Arm® Cortex®-M

32-bit Microcontroller

NuMicro® Family
Nano100 Series
Technical Reference Manual

For additional information or questions, please contact: Nuvoton Technology Corporation.

www.nuvoton.com

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1 GENERAL DESCRIPTION

The Nano100 series ultra-low power 32-bit microcontroller is embedded with ARM® Cortex®-M0 core operated at a wide voltage range from 1.8 V to 3.6 V and runs up to 42 MHz frequency with 32K/64K/128K bytes embedded Flash and 8K/16K-byte embedded SRAM. Integrating LCD 4x40 or 6x38 (COM/Segment), USB 2.0 full-speed function, RTC, 12-bit SAR ADC, 12-bit DAC and provides high performance connectivity peripheral interfaces such as UART, SPI, I^2C, I^2S, GPIOs, EBI (External Bus Interface) for external memory-mapped device access and ISO-7816-3 for Smart card, the Nano100 series supports Brown-out Detector, Power-down mode with RAM retention and fast wake-up via many peripheral interfaces.

The Nano100 series provides low power voltage, low power consumption, low standby current, high integration peripherals, high-efficiency operation, fast wake-up function and the lowest cost 32-bit microcontrollers. The Nano100 series is suitable for a wide range of battery device applications such as:

- Portable Data Collector
- Portable Medical Monitor
- Portable RFID Reader
- Portable Barcode Scanner
- Security Alarm System
- System Supervisors
- Power Metering
- USB Accessories
- Smart Card Reader
- Wireless Game Control Device
- IPTV Remote Smart Keyboard
- Wireless Sensors Node Device (WSN)
- Wireless RF4CE Remote Control
- Wireless Audio
- Wireless Automatic Meter Reader (AMR)
- Electronic Toll Collection (ETC)

The Nano100 Base line, an ultra-low power 32-bit microcontroller with the embedded ARM® Cortex®-M0 core, operates at wide voltage range from 1.8V to 3.6V and runs up to 42 MHz frequency with 32K/64K/128K bytes embedded flash and 8K/16K bytes embedded SRAM. It integrates RTC, 12-channels 12-bit SAR ADC, 2-channels 12-bit DAC and provides high performance connectivity peripheral interfaces such as 2xUART, 3xSPI, $2xI^2C$, I^2S, GPIOs, EBI (External Bus Interface) for external memory-mapped device access and 3xISO-7816-3 for Smart card. The Nano100 Base line supports Brown-out Detector, Power-down mode with RAM retention and fast wake-up via many peripheral interfaces.

The Nano110 LCD line, an ultra-low power 32-bit microcontroller with the embedded ARM® Cortex®-M0 core, operates at wide voltage range from 1.8V to 3.6V and runs up to 42 MHz frequency with 32K/64K/128K bytes embedded flash and 8K/16K bytes embedded SRAM. It integrates LCD 4x40 or 6x38 (COM/Segment). RTC, 12-channels 12-bit SAR ADC, 2-channels 12-bit DAC and provides high performance connectivity peripheral interfaces such as 2xUART, 2xSPI, $2xI^2C$, I^2S, GPIOs, EBI (External Bus Interface) for external memory-mapped device access and 3xISO-7816-3 for Smart card. The Nano110 LCD line supports Brown-out Detector, Power-down mode with RAM retention and fast wake-up via many peripheral interfaces.

The Nano120 USB Connectivity line, an ultra-low power 32-bit microcontroller with the embedded

ARM® Cortex®-M0 core, operates at wide voltage range from 1.8V to 3.6V and runs up to 42 MHz frequency with 32K/64K/128K bytes embedded flash and 8K/16K bytes embedded SRAM. It integrates USB 2.0 full-speed device function, RTC, 12-channels12-bit SAR ADC, 2-channels 12-bit DAC and provides high performance connectivity peripheral interfaces such as 2xUART, 3xSPI, 2xI2C, I2S, GPIOs, EBI (External Bus Interface) for external memory-mapped device access and 3xISO-7816-3 for Smart card. The Nano120 USB Connectivity line supports Brown-out Detector, Power-down mode with RAM retention and fast wake-up via many peripheral interfaces.

The Nano130 Advanced line, an ultra-low power 32-bit microcontroller with the embedded ARM® Cortex®-M0 core, operates at wide voltage range from 1.8V to 3.6V and runs up to 42 MHz frequency with 32K/64K/128K bytes embedded flash and 8K/16K bytes embedded SRAM. It integrated LCD 4x40 or 6x38 (COM/Segment), USB 2.0 full-speed device function, RTC, 8-channels 12-bit SAR ADC, 2-channels 12-bit DAC and provides high performance connectivity peripheral interfaces such as 2xUART, 2xSPI, 2xI^2C, I^2S, GPIOs, EBI (External Bus Interface) for external memory-mapped device access and 3xISO-7816-3 for Smart card. The Nano130 Advanced line supports Brown-out Detector, Power-down mode with RAM retention and fast wake-up via many peripheral interfaces.

Product Line	UART	SPI	I^2C	I^2S	USB	LCD	ADC	DAC	RTC	EBI	SC	Timer
Nano100	●	●	●	●			●	●	●	●	●	●
Nano110	●	●	●	●		●	●	●	●	●	●	●
Nano120	●	●	●	●	●		●	●	●	●	●	●
Nano130	●	●	●	●	●	●	●	●	●	●	●	●

Table 2.1-1 Connectivity Support Table

2 FEATURES

The equipped features are dependent on the product line and their sub products.

2.1 Nano100 Features – Base Line

- Core
  - ARM® Cortex®-M0 core running up to 42 MHz
  - One 24-bit system timer
  - Supports Low Power Sleep mode
  - Single-cycle 32-bit hardware multiplier
  - NVIC for the 32 interrupt inputs, each with 4-levels of priority
  - Serial Wire Debug supports with 2 watchpoints/4 breakpoints
- Brown-out
  - Built-in 2.5V/2.0V/1.7V BOD for wide operating voltage range operation
- Flash EPROM Memory
  - Runs up to 42 MHz with zero wait state for discontinuous address read access
  - 64K/32K/123K bytes application program memory (APROM)
  - 4 KB in system programming (ISP) loader program memory (LDROM)
  - Programmable data flash start address and memory size with 512 bytes page erase unit
  - In System Program (ISP)/In Application Program (IAP) to update on-chip Flash EPROM
- SRAM Memory
  - 16K/8K bytes embedded SRAM
  - Supports DMA mode
- DMA: Supports 8 channels: one VDMA channel, 6 PDMA channels and one CRC channel
  - VDMA
    - Memory-to-memory transfer
    - Supports block transfer with stride
    - Supports word/half-word/byte boundary address
    - Supports address direction: increment and decrement
  - PDMA
    - Peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfer
    - Supports word boundary address
    - Supports word alignment transfer length in memory-to-memory mode
    - Supports word/half-word/byte alignment transfer length in peripheral-to-memory and memory-to-peripheral mode
    - Supports word/half-word/byte transfer data width from/to peripheral
    - Supports address direction: increment, fixed, and wrap around
  - CRC
    - Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32

- CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
- CRC-8: $X^8 + X^2 + X + 1$
- CRC-16: $X^{16} + X^{15} + X^2 + 1$
- CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

- Clock Control
  - Flexible selection for different applications
  - Built-in 12 MHz OSC, can be trimmed to 0.25 % deviation within all temperature range when turning on auto-trim function (system must have external 32.768 kHz crystal input) otherwise 12 MHz OSC has 2 % deviation within all temperarure range.
  - Low power 10 kHz OSC for watchdog and low power system operation
  - Supports one PLL, up to 120 MHz, for high performance system operation and USB application (48 MHz).
  - External 4~24 MHz crystal input for precise timing operation
  - External 32.768 kHz crystal input for RTC function and low power system operation
- GPIO
  - Three I/O modes:
    - Push-Pull output
    - Open-Drain output
    - Input only with high impendence
  - All inputs with Schmitt trigger
  - I/O pin configured as interrupt source with edge/level setting
  - Supports High Driver and High Sink I/O mode
  - Supports input 5V tolerance, except PA.0 ~ PA.7, PD.0 ~ PD.1 and PC.6 ~ PC.7
- Timer
  - Supports 4 sets of 32-bit timers, each with 24-bit up-counting timer and one 8-bit pre-scale counter
  - Independent Clock Source for each timer
  - Provides one-shot,periodic, output toggle and continuous operation modes
  - Internal trigger event to ADC, DAC and PDMA
  - Supports PDMA mode
  - Wake system up from Power-down mode
- Watchdog Timer
  - Clock Source from LIRC (Internal 10 kHz Low Speed Oscillator Clock)
  - Selectable time-out period from 1.6 ms ~ 26 sec (depending on clock source)
  - Interrupt or reset selectable when watchdog time-out
  - Wake system up from Power-down mode
- Window Watchdog Timer (WWDT)
  - 6-bit down counter and 6-bit compare value to make the window period flexible

- Selectable WWDT clock pre-scale counter to make WWDT time-out interval variable.

- RTC
  - Supports software compensation by setting frequency compensate register (FCR)
  - Supports RTC counter (second, minute, hour) and calendar counter (day, month, year)
  - Supports Alarm registers (second, minute, hour, day, month, year)
  - Selectable 12-hour or 24-hour mode
  - Automatic leap year recognition
  - Supports periodic time tick interrupt with 8 periodic options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
  - Wake system up from Power-down mode
  - Supports 80 bytes spare registers and a snoop pin to clear the content of these spare registers
- PWM/Capture
  - Supports 2 PWM modules, each has two 16-bit PWM generators
  - Provides eight PWM outputs or four complementary paired PWM outputs
  - Each PWM generator equipped with one clock divider, one 8-bit prescaler, two clock selectors, and one Dead-zone generator for complementary paired PWM
  - (Shared with PWM timers) with eight 16-bit digital capture timers provides eight rising/ falling/both capture inputs.
  - Supports One-shot and Continuous mode
  - Supports Capture interrupt
- UART
  - Up to two 16-byte FIFO UART controllers
  - UART ports with flow control (TX, RX, CTSn and RTSn)
  - Supports IrDA (SIR) function
  - Supports LIN function
  - Supports RS-485 9 bit mode and direction control.
  - Programmable baud rate generator
  - Supports PDMA mode
  - Wake system up from Power-down mode
- SPI
  - Up to three sets of SPI controller
  - Master up to 32 MHz, and Slave up to 16 MHz
  - Supports SPI/MICROWIRE Master/Slave mode
  - Full duplex synchronous serial data transfer
  - Variable length of transfer data from 4 to 32 bits
  - MSB or LSB first data transfer
  - RX and TX on both rising or falling edge of serial clock independently
  - Two slave/device select lines when SPI controller is used as the master, and 1 slave/device

select line when SPI controller is used as the slave
- Supports byte suspend mode in 32-bit transmission
- Supports two channel PDMA requests, one for transmit and another for receive
- Supports three wire mode, no slave select signal, bi-direction interface
- Wake system up from Power-down mode

- I^2C
  - Up to two sets of I^2C device
  - Master/Slave up to 1 Mbit/s
  - Bi-directional data transfer between masters and slaves
  - Multi-master bus (no central master)
  - Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
  - Serial clock synchronization allows devices with different bit rates to communicate via one serial bus
  - Serial clock synchronization used as a handshake mechanism to suspend and resume serial transfer
  - Built-in 14-bit time-out counter requesting the I^2C interrupt if the I^2C bus hangs up and timer-out counter overflows
  - Programmable clocks allowing for versatile rate control
  - Supports 7-bit addressing mode
  - Supports multiple address recognition (four slave addresses with mask option)
- I^2S
  - Interface with external audio CODEC
  - Operated as either Master or Slave mode
  - Capable of handling 8, 16, 24 and 32 bit word sizes
  - Supports Mono and stereo audio data
  - Supports I^2S and MSB justified data format
  - Provides two 8 word FIFO data buffers: one for transmitting and the other for receiving
  - Generates interrupt requests when buffer levels cross a programmable boundary
  - Supports two PDMA requests: one for transmitting and the other for receiving
- ADC
  - 12-bit SAR ADC up to 2MSPS conversion rate
  - Up to 12-ch single-ended input from external pin (PA.0 ~ PA.7 and PD.0 ~ PD.3)
  - Six internal channels from DAC0, DAC1, internal reference voltage (Int_VREF), Temperature sensor, AV_{DD}, and AV_{SS}.
  - Supports three reference voltage sources from V_{REF} pin, internal reference voltage (Int_VREF), and AV_{DD}.
  - Supports Single Scan, Single Cycle Scan, and Continuous Scan mode
  - Each channel with individual result register

- Only scan on enabled channels
- Threshold voltage detection (comparator function)
- Conversion started by software programming or external input
- Supports PDMA mode
- Supports up to four timer time-out events (TMR0, TMR1, TMR2 and TMR3) to enable ADC

- DAC
  - 12-bit monotonic output with 400K conversion rate
  - Supports three reference voltage sources from V_{REF} pin, internal reference voltage (Int_VREF), and AV_{DD}.
  - Synchronized update capability for two DACs (group function)
  - Supports up to four timer time-out events (TMR0, TMR1, TMR2 and TMR3), software or PDMA to trigger DAC to conversion
- SmartCard (SC)
  - Compliant to ISO-7816-3 T=0, T=1
  - Supports up to three ISO-7816-3 ports
  - Separates receive/transmit 4 bytes entry FIFO for data payloads
  - Programmable transmission clock frequency
  - Programmable receiver buffer trigger level
  - Programmable guard time selection (11 ETU ~ 266 ETU)
  - A 24-bit and two 8-bit time-out counters for Answer to Reset (ATR) and waiting times processing
  - Supports auto inverse convention function
  - Supports stop clock level and clock stop (clock keep) function
  - Supports transmitter and receiver error retry and error limit function
  - Supports hardware activation sequence process
  - Supports hardware warm reset sequence process
  - Supports hardware deactivation sequence process
  - Supports hardware auto deactivation sequence when detect the card is removal
  - Supports UART mode (Half Duplex)
- EBI (External bus interface) support
  - Accessible space: 64 KB in 8-bit mode or 128 KB in 16-bit mode
  - Supports 8bit/16bit data width
  - Supports byte write in 16-bit Data Width mode
- One built-in temperature sensor with 1℃ resolution
- 96-bit unique ID
- 128-bit unique customer ID
- Operating Temperature: -40℃~85℃
- Packages:

- All Green package (RoHS)
- LQFP 128-pin(14x14) / 64-pin(7x7) / 48-pin(7x7) / QFN 48-pin(7x7)

2.2 Nano110 Features – LCD Line

- Core
  - ARM® Cortex®-M0 core running up to 42 MHz
  - One 24-bit system timer
  - Supports Low Power Sleep mode
  - Single-cycle 32-bit hardware multiplier
  - NVIC for the 32 interrupt inputs, each with 4-levels of priority
  - Serial Wire Debug supports with 2 watchpoints/4 breakpoints
- Brown-out
  - Built-in 2.5V/2.0V/1.7V BOD for wide operating voltage range operation
- Flash EPROM Memory
  - Runs up to 42 MHz with zero wait state for discontinuous address read access.
  - 64K/32K/123K bytes application program memory (APROM)
  - 4 KB In System Programming (ISP) loader program memory (LDROM)
  - Programmable data flash start address and memory size with 512 bytes page erase unit
  - In System Program (ISP)/In Application Program (IAP) to update on chip Flash EPROM
- SRAM Memory
  - 16K/8K bytes embedded SRAM
  - Supports DMA mode
- DMA : Supports 8 channels: one VDMA channel,6 PDMA channels, and one CRC channel
  - VDMA
    - Memory-to-memory transfer
    - Supports block transfer with stride
    - Supports word/half-word/byte boundary address
    - Supports address direction: increment and decrement
  - PDMA
    - Peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfer
    - Supports word boundary address
    - Supports word alignment transfer length in memory-to-memory mode
    - Supports word/half-word/byte alignment transfer length in peripheral-to-memory and memory-to-peripheral mode
    - Supports word/half-word/byte transfer data width from/to peripheral
    - Supports address direction: increment, fixed, and wrap around
  - CRC
    - Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32
      - CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
      - CRC-8: $X^8 + X^2 + X + 1$

- CRC-16: $X^{16} + X^{15} + X^2 + 1$
- CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

- Clock Control
  - Flexible selection for different applications
  - Built-in 12 MHz OSC, can be trimmed to 0.25% deviation within all temperature range when turning on auto-trim function (system must have external 32.768 kHz crystal input) otherwise 12 MHz OSC has 2 % deviation within all temperarure range.
  - Low power 10 kHz OSC for watchdog and low power system operation
  - Supports one PLL, up to 120 MHz, for high performance system operation and USB application (48 MHz).
  - External 4~24 MHz crystal input for precise timing operation
  - External 32.768 kHz crystal input for RTC function and low power system operation
- GPIO
  - Three I/O modes:
    - Push-Pull output
    - Open-Drain output
    - Input only with high impendence
  - All inputs with Schmitt trigger
  - I/O pin configured as interrupt source with edge/level setting
  - Supports High Driver and High Sink I/O mode
  - Supports input 5V tolerance, except PA.0 ~ PA.7, PD.0 ~ PD.1 and PC.6 ~ PC.7)
- Timer
  - Supports 4 sets of 32-bit timers, each with 24-bit up-timer and one 8-bit pre-scale counter
  - Independent Clock Source for each timer
  - Provides one-shot,periodic, output toggle and continuous operation modes
  - Internal trigger event to ADC, DAC and PDMA module
  - Supports PDMA mode
  - Wake system up from Power-down mode
- Watchdog Timer
  - Clock Source from LIRC (Internal 10 kHz Low Speed Oscillator Clock)
  - Selectable time-out period from 1.6 ms ~ 26 sec (depending on clock source)
  - Interrupt or reset selectable when watchdog time-out
  - Wake system up from Power-down mode
- Window Watchdog Timer(WWDT)
  - 6-bit down counter and 6-bit compare value to make the window period flexible
  - Selectable WWDT clock pre-scale counter to make WWDT time-out interval variable.
- RTC
  - Supports software compensation by setting frequency compensate register (FCR)

- Supports RTC counter (second, minute, hour) and calendar counter (day, month, year)
- Supports Alarm registers (second, minute, hour, day, month, year)
- Selectable 12-hour or 24-hour mode
- Automatic leap year recognition
- Supports periodic time tick interrupt with 8 periodic options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
- Wake system up from Power-down mode
- Supports 80 bytes spare registers and a snoop pin to clear the content of these spare registers

● PWM/Capture
  - Supports 2 PWM modules, each has two 16-bit PWM generators
  - Provides eight PWM outputs or four complementary paired PWM outputs
  - Each PWM generator equipped with one clock divider, one 8-bit prescaler, two clock selectors, and one Dead-zone generator for complementary paired PWM
  - (Shared with PWM timers) with eight 16-bit digital capture timers provides eight rising/ falling/both capture inputs.
  - Supports Capture interrupt

● UART
  - Up to two 16-byte FIFO UART controllers
  - UART ports with flow control (TX, RX, CTSn and RTSn)
  - Supports IrDA (SIR) function
  - Supports LIN function
  - Supports RS-485 9 bit mode and direction control (Low Density Only)
  - Programmable baud rate generator
  - Supports PDMA mode
  - Wake system up from Power-down mode

● SPI
  - Up to three sets of SPI controller
  - Master up to 32 MHz, and Slave up to 16 MHz
  - Supports SPI/MICROWIRE Master/Slave mode
  - Full duplex synchronous serial data transfer
  - Variable length of transfer data from 4 to 32 bits
  - MSB or LSB first data transfer
  - RX and TX on both rising or falling edge of serial clock independently
  - Two slave/device select lines when SPI controller is as the master, and 1 slave/device select line when SPI controller is as the slave
  - Supports byte suspend mode in 32-bit transmission
  - Supports two channel PDMA requests, one for transmit and another for receive
  - Supports three wire mode, no slave select signal, bi-direction interface

- Wake system up from Power-down mode

- I^2C
  - Up to two sets of I^2C device
  - Master/Slave up to 1Mbit/s
  - Bidirectional data transfer between masters and slaves
  - Multi-master bus (no central master)
  - Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
  - Serial clock synchronization allowing devices with different bit rates to communicate via one serial bus
  - Serial clock synchronization used as a handshake mechanism to suspend and resume serial transfer
  - Built-in 14-bit time-out counter requestING the I^2C interrupt if the I^2C bus hangs up and timer-out counter overflows
  - Programmable clocks allow versatile rate control
  - Supports 7-bit addressing mode
  - Supports multiple address recognition (four slave address with mask option)
- I^2S
  - Interface with external audio CODEC
  - Operated as either Master or Slave mode
  - Capable of handling 8, 16, 24 and 32 bit word sizes
  - Supports Mono and stereo audio data
  - Supports I^2S and MSB justified data format
  - Provides two 8 word FIFO data buffers: one for transmitting and the other for receiving
  - Generates interrupt requests when buffer levels cross a programmable boundary
  - Supports two PDMA requests: one for transmitting and the other for receiving
- ADC
  - 12-bit SAR ADC up to 2 MSPS conversion rate
  - Up to 12-ch single-ended input from external pin (PA.0 ~ PA.7 and PD.0 ~ PD.3)
  - Six internal channels from DAC0, DAC1, internal reference voltage (Int_VREF), Temperature sensor, AV_{DD}, and AV_{SS}
  - Supports three reference voltage sources from V_{REF} pin, internal reference voltage (Int_VREF), and AV_{DD}.
  - Single scan/single cycle scan/continuous scan
  - Each channel with individual result register
  - Only scan on enabled channels
  - Threshold voltage detection (comparator function)
  - Conversion start by software programming or external input
  - Supports PDMA mode

- Supports up to four timer time-out events (TMR0, TMR1, TMR2, and TMR3) to enable ADC

- DAC
  - 12-bit monotonic output with 400K conversion rate
  - Supports three reference voltage sources from V_{REF} pin, internal reference voltage (Int_VREF), and AV_{DD}.
  - Synchronized update capability for two DACs (group function)
  - Supports up to four timer time-out events (TMR0, TMR1, TMR2 and TMR3), software or PDMA to trigger DAC to conversion
- SmartCard (SC)
  - Compliant to ISO-7816-3 T=0, T=1
  - Supports up to three ISO-7816-3 ports
  - Separates receive / transmit 4 bytes entry FIFO for data payloads
  - Programmable transmission clock frequency
  - Programmable receiver buffer trigger level
  - Programmable guard time selection (11 ETU ~ 266 ETU)
  - A 24-bit and two 8-bit time-out counter for Answer to Reset (ATR) and waiting times processing
  - Supports auto inverse convention function
  - Supports stop clock level and clock stop (clock keep) function
  - Supports transmitter and receiver error retry and error limit function
  - Supports hardware activation sequence process
  - Supports hardware warm reset sequence process
  - Supports hardware deactivation sequence process
  - Supports hardware auto deactivation sequence when detect the card is removal
  - Supports UART mode (Half Duplex)
- LCD
  - LCD driver for up to 4 COM x 40 SEG or 6 COM x 38 SEG
  - Supports Static,1/2 bias and 1/3 bias voltage
  - Four display modes; Static, 1/2 duty, 1/3 duty,1/4 duty, 1/5 duty and 1/6 duty.
  - Selectable LCD frequency by frequency divider
  - Configurable frame frequency
  - Internal Charge pump, adjustable contrast adjustment
  - Configurable Charge pump frequency
  - Blinking capability
  - Supports R-type/C-type method
  - LCD frame interrupt
- One built-in temperature sensor with 1℃ resolution
- 96-bit unique ID

- 128-bit unique customer ID
- Operating Temperature: -40℃~85℃
- Packages:
  - All Green package (RoHS)
  - LQFP 128-pin(14x14) / 64-pin(10x10) / 64-pin(7x7)

2.3 Nano120 Features – USB Connectivity Line

- Core
  - ARM® Cortex®-M0 core running up to 42 MHz
  - One 24-bit system timer
  - Supports Low Power Sleep mode
  - Single-cycle 32-bit hardware multiplier
  - NVIC for the 32 interrupt inputs, each with 4-levels of priority
  - Serial Wire Debug supports with 2 watchpoints/4 breakpoints
- Brown-out
  - Built-in 2.5V/2.0V/1.7V BOD for wide operating voltage range operation
- Flash EPROM Memory
  - Runs up to 42 MHz with zero wait state for discontinuous address read access.
  - 64K/32K/123K bytes application program memory (APROM)
  - 4KB in system programming (ISP) loader program memory (LDROM)
  - Programmable data flash start address and memory size with 512 bytes page erase unit
  - In System Program (ISP)/In Application Program (IAP) to update on chip Flash EPROM
- SRAM Memory
  - 16K/8K bytes embedded SRAM
  - Supports PDMA mode
- DMA: Support 8 channels: one VDMA channel, 6 PDMA channels, and one CRC channel
  - VDMA
    - Memory-to-memory transfer
    - Supports block transfer with stride
    - Supports word/half-word/byte boundary address
    - Supports address direction: increment and decrement
  - PDMA
    - Peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfer
    - Supports word boundary address
    - Supports word alignment transfer length in memory-to-memory mode
    - Supports word/half-word/byte alignment transfer length in peripheral-to-memory and memory-to-peripheral mode
    - Supports word/half-word/byte transfer data width from/to peripheral
    - Supports address: increment, fixed, and wrap around
  - CRC
    - Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32
      - CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
      - CRC-8: $X^8 + X^2 + X + 1$

- CRC-16: $X^{16} + X^{15} + X^2 + 1$
- CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

- Clock Control
  - Flexible selection for different applications
  - Built-in 12MHz OSC, can be trimmed to 0.25% deviation within all temperature range when turning on auto-trim function (system must have external 32.768 kHz crystal input) otherwise 12 MHz OSC has 2 % deviation within all temperarure range
  - Low power 10 kHz OSC for watchdog and low power system operatin
  - Supports one PLL, up to 120 MHz, for high performance system operation and USB application (48 MHz).
  - External 4~24 MHz crystal input for precise timing operation
  - External 32.768 kHz crystal input for RTC function and low power system operation
- GPIO
  - Three I/O modes:
    - Push-Pull output
    - Open-Drain output
    - Input only with high impendence
  - All inputs with Schmitt trigger
  - I/O pin can be configured as interrupt source with edge/level setting
  - High driver and high sink IO mode support
  - Supports input 5V tolerance (except ADC and DAC shared pins)
- Timer
  - Supports 4 sets of 32-bit timers, each with 24-bit up-timer and one 8-bit pre-scale counter
  - Independent Clock Source for each timer
  - Provides one-shot,periodic, output toggle and continuous operation modes
  - Internal trigger event to ADC, DAC and PDMA module
  - Supports PDMA mode
  - Wake system up from Power-down mode
- Watchdog Timer
  - Clock Source from LIRC. (Internal 10 kHz Low Speed Oscillator Clock)
  - Selectable time-out period from 1.6 ms ~ 26 sec (depending on clock source)
  - Interrupt or reset selectable on watchdog time-out
  - Wake system up from Power-down mode
- Window Watchdog Timer(WWDT)
  - 6-bit down counter and 6-bit compare value to make the window period flexible
  - Selectable WWDT clock pre-scale counter to make WWDT time-out interval variable.
- RTC
  - Supports software compensation by setting frequency compensate register (FCR)

- Supports RTC counter (second, minute, hour) and calendar counter (day, month, year)
- Supports Alarm registers (second, minute, hour, day, month, year)
- Selectable 12-hour or 24-hour mode
- Automatic leap year recognition
- Supports periodic time tick interrupt with 8 periodic options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
- Wake system up from Power-down or Idle mode
- Support 80 bytes spare registers and a snoop pin to clear the content of these spare registers

- PWM/Capture
  - Supports 2 PWM module, each has two 16-bit PWM generators
  - Provide eight PWM outputs or four complementary paired PWM outputs
  - Each PWM generator equipped with one clock divider, one 8-bit prescaler, two clock selectors, and one Dead-Zone generator for complementary paired PWM
  - (Shared with PWM timers) with eight 16-bit digital capture timers provides eight rising/ falling/both capture inputs.
  - Supports one shot and continuous mode
  - Supports Capture interrupt
- UART
  - Up to two 16-byte FIFO UART controllers
  - UART ports with flow control (TX, RX, CTSn and RTSn)
  - Supports IrDA (SIR) function
  - Supports LIN function
  - Supports RS-485 9 bit mode and direction control. (Low Density Only)
  - Programmable baud rate generator
  - Supports PDMA mode
  - Wake system up from Power-down mode
- SPI
  - Up to three sets of SPI controller
  - Master up to 32 MHz, and Slave up to 16 MHz
  - Supports SPI/MICROWIRE Master/Slave mode
  - Full duplex synchronous serial data transfer
  - Variable length of transfer data from 4 to 32 bits
  - MSB or LSB first data transfer
  - RX and TX on both rising or falling edge of serial clock independently
  - Two slave/device select lines when SPI controller is as the master, and 1 slave/device select line when SPI controller is as the slave
  - Supports byte suspend mode in 32-bit transmission
  - Supports two channel PDMA requests, one for transmit and another for receive

- Supports three wire, no slave select signal, bi-direction interface
- Wake system up from Power-down mode

- I^2C
  - Up to two sets of I^2C device
  - Master/Slave up to 1Mbit/s
  - Bi-directional data transfer between masters and slaves
  - Multi-master bus (no central master)
  - Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
  - Serial clock synchronization allowing devices with different bit rates to communicate via one serial bus
  - Serial clock synchronization used as a handshake mechanism to suspend and resume serial transfer
  - Built-in 14-bit time-out counter requesting the I^2C interrupt if the I^2C bus hangs up and timer-out counter overflows
  - Programmable clocks allow versatile rate control
  - Supports 7-bit addressing mode
  - Supports multiple address recognition (four slave addresses with mask option)
- I^2S
  - Interface with external audio CODEC
  - Operated as either Master or Slave mode
  - Capable of handling 8, 16, 24 and 32 bit word sizes
  - Supports Mono and stereo audio data
  - Supports I^2S and MSB justified data format
  - Provides two 8 word FIFO data buffers: one for transmitting and the other for receiving
  - Generates interrupt requests when buffer levels cross a programmable boundary
  - Supports two PDMA requests: one for transmitting and the other for receiving
- ADC
  - 12-bit SAR ADC up to 2MSPS conversion rate
  - Up to 12-ch single-ended input from external pin (PA.0 ~ PA.7 and PD.0 ~ PD.3).
  - Six internal channels from DAC0, DAC1, internal reference voltage (Int_VREF), Temperature sensor, AV_{DD}, and AV_{SS}.
  - Supports three reference voltage sources from V_{REF} pin, internal reference voltage (Int_VREF), and AV_{DD}
  - Supports single scan, single cycle scan, and continuous scan modes
  - Each channel with individual result register
  - Only scan on enabled channels
  - Threshold voltage detection (comparator function)
  - Conversion start by software programming or external input

- Supports PDMA mode
- Supports up to four timer time-out events (TMR0, TMR1, TMR2 and TMR3) to enable ADC

- DAC
  - 12-bit monotonic output with 400K conversion rate
  - Supports three reference voltage sources from V_{REF} pin, internal reference voltage (Int_VREF), and AV_{DD}.
  - Synchronized update capability for two DACs (group function)
  - Supports up to four timer time-out event (TMR0, TMR1, TMR2 and TMR3), software or PDMA to trigger DAC to conversion
- SmartCard (SC)
  - Compliant to ISO-7816-3 T=0, T=1
  - Supports up to three ISO-7816-3 ports
  - Separates receive / transmit 4 bytes entry FIFO for data payloads
  - Programmable transmission clock frequency
  - Programmable receiver buffer trigger level
  - Programmable guard time selection (11 ETU ~ 266 ETU)
  - A 24-bit and two 8-bit time-out counter for Answer to Reset (ATR) and waiting times processing
  - Supports auto inverse convention function
  - Supports stop clock level and clock stop (clock keep) function
  - Supports transmitter and receiver error retry and error limit function
  - Supports hardware activation sequence process
  - Supports hardware warm reset sequence process
  - Supports hardware deactivation sequence process
  - Supports hardware auto deactivation sequence when detect the card is removal
  - Supports UART mode (Half Duplex)
- USB 2.0 Full-Speed Device
  - One set of USB 2.0 FS Device 12 Mbps
  - On-chip USB Transceiver
  - Provides 1 interrupt source with 4 interrupt events
  - Supports Control, Bulk In/Out, Interrupt and Isochronous transfers
  - Auto suspend function when no bus signaling for 3 ms
  - Provides 8 programmable endpoints
  - Includes 512 Bytes internal SRAM as USB buffer
  - Provides remote wake-up capability
- EBI (External bus interface) support
  - Accessible space: 64 KB in 8-bit mode or 128 KB in 16-bit mode
  - Supports 8bit/16bit data width

- Supports byte write in 16-bit Data Width mode
- One built-in temperature sensor with 1℃ resolution
- 96-bit unique ID
- 128-bit unique customer ID
- Operating Temperature: -40℃~85℃
- Packages:
  - All Green package (RoHS)
  - LQFP 128-pin(14x14) / 64-pin(7x7) / 48-pin(7x7)

2.4 Nano130 Features – Advanced Line

- Core
  - ARM® Cortex®-M0 core running up to 42 MHz
  - One 24-bit system timer
  - Supports Low Power Sleep mode
  - Single-cycle 32-bit hardware multiplier
  - NVIC for the 32 interrupt inputs, each with 4-levels of priority
  - Serial Wire Debug supports with 2 watchpoints/4 breakpoints
- Brown-out
  - Built-in 2.5V/2.0V/1.7V BOD for wide operating voltage range operation
- Flash EPROM Memory
  - Runs up to 42 MHz with zero wait state for discontinuous address read access.
  - 64K/32K/123K bytes application program memory (APROM)
  - 4KB in system programming (ISP) loader program memory (LDROM)
  - Programmable data flash start address and memory size with 512 bytes page erase unit
  - In System Program (ISP)/In Application Program (IAP) to update on chip Flash EPROM
- SRAM Memory
  - 16K/8K bytes embedded SRAM
  - Supports DMA mode
- DMA : Supports 8 channels: one VDMA channel,6 PDMA channels, and one CRC egiste
  - VDMA
    - Memory-to-memory transfer
    - Supports block transfer with stride
    - Supports word/half-word/byte boundary address
    - Supports address direction: increment and decrement
  - PDMA
    - Peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfer
    - Supports word boundary address
    - Supports word alignment transfer length in memory-to-memory mode
    - Supports word/half-word/byte alignment transfer length in peripheral-to-memory and memory-to-peripheral mode
    - Supports word/half-word/byte transfer data width from/to peripheral
    - Supports address direction: increment, fixed, and wrap around
  - CRC
    - Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32
      - CRC-CCITT: $X^{16} + X^{12} + X^5 + 1$
      - CRC-8: $X^8 + X^2 + X + 1$

- $CRC-16: X^{16} + X^{15} + X^2 + 1$
- $CRC-32: X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

- Clock Control
  - Flexible selection for different applications
  - Built-in 12MHz OSC, can be trimmed to 0.25% deviation within all temperature range when turning on auto-trim function (system must have external 32.768 kHz crystal input) otherwise 12 MHz OSC has 2 % deviation within all temperarure range.
  - Low power 10 kHz OSC for watchdog and low power system operation
  - Supports one PLL, up to 120 MHz, for high performance system operation and USB application (48 MHz).
  - External 4~24 MHz crystal input for precise timing operation
  - External 32.768 kHz crystal input for RTC function and low power system operation
- GPIO
  - Three I/O modes:
    - Push-Pull output
    - Open-Drain output
    - Input only with high impendence
  - All inputs with Schmitt trigger
  - I/O pin configured as interrupt source with edge/level setting
  - Supports High Driver and High Sink I/O mode
  - Supports input 5V tolerance (except ADC and DAC shared pins)
- Timer
  - Supports 4 sets of 32-bit timers with 24-bit up-timer and one 8-bit pre-scale counter
  - Independent Clock Source for each timer
  - Provides one-shot,periodic, output toggle and continuous operation modes
  - Supports internal trigger event to ADC, DAC and PDMA module
  - Wake system up from Power-down mode
- Watchdog Timer
  - Clock Source is from LIRC. (Internal 10 kHz Low Speed Oscillator Clock)
  - Selectable time-out period from 1.6ms ~ 26sec (depends on clock source)
  - Interrupt or reset selectable on watchdog time-out
  - WDT can wake system up from Power-down mode
- Window Watchdog Timer(WWDT)
  - 6-bit down counter and 6-bit compare value to make the window period flexible
  - Selectable WWDT clock pre-scale counter to make WWDT time-out interval variable.
- RTC
  - Supports software compensation by setting frequency compensate register (FCR)
  - Supports RTC counter (second, minute, hour) and calendar counter (day, month, year)

- Supports Alarm registers (second, minute, hour, day, month, year)
- Selectable 12-hour or 24-hour mode
- Automatic leap year recognition
- Supports periodic time tick interrupt with 8 periodic options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
- Wake system up from Power-down or Idle mode
- Supports 80 bytes spare registers and a snoop pin to clear the content of these spare registers

- PWM/Capture
  - Supports 2 PWM module, each with two 16-bit PWM generators
  - Provides eight PWM outputs or four complementary paired PWM outputs
  - Each PWM generator equipped with one clock divider, one 8-bit prescaler, two clock selectors, and one Dead-Zone generator for complementary paired PWM
  - (Shared with PWM timers) with eight 16-bit digital capture timers provides eight rising/ falling/both capture inputs.
  - Supports Capture interrupt
- UART
  - Up to two 16-byte FIFO UART controllers
  - UART ports with flow control (TX, RX, CTSn and RTSn)
  - Supports IrDA (SIR) function
  - Supports LIN function
  - Supports RS-485 9 bit mode and direction control (Low Density Only)
  - Programmable baud rate generator
  - Supports PDMA mode
  - Wake system up from Power-down or Idle mode
- SPI
  - Up to 3 sets of SPI controller
  - Master up to 32 MHz, and Slave up to 16 MHz
  - Supports SPI/MICROWIRE Master/Slave mode
  - Full duplex synchronous serial data transfer
  - Variable length of transfer data from 4 to 32 bits
  - MSB or LSB first data transfer
  - RX and TX on both rising or falling edge of serial clock independently
  - Two slave/device select lines when used as the master, and 1 slave/device select line when used as the slave
  - Supports byte suspend mode in 32-bit transmission
  - Supports two channel PDMA request, one for transmit and another for receive
  - Supports three wire, no slave select signal, bi-direction interface
  - Wake system up from Power-down or Idle mode

- I^2C
  - Up to two sets of I^2C device
  - Master/Slave up to 1Mbit/s
  - Bi-directional data transfer between masters and slaves
  - Multi-master bus (no central master)
  - Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
  - Serial clock synchronization allowing devices with different bit rates to communicate via one serial bus
  - Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer
  - Built-in 14-bit time-out counter will request the I^2C interrupt if the I^2C bus hangs up and timer-out counter overflows
  - Programmable clocks allowing for versatile rate control
  - Supports 7-bit addressing mode
  - Supports multiple address recognition (four slave addresses with mask option)
- I^2S
  - Interface with external audio CODEC
  - Operate as either Master or Slave mode
  - Capable of handling 8, 16, 24 and 32 bit word sizes
  - Supports Mono and stereo audio data
  - Supports I^2S and MSB justified data format
  - Provides two 8 word FIFO data buffers: one for transmitting and the other for receiving
  - Generates interrupt requests when buffer levels cross a programmable boundary
  - Supports two PDMA requests: one for transmitting and the other for receiving
- ADC
  - 12-bit SAR ADC up to 2MSPS conversion rate
  - Up to 12-ch single-ended input from external pin (PA.0 ~ PA.7 and PD.0 ~ PD.3)
  - Six internal channels from DAC0, DAC1, internal reference voltage (Int_VREF), Temperature sensor, AV_{DD}, and AV_{SS}.
  - Supports three reference voltage sources from V_{REF} pin, internal reference voltage (Int_VREF), and AV_{DD}
  - Single scan/single cycle scan/continuous scan
  - Each channel with individual result register
  - Scan on enabled channels
  - Threshold voltage detection (comparator function)
  - Conversion start by software programming or external input
  - Supports PDMA mode
  - Supports up to four timer time-out events (TMR0, TMR1, TMR2 and TMR3) to enable ADC

- DAC
  - 12-bit monotonic output with 400K conversion rate
  - Supports three reference voltage sources from V_{REF} pin, internal reference voltage (Int_VREF), and AV_{DD}.
  - Synchronized update capability for two DACs (group function)
  - Supports up to four timer time-out events (TMR0, TMR1, TMR2 and TMR3), software or PDMA to trigger DAC to conversion
- SmartCard (SC)
  - Compliant to ISO-7816-3 T=0, T=1
  - Supports up to three ISO-7816-3 ports
  - Separates receive/transmit 4 bytes entry FIFO for data payloads
  - Programmable transmission clock frequency
  - Programmable receiver buffer trigger level
  - Programmable guard time selection (11 ETU ~ 266 ETU)
  - A 24-bit and two 8-bit time-out counter for Answer to Reset (ATR) and waiting times processing
  - Supports auto inverse convention function
  - Supports stop clock level and clock stop (clock keep) function
  - Supports transmitter and receiver error retry and error limit function
  - Supports hardware activation sequence process
  - Supports hardware warm reset sequence process
  - Supports hardware deactivation sequence process
  - Supports hardware auto deactivation sequence when detecting the card is removed
  - Support UART mode (Half Duplex)
- LCD
  - LCD driver for up to 4 COM x 40 SEG or 6 COM x 38 SEG
  - Supports Static,1/2 bias and 1/3 bias voltage
  - Four display modes: Static, 1/2 duty, 1/3 duty, 1/4 duty, 1/5 duty and 1/6 duty.
  - Selectable LCD frequency by frequency divider
  - Configurable frame frequency
  - Internal Charge pump, adjustable contrast adjustment
  - Configurable Charge pump frequency
  - Blinking capability
  - Supports R-type/C-type method
  - LCD frame interrupt
- USB 2.0 Full-speed Device
  - One set of USB 2.0 FS Device 12 Mbps
  - On-chip USB Transceiver

- Provides 1 interrupt source with 4 interrupt events
- Supports Control, Bulk In/Out, Interrupt and Isochronous transfers
- Auto suspend function when no bus signaling for 3 ms
- Provides 8 programmable endpoints
- Includes 512 Bytes internal SRAM as USB buffer
- Provides remote wake-up capability

- EBI (External bus interface)
  - Accessible space: 64 KB in 8-bit mode or 128 KB in 16-bit mode
  - Supports 8bit/16bit data width
  - Supports byte write in 16-bit data width mode
- One built-in temperature sensor with 1℃ resolution
- 96-bit unique ID
- 128-bit unique customer ID
- Operating Temperature: -40℃~85℃
- Packages:
  - All Green package (RoHS)
  - LQFP 128-pin(14x14) / 64-pin (7x7)

3 PARTS INFORMATION LIST AND PIN CONFIGURATION

3.1 NuMicro® Nano100 Series Selection Code

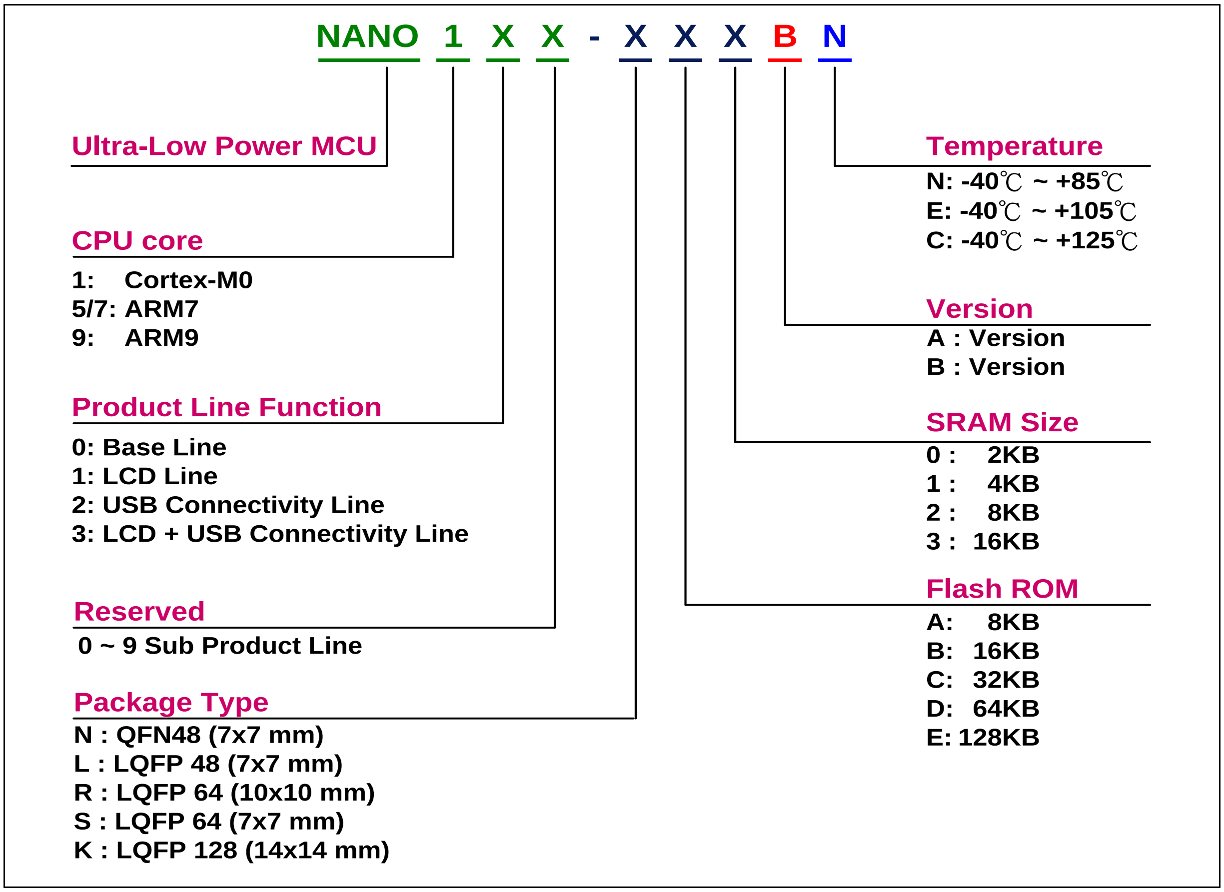


Figure 3.1-1 NuMicro® Nano100 Series Selection Code

3.2 NuMicro® Nano100 Products Selection Guide

3.2.1 NuMicro® Nano100 Base Line Selection Guide

Part No.	Flash (KB)	SRAM (KB)	Data Flash (KB)	LDROM (KB)	I/O	Timer (32-bit)	Connectivity				I^2S	PWM (16-bit)	ADC (12-bit)	RTC	EBI	IRC 10 kHz 12 MHz	PDMA	LCD	DAC (12-bit)	ISO-7816-3	ISP/ICP	Package
							UART	SPI	I^2C	USB												
NANO100NC2BN	32	8	Configurable	4	38	4	4	3	2	-	1	6	7	V	-	V	8	-	2	2	V	QFN48*
NANO100ND2BN	64	8	Configurable	4	38	4	4	3	2	-	1	6	7	V	-	V	8	-	2	2	V	QFN48*
NANO100ND3BN	64	16	Configurable	4	38	4	4	3	2	-	1	6	7	V	-	V	8	-	2	2	V	QFN48*
NANO100NE3BN	128	16	Configurable	4	38	4	4	3	2	-	1	6	7	V	-	V	8	-	2	2	V	QFN48*
NANO100LC2BN	32	8	Configurable	4	38	4	4	3	2	-	1	6	7	V	-	V	8	-	2	2	V	LQFP48
NANO100LD2BN	64	8	Configurable	4	38	4	4	3	2	-	1	6	7	V	-	V	8	-	2	2	V	LQFP48
NANO100LD3BN	64	16	Configurable	4	38	4	4	3	2	-	1	6	7	V	-	V	8	-	2	2	V	LQFP48
NANO100LE3BN	128	16	Configurable	4	38	4	4	3	2	-	1	6	7	V	-	V	8	-	2	2	V	LQFP48
NANO100SC2BN	32	8	Configurable	4	52	4	5	3	2	-	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO100SD2BN	64	8	Configurable	4	52	4	5	3	2	-	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO100SD3BN	64	16	Configurable	4	52	4	5	3	2	-	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO100SE3BN	128	16	Configurable	4	52	4	5	3	2	-	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO100KD3BN	64	16	Configurable	4	86	4	5	3	2	-	1	8	12	V	V	V	8	-	2	3	V	LQFP128
NANO100KE3BN	128	16	Configurable	4	86	4	5	3	2	-	1	8	12	V	V	V	8	-	2	3	V	LQFP128

QFN48*: 7x7, pitch 0.5 mm; LQFP48: 7x7, pitch 0.5 mm; LQFP64: 7x7, pitch 0.4 mm; LQFP128: 14x14, pitch 0.4 mm

Table 3.2-1 Nano100 Base Line Selection Table

3.2.2 NuMicro® Nano110 LCD Line Selection Guide

Part No.	Flash (KB)	SRAM (KB)	Data Flash (KB)	LDROM (KB)	I/O	Timer (32-bit)	Connectivity				I^2S	PWM (16-bit)	ADC (12-bit)	RTC	EBI	IRC 10 kHz 12 MHz	PDMA	LCD	DAC (12-bit)	ISO-7816-3	ISP/ICP	Package
							UART	SPI	I^2C	USB												
NANO110SC2BN	32	8	Configurable	4	51	4	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO110SD2BN	64	8	Configurable	4	51	4	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO110SD3BN	64	16	Configurable	4	51	4	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO110SE3BN	128	16	Configurable	4	51	4	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO110RC2BN	32	8	Configurable	4	51	4	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64*
NANO110RD2BN	64	8	Configurable	4	51	4	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64*
NANO110RD3BN	64	16	Configurable	4	51	4	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64*
NANO110RE3BN	128	16	Configurable	4	51	4	5	3	2	-	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64*
NANO110KC2BN	32	8	Configurable	4	86	4	5	3	2	-	1	8	12	V	V	V	8	4x40, 6x38	2	3	V	LQFP128
NANO110KD2BN	64	8	Configurable	4	86	4	5	3	2	-	1	8	12	V	V	V	8	4x40, 6x38	2	3	V	LQFP128
NANO110KD3BN	64	16	Configurable	4	86	4	5	3	2	-	1	8	12	V	V	V	8	4x40, 6x38	2	3	V	LQFP128
NANO110KE3BN	128	16	Configurable	4	86	4	5	3	2	-	1	8	12	V	V	V	8	4x40, 6x38	2	3	V	LQFP128

LQFP64: 7x7, pitch 0.4 mm; LQFP64*: 10x10, pitch 0.5 mm; LQFP128: 14x14, pitch 0.4 mm

Table 3.2-2 Nano110 LCD Line Selection Table

3.2.3 NuMicro® Nano120 USB Connectivity Line Selection Guide

Part No.	Flash (KB)	SRAM (KB)	Data Flash (KB)	LDROM (KB)	I/O	Timer (32-bit)	Connectivity				I^2S	PWM (16-bit)	ADC (12-bit)	RTC	EBI	IRC 10 kHz 12 MHz	PDMA	LCD	DAC (12-bit)	ISO-7816-3	ISP/ICP	Package
							UART	SPI	I^2C	USB												
NANO120LC2BN	32	8	Configurable	4	34	4	4	3	2	1	1	4	7	V	-	V	8	-	2	2	V	LQFP48
NANO120LD2BN	64	8	Configurable	4	34	4	4	3	2	1	1	4	7	V	-	V	8	-	2	2	V	LQFP48
NANO120LD3BN	64	16	Configurable	4	34	4	4	3	2	1	1	4	7	V	-	V	8	-	2	2	V	LQFP48
NANO120LE3BN	128	16	Configurable	4	34	4	4	3	2	1	1	4	7	V	-	V	8	-	2	2	V	LQFP48
NANO120SC2BN	32	8	Configurable	4	48	4	5	3	2	1	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO120SD2BN	64	8	Configurable	4	48	4	5	3	2	1	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO120SD3BN	64	16	Configurable	4	48	4	5	3	2	1	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO120SE3BN	128	16	Configurable	4	48	4	5	3	2	1	1	8	7	V	-	V	8	-	2	3	V	LQFP64
NANO120KD3BN	64	16	Configurable	4	86	4	5	3	2	1	1	8	8	V	V	V	8	-	2	3	V	LQFP128
NANO120KE3BN	128	16	Configurable	4	86	4	5	3	2	1	1	8	8	V	V	V	8	-	2	3	V	LQFP128

LQFP48: 7x7, pitch 0.5 mm; LQFP64: 7x7, pitch 0.4 mm; LQFP128: 14x14, pitch 0.4 mm

Table 3.2-3 Nano120 USB Connectivity Line Selection Table

3.2.4 NuMicro® Nano130 Advanced Line Selection Guide

Part No.	Flash (KB)	SRAM (KB)	Data Flash (KB)	LDROM (KB)	I/O	Timer (32-bit)	Connectivity				I^2S	PWM (16-bit)	ADC (12-bit)	RTC	EBI	IRC 10 kHz 12 MHz	PDMA	LCD	DAC (12-bit)	ISO-7816-3	ISP/ICP	Package
							UART	SPI	I^2C	USB												
NANO130SC2BN	32	8	Configurable	4	47	4	5	3	2	1	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO130SD2BN	64	8	Configurable	4	47	4	5	3	2	1	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO130SD3BN	64	16	Configurable	4	47	4	5	3	2	1	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO130SE3BN	128	16	Configurable	4	47	4	5	3	2	1	1	7	7	V	-	V	8	4x31, 6x29	2	3	V	LQFP64
NANO130KC2BN	32	8	Configurable	4	86	4	5	3	2	1	1	8	8	V	V	V	8	4x40, 6x38	2	3	V	LQFP128
NANO130KD2BN	64	8	Configurable	4	86	4	5	3	2	1	1	8	8	V	V	V	8	4x40, 6x38	2	3	V	LQFP128
NANO130KD3BN	64	16	Configurable	4	86	4	5	3	2	1	1	8	8	V	V	V	8	4x40, 6x38	2	3	V	LQFP128
NANO130KE3BN	128	16	Configurable	4	86	4	5	3	2	1	1	8	8	V	V	V	8	4x40, 6x38	2	3	V	LQFP128

LQFP64: 7x7, pitch 0.4 mm; LQFP128: 14x14, pitch 0.4 mm

Table 3.2-4 Nano130 Advanced Line Selection Table

3.3 Pin Configuration

3.3.1 NuMicro® Nano100 Pin Diagrams

3.3.1.1 NuMicro® Nano100 LQFP 128-pin

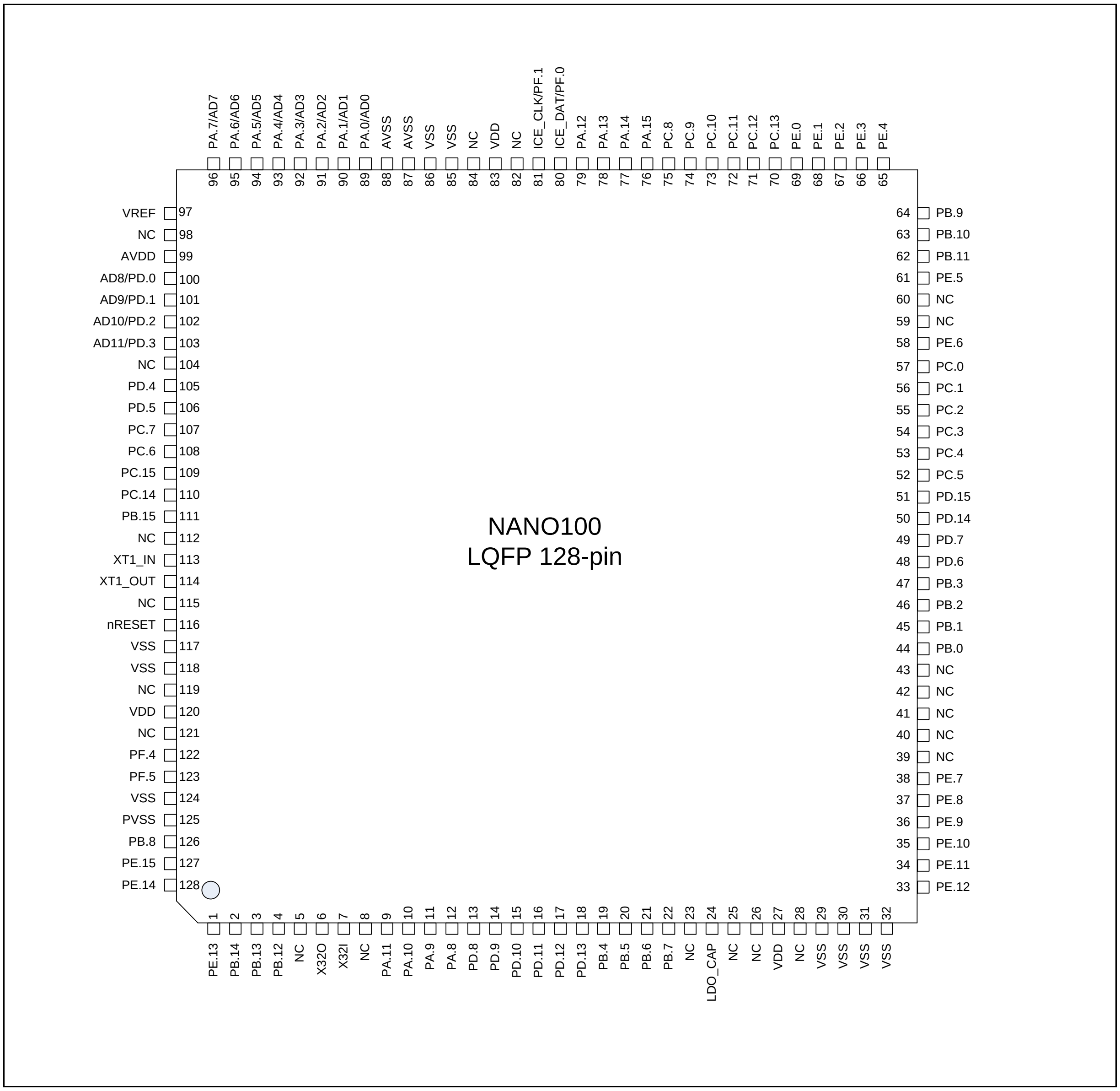


Figure 3.3-1 NuMicro® Nano100 LQFP 128-pin Diagram

3.3.1.2 NuMicro® Nano100 LQFP 64-pin

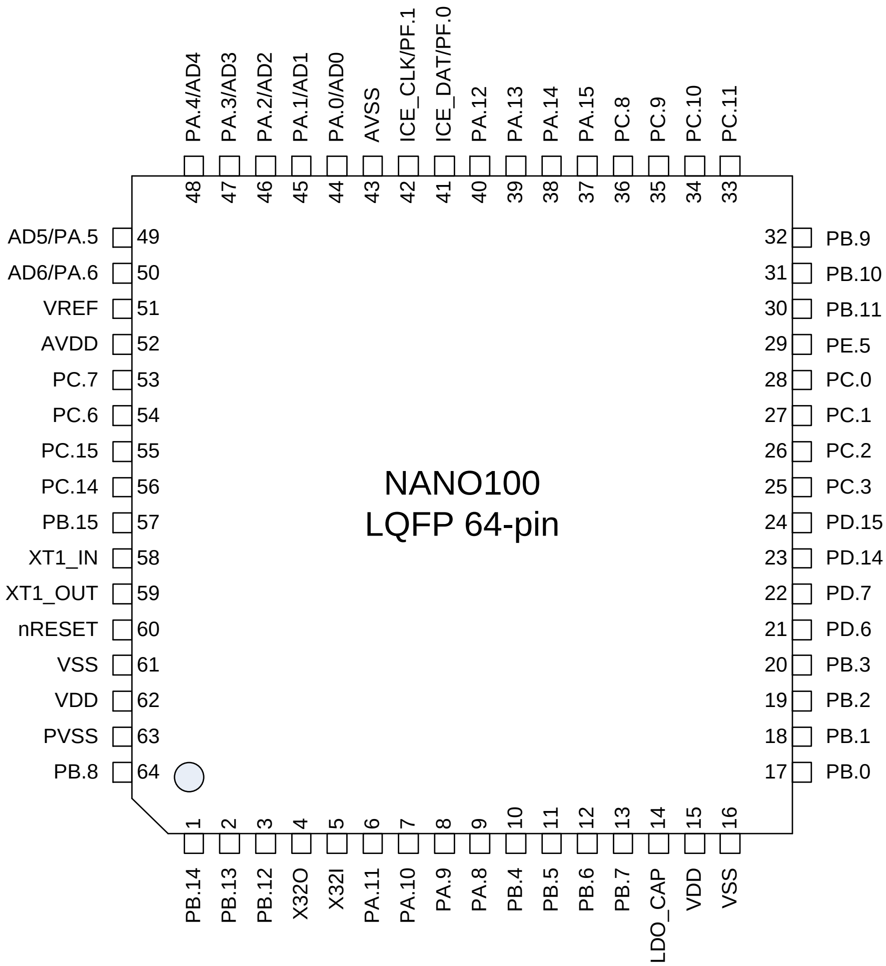


Figure 3.3-2 NuMicro® Nano100 LQFP 64-pin Diagram

3.3.1.3 NuMicro® Nano100 LQFP/QFN 48-pin

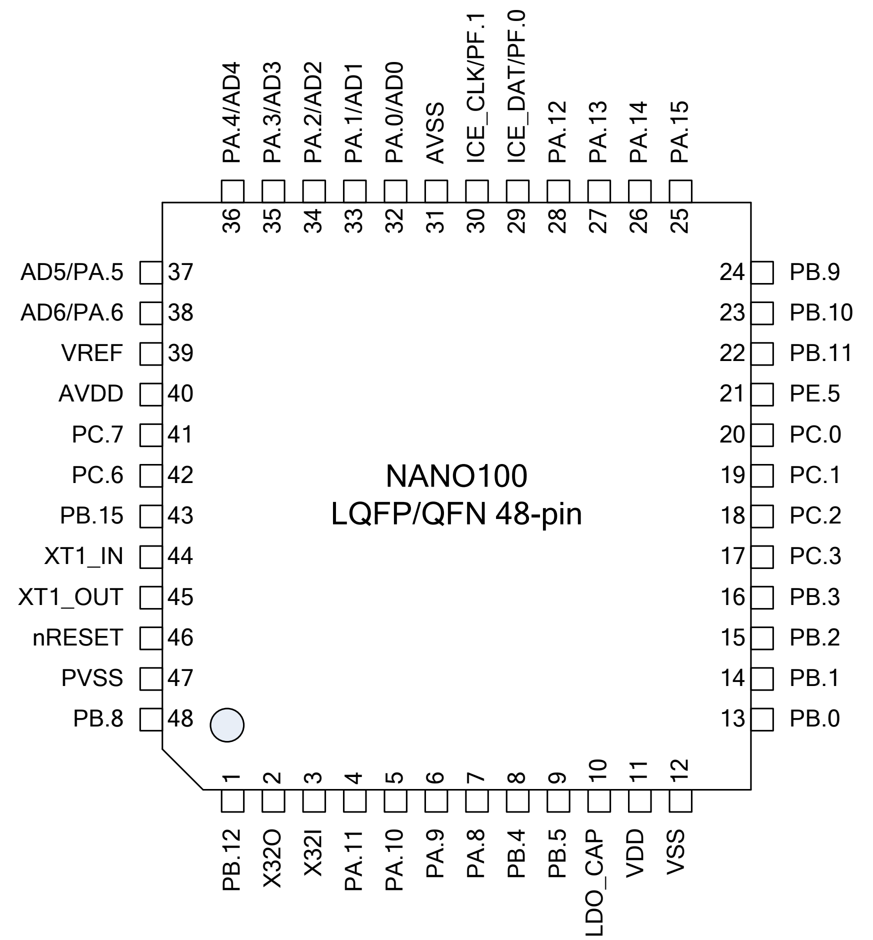


Figure 3.3-3 NuMicro® Nano100 LQFP 48-pin Diagram

3.3.2 NuMicro® Nano110 Pin Diagrams

3.3.2.1 NuMicro® Nano110 LQFP 128-pin

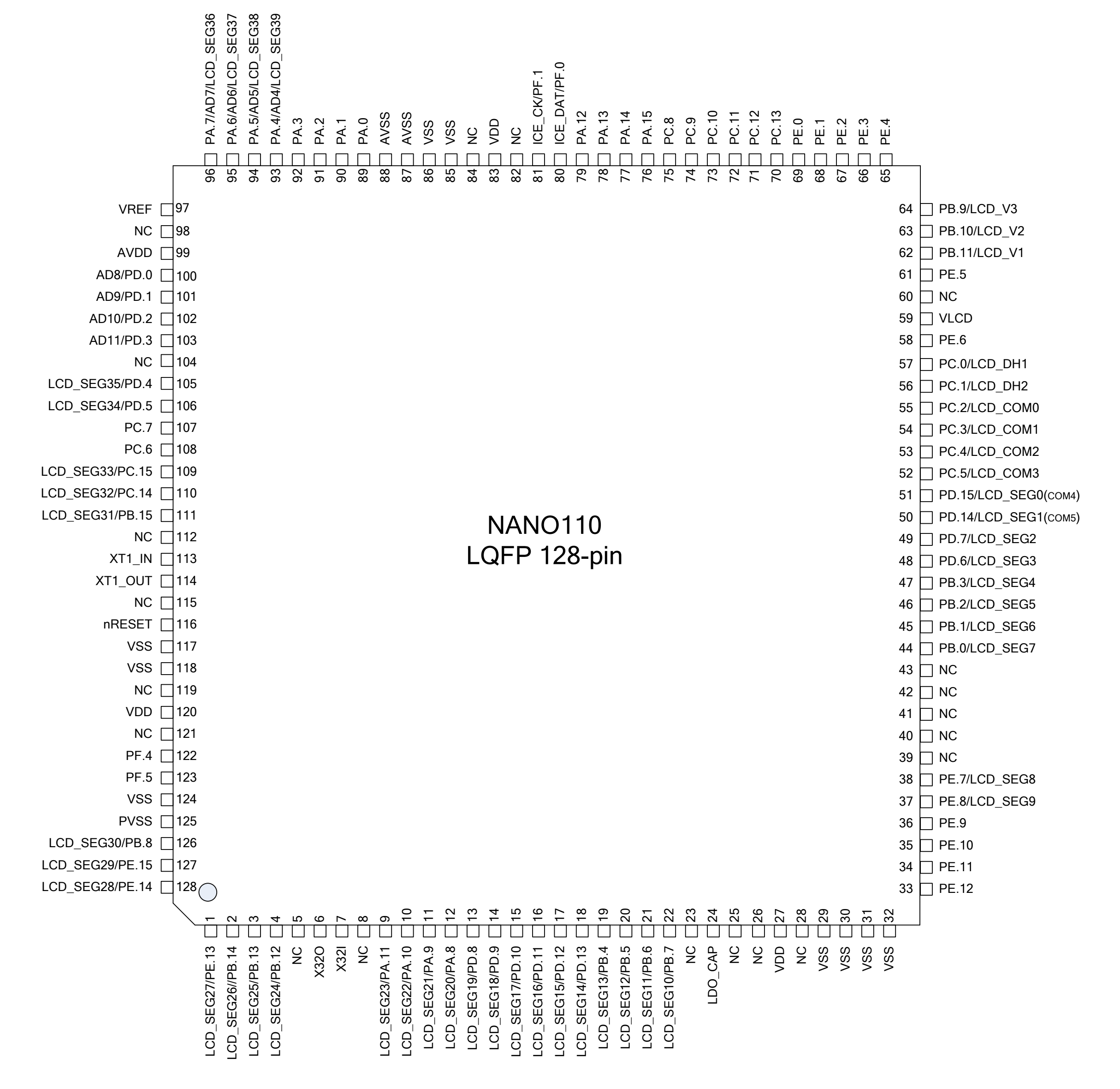


Figure 3.3-4 NuMicro® Nano110 LQFP 128-pin Diagram

3.3.2.2 NuMicro® Nano110 LQFP 64-pin

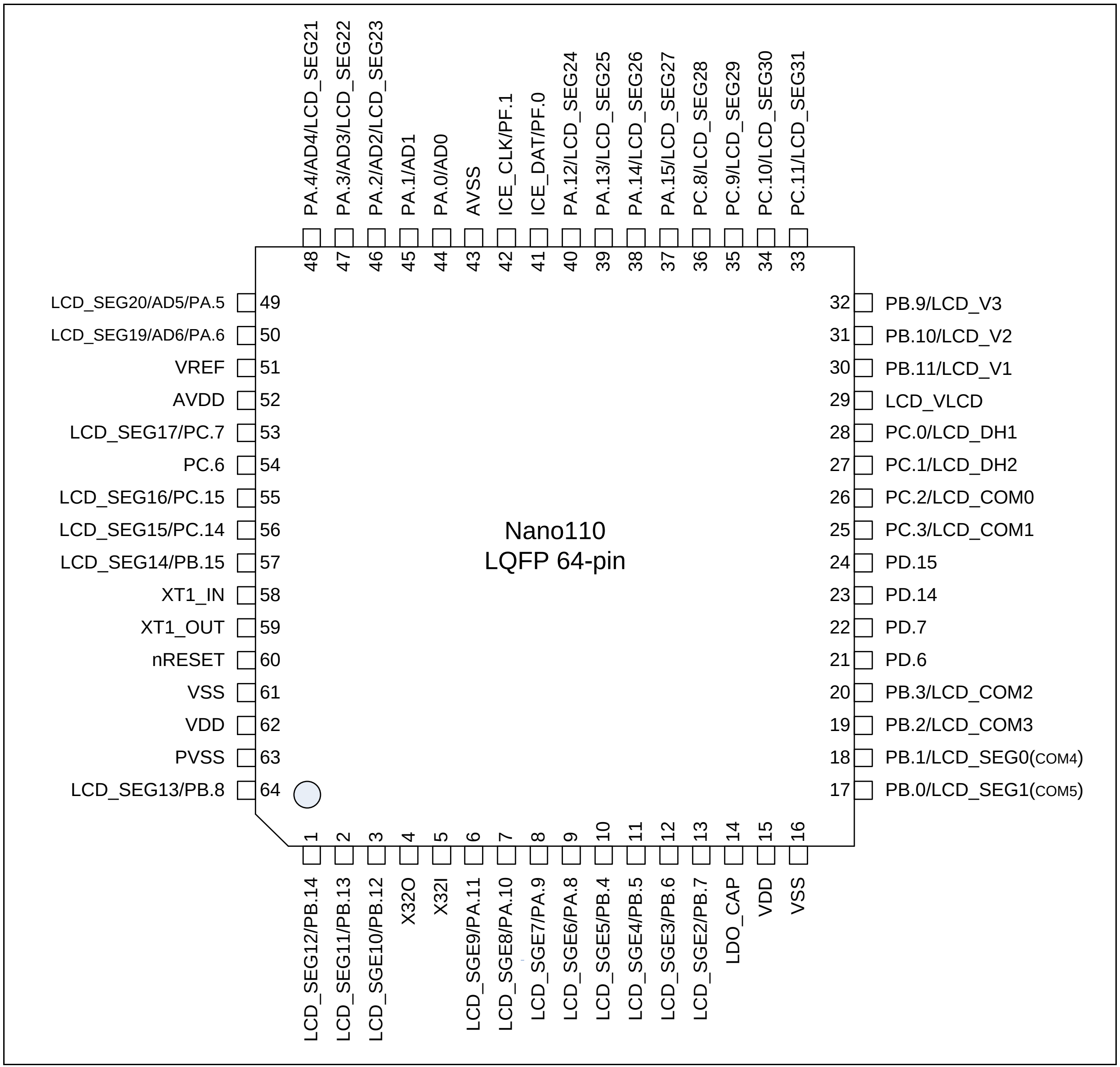


Figure 3.3-5 NuMicro® Nano110 LQFP 64-pin Diagram

3.3.3 NuMicro® Nano120 Pin Diagrams

3.3.3.1 NuMicro® Nano120 LQFP 128-pin

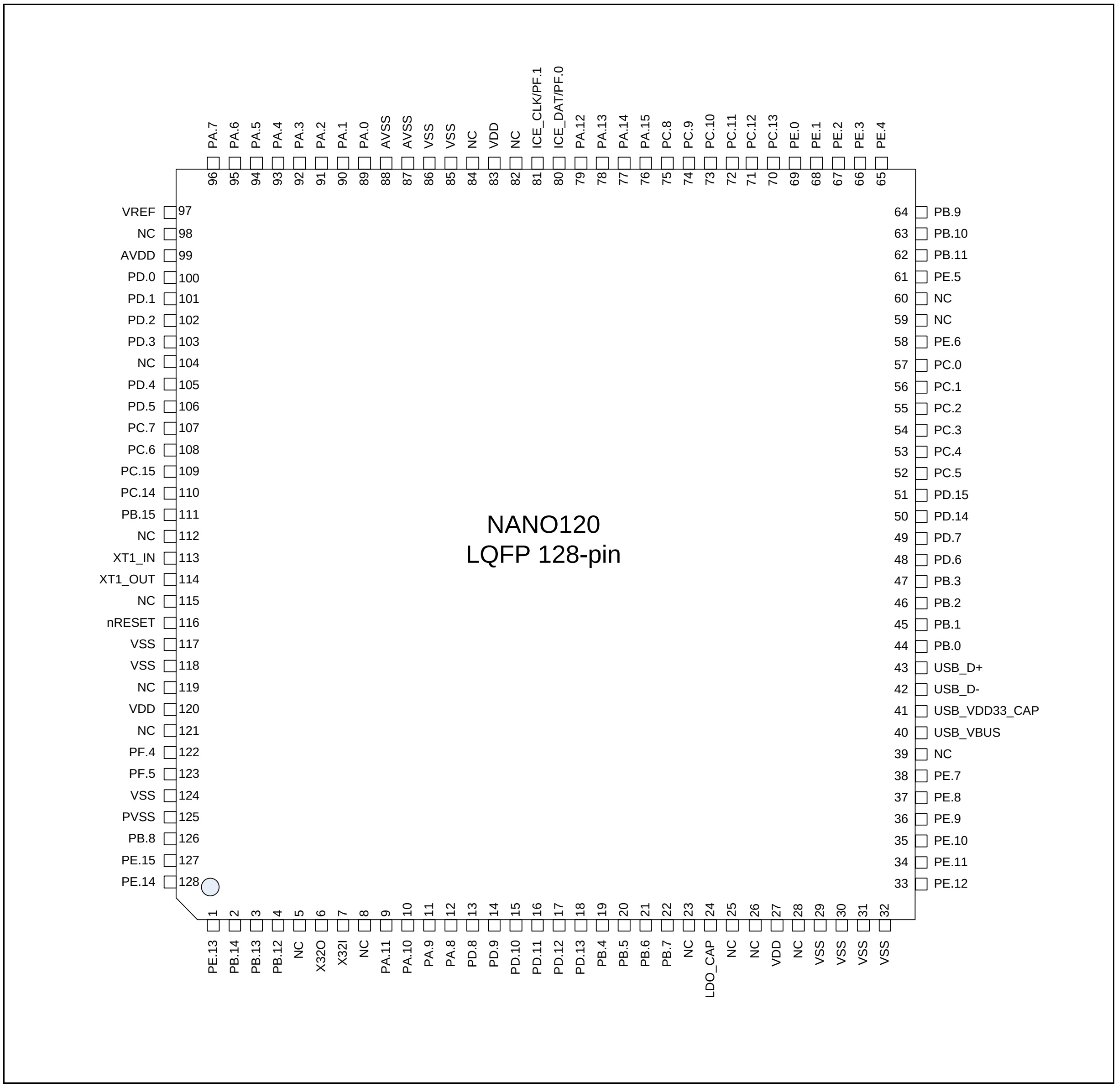


Figure 3.3-6 NuMicro® Nano120 LQFP 128-pin Diagram

3.3.3.2 NuMicro® Nano120 LQFP 64-pin

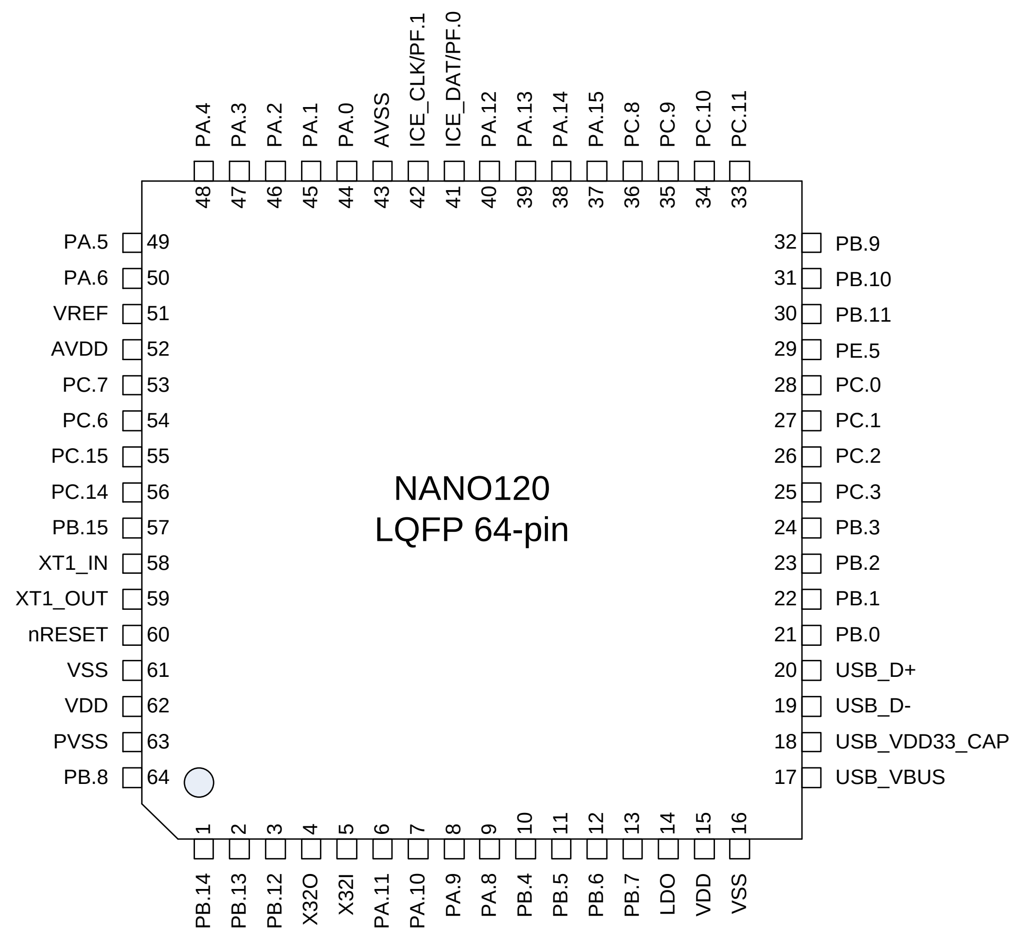


Figure 3.3-7 NuMicro® Nano120 LQFP 64-pin Diagram

3.3.3.3 NuMicro® Nano120 LQFP 48-pin

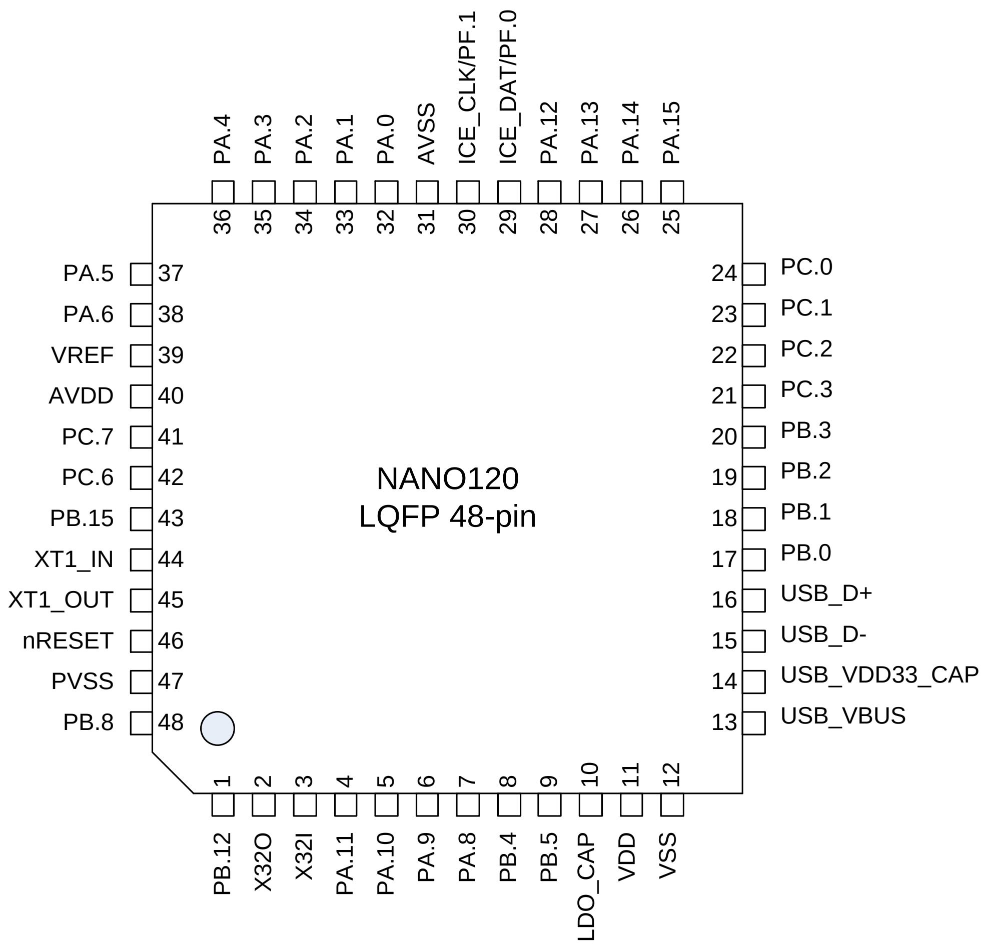


Figure 3.3-8 NuMicro® Nano120 LQFP 48-pin Diagram

3.3.4 NuMicro® Nano130 Pin Diagrams

3.3.4.1 NuMicro® Nano130 LQFP 128-pin

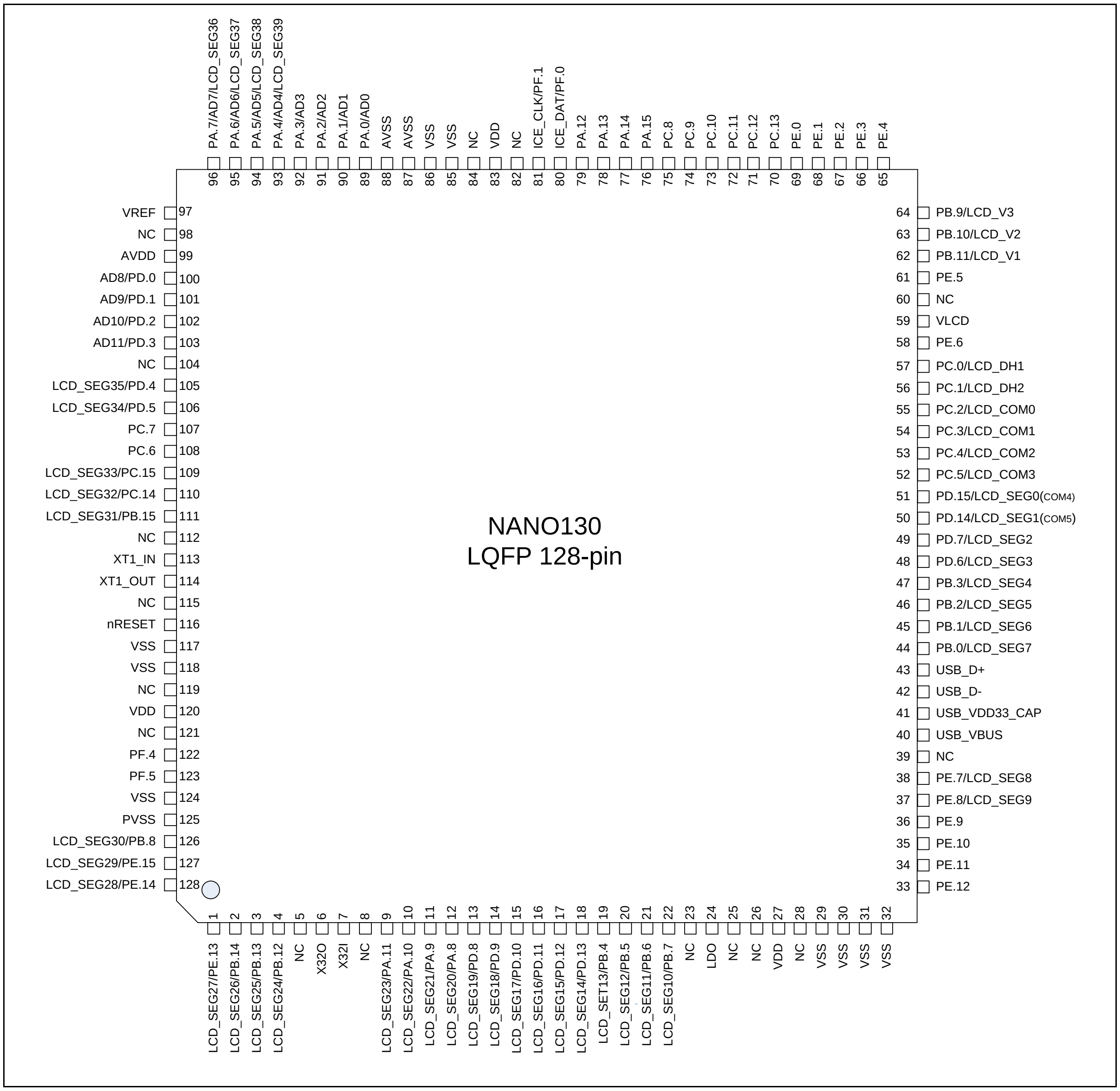


Figure 3.3-9 NuMicro® Nano130 LQFP 128-pin Diagram

3.3.4.2 NuMicro® Nano130 LQFP 64-pin

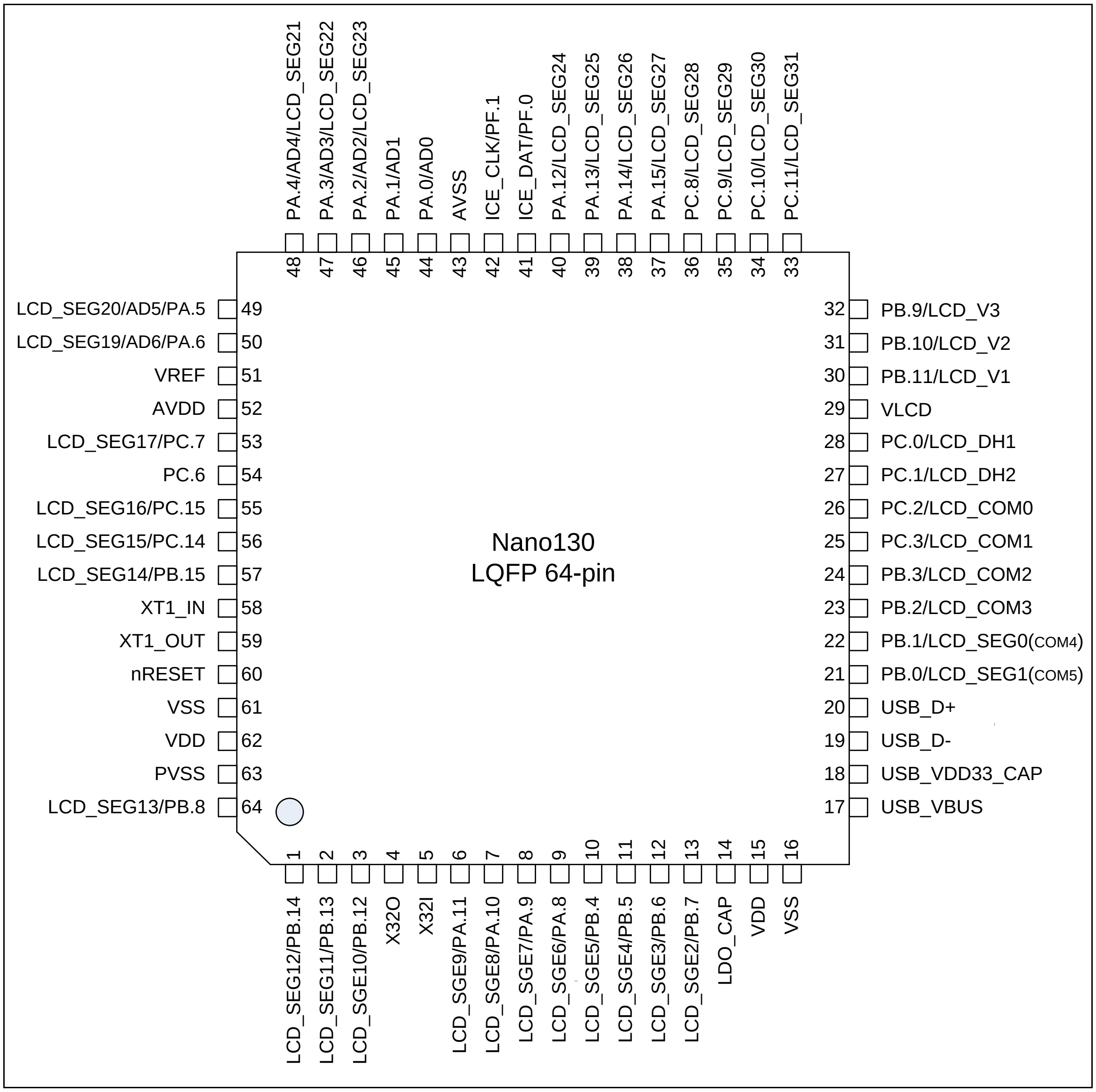


Figure 3.3-10 NuMicro® Nano130 LQFP 64-pin Diagram

3.4 Pin Description

3.4.1 NuMicro® Nano100 Pin Description

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP/QFN 48-pin			
1			PE.13	**I/O**	General purpose digital I/O pin
2	1		PB.14	**I/O**	General purpose digital I/O pin
			INT0	**I**	External interrupt0 input pin
			SC2_CD	**I**	SmartCard2 card detect pin
			SPI2_SS1	**I/O**	SPI2 2nd slave select pin
3	2		PB.13	**I/O**	General purpose digital I/O pin
			EBI_AD1	**I/O**	EBI Address/Data bus bit1
4	3	1	PB.12	**I/O**	General purpose digital I/O pin
			EBI_AD0	**I/O**	EBI Address/Data bus bit0
			FCLKO	**O**	Frequency Divider output pin
5					NC
6	4	2	X32O	**O**	External 32.768 kHz crystal output pin
7	5	3	X32I	**I**	External 32.768 kHz crystal input pin
8					NC
9	6	4	PA.11	**I/O**	General purpose digital I/O pin
			I2C1_SCL	**I/O**	I^2C1 clock pin
			EBI_nRD	**O**	EBI read enable output pin
			SC0_RST	**O**	SmartCard0 RST pin
			SPI2_MOSI0	**I/O**	SPI2 1st MOSI (Master Out, Slave In) pin
10	7	5	PA.10	**I/O**	General purpose digital I/O pin
			I2C1_SDA	**I/O**	I^2C1 data I/O pin
			EBI_nWR	**O**	EBI write enable output pin
			SC0_PWR	**O**	SmartCard0 Power pin
			SPI2_MISO0	**I/O**	SPI2 1st MISO (Master In, Slave Out) pin
11	8	6	PA.9	**I/O**	General purpose digital I/O pin
			I2C0_SCL	**I/O**	I^2C0 clock pin
			SC0_DAT	**I/O**	SmartCard0 DATA pin(SC0_UART_RXD)
			SPI2_CLK	**I/O**	SPI2 serial clock pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP/QFN 48-pin			
12	9	7	PA.8	I/O	General purpose digital I/O pin
			I2C0_SDA	I/O	I^2C0 data I/O pin
			SC0_CLK	O	SmartCard0 clock pin(SC0_UART_TXD)
			SPI2_SS0	I/O	SPI2 1st slave select pin
13			PD.8	I/O	General purpose digital I/O pin
14			PD.9	I/O	General purpose digital I/O pin
15			PD.10	I/O	General purpose digital I/O pin
16			PD.11	I/O	General purpose digital I/O pin
17			PD.12	I/O	General purpose digital I/O pin
18			PD.13	I/O	General purpose digital I/O pin
19	10	8	PB.4	I/O	General purpose digital I/O pin
			UART1_RXD	I	UART1 Data receiver input pin
			SC0_CD	I	SmartCard0 card detect pin
			SPI2_SS0	I/O	SPI2 1st slave select pin
20	11	9	PB.5	I/O	General purpose digital I/O pin
			UART1_TXD	O	UART1 Data transmitter output pin
			SC0_RST	O	SmartCard0 RST pin
			SPI2_CLK	I/O	SPI2 serial clock pin
21	12		PB.6	I/O	General purpose digital I/O pin
			UART1_RTSn	O	UART1 Request to Send output pin
			EBI_ALE	O	EBI address latch enable output pin
			SPI2_MISO0	I/O	SPI2 1st MISO (Master In, Slave Out) pin
22	13		PB.7	I/O	General purpose digital I/O pin
			UART1_CTSn	I	UART1 Clear to Send input pin
			EBI_nCS	O	EBI chip select enable output pin
			SPI2_MOSI0	I/O	SPI2 1st MOSI (Master Out, Slave In) pin
23					NC
24	14	10	LDO_CAP	P	LDO output pin
25					NC
26					NC

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP/QFN 48-pin			
27	15	11	V_{DD}	**P**	Power supply for I/O ports and LDO source
28					NC
29	16	12	V_{SS}	**P**	Ground
30			V_{SS}	**P**	Ground
31			V_{SS}	**P**	Ground
32			V_{SS}	**P**	Ground
33			PE.12	**I/O**	General purpose digital I/O pin
34			PE.11	**I/O**	General purpose digital I/O pin
35			PE.10	**I/O**	General purpose digital I/O pin
36			PE.9	**I/O**	General purpose digital I/O pin
37			PE.8	**I/O**	General purpose digital I/O pin
38			PE.7	**I/O**	General purpose digital I/O pin
39					NC
40					NC
41					NC
42					NC
43					NC
44	17	13	PB.0	**I/O**	General purpose digital I/O pin
			UART0_RXD	**I**	UART0 Data receiver input pin
			SPI1_MOSI0	**I/O**	SPI1 1st MOSI (Master Out, Slave In) pin
45	18	14	PB.1	**I/O**	General purpose digital I/O pin
			UART0_TXD	**O**	UART0 Data transmitter output pin
			SPI1_MISO0	**I/O**	SPI1 1st MISO (Master In, Slave Out) pin
46	19	15	PB.2	**I/O**	General purpose digital I/O pin
			UART0_RTSn	**O**	UART0 Request to Send output pin
			EBI_nWRL	**O**	EBI low byte write enable output pin
			SPI1_CLK	**I/O**	SPI1 serial clock pin
47	20	16	PB.3	**I/O**	General purpose digital I/O pin
			UART0_CTSn	**I**	UART0 Clear to Send input pin
			EBI_nWRH	**O**	EBI high byte write enable output pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP/QFN 48-pin			
			SPI1_SS0	I/O	SPI1 1st slave select pin
48	21		PD.6	I/O	General purpose digital I/O pin
49	22		PD.7	I/O	General purpose digital I/O pin
50	23		PD.14	I/O	General purpose digital I/O pin
51	24		PD.15	I/O	General purpose digital I/O pin
52			PC.5	I/O	General purpose digital I/O pin
			SPI0_MOSI1	I/O	SPI0 2nd MOSI (Master Out, Slave In) pin
53			PC.4	I/O	General purpose digital I/O pin
			SPI0_MISO1	I/O	SPI0 2nd MISO (Master In, Slave Out) pin
54	25	17	PC.3	I/O	General purpose digital I/O pin
			SPI0_MOSI0	I/O	SPI0 1st MOSI (Master Out, Slave In) pin
			I2S_DO	O	I^2S data output
			SC1_RST	O	SmartCard1 RST pin
55	26	18	PC.2	I/O	General purpose digital I/O pin
			SPI0_MISO0	I/O	SPI0 1st MISO (Master In, Slave Out) pin
			I2S_DI	I	I^2S data input
			SC1_PWR	O	SmartCard1 PWR pin
56	27	19	PC.1	I/O	General purpose digital I/O pin
			SPI0_CLK	I/O	SPI0 serial clock pin
			I2S_BCLK	I/O	I^2S bit clock pin
			SC1_DAT	I/O	SmartCard1 DATA pin(SC1_UART_RXD)
57	28	20	PC.0 / MCLKO	I/O	General purpose digital I/O pin / Module clock output pin
			SPI0_SS0	I/O	SPI0 1st slave select pin
			I2S_LRCLK	I/O	I^2S left right channel clock
			SC1_CLK	O	SmartCard1 clock pin(SC1_UART_TXD)
58			PE.6	I/O	General purpose digital I/O pin
59					NC
60					NC
61	29	21	PE.5	I/O	General purpose digital I/O pin
			PWM1_CH1	I/O	PWM1 Channel1 output

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP/QFN 48-pin			
62	30	22	PB.11	I/O	General purpose digital I/O pin
			PWM1_CH0	I/O	PWM1 Channel0 output
			TM3	O	Timer3 external counter input
			SC2_DAT	I/O	SmartCard2 DATA pin(SC2_UART_RXD)
			SPI0_MISO0	I/O	SPI0 1st MISO (Master In, Slave Out) pin
63	31	23	PB.10	I/O	General purpose digital I/O pin
			SPI0_SS1	I/O	SPI0 2nd slave select pin
			TM2	O	Timer2 external counter input
			SC2_CLK	O	SmartCard2 clock pin(SC2_UART_TXD)
			SPI0_MOSI0	I/O	SPI0 1st MOSI (Master Out, Slave In) pin
64	32	24	PB.9	I/O	General purpose digital I/O pin
			SPI1_SS1	I/O	SPI1 2nd slave select pin
			TM1	O	Timer1 external counter input
			SC2_RST	O	SmartCard2 RST pin
			INT0	I	External interrupt0 input pin
65			PE.4	I/O	General purpose digital I/O pin
			SPI0_MOSI0	I/O	SPI0 1st MOSI (Master Out, Slave In) pin
66			PE.3	I/O	General purpose digital I/O pin
			SPI0_MISO0	I/O	SPI0 1st MISO (Master In, Slave Out) pin
67			PE.2	I/O	General purpose digital I/O pin
			SPI0_CLK	I/O	SPI0 serial clock pin
68			PE.1	I/O	General purpose digital I/O pin.
			PWM1_CH3	I/O	PWM1 Channel3 output
			SPI0_SS0	I/O	SPI0 1st slave select pin
69			PE.0	I/O	General purpose digital I/O pin
			PWM1_CH2	I/O	PWM1 Channel2 output
			I2S_MCLK	O	I^2S master clock output pin
70			PC.13	I/O	General purpose digital I/O pin
			SPI1_MOSI1	I/O	SPI1 2nd MOSI (Master Out, Slave In) pin
			PWM1_CH1	I/O	PWM1 Channel1 output

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP/QFN 48-pin			
			SNOOPER	I	Snooper pin
			INT1	I	External interrupt 1
			I2C0_SCL	O	I^2C0 clock pin
71			PC.12	I/O	General purpose digital I/O pin
			SPI1_MISO1	I/O	SPI1 2^{nd} MISO (Master In, Slave Out) pin
			PWM1_CH0	I/O	PWM1 Channel0 output
			INT0	I	External interrupt0 input pin
			I2C0_SDA	I/O	I^2C0 data I/O pin
72	33		PC.11	I/O	General purpose digital I/O pin
			SPI1_MOSI0	I/O	SPI1 1^{st} MOSI (Master Out, Slave In) pin
			UART1_TXD	O	UART1 Data transmitter output pin
73	34		PC.10	I/O	General purpose digital I/O pin
			SPI1_MISO0	I/O	SPI1 1^{st} MISO (Master In, Slave Out) pin
			UART1_RXD	I	UART1 Data receiver input pin
74	35		PC.9	I/O	General purpose digital I/O pin
			SPI1_CLK	I/O	SPI1 serial clock pin
			I2C1_SCL	I/O	I^2C1 clock pin
75	36		PC.8	I/O	General purpose digital I/O pin
			SPI1_SS0	I/O	SPI1 1^{st} slave select pin
			EBI_MCLK	O	EBI external clock output pin
			I2C1_SDA	I/O	I^2C1 data I/O pin
76	37	25	PA.15	I/O	General purpose digital I/O pin
			PWM0_CH3	I/O	PWM0 Channel3 output
			I2S_MCLK	O	I^2S master clock output pin
			TC3	I	Timer3 capture input
			SC0_PWR	O	SmartCard0 Power pin
			UART0_TXD	O	UART0 Data transmitter output pin
77	38	26	PA.14	I/O	General purpose digital I/O pin
			PWM0_CH2	I/O	PWM0 Channel2 output
			EBI_AD15	I/O	EBI Address/Data bus bit15

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP/QFN 48-pin			
			TC2	**I**	Timer2 capture input
			UART0_RXD	**I**	UART0 Data receiver input pin
78	39	27	PA.13	**I/O**	General purpose digital I/O pin
			PWM0_CH1	**I/O**	PWM0 Channel1 output
			EBI_AD14	**I/O**	EBI Address/Data bus bit14
			TC1	**I**	Timer1 capture input
			I2C0_SCL	**I/O**	I^2C0 clock pin
79	40	28	PA.12	**I/O**	General purpose digital I/O pin
			PWM0_CH0	**I/O**	PWM0 Channel0 output
			EBI_AD13	**I/O**	EBI Address/Data bus bit13
			TC0	**I**	Timer0 capture input
			I2C0_SDA	**I/O**	I^2C0 data I/O pin
80	41	29	ICE_DAT	**I/O**	Serial Wired Debugger Data pin **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_DAT pin.
			PF.0	**I/O**	General purpose digital I/O pin
			INT0	**I**	External interrupt0 input pin
81	42	30	ICE_CLK	**I**	Serial Wired Debugger Clock pin **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_CLK pin.
			PF.1	**I/O**	General purpose digital I/O pin
			FCLKO	**O**	Frequency Divider output pin
			INT1	**I**	External interrupt1 input pin
82					NC
83			V_{DD}	**P**	Power supply for I/O ports and LDO source for internal PLL and digital circuit
84					NC
85			V_{SS}	**P**	Ground
86			V_{SS}	**P**	Ground
87	43	31	AV_{SS}	**AP**	Ground Pin for analog circuit
88			AV_{SS}	**AP**	Ground Pin for analog circuit
89	44	32	PA.0	**I/O**	General purpose digital I/O pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP/QFN 48-pin			
			AD0	AI	ADC analog input0
			SC2_CD	I	SmartCard2 card detect
90	45	33	PA.1	I/O	General purpose digital I/O pin
			AD1	AI	ADC analog input1
			EBI_AD12	I/O	EBI Address/Data bus bit12
91	46	34	PA.2	I/O	General purpose digital I/O pin
			AD2	AI	ADC analog input2
			EBI_AD11	I/O	EBI Address/Data bus bit11
			UART1_RXD	I	UART1 Data receiver input pin
92	47	35	PA.3	I/O	General purpose digital I/O pin
			AD3	AI	ADC analog input3
			EBI_AD10	I/O	EBI Address/Data bus bit10
			UART1_TXD	O	UART1 Data transmitter output pin
93	48	36	PA.4	I/O	General purpose digital I/O pin
			AD4	AI	ADC analog input4
			EBI_AD9	I/O	EBI Address/Data bus bit9
			SC2_PWR	O	SmartCard2 Power pin
			I2C0_SDA	I/O	I^2C0 data I/O pin
94	49	37	PA.5	I/O	General purpose digital I/O pin
			AD5	AI	ADC analog input5
			EBI_AD8	I/O	EBI Address/Data bus bit8
			SC2_RST	O	SmartCard2 RST pin
			I2C0_SCL	I/O	I^2C0 clock pin
95	50	38	PA.6	I/O	General purpose digital I/O pin
			AD6	AI	ADC analog input6
			EBI_AD7	I/O	EBI Address/Data bus bit7
			TC3	I	Timer3 capture input
			SC2_CLK	O	SmartCard2 clock pin(SC2_UART_TXD)
			PWM0_CH3	I/O	PWM0 Channel3 output
96			PA.7	I/O	General purpose digital I/O pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP/QFN 48-pin			
			AD7	**AI**	ADC analog input7
			EBI_AD6	**I/O**	EBI Address/Data bus bit6
			TC2	**I**	Timer2 capture input
			SC2_DAT	**I/O**	SmartCard2 DATA pin(SC2_UART_RXD)
			PWM0_CH2	**I/O**	PWM0 Channel2 output
97	51	39	V_{REF}	**AP**	Voltage reference input for ADC
98					NC
99	52	40	AV_{DD}	**AP**	Power supply for internal analog circuit
100			PD.0	**I/O**	General purpose digital I/O pin
			UART1_RXD	**I**	UART1 Data receiver input pin
			SPI2_SS0	**I/O**	SPI2 1st slave select pin
			SC1_CLK	**O**	SmartCard1 clock pin(SC1_UART_TXD)
			AD8	**AI**	ADC analog input8
101			PD.1	**I/O**	General purpose digital I/O pin
			UART1_TXD	**O**	UART1 Data transmitter output pin
			SPI2_CLK	**I/O**	SPI2 serial clock pin
			SC1_DAT	**I/O**	SmartCard1 DATA pin(SC1_UART_RXD).
			AD9	**AI**	ADC analog input9
102			PD.2	**I/O**	General purpose digital I/O pin
			UART1_RTSn	**O**	UART1 Request to Send output pin
			I2S_LRCLK	**I/O**	I^{2}S left right channel clock
			SPI2_MISO0	**I/O**	SPI2 1st MISO (Master In, Slave Out) pin
			SC1_PWR	**O**	SmartCard1 Power pin
			AD10	**AI**	ADC analog input10
103			PD.3	**I/O**	General purpose digital I/O pin
			UART1_CTSn	**I**	UART1 Clear to Send input pin
			I2S_BCLK	**I/O**	I^{2}S bit clock pin
			SPI2_MOSI0	**I/O**	SPI2 1st MOSI (Master Out, Slave In) pin
			SC1_RST	**O**	SmartCard1 RST pin
			AD11	**AI**	ADC analog input11

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP/QFN 48-pin			
104					NC
105			PD.4	**I/O**	General purpose digital I/O pin
			I2S_DI	**I**	I^2S data input
			SPI2_MISO1	**I/O**	SPI2 2^{nd} MISO (Master In, Slave Out) pin
			SC1_CD	**I**	SmartCard1 card detect
106			PD.5	**I/O**	General purpose digital I/O pin
			I2S_DO	**O**	I^2S data output
			SPI2_MOSI1	**I/O**	SPI2 2^{nd} MOSI (Master Out, Slave In) pin
107	53	41	PC.7	**I/O**	General purpose digital I/O pin
			DA1_OUT	**AO**	DAC 1 output
			EBI_AD5	**I/O**	EBI Address/Data bus bit5
			TC1	**I**	Timer1 capture input
			PWM0_CH1	**I/O**	PWM0 Channel1 output
108	54	42	PC.6	**I/O**	General purpose digital I/O pin
			DA0_OUT	**I**	DAC0 output
			EBI_AD4	**I/O**	EBI Address/Data bus bit4
			TC0	**I**	Timer0 capture input
			SC1_CD	**I**	SmartCard1 card detect pin
			PWM0_CH0	**I/O**	PWM0 Channel0 output
109	55		PC.15	**I/O**	General purpose digital I/O pin
			EBI_AD3	**I/O**	EBI Address/Data bus bit3
			TC0	**I**	Timer0 capture input
			PWM1_CH2	**I/O**	PWM1 Channel2 output
110	56		PC.14	**I/O**	General purpose digital I/O pin
			EBI_AD2	**I/O**	EBI Address/Data bus bit2
			PWM1_CH3	**I/O**	PWM1 Channel3 output
111	57	43	PB.15	**I/O**	General purpose digital I/O pin
			INT1	**I**	External interrupt1 input pin
			SNOOPER	**I**	Snooper pin
			SC1_CD	**I**	SmartCard1 card detect

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP/QFN 48-pin			
112					NC
113	58	44	XT1_IN	**O**	External 4~24 MHz crystal output pin
			PF.3	**I/O**	General purpose digital I/O pin
114	59	45	XT1_OUT	**I**	External 4~24 MHz crystal input pin
			PF.2	**I/O**	General purpose digital I/O pin
115					NC
116	60	46	nRESET	**I**	External reset input: Low active, set this pin low reset chip to initial state. With internal pull-up. **Note:** It is recommended to use 10 kΩ pull-up resistor and 10 µF capacitor on nRESET pin.
117	61		V_{SS}	**P**	Ground
118			V_{SS}	**P**	Ground
119					NC
120	62		V_{DD}	**P**	Power supply for I/O ports and LDO source for internal PLL and digital circuit
121					NC
122			PF.4	**I/O**	General purpose digital I/O pin
			I2C0_SDA	**I/O**	I^2C0 data I/O pin
123			PF.5	**I/O**	General purpose digital I/O pin
			I2C0_SCL	**I/O**	I^2C0 clock pin
124			VSS	**P**	Ground
125	63	47	PVSS	**P**	PLL Ground
126	64	48	PB.8	**I/O**	General purpose digital I/O pin
			STADC	**I**	ADC external trigger input.
			TM0	**I**	Timer0 external counter input
			INT0	**I**	External interrupt0 input pin
			SC2_PWR	**O**	SmartCard2 Power pin
127			PE.15	**I/O**	General purpose digital I/O pin
128			PE.14	**I/O**	General purpose digital I/O pin

Note:

Pin Type: I = Digital Input, O = Digital Output; AI = Analog Input; AO = Analog Output; P = Power Pin; AP = Analog Power.

3.4.2 NuMicro® Nano110 Pin Description

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
1			PE.13	I/O	General purpose digital I/O pin
			LCD_SEG27	O	LCD segment output 27 at LQFP128
2	1		PB.14	I/O	General purpose digital I/O pin
			INT0	I	External interrupt0 input pin
			SC2_CD	I	SmartCard2 card detect
			SPI2_SS1	I/O	SPI2 2nd slave select pin
			LCD_SEG12	O	LCD segment output 12 at LQFP64
			LCD_SEG26	O	LCD segment output 26 at LQFP128
3	2		PB.13	I/O	General purpose digital I/O pin
			EBI_AD1	I/O	EBI Address/Data bus bit1
			LCD_SEG11	O	LCD segment output 11 at LQFP64
			LCD_SEG25	O	LCD segment output 25 at LQFP128
4	3		PB.12	I/O	General purpose digital I/O pin
			EBI_AD0	I/O	EBI Address/Data bus bit0
			FCLKO	O	Frequency Divider output pin
			LCD_SEG10	O	LCD segment output 10 at LQFP64
			LCD_SEG24	O	LCD segment output 24 at LQFP128
5					NC
6	4		X32O	O	External 32.768 kHz crystal output pin
7	5		X32I	I	External 32.768 kHz crystal input pin
8					NC
9	6		PA.11	I/O	General purpose digital I/O pin
			I2C1_SCL	I/O	I^2C1 clock pin
			EBI_nRD	O	EBI read enable output pin
			SC0_RST	O	SmartCard0 RST pin
			SPI2_MOSI0	I/O	SPI2 1st MOSI (Master Out, Slave In) pin
			LCD_SEG9	O	LCD segment output 9 at LQFP64
			LCD_SEG23	O	LCD segment output 23 at LQFP128
10	7		PA.10	I/O	General purpose digital I/O pin
			I2C1_SDA	I/O	I^2C1 data I/O pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			EBI_nWR	O	EBI write enable output pin
			SC0_PWR	O	SmartCard0 Power pin
			SPI2_MISO0	I/O	SPI2 1st MISO (Master In, Slave Out) pin
			LCD_SEG8	O	LCD segment output 8 at LQFP64
			LCD_SEG22	O	LCD segment output 22 at LQFP128
11	8		PA.9	I/O	General purpose digital I/O pin
			I2C0_SCL	I/O	I^2C0 clock pin
			SC0_DAT	I/O	SmartCard0 DATA pin(SC0_UART_RXD)
			SPI2_CLK	I/O	SPI2 serial clock pin
			LCD_SEG7	O	LCD segment output 7 at LQFP64
			LCD_SEG21	O	LCD segment output 21 at LQFP128
12	9		PA.8	I/O	General purpose digital I/O pin
			I2C0_SDA	I/O	I^2C0 data I/O pin
			SC0_CLK	O	SmartCard0 clock pin(SC0_UART_TXD)
			SPI2_SS0	I/O	SPI2 1st slave select pin
			LCD_SEG6	O	LCD segment output 6 at LQFP64
			LCD_SEG20	O	LCD segment output 20 at LQFP128
13			PD.8	I/O	General purpose digital I/O pin
			LCD_SEG19	O	LCD segment output 19 at LQFP128
14			PD.9	I/O	General purpose digital I/O pin
			LCD_SEG18	O	LCD segment output 18 at LQFP128
15			PD.10	I/O	General purpose digital I/O pin
			LCD_SEG17	O	LCD segment output 17 at LQFP128
16			PD.11	I/O	General purpose digital I/O pin
			LCD_SEG16	O	LCD segment output 16 at LQFP128
17			PD.12	I/O	General purpose digital I/O pin
			LCD_SEG15	O	LCD segment output 15 at LQFP128
18			PD.13	I/O	General purpose digital I/O pin
			LCD_SEG14	O	LCD segment output 14 at LQFP128
19	10		PB.4	I/O	General purpose digital I/O pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			UART1_RXD	I	UART1 Data receiver input pin
			SC0_CD	I	SmartCard0 card detect pin
			SPI2_SS0	I/O	SPI2 1st slave select pin
			LCD_SEG5	O	LCD segment output 5 at LQFP64
			LCD_SEG13	O	LCD segment output 13 at LQFP128
20	11		PB.5	I/O	General purpose digital I/O pin
			UART1_TXD	O	UART1 Data transmitter output pin
			SC0_RST	O	SmartCard0 RST pin
			SPI2_CLK	I/O	SPI2 serial clock pin
			LCD_SEG4	O	LCD segment output 4 at LQFP64
			LCD_SEG12	O	LCD segment output 12 at LQFP128
21	12		PB.6	I/O	General purpose digital I/O pin
			UART1_RTSn	O	UART1 Request to Send output pin
			EBI_ALE	O	EBI address latch enable output pin
			SPI2_MISO0	I/O	SPI2 1st MISO (Master In, Slave Out) pin
			LCD_SEG3	O	LCD segment output 3 at LQFP64
			LCD_SEG11	O	LCD segment output 11 at LQFP128
22	13		PB.7	I/O	General purpose digital I/O pin
			UART1_CTSn	I	UART1 Clear to Send input pin
			EBI_nCS	O	EBI chip select enable output pin
			SPI2_MOSI0	I/O	SPI2 1st MOSI (Master Out, Slave In) pin
			LCD_SEG2	O	LCD segment output 2 at LQFP64
			LCD_SEG10	O	LCD segment output 10 at LQFP128
23					NC
24	14		LDO_CAP	P	LDO output pin
25					NC
26					NC
27	15		V_{DD}	P	Power supply for I/O ports and LDO source
28					NC
29	16		V_{SS}	P	Ground

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
30			V_{SS}	P	Ground
31			V_{SS}	P	Ground
32			V_{SS}	P	Ground
33			PE.12	I/O	General purpose digital I/O pin
			UART1_CTSn	I	UART1 Clear to Send input pin
34			PE.11	I/O	General purpose digital I/O pin
			UART1_RTSn	O	UART1 Request to Send output pin
35			PE.10	I/O	General purpose digital I/O pin
			UART1_TXD	O	UART1 Data transmitter output pin
36			PE.9	I/O	General purpose digital I/O pin
			UART1_RXD	I	UART1 Data receiver input pin
37			PE.8	I/O	General purpose digital I/O pin
			LCD_SEG9	O	LCD segment output 9 at LQFP128
38			PE.7	I/O	General purpose digital I/O pin
			LCD_SEG8	O	LCD segment output 8 at LQFP128
39					NC
40					NC
41					NC
42					NC
43					NC
44	17		PB.0	I/O	General purpose digital I/O pin
			UART0_RXD	I	UART0 Data receiver input pin
			SPI1_MOSI0	I/O	SPI1 1st MOSI (Master Out, Slave In) pin
			LCD_SEG1	O	LCD segment output 1 at LQFP64 (or as LD_COM5)
			LCD_SEG7	O	LCD segment output 7 at LQFP128
45	18		PB.1	I/O	General purpose digital I/O pin
			UART0_TXD	O	UART0 Data transmitter output pin
			SPI1_MISO0	I/O	SPI1 1st MISO (Master In, Slave Out) pin
			LCD_SEG0	O	LCD segment output 0 at LQFP64 (or as LCD_COM4)

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			LCD_SEG6	O	LCD segment output 6 at LQFP128
46	19		PB.2	I/O	General purpose digital I/O pin
			UART0_RTSn	O	UART0 Request to Send output pin
			EBI_nWRL	O	EBI low byte write enable output pin
			SPI1_CLK	I/O	SPI1 serial clock pin
			LCD_COM3	O	LCD common output 3 at LQFP64
			LCD_SEG5	O	LCD segment output 5 at LQFP128
47	20		PB.3	I/O	General purpose digital I/O pin
			UART0_CTSn	I	UART0 Clear to Send input pin
			EBI_nWRH	O	EBI high byte write enable output pin
			SPI1_SS0	I/O	SPI1 1st slave select pin
			LCD_COM2	O	LCD common output 2 at LQFP64
			LCD_SEG4	O	LCD segment output 4 at LQFP128
48	21		PD.6	I/O	General purpose digital I/O pin
			LCD_SEG3	O	LCD segment output 3 at LQFP128
49	22		PD.7	I/O	General purpose digital I/O pin
			LCD_SEG2	O	LCD segment output 2 at LQFP128
50	23		PD.14	I/O	General purpose digital I/O pin
			LCD_SEG1	O	LCD segment output 1 at LQFP128 (or as LCD_COM5)
51	24		PD.15	I/O	General purpose digital I/O pin
			LCD_SEG0	O	LCD segment output 0 at LQFP128 (or as LCD_COM4)
52			PC.5	I/O	General purpose digital I/O pin
			SPI0_MOSI1	I/O	SPI0 2nd MOSI (Master Out, Slave In) pin
			LCD_COM3	O	LCD common output 3 at LQFP128
53			PC.4	I/O	General purpose digital I/O pin
			SPI0_MISO1	I/O	SPI0 2nd MISO (Master In, Slave Out) pin
			LCD_COM2	O	LCD common output 2 at LQFP128
54	25		PC.3	I/O	General purpose digital I/O pin
			SPI0_MOSI0	I/O	SPI0 1st MOSI (Master Out, Slave In) pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			I2S_DO	O	I^2S data output
			SC1_RST	O	SmartCard1 RST pin
			LCD_COM1	O	LCD common output 1 at LQFP64
			LCD_COM1	O	LCD common output 1 at LQFP128
55	26		PC.2	I/O	General purpose digital I/O pin
			SPI0_MISO0	I/O	SPI0 1st MISO (Master In, Slave Out) pin
			I2S_DI	I	I^2S data input
			SC1_PWR	O	SmartCard1 PWR pin
			LCD_COM0	O	LCD common output 0 at LQFP64
			LCD_COM0	O	LCD common output 0 at LQFP128
56	27		PC.1	I/O	General purpose digital I/O pin
			SPI0_CLK	I/O	SPI0 serial clock pin
			I2S_BCLK	I/O	I^2S bit clock pin
			SC1_DAT	I/O	SmartCard1 DATA pin(SC1_UART_RXD)
			LCD_DH2	O	LCD externl capacitor pin of charge pump circuit at LQFP64
			LCD_DH2	O	LCD externl capacitor pin of charge pump circuit at LQFP128
57	28		PC.0 / MCLKO	I/O	General purpose digital I/O pin / Module clock output pin
			SPI0_SS0	I/O	SPI0 1st slave select pin
			I2S_LRCLK	I/O	I^2S left right channel clock
			SC1_CLK	O	SmartCard1 clock pin(SC1_UART_TXD)
			LCD_DH1	O	LCD externl capacitor pin of charge pump circuit at LQFP64
			LCD_DH1	O	LCD externl capacitor pin of charge pump circuit at LQFP128
58			PE.6	I/O	General purpose digital I/O pin
59	29		LCD_VLCD	AO	LCD power supply pin
60					NC
61			PE.5	I/O	General purpose digital I/O pin
			PWM1_CH1	I/O	PWM1 Channel1 output
62	30		PB.11	I/O	General purpose digital I/O pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			PWM1_CH0	**I/O**	PWM1 Channel0 output
			TM3	**O**	Timer3 external counter input
			SC2_DAT	**I/O**	SmartCard2 DATA pin(SC2_UART_RXD)
			SPI0_MISO0	**I/O**	SPI0 1st MISO (Master In, Slave Out) pin
			LCD_V1	**O**	Unit voltage for LCD charge pump circuit at LQFP64
			LCD_V1	**O**	LCD Unit voltage for LCD charge pump circuit at LQFP128
63	31		PB.10	**I/O**	General purpose digital I/O pin
			SPI0_SS1	**I/O**	SPI0 2nd slave select pin
			TM2	**O**	Timer2 external counter input
			SC2_CLK	**O**	SmartCard2 clock pin(SC2_UART_TXD)
			SPI0_MOSI0	**I/O**	SPI0 1st MOSI (Master Out, Slave In) pin
			LCD_V2	**O**	LCD driver biasing voltage at LQFP64
			LCD_V2	**O**	LCD driver biasing voltage at LQFP128
64	32		PB.9	**I/O**	General purpose digital I/O pin
			SPI1_SS1	**I/O**	SPI1 2nd slave select pin
			TM1	**O**	Timer1 external counter input
			SC2_RST	**O**	SmartCard2 RST pin
			INT0	**I**	External interrupt0 input pin
			LCD_V3	**O**	LCD driver biasing voltage at LQFP64
			LCD_V3	**O**	LCD driver biasing voltage at LQFP128
65			PE.4	**I/O**	General purpose digital I/O pin
			SPI0_MOSI0	**I/O**	SPI0 1st MOSI (Master Out, Slave In) pin
66			PE.3	**I/O**	General purpose digital I/O pin
			SPI0_MISO0	**I/O**	SPI0 1st MISO (Master In, Slave Out) pin
67			PE.2	**I/O**	General purpose digital I/O pin
			SPI0_CLK	**I/O**	SPI0 serial clock pin
68			PE.1	**I/O**	General purpose digital I/O pin
			PWM1_CH3	**I/O**	PWM1 Channel3 output
			SPI0_SS0	**I/O**	SPI0 1st slave select pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
69			PE.0	I/O	General purpose digital I/O pin
			PWM1_CH2	I/O	PWM1 Channel2 output
			I2S_MCLK	O	I^2S master clock output pin
70			PC.13	I/O	General purpose digital I/O pin
			SPI1_MOSI1	I/O	SPI1 2nd MOSI (Master Out, Slave In) pin
			PWM1_CH1	I/O	PWM1 Channel1 output
			SNOOPER	I	Snooper pin
			INT1	I	External interrupt 1
			I2C0_SCL	O	I^2C0 clock pin
71			PC.12	I/O	General purpose digital I/O pin
			SPI1_MISO1	I/O	SPI1 2nd MISO (Master In, Slave Out) pin
			PWM1_CH0	I/O	PWM1 Channel0 output
			INT0	I	External interrupt0 input pin
			I2C0_SDA	I/O	I^2C0 data I/O pin
72	33		PC.11	I/O	General purpose digital I/O pin
			SPI1_MOSI0	I/O	SPI1 1st MOSI (Master Out, Slave In) pin
			UART1_TXD	O	UART1 Data transmitter output pin
			LCD_SEG31	O	LCD segment output 31 at LQFP64
73	34		PC.10	I/O	General purpose digital I/O pin
			SPI1_MISO0	I/O	SPI1 1st MISO (Master In, Slave Out) pin
			UART1_RXD	I	UART1 Data receiver input pin
			LCD_SEG30	O	LCD segment output 30 at LQFP64
74	35		PC.9	I/O	General purpose digital I/O pin
			SPI1_CLK	I/O	SPI1 serial clock pin
			I2C1_SCL	I/O	I^2C1 clock pin
			LCD_SEG29	O	LCD segment output 29 at LQFP64
75	36		PC.8	I/O	General purpose digital I/O pin
			SPI1_SS0	I/O	SPI1 1st slave select pin
			EBI_MCLK	O	EBI external clock output pin
			I2C1_SDA	I/O	I^2C1 data I/O pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			LCD_SEG28	O	LCD segment output 28 at LQFP64
76	37		PA.15	I/O	General purpose digital I/O pin
			PWM0_CH3	I/O	PWM0 Channel3 output
			I2S_MCLK	O	I^2S master clock output pin
			TC3	I	Timer3 capture input
			SC0_PWR	O	SmartCard0 Power pin
			UART0_TXD	O	UART0 Data transmitter output pin
			LCD_SEG27	O	LCD segment output 27 at LQFP64
77	38		PA.14	I/O	General purpose digital I/O pin
			PWM0_CH2	I/O	PWM0 Channel2 output
			EBI_AD15	I/O	EBI Address/Data bus bit15
			TC2	I	Timer2 capture input
			UART0_RXD	I	UART0 Data receiver input pin
			LCD_SEG26	O	LCD segment output 26 at LQFP64
78	39		PA.13	I/O	General purpose digital I/O pin
			PWM0_CH1	I/O	PWM0 Channel1 output
			EBI_AD14	I/O	EBI Address/Data bus bit14
			TC1	I	Timer1 capture input
			I2C0_SCL	I/O	I^2C0 clock pin
			LCD_SEG25	O	LCD segment output 25 at LQFP64
79	40		PA.12	I/O	General purpose digital I/O pin
			PWM0_CH0	I/O	PWM0 Channel0 output
			EBI_AD13	I/O	EBI Address/Data bus bit13
			TC0	I	Timer0 capture input
			I2C0_SDA	I/O	I^2C0 data I/O pin
			LCD_SEG24	O	LCD segment output 24 at LQFP64
80	41		ICE_DAT	I/O	Serial Wired Debugger Data pin **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_DAT pin.
			PF.0	I/O	General purpose digital I/O pin
			INT0	I	External interrupt0 input pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
81	42		ICE_CLK	I	Serial Wired Debugger Clock pin **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_CLK pin.
			PF.1	I/O	General purpose digital I/O pin
			FCLKO	O	Frequency Divider output pin
			INT1	I	External interrupt1 input pin
82					NC
83			V_{DD}	P	Power supply for I/O ports and LDO source for internal PLL and digital circuit
84					NC
85			V_{SS}	P	Ground
86			V_{SS}	P	Ground
87	43		AV_{SS}	AP	Ground Pin for analog circuit
88			AV_{SS}	AP	Ground Pin for analog circuit
89	44		PA.0	I/O	General purpose digital I/O pin
			AD0	AI	ADC analog input0
			SC2_CD	I	SmartCard2 card detect
90	45		PA.1	I/O	General purpose digital I/O pin
			AD1	AI	ADC analog input1
			EBI_AD12	I/O	EBI Address/Data bus bit12
91	46		PA.2	I/O	General purpose digital I/O pin
			AD2	AI	ADC analog input2
			EBI_AD11	I/O	EBI Address/Data bus bit11
			UART1_RXD	I	UART1 Data receiver input pin
			LCD_SEG23*	AO	LCD segment output 23 at LQFP64
92	47		PA.3	I/O	General purpose digital I/O pin
			AD3	AI	ADC analog input3
			EBI_AD10	I/O	EBI Address/Data bus bit10
			UART1_TXD	O	UART1 Data transmitter output pin
			LCD_SEG22*	AO	LCD segment output 22 at LQFP64
93	48		PA.4	I/O	General purpose digital I/O pin
			AD4	AI	ADC analog input4

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			EBI_AD9	**I/O**	EBI Address/Data bus bit9
			SC2_PWR	**O**	SmartCard2 Power pin
			I2C0_SDA	**I/O**	I^2C0 data I/O pin
			LCD_SEG21*	**AO**	LCD segment output 21 at LQFP64
			LCD_SEG39*	**AO**	LCD segment output 39 at LQFP128
94	49		PA.5	**I/O**	General purpose digital I/O pin
			AD5	**AI**	ADC analog input5
			EBI_AD8	**I/O**	EBI Address/Data bus bit8
			SC2_RST	**O**	SmartCard2 RST pin
			I2C0_SCL	**I/O**	I^2C0 clock pin
			LCD_SEG20*	**AO**	LCD segment output 19 at LQFP64
			LCD_SEG38*	**AO**	LCD segment output 37 at LQFP128
95	50		PA.6	**I/O**	General purpose digital I/O pin
			AD6	**AI**	ADC analog input6
			EBI_AD7	**I/O**	EBI Address/Data bus bit7
			TC3	**I**	Timer3 capture input
			SC2_CLK	**O**	SmartCard2 clock pin(SC2_UART_TXD)
			PWM0_CH3	**I/O**	PWM0 Channel3 output
			LCD_SEG19*	**AO**	LCD segment output 19 at LQFP64
			LCD_SEG37*	**AO**	LCD segment output 37 at LQFP128
96			PA.7	**I/O**	General purpose digital I/O pin
			AD7	**AI**	ADC analog input7
			EBI_AD6	**I/O**	EBI Address/Data bus bit6
			TC2	**I**	Timer2 capture input
			SC2_DAT	**I/O**	SmartCard2 DATA pin(SC2_UART_RXD)
			PWM0_CH2	**I/O**	PWM0 Channel2 output
			LCD_SEG36*	**AO**	LCD segment output 36 output at LQFP128
97	51		V_{REF}	**AP**	Voltage reference input for ADC
98					NC
99	52		AV_{DD}	**AP**	Power supply for internal analog circuit

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
100			PD.0	I/O	General purpose digital I/O pin
			UART1_RXD	I	UART1 Data receiver input pin
			SPI2_SS0	I/O	SPI2 1st slave select pin
			SC1_CLK	O	SmartCard1 clock pin(SC1_UART_TXD)
			AD8	AI	ADC analog input8
101			PD.1	I/O	General purpose digital I/O pin
			UART1_TXD	O	UART1 Data transmitter output pin
			SPI2_CLK	I/O	SPI2 serial clock pin
			SC1_DAT	I/O	SmartCard1 DATA pin(SC1_UART_RXD)
			AD9	AI	ADC analog input9
102			PD.2	I/O	General purpose digital I/O pin
			UART1_RTSn		UART1 Request to Send output pin
			I2S_LRCLK	I/O	I^2S left right channel clock
			SPI2_MISO0	I/O	SPI2 1st MISO (Master In, Slave Out) pin
			SC1_PWR	O	SmartCard1 Power pin
			AD10	AI	ADC analog input10
103			PD.3	I/O	General purpose digital I/O pin
			UART1_CTSn		UART1 Clear to Send input pin
			I2S_BCLK	I/O	I^2S bit clock pin
			SPI2_MOSI0	I/O	SPI2 1st MOSI (Master Out, Slave In) pin
			SC1_RST	O	SmartCard1 RST pin
			AD11	AI	ADC analog input11
104					NC
105			PD.4	I/O	General purpose digital I/O pin
			I2S_DI	I	I^2S data input
			SPI2_MISO1	I/O	SPI2 2nd MISO (Master In, Slave Out) pin
			SC1_CD	I	SmartCard1 card detect
			LCD_SEG35	AO	LCD segment output 35 at LQFP10
106			PD.5	I/O	General purpose digital I/O pin
			I2S_DO	O	I^2S data output

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			SPI2_MOSI1	I/O	SPI2 2nd MOSI (Master Out, Slave In) pin
			LCD_SEG34	AO	LCD segment output 34 at LQFP128
107	53		PC.7	I/O	General purpose digital I/O pin
			DA1_OUT	AO	DAC 1 output
			EBI_AD5	I/O	EBI Address/Data bus bit5
			TC1	I	Timer1 capture input
			PWM0_CH1	I/O	PWM0 Channel1 output
			LCD_SEG17*	AO	LCD segment output 17 at LQFP64
108	54		PC.6	I/O	General purpose digital I/O pin
			DA0_OUT	I	DAC0 output
			EBI_AD4	I/O	EBI Address/Data bus bit4
			TC0	I	Timer0 capture input
			SC1_CD	I	SmartCard1 card detect pin
			PWM0_CH0	I/O	PWM0 Channel0 output
109	55		PC.15	I/O	General purpose digital I/O pin
			EBI_AD3	I/O	EBI Address/Data bus bit3
			TC0	I	Timer0 capture input
			PWM1_CH2	I/O	PWM1 Channel2 output
			LCD_SEG16	AO	LCD segment output 16 at LQFP64
			LCD_SEG33	AO	LCD segment output 33 at LQFP128
110	56		PC.14	I/O	General purpose digital I/O pin
			EBI_AD2	I/O	EBI Address/Data bus bit2
			PWM1_CH3	I/O	PWM1 Channel3 output
			LCD_SEG15	AO	LCD segment output 15 at LQFP64
			LCD_SEG32	AO	LCD segment output 32 at LQFP128
111	57		PB.15	I/O	General purpose digital I/O pin
			INT1	I	External interrupt1 input pin
			SNOOPER	I	Snooper pin
			LCD_SEG14	AO	LCD segment output 14 at LQFP64
			LCD_SEG31	AO	LCD segment output 31 at LQFP128

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
112					NC
113	58		XT1_IN	**O**	External 4~24 MHz crystal output pin
			PF.3	**I/O**	General purpose digital I/O pin
114	59		XT1_OUT	**I**	External 4~24 MHz crystal input pin
			PF.2	**I/O**	General purpose digital I/O pin
115					NC
116	60		nRESET	**I**	External reset input: Low active, set this pin low reset chip to initial state. With internal pull-up. **Note:** It is recommended to use 10 kΩ pull-up resistor and 10 μF capacitor on nRESET pin.
117	61		V_{SS}	**P**	Ground
118			V_{SS}	**P**	Ground
119					NC
120	62		V_{DD}	**P**	Power supply for I/O ports and LDO source for internal PLL and digital circuit
121					NC
122			PF.4	**I/O**	General purpose digital I/O pin
			I2C0_SDA	**I/O**	I^2C0 data I/O pin
123			PF.5	**I/O**	General purpose digital I/O pin
			I2C0_SCL	**I/O**	I^2C0 clock pin
124			V_{SS}	**P**	Ground
125	63		PV_{SS}	**P**	PLL Ground
126	64		PB.8	**I/O**	General purpose digital I/O pin
			STADC	**I**	ADC external trigger input.
			TM0	**I**	Timer0 external counter input
			INT0	**I**	External interrupt0 input pin
			SC2_PWR	**O**	SmartCard2 Power pin
			LCD_SEG13	**AO**	LCD segment output 13 at LQFP64
			LCD_SEG30	**AO**	LCD segment output 30 at LQFP128
127			PE.15	**I/O**	General purpose digital I/O pin
			LCD_SEG29	**O**	LCD segment output 29 at LQFP128
128			PE.14	**I/O**	General purpose digital I/O pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			LCD_SEG28	O	LCD segment output 28 at LQFP128

Note:

1. Pin Type: I = Digital Input, O=Digital Output; AI=Analog Input; AO= Analog Output; P=Power Pin; AP=Analog Power;
2. * : Output voltage for ADC/LCD shared pins cannot be higher than V_{DD} because these pins are without 5V tolerance.

3.4.3 NuMicro® Nano120 Pin Description

Pin No.			Pin Name	Pin Type	Description
LQFP 128	LQFP 64	LQFP 48			
1			PE.13	I/O	General purpose digital IO pin
2	1		PB.14	I/O	General purpose digital IO pin
			INT0	I	External interrupt0 input pin
			SC2_CD	I	SmartCard2 card detect
			SPI2_SS1	I/O	SPI2 2nd slave select pin
3	2		PB.13	I/O	General purpose digital IO pin
			EBI_AD1	I/O	EBI Address/Data bus bit1
4	3	1	PB.12	I/O	General purpose digital IO pin
			EBI_AD0	I/O	EBI Address/Data bus bit0
			FCLKO	O	Frequency Divider output pin
5					NC
6	4	2	X32O	O	External 32.768 kHz crystal output pin
7	5	3	X32I	I	External 32.768 kHz crystal input pin
8					NC
9	6	4	PA.11	I/O	General purpose digital IO pin
			I2C1_SCL	I/O	I^2C 1 clock pin
			EBI_nRD	O	EBI read enable output pin
			SC0_RST	O	SmartCard0 RST pin
			SPI2_MOSI0	I/O	SPI2 1st MOSI (Master Out, Slave In) pin
10	7	5	PA.10	I/O	General purpose digital IO pin
			I2C1_SDA	I/O	I^2C 1 data I/O pin
			EBI_nWR	O	EBI write enable output pin
			SC0_PWR	O	SmartCard0 Power pin
			SPI2_MISO0	I/O	SPI2 1st MISO (Master In, Slave Out) pin
11	8	6	PA.9	I/O	General purpose digital IO pin
			I2C0_SCL	I/O	I^2C 0 clock pin
			SC0_DAT	I/O	SmartCard0 DATA pin(SC0_UART_RXD)
			SPI2_CLK	I/O	SPI2 serial clock pin
12	9	7	PA.8	I/O	General purpose digital IO pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128	LQFP 64	LQFP 48			
			I2C0_SDA	**I/O**	I^2C 0 data I/O pin
			SC0_CLK	**O**	SmartCard0 clock pin(SC0_UART_TXD)
			SPI2_SS0	**I/O**	SPI2 1st slave select pin
13			PD.8	**I/O**	General purpose digital IO pin
14			PD.9	**I/O**	General purpose digital IO pin
15			PD.10	**I/O**	General purpose digital IO pin
16			PD.11	**I/O**	General purpose digital IO pin
17			PD.12	**I/O**	General purpose digital IO pin
18			PD.13	**I/O**	General purpose digital IO pin
19	10	8	PB.4	**I/O**	General purpose digital IO pin
			UART1_RXD	**I**	UART1 Data receiver input pin
			SC0_CD	**I**	SmartCard0 card detect pin
			SPI2_SS0	**I/O**	SPI2 1st slave select pin
20	11	9	PB.5	**I/O**	General purpose digital IO pin
			UART1_TXD	**O**	UART1 Data transmitter output pin
			SC0_RST	**O**	SmartCard0 RST pin
			SPI2_CLK	**I/O**	SPI2 serial clock pin
21	12		PB.6	**I/O**	General purpose digital IO pin
			UART1_nRTS	**O**	UART1 Request to Send output pin
			EBI_ALE	**O**	EBI address latch enable output pin
			SPI2_MISO0	**I/O**	SPI2 1st MISO (Master In, Slave Out) pin
22	13		PB.7	**I/O**	General purpose digital IO pin
			UART1_nCTS	**I**	UART1 Clear to Send input pin
			EBI_nCS	**O**	EBI chip select enable output pin
			SPI2_MOSI0	**I/O**	SPI2 1st MOSI (Master Out, Slave In) pin
23					NC
24	14	10	LDO_CAP	**P**	LDO output pin
25					NC
26					NC
27	15	11	V_{DD}	**P**	Power supply for I/O ports and LDO source

Pin No.			Pin Name	Pin Type	Description
LQFP 128	LQFP 64	LQFP 48			
28					NC
29	16	12	V_{SS}	P	Ground
30			V_{SS}	P	Ground
31			V_{SS}	P	Ground
32			V_{SS}	P	Ground
33			PE.12	I/O	General purpose digital IO pin
34			PE.11	I/O	General purpose digital IO pin
35			PE.10	I/O	General purpose digital IO pin
36			PE.9	I/O	General purpose digital IO pin
37			PE.8	I/O	General purpose digital IO pin
38			PE.7	I/O	General purpose digital IO pin
39					NC
40	17	13	USB_VBUS	USB	POWER SUPPLY: From USB Host or HUB.
41	18	14	USB_VDD33_CAP	USB	Internal Power Regulator Output 3.3V Decoupling Pin
42	19	15	USB_D-	USB	USB Differential Signal D-
43	20	16	USB_D+	USB	USB Differential Signal D+
44	21	17	PB.0	I/O	General purpose digital IO pin
			UART0_RXD	I	UART0 Data receiver input pin
			SPI1_MOSI0	I/O	SPI1 1st MOSI (Master Out, Slave In) pin
45	22	18	PB.1	I/O	General purpose digital IO pin
			UART0_TXD	O	UART0 Data transmitter output pin
			SPI1_MISO0	I/O	SPI1 1st MISO (Master In, Slave Out) pin
46	23	19	PB.2	I/O	General purpose digital IO pin
			UART0_nRTS	O	UART0 Request to Send output pin
			EBI_nWRL	O	EBI low byte write enable output pin
			SPI1_CLK	I/O	SPI1 serial clock pin
47	24	20	PB.3	I/O	General purpose digital IO pin
			UART0_nCTS	I	UART0 Clear to Send input pin
			EBI_nWRH	O	EBI high byte write enable output pin
			SPI1_SS0	I/O	SPI1 1st slave select pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128	LQFP 64	LQFP 48			
48			PD.6	I/O	General purpose digital IO pin
49			PD.7	I/O	General purpose digital IO pin
50			PD.14	I/O	General purpose digital IO pin
51			PD.15	I/O	General purpose digital IO pin
52			PC.5	I/O	General purpose digital IO pin
			SPI0_MOSI1	I/O	SPI0 2nd MOSI (Master Out, Slave In) pin
53			PC.4	I/O	General purpose digital IO pin
			SPI0_MISO1	I/O	SPI0 2nd MISO (Master In, Slave Out) pin
54	25	21	PC.3	I/O	General purpose digital IO pin
			SPI0_MOSI0	I/O	SPI0 1st MOSI (Master Out, Slave In) pin
			I2S_DO	O	I^2S data output
			SC1_RST	O	SmartCard1 RST pin
55	26	22	PC.2	I/O	General purpose digital IO pin
			SPI0_MISO0	I/O	SPI0 1st MISO (Master In, Slave Out) pin
			I2S_DI	I	I^2S data input
			SC1_PWR	O	SmartCard1 PWR pin
56	27	23	PC.1	I/O	General purpose digital IO pin
			SPI0_CLK	I/O	SPI0 serial clock pin
			I2S_BCLK	I/O	I^2S bit clock pin
			SC1_DAT	I/O	SmartCard1 DATA pin(SC1_UART_RXD)
57	28	24	PC.0 / MCLKO	I/O	General purpose digital IO pin / Module clock output pin
			SPI0_SS0	I/O	SPI0 1st slave select pin
			I2S_LRCLK	I/O	I^2S left right channel clock
			SC1_CLK	O	SmartCard1 clock pin(SC1_UART_TXD)
58			PE.6	I/O	General purpose digital IO pin
59					NC
60					NC
61	29		PE.5	I/O	General purpose digital IO pin
			PWM1_CH1	I/O	PWM1 Channel1 output
62	30		PB.11	I/O	General purpose digital IO pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128	LQFP 64	LQFP 48			
			PWM1_CH0	I/O	PWM1 Channel0 output
			TM3	O	Timer3 external counter input
			SC2_DAT	I/O	SmartCard2 DATA pin(SC2_UART_RXD)
			SPI0_MISO0	I/O	SPI0 1st MISO (Master In, Slave Out) pin
63	31		PB.10	I/O	General purpose digital IO pin
			SPI0_SS1	I/O	SPI0 2nd slave select pin
			TM2	O	Timer2 external counter input
			SC2_CLK	O	SmartCard2 clock pin(SC2_UART_TXD)
			SPI0_MOSI0	I/O	SPI0 1st MOSI (Master Out, Slave In) pin
64	32		PB.9	I/O	General purpose digital IO pin
			SPI1_SS1	I/O	SPI1 2nd slave select pin
			TM1	O	Timer1 external counter input
			SC2_RST	O	SmartCard2 RST pin
			INT0	I	External interrupt0 input pin
65			PE.4	I/O	General purpose digital IO pin
			SPI0_MOSI0	I/O	SPI0 1st MOSI (Master Out, Slave In) pin
66			PE.3	I/O	General purpose digital IO pin
			SPI0_MISO0	I/O	SPI0 1st MISO (Master In, Slave Out) pin
67			PE.2	I/O	General purpose digital IO pin
			SPI0_CLK	I/O	SPI0 serial clock pin
68			PE.1	I/O	General purpose digital IO pin
			PWM1_CH3	I/O	PWM1 Channel3 output
			SPI0_SS0	I/O	SPI0 1st slave select pin
69			PE.0	I/O	General purpose digital IO pin
			PWM1_CH2	I/O	PWM1 Channel2 output
			I2S_MCLK	O	I^{2}S master clock output pin
70			PC.13	I/O	General purpose digital IO pin
			SPI1_MOSI1	I/O	SPI1 2nd MOSI (Master Out, Slave In) pin
			PWM1_CH1	I/O	PWM1 Channel1 output
			SNOOPER	I	Snooper pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128	LQFP 64	LQFP 48			
			INT1	I	External interrupt 1 input pin
			I2C0_SCL	O	I^2C 0 clock pin
71			PC.12	I/O	General purpose digital IO pin
			SPI1_MISO1	I/O	SPI1 2nd MISO (Master In, Slave Out) pin
			PWM1_CH0	I/O	PWM1 Channel 0 output
			INT0	I	External interrupt 0 input pin
			I2C0_SDA	I/O	I^2C 0 data I/O pin
72	33		PC.11	I/O	General purpose digital IO pin
			SPI1_MOSI0	I/O	SPI1 1st MOSI (Master Out, Slave In) pin
			UART1_TXD	O	UART1 Data transmitter output pin
73	34		PC.10	I/O	General purpose digital IO pin
			SPI1_MISO0	I/O	SPI1 1st MISO (Master In, Slave Out) pin
			UART1_RXD	I	UART1 Data receiver input pin
74	35		PC.9	I/O	General purpose digital IO pin
			SPI1_CLK	I/O	SPI1 serial clock pin
			I2C1_SCL	I/O	I^2C 1 clock pin
75	36		PC.8	I/O	General purpose digital IO pin
			SPI1_SS0	I/O	SPI1 1st slave select pin
			EBI_MCLK	O	EBI external clock output pin
			I2C1_SDA	I/O	I^2C 1 data I/O pin
76	37	25	PA.15	I/O	General purpose digital IO pin
			PWM0_CH3	I/O	PWM0 Channel3 output
			I2S_MCLK	O	I^2S master clock output pin
			TC3	I	Timer3 capture input
			SC0_PWR	O	SmartCard0 Power pin
			UART0_TXD	O	UART0 Data transmitter output pin
77	38	26	PA.14	I/O	General purpose digital IO pin
			PWM0_CH2	I/O	PWM0 Channel2 output
			EBI_AD15	I/O	EBI Address/Data bus bit15
			TC2	I	Timer 2 capture input

Pin No.			Pin Name	Pin Type	Description
LQFP 128	LQFP 64	LQFP 48			
			UART0_RXD	**I**	UART0 Data receiver input pin
78	39	27	PA.13	**I/O**	General purpose digital IO pin
			PWM0_CH1	**I/O**	PWM0 Channel1 output
			EBI_AD14	**I/O**	EBI Address/Data bus bit14
			TC1	**I**	Timer1 capture input
			I2C0_SCL	**I/O**	I^2C 0 clock pin
79	40	28	PA.12	**I/O**	General purpose digital IO pin
			PWM0_CH0	**I/O**	PWM0 Channel0 output
			EBI_AD13	**I/O**	EBI Address/Data bus bit13
			TC0	**I**	Timer 0 capture input
			I2C0_SDA	**I/O**	I^2C 0 data I/O pin
80	41	29	ICE_DAT	**I/O**	Serial Wired Debugger Data pin **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_DAT pin.
			PF.0	**I/O**	General purpose digital IO pin
			INT0	**I**	External interrupt0 input pin
81	42	30	ICE_CLK	**I**	Serial Wired Debugger Clock pin **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_CLK pin.
			PF.1	**I/O**	General purpose digital IO pin
			FCLKO	**O**	Frequency Divider output pin
			INT1	**I**	External interrupt1 input pin
82					NC
83			V_{DD}	**P**	Power supply for I/O ports and LDO source for internal PLL and digital circuit
84					NC
85			V_{SS}	**P**	Ground
86			V_{SS}	**P**	Ground
87	43	31	AV_{SS}	**AP**	Ground Pin for analog circuit
88			AV_{SS}	**AP**	Ground Pin for analog circuit
89	44	32	PA.0	**I/O**	General purpose digital IO pin
			AD0	**AI**	ADC analog input0

Pin No.			Pin Name	Pin Type	Description
LQFP 128	LQFP 64	LQFP 48			
			SC2_CD	I	SmartCard2 card detect
90	45	33	PA.1	I/O	General purpose digital IO pin
			AD1	AI	ADC analog input1
			EBI_AD12	I/O	EBI Address/Data bus bit12
91	46	34	PA.2	I/O	General purpose digital IO pin
			AD2	AI	ADC analog input2
			EBI_AD11	I/O	EBI Address/Data bus bit11
			UART1_RXD	I	UART1 Data receiver input pin
92	47	35	PA.3	I/O	General purpose digital IO pin
			AD3	AI	ADC analog input3
			EBI_AD10	I/O	EBI Address/Data bus bit10
			UART1_TXD	O	UART1 Data transmitter output pin
93	48	36	PA.4	I/O	Digital GPIO pin
			AD4	AI	ADC analog input4
			EBI_AD9	I/O	EBI Address/Data bus bit9
			SC2_PWR	O	SmartCard2 Power pin
			I2C0_SDA	I/O	I^2C 0 data I/O pin
94	49	37	PA.5	I/O	General purpose digital IO pin
			AD5	AI	ADC analog input5
			EBI_AD8	I/O	EBI Address/Data bus bit8
			SC2_RST	O	SmartCard2 RST pin
			I2C0_SCL	I/O	I^2C 0 clock pin
95	50	38	PA.6	I/O	General purpose digital IO pin
			AD6	AI	ADC analog input6
			EBI_AD7	I/O	EBI Address/Data bus bit7
			TC3	I	Timer3 capture input
			SC2_CLK	O	SmartCard2 clock pin(SC2_UART_TXD)
			PWM0_CH3	I/O	PWM0 Channel3 output
96			PA.7	I/O	General purpose digital IO pin
			AD7	AI	ADC analog input7

Pin No.			Pin Name	Pin Type	Description
LQFP 128	LQFP 64	LQFP 48			
			EBI_AD6	**I/O**	EBI Address/Data bus bit6
			TC2	**I**	Timer2 capture input
			SC2_DAT	**I/O**	SmartCard2 DATA pin(SC2_UART_RXD)
			PWM0_CH2	**I/O**	PWM0 Channel2 output
97	51	39	V_{REF}	**AP**	Voltage reference input for ADC
98					NC
99	52	40	AV_{DD}	**AP**	Power supply for internal analog circuit
100			PD.0	**I/O**	General purpose digital IO pin
			UART1_RXD	**I**	UART1 Data receiver input pin
			SPI2_SS0	**I/O**	SPI2 1^{st} slave select pin
			SC1_CLK	**O**	SmartCard1 clock pin(SC1_UART_TXD)
			AD8	**AI**	ADC analog input8
101			PD.1	**I/O**	General purpose digital IO pin
			UART1_TXD	**O**	UART1 Data transmitter output pin
			SPI2_CLK	**I/O**	SPI2 serial clock pin
			SC1_DAT	**I/O**	SmartCard1 DATA pin(SC1_UART_RXD)
			AD9	**AI**	ADC analog input9
102			PD.2	**I/O**	General purpose digital IO pin
			UART1_nRTS	**O**	UART1 Request to Send output pin
			I2S_LRCLK	**I/O**	I^2S left right channel clock
			SPI2_MISO0	**I/O**	SPI2 1^{st} MISO (Master In, Slave Out) pin
			SC1_PWR	**O**	SmartCard1 Power pin
			AD10	**AI**	ADC analog input10
103			PD.3	**I/O**	General purpose digital IO pin
			UART1_nCTS	**I**	UART1 Clear to Send input pin
			I2S_BCLK	**I/O**	I^2S bit clock pin
			SPI2_MOSI0	**I/O**	SPI2 1^{st} MOSI (Master Out, Slave In) pin
			SC1_RST	**O**	SmartCard1 RST pin
			AD11	**AI**	ADC analog input11
104					NC

Pin No.			Pin Name	Pin Type	Description
LQFP 128	LQFP 64	LQFP 48			
105			PD.4	I/O	General purpose digital IO pin
			I2S_DI	I	I^2S data input
			SPI2_MISO1	I/O	SPI2 2^{nd} MISO (Master In, Slave Out) pin
			SC1_CD	I	SmartCard1 card detect
106			PD.5	I/O	General purpose digital IO pin
			I2S_DO	O	I^2S data output
			SPI2_MOSI1	I/O	SPI2 2^{nd} MOSI (Master Out, Slave In) pin
107	53	41	PC.7	I/O	General purpose digital IO pin
			DA1_OUT	AO	DAC 1 output
			EBI_AD5	I/O	EBI Address/Data bus bit5
			TC1	I	Timer1 capture input
			PWM0_CH1	I/O	PWM0 Channel1 output
108	54	42	PC.6	I/O	General purpose digital IO pin
			DA0_OUT	I	DAC0 output
			EBI_AD4	I/O	EBI Address/Data bus bit4
			TC0	I	Timer 0 capture input
			SC1_CD		SmartCard1 card detect pin
			PWM0_CH0	I/O	PWM0 Channel0 output
109	55		PC.15	I/O	General purpose digital IO pin
			EBI_AD3	I/O	EBI Address/Data bus bit3
			TC0	I	Timer0 capture input
			PWM1_CH2	I/O	PWM1 Channel2 output
110	56		PC.14	I/O	General purpose digital IO pin
			EBI_AD2	I/O	EBI Address/Data bus bit2
			PWM1_CH3	I/O	PWM1 Channel3 output
111	57	43	PB.15	I/O	General purpose digital IO pin
			INT1	I	External interrupt1 input pin
			SNOOPER	I	Snooper pin
			SC1_CD	I	SmartCard1 card detect
112					NC

Pin No.			Pin Name	Pin Type	Description
LQFP 128	LQFP 64	LQFP 48			
113	58	44	XT1_IN	O	External 4~24 MHz crystal output pin
			PF.3	I/O	General purpose digital I/O pin
114	59	45	XT1_OUT	I	External 4~24 MHz crystal input pin
			PF.2	I/O	General purpose digital I/O pin
115					NC
116	60	46	nRESET	I	External reset input: Low active, set this pin low reset chip to initial state. With internal pull-up. **Note:** It is recommended to use 10 kΩ pull-up resistor and 10 μF capacitor on nRESET pin.
117	61		V_{SS}	P	Ground
118			V_{SS}	P	Ground
119					NC
120	62		V_{DD}	P	Power supply for I/O ports and LDO source for internal PLL and digital circuit
121					NC
122			PF.4	I/O	General purpose digital IO pin
			I2C0_SDA	I/O	I^2C 0 data I/O pin
123			PF.5	I/O	General purpose digital IO pin
			I2C0_SCL	I/O	I^2C 0 clock pin
124			V_{SS}	P	Ground
125	63	47	PV_{SS}	P	PLL Ground
126	64	48	PB.8	I/O	General purpose digital IO pin
			STADC	I	ADC external trigger input.
			TM0	I	Timer0 external counter input
			INT0	I	External interrupt0 input pin
			SC2_PWR	O	SmartCard2 Power pin
127			PE.15	I/O	General purpose digital IO pin
128			PE.14	I/O	General purpose digital IO pin

Note:

1. Pin Type: I = Digital Input, O=Digital Output; AI=Analog Input; AO= Analog Output; P=Power Pin; AP=Analog Power;

3.4.4 NuMicro® Nano130 Pin Description

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
1			PE.13	I/O	General purpose digital I/O pin
			LCD_SEG27	O	LCD segment output 27 at LQFP128
2	1		PB.14	I/O	General purpose digital I/O pin
			INT0	I	External interrupt0 input pin
			SC2_CD	I	SmartCard2 card detect
			SPI2_SS1	I/O	SPI2 2nd slave select pin
			LCD_SEG12	O	LCD segment output 12 at LQFP64
			LCD_SEG26	O	LCD segment output 26 at LQFP128
3	2		PB.13	I/O	General purpose digital I/O pin
			EBI_AD1	I/O	EBI Address/Data bus bit1
			LCD_SEG11	O	LCD segment output 11 at LQFP64
			LCD_SEG25	O	LCD segment output 25 at LQFP128
4	3		PB.12	I/O	General purpose digital I/O pin
			EBI_AD0	I/O	EBI Address/Data bus bit0
			FCLKO	O	Frequency Divider output pin
			LCD_SEG10	O	LCD segment output 10 at LQFP64
			LCD_SEG24	O	LCD segment output 24 at LQFP128
5					NC
6	4		X32O	O	External 32.768 kHz crystal output pin
7	5		X32I	I	External 32.768 kHz crystal input pin
8					NC
9	6		PA.11	I/O	General purpose digital I/O pin
			I2C1_SCL	I/O	I^{2}C1 clock pin
			EBI_nRD	O	EBI read enable output pin
			SC0_RST	O	SmartCard0 RST pin
			SPI2_MOSI0	I/O	SPI2 1st MOSI (Master Out, Slave In) pin
			LCD_SEG9	O	LCD segment output 9 at LQFP64
			LCD_SEG23	O	LCD segment output 23 at LQFP128
10	7		PA.10	I/O	General purpose digital I/O pin
			I2C1_SDA	I/O	I^{2}C1 data I/O pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			EBI_nWR	O	EBI write enable output pin
			SC0_PWR	O	SmartCard0 Power pin
			SPI2_MISO0	I/O	SPI2 1st MISO (Master In, Slave Out) pin
			LCD_SEG8	O	LCD segment output 8 at LQFP64
			LCD_SEG22	O	LCD segment output 22 at LQFP128
11	8		PA.9	I/O	General purpose digital I/O pin
			I2C0_SCL	I/O	I^2C0 clock pin
			SC0_DAT	I/O	SmartCard0 DATA pin(SC0_UART_RXD)
			SPI2_CLK	I/O	SPI2 serial clock pin
			LCD_SEG7	O	LCD segment output 7 at LQFP64
			LCD_SEG21	O	LCD segment output 21 at LQFP128
12	9		PA.8	I/O	General purpose digital I/O pin
			I2C0_SDA	I/O	I^2C0 data I/O pin
			SC0_CLK	O	SmartCard0 clock pin(SC0_UART_TXD)
			SPI2_SS0	I/O	SPI2 1st slave select pin
			LCD_SEG6	O	LCD segment output 6 at LQFP64
			LCD_SEG20	O	LCD segment output 20 at LQFP128
13			PD.8	I/O	General purpose digital I/O pin
			LCD_SEG19	O	LCD segment output 19 at LQFP128
14			PD.9	I/O	General purpose digital I/O pin
			LCD_SEG18	O	LCD segment output 18 at LQFP128
15			PD.10	I/O	General purpose digital I/O pin
			LCD_SEG17	O	LCD segment output 17 at LQFP128
16			PD.11	I/O	General purpose digital I/O pin
			LCD_SEG16	O	LCD segment output 16 at LQFP128
17			PD.12	I/O	General purpose digital I/O pin
			LCD_SEG15	O	LCD segment output 15 at LQFP128
18			PD.13	I/O	General purpose digital I/O pin
			LCD_SEG14	O	LCD segment output 14 at LQFP128
19	10		PB.4	I/O	General purpose digital I/O pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			UART1_RXD	I	UART1 Data receiver input pin
			SC0_CD	I	SmartCard0 card detect pin
			SPI2_SS0	I/O	SPI2 1st slave select pin
			LCD_SEG5	O	LCD segment output 5 at LQFP64
			LCD_SEG13	O	LCD segment output 13 at LQFP128
20	11		PB.5	I/O	General purpose digital I/O pin
			UART1_TXD	O	UART1 Data transmitter output pin
			SC0_RST	O	SmartCard0 RST pin
			SPI2_CLK	I/O	SPI2 serial clock pin
			LCD_SEG4	O	LCD segment output 4 at LQFP64
			LCD_SEG12	O	LCD segment output 12 at LQFP128
21	12		PB.6	I/O	General purpose digital I/O pin
			UART1_RTSn	O	UART1 Request to Send output pin
			EBI_ALE	O	EBI address latch enable output pin
			SPI2_MISO0	I/O	SPI2 1st MISO (Master In, Slave Out) pin
			LCD_SEG3	O	LCD segment output 3 at LQFP64
			LCD_SEG11	O	LCD segment output 11 at LQFP128
22	13		PB.7	I/O	General purpose digital I/O pin
			UART1_CTSn	I	UART1 Clear to Send input pin
			EBI_nCS	O	EBI chip select enable output pin
			SPI2_MOSI0	I/O	SPI2 1st MOSI (Master Out, Slave In) pin
			LCD_SEG2	O	LCD segment output 2 at LQFP64
			LCD_SEG10	O	LCD segment output 10 at LQFP128
23					NC
24	14		LDO_CAP	P	LDO output pin
25					NC
26					NC
27	15		V_{DD}	P	Power supply for I/O ports and LDO source
28					NC
29	16		V_{SS}	P	Ground

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
30			V_{SS}	**P**	Ground
31			V_{SS}	**P**	Ground
32			V_{SS}	**P**	Ground
33			PE.12	**I/O**	General purpose digital I/O pin
34			PE.11	**I/O**	General purpose digital I/O pin
35			PE.10	**I/O**	General purpose digital I/O pin
36			PE.9	**I/O**	General purpose digital I/O pin
37			PE.8	**I/O**	General purpose digital I/O pin
			LCD_SEG9	**O**	LCD segment output 9 at LQFP128
38			PE.7	**I/O**	General purpose digital I/O pin
			LCD_SEG8	**O**	LCD segment output 8 at LQFP128
39					NC
40	17		USB_VBUS	**USB**	POWER SUPPLY: From USB Host or HUB.
41	18		USB_VDD33_CAP	**USB**	Internal Power Regulator Output 3.3V Decoupling Pin
42	19		USB_D-	**USB**	USB Differential Signal D-
43	20		USB_D+	**USB**	USB Differential Signal D+
44	21		PB.0	**I/O**	General purpose digital I/O pin
			UART0_RXD	**I**	UART0 Data receiver input pin
			SPI1_MOSI0	**I/O**	SPI1 1st MOSI (Master Out, Slave In) pin
			LCD_SEG1	**O**	LCD segment output 1 at LQFP64 (or as LCD_COM5)
			LCD_SEG7	**O**	LCD segment output 7 at LQFP128
45	22		PB.1	**I/O**	General purpose digital I/O pin
			UART0_TXD	**O**	UART0 Data transmitter output pin
			SPI1_MISO0	**I/O**	SPI1 1st MISO (Master In, Slave Out) pin
			LCD_SEG0	**O**	LCD segment output 0 at LQFP64 (or as LCD_COM4)
			LCD_SEG6	**O**	LCD segment output 6 at LQFP128
46	23		PB.2	**I/O**	General purpose digital I/O pin
			UART0_RTSn	**O**	UART0 Request to Send output pin
			EBI_nWRL	**O**	EBI low byte write enable output pin
			SPI1_CLK	**I/O**	SPI1 serial clock pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			LCD_COM3	O	LCD common output 3 at LQFP64
			LCD_SEG5	O	LCD segment output 5 at LQFP128
47	24		PB.3	I/O	General purpose digital I/O pin
			UART0_CTSn	I	UART0 Clear to Send input pin
			EBI_nWRH	O	EBI high byte write enable output pin
			SPI1_SS0	I/O	SPI1 1st slave select pin
			LCD_COM2	O	LCD common output 2 at LQFP64
			LCD_SEG4	O	LCD segment output 4 at LQFP128
48			PD.6	I/O	General purpose digital I/O pin
			LCD_SEG3	O	LCD segment output 3 at LQFP128
49			PD.7	I/O	General purpose digital I/O pin
			LCD_SEG2	O	LCD segment output 2 at LQFP128
50			PD.14	I/O	General purpose digital I/O pin
			LCD_SEG1	O	LCD segment output 1 at LQFP128 (or as LCD_COM5)
51			PD.15	I/O	General purpose digital I/O pin
			LCD_SEG0	O	LCD segment output 0 at LQFP128 (or as LCD_COM4)
52			PC.5	I/O	General purpose digital I/O pin
			SPI0_MOSI1	I/O	SPI0 2nd MOSI (Master Out, Slave In) pin
			LCD_COM3	O	LCD common output 3 at LQFP128
53			PC.4	I/O	General purpose digital I/O pin
			SPI0_MISO1	I/O	SPI0 2nd MISO (Master In, Slave Out) pin
			LCD_COM2	O	LCD common output 2 at LQFP128
54	25		PC.3	I/O	General purpose digital I/O pin
			SPI0_MOSI0	I/O	SPI0 1st MOSI (Master Out, Slave In) pin
			I2S_DO	O	I^2S data output
			SC1_RST	O	SmartCard1 RST pin
			LCD_COM1	O	LCD common output 1 at LQFP64
			LCD_COM1	O	LCD common output 1 at LQFP128
55	26		PC.2	I/O	General purpose digital I/O pin
			SPI0_MISO0	I/O	SPI0 1st MISO (Master In, Slave Out) pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			I2S_DI	I	I^2S data input
			SC1_PWR	O	SmartCard1 PWR pin
			LCD_COM0	O	LCD common output 0 at LQFP64
			LCD_COM0	O	LCD common output 0 at LQFP128
56	27		PC.1	I/O	General purpose digital I/O pin
			SPI0_CLK	I/O	SPI0 serial clock pin
			I2S_BCLK	I/O	I^2S bit clock pin
			SC1_DAT	I/O	SmartCard1 DATA pin(SC1_UART_RXD)
			LCD_DH2	O	LCD externl capacitor pin of charge pump circuit at LQFP64
			LCD_DH2	O	LCD externl capacitor pin of charge pump circuit at LQFP128
57	28		PC.0 / MCLKO	I/O	General purpose digital I/O pin / Module clock output pin
			SPI0_SS0	I/O	SPI0 1st slave select pin
			I2S_LRCLK	I/O	I^2S left right channel clock
			SC1_CLK	O	SmartCard1 clock pin(SC1_UART_TXD)
			LCD_DH1	O	LCD externl capacitor pin of charge pump circuit at LQFP64
			LCD_DH1	O	LCD externl capacitor pin of charge pump circuit at LQFP128
58			PE.6	I/O	General purpose digital I/O pin
59	29		LCD_VLCD	AO	LCD power supply pin
60					NC
61			PE.5	I/O	General purpose digital I/O pin
			PWM1_CH1	I/O	PWM1 Channel1 output
62	30		PB.11	I/O	General purpose digital I/O pin
			PWM1_CH0	I/O	PWM1 Channel0 output
			TM3	O	Timer3 external counter input
			SC2_DAT	I/O	SmartCard2 DATA pin(SC2_UART_RXD)
			SPI0_MISO0	I/O	SPI0 1st MISO (Master In, Slave Out) pin
			LCD_V1	O	LCD Unit voltage for LCD charge pump circuit at LQFP64

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			LCD_V1	O	LCD Unit voltage for LCD charge pump circuit at LQFP128
63	31		PB.10	I/O	General purpose digital I/O pin
			SPI0_SS1	I/O	SPI0 2nd slave select pin
			TM2	O	Timer2 external counter input
			SC2_CLK	O	SmartCard2 clock pin(SC2_UART_TXD)
			SPI0_MOSI0	I/O	SPI0 1st MOSI (Master Out, Slave In) pin
			LCD_V2	O	LCD driver biasing voltage at LQFP64
			LCD_V2	O	LCD driver biasing voltage at LQFP128
64	32		PB.9	I/O	General purpose digital I/O pin
			SPI1_SS1	I/O	SPI1 2nd slave select pin
			TM1	O	Timer1 external counter input
			SC2_RST	O	SmartCard2 RST pin
			INT0	I	External interrupt0 input pin
			LCD_V3	O	LCD driver biasing voltage at LQFP64
			LCD_V3	O	LCD driver biasing voltage at LQFP128
65			PE.4	I/O	General purpose digital I/O pin
			SPI0_MOSI0	I/O	SPI0 1st MOSI (Master Out, Slave In) pin
66			PE.3	I/O	General purpose digital I/O pin
			SPI0_MISO0	I/O	SPI0 1st MISO (Master In, Slave Out) pin
67			PE.2	I/O	General purpose digital I/O pin
			SPI0_CLK	I/O	SPI0 serial clock pin
68			PE.1	I/O	General purpose digital I/O pin
			PWM1_CH3	I/O	PWM1 Channel3 output
			SPI0_SS0	I/O	SPI0 1st slave select pin
69			PE.0	I/O	General purpose digital I/O pin
			PWM1_CH2	I/O	PWM1 Channel2 output
			I2S_MCLK	O	I^2S master clock output pin
70			PC.13	I/O	General purpose digital I/O pin
			SPI1_MOSI1	I/O	SPI1 2nd MOSI (Master Out, Slave In) pin
			PWM1_CH1	I/O	PWM1 Channel1 output

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			SNOOPER	I	Snooper pin
			INT1	I	External interrupt 1 input pin
			I2C0_SCL	O	I^2C0 clock pin
71			PC.12	I/O	General purpose digital I/O pin
			SPI1_MISO1	I/O	SPI1 2^{nd} MISO (Master In, Slave Out) pin
			PWM1_CH0	I/O	PWM1 Channel0 output
			INT0	I	External interrupt0 input pin
			I2C0_SDA	I/O	I^2C0 data I/O pin
72	33		PC.11	I/O	General purpose digital I/O pin
			SPI1_MOSI0	I/O	SPI1 1^{st} MOSI (Master Out, Slave In) pin
			UART1_TXD	O	UART1 Data transmitter output pin
			LCD_SEG31	O	LCD segment output 31 at LQFP64
73	34		PC.10	I/O	General purpose digital I/O pin
			SPI1_MISO0	I/O	SPI1 1^{st} MISO (Master In, Slave Out) pin
			UART1_RXD	I	UART1 Data receiver input pin
			LCD_SEG30	O	LCD segment output 30 at LQFP64
74	35		PC.9	I/O	General purpose digital I/O pin
			SPI1_CLK	I/O	SPI1 serial clock pin
			I2C1_SCL	I/O	I^2C1 clock pin
			LCD_SEG29	O	LCD segment output 29 at LQFP64
75	36		PC.8	I/O	General purpose digital I/O pin
			SPI1_SS0	I/O	SPI1 1^{st} slave select pin
			EBI_MCLK	O	EBI external clock output pin
			I2C1_SDA	I/O	I^2C1 data I/O pin
			LCD_SEG28	O	LCD segment output 28 at LQFP64
76	37		PA.15	I/O	General purpose digital I/O pin
			PWM0_CH3	I/O	PWM0 Channel3 output
			I2S_MCLK	O	I^2S master clock output pin
			TC3	I	Timer3 capture input
			SC0_PWR	O	SmartCard0 Power pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			UART0_TXD	O	UART0 Data transmitter output pin
			LCD_SEG27	O	LCD segment output 27 at LQFP64
77	38		PA.14	I/O	General purpose digital I/O pin
			PWM0_CH2	I/O	PWM0 Channel2 output
			EBI_AD15	I/O	EBI Address/Data bus bit15
			TC2	I	Timer2 capture input
			UART0_RXD	I	UART0 Data receiver input pin
			LCD_SEG26	O	LCD segment output 26 at LQFP64
78	39		PA.13	I/O	General purpose digital I/O pin
			PWM0_CH1	I/O	PWM0 Channel1 output
			EBI_AD14	I/O	EBI Address/Data bus bit14
			TC1	I	Timer1 capture input
			I2C0_SCL	I/O	I^2C0 clock pin
			LCD_SEG25	O	LCD segment output 25 at LQFP64
79	40		PA.12	I/O	General purpose digital I/O pin
			PWM0_CH0	I/O	PWM0 Channel0 output
			EBI_AD13	I/O	EBI Address/Data bus bit13
			TC0	I	Timer0 capture input
			I2C0_SDA	I/O	I^2C0 data I/O pin
			LCD_SEG24	O	LCD segment output 24 at LQFP64
80	41		ICE_DAT	I/O	Serial Wired Debugger Data pin **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_DAT pin.
			PF.0	I/O	General purpose digital I/O pin
			INT0	I	External interrupt0 input pin
81	42		ICE_CLK	I	Serial Wired Debugger Clock pin **Note:** It is recommended to use 100 kΩ pull-up resistor on ICE_CLK pin.
			PF.1	I/O	General purpose digital I/O pin
			FCLKO	O	Frequency Divider output pin
			INT1	I	External interrupt1 input pin
82					NC

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
83			V_{DD}	**P**	Power supply for I/O ports and LDO source for internal PLL and digital circuit
84					NC
85			V_{SS}	**P**	Ground
86			V_{SS}	**P**	Ground
87	43		AV_{SS}	**AP**	Ground Pin for analog circuit
88			AV_{SS}	**AP**	Ground Pin for analog circuit
89	44		PA.0	**I/O**	General purpose digital I/O pin
			AD0	**AI**	ADC analog input0
			SC2_CD	**I**	SmartCard2 card detect
90	45		PA.1	**I/O**	General purpose digital I/O pin
			AD1	**AI**	ADC analog input1
			EBI_AD12	**I/O**	EBI Address/Data bus bit12
91	46		PA.2	**I/O**	General purpose digital I/O pin
			AD2	**AI**	ADC analog input2
			EBI_AD11	**I/O**	EBI Address/Data bus bit11
			UART1_RXD	**I**	UART1 Data receiver input pin
			LCD_SEG23*	**AO**	LCD segment output 23 at LQFP64
92	47		PA.3	**I/O**	General purpose digital I/O pin
			AD3	**AI**	ADC analog input3
			EBI_AD10	**I/O**	EBI Address/Data bus bit10
			UART1_TXD	**O**	UART1 Data transmitter output pin
			LCD_SEG22*	**AO**	LCD segment output 22 at LQFP64
93	48		PA.4	**I/O**	General purpose digital I/O pin
			AD4	**AI**	ADC analog input4
			EBI_AD9	**I/O**	EBI Address/Data bus bit9
			SC2_PWR	**O**	SmartCard2 Power pin
			I2C0_SDA	**I/O**	I^2C0 data I/O pin
			LCD_SEG21*	**AO**	LCD segment output 21 at LQFP64
			LCD_SEG39*	**AO**	LCD segment output 39 at LQFP128
94	49		PA.5	**I/O**	General purpose digital I/O pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			AD5	**AI**	ADC analog input5
			EBI_AD8	**I/O**	EBI Address/Data bus bit8
			SC2_RST	**O**	SmartCard2 RST pin
			I2C0_SCL	**I/O**	I^2C0 clock pin
			LCD_SEG20*	**AO**	LCD segment output 20 at LQFP64
			LCD_SEG38*	**AO**	LCD segment output 38 at LQFP128
95	50		PA.6	**I/O**	General purpose digital I/O pin
			AD6	**AI**	ADC analog input6
			EBI_AD7	**I/O**	EBI Address/Data bus bit7
			TC3	**I**	Timer3 capture input
			SC2_CLK	**O**	SmartCard2 clock pin(SC2_UART_TXD)
			PWM0_CH3	**I/O**	PWM0 Channel3 output
			LCD_SEG19*	**AO**	LCD segment output 19 at LQFP64
			LCD_SEG37*	**AO**	LCD segment output 37 at LQFP128
96			PA.7	**I/O**	General purpose digital I/O pin
			AD7	**AI**	ADC analog input7
			EBI_AD6	**I/O**	EBI Address/Data bus bit6
			TC2	**I**	Timer2 capture input
			SC2_DAT	**I/O**	SmartCard2 DATA pin(SC2_UART_RXD)
			PWM0_CH2	**I/O**	PWM0 Channel2 output
			LCD_SEG36*	**AO**	LCD segment output 36 output at LQFP128
97	51		V_{REF}	**AP**	Voltage reference input for ADC
98					NC
99	52		AV_{DD}	**AP**	Power supply for internal analog circuit
100			PD.0	**I/O**	General purpose digital I/O pin
			UART1_RXD	**I**	UART1 Data receiver input pin
			SPI2_SS0	**I/O**	SPI2 1st slave select pin
			SC1_CLK	**O**	SmartCard1 clock pin(SC1_UART_TXD)
			AD8	**AI**	ADC analog input8
101			PD.1	**I/O**	General purpose digital I/O pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			TX1	O	UART1 Data transmitter output pin
			SPI2_CLK	I/O	SPI2 serial clock pin
			SC1_DAT	I/O	SmartCard1 DATA pin(SC1_UART_RXD)
			AD9	AI	ADC analog input9
102			PD.2	I/O	General purpose digital I/O pin
			UART1_RTSn	O	UART1 Request to Send output pin
			I2S_LRCLK	I/O	I^2S left right channel clock
			SPI2_MISO0	I/O	SPI2 1st MISO (Master In, Slave Out) pin
			SC1_PWR	O	SmartCard1 Power pin
			AD10	AI	ADC analog input10
103			PD.3	I/O	General purpose digital I/O pin
			UART1_CTSn	I	UART1 Clear to Send input pin
			I2S_BCLK	I/O	I^2S bit clock pin
			SPI2_MOSI0	I/O	SPI2 1st MOSI (Master Out, Slave In) pin
			SC1_RST	O	SmartCard1 RST pin
			AD11	AI	ADC analog input11
104					NC
105			PD.4	I/O	General purpose digital I/O pin
			I2S_DI	I	I^2S data input
			SPI2_MISO1	I/O	SPI2 2nd MISO (Master In, Slave Out) pin
			SC1_CD	I	SmartCard1 card detect
			LCD_SEG35	AO	LCD segment output 35 at LQFP128
106			PD.5	I/O	General purpose digital I/O pin
			I2S_DO	O	I^2S data output
			SPI2_MOSI1	I/O	SPI2 2nd MOSI (Master Out, Slave In) pin
			LCD_SEG34	AO	LCD segment output 34 at LQFP128
107	53		PC.7	I/O	General purpose digital I/O pin
			DA1_OUT	AO	DAC 1 output
			EBI_AD5	I/O	EBI Address/Data bus bit5
			TC1	I	Timer1 capture input

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
			PWM0_CH1	O	PWM0 Channel1 output
			LCD_SEG17*	AO	LCD segment output 17 at LQFP64
108	54		PC.6	I/O	General purpose digital I/O pin
			DA0_OUT	I	DAC0 output
			EBI_AD4	I/O	EBI Address/Data bus bit4
			TC0	I	Timer0 capture input
			SC1_CD		SmartCard1 card detect pin
			PWM0_CH0	I/O	PWM0 Channel0 output
109	55		PC.15	I/O	General purpose digital I/O pin
			EBI_AD3	I/O	EBI Address/Data bus bit3
			TC0	I	Timer0 capture input
			PWM1_CH2	I/O	PWM1 Channel2 output
			LCD_SEG16	AO	LCD segment output 16 at LQFP64
			LCD_SEG33	AO	LCD segment output 33 at LQFP128
110	56		PC.14	I/O	General purpose digital I/O pin
			EBI_AD2	I/O	EBI Address/Data bus bit2
			PWM1_CH3	I/O	PWM1 Channel3 output
			LCD_SEG15	AO	LCD segment output 15 at LQFP64
			LCD_SEG32	AO	LCD segment output 32 at LQFP128
111	57		PB.15	I/O	General purpose digital I/O pin
			INT1	I	External interrupt1 input pin
			SNOOPER	I	Snooper pin
			SC1_CD	I	SmartCard1 card detect
			LCD_SEG14	AO	LCD segment output 14 at LQFP64
			LCD_SEG31	AO	LCD segment output 31 at LQFP128
112					NC
113	58		XT1_IN	O	External 4~24 MHz crystal output pin
			PF.3	I/O	General purpose digital I/O pin
114	59		XT1_OUT	I	External 4~24 MHz crystal input pin
			PF.2	I/O	General purpose digital I/O pin

Pin No.			Pin Name	Pin Type	Description
LQFP 128-pin	LQFP 64-pin	LQFP 48-pin			
115					NC
116	60		nRESET	**I**	External reset input: Low active, set this pin low reset chip to initial state. With internal pull-up. **Note:** It is recommended to use 10 kΩ pull-up resistor and 10 μF capacitor on nRESET pin.
117	61		V_{SS}	**P**	Ground
118			V_{SS}	**P**	Ground
119					NC
120	62		V_{DD}	**P**	Power supply for I/O ports and LDO source for internal PLL and digital circuit
121					NC
122			PF.4	**I/O**	General purpose digital I/O pin
			I2C0_SDA	**I/O**	I^2C0 data I/O pin
123			PF.5	**I/O**	Digital GPI/O pin
			I2C0_SCL	**I/O**	I^2C0 clock pin
124			VSS	**P**	Ground
125	63		PVSS	**I/O**	PLL Ground
126	64		PB.8	**I/O**	General purpose digital I/O pin
			STADC	**I**	ADC external trigger input.
			TM0	**I**	Timer0 external counter input
			INT0	**I**	External interrupt0 input pin
			SC2_PWR	**O**	SmartCard2 Power pin
			LCD_SEG13	**AO**	LCD segment output 13 at LQFP64
			LCD_SEG30	**AO**	LCD segment output 30 at LQFP128
127			PE.15	**I/O**	General purpose digital I/O pin
			LCD_SEG29	**O**	LCD segment output 29 at LQFP128
128			PE.14	**I/O**	General purpose digital I/O pin
			LCD_SEG28	**O**	LCD segment output 28 at LQFP128

Note:

1. Pin Type: I=Digital Input, O=Digital Output; AI=Analog Input; AO=Analog Output; P=Power Pin; AP=Analog Power
2. * : Output voltage for ADC/LCD shared pins cannot be higher than V_{DD} because these pins are without 5V tolerance.

4 BLOCK DIAGRAM

4.1 Nano100 Block Diagram

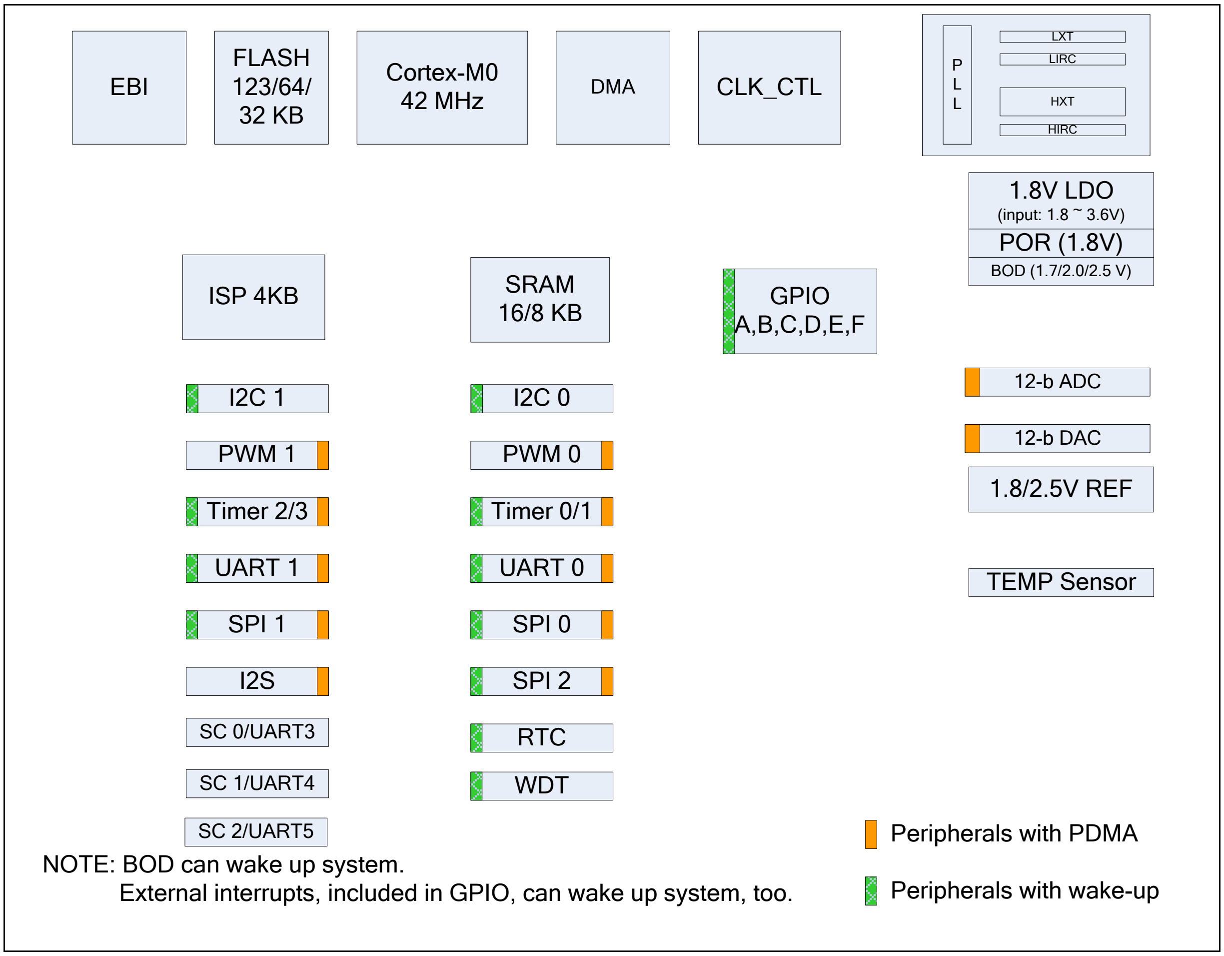


Figure 4.1-1 NuMicro® Nano100 Block Diagram

4.2 Nano110 Block Diagram

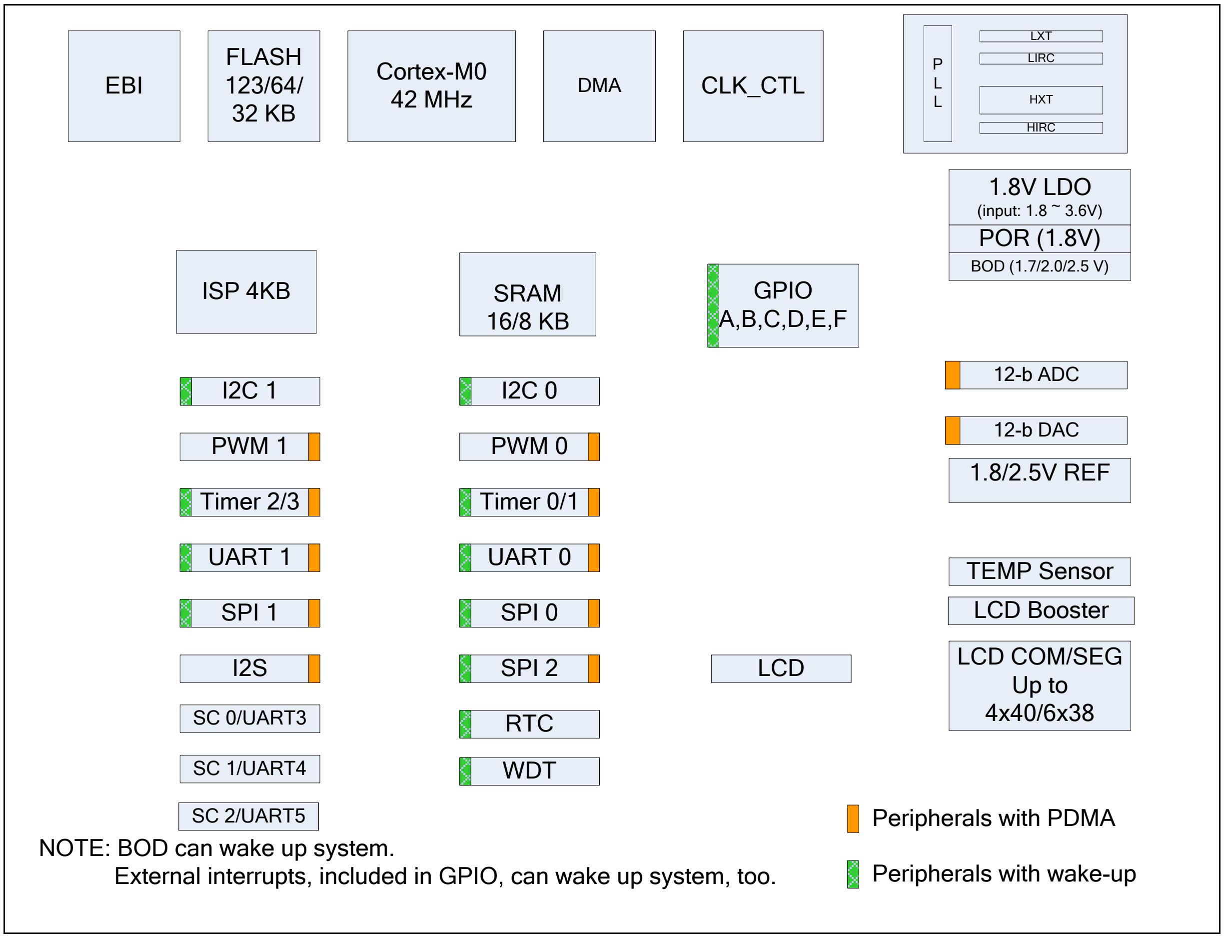


Figure 4.2-1 NuMicro® Nano110 Block Diagram

4.3 Nano120 Block Diagram

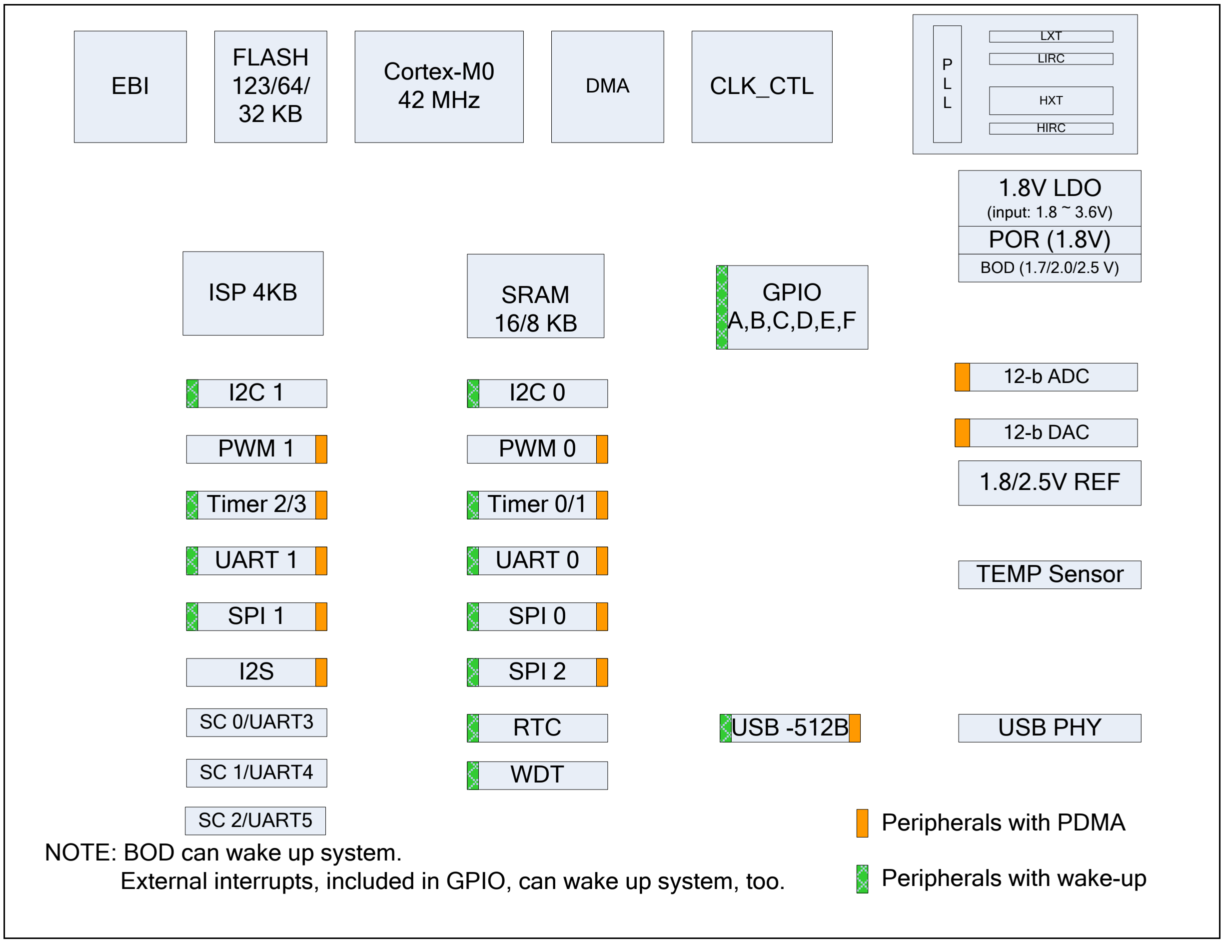


Figure 4.3-1 NuMicro® Nano120 Block Diagram

4.4 Nano130 Block Diagram

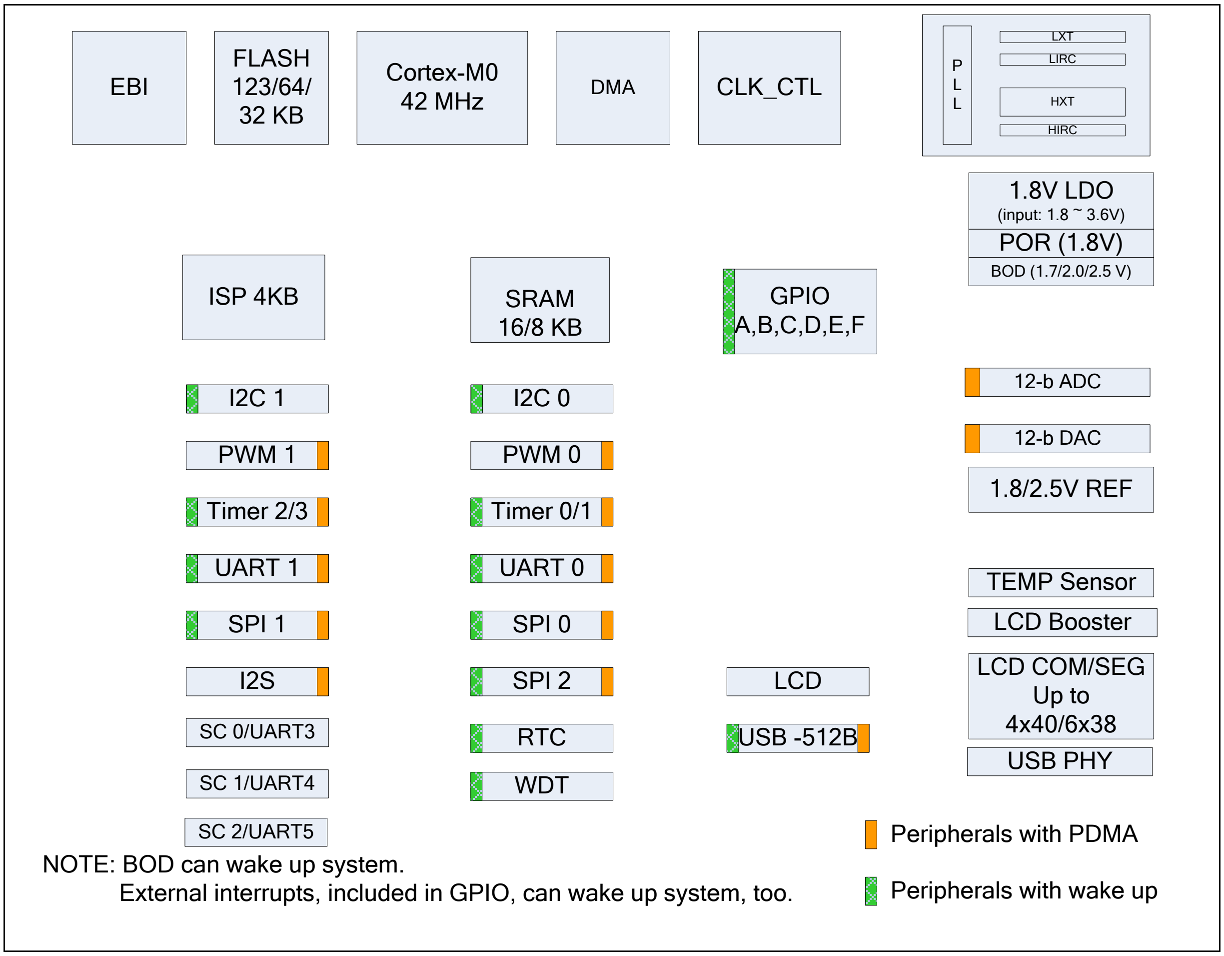


Figure 4.4-1 NuMicro® Nano130 Block Diagram

5 FUNCTIONAL DESCRIPTION

5.1 ARM® CORTEX®-M0 CORE

The Cortex®-M0 processor is a configurable, multistage, 32-bit RISC processor. It has an AMBA AHB-Lite interface and includes an NVIC component. It also has optional hardware debug functionality. The processor can execute Thumb code and is compatible with other Cortex-M profile processor. The profile supports two modes – Thread mode and Handler mode. Handler mode is entered as a result of an exception. An exception return can only be issued in Handler mode. Thread mode is entered on Reset, and can be entered as a result of an exception return. The following figure shows the functional controller of processor.

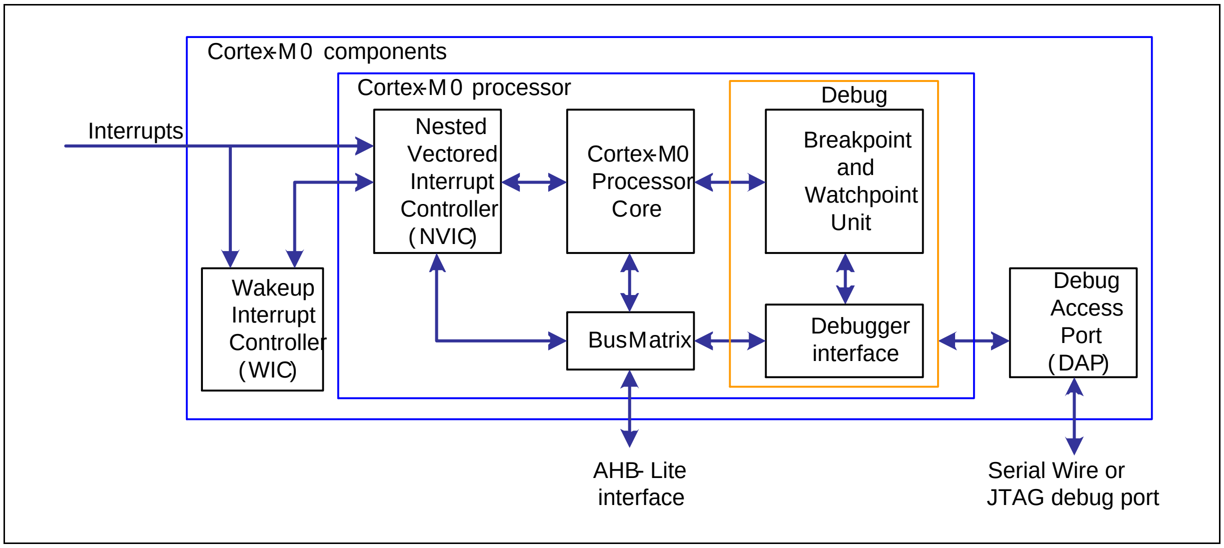


Figure 5.1-1 M0 Functional Block

The implemented device provides:

- A low gate count processor:
  - ARMv6-M Thumb® instruction set
  - Thumb-2 technology
  - ARMv6-M compliant 24-bit SysTick timer
  - A 32-bit hardware multiplier
  - Supports little-endian data accesses
  - Capable of deterministic, fixed-latency, interrupt handling
  - Load/store-multiples and multi-cycle-multiplies that can be abandoned and restarted to facilitate rapid interrupt handling
  - C Application Binary Interface compliant exception model. This is the ARMv6-M, C Application Binary Interface (C-ABI) compliant exception model that enables the use of pure C functions as interrupt handlers
  - Low Power Sleep mode entry using Wait For Interrupt (WFI), Wait For Event (WFE) instructions, or return from interrupt sleep-on-exit feature
- NVIC:
  - 32 external interrupt inputs, each with four levels of priority

- ◆ Dedicated Non-Maskable Interrupt (NMI) input
- ◆ Supports for both level-sensitive and pulse-sensitive interrupt lines
- ◆ Wake-up Interrupt Controller (WIC), providing Ultra-low Power Sleep mode support

- Debug support:
  - ◆ Four hardware breakpoints
  - ◆ Two watch points
  - ◆ Program Counter Sampling Register (PCSR) for non-intrusive code profiling
  - ◆ Single step and vector catch capabilities
- Bus interfaces:
  - ◆ Single 32-bit AMBA-3 AHB-Lite system interface providing simple integration to all system peripherals and memory
  - ◆ Single 32-bit slave port that supports the DAP (Debug Access Port)

5.1.1 System Timer (SysTick)

The Cortex-M0 includes an integrated system timer, SysTick. SysTick provides a simple, 24-bit cleared-on-write, decrementing, wrap-on-zero counter with a flexible control mechanism. The counter can be used in several different ways, for example:

An RTOS tick timer which fires at a programmable rate (for example 100Hz) and invokes a SysTick routine.

A high speed alarm timer using Core clock.

A variable rate alarm or signal timer – the duration range dependent on the reference clock used and the dynamic range of the counter.

A simple counter. Software can use this to measure task completion time.

An internal Clock Source control based on missing/meeting durations. The COUNTFLAG bit-field in the control and status register can be used to determine if an action completed within a set duration, as part of a dynamic clock management control loop.

When enabled, the timer will count down from the value in the SysTick Current Value Register (SYST_CVR) to zero, and reload (wrap) to the value in the SysTick Reload Value Register (SYST_RVR) on the next clock edge, and then decrement on subsequent clocks. When the counter transitions to zero, the COUNTFLAG status bit is set. The COUNTFLAG bit clears on read.

The SYST_CVR value is UNKNOWN on reset. Software should write to the register to clear it to zero before enabling the feature. This ensures the timer will count from the SYST_RVR value rather than an arbitrary value when it is enabled.

If the SYST_RVR is zero, the timer will be maintained with a current value of zero after it is reloaded with this value. This mechanism can be used to disable the feature independently from the timer enable bit.

For more detailed information, please refer to the “ARM® Cortex®-M0 Technical Reference Manual” and “ARM® v6-M Architecture Reference Manual”.

5.1.2 System Timer Control Register Map

R: read only, W: write only, R/W: both read and write, W&C: Write 1 clear

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **SCS_BA = 0xE000_E000**				
SYST_CTL	SCS_BA+0x10	R/W	SysTick Control and Status	0x0000_0004
SYST_RVR	SCS_BA+0x14	R/W	SysTick Reload Value Register	0xXXXX_XXXX
SYST_CVR	SCS_BA+0x18	R/W	SysTick Current Value Register	0xXXXX_XXXX

5.1.3 System Timer Control Register Description

SysTick Control and Status (SYST_CTL)

Register	Offset	R/W	Description	Reset Value
SYST_CTL	SCS_BA+0x10	R/W	SysTick Control and Status	0x0000_0004

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							COUNTFLAG
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved					CLKSRC	TICKINT	ENABLE

Bits	Description	
[31:17]	Reserved	**Reserved**
[16]	**COUNTFLAG**	**Returns 1 if timer counted to 0 since last time this register was read.** COUNTFLAG is set by a count transition from 1 to 0. COUNTFLAG is cleared on read or by a write to the Current Value register.
[15:3]	Reserved	**Reserved**
[2]	**CLKSRC**	1 = Core clock used for SysTick If no external clock provided, this bit will read as 1 and ignore writes. 0 = Clock Source is (optional) external reference clock
[1]	**TICKINT**	1 = Counting down to 0 will cause the SysTick exception to be pended. Clearing the SysTick Current Value register by a register write in software will not cause SysTick to be pended. 0 = Counting down to 0 does not cause the SysTick exception to be pended. Software can use COUNTFLAG to determine if a count to zero has occurred.
[0]	**ENABLE**	1 = The counter will operate in a multi-shot manner. 0 = The counter is Disabled

SysTick Reload Value Register (SYST_RVR)

Register	Offset	R/W	Description	Reset Value
SYST_RVR	SCS_BA+0x14	R/W	SysTick Reload Value Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RELOAD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RELOAD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RELOAD							

Bits	Description	
[31:24]	**Reserved**	**Reserved.**
[23:0]	**RELOAD**	The value to load into the Current Value register when the counter reaches 0.

SysTick Current Value Register (SYST_CVR)

Register	Offset	R/W	Description	Reset Value
SYST_CVR	SCS_BA+0x18	R/W	SysTick Current Value Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CURRENT							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CURRENT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CURRENT							

Bits	Description	
[31:24]	**Reserved**	**Reserved**
[23:0]	**CURRENT**	Current counter value. This is the value of the counter at the time it is sampled. The counter does not provide read-modify-write protection. The register is write-clear. A software write of any value will clear the register to 0. Unsupported bits RAZ (Read As Zero, writes ignore) (See SysTick Reload Value register).

5.1.4 System Control Registers

Key control and status features of Coterx-M0 are managed centrally in a System Control Block within the System Control Registers.

For more detailed information, please refer to the “ARM® Cortex®-M0 Technical Reference Manual” and “ARM® v6-M Architecture Reference Manual”.

5.1.5 System Control Register Memory Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **SCS_BA = 0xE000_E000**				
CPUID	SCS_BA+0xD00	R	CPUID Base Register	0x410C_C200
ICSR	SCS_BA+0xD04	R/W	Interrupt Control State Register	0x0000_0000
SCR	SCS_BA+0xD10	R/W	System Control Register	0x0000_0000
SHPR2	SCS_BA+0xD1C	R/W	System Handler Priority Register 2	0x0000_0000
SHPR3	SCS_BA+0xD20	R/W	System Handler Priority Register 3	0x0000_0000

5.1.6 System Control Register Description

CPUID Base Register (CPUID)

Register	Offset	R/W	Description	Reset Value
CPUID	SCS_BA+0xD00	R	CPUID Base Register	0x410C_C200

31	30	29	28	27	26	25	24
IMPLEMENTER							
23	22	21	20	19	18	17	16
Reserved				PART			
15	14	13	12	11	10	9	8
PARTNO							
7	6	5	4	3	2	1	0
PARTNO				REVISION			

Bits	Description	
[31:24]	**IMPLEMENTER**	Implementer code assigned by ARM (ARM = 0x41).
[23:20]	**Reserved**	**Reserved**
[19:16]	**PART**	Reads as 0xC for ARMv6-M parts
[15:4]	**PARTNO**	Reads as 0xC20.
[3:0]	**REVISION**	Reads as 0x0

Interrupt Control State Register (ICSR)

Register	Offset	R/W	Description	Reset Value
ICSR	SCS_BA+0xD04	R/W	Interrupt Control State Register	0x0000_0000

31	30	29	28	27	26	25	24
NMIPENDSET	Reserved		PENDSVSET	PENDSVCLR	PENDSTSET	PENDSTCLR	Reserved
23	22	21	20	19	18	17	16
ISRPREEMPT	ISRPENDING	Reserved	VECTPENDING				
15	14	13	12	11	10	9	8
VECTPENDING				Reserved			VECTACTIVE
7	6	5	4	3	2	1	0
VECTACTIVE							

Bits	Description	
[31]	**NMIPENDSET**	Setting this bit will activate an NMI. Since NMI is the highest priority exception, it will activate as soon as it is registered. Reads back with current state (1 if Pending, 0 if not).
[28]	**PENDSVSET**	Set a pending PendSV interrupt. This is normally used to request a context switch. Reads back with current state (1 if Pending, 0 if not).
[27]	**PENDSVCLR**	Write 1 to clear a pending PendSV interrupt.
[26]	**PENDSTSET**	Set a pending SysTick. Reads back with current state (1 if Pending, 0 if not).
[25]	**PENDSTCLR**	Write 1 to clear a pending SysTick.
[23]	**ISRPREEMPT**	If set, a pending exception will be serviced on exit from the debug halt state.
[22]	**ISRPENDING**	Indicates if an external configurable (NVIC generated) interrupt is pending.
[20:12]	**VECTPENDING**	Indicates the exception number for the highest priority pending exception. The pending state includes the effect of memory-mapped enable and mask registers. It does not include the PRIMASK special-purpose register qualifier. A value of zero indicates no pending exceptions.
[8:0]	**VECTACTIVE**	0 = Thread mode If value of VECTACTIVE > 1: the exception number for the current executing exception.

System Control Register (SCR)

Register	Offset	R/W	Description	Reset Value
SCR	SCS_BA+0xD10	R/W	System Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**SEVONPEND**	**Reserved**	**SLEEPDEEP**	**SLEEPONEXIT**	**Reserved**

Bits	Description	
[4]	**SEVONPEND**	When enabled, interrupt transitions from Inactive to Pending are included in the list of wake-up events for the WFE instruction.
[2]	**SLEEPDEEP**	A qualifying hint that indicates waking from sleep might take longer.
[1]	**SLEEPONEXIT**	When set to 1, the core can enter a sleep state on an exception return to Thread mode. This is the mode and exception level entered at reset, the base level of execution.

System Handler Priority Register 2 (SHPR2)

Register	Offset	R/W	Description	Reset Value
SHPR2	SCS_BA+0xD1C	R/W	System Handler Priority Register 2	0x0000_0000

31	30	29	28	27	26	25	24
PRI_11							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**

Bits	Description	
[31:30]	**PRI_11**	**Priority of system handler 11 – SVCall** “0” denotes the highest priority and “3” denotes the lowest priority.

System Handler Priority Register 3 (SHPR3)

Register	Offset	R/W	Description	Reset Value
SHPR3	SCS_BA+0xD20	R/W	System Handler Priority Register 3	0x0000_0000

31	30	29	28	27	26	25	24
PRI_15		Reserved					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_14		Reserved					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:30]	**PRI_15**	**Priority of system handler 15 – SysTick** “0” denotes the highest priority and “3” denotes the lowest priority.
[23:22]	**PRI_14**	**Priority of system handler 14 – PendSV** “0” denotes the highest priority and “3” denotes the lowest priority.

5.2 Memory Organization

5.2.1 Overview

The Nano100 provides 4G-byte addressing space. The memory locations assigned to each on-chip modules are shown in following. The detailed register definition, memory space, and programming detailed will be described in the following sections for each on-chip module. The Nano100 series only supports little-endian data format.

5.2.2 Memory Map

The memory locations assigned to each on-chip controllers are shown in the following table.

Address Space	Token	Modules
Flash & SRAM Memory Space		
0x0000_0000 – 0x0001_FFFF	FLASH_BA	FLASH Memory Space (128KB)
0x2000_0000 – 0x2000_3FFF	SRAM_BA	SRAM Memory Space (16KB)
0x6000_0000 --- 0x6001_FFFF	EXTMEM_BA	External Memory Space(128KB)
AHB Modules Space (0x5000_0000 – 0x501F_FFFF)		
0x5000_0000 – 0x5000_01FF	GCR_BA	System Management Control Registers
0x5000_0200 – 0x5000_02FF	CLK_BA	Clock Control Registers
0x5000_0300 – 0x5000_03FF	INT_BA	Interrupt Multiplexer Control Registers
0x5000_4000 – 0x5000_7FFF	GPIO_BA	GPIO Control Registers
0x5000_8000 – 0x5000_BFFF	DMA_BA	DMA Control Registers
0x5000_C000 – 0x5000_FFFF	FMC_BA	Flash Memory Control Registers
0x5001_0000 – 0x5001_03FF	EBI_BA	External Bus Interface Control Registers
APB1 Modules Space (0x4000_0000 ~ 0x400F_FFFF)		
0x4000_4000 – 0x4000_7FFF	WDT_BA	Watchdog Timer Control Registers
0x4000_8000 – 0x4000_BFFF	RTC_BA	Real Time Clock (RTC) Control Register
0x4001_0000 – 0x4001_3FFF	TMR01_BA	Timer0 and Timer1 Control Registers
0x4002_0000 – 0x4002_3FFF	I2C0_BA	I^2C0 Interface Control Registers
0x4003_0000 – 0x4003_3FFF	SPI0_BA	SPI0 with Master/Slave function Control Registers
0x4004_0000 – 0x4004_3FFF	PWM0_BA	PWM0 Control Registers
0x4005_0000 – 0x4005_3FFF	UART0_BA	UART0 Control Registers
0x4006_0000 – 0x4006_3FFF	USBD_BA	USB FS device Controller Registers
0x400A_0000 – 0x400A_3FFF	DAC_BA	Digital-Analog-Converter (DAC) Control Registers
0x400B_0000 – 0x400B_3FFF	LCD_BA	LCD Control Registers
0x400D_0000 – 0x400D_3FFF	SPI2_BA	SPI2 with Master/Slave function Control Registers

0x400E_0000 – 0x400E_3FFF	ADC12_BA	12-bit Analog-Digital-Converter (ADC12) Control Registers
APB2 Modules Space (0x4010_0000 ~ 0x401F_FFFF)		
0x4011_0000 – 0x4011_3FFF	TMR23_BA	Timer2 and Timer3 Control Registers
0x4012_0000 – 0x4012_3FFF	I2C1_BA	I^2C1 Interface Control Registers
0x4013_0000 – 0x4013_3FFF	SPI1_BA	SPI1 with Master/Slave function Control Registers
0x4014_0000 – 0x4014_3FFF	PWM1_BA	PWM1 Control Registers
0x4015_0000 – 0x4015_3FFF	UART1_BA	UART1 Control Registers
0x4019_0000 – 0x4019_3FFF	SC0_BA	SmartCard0 Control Registers
0x401A_0000 – 0x401A_3FFF	I2S_BA	I^2S Control Registers
0x401B_0000 – 0x401B_3FFF	SC1_BA	SmartCard1 Control Registers
0x401C_0000 – 0x401C_3FFF	SC2_BA	SmartCard2 Control Registers
System Control Space (0xE000_E000 ~ 0xE000_EFFF)		
0xE000_E010 – 0xE000_E0FF	SCS_BA	System Timer Control Registers
0xE000_E100 – 0xE000_ECFF	SCS_BA	External Interrupt Controller Control Registers
0xE000_ED00 – 0xE000_ED8F	SCS_BA	System Control Registers

5.3 Nested Vectored Interrupt Controller (NVIC)

5.3.1 Overview

The Cortex-M0 provides an interrupt controller as an integral part of the exception mode, named as "Nested Vectored Interrupt Controller (NVIC)". It is closely coupled to the processor kernel and provides following features.

5.3.2 Features

- Nested and Vectored interrupt support
- Automatic processor state saving and restoration
- Dynamic priority changing
- Reduced and deterministic interrupt latency

The NVIC prioritizes and handles all supported exceptions. All exceptions are handled in "Handler Mode". This NVIC architecture supports 32 (IRQ[31:0]) discrete interrupts with 4 levels of priority. All of the interrupts and most of the system exceptions can be configured to different priority levels. When an interrupt occurs, the NVIC will compare the priority of the new interrupt to the current running one's priority. If the priority of the new interrupt is higher than the current one, the new interrupt handler will override the current handler.

When any interrupts is accepted, the starting address of the interrupt service routine (ISR) is fetched from a vector table in memory. There is no need to determine which interrupt is accepted and branch to the starting address of the correlated ISR by software. While the starting address is fetched, NVIC will also automatically save processor state including the registers "PC, PSR, LR, R0~R3, R12" to the stack. At the end of the ISR, the NVIC will restore the mentioned registers from stack and resume the normal execution. Thus it will take less and deterministic time to process the interrupt request.

The NVIC supports "Tail Chaining" which handles back-to-back interrupts efficiently without the overhead of states saving and restoration and therefore reduces delay time in switching to pending ISR at the end of current ISR. The NVIC also supports "Late Arrival" which improves the efficiency of concurrent ISRs. When a higher priority interrupt request occurs before the current ISR starts to execute (at the stage of state saving and starting address fetching), the NVIC will give priority to the higher one without delay penalty. Thus it advances the real-time capability.

For more detailed information, please refer to the "ARM® Cortex®-M0 Technical Reference Manual" and "ARM® v6-M Architecture Reference Manual".

5.3.3 Exception Model and System Interrupt Map

The following table lists the exception model supported by Nano100 serials. Software can set four levels of priority on some of these exceptions as well as on all interrupts. The highest user-configurable priority is denoted as "0" and the lowest priority is denoted as "3". The default priority of all the user-configurable interrupts is "0". Note that priority "0" is treated as the fourth priority on the system, after three system exceptions "Reset", "NMI" and "Hard Fault".

Exception Name	Vector Number	Priority
Reset	1	-3
NMI	2	-2
Hard Fault	3	-1
Reserved	4 ~ 10	Reserved
SVCall	11	Configurable
Reserved	12 ~ 13	Reserved
PendSV	14	Configurable
SysTick	15	Configurable
Interrupt (IRQ0 ~ IRQ31)	16 ~ 47	Configurable

Table 5.3-1 Exception Model

Vector Number	Interrupt Number (Bit in Interrupt Registers)	Interrupt Name	Source IP	Interrupt Description
0 ~ 15	-	-	-	System exceptions
16	0	BOD_INT	Brown-out	Brown-out low voltage detected interrupt
17	1	WDT_INT	WDT	Watchdog Timer interrupt
18	2	EINT0	GPIO	External signal interrupt from PB.14 pin
19	3	EINT1	GPIO	External signal interrupt from PB.15 pin
20	4	GPABC_INT	GPIO	External signal interrupt from PA[15:0] / PB[13:0]/PC[15:0]
21	5	GPDEF_INT	GPIO	External interrupt from PD[15:0]/PE[15:0]/PF[5:0]
22	6	PWM0_INT	PWM0	PWM0 interrupt
23	7	PWM1_INT	PWM1	PWM1interrupt
24	8	TMR0_INT	TMR0	Timer0 interrupt
25	9	TMR1_INT	TMR1	Timer1 interrupt

Vector Number	Interrupt Number (Bit in Interrupt Registers)	Interrupt Name	Source IP	Interrupt Description
26	10	TMR2_INT	TMR2	Timer2 interrupt
27	11	TMR3_INT	TMR3	Timer3 interrupt
28	12	UART0_INT	UART0	UART0 interrupt
29	13	UART1_INT	UART1	UART1 interrupt
30	14	SPI0_INT	SPI0	SPI0 interrupt
31	15	SPI1_INT	SPI1	SPI1 interrupt
32	16	SPI2_INT	SPI2	SPI2 interrupt
33	17	IRC_INT	IRC	IRC TRIM interrupt
34	18	I2C0_INT	I2C0	I^2C0 interrupt
35	19	I2C1_INT	I2C1	I^2C1 interrupt
36	20	SC2_INT	SC2	Smart Card2 interrupt
37	21	SC0_INT	SC0	Smart Card0 interrupt
38	22	SC1_INT	SC1	Smart Card1 interrupt
39	23	USB_INT	USBD	USB FS Device interrupt
41	25	LCD_INT	LCD	LCD interrupt
42	26	DMA_INT	DMA	DMA interrupt
43	27	I2S_INT	I^2S	I^2S interrupt
44	28	PD_WU_INT	CLKC	Clock controller interrupt for chip wake-up from power-down state
45	29	ADC_INT	ADC	ADC interrupt
46	30	DAC_INT	DAC	DAC interrupt
47	31	RTC_INT	RTC	Real time clock interrupt

Table 5.3-2 System Interrupt Map

5.3.4 Vector Table

When any interrupts is accepted, the processor will automatically fetch the starting address of the interrupt service routine (ISR) from a vector table in memory. For ARMv6-M, the vector table base address is fixed at 0x00000000. The vector table contains the initialization value for the stack pointer on reset, and the entry point addresses for all exception handlers. The vector number on previous page defines the order of entries in the vector table associated with exception handler entry as illustrated in previous section.

Vector Table Word Offset	Description
0	SP_main – The Main stack pointer
Vector Number	Exception Entry Pointer using that Vector Number

Table 5.3-3 Vector Table Format

5.3.5 Operation Description

The NVIC interrupts can be enabled and disabled by writing to their corresponding Interrupt Set-Enable or Interrupt Clear-Enable register bit-field. The registers use a write-1-to-enable and write-1-to-clear policy, both registers reading back the current enabled state of the corresponding interrupts. When an interrupt is disabled, interrupt assertion will cause the interrupt to become Pending, however, the interrupt will not activate. If an interrupt is Active when it is disabled, it remains in its Active state until cleared by reset or an exception return. Clearing the enable bit prevents new activations of the associated interrupt.

The NVIC interrupts can be pended/un-pended using a complementary pair of registers to those used to enable/disable the interrupts, named the Set-Pending Register and Clear-Pending Register respectively. The registers use a write-1-to-enable and write-1-to-clear policy, both registers reading back the current pended state of the corresponding interrupts. The Clear-Pending Register has no effect on the execution status of an Active interrupt.

The NVIC interrupts are prioritized by updating an 8-bit field within a 32-bit register (each register supporting four interrupts).

The general registers associated with the NVIC are all accessible from a block of memory in the System Control Space and will be described in next section.

5.3.6 NVIC Control Register Map

R: read only**, W:** write only**, R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **SCS_BA = 0xE000_E000**				
NVIC_ISER	SCS_BA+0x100	R/W	IRQ0~IRQ31 Set-Enable Control Register	0x0000_0000
NVIC_ICER	SCS_BA+0x180	R/W	IRQ0~IRQ31 Clear-Enable Control Register	0x0000_0000
NVIC_ISPR	SCS_BA+0x200	R/W	IRQ0~IRQ31 Set-Pending Control Register	0x0000_0000
NVIC_ICPR	SCS_BA+0x280	R/W	IRQ0~IRQ31Clear-Pending Control Register	0x0000_0000
NVIC_IPR0	SCS_BA+0x400	R/W	IRQ0~IRQ3 Priority Control Register	0x0000_0000
NVIC_IPR1	SCS_BA+0x404	R/W	IRQ4~IRQ7 Priority Control Register	0x0000_0000
NVIC_IPR2	SCS_BA+0x408	R/W	IRQ8~IRQ11 Priority Control Register	0x0000_0000
NVIC_IPR3	SCS_BA+0x40C	R/W	IRQ12~IRQ15 Priority Control Register	0x0000_0000
NVIC_IPR4	SCS_BA+0x410	R/W	IRQ16~IRQ19 Priority Control Register	0x0000_0000
NVIC_IPR5	SCS_BA+0x414	R/W	IRQ20~IRQ23 Priority Control Register	0x0000_0000
NVIC_IPR6	SCS_BA+0x418	R/W	IRQ24~IRQ27 Priority Control Register	0x0000_0000
NVIC_IPR7	SCS_BA+0x41C	R/W	IRQ28~IRQ31 Priority Control Register	0x0000_0000

5.3.7 NVIC Control Register Description

IRQ0 ~ IRQ31 Set-enable Control Register (NVIC_ISER)

Register	Offset	R/W	Description	Reset Value
NVIC_ISER	SCS_BA+0x100	R/W	IRQ0~IRQ31 Set-Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SETENA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SETENA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SETENA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SETENA							

Bits	Description	
[31:0]	**SETENA**	Enable one or more interrupts within a group of 32. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Writing 1 will enable the associated interrupt. Writing 0 has no effect. The register reads back with the current enable state.

IRQ0 ~ IRQ31 Clear-enable Control Register (NVIC_ICER)

Register	Offset	R/W	Description	Reset Value
NVIC_ICER	SCS_BA+0x180	R/W	IRQ0~IRQ31 Clear-Enable Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CLRENA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CLRENA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CLRENA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CLRENA							

Bits	Description	
[31:0]	**CLRENA**	Disable one or more interrupts within a group of 32. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Writing 1 will disable the associated interrupt. Writing 0 has no effect. The register reads back with the current enable state.

IRQ0 ~ IRQ31 Set-pending Control Register (NVIC_ISPR)

Register	Offset	R/W	Description	Reset Value
NVIC_ISPR	SCS_BA+0x200	R/W	IRQ0~IRQ31 Set-Pending Control Register	0x0000_0000

31	30	29	28	27	26	25	24
SETPEND							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SETPEND							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SETPEND							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SETPEND							

Bits	Description	
[31:0]	**SETPEND**	Writing 1 to a bit to set pending state of the associated interrupt under software control. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Writing 0 has no effect. The register reads back with the current pending state.

IRQ0 ~ IRQ31 Clear-pending Control Register (NVIC_ICPR)

Register	Offset	R/W	Description	Reset Value
NVIC_ICPR	SCS_BA+0x280	R/W	IRQ0~IRQ31Clear-Pending Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CLRPEND							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CLRPEND							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CLRPEND							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CLRPEND							

Bits	Description	
[31:0]	**CLRPEND**	Writing 1 to a bit to remove the pending state of associated interrupt under software control. Each bit represents an interrupt number from IRQ0 ~ IRQ31 (Vector number from 16 ~ 47). Writing 0 has no effect. The register reads back with the current pending state.

IRQ0 ~ IRQ3 Interrupt Priority Register (NVIC_IPR0)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR0	SCS_BA+0x400	R/W	IRQ0~IRQ3 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_3		-					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_2		-					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRI_1		-					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_0		-					

Bits	Description	
[31:30]	**PRI_3**	**Priority of IRQ3** “0” denotes the highest priority and “3” denotes the lowest priority.
[23:22]	**PRI_2**	**Priority of IRQ2** “0” denotes the highest priority and “3” denotes the lowest priority.
[15:14]	**PRI_1**	**Priority of IRQ1** “0” denotes the highest priority and “3” denotes the lowest priority.
[7:6]	**PRI_0**	**Priority of IRQ0** “0” denotes the highest priority and “3” denotes the lowest priority.

IRQ4 ~ IRQ7 Interrupt Priority Register (NVIC_IPR1)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR1	SCS_BA+0x404	R/W	IRQ4~IRQ7 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_7		-					
23	22	21	20	19	18	17	16
PRI_6		-					
15	14	13	12	11	10	9	8
PRI_5		-					
7	6	5	4	3	2	1	0
PRI_4		-					

Bits	Description	
[31:30]	PRI_7	**Priority of IRQ7** "0" denotes the highest priority and "3" denotes the lowest priority.
[23:22]	PRI_6	**Priority of IRQ6** "0" denotes the highest priority and "3" denotes the lowest priority.
[15:14]	PRI_5	**Priority of IRQ5** "0" denotes the highest priority and "3" denotes the lowest priority.
[7:6]	PRI_4	**Priority of IRQ4** "0" denotes the highest priority and "3" denotes the lowest priority.

IRQ8 ~ IRQ11 Interrupt Priority Register (NVIC_IPR2)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR2	SCS_BA+0x408	R/W	IRQ8~IRQ11 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_11		-					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_10		-					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRI_9		-					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_8		-					

Bits	Description	
[31:30]	**PRI_11**	**Priority of IRQ11** “0” denotes the highest priority and “3” denotes the lowest priority.
[23:22]	**PRI_10**	**Priority of IRQ10** “0” denotes the highest priority and “3” denotes the lowest priority.
[15:14]	**PRI_9**	**Priority of IRQ9** “0” denotes the highest priority and “3” denotes the lowest priority.
[7:6]	**PRI_8**	**Priority of IRQ8** “0” denotes the highest priority and “3” denotes the lowest priority.

IRQ12 ~ IRQ15 Interrupt Priority Register (NVIC_IPR3)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR3	SCS_BA+0x40C	R/W	IRQ12~IRQ15 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_15		-					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_14		-					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRI_13		-					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_12		-					

Bits	Description	
[31:30]	**PRI_15**	**Priority** of IRQ15 "0" denotes the highest priority and "3" denotes the lowest priority.
[23:22]	**PRI_14**	**Priority of IRQ14** "0" denotes the highest priority and "3" denotes the lowest priority.
[15:14]	**PRI_13**	**Priority of IRQ13** "0" denotes the highest priority and "3" denotes the lowest priority.
[7:6]	**PRI_12**	**Priority of IRQ12** "0" denotes the highest priority and "3" denotes the lowest priority.

IRQ16 ~ IRQ19 Interrupt Priority Register (NVIC_IPR4)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR4	SCS_BA+0x410	R/W	IRQ16~IRQ19 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_19		-					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_18		-					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRI_17		-					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_16		-					

Bits	Description	
[31:30]	**PRI_19**	**Priority of IRQ19** “0” denotes the highest priority and “3” denotes the lowest priority.
[23:22]	**PRI_18**	**Priority of IRQ18** “0” denotes the highest priority and “3” denotes the lowest priority.
[15:14]	**PRI_17**	**Priority of IRQ17** “0” denotes the highest priority and “3” denotes the lowest priority.
[7:6]	**PRI_16**	**Priority of IRQ16** “0” denotes the highest priority and “3” denotes the lowest priority.

IRQ20 ~ IRQ23 Interrupt Priority Register (NVIC_IPR5)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR5	SCS_BA+0x414	R/W	IRQ20~IRQ23 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_23		-					
23	22	21	20	19	18	17	16
PRI_22		-					
15	14	13	12	11	10	9	8
PRI_21		-					
7	6	5	4	3	2	1	0
PRI_20		-					

Bits	Description	
[31:30]	PRI_23	**Priority of IRQ23** "0" denotes the highest priority and "3" denotes the lowest priority.
[23:22]	PRI_22	**Priority of IRQ22** "0" denotes the highest priority and "3" denotes the lowest priority.
[15:14]	PRI_21	**Priority of IRQ21** "0" denotes the highest priority and "3" denotes the lowest priority.
[7:6]	PRI_20	**Priority of IRQ20** "0" denotes the highest priority and "3" denotes the lowest priority.

IRQ24 ~ IRQ27 Interrupt Priority Register (NVIC_IPR6)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR6	SCS_BA+0x418	R/W	IRQ24~IRQ27 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_27		-					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_26		-					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRI_25		-					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_24		-					

Bits	Description	
[31:30]	PRI_27	**Priority of IRQ27** "0" denotes the highest priority and "3" denotes the lowest priority.
[23:22]	PRI_26	**Priority of IRQ26** "0" denotes the highest priority and "3" denotes the lowest priority.
[15:14]	PRI_25	**Priority of IRQ25** "0" denotes the highest priority and "3" denotes the lowest priority.
[7:6]	PRI_24	**Priority of IRQ24** "0" denotes the highest priority and "3" denotes the lowest priority.

IRQ28 ~ IRQ31 Interrupt Priority Register (NVIC_IPR7)

Register	Offset	R/W	Description	Reset Value
NVIC_IPR7	SCS_BA+0x41C	R/W	IRQ28~IRQ31 Priority Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PRI_31		-					
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PRI_30		-					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PRI_29		-					
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRI_28		-					

Bits	Description	
[31:30]	**PRI_31**	**Priority of IRQ31** "0" denotes the highest priority and "3" denotes the lowest priority.
[23:22]	**PRI_30**	**Priority of IRQ30** "0" denotes the highest priority and "3" denotes the lowest priority.
[15:14]	**PRI_29**	**Priority of IRQ29** "0" denotes the highest priority and "3" denotes the lowest priority.
[7:6]	**PRI_28**	**Priority of IRQ28** "0" denotes the highest priority and "3" denotes the lowest priority.

5.3.8 Interrupt Source Control Registers

Besides the interrupt control registers associated with the NVIC, the Nano100 serials also implement some specific control registers to facilitate the interrupt functions, including “interrupt source identify”, and ”NMI source selection”, which are described below.

5.3.8.1 Interrupt Source Control Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
SCS Base Address: **INT_BA = 0x5000_0300**				
IRQ0_SRC	INT_BA+0x00	R	MCU IRQ0 (BOD_INT) interrupt source identify	0xXXXX_XXXX
IRQ1_SRC	INT_BA+0x04	R	MCU IRQ1 (WDT_INT) interrupt source identify	0xXXXX_XXXX
IRQ2_SRC	INT_BA+0x08	R	MCU IRQ2 (EINT0) interrupt source identify	0xXXXX_XXXX
IRQ3_SRC	INT_BA+0x0C	R	MCU IRQ3 (EINT1) interrupt source identify	0xXXXX_XXXX
IRQ4_SRC	INT_BA+0x10	R	MCU IRQ4 (GPABC_INT) interrupt source identify	0xXXXX_XXXX
IRQ5_SRC	INT_BA+0x14	R	MCU IRQ5 (GPDEF_INT) interrupt source identify	0xXXXX_XXXX
IRQ6_SRC	INT_BA+0x18	R	MCU IRQ6 (PWM0_INT) interrupt source identify	0xXXXX_XXXX
IRQ7_SRC	INT_BA+0x1C	R	MCU IRQ7 (PWM1_INT) interrupt source identify	0xXXXX_XXXX
IRQ8_SRC	INT_BA+0x20	R	MCU IRQ8 (TMR0_INT) interrupt source identify	0xXXXX_XXXX
IRQ9_SRC	INT_BA+0x24	R	MCU IRQ9 (TMR1_INT) interrupt source identify	0xXXXX_XXXX
IRQ10_SRC	INT_BA+0x28	R	MCU IRQ10 (TMR2_INT) interrupt source identify	0xXXXX_XXXX
IRQ11_SRC	INT_BA+0x2C	R	MCU IRQ11 (TMR3_INT) interrupt source identify	0xXXXX_XXXX
IRQ12_SRC	INT_BA+0x30	R	MCU IRQ12 (UART0_INT) interrupt source identify	0xXXXX_XXXX
IRQ13_SRC	INT_BA+0x34	R	MCU IRQ13 (UART1_INT) interrupt source identify	0xXXXX_XXXX
IRQ14_SRC	INT_BA+0x38	R	MCU IRQ14 (SPI0_INT) interrupt source identify	0xXXXX_XXXX
IRQ15_SRC	INT_BA+0x3C	R	MCU IRQ15 (SPI1_INT) interrupt source identify	0xXXXX_XXXX
IRQ16_SRC	INT_BA+0x40	R	MCU IRQ16 (SPI2_INT) interrupt source identify	0xXXXX_XXXX
IRQ17_SRC	INT_BA+0x44	R	MCU IRQ17 (IRC_INT) interrupt source identify	0xXXXX_XXXX

IRQ18_SRC	INT_BA+0x48	R	MCU IRQ18 (I2C0_INT) interrupt source identify	0xXXXX_XXXX
IRQ19_SRC	INT_BA+0x4C	R	MCU IRQ19 (I2C1_INT) interrupt source identify	0xXXXX_XXXX
IRQ20_SRC	INT_BA+0x50	R	MCU IRQ20 (SC2_INT) interrupt source identify	0xXXXX_XXXX
IRQ21_SRC	INT_BA+0x54	R	MCU IRQ21 (SC0_INT) interrupt source identify	0xXXXX_XXXX
IRQ22_SRC	INT_BA+0x58	R	MCU IRQ22 (SC1_INT) interrupt source identify	0xXXXX_XXXX
IRQ23_SRC	INT_BA+0x5C	R	MCU IRQ23 (USB_INT) interrupt source identify	0xXXXX_XXXX
IRQ25_SRC	INT_BA+0x64	R	MCU IRQ25 (LCD_INT) interrupt source identify	0xXXXX_XXXX
IRQ26_SRC	INT_BA+0x68	R	MCU IRQ26 (DMA_INT) interrupt source identify	0xXXXX_XXXX
IRQ27_SRC	INT_BA+0x6C	R	MCU IRQ27 (I2S_INT) interrupt source identify	0xXXXX_XXXX
IRQ28_SRC	INT_BA+0x70	R	MCU IRQ28 (PDWU_INT) interrupt source identify	0xXXXX_XXXX
IRQ29_SRC	INT_BA+0x74	R	MCU IRQ29 (ADC_INT) interrupt source identify	0xXXXX_XXXX
IRQ30_SRC	INT_BA+0x78	R	MCU IRQ30 (DAC_INT) interrupt source identify	0xXXXX_XXXX
IRQ31_SRC	INT_BA+0x7C	R	MCU IRQ31 (RTC_INT) interrupt source identify	0xXXXX_XXXX
NMI_SEL	INT_BA+0x80	R/W	NMI source interrupt select control register	0x0000_0000
MCU_IRQ	INT_BA+0x84	R/W	MCU interrupt request source register	0x0000_0000

5.3.8.2 *Interrupt source control register description*

Interrupt Source Identify Register (IRQn_SRC)

Register	Offset	R/W	Description	Reset Value
IRQ0_SRC	INT_BA+0x00	R	MCU IRQ0 (BOD_INT) interrupt source identify	0xXXXX_XXXX
IRQ1_SRC	INT_BA+0x04	R	MCU IRQ1 (WDT_INT) interrupt source identify	0xXXXX_XXXX
IRQ2_SRC	INT_BA+0x08	R	MCU IRQ2 (EINT0) interrupt source identify	0xXXXX_XXXX
IRQ3_SRC	INT_BA+0x0C	R	MCU IRQ3 (EINT1) interrupt source identify	0xXXXX_XXXX
IRQ4_SRC	INT_BA+0x10	R	MCU IRQ4 (GPABC_INT) interrupt source identify	0xXXXX_XXXX
IRQ5_SRC	INT_BA+0x14	R	MCU IRQ5 (GPDEF_INT) interrupt source identify	0xXXXX_XXXX
IRQ6_SRC	INT_BA+0x18	R	MCU IRQ6 (PWM0_INT) interrupt source identify	0xXXXX_XXXX
IRQ7_SRC	INT_BA+0x1C	R	MCU IRQ7 (PWM1_INT) interrupt source identify	0xXXXX_XXXX
IRQ8_SRC	INT_BA+0x20	R	MCU IRQ8 (TMR0_INT) interrupt source identify	0xXXXX_XXXX
IRQ9_SRC	INT_BA+0x24	R	MCU IRQ9 (TMR1_INT) interrupt source identify	0xXXXX_XXXX
IRQ10_SRC	INT_BA+0x28	R	MCU IRQ10 (TMR2_INT) interrupt source identify	0xXXXX_XXXX
IRQ11_SRC	INT_BA+0x2C	R	MCU IRQ11 (TMR3_INT) interrupt source identify	0xXXXX_XXXX
IRQ12_SRC	INT_BA+0x30	R	MCU IRQ12 (UART0_INT) interrupt source identify	0xXXXX_XXXX
IRQ13_SRC	INT_BA+0x34	R	MCU IRQ13 (UART1_INT) interrupt source identify	0xXXXX_XXXX
IRQ14_SRC	INT_BA+0x38	R	MCU IRQ14 (SPI0_INT) interrupt source identify	0xXXXX_XXXX
IRQ15_SRC	INT_BA+0x3C	R	MCU IRQ15 (SPI1_INT) interrupt source identify	0xXXXX_XXXX
IRQ16_SRC	INT_BA+0x40	R	MCU IRQ16 (SPI2_INT) interrupt source identify	0xXXXX_XXXX
IRQ17_SRC	INT_BA+0x44	R	MCU IRQ17 (IRC_INT) interrupt source identify	0xXXXX_XXXX
IRQ18_SRC	INT_BA+0x48	R	MCU IRQ18 (I2C0_INT) interrupt source identify	0xXXXX_XXXX
IRQ19_SRC	INT_BA+0x4C	R	MCU IRQ19 (I2C1_INT) interrupt source identify	0xXXXX_XXXX
IRQ20_SRC	INT_BA+0x50	R	MCU IRQ20 (SC2_INT) interrupt source identify	0xXXXX_XXXX
IRQ21_SRC	INT_BA+0x54	R	MCU IRQ21 (SC0_INT) interrupt source identify	0xXXXX_XXXX
IRQ22_SRC	INT_BA+0x58	R	MCU IRQ22 (SC1_INT) interrupt source identify	0xXXXX_XXXX

IRQ23_SRC	INT_BA+0x5C	R	MCU IRQ23 (USB_INT) interrupt source identify	0xXXXX_XXXX
IRQ25_SRC	INT_BA+0x64	R	MCU IRQ25 (LCD_INT) interrupt source identify	0xXXXX_XXXX
IRQ26_SRC	INT_BA+0x68	R	MCU IRQ26 (DMA_INT) interrupt source identify	0xXXXX_XXXX
IRQ27_SRC	INT_BA+0x6C	R	MCU IRQ27 (I2S_INT) interrupt source identify	0xXXXX_XXXX
IRQ28_SRC	INT_BA+0x70	R	MCU IRQ28 (PDWU_INT) interrupt source identify	0xXXXX_XXXX
IRQ29_SRC	INT_BA+0x74	R	MCU IRQ29 (ADC_INT) interrupt source identify	0xXXXX_XXXX
IRQ30_SRC	INT_BA+0x78	R	MCU IRQ30 (DAC_INT) interrupt source identify	0xXXXX_XXXX
IRQ31_SRC	INT_BA+0x7C	R	MCU IRQ31 (RTC_INT) interrupt source identify	0xXXXX_XXXX

31	30	29	28	27	26	25	24
-							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
-							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
-							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-				**INT_SRC**			

Bits	Description	
[31:4]	-	**Reserved**
[3:0]	**INT_SRC**	**Interrupt Source** Define the interrupt sources for interrupt event.

Address	INT Number	Bits	Description
INT_BA+0x00	0	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = BOD_INT
INT_BA+0x04	1	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = WDT_INT
INT_BA+0x08	2	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0

Address	INT Number	Bits	Description
			Bit0 = EINT0 – external interrupt 0 from PB.14
INT_BA+0x0C	3	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = EINT1 – external interrupt 1 from PB.15
INT_BA+0x10	4	[2:0]	Bit2 = GPC_INT Bit1 = GPB_INT Bit0 = GPA_INT
INT_BA+0x14	5	[2:0]	Bit2 = GPF_INT Bit1 = GPE_INT Bit0 = GPD_INT
INT_BA+0x18	6	[3:0]	Bit3 = PWM0_CH3_INT Bit2 = PWM0_CH2_INT Bit1 = PWM0_CH1_INT Bit0 = PWM0_CH0_INT
INT_BA+0x1C	7	[3:0]	Bit3 = PWM1_CH3_INT Bit2 = PWM1_CH2_INT Bit1 = PWM1_CH1_INT Bit0 = PWM1_CH0_INT
INT_BA+0x20	8	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = TMR0_INT
INT_BA+0x24	9	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = TMR1_INT
INT_BA+0x28	10	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = TMR2_INT
INT_BA+0x2C	11	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = TMR3_INT
INT_BA+0x30	12	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = UART0_INT
INT_BA+0x34	13	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = UART1_INT
INT_BA+0x38	14	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = SPI0_INT

Address	INT Number	Bits	Description
INT_BA+0x3C	15	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = SPI1_INT
INT_BA+0x40	16	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = SPI2_INT
INT_BA+0x44	17	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = IRC_INT
INT_BA+0x48	18	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = I2C0_INT
INT_BA+0x4C	19	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = I2C1_INT
INT_BA+0x50	20	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = SC2_INT
INT_BA+0x54	21	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = SC0_INT
INT_BA+0x58	22	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = SC1_INT
INT_BA+0x5C	23	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = USB_INT
INT_BA+0x64	25	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = LCD_INT
INT_BA+0x68	26	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = DMA_INT
INT_BA+0x6C	27	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = I2S_INT
INT_BA+0x70	28	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = PD_WU_INT

Address	INT Number	Bits	Description
INT_BA+0x74	29	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = ADC_INT
INT_BA+0x78	30	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = DAC_INT
INT_BA+0x7C	31	[2:0]	Bit2 = 1'b0 Bit1 = 1'b0 Bit0 = RTC_INT

NMI Interrupt Source Select Control Register (NMI_SEL)

Register	Offset	R/W	Description	Reset Value
NMI_SEL	INT_BA+0x80	R/W	NMI source interrupt select control register	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
-							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
-							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-			NMI_SEL[4:0]				

Bits	Description	
[31:8]	-	**Reserved**
[7:5]	-	**Reserved**
[4:0]	**NMI_SEL**	The NMI interrupt to Cortex-M0 can be selected from one of the interrupt[31:0] The NMI_SEL bit[4:0] used to select the NMI interrupt source

MCU Interrupt Request Source Register (MCU_IRQ)

Register	Offset	R/W	Description	Reset Value
MCU_IRQ	INT_BA+0x84	R/W	MCU interrupt request source register	0x0000_0000

31	30	29	28	27	26	25	24
MCU_IRQ							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
MCU_IRQ							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MCU_IRQ							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MCU_IRQ							

Bits	Description	
[31:0]	**MCU_IRQ**	**MCU IRQ Source Register** The MCU_IRQ collects all the interrupts from the peripherals and generates the synchronous interrupt to MCU Cortex-M0. There are two modes to generate interrupt to Cortex-M0, the normal mode. The MCU_IRQ collects all interrupts from each peripheral and synchronizes them and then interrupts the Cortex-M0. When the MCU_IRQ[n] is “0”, setting MCU_IRQ[n] “1” will generate an interrupt to Cortex_M0 NVIC[n]. When the MCU_IRQ[n] is “1” (means an interrupt is asserted), setting the MCU_bit[n] will clear the interrupt Set MCU_IRQ[n] “0” : no any effect

5.4 System Manager

5.4.1 Overview

System manager mainly controls the power modes, wake-up source, system resets and system memory map. It also provides information about product ID, chip reset, IP reset, and multi-function pin control.

5.4.2 Features

- Power modes and wake-up sources
- System resets
- System Memory Map
- System manager registers for :
  - Product ID
  - Chip and IP reset
  - Multi-functional pin control

5.4.3 Functional Description

5.4.3.1 Power modes and Wake-up sources

There are several power modes in this chip, depending on the clock status (ON or OFF).

Clocks:

- External 32.768 kHz Low Speed Crystal (LXT)
- External 4~ 24 MHz High Speed Crystal (HXT)
- Internal RC 12 MHz High Speed Oscillator Clock (HIRC)
- Internal 10 kHz Low Speed Oscillator Clock (LIRC)

Power Modes:

- Normal mode: CPU runs normally and all clocks ON.
- Idle mode: CPU entered sleep mode. CPU clock stops and other clocks ON.
- Power-down mode: All clocks stop, except LXT and LIRC, and SRAM retention

After chip enters power down, the following wake-up sources can wake chip up to Normal mode and list the condition about how to enter pown-down mode again for each peripheral.

Wake-up Source	Wake-up condition	System can enter Power-down mode:
External Interrupts	-	After software writes 1 to clear the GPIOx_ISRC bit.
UART	Data wake-up	Immediately after wake-up.
	CTSn wake-up	Requiring 2 UART_CLK after wake-up.
GPIO	-	After Software writes 1 to clear the GPIOx_ISRC bit.
RTC	-	Requiring 1 RTC_CLK (about 30 us) after wake-up.
USB	-	Immediately after wake up
SPI	-	After SPI slave clock goes from high to low.
Timer	TMRx_ISR[TCAP_IS]	After software sets timer TMRx_ISR[SW_RST] (software reset) or software writes 1 to clear TMRx_ISR[TCAP_IS].
	TMRx_ISR[TMR_IS]	After software sets timer TMRx_ISR[SW_RST] (software reset) or software writes 1 to clear TMRx_ISR[TMR_IS].
WDT	-	Immediately after wake-up.
BOD	-	After voltage is raised higher than target voltage or BOFx_INT_EN is set to low.
I^2C	-	Immediately after wake-up.

Table 5.4-1 Condition of Entering Power-down Mode Again Table

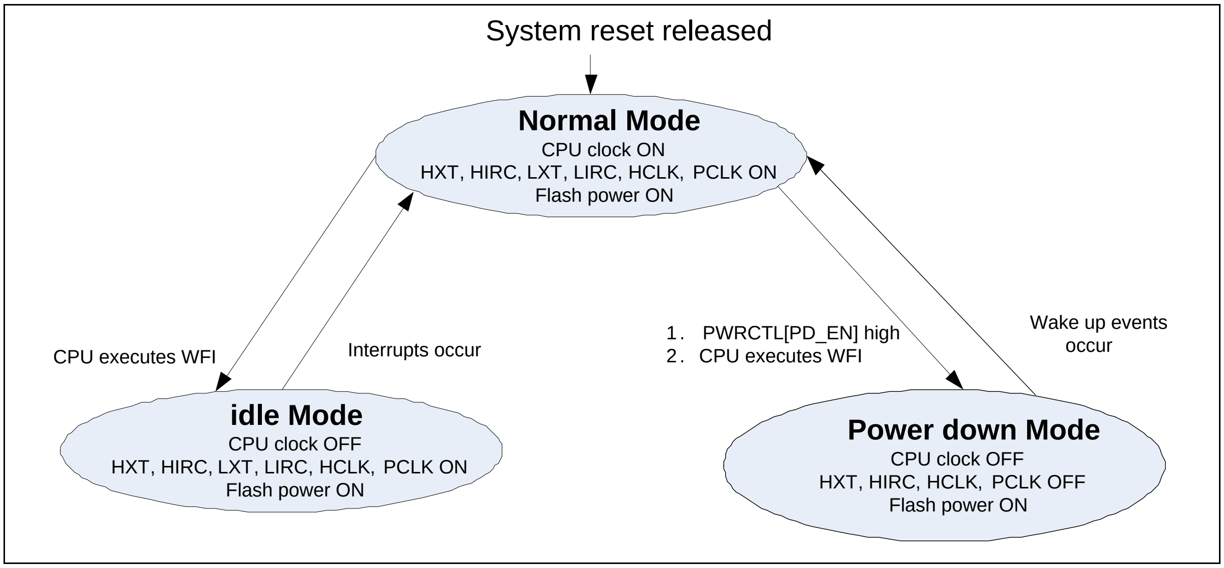


Figure 5.4-1 Power Modes

Register	Normal Mode	Idle Mode	Power-down Mode
Wake-up time to Normal mode	-	-	7us: from wake-up event to first CPU core valid clock 10us: from interrupt event to interrupt service routine first instruction.
HXT (4~24 MHz XTL)	ON	ON	Halt
HIRC (12 MHz OSC)	ON	ON	Halt
LXT (32 kHz XTL)	ON	ON	ON/OFF[1]
LIRC (10 kHz OSC)	ON	ON	ON/OFF[2]
PLL	ON/OFF	ON/OFF	Halt
LDO	ON	ON	ON
CPU	ON	Halt	Halt
HCLK/PCLK	ON	ON	Halt
SRAM retention	ON	ON	ON
FLASH	ON	ON	ON
GPIO	ON	ON	Halt
DMA	ON	ON	Halt
I^2C	ON	ON	Halt
PWM	ON	ON	Halt

TIMER	ON	ON	ON/OFF[3]
UART	ON	ON	ON/OFF[4]
SPI	ON	ON	Halt
RTC	ON	ON	ON/OFF[5]
WDT	ON	ON	ON/OFF[6]
WWDT	ON	ON	Halt
USB	ON	ON	Halt
LCD	ON	ON	ON/OFF[7]
I^2S	ON	ON	Halt
ADC	ON	ON	Halt
DAC	ON	ON	Halt

Table 5.4-2 IP clock ON/OFF in Power Modes

1. LXT (32 kHz XTL) ON or OFF is depended on SW setting in run mode
2. LIRC (10 kHz OSC) ON or OFF is depended on S/W setting in run mode
3. TIMER can work if LXT or LIRC is ON.
4. UART can work if LXT is ON.
5. RTC can work if LXT is ON
6. WDT can work if LIRC is ON.
7. LCD can work if LXT is ON.

Note: If CPU clock is not from HIRC (12 MHz RC oscillator), users must enable HIRC before entering Power-down mode to avoid wake-up fail. The power-down circuit will automatically disable HIRC in Power-down mode to save power consumption. Users can disable HIRC after wake-up to Normal mode to reduce extra current consumption if HIRC is not needed anymore in Normal mode.

5.4.3.2 System Resets

The system reset includes the events listed bellow. These reset events can be read by the RST_SRC register.

- Power-on Reset (POR)
- Brown-out Reset (BOD)
- Low level on the nRESET pin
- Watchdog Timer Time-out Reset
- Cortex-M0 MCU Reset

5.4.3.3 Auto-trim

This chip supports auto-trim function: the HIRC trim (12 MHz RC oscillator), according to the accurate LXT (32.768 kHz crystal oscillator), automatically gets accurate HIRC output frequency, 0.25% deviation within all temperature ranges. For instance, PLL needs an accurate 12 MHz input clock to output 48 MHz clock in USB applications. In such case, if users do not want to use 12 MHz HXT as PLL input clock, they need to solder 32.768 kHz crystal in system, and set the HIRC target output clock to 12 MHz by setting HIRCTRIMCTL[TRIM_SEL] to “10”, and the auto-trim function will be enabled. It is recommended to set both TRIM_LOOP and TRIM_RETRY_CNT to “11” to get better results.

5.4.4 Register and Memory Map

Open lock sequence for protected registers

Some of the system control registers need to be protected to avoid inadvertent write and disturb the chip operation. These system control registers are locked after the power-on reset till users to open the lock. For users to program these protected registers, an open lock sequence needs to be followed by a special programming sequence. The open sequence is to continually write the data “59h”, “16h” “88h” to the key controller address 0x5000_0100(RegLockAddr). Any different data value or different sequence or any other write operations to any other address during these three data program aborts the whole sequence. Therefore, users only follow the order of these three data and do not need to care the time interval when writing them.

After the lock is opened, users can check the lock bit RegLockAddr[0]. “1” is unlocked, “0” is locked. Then users can update the target register value if RegUnLock is high and write any data to the address “0x5000_0100” to re-lock the protected registers

Register	Offset	R/W	Description	Reset Value
GCR Base Address: **GCR_BA = 0x5000_0000**				
PDID	GCR_BA+0x00	R	Part Device Identification number Register	0x0014_0018[1]
RST_SRC	GCR_BA+0x04	R/W	System Reset Source Register	0x0000_00xx
IPRST_CTL1	GCR_BA+0x08	R/W	IP Reset Control Resister1	0x0000_0000
IPRST_CTL2	GCR_BA+0x0C	R/W	IP Reset Control Resister2	0x0000_0000
TEMPCTL	GCR_BA+0x20	R/W	Temperature Sensor Control Register	0x0000_0000
PA_L_MFP	GCR_BA+0x30	R/W	Port A low byte multiple function control register	0x0000_0000
PA_H_MFP	GCR_BA+0x34	R/W	Port A high byte multiple function control register	0x0000_0000
PB_L_MFP	GCR_BA+0x38	R/W	Port B low byte multiple function control register	0x0000_0000
PB_H_MFP	GCR_BA+0x3C	R/W	Port B high byte multiple function control register	0x0000_0000
PC_L_MFP	GCR_BA+0x40	R/W	Port C low byte multiple function control register	0x0000_0000
PC_H_MFP	GCR_BA+0x44	R/W	Port C high byte multiple function control register	0x0000_0000
PD_L_MFP	GCR_BA+0x48	R/W	Port D low byte multiple function control register	0x0000_0000
PD_H_MFP	GCR_BA+0x4C	R/W	Port D high byte multiple function control register	0x0000_0000
PE_L_MFP	GCR_BA+0x50	R/W	Port E low byte multiple function control register	0x0000_0000
PE_H_MFP	GCR_BA+0x54	R/W	Port E high byte multiple function control register	0x0000_0000
PF_L_MFP	GCR_BA+0x58	R/W	Port F low byte multiple function control register	0x0077_7777
PORCTL	GCR_BA+0x60	R/W	Power-On-Reset Controller Register	0x0000_0000
BODCTL	GCR_BA+0x64	R/W	Brown-out Detector Controller Register	0x00FF_F0xx
BODSTS	GCR_BA+0x68	R	Brown-out Detector Status Register	0x0000_0000

Int_VREFCTL	GCR_BA+0x6C	R/W	Voltage reference Control register	0x0000_0F00
IRCTRIMCTL	GCR_BA+0x80	R/W	HIRC Trim Control Register	0x0000_0000
IRCTRIMIEN	GCR_BA+0x84	R/W	HIRC Trim Interrupt Enable Register	0x0000_0000
IRCTRIMINT	GCR_BA+0x88	R/W	HIRC Trim Interrupt Status Register	0x0000_0000
RegLockAddr	GCR_BA+0x100	R/W	Register Lock Key address	0x0000_0000

5.4.5 Register Description

Part Device Identification Number Register (PDID)

Register	Offset	R/W	Description	Reset Value
PDID	GCR_BA+0x00	R	Part Device Identification number Register	0x0014_0018[1]

[1] Every part device identification has a unique default reset value.

31	30	29	28	27	26	25	24
PDID							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PDID							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDID							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDID							

Bits	Description	
[31:0]	**PDID**	**Part Device ID** This register reflects device part number code. Software can read this register to identify which device is used.

System Reset Source Register (RST_SRC)

This register provides specific information for software to identify the chip's reset source from the last operation.

Register	Offset	R/W	Description	Reset Value
RST_SRC	GCR_BA+0x04	R/W	System Reset Source Register	0x0000_00xx

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RSTS_CPU	**Reserved**	**RSTS_SYS**	**RSTS_BOD**	**Reserved**	**RSTS_WDT**	**RSTS_PAD**	**RSTS_POR**

Bits	Description	
[31:8]	**Reserved**	**Reserved**
[7]	**RSTS_CPU**	The RSTS_CPU flag is set by hardware if software writes CPU_RST (IPRST_CTL1[1]) "1" to rest Cortex-M0 CPU kernel and Flash memory controller (FMC). 1 = Cortex-M0 CPU kernel and FMC are reset by software setting CPU_RST to 1. 0 = No reset from CPU This bit is cleared by writing 1 to itself.
[6]	**Reserved**	**Reserved**
[5]	**RSTS_SYS**	The RSTS_SYS flag is set by the "reset signal" from the Cortex_M0 kernel to indicate the previous reset source. 1 = Cortex_M0 had issued the reset signal to reset the system by writing 1 to the bit SYSRESTREQ(AIRCR[2], Application Interrupt and Reset Control Register) in system control registers of Cortex_M0 kernel. 0 = No reset from Cortex_M0 This bit is cleared by writing 1 to itself.
[4]	**RSTS_BOD**	The RSTS_BOD flag is set by the "reset signal" from the Brown-out-Detected module to indicate the previous reset source. 1 = Brown-out-Detected module had issued the reset signal to reset the system. 0 = No reset from BOD This bit is cleared by writing 1 to itself.
[3]	**Reserved**	**Reserved**
[2]	**RSTS_WDT**	The RSTS_WDT flag is set by the "reset signal" from the Watchdog Timer module to indicate the previous reset source. 1 = The Watchdog Timer module had issued the reset signal to reset the system. 0 = No reset from Watchdog Timer

Bits	Description	
		This bit is cleared by writing 1 to itself.
[1]	**RSTS_PAD**	The RSTS_PAD flag is set by the "reset signal" from the /RESET pin to indicate the previous reset source. 1 = The /RESET pin had issued the reset signal to reset the system. 0 = No reset from /RESET pin This bit is cleared by writing 1 to itself.
[0]	**RSTS_POR**	The RSTS_POR flag is set by the "reset signal" from the Power-on Reset (POR) module or bit CHIP_RST (IPRSTC1[0]) to indicate the previous reset source. 1 = Power-on Reset (POR) or CHIP_RST had issued the reset signal to reset the system. 0 = No reset from POR or CHIP_RST This bit is cleared by writing 1 to itself.

IP Reset Control Register1 (IPRST_CTL1)

Register	Offset	R/W	Description	Reset Value
IPRST_CTL1	GCR_BA+0x08	R/W	IP Reset Control Resister1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
20	**20**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
12	**12**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**EBI_RST**	**DMA_RST**	**CPU_RST**	**CHIP_RST**

Bits	Description	
[31:4]	**Reserved**	**Reserved**
[3]	**EBI_RST**	**EBI Controller Reset** This is a protected register. Please refer to open lock sequence to program it. Set this bit "1" will generate a reset signal to the EBI. SW needs to set this bit to low to release reset signal. 0 = Normal operation 1 = EBI IP reset
[2]	**DMA_RST**	**DMA Controller Reset** This is a protected register. Please refer to open lock sequence to program it. Set this bit "1" will generate a reset signal to the DMA. SW needs to set this bit to low to release reset signal. 0 = Normal operation 1 = DMA IP reset
[1]	**CPU_RST**	**CPU kernel one shot reset.** This is a protected register. Please refer to open lock sequence to program it. Setting this bit will only reset the CPU kernel and Flash Memory Controller(FMC), and this bit will automatically return to "0" after the 2 clock cycles 0 = Normal 1 = Reset CPU
[0]	**CHIP_RST**	**CHIP one shot reset.** This is a protected register. Please refer to open lock sequence to program it. Setting this bit will reset the whole chip, including CPU kernel and all peripherals like power-on reset and this bit will automatically return to "0" after the 2 clock cycles. The chip setting from flash will be also reloaded when chip one shot reset. 0 = Normal

Bits	Description	
		1 = Reset CHIP **Note:** In the following conditions, chip setting from flash will be reloaded. Power-on Reset Brown-out-Detected Reset Low level on the /RESET pin Set IPRST_CTL1[CHIP_RST]

IP Reset Control Register2 (IPRST_CTL2)

Register	Offset	R/W	Description	Reset Value
IPRST_CTL2	GCR_BA+0x0C	R/W	IP Reset Control Resister2	0x0000_0000

31	30	29	28	27	26	25	24
SC1_RST	SC0_RST	I2S_RST	ADC_RST	USBD_RST	LCD_RST	DAC_RST	Reserved
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		PWM1_RST	PWM0_RST	Reserved		UART1_RST	UART0_RST
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	SPI2_RST	SPI1_RST	SPI0_RST	Reserved		I2C1_RST	I2C0_RST
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC2_RST	Reserved	TMR3_RST	TMR2_RST	TMR1_RST	TMR0_RST	GPIO_RST	Reserved

Bits	Description	
[31]	**SC1_RST**	**SmartCard1 Controller Reset** 0 = SmartCard block normal operation 1 = SmartCard block reset
[30]	**SC0_RST**	**SmartCard 0 Controller Reset** 0 = SmartCard block normal operation 1 = SmartCard block reset
[29]	**I2S_RST**	**I2S Controller Reset** 0 = I2S block normal operation 1 = I2S block reset
[28]	**ADC_RST**	**ADC Controller Reset** 0 = ADC block normal operation 1 = ADC block reset
[27]	**USBD_RST**	**USB Device Controller Reset** 0 = USB block normal operation 1 = USB block reset
[26]	**LCD_RST**	**LCD Controller Reset** 0 = LCD block normal operation 1 = LCD block reset
[25]	**DAC_RST**	**DAC Controller Reset** 0 = DAC block normal operation 1 = DAC block reset
[24:22]	**Reserved**	**Reserved**
[21]	**PWM1_RST**	**PWM1 controller Reset**

Bits	Description	
		0 = PWM1 block normal operation 1 = PWM1 block reset
[20]	PWM0_RST	**PWM0 controller Reset** 0 = PWM0 block normal operation 1 = PWM0 block reset
[19:18]	Reserved	Reserved
[17]	UART1_RST	**UART1 controller Reset** 0 = UART1 normal operation 1 = UART1 block reset
[16]	UART0_RST	**UART0 controller Reset** 0 = UART0 normal operation 1 = UART0 block reset
[15]	Reserved	Reserved
[14]	SPI2_RST	**SPI2 controller Reset** 0 = SPI2 normal operation 1 = SPI2 block reset
[13]	SPI1_ RST	**SPI1 controller Reset** 0 = SPI1 normal operation 1 = SPI1 block reset
[12]	SPI0_ RST	**SPI0 controller Reset** 0 = SPI0 block normal operation 1 = SPI0 block reset
[11:10]	Reserved	Reserved
[9]	I2C1_RST	**I2C1 controller Reset** 0 = I2C1 block normal operation 1 = I2C1 block reset
[8]	I2C0_RST	**I2C0 controller Reset** 0 = I2C0 normal operation 1 = I2C0 block reset
[7]	SC2_RST	**SmartCard 2 Controller Reset** 0 = SmartCard 2 block normal operation 1 = SmartCard 2 block reset
[6]	Reserved	Reserved
[5]	TMR3_RST	**Timer3 controller Reset** 0 = Timer3 normal operation 1 = Timer3 block reset
[4]	TMR2_RST	**Timer2 controller Reset** 0 = Timer2 normal operation

Bits	Description	
		1 = Timer2 block reset
[3]	**TMR1_RST**	**Timer1 controller Reset** 0 = Timer1 normal operation 1 = Timer1 block reset
[2]	**TMR0_RST**	**Timer0 controller Reset** 0 = Timer0 normal operation 1 = Timer0 reset
[1]	**GPIO_RST**	**GPIO controller Reset** 0 = GPIO normal operation 1 = GPIO reset
[0]	**Reserved**	**Reserved**

Temperature Sensor Control Register (TEMPCTL)

Register	Offset	R/W	Description	Reset Value
TEMPCTL	GCR_BA+0x20	R/W	Temperature Sensor Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**VTEMP_EN**

Bits	Description	
[31:1]	**Reserved**	**Reserved**
[0]	**VTEMP_EN**	**Temperature Sensor Enable** 1 = Temperature sensor function Enabled 0 = Temperature sensor function Disabled (default).

Multiple Function Port A Low Byte Control Register (PA_L_MFP)

Register	Offset	R/W	Description	Reset Value
PA_L_MFP	GCR_BA+0x30	R/W	Port A low byte multiple function control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	**PA7_MFP**			**Reserved**	**PA6_MFP**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**PA5_MFP**			**Reserved**	**PA4_MFP**		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**PA3_MFP**			**Reserved**	**PA2_MFP**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**PA1_MFP**			**Reserved**	**PA0_MFP**		

Bits	Description	
[31]	**Reserved**	**Reserved**
[30:28]	**PA7_MFP**	**PA.7 Pin Function Selection** **At LQFP-128 Package :** PA7_MFP / Function: 111 — LCD SEG 36 101 — PWM0 Channel 2 100 — SmartCard 2 data pin 011 — Timer 2 capture event 010 — EBI AD[6] 001 — ADC input channel 7 Others — GPIOA[7] **At LQFP-64 Package :** None **At LQFP-48 Package :** None
[27]	**Reserved**	**Reserved**
[26:24]	**PA6_MFP**	**PA.6 Pin Function Selection** **At LQFP-128 Package :** PA6_MFP / Function

Bits	Description		
		111	LCD SEG 37
		101	PWM0 Channel 3
		100	SmartCard 2 clock
		011	Timer 3 Capture event
		010	EBI AD[7]
		001	ADC input channel 6
		Others	GPIOA[6]
		At LQFP-64 Package :	
		PA6_MFP	**Function**
		111	LCD SEG 19
		101	PWM0 Channel 3
		100	SmartCard 2 clock
		011	Timer 3 Capture event
		010	EBI AD[7]
		001	ADC input channel 6
		Others	GPIOA[6]
		At LQFP-48 Package :	
		PA6_MFP	**Function**
		101	PWM0 Channel 3
		100	SmartCard 2 clock
		011	Timer 3 Capture event
		010	EBI AD[7]
		001	ADC input channel 6
		Others	GPIOA[6]
[23]	**Reserved**	**Reserved**	
[22:20]	**PA5_MFP**	**PA.5 Pin Function Selection**	
		At LQFP-128 Package :	
		PA5_MFP	**Function**
		111	LCD SEG 38
		101	I^2C0 SCL
		100	SmartCard2 RST
		010	EBI AD[8]
		001	ADC input channel 5

Bits	Description	
		Others: GPIOA[5] **At LQFP-64 Package :** PA5_MFP / Function: 111: LCD SEG 20 101: I^2C0 SCL 100: SmartCard2 RST 010: EBI AD[8] 001: ADC input channel 5 Others: GPIOA[5] **At LQFP-48 Package :** PA5_MFP / Function: 101: I^2C0 SCL 100: SmartCard2 RST 010: EBI AD[8] 001: ADC input channel 5 Others: GPIOA[5]
[19]	**Reserved**	**Reserved**
[18:16]	**PA4_MFP**	**PA.4 Pin Function Selection** **At LQFP-128 Package :** PA4_MFP / Function: 111: LCD SEG 39 101: I^2C0 SDA 100: SmartCard 2 power 010: EBI AD[9] 001: ADC input channel 4 Others: GPIOA[4] **At LQFP-64 Package :** PA4_MFP / Function: 111: LCD SEG 21 101: I^2C0 SDA 100: SmartCard 2 power 010: EBI AD[9]

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>001</td><td>ADC input channel 4</td></tr>
<tr><td>Others</td><td>GPIOA[4]</td></tr>
</table>
At LQFP-48 Package :
<table>
<tr><th>PA4_MFP</th><th>Function</th></tr>
<tr><td>101</td><td>I2C0SDA</td></tr>
<tr><td>100</td><td>SmartCard 2 power</td></tr>
<tr><td>010</td><td>EBI AD[9]</td></tr>
<tr><td>001</td><td>ADC input channel 4</td></tr>
<tr><td>Others</td><td>GPIOA[4]</td></tr>
</table>
</td></tr>
<tr><td>[15]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[14:12]</td><td>PA3_MFP</td><td>
PA.3 Pin Function Selection
At LQFP-128 Package :
<table>
<tr><th>PA3_MFP</th><th>Function</th></tr>
<tr><td>101</td><td>UART1_TXD</td></tr>
<tr><td>010</td><td>EBI AD[10]</td></tr>
<tr><td>001</td><td>ADC input channel 3</td></tr>
<tr><td>Others</td><td>GPIOA[3]</td></tr>
</table>
At LQFP-64 Package :
<table>
<tr><th>PA3_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 22</td></tr>
<tr><td>101</td><td>UART1_TXD</td></tr>
<tr><td>010</td><td>EBI AD[10]</td></tr>
<tr><td>001</td><td>ADC input channel 3</td></tr>
<tr><td>Others</td><td>GPIOA[3]</td></tr>
</table>
At LQFP-48 Package :
<table>
<tr><th>PA3_MFP</th><th>Function</th></tr>
<tr><td>101</td><td>UART1_TXD</td></tr>
<tr><td>010</td><td>EBI AD[10]</td></tr>
<tr><td>001</td><td>ADC input channel 3</td></tr>
<tr><td>Others</td><td>GPIOA[3]</td></tr>
</table>
</td></tr>
<tr><td>[11]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[10:8]</td><td>PA2_MFP</td><td>PA.2 Pin Function Selection</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>
At LQFP-128 Package :
<table>
<tr><th>PA2_MFP</th><th>Function</th></tr>
<tr><td>101</td><td>UART1_RXD</td></tr>
<tr><td>010</td><td>EBI AD[11]</td></tr>
<tr><td>001</td><td>ADC input channel 2</td></tr>
<tr><td>Others</td><td>GPIOA[2]</td></tr>
</table>
At LQFP-64 Package :
<table>
<tr><th>PA2_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 23</td></tr>
<tr><td>101</td><td>UART1_RXD</td></tr>
<tr><td>010</td><td>EBI AD[11]</td></tr>
<tr><td>001</td><td>ADC input channel 2</td></tr>
<tr><td>Others</td><td>GPIOA[2]</td></tr>
</table>
At LQFP-48 Package :
<table>
<tr><th>PA2_MFP</th><th>Function</th></tr>
<tr><td>101</td><td>UART1_RXD</td></tr>
<tr><td>010</td><td>EBI AD[11]</td></tr>
<tr><td>001</td><td>ADC input channel 2</td></tr>
<tr><td>Others</td><td>GPIOA[2]</td></tr>
</table>
</td></tr>
<tr><td>[7]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[6:4]</td><td>PA1_MFP</td><td>
PA.1 Pin Function Selection
At LQFP-128 Package :
<table>
<tr><th>PA1_MFP</th><th>Function</th></tr>
<tr><td>010</td><td>EBI AD[12]</td></tr>
<tr><td>001</td><td>ADC input channel 1</td></tr>
<tr><td>Others</td><td>GPIOA[1]</td></tr>
</table>
At LQFP-64 Package :
<table>
<tr><th>PA1_MFP</th><th>Function</th></tr>
<tr><td>010</td><td>EBI AD[12]</td></tr>
<tr><td>001</td><td>ADC input channel 1</td></tr>
<tr><td>Others</td><td>GPIOA[1]</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>

At LQFP-48 Package :

<table>
<tr><th>PA1_MFP</th><th>Function</th></tr>
<tr><td>010</td><td>EBI AD[12]</td></tr>
<tr><td>001</td><td>ADC input channel 1</td></tr>
<tr><td>Others</td><td>GPIOA[1]</td></tr>
</table>
</td></tr>
<tr><td>[3]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[2:0]</td><td>PA0_MFP</td><td>

PA.0 Pin Function Selection

At LQFP-128 Package :

<table>
<tr><th>PA0_MFP</th><th>Function</th></tr>
<tr><td>100</td><td>SmartCard 2 card detect</td></tr>
<tr><td>001</td><td>ADC input channel 0</td></tr>
<tr><td>Others</td><td>GPIOA[0]</td></tr>
</table>

At LQFP-64 Package :

<table>
<tr><th>PA0_MFP</th><th>Function</th></tr>
<tr><td>100</td><td>SmartCard 2 card detect</td></tr>
<tr><td>001</td><td>ADC input channel 0</td></tr>
<tr><td>Others</td><td>GPIOA[0]</td></tr>
</table>

At LQFP-48 Package :

<table>
<tr><th>PA0_MFP</th><th>Function</th></tr>
<tr><td>100</td><td>SmartCard 2 card detect</td></tr>
<tr><td>001</td><td>ADC input channel 0</td></tr>
<tr><td>Others</td><td>GPIOA[0]</td></tr>
</table>
</td></tr>
</table>

Multiple Function Port A High Byte Control Register (PA_H_MFP)

Register	Offset	R/W	Description	Reset Value
PA_H_MFP	GCR_BA+0x34	R/W	Port A high byte multiple function control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	**PA15_MFP**			**Reserved**	**PA14_MFP**		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	**PA13_MFP**			**Reserved**	**PA12_MFP**		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**PA11_MFP**			**Reserved**	**PA10_MFP**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**PA9_MFP**			**Reserved**	**PA8_MFP**		

Bits	Description	
[31]	**Reserved**	**Reserved.**
[30:28]	**PA15_MFP**	**PA.15 Pin Function Selection** **At LQFP-128 Package :** PA15_MFP / Function: 110: UART0 TX 100: SmartCard 0 power 011: Timer3 capture event 010: I²S MCLK 001: PWM0 Channel 3 Others: GPIOA[15] **At LQFP-64 Package :** PA15_MFP / Function: 111: LCD SEG 27 110: UART0 TX 100: SmartCard 0 power 011: Timer3 capture event 010: I²S MCLK 001: PWM0 Channel 3 Others: GPIOA[15]

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>

At LQFP-48 Package :

PA15_MFP	Function
110	UART0 TX
100	SmartCard 0 power
011	Timer3 capture event
010	I2S MCLK
001	PWM0 Channel 3
Others	GPIOA[15]

</td></tr>
<tr><td>[27]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[26:24]</td><td>PA14_MFP</td><td>

PA.14 Pin Function Selection

At LQFP-128 Package :

PA14_MFP	Function
110	UART0 RX
011	Timer2 capture event
010	EBI AD[15]
001	PWM0 Channel 2
Others	GPIOA[14]

At LQFP-64 Package :

PA14_MFP	Function
111	LCD SEG 26
110	UART0 RX
011	Timer2 capture event
010	EBI AD[15]
001	PWM0 Channel 2
Others	GPIOA[14]

At LQFP-48 Package :

PA14_MFP	Function
110	UART0 RX
011	Timer2 capture event
010	EBI AD[15]
001	PWM0 Channel 2
Others	GPIOA[14]

</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[23]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[22:20]</td><td>PA13_MFP</td><td>PA.13 Pin Function Selection
At LQFP-128 Package :
<table>
<tr><th>PA13_MFP</th><th>Function</th></tr>
<tr><td>101</td><td>I^{2}C0 SCL</td></tr>
<tr><td>011</td><td>Timer1 capture event</td></tr>
<tr><td>010</td><td>EBI AD[14]</td></tr>
<tr><td>001</td><td>PWM0 Channel 1</td></tr>
<tr><td>Others</td><td>GPIOA[13]</td></tr>
</table>
At LQFP-64 Package :
<table>
<tr><th>PA13_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 25</td></tr>
<tr><td>101</td><td>I^{2}C0 SCL</td></tr>
<tr><td>011</td><td>Timer1 capture event</td></tr>
<tr><td>010</td><td>EBI AD[14]</td></tr>
<tr><td>001</td><td>PWM0 Channel 1</td></tr>
<tr><td>Others</td><td>GPIOA[13]</td></tr>
</table>
At LQFP-48 Package :
<table>
<tr><th>PA13_MFP</th><th>Function</th></tr>
<tr><td>101</td><td>I^{2}C0 SCL</td></tr>
<tr><td>011</td><td>Timer1 capture event</td></tr>
<tr><td>010</td><td>EBI AD[14]</td></tr>
<tr><td>001</td><td>PWM0 Channel 1</td></tr>
<tr><td>Others</td><td>GPIOA[13]</td></tr>
</table></td></tr>
<tr><td>[19]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[18:16]</td><td>PA12_MFP</td><td>PA.12 Pin Function Selection
At LQFP-128 Package :
<table>
<tr><th>PA12_MFP</th><th>Function</th></tr>
<tr><td>101</td><td>I^{2}C0 SDA</td></tr>
<tr><td>011</td><td>Timer0 capture event</td></tr>
<tr><td>010</td><td>EBI AD[13]</td></tr>
<tr><td>001</td><td>PWM0 Channel 0</td></tr>
<tr><td>Others</td><td>GPIOA[12]</td></tr>
</table></td></tr>
</table>

Bits	Description	
		At LQFP-64 Package : | PA12_MFP | Function | |---|---| | 111 | LCD SEG 24 | | 101 | I^2C0 SDA | | 011 | Timer0 capture event | | 010 | EBI AD[13] | | 001 | PWM0 Channel 0 | | Others | GPIOA[12] | At LQFP-48 Package : | PA12_MFP | Function | |---|---| | 101 | I^2C0 SDA | | 011 | Timer0 capture event | | 010 | EBI AD[13] | | 001 | PWM0 Channel 0 | | Others | GPIOA[12] |
[15]	**Reserved**	**Reserved**
[14:12]	**PA11_MFP**	**PA.11 Pin Function Selection** At LQFP-128 Package : | PA11_MFP | Function | |---|---| | 111 | LCD SEG 23 | | 100 | SPI2 MOSI0 | | 011 | SmartCard0 RST | | 010 | EBI nRD | | 001 | I^2C1 SCL | | Others | GPIOA[11] | At LQFP-64 Package : | PA11_MFP | Function | |---|---| | 111 | LCD SEG 9 | | 100 | SPI2 MOSI0 | | 011 | SmartCard0 RST | | 010 | EBI nRD | | 001 | I^2C1 SCL |

<table>
<tr><th>Bits</th><th>Description</th><th></th></tr>
<tr><td></td><td></td><td>
<table><tr><td>Others</td><td>GPIOA[11]</td></tr></table>
At LQFP-48 Package :
<table>
<tr><th>PA11_MFP</th><th>Function</th></tr>
<tr><td>100</td><td>SPI2 MOSI0</td></tr>
<tr><td>011</td><td>SmartCard0 RST</td></tr>
<tr><td>010</td><td>EBI nRD</td></tr>
<tr><td>001</td><td>I2C1 SCL</td></tr>
<tr><td>Others</td><td>GPIOA[11]</td></tr>
</table>
</td></tr>
<tr><td>[11]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[10:8]</td><td>PA10_MFP</td><td>
PA.10 Pin Function Selection
At LQFP-128 Package :
<table>
<tr><th>PA10_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 22</td></tr>
<tr><td>100</td><td>SPI2 MISO0</td></tr>
<tr><td>011</td><td>SmartCard0 Power</td></tr>
<tr><td>010</td><td>EBI nWR</td></tr>
<tr><td>001</td><td>I2C1 SDA</td></tr>
<tr><td>Others</td><td>GPIOA[10]</td></tr>
</table>
At LQFP-64 Package :
<table>
<tr><th>PA10_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 8</td></tr>
<tr><td>100</td><td>SPI2 MISO0</td></tr>
<tr><td>011</td><td>SmartCard0 Power</td></tr>
<tr><td>010</td><td>EBI nWR</td></tr>
<tr><td>001</td><td>I2C1 SDA</td></tr>
<tr><td>Others</td><td>GPIOA[10]</td></tr>
</table>
At LQFP-48 Package :
<table>
<tr><th>PA10_MFP</th><th>Function</th></tr>
<tr><td>100</td><td>SPI2 MISO0</td></tr>
<tr><td>011</td><td>SmartCard0 Power</td></tr>
<tr><td>010</td><td>EBI nWR</td></tr>
<tr><td>001</td><td>I2C1 SDA</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Bits</th><th>Description</th><th></th></tr>
<tr><td></td><td></td><td><table><tr><td>Others</td><td>GPIOA[10]</td></tr></table></td></tr>
<tr><td>[7]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[6:4]</td><td>PA9_MFP</td><td>PA.9 Pin Function Selection
At LQFP-128 Package :
<table><tr><th>PA9_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 21</td></tr>
<tr><td>100</td><td>SPI2 SCLK</td></tr>
<tr><td>011</td><td>SmartCard0 DATA</td></tr>
<tr><td>001</td><td>I2C0 SCL</td></tr>
<tr><td>Others</td><td>GPIOA[9]</td></tr></table>
At LQFP-64 Package :
<table><tr><th>PA9_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 7</td></tr>
<tr><td>100</td><td>SPI2 SCLK</td></tr>
<tr><td>011</td><td>SmartCard0 DATA</td></tr>
<tr><td>001</td><td>I^2C0 SCL</td></tr>
<tr><td>Others</td><td>GPIOA[9]</td></tr></table>
At LQFP-48 Package :
<table><tr><th>PA9_MFP</th><th>Function</th></tr>
<tr><td>100</td><td>SPI2 SCLK</td></tr>
<tr><td>011</td><td>SmartCard0 DATA</td></tr>
<tr><td>001</td><td>I^2C0 SCL</td></tr>
<tr><td>Others</td><td>GPIOA[9]</td></tr></table></td></tr>
<tr><td>[3]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[2:0]</td><td>PA8_MFP</td><td>PA.8 Pin Function Selection
At LQFP-128 Package :
<table><tr><th>PA8_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 20</td></tr>
<tr><td>100</td><td>SPI2 1^{st} slave select pin</td></tr>
<tr><td>011</td><td>SmartCard0 clock</td></tr>
<tr><td>001</td><td>I^2C0 SDA</td></tr>
<tr><td>Others</td><td>GPIOA[8]</td></tr></table>
At LQFP-64 Package:</td></tr>
</table>

Bits	Description

PA8_MFP	Function
111	LCD SEG 6
100	SPI2 1st slave select pin
011	SmartCard0 clock
001	I2C0 SDA
Others	GPIOA[8]

At LQFP-48 Package :

PA8_MFP	Function
100	SPI2 1st slave select pin
011	SmartCard0 clock
001	I2C0 SDA
Others	GPIOA[8]

Multiple Function Port B Low Byte Control Register (PB_L_MFP)

Register	Offset	R/W	Description	Reset Value
PB_L_MFP	GCR_BA+0x38	R/W	Port B low byte multiple function control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	PB7_MFP			Reserved	PB6_MFP		
23	22	21	20	19	18	17	16
Reserved	PB5_MFP			Reserved	PB4_MFP		
15	14	13	12	11	10	9	8
Reserved	PB3_MFP			Reserved	PB2_MFP		
7	6	5	4	3	2	1	0
Reserved	PB1_MFP			Reserved	PB0_MFP		

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[30:28]</td><td>PB7_MFP</td><td>PB.7 Pin Function Selection
At LQFP-128 Package :
<table>
<tr><th>PB7_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 10</td></tr>
<tr><td>100</td><td>SPI2 MOSI0</td></tr>
<tr><td>010</td><td>EBI nCS</td></tr>
<tr><td>001</td><td>UART1 CTSn</td></tr>
<tr><td>Others</td><td>GPIOB[7]</td></tr>
</table>
At LQFP-64 Package :
<table>
<tr><th>PB7_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 2</td></tr>
<tr><td>100</td><td>SPI2 MOSI0</td></tr>
<tr><td>010</td><td>EBI nCS</td></tr>
<tr><td>001</td><td>UART1 CTSn</td></tr>
<tr><td>Others</td><td>GPIOB[7]</td></tr>
</table>
At LQFP-48 Package :
None</td></tr>
<tr><td>[27]</td><td>Reserved</td><td>Reserved</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[26:24]</td><td>PB6_MFP</td><td>PB.6 Pin Function Selection
At LQFP-128 Package :
<table><tr><th>PB6_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 11</td></tr><tr><td>100</td><td>SPI2 MISO0</td></tr><tr><td>010</td><td>EBI ALE</td></tr><tr><td>001</td><td>UART1 RTSn</td></tr><tr><td>Others</td><td>GPIOB[6]</td></tr></table>
At LQFP-64 Pckage :
<table><tr><th>PB6_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 3</td></tr><tr><td>100</td><td>SPI2 MISO0</td></tr><tr><td>010</td><td>EBI ALE</td></tr><tr><td>001</td><td>UART1 RTSn</td></tr><tr><td>Others</td><td>GPIOB[6]</td></tr></table>
At LQFP-48 Package :
None.</td></tr>
<tr><td>[23]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[22:20]</td><td>PB5_MFP</td><td>PB.5 Pin Function Selection
At LQFP-128 Package :
<table><tr><th>PB5_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 12</td></tr><tr><td>100</td><td>SPI2 SCLK</td></tr><tr><td>011</td><td>SmartCard0 RST</td></tr><tr><td>001</td><td>UART1 TX</td></tr><tr><td>Others</td><td>GPIOB[5]</td></tr></table>
At LQFP-64 Package:
<table><tr><th>PB5_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 4</td></tr><tr><td>011</td><td>SmartCard0 RST</td></tr><tr><td>100</td><td>SPI2 SCLK</td></tr><tr><td>001</td><td>UART1 TX</td></tr></table></td></tr>
</table>

Bits	Description	
		<table><tr><td>Others</td><td>GPIOB[5]</td></tr></table> **At LQFP-48 Package :** <table><tr><th>PB5_MFP</th><th>Function</th></tr><tr><td>011</td><td>SmartCard0 RST</td></tr><tr><td>100</td><td>SPI2 SCLK</td></tr><tr><td>001</td><td>UART1 TX</td></tr><tr><td>Others</td><td>GPIOB[5]</td></tr></table>
[19]	**Reserved**	**Reserved**
[18:16]	**PB4_MFP**	**PB.4 Pin Function Selection** **At LQFP-128 Package :** <table><tr><th>PB4_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 13</td></tr><tr><td>100</td><td>SPI2 1st slave select pin</td></tr><tr><td>011</td><td>SmartCard0 card detection</td></tr><tr><td>001</td><td>UART1 RX</td></tr><tr><td>Others</td><td>GPIOB[4]</td></tr></table> **At LQFP-64 Package :** <table><tr><th>PB4_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 5</td></tr><tr><td>100</td><td>SPI2 1st slave select pin</td></tr><tr><td>011</td><td>SmartCard0 card detection</td></tr><tr><td>001</td><td>UART1 RX</td></tr><tr><td>Others</td><td>GPIOB[4]</td></tr></table> **At LQFP-48 Package :** <table><tr><th>PB4_MFP</th><th>Function</th></tr><tr><td>100</td><td>SPI2 1st slave select pin</td></tr><tr><td>011</td><td>SmartCard0 card detection</td></tr><tr><td>001</td><td>UART1 RX</td></tr><tr><td>Others</td><td>GPIOB[4]</td></tr></table>
[15]	**Reserved**	**Reserved**
[14:12]	**PB3_MFP**	**PB.3 Pin Function Selection** **At LQFP-128 Package :**

<table>
<tr><th>Bits</th><th>Description</th><th></th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>PB3_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 4</td></tr>
<tr><td>011</td><td>SPI1 1st slave select pin</td></tr>
<tr><td>010</td><td>EBI nWRH</td></tr>
<tr><td>001</td><td>UART0 CTSn</td></tr>
<tr><td>Others</td><td>GPIOB[3]</td></tr>
</table>
At LQFP-64 Package :
<table>
<tr><th>PB3_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD COM 2</td></tr>
<tr><td>011</td><td>SPI1 1st slave select pin</td></tr>
<tr><td>010</td><td>EBI nWRH</td></tr>
<tr><td>001</td><td>UART0 CTSn</td></tr>
<tr><td>Others</td><td>GPIOB[3]</td></tr>
</table>
At LQFP-48 Package :
<table>
<tr><th>PB3_MFP</th><th>Function</th></tr>
<tr><td>011</td><td>SPI1 1st slave select pin</td></tr>
<tr><td>010</td><td>EBI nWRH</td></tr>
<tr><td>001</td><td>UART0 nCTS</td></tr>
<tr><td>Others</td><td>GPIOB[3]</td></tr>
</table>
</td></tr>
<tr><td>[11]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[10:8]</td><td>PB2_MFP</td><td>
PB.2 Pin Function Selection
At LQFP-128 Package :
<table>
<tr><th>PB2_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 5</td></tr>
<tr><td>011</td><td>SPI1 SCLK</td></tr>
<tr><td>010</td><td>EBI nWRL</td></tr>
<tr><td>001</td><td>UART0 RTSn</td></tr>
<tr><td>Others</td><td>GPIOB[2]</td></tr>
</table>
At LQFP-64 Package :
<table>
<tr><th>PB2_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD COM 3</td></tr>
</table>
</td></tr>
</table>

Bits	Description	
		011 — SPI1 SCLK 010 — EBI nWRL 001 — UART0 RTSn Others — GPIOB[2] **At LQFP-48 Package :** **PB2_MFP** — **Function** 011 — SPI1 SCLK 010 — EBI nWRL 001 — UART0 RTSn Others — GPIOB[2]
[7]	**Reserved**	**Reserved**
[6:4]	**PB1_MFP**	**PB.1 Pin Function Selection** **At LQFP-128 Package :** **PB1_MFP** — **Function** 111 — LCD SEG 6 011 — SPI1 MISO0 001 — UART0 TX Others — GPIOB[1] **At LQFP-64 Package :** **PB1_MFP** — **Function** 111 — LCD SEG 0 011 — SPI1 MISO0 001 — UART0 TX Others — GPIOB[1] **At LQFP-48 Package :** **PB1_MFP** — **Function** 011 — SPI1 MISO0 001 — UART0 TX Others — GPIOB[1]
[3]	**Reserved**	**Reserved**
[2:0]	**PB0_MFP**	**PB.0 Pin Function Selection** **At LQFP-128 Package :**

Bits	Description

PB0_MFP	Function
111	LCD SEG 7
011	SPI1 MOSI0
001	UART0 RX
Others	GPIOB[0]

At LQFP-64 Package :

PB0_MFP	Function
111	LCD SEG 1
011	SPI1 MOSI0
001	UART0 RX
Others	GPIOB[0]

At LQFP-48 Package :

PB0_MFP	Function
011	SPI1 MOSI0
001	UART0 RX
Others	GPIOB[0]

Multiple Function Port B High Byte Control Register (PB_H_MFP)

Register	Offset	R/W	Description	Reset Value
PB_H_MFP	GCR_BA+0x3C	R/W	Port B high byte multiple function control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	PB15_MFP			Reserved	PB14_MFP		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	PB13_MFP			Reserved	PB12_MFP		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	PB11_MFP			Reserved	PB10_MFP		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	PB9_MFP			Reserved	PB8_MFP		

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[30:28]</td><td>PB15_MFP</td><td>
PB.15 Pin Function Selection
At LQFP-128 Package :
<table>
<tr><th>PB15_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 31</td></tr>
<tr><td>100</td><td>SmartCard1 card detect</td></tr>
<tr><td>011</td><td>Snooper pin</td></tr>
<tr><td>001</td><td>External interrupt 1</td></tr>
<tr><td>Others</td><td>GPIOB[15]</td></tr>
</table>
At LQFP-64 Package :
<table>
<tr><th>PB15_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 14</td></tr>
<tr><td>100</td><td>SmartCard1 card detect</td></tr>
<tr><td>011</td><td>Snooper pin</td></tr>
<tr><td>001</td><td>External interrupt 1</td></tr>
<tr><td>Others</td><td>GPIOB[15]</td></tr>
</table>
At LQFP-48 Package :
<table>
<tr><th>PB15_MFP</th><th>Function</th></tr>
<tr><td>100</td><td>SmartCard1 card detect</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Bits</th><th>Description</th><th></th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>011</td><td>Snooper pin</td></tr>
<tr><td>001</td><td>External interrupt 1</td></tr>
<tr><td>Others</td><td>GPIOB[15]</td></tr>
</table>
</td></tr>
<tr><td>[27]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[26:24]</td><td>PB14_MFP</td><td>
PB.14 Pin Function Selection
At LQFP-128 Package :
<table>
<tr><th>PB14_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 26</td></tr>
<tr><td>100</td><td>SPI2 2nd slave select pin</td></tr>
<tr><td>011</td><td>SmartCard 2 card detect</td></tr>
<tr><td>001</td><td>External interrupt 0</td></tr>
<tr><td>Others</td><td>GPIOB[14]</td></tr>
</table>
At LQFP-64 Package :
<table>
<tr><th>PB14_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 12</td></tr>
<tr><td>100</td><td>SPI2 2nd slave select pin</td></tr>
<tr><td>011</td><td>SmartCard 2 card detect</td></tr>
<tr><td>001</td><td>External interrupt 0</td></tr>
<tr><td>Others</td><td>GPIOB[14]</td></tr>
</table>
At LQFP-64 Package :
None
</td></tr>
<tr><td>[23]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[22:20]</td><td>PB13_MFP</td><td>
PB.13 Pin Function Selection
At LQFP-128 Package :
<table>
<tr><th>PB13_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 25</td></tr>
<tr><td>010</td><td>EBI AD[1]</td></tr>
<tr><td>Others</td><td>GPIOB[13]</td></tr>
</table>
At LQFP-64 Package :
<table>
<tr><th>PB13_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 11</td></tr>
<tr><td>010</td><td>EBI AD[1]</td></tr>
</table>
</td></tr>
</table>

Bits	Description	
		Others: GPIOB[13] **At LQFP-48 Package :** **None**
[19]	**Reserved**	**Reserved**
[18:16]	**PB12_MFP**	**PB.12 Pin Function Selection** **At LQFP-128 Package**: PB12_MFP / Function: 111: LCD SEG 24 100: FRQDIV_CLK 010: EBI AD[0] Others: GPIOB[12] **At LQFP-64 Package :** PB12_MFP / Function: 111: LCD SEG 10 100: FRQDIV_CLK 010: EBI AD[0] Others: GPIOB[12] **At LQFP-48 Package :** PB12_MFP / Function: 100: FRQDIV_CLK 010: EBI AD[0] Others: GPIOB[12]
[15]	**Reserved**	**Reserved**
[14:12]	**PB11_MFP**	**PB.11 Pin Function Selection** **At LQFP-128 Package :** PB11_MFP / Function: 111: LCD V1 101: SPI0 MISO0 100: SmartCard 2 DATA 010: Timer3 external event input or Timer3 toggle output 001: PWM1 Channel 0

Bits	Description	
		Others: GPIOB[11]
		At LQFP-64 Package :
		PB11_MFP 111: LCD V1; 101: SPI0 MISO0; 100: SmartCard 2 DATA; 010: Timer3 external event input or Timer3 toggle output; 001: PWM1 Channel 0; Others: GPIOB[11]
		At LQFP-48 Package :
		PB11_MFP 111: LCD V1; 101: SPI0 MISO0; 100: SmartCard 2 DATA; 010: Timer3 external event input or Timer3 toggle output; 001: PWM1 Channel 0; Others: GPIOB[11]
[11]	**Reserved**	**Reserved**
[10:8]	**PB10_MFP**	**PB.10 Pin Function Selection**

At LQFP-128 Package :

PB10_MFP	Function
111	LCD V2
101	SPI0 MOSI0
100	SmartCard 2 clock
010	Timer2 external event input or Timer2 toggle output
001	SPI0 2nd slave select pin
Others	GPIOB[10]

At LQFP-64 Package :

PB10_MFP	Function
111	LCD V2
101	SPI0 MOSI0

Bits	Description	
		<table><tr><td>100</td><td>SmartCard 2 clock</td></tr><tr><td>010</td><td>Timer2 external event input or Timer2 toggle output</td></tr><tr><td>001</td><td>SPI0 2nd slave select pin</td></tr><tr><td>Others</td><td>GPIOB[10]</td></tr></table> **At LQFP-48 Package :** **None**
[7]	**Reserved**	**Reserved**
[6:4]	**PB9_MFP**	**PB.9 Pin Function Selection** **At LQFP-128 Package :**<table><tr><th>PB9_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD V3</td></tr><tr><td>101</td><td>External interrupt 0</td></tr><tr><td>100</td><td>SmartCard 2 RST</td></tr><tr><td>010</td><td>Timer1 external event input or Timer1 toggle output</td></tr><tr><td>001</td><td>SPI1 2nd slave select pin</td></tr><tr><td>Others</td><td>GPIOB[9]</td></tr></table>**At LQFP-64 Package :**<table><tr><th>PB9_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD V3</td></tr><tr><td>101</td><td>External interrupt 0</td></tr><tr><td>100</td><td>SmartCard 2 RST</td></tr><tr><td>010</td><td>Timer1 external event input or Timer1 toggle output</td></tr><tr><td>001</td><td>SPI1 2nd slave select pin</td></tr><tr><td>Others</td><td>GPIOB[9]</td></tr></table>**At LQFP-48 Package :** **None**
[3]	**Reserved**	**Reserved**
[2:0]	**PB8_MFP**	**PB.8 Pin Function Selection** **At LQFP-128 Package :**<table><tr><th>PB8_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 30</td></tr></table>

Bits	Description

100	SmartCard 2 power
011	External interrupt 0
010	Timer0 external event input or Timer0 toggle output
001	ADC external trigger
Others	GPIOB[8]

At LQFP-64 Package :

PB8_MFP	Function
111	LCD SEG 13
100	SmartCard 2 power
011	External interrupt 0
010	Timer0 external event input or Timer0 toggle output
001	ADC external trigger
Others	GPIOB[8]

At LQFP-48 Package :

PB8_MFP	Function
100	SmartCard 2 power
011	External interrupt 0
010	Timer0 external event input or Timer0 toggle output
001	ADC external trigger
Others	GPIOB[8]

Multiple Function Port C Low Byte Control Register (PC_MFP)

Register	Offset	R/W	Description	Reset Value
PC_L_MFP	GCR_BA+0x40	R/W	Port C low byte multiple function control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	PC7_MFP			Reserved	PC6_MFP		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	PC5_MFP			Reserved	PC4_MFP		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	PC3_MFP			Reserved	PC2_MFP		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	PC1_MFP			Reserved	PC0_MFP		

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[30:28]</td><td>PC7_MFP</td><td>PC.7 Pin Function Selection
At LQFP-128 Package :
<table>
<tr><th>PC7_MFP</th><th>Function</th></tr>
<tr><td>101</td><td>PWM0 Channel 1</td></tr>
<tr><td>011</td><td>Timer1capture event</td></tr>
<tr><td>010</td><td>EBI AD[5]</td></tr>
<tr><td>001</td><td>DA out1</td></tr>
<tr><td>Others</td><td>GPIOC[7]</td></tr>
</table>
At LQFP-64 Package :
<table>
<tr><th>PC7_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 17</td></tr>
<tr><td>101</td><td>PWM0 Channel 1</td></tr>
<tr><td>011</td><td>Timer1capture event</td></tr>
<tr><td>010</td><td>EBI AD[5]</td></tr>
<tr><td>001</td><td>DA out1</td></tr>
<tr><td>Others</td><td>GPIOC[7]</td></tr>
</table>
At LQFP-48 Package :
<table>
<tr><th>PC7_MFP</th><th>Function</th></tr>
</table>
</td></tr>
</table>

Bits	Description	
		101 — PWM0 Channel 1 011 — Timer1capture event 010 — EBI AD[5] 001 — DA out1 Others — GPIOC[7]
[27]	**Reserved**	**Reserved**
[26:24]	**PC6_MFP**	**PC.6 Pin Function Selection** **At LQFP-128 Package :** PC6_MFP — Function 101 — PWM0 Channel 0 100 — SmartCard1 card detection 011 — Timer0 capture event 010 — EBI AD[4] 001 — DA out0 Others — GPIOC[6] **At LQFP-64 Package :** PC6_MFP — Function 101 — PWM0 Channel 0 100 — SmartCard1 card detection 011 — Timer0 capture event 010 — EBI AD[4] 001 — DA out0 Others — GPIOC[6] **At LQFP-48 Package :** PC6_MFP — Function 101 — PWM0 Channel 0 100 — SmartCard1 card detection 011 — Timer0 capture event 010 — EBI AD[4] 001 — DA out0 Others — GPIOC[6]
[23]	**Reserved**	**Reserved**
[22:20]	**PC5_MFP**	**PC.5 Pin Function Selection**

Bits	Description	
		At LQFP-128 Package : | **PC5_MFP** | **Function** | | 111 | LCD COM 3 | | 001 | SPI0 MOSI1 | | Others | GPIOC[5] | **At LQFP-64 Package :** **None** **At LQFP-48 Package :** **None**
[19]	**Reserved**	**Reserved**
[18:16]	**PC4_MFP**	**PC.4 Pin Function Selection** **At LQFP-128 Package :** | **PC4_MFP** | **Function** | | 111 | LCD COM 2 | | 001 | SPI0 MISO1 | | Others | GPIOC[4] | **At LQFP-64 Package :** None **At LQFP-48 Package :** None
[15]	**Reserved**	**Reserved**
[14:12]	**PC3_MFP**	**PC.3 Pin Function Selection** **At LQFP-128 Package :** | **PC3_MFP** | **Function** | | 111 | LCD COM 1 | | 100 | SmartCard1 RST | | 010 | I^2S Dout | | 001 | SPI0 MOSI1 | | Others | GPIOC[3] | **At LQFP-64 Package :** | **PC3_MFP** | **Function** | | 111 | LCD COM 1 |

Bits	Description	
		(continued)

PC3_MFP	Function
100	SmartCard1 RST
010	I^2S Dout
001	SPI0 MOSI0
Others	GPIOC[3]

At LQFP-48 Package :

PC3_MFP	Function
100	SmartCard1 RST
010	I^2S Dout
001	SPI0 MOSI0
Others	GPIOC[3]

Bits	Description	
[11]	**Reserved**	**Reserved**
[10:8]	**PC2_MFP**	**PC.2 Pin Function Selection** (see below)

PC.2 Pin Function Selection

At LQFP-128 Package :

PC2_MFP	Function
111	LCD COM 0
100	SmartCard1 Power
010	I2S Din
001	SPI0 MISO0
Others	GPIOC[2]

At LQFP-64 Package :

PC2_MFP	Function
111	LCD COM 0
100	SmartCard1 Power
010	I^2S Din
001	SPI0 MISO0
Others	GPIOC[2]

At LQFP-48 Package :

PC2_MFP	Function
100	SmartCard1 Power
010	I^2S Din
001	SPI0 MISO0
Others	GPIOC[2]

Bits	Description	
[7]	**Reserved**	**Reserved**
[6:4]	**PC1_MFP**	**PC.1 Pin Function Selection** **At LQFP-128 Package :** PC1_MFP / Function: 111 – LCD DH2 100 – SmartCard1 DATA 010 – I2S BCLK 001 – SPI0 SCLK Others – GPIOC[1] **At LQFP-64 Package :** PC1_MFP / Function: 111 – LCD DH2 100 – SmartCard1 DATA 010 – I^{2}S BCLK 001 – SPI0 SCLK Others – GPIOC[1] **At LQFP-48 Package :** PC1_MFP / Function: 100 – SmartCard1 DATA 010 – I^{2}S BCLK 001 – SPI0 SCLK Others – GPIOC[1]
[3]	**Reserved**	**Reserved**
[2:0]	**PC0_MFP**	**PC.0 Pin Function Selection** **At LQFP-128 Package :** PC0_MFP / Function: 111 – LCD DH1 100 – SmartCard1 clock 010 – I^{2}S WS 001 – SPI0 1st slave select pin Others – GPIOC[0] **At LQFP-64 Package :**

Bits	Description

PC0_MFP	Function
111	LCD DH1
100	SmartCard1 clock
010	I^2S WS
001	SPI0 1st slave select pin
Others	GPIOC[0]

At LQFP-48 Package :

PC0_MFP	Function
100	SmartCard1 clock
010	I^2S WS
001	SPI0 1st slave select pin
Others	GPIOC[0]

Multiple Function Port C High Byte Control Register (PC_H_MFP)

Register	Offset	R/W	Description	Reset Value
PC_H_MFP	GCR_BA+0x44	R/W	Port C high byte multiple function control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	**PC15_MFP**			**Reserved**	**PC14_MFP**		
23	**22**	**19**	**20**	**19**	**18**	**17**	**16**
Reserved	**PC13_MFP**			**Reserved**	**PC12_MFP**		
15	**14**	**11**	**12**	**11**	**10**	**9**	**8**
Reserved	**PC11_MFP**			**Reserved**	**PC10_MFP**		
7	**6**	**3**	**4**	**3**	**2**	**1**	**0**
Reserved	**PC9_MFP**			**Reserved**	**PC8_MFP**		

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[30:28]</td><td>PC15_MFP</td><td>PC.15 Pin Function Selection
At LQFP-128 Package:
<table>
<tr><th>PC15_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 33</td></tr>
<tr><td>100</td><td>PWM1 Channel 2</td></tr>
<tr><td>011</td><td>Timer0 capture event</td></tr>
<tr><td>010</td><td>EBI AD[3]</td></tr>
<tr><td>Others</td><td>GPIOC[15]</td></tr>
</table>
At LQFP-64 Package:
<table>
<tr><th>PC15_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 16</td></tr>
<tr><td>100</td><td>PWM1 Channel 2</td></tr>
<tr><td>011</td><td>Timer0 capture event</td></tr>
<tr><td>010</td><td>EBI AD[3]</td></tr>
<tr><td>Others</td><td>GPIOC[15]</td></tr>
</table>
At LQFP-48 Package:
None</td></tr>
<tr><td>[27]</td><td>Reserved</td><td>Reserved</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[26:24]</td><td>PC14_MFP</td><td>PC.14 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PC14_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 32</td></tr><tr><td>100</td><td>PWM1 Channel 3</td></tr><tr><td>010</td><td>EBI AD[2]</td></tr><tr><td>Others</td><td>GPIOC[14]</td></tr></table>
At LQFP-64 Package:
<table><tr><th>PC14_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 15</td></tr><tr><td>100</td><td>PWM1 Channel 3</td></tr><tr><td>010</td><td>EBI AD[2]</td></tr><tr><td>Others</td><td>GPIOC[14]</td></tr></table>
At LQFP-48 Package:
None</td></tr>
<tr><td>[23]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[22:20]</td><td>PC13_MFP</td><td>PC.13 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PC13_MFP</th><th>Function</th></tr><tr><td>110</td><td>I²C0 SCL</td></tr><tr><td>101</td><td>External interrupt 1</td></tr><tr><td>100</td><td>Snooper pin</td></tr><tr><td>010</td><td>PWM1 Channel 1</td></tr><tr><td>001</td><td>SPI1 MOSI1</td></tr><tr><td>Others</td><td>GPIOC[13]</td></tr></table>
At LQFP-64 Package:
None
At LQFP-48 Package:
None</td></tr>
<tr><td>[19]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[18:16]</td><td>PC12_MFP</td><td>PC.12 Pin Function Selection</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>
At LQFP-128 Package:
<table>
<tr><th>PC12_MFP</th><th>Function</th></tr>
<tr><td>110</td><td>I^2C0 SDA</td></tr>
<tr><td>101</td><td>External interrupt 0</td></tr>
<tr><td>010</td><td>PWM1 Channel 0</td></tr>
<tr><td>001</td><td>SPI1 MISO1</td></tr>
<tr><td>Others</td><td>GPIOC[12]</td></tr>
</table>
At LQFP-64 Package:
None
At LQFP-48 Package:
None
</td></tr>
<tr><td>[15]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[14:12]</td><td>PC11_MFP</td><td>
PC.11 Pin Function Selection
At LQFP-128 Package:
<table>
<tr><th>PC11_MFP</th><th>Function</th></tr>
<tr><td>101</td><td>UART1 TX</td></tr>
<tr><td>001</td><td>SPI1 MOSI0</td></tr>
<tr><td>Others</td><td>GPIOC[11]</td></tr>
</table>
At LQFP-64 Package:
<table>
<tr><th>PC11_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 31</td></tr>
<tr><td>101</td><td>UART1 TX</td></tr>
<tr><td>001</td><td>SPI1 MOSI0</td></tr>
<tr><td>Others</td><td>GPIOC[11]</td></tr>
</table>
At LQFP-48 Package:
None
</td></tr>
<tr><td>[11]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[10:8]</td><td>PC10_MFP</td><td>
PC.10 Pin Function Selection
At LQFP-128 Package:
<table>
<tr><th>PC10_MFP</th><th>Function</th></tr>
<tr><td>101</td><td>UART1 RX</td></tr>
</table>
</td></tr>
</table>

Bits	Description	

001	SPI1 MISO0
Others	GPIOC[10]

At LQFP-64 Package:

PC10_MFP	Function
111	LCD SEG 30
101	UART1 RX
001	SPI1 MISO0
Others	GPIOC[10]

At LQFP-48 Package:

None

Bits	Name	Description
[7]	**Reserved**	**Reserved**
[6:4]	**PC9_MFP**	**PC.9 Pin Function Selection**

At LQFP-128 Package:

PC9_MFP	Function
101	I^2C1 SCL
001	SPI1 SCLK
Others	GPIOC[9]

At LQFP-64 Package:

PC9_MFP	Function
111	LCD SEG 29
101	I2C1 SCL
001	SPI1 SCLK
Others	GPIOC[9]

At LQFP-48 Package:

None

Bits	Name	Description
[3]	**Reserved**	**Reserved**
[2:0]	**PC8_MFP**	**PC.8 Pin Function Selection**

At LQFP-128 Package:

PC8_MFP	Function
101	I^2C1 SDA
010	EBI MCLK
001	SPI1 1st slave select pin

Bits	Description	

Others	GPIOC[8]

At LQFP-64 Package:

PC8_MFP	Function
111	LCD SEG 28
101	I^2C1 SDA
010	EBI XCLK
001	SPI1 1st slave select pin]
Others	GPIOC[8]

At LQFP-48 Package:

None

Multiple Function Port D Low Byte Control Register (PD_L_MFP)

Register	Offset	R/W	Description	Reset Value
PD_L_MFP	GCR_BA+0x48	R/W	Port D low byte multiple function control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	PD7_MFP			Reserved	PD6_MFP		
23	**22**	**19**	**20**	**19**	**18**	**17**	**16**
Reserved	PD5_MFP			Reserved	PD4_MFP		
15	**14**	**11**	**12**	**11**	**10**	**9**	**8**
Reserved	PD3_MFP			Reserved	PD2_MFP		
7	**6**	**3**	**4**	**3**	**2**	**1**	**0**
Reserved	PD1_MFP			Reserved	PD0_MFP		

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[30:28]</td><td>PD7_MFP</td><td>PD.7 Pin Function Selection
At LQFP-128 Package:
<table>
<tr><th>PD7_MDP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 2</td></tr>
<tr><td>Others</td><td>GPIOD[7]</td></tr>
</table>
At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[27]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[26:24]</td><td>PD6_MFP</td><td>PD.6 Pin Function Selection
At LQFP-128 Package:
<table>
<tr><th>PD6_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 3</td></tr>
<tr><td>Others</td><td>GPIOD[6]</td></tr>
</table>
At LQFP-64 Package:
None

At LQFP-48 Package:</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>None</td></tr>
<tr><td>[23]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[22:20]</td><td>PD5_MFP</td><td>PD.5 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PD5_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 34</td></tr><tr><td>011</td><td>SPI2 MOSI1</td></tr><tr><td>010</td><td>I^2S Dout</td></tr><tr><td>Others</td><td>GPIOD[5]</td></tr></table>
At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[19]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[18:16]</td><td>PD4_MFP</td><td>PD.4 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PD4_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 35</td></tr><tr><td>100</td><td>SmartCard1 card detection</td></tr><tr><td>011</td><td>SPI2 MISO1</td></tr><tr><td>010</td><td>I^2S Din</td></tr><tr><td>Others</td><td>GPIOD[4]</td></tr></table>At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[15]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[14:12]</td><td>PD3_MFP</td><td>PD.3 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PD3_MFP</th><th>Function</th></tr><tr><td>101</td><td>ADC input channel11</td></tr><tr><td>100</td><td>SmartCard1 reset</td></tr><tr><td>011</td><td>SPI2 MOSI0</td></tr><tr><td>010</td><td>I^2S BCLK</td></tr></table></td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td>001</td><td>UART1 CTSn</td></tr>
<tr><td>Others</td><td>GPIOD[3]</td></tr>
</table>
At LQFP-64 Package:
None

At LQFP-48 Package:
None
</td></tr>
<tr><td>[11]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[10:8]</td><td>PD2_MFP</td><td>
PD.2 Pin Function Selection
At LQFP-128 Package:
<table>
<tr><th>PD2_MFP</th><th>Function</th></tr>
<tr><td>101</td><td>ADC input channel10</td></tr>
<tr><td>100</td><td>SmartCard1 power</td></tr>
<tr><td>011</td><td>SPI2 MISO0</td></tr>
<tr><td>010</td><td>I^{2}S WS</td></tr>
<tr><td>001</td><td>UART1 RTSn</td></tr>
<tr><td>Others</td><td>GPIOD[2]</td></tr>
</table>
At LQFP-64 Package:
None

At LQFP-48 Package:
None
</td></tr>
<tr><td>[7]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[6:4]</td><td>PD1_MFP</td><td>
PD.1 Pin Function Selection
At LQFP-128 Package:
<table>
<tr><th>PD1_MFP</th><th>Function</th></tr>
<tr><td>101</td><td>ADC input channel9</td></tr>
<tr><td>100</td><td>SmartCard1 DATA</td></tr>
<tr><td>011</td><td>SPI2 SCLK</td></tr>
<tr><td>001</td><td>UART1 TX</td></tr>
<tr><td>Others</td><td>GPIOD[1]</td></tr>
</table>
At LQFP-64 Package:
None

At LQFP-48 Package:
None
</td></tr>
<tr><td>[3]</td><td>Reserved</td><td>Reserved</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[2:0]</td><td>PD0_MFP</td><td>PD.0 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PD0_MFP</th><th>Function</th></tr><tr><td>101</td><td>ADC input channel8</td></tr><tr><td>100</td><td>SmartCard1 clock</td></tr><tr><td>011</td><td>SPI2 1st slave select pin</td></tr><tr><td>001</td><td>UART1 RX</td></tr><tr><td>Others</td><td>GPIOD[0]</td></tr></table>At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
</table>

Multiple Function Port D High Byte Control Register (PD_H_MFP)

Register	Offset	R/W	Description	Reset Value
PD_H_MFP	GCR_BA+0x4C	R/W	Port D high byte multiple function control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	PD15_MFP			Reserved	PD14_MFP		
23	**22**	**19**	**20**	**19**	**18**	**17**	**16**
Reserved	PD13_MFP			Reserved	PD12_MFP		
15	**14**	**11**	**12**	**11**	**10**	**9**	**8**
Reserved	PD11_MFP			Reserved	PD10_MFP		
7	**6**	**3**	**4**	**3**	**2**	**1**	**0**
Reserved	PD9_MFP			Reserved	PD8_MFP		

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[30:28]</td><td>PD15_MFP</td><td>PD.15 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PD15_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 0</td></tr><tr><td>Others</td><td>GPIOD[15]</td></tr></table>At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[27]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[26:24]</td><td>PD14_MFP</td><td>PD.14 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PD14_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 1</td></tr><tr><td>Others</td><td>GPIOD[14]</td></tr></table>At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[23]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[22:20]</td><td>PD13_MFP</td><td>PD.13 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PD13_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 14</td></tr><tr><td>Others</td><td>GPIOD[13]</td></tr></table>At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[19]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[18:16]</td><td>PD12_MFP</td><td>PD.12 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PD12_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 15</td></tr><tr><td>Others</td><td>GPIOD[12]</td></tr></table>At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[15]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[14:12]</td><td>PD11_MFP</td><td>PD.11 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PD11_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 16</td></tr><tr><td>Others</td><td>GPIOD[11]</td></tr></table>At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[11]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[10:8]</td><td>PD10_MFP</td><td>PD.10 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PD10_MFP</th><th>Function</th></tr></table></td></tr>
</table>

Bits	Description	
		<table><tr><td>111</td><td>LCD SEG 17</td></tr><tr><td>Others</td><td>GPIOD[10]</td></tr></table> **At LQFP-64 Package:** **None** **At LQFP-48 Package:** **None**
[7]	**Reserved**	**Reserved**
[6:4]	**PD9_MFP**	**PD.9 Pin Function Selection** **At LQFP-128 Package:** <table><tr><th>PD9_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 18</td></tr><tr><td>Others</td><td>GPIOD[9]</td></tr></table> **At LQFP-64 Package:** **None** **At LQFP-48 Package:** **None**
[3]	**Reserved**	**Reserved**
[2:0]	**PD8_MFP**	**PD.8 Pin Function Selection** **At LQFP-128 Package:** <table><tr><th>PD8_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 19</td></tr><tr><td>Others</td><td>GPIOD[8]</td></tr></table> **At LQFP-64 Package:** **None** **At LQFP-48 Package:** **None**

Multiple Function Port E Low Byte Control Register (PE_L_MFP)

Register	Offset	R/W	Description	Reset Value
PE_L_MFP	GCR_BA+0x50	R/W	Port E low byte multiple function control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	**PE7_MFP**			**Reserved**	**PE6_MFP**		
23	**22**	**19**	**20**	**19**	**18**	**17**	**16**
Reserved	**PE5_MFP**			**Reserved**	**PE4_MFP**		
15	**14**	**11**	**12**	**11**	**10**	**9**	**8**
Reserved	**PE3_MFP**			**Reserved**	**PE2_MFP**		
7	**6**	**3**	**4**	**3**	**2**	**1**	**0**
Reserved	**PE1_MFP**			**Reserved**	**PE0_MFP**		

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[30:28]</td><td>PE7_MFP</td><td>PE.7 Pin Function Selection
At LQFP-128 Package:
<table>
<tr><th>PE7_MFP</th><th>Function</th></tr>
<tr><td>111</td><td>LCD SEG 8</td></tr>
<tr><td>Others</td><td>GPIOE[7]</td></tr>
</table>
At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[27]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[26:24]</td><td>PE6_MFP</td><td>PE.6 Pin Function Selection
At LQFP-128 Package:
GPIOE[6]
At LQFP-64 Package:
GPIOE[6]
At LQFP-48 Package:
GPIOE[6]</td></tr>
<tr><td>[23]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[22:20]</td><td>PE5_MFP</td><td>PE.5 Pin Function Selection
At LQFP-128 Package:</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><th>PE5_MFP</th><th>Function</th></tr>
<tr><td>001</td><td>PWM1 Channel 1</td></tr>
<tr><td>Others</td><td>GPIOE[5]</td></tr>
</table>

At LQFP-64 Package:
NANO100/NANO120 series:
<table>
<tr><th>PE5_MFP</th><th>Function</th></tr>
<tr><td>001</td><td>PWM1 Channel 1</td></tr>
<tr><td>Others</td><td>GPIOE[5]</td></tr>
</table>
NANO110/130 series:
None

At LQFP-48 Package:
NANO100 series:
<table>
<tr><th>PE5_MFP</th><th>Function</th></tr>
<tr><td>001</td><td>PWM1 Channel 1</td></tr>
<tr><td>Others</td><td>GPIOE[5]</td></tr>
</table>
NANO110/NANO120/NANO130 series:
None
</td></tr>
<tr><td>[19]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[18:16]</td><td>PE4_MFP</td><td>
PE.4 Pin Function Selection
At LQFP-128 Package:
<table>
<tr><th>PE4_MFP</th><th>Function</th></tr>
<tr><td>110</td><td>SPI0 MOSI0</td></tr>
<tr><td>Others</td><td>GPIOE[4]</td></tr>
</table>
At LQFP-64 Package:
None

At LQFP-48 Package:
None
</td></tr>
<tr><td>[15]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[14:12]</td><td>PE3_MFP</td><td>
PE.3 Pin Function Selection
At LQFP-128 Package:
</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>
<table><tr><th>PE3_MFP</th><th>Function</th></tr><tr><td>110</td><td>SPI0 MISO0</td></tr><tr><td>Others</td><td>GPIOE[3]</td></tr></table>
At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[11]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[10:8]</td><td>PE2_MFP</td><td>PE.2 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PE2_MFP</th><th>Function</th></tr><tr><td>110</td><td>SPI0 SCLK</td></tr><tr><td>Others</td><td>GPIOE[2]</td></tr></table>
At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[7]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[6:4]</td><td>PE1_MFP</td><td>PE.1 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PE1_MFP</th><th>Function</th></tr><tr><td>110</td><td>SPI0 1st slave select pin</td></tr><tr><td>001</td><td>PWM1 Channel 3</td></tr><tr><td>Others</td><td>GPIOE[1]</td></tr></table>
At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[3]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[2:0]</td><td>PE0_MFP</td><td>PE.0 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PE0_MFP</th><th>Function</th></tr><tr><td>010</td><td>I^{2}S MCLK</td></tr></table></td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="3">Description</th></tr>
<tr><td rowspan="3"></td><td rowspan="3"></td><td>001</td><td>PWM1 Channel 2</td></tr>
<tr><td>Others</td><td>GPIOE[0]</td></tr>
<tr><td colspan="2">At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
</table>

Multiple Function Port E High Byte Control Register (PE_H_MFP)

Register	Offset	R/W	Description	Reset Value
PE_H_MFP	GCR_BA+0x54	R/W	Port E high byte multiple function control register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved	PE15_MFP			Reserved	PE14_MFP		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	PE13_MFP			Reserved	PE12_MFP		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	PE11_MFP			Reserved	PE10_MFP		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	PE9_MFP			Reserved	PE8_MFP		

Bits	Description	
[31]	**Reserved**	**Reserved**
[30:28]	**PE15_MFP**	**PE.15 Pin Function Selection** **At LQFP-128 Package:** <table><tr><th>PE15_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 2</td></tr><tr><td>Others</td><td>GPIOE[15]</td></tr></table>**At LQFP-64 Package:** **None** **At LQFP-48 Package:** **None**
[27]	**Reserved**	**Reserved**
[26:24]	**PE14_MFP**	**PE.14 Pin Function Selection** **At LQFP-128 Package:** <table><tr><th>PE14_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 28</td></tr><tr><td>Others</td><td>GPIOE[14]</td></tr></table>**At LQFP-64 Package:** **None** **At LQFP-48 Package:** **None**

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[23]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[22:20]</td><td>PE13_MFP</td><td>PE.13 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PE13_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 27</td></tr><tr><td>Others</td><td>GPIOE[13]</td></tr></table>At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[19]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[18:16]</td><td>PE12_MFP</td><td>PE.12 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PE12_MFP</th><th>Function</th></tr><tr><td>111</td><td>UART1 CTSn (only valid in NANO110)</td></tr><tr><td>Others</td><td>GPIOE[12]</td></tr></table>At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[15]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[14:12]</td><td>PE11_MFP</td><td>PE.11 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PE11_MFP</th><th>Function</th></tr><tr><td>111</td><td>UART1 RTSn (only valid in NANO110)</td></tr><tr><td>Others</td><td>GPIOE[11]</td></tr></table>At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[11]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[10:8]</td><td>PE10_MFP</td><td>PE.10 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PE10_MFP</th><th>Function</th></tr></table></td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="3">Description</th></tr>
<tr><td></td><td></td><td colspan="2"><table><tr><td>111</td><td>UART1 TX (only valid in NANO110)</td></tr><tr><td>Others</td><td>GPIOE[10]</td></tr></table>At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[7]</td><td>Reserved</td><td colspan="2">Reserved</td></tr>
<tr><td>[6:4]</td><td>PE9_MFP</td><td colspan="2">PE.9 Pin Function Selection
At LQFP-128 Package:<table><tr><th>PE9_MFP</th><th>Function</th></tr><tr><td>111</td><td>UART1 RX (only valid in NANO110)</td></tr><tr><td>Others</td><td>GPIOE[9</td></tr></table>At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[3]</td><td>Reserved</td><td colspan="2">Reserved</td></tr>
<tr><td>[2:0]</td><td>PE8_MFP</td><td colspan="2">PE.8 Pin Function Selection
At LQFP-128 Package:<table><tr><th>PE8_MFP</th><th>Function</th></tr><tr><td>111</td><td>LCD SEG 9</td></tr><tr><td>Others</td><td>GPIOE[8]</td></tr></table>At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
</table>

Multiple Function Port F Low Byte Control Register (PF_L_MFP)

Register	Offset	R/W	Description	Reset Value
PF_L_MFP	GCR_BA+0x58	R/W	Port F low byte multiple function control register	0x0077_7777

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	PF5_MFP			Reserved	PF4_MFP		
15	14	13	12	11	10	9	8
Reserved	PF3_MFP			Reserved	PF2_MFP		
7	6	5	4	3	2	1	0
Reserved	PF1_MFP			Reserved	PF0_MFP		

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:23]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[22:20]</td><td>PF5_MFP</td><td>PF.5 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PF5_MFP</th><th>Function</th></tr><tr><td>001</td><td>I2C0 SCL</td></tr><tr><td>Others</td><td>GPIOF[5]</td></tr></table>At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
<tr><td>[19]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[18:16]</td><td>PF4_MFP</td><td>PF.4 Pin Function Selection
At LQFP-128 Package:
<table><tr><th>PF4_MFP</th><th>Function</th></tr><tr><td>001</td><td>I2C0 SDA</td></tr><tr><td>Others</td><td>GPIOF[4]</td></tr></table>At LQFP-64 Package:
None

At LQFP-48 Package:
None</td></tr>
</table>

Bits	Description	
[15]	**Reserved**	**Reserved**
[14:12]	**PF3_MFP**	**PF.3 Pin Function Selection** **At LQFP-128 Package:** PF3_MFP: 111 = HXT IN; Others = GPIOF[3] **At LQFP-64 Package:** PF3_MFP: 111 = HXT IN; Others = GPIOF[3] **At LQFP-48 Package:** PF3_MFP: 111 = HXT IN; Others = GPIOF[3]
[11]	**Reserved**	**Reserved**
[10:8]	**PF2_MFP**	**PF.2 Pin Function Selection** **At LQFP-128 Package:** PF2_MFP: 111 = HXT OUT; Others = GPIOF[2] **At LQFP-64 Package:** PF2_MFP: 111 = HXT OUT; Others = GPIOF[2] **At LQFP-48 Package:** PF2_MFP: 111 = HXT OUT; Others = GPIOF[2]
[7]	**Reserved**	**Reserved**
[6:4]	**PF1_MFP**	**PF.1 Pin Function Selection**

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>

At LQFP-128 Package:

PF1_MFP	Function
111	ICE CLOCK
101	External interrupt 1
100	FRQDIV_CLK
Others	GPIOF[1]

At LQFP-64 Package:

PF1_MFP	Function
111	ICE CLOCK
101	External interrupt 1
100	FRQDIV_CLK
Others	GPIOF[1]

At LQFP-48 Package:

PF1_MFP	Function
111	ICE clock
101	External interrupt 1
100	FRQDIV_CLK
Others	GPIOF[1]

</td></tr>
<tr><td>[3]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[2:0]</td><td>PF0_MFP</td><td>

PF.0 Pin Function Selection

At LQFP-128 Package:

PF0_MFP	Function
111	ICE DATA
101	External interrupt 0
Others	GPIOF[1]

At LQFP-64 Package:

PF0_MFP	Function
111	ICE DATA
101	External interrupt 0
Others	GPIOF[1]

At LQFP-48 Package:

</td></tr>
</table>

Bits	Description		
		PF0_MFP	**Function**
		111	ICE DATA
		101	External interrupt 0
		Others	GPIOF[1]

Power-On-reset Control Register (PORCTL)

Register	Offset	R/W	Description	Reset Value
PORCTL	GCR_BA+0x60	R/W	Power-On-Reset Controller Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
POR_DIS_CODE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
POR_DIS_CODE							

Bits	Description	
[31:16]	**Reserved**	**Reserved**
[15:0]	**POR_DIS_CODE**	**Power-on Reset Enable Control** This is a protected register. Please refer to open lock sequence to program it. When powered on, the POR circuit generates a reset signal to reset the whole chip function, but noise on the power may cause the POR active again. If setting the POR_DIS_CODE to 0x5AA5, the POR reset function will be disabled and the POR function will be active again when POR_DIS_CODE is set to another value or POR_DIS_CODE is reset by chip other reset functions, including: /RESET, Watchdog Timer reset, BOD reset, ICE reset command and the software-chip reset function

Brown-out Detect Control Register (BODCTL)

Partial of the BODCTL control registers bits are initiated by the flash configuration

Register	Offset	R/W	Description	Reset Value
BODCTL	GCR_BA+0x64	R/W	Brown-out Detector Controller Register	0x00FF_F0xx

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		**BOD25_INT_EN**		**BOD20_INT_EN**		**BOD17_INT_EN**	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**BOD25_RST_EN**	**BOD20_RST_EN**	**BOD17_RST_EN**	**Reserved**	**BOD25_EN**	**BOD20_EN**	**BOD17_EN**

Bits	Description	
[31:11]	**Reserved**	**Reserved**
[10]	**BOD25_INT_EN**	**BOD 2.5 V interrupt Enable** This is a protected register. Please refer to open lock sequence to program it. 1 = Interrupt issues when BOD25 occurs 0 = Interrupt does not issue when BOD25 occurs
[9]	**BOD20_INT_EN**	**BOD 2.0 V interrupt Enable** This is a protected register. Please refer to open lock sequence to program it. 1 = Interrupt issues when BOD20 occurs 0 = Interrupt does not issue when BOD20 occurs
[8]	**BOD17_INT_EN**	**BOD 1.7 V interrupt Enable** This is a protected register. Please refer to open lock sequence to program it. 1 = Interrupt issues when BOD17 occurs 0 = Interrupt does not issue when BOD17 occurs
[7]	**Reserved**	**Reserved**
[6]	**BOD25_RST_EN**	**BOD 2.5 V Reset Enable** This is a protected register. Please refer to open lock sequence to program it. 1 = Reset issues when BOD25 occurs 0 = Reset does not issue when BOD25 occurs The default value is set by flash controller user configuration register config0 bit[20:19]
[5]	**BOD20_RST_EN**	**BOD 2.0 V Reset Enable** This is a protected register. Please refer to open lock sequence to program it.

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>1 = Reset issues when BOD20 occurs
0 = Reset does not issue when BOD20 occurs
The default value is set by flash controller user configuration register config0 bit[20:19]</td></tr>
<tr><td>[4]</td><td>BOD17_RST_EN</td><td>BOD 1.7 V Reset Enable
This is a protected register. Please refer to open lock sequence to program it.
1 = Reset issues when BOD17 occurs
0 = Reset does not issue when BOD17 occurs
The default value is set by flash controller user configuration register config0 bit[20:19]
BOD17_RST_EN can be controlled (enable or disable) only when BOD17_EN is high.
<table><tr><th>BOD17_EN Status</th><th>BOD17 RST Function</th></tr><tr><td>Low</td><td>Enabled</td></tr><tr><td>High</td><td>Controledl by BOD17_RST_EN</td></tr></table></td></tr>
<tr><td>[3]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[2]</td><td>BOD25_EN</td><td>Brown-out Detector 2.5 V Function Enable
This is a protected register. Please refer to open lock sequence to program it.
1 = Brown-out Detector 2.5 V function Enabled
0 = Brown-out Detector 2.5 V function Disabled</td></tr>
<tr><td>[1]</td><td>BOD20_EN</td><td>Brown-out Detector 2.0 V Function Enable
This is a protected register. Please refer to open lock sequence to program it.
1 = Brown-out Detector 2.0 V function Enabled
0 = Brown-out Detector 2.0 V function Disabled
BOD20_EN is default on. If SW disables it, Brown-out Detector 2.0 V function is not disabled until chip enters power-down mode. If system is not in power-down mode, BOD20_EN will be enabled by hardware automatically.</td></tr>
<tr><td>[0]</td><td>BOD17_EN</td><td>Brown-out Detector 1.7V Function Enable
This is a protected register. Please refer to open lock sequence to program it.
The default value is set by flash controller user configuration register config0 bit[20:19]
Users can disable BOD17_EN but it takes effective (disabled) only in Power-down mode. Once existing Power-down mode, BOD17 will be enabled by HW automatically.
When CPU reads this bit, CPU will read whether BOD17 function enabled or not. In other words,CPU will always read high.
1 = Brown-out Detector 1.7V function Enabled
0 = Brown-out Detector 1.7V function Disabled
<table><tr><th>Operating Mode</th><th>BOD17 Function</th></tr><tr><td>Normal mode</td><td>Enabled</td></tr><tr><td>Power-down mode</td><td>Controlled by BOD17_EN</td></tr></table></td></tr>
</table>

Brown-out Detector Status Register (BODSTS)

Register	Offset	R/W	Description	Reset Value
BODSTS	GCR_BA+0x68	R/W	Brown-out Detector Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**BOD25_rise**	**BOD20_rise**	**BOD17_rise**	**BOD25_drop**	**BOD20_drop**	**BOD17_drop**	**BOD_INT**

Bits	Description	
[31:7]	**Reserved**	**Reserved**
[6]	**BOD25_rise**	**Brown-out Detector higher than 2.5V Status** Setting BOD25_rise high means once the detected voltage is higher than target detected voltage setting (2.5V). Software can write 1 to clear BOD25_rise.
[5]	**BOD20_rise**	**Brown-out Detector higher than 2.0V Status** Setting BOD20_rise high means once the detected voltage is higher than target detected voltage setting (2.0V). Software can write 1 to clear BOD20_rise
[4]	**BOD17_rise**	**Brown-out Detector higher than 1.7V Status** Setting BOD17_rise high means once the detected voltage is higher than target detected voltage setting (1.7V). Software can write 1 to clear BOD17_rise
[3]	**BOD25_drop**	**Brown-out Detector lower than 2.5V Status** Setting BOD25_drop high means once the detected voltage is lower than target detected voltage setting (2.5V). Software can write 1 to clear BOD25_drop
[2]	**BOD20_drop**	**Brown-out Detector lower than 2.0V Status** Setting BOD20_drop high means once the detected voltage is lower than target detected voltage setting (2.0V). Software can write 1 to clear BOD20_drop
[1]	**BOD17_drop**	**Brown-out Detector lower than 1.7V Status** Setting BOD17_drop high means once the detected voltage is lower than target detected voltage setting (1.7V). Software can write 1 to clear BOD17_drop
[0]	**BOD_INT**	**Brown-out Detector interrupt status** 1 = When Brown-out Detector detects the V_{DD} is dropped down through the target detected voltage or the V_{DD} is raised up through the target detected voltage and Brown-out interrupt is enabled, this bit will be set to 1. 0 = Brown-out Detector does not detect any voltage drift at V_{DD} down through or up through the target detected voltage after interrupt is enabled. This bit is cleared by writing 1 to itself.

Internal Voltage Reference Generator Control Register (Int_VREFCTL)

Register	Offset	R/W	Description	Reset Value
Int_VREFCTL	GCR_BA+0x6C	R/W	Voltage reference Control register	0x0000_0F00

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**EXT_MODE**	**SEL25**	**REG_EN**	**BGP_EN**

Bits	Description	
[31:4]	**Reserved**	**Reserved**
[3]	**EXT_MODE**	**Regulator External Mode** This is a protected register. Please refer to open lock sequence to program it. Users can output regulator output voltage in V_{REF} pin if EXT_MODE is high. 0 = No connection with external V_{REF} pin. 1 = Connet to external V_{REF} pin. Connect a 1 uF to 10 uF capacitor to AV_{SS} will let internal voltage reference be more stable.
[2]	**SEL25**	**Regulator Output Voltage Selection** Select internal reference voltage level. This is a protected register. Please refer to open lock sequence to program it. 0 = 1.8V 1 = 2.5V
[1]	**REG_EN**	**Regulator Enable** Enable internal 1.8V or 2.5V reference voltage. This is a protected register. Please refer to open lock sequence to program it. 0 = Disabled 1 = Enabled
[0]	**BGP_EN**	**Band-gap Enable** This is a protected register. Please refer to open lock sequence to program it. Band-gap is the reference voltage of internal reference voltage. User must enable band-gap if want to enable internal 1.8V or 2.5V reference voltage. 0 = Disabled 1 = Enabled

HIRC Trim Control Register (IRCTRIMCTL)

Register	Offset	R/W	Description	Reset Value
IRCTRIMCTL	GCR_BA+0x80	R/W	HIRC Trim Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							ERR_STOP
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRIM_RETRY_CNT		TRIM_LOOP		Reserved		TRIM_SEL	

Bits	Description	
[31:8]	**Reserved**	**Reserved**
[8]	**ERR_STOP**	**Trim Stop When 32.768 kHz Error Detected** This bit is used to control if stop the HIRC trim operation when 32.768 kHz clock error is detected. If set this bit high and 32.768 kHz clock error detected, the status 32K_ERR_INT would be set high and HIRC trim operation was stopped. If this bit is low and 32.768 kHz clock error detected, the status 23K_ERR_INT would be set high and HIRC trim operation is continuously. 0 = Continue the HIRC trim operation even if 32.768 kHz clock error detected. 1 = Stop the HIRC trim operation if 32.768 kHz clock error detected.
[7:6]	**TRIM_RETRY_CNT**	**Trim Value Update Limitation Count** This field defines that how many times the auto trim circuit will try to update the HIRC trim value before the frequency of HIRC locked. Once the HIRC locked, the internal trim value update counter will be reset. If the trim value update counter reached this limitation value and frequency of HIRC still doesn't lock, the auto trim operation will be disabled and TRIM_SEL will be cleared to 00. **TRIM_RETRY_CNT** / **Trim Retry Count Limitation**: 00 = Trim retry count limitation is 64 01 = Trim retry count limitation is 128 10 = Trim retry count limitation is 256 11 = Trim retry count limitation is 512
[5:4]	**TRIM_LOOP**	**Trim Calculation Loop** This field defines that trim value calculation is based on how many 32.768 kHz clock. For example, if TRIM_LOOP is set as 00, auto trim circuit will calculate trim value based on the average frequency difference in 4 32.768 kHz clock.

NANO100 SERIES TECHNICAL REFERENCE MANUAL

Bits	Description	

TRIM_LOOP	Average Frequency Difference
00	4 32.768 kHz clock
01	8 32.768 kHz clock
10	16 32.768 kHz clock
11	32 32.768 kHz clock

Bits	Name	Description
[3:2]	**Reserved**	**Reserved**
[1:0]	**TRIM_SEL**	**Trim Frequency Selection** This field indicates the target frequency of HIRC auto trim. If no any target frequency is selected (TRIM_SEL is 00), the HIRC auto trim function is disabled. During auto trim operation, if 32.768 kHz clock error detected or trim retry limitation count reached, this field will be cleared to 00 automatically.

TRIM_SEL	Function
00	Disable HIRC auto trim function
01	Enable HIRC auto trim function and trim HIRC to 11.0592 MHz
10	Enable HIRC auto trim function and trim HIRC to 12 MHz
11	Enable HIRC auto trim function and trim HIRC to 12.288 MHz

HIRC Trim Interrupt Enable Register (IRCTRIMIEN)

Register	Offset	R/W	Description	Reset Value
IRCTRIMIEN	GCR_BA+0x84	R/W	HIRC Trim Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					32K_ERR_IEN	TRIM_FAIL_IEN	Reserved

Bits	Description	
[31:3]	**Reserved**	**Reserved**
[2]	**32K_ERR_IEN**	**32.768 kHz Clock Error Interrupt Enable** This bit controls if CPU would get an interrupt while 32.768 kHz clock is inaccuracy during auto trim operation. If this bit is high, and 32K_ERR_INT is set during auto trim operation, an interrupt will be triggered to notify the 32.768 kHz clock frequency is inaccuracy. 0 = 32K_ERR_INT status Disabled to trigger an interrupt to CPU. 1 = 32K_ERR_INT status Enabled to trigger an interrupt to CPU.
[1]	**TRIM_FAIL_IEN**	**Trim Failure Interrupt Enable** This bit controls if an interrupt will be triggered while HIRC trim value update limitation count reached and HIRC frequency still not locked on target frequency set by TRIM_SEL. If this bit is high and TRIM_FAIL_INT is set during auto trim operation, an interrupt will be triggered to notify that HIRC trim value update limitation count was reached. 0 = TRIM_FAIL_INT status Disabled to trigger an interrupt to CPU. 1 = TRIM_FAIL_INT status Enabled to trigger an interrupt to CPU.
[0]	**Reserved**	**Reserved**

HIRC Trim Interrupt Status Register (IRCTRIMINT)

Register	Offset	R/W	Description	Reset Value
IRCTRIMINT	GCR_BA+0x88	R/W	HIRC Trim Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**32K_ERR_INT**	**TRIM_FAIL_INT**	**FREQ_LOCK**

Bits	Description	
[31:3]	**Reserved**	**Reserved.**
[2]	**32K_ERR_INT**	**32.768 kHz Clock Error Interrupt Status** This bit indicates that 32.768 kHz clock frequency is inaccuracy. Once this bit is set, the auto trim operation stopped and TRIM_SEL will be cleared to 00 by hardware automatically. If this bit is set and 32K_ERR_IEN is high, an interrupt will be triggered to notify the 32.768 kHz clock frequency is inaccuracy. Write 1 to clear this to zero. 0 = 32.768 kHz clock frequency is accuracy. 1 = 32.768 kHz clock frequency is inaccuracy.
[1]	**TRIM_FAIL_INT**	**Trim Failure Interrupt Status** This bit indicates that HIRC trim value update limitation count reached and HIRC clock frequency still doesn't lock. Once this bit is set, the auto trim operation stopped and TRIM_SEL will be cleared to 00 by hardware automatically. If this bit is set and TRIM_FAIL_IEN is high, an interrupt will be triggered to notify that HIRC trim value update limitation count was reached. Write 1 to clear this to zero. 0 = Trim value update limitation count doesn't reach. 1 = Trim value update limitation count reached and HIRC frequency still doesn't lock.
[0]	**FREQ_LOCK**	**HIRC Frequency Lock Status** This bit indicates the HIRC frequency lock. This is a status bit and doesn't trigger any interrupt.

Register Lock Key Address Register (RegLockAddr)

Some of the system control registers need to be protected to avoid inadvertent write and disturb the chip operation. These system control registers are locked after the power-on reset till users to open the lock. For users to program these protected registers, an open lock sequence needs to be followed by a special programming sequence. The open sequence is to continue write the data “59h”, “16h” “88h” to the key controller address 0x5000_0100. Any different data value or different sequence or any other write operations to any other address during these three data program aborts the whole sequence.

After the lock is opened, users can check the lock bit RegLockAddr[0]. “1” is unlocked, “0” is locked. Then users can update the target register value if RegUnLock is high and write any data to the address “0x5000_0100” to re-lock the protected registers

This register is written for open the RegUnLock key and read for the RegUnLock status

Register	Offset	R/W	Description	Reset Value
RegLockAddr	GCR_BA+0x100	R/W	Register Lock Key address	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							RegUnLock

Bits	Description	
[31:1]	**Reserved**	**Reserved**
[0]	**RegUnLock**	1 = Protected registers are Unlocked. 0 = Protected register are Locked. Any write to the target register is ignored.

5.5 Clock Controller

5.5.1 Overview

The clock controller generates clocks for the whole chip, Iincluding system clocks (CPU clock, HCLKx, and PCLKx) and all peripheral engine clocks. HCLKx means AHB bus clock for peripherals on AHB bus. PCLKx means APB bus clock for peripherals on APB bus. The clock controller also implements the power control function with the individually clock ON/OFF control, clock source selection and a 4-bit clock divider. The chip will not enter power-down mode until CPU sets the power down enable bit (PD_EN) and CPU executes the WFI instruction. In the Power-down mode, clock controller turns off the external high frequency crystal, internal high frequency oscillator, and system clocks (CPU clock, HCLKx, and PCLKx) to reduce the power consumption to minimum.

5.5.2 Features

- Generates clocks for system clocks and all peripheral engine clocks.
- Each peripheral engine clock can be turned on/off.
- High frequency crystal, internal high frequency oscillator, and system clocks will be turned off when chip is in Power-down mode.

5.5.3 Block Diagram

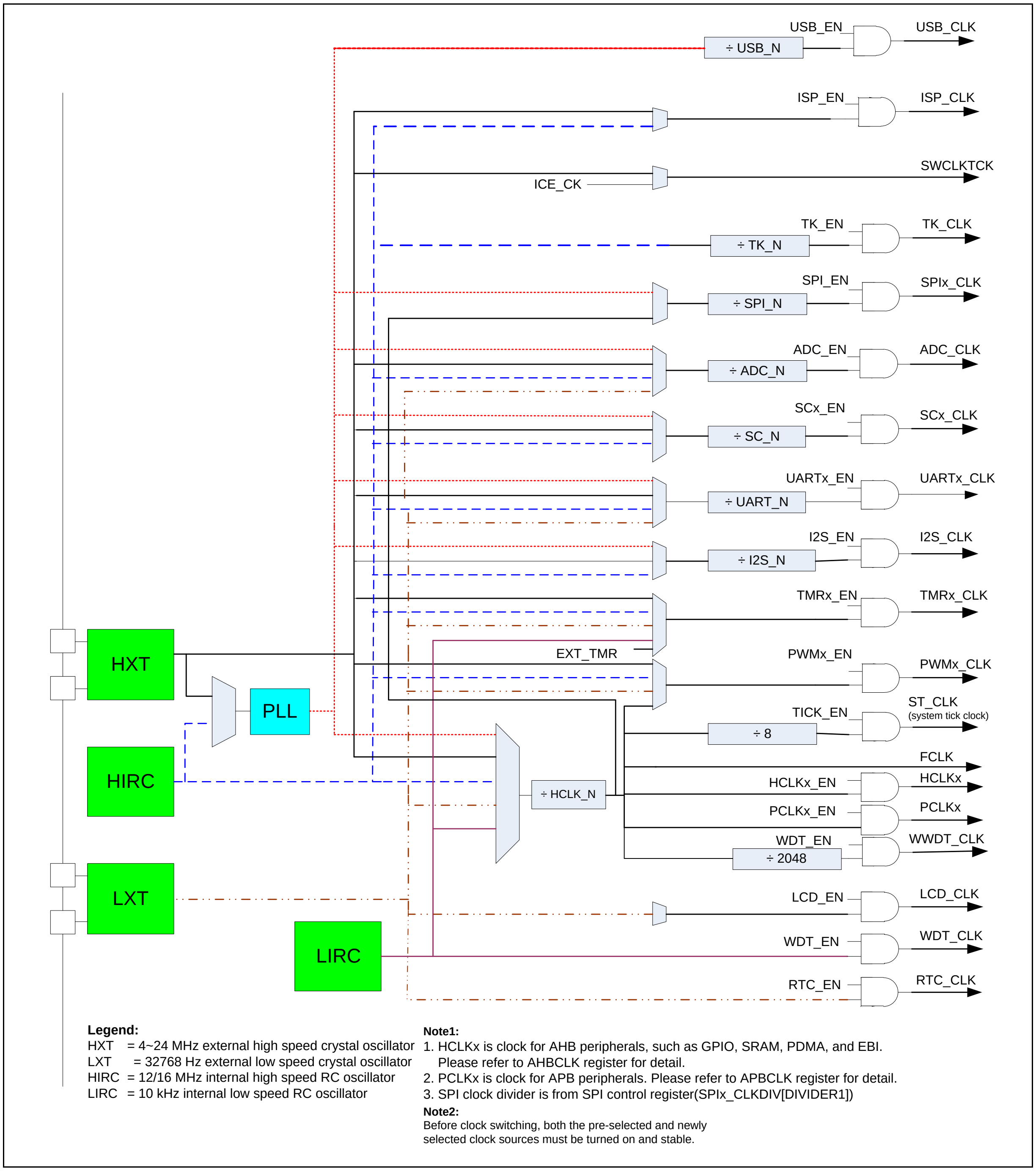


Figure 5.5-1 Clock Controller Block Diagram

5.5.4 Functional Description

5.5.4.1 System Clock

The clock controller consists of 5 sources as listed below:

- 32.768 kHz low speed external crystal (LXT)
- 4~ 24 MHz high speed external crystal (HXT)
- One programmable PLL FOUT (PLL source consists of HXT and HIRC)
- 12 MHz high speed internal RC oscillator (HIRC)
- 10 kHz low speed internal RC oscillator (LIRC)

5.5.4.2 Peripheral engine clocks

Each peripheral engine clock has different Clock Source switching setting. Please refer to the CLKSEL1 and CLKSEL2 description.

	HXT	LXT	HIRC	LIRC	PCLK/HCLK	PLL	Ext. Pin
LCD	-	Yes	-	-	-	-	-
TM	Yes	Yes	Yes	Yes	-	-	Yes
PWM	Yes	Yes	Yes	-	-	Yes	-
ADC	Yes	Yes	Yes	-	-	Yes	-
UART	Yes	Yes	Yes	-	-	Yes	-
SC	Yes	-	Yes	-	-	Yes	-
I^2S	Yes	-	Yes	-	-	Yes	-
SPI	-	-	Yes	-	-	Yes	-
I^2C	-	-	-	-	Yes	-	-
USB	-	-	-	-	-	Yes	-

Table 5.5-1 Peripheral Engine Clocks

5.5.4.3 Clocks in Power-down Mode

When chip enters Power-down mode, system clocks (CPU clock, HCLKx, and PCLKx), HXT, HIRC will be disabled directly. LXT and LIRC could be still active in Power-down mode if CPU does not disable these clocks before entering Power-down mode. IP engine clock could be still active in Power-down mode if IP adopts LXT or LIRC and LXT or LIRC does not be disabled respectively.

5.5.4.4 Frequency Divider Output

This device is equipped a power-of-2 frequency divider which is composed by16 chained divide-by-2 shift registers. One of the 16 shift register outputs selected by a sixteen to one multiplexer is reflected to GPIOB.12/GPIOF.1. Therefore there are 16 options of power-of-2 divided clocks with the frequency from Fin/2^1 to Fin/2^16 where Fin is input clock frequency to the clock divider.

The output formula is Fout = Fin/2^(N+1), where Fin is the input clock frequency, Fout is the clock divider output frequency and N is the 4-bit value in FSEL (FRQDIV[3:0]).

When FDIV_EN (FRQDIV[4]) is set to high, the rising transition of FDIV_EN will reset the chained counter and then chained counter starts to count. When FDIV_EN (FRQDIV[4]) is written with zero, the chained counter continuously runs till divided clock reaches low state and stay in low state.

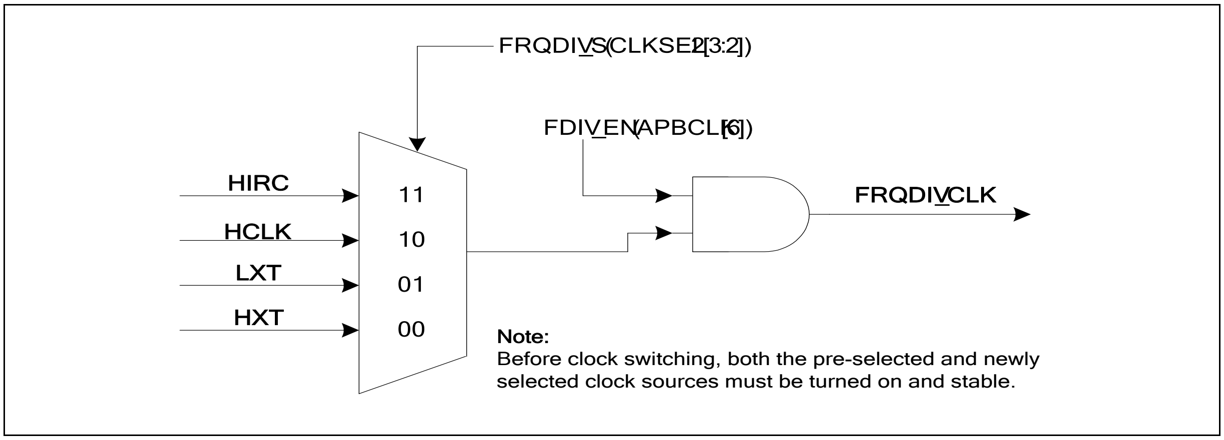


Figure 5.5-2 Clock Sources of Frequency Divider

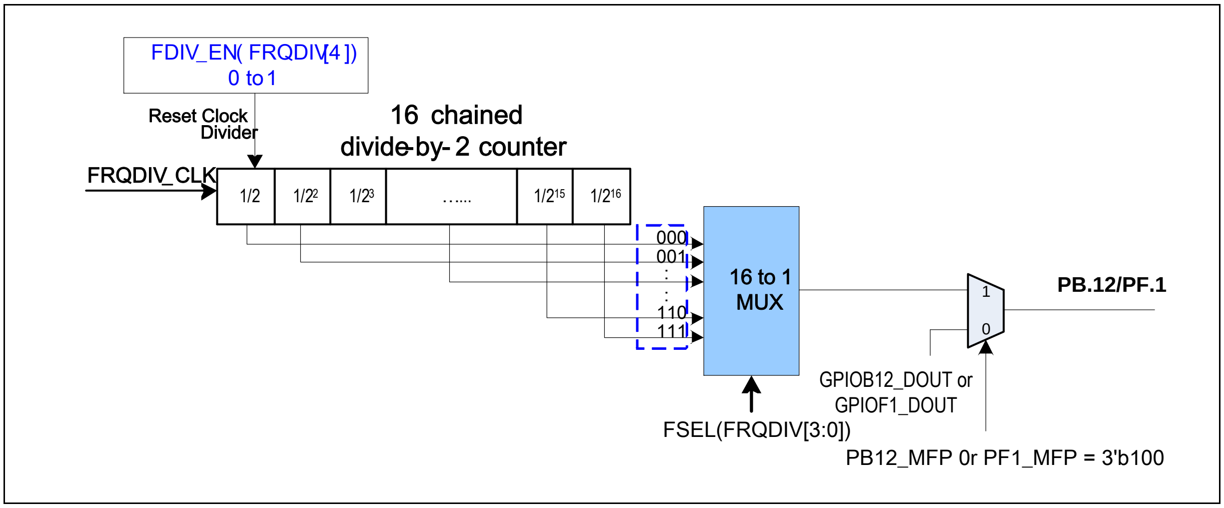


Figure 5.5-3 Frequency Divider Block Diagram

5.5.5 Register and Memory Map

R: read only**, W:** write only**, R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
CLK Base Address: **CLK_BA = 0x5000_0200**				
PWRCTL	CLK_BA+0x00	R/W	System Power Down Control Register	0x0000_031x
AHBCLK	CLK_BA+0x04	R/W	AHB Devices Clock Enable Control Register	0x0000_0035
APBCLK	CLK_BA+0x08	R/W	APB Devices Clock Enable Control Register	0x0000_0001
CLKSTATUS	CLK_BA+0x0C	R	Clock status monitor Register	0x0000_001x
CLKSEL0	CLK_BA+0x10	R/W	Clock Source Select Control Register 0	0x0000_000x
CLKSEL1	CLK_BA+0x14	R/W	Clock Source Select Control Register 1	0x0007_FFFF
CLKSEL2	CLK_BA+0x18	R/W	Clock Source Select Control Register 2	0x000F_FFFF
CLKDIV0	CLK_BA+0x1C	R/W	Clock Divider Number Register 0	0x0000_0000
CLKDIV1	CLK_BA+0x20	R/W	Clock Divider Number Register 1	0x0000_0000
PLLCTL	CLK_BA+0x24	R/W	PLL Control Register	0x0003_1120
FRQDIV	CLK_BA+0x28	R/W	Frequency Divider Control Register	0x0000_0000
MCLKO	CLK_BA+0x2C	R/W	Module Clock Output Register	0x0000_0000
WK_INTSTS	CLK_BA+0x30	R	Wake-up interrupt status	0x0000_0000

5.5.6 Register Description

Power Down Control Register (PWRCTL)

Register	Offset	R/W	Description	Reset Value
PWRCTL	CLK_BA+0x00	R/W	System Power Down Control Register	0x0000_031x

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			HXT_HF_ST		LXT_SCNT	HXT_GAIN	HXT_SELXT
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	PD_EN	PD_WK_IE	WK_DLY	LIRC_EN	HIRC_EN	LXT_EN	HXT_EN

Bits	Description	
[31:13]	**Reserved**	**Reserved**
[12:11]	**HXT_HF_ST**	**HXT Frequency Selection** Set this bit to meet HXT frequency selection **(Recommended)** 00 = HXT frequency is from 4 MHz to 12 MHz. 01 = HXT frequency is from 12 MHz to 16 MHz. 10 = HXT frequency is from 16 MHz to 24 MHz. 11 = Reserved.
[10]	**LXT_SCNT**	**LXT Stable Time Control** This is a protected register. Please refer to open lock sequence to program it. 0 = Delay 4096 LXT before LXT output 1 = Delay 8192 LXT before LXT output
[9]	**HXT_GAIN**	**HXT Gain Control Bit** This is a protected register. Please refer to open lock sequence to program it. Gain control is used to enlarge the gain of crystal to make sure crystal wok normally. If gain control is enabled, crystal will consume more power than gain control off. 0 = Gain control Disabled. It means HXT gain is always high. For 16MHz to 24MHz crystal. 1 = Gain control Enabled. HXT gain will be high lasting 2ms then low. This is for power saving. For 4MHz to 16MHz crystal.
[8]	**HXT_SELXT**	**HXT SELXT** This is a protected register. Please refer to open lock sequence to program it. 0 = High frequency crystal loop back path Disabled. It is used for external oscillator.

Bits	Description	
		1 = High frequency crystal loop back path Enabled. It is used for external crystal.
[7]	**Reserved**	**Reserved**
[6]	**PD_EN**	**Chip Power-down mode Enable Bit** This is a protected register. Please refer to open lock sequence to program it. When CPU sets this bit, the chip power down is enabled and chip will not enter Power-down mode until CPU sleep mode is also active. When chip wakes up from Power-down mode, this bit will be auto cleared. When chip is in Power-down mode, the LDO, HXT and HIRC will be disabled, but LXT and LIRC are not controlled by Power-down mode. When power down, the PLL and system clock (CPU, HCLKx and PCLKx) are also disabled no matter the Clock Source selection. Peripheral clocks are not controlled by this bit, if peripheral Clock Source is from LXT or LIRC. In Power-down mode, flash macro power is ON. 1 = Chip power down Enabled. 0 = Chip operated in Normal mode.
[5]	**PD_WK_IE**	**Power-down Mode Wake-up Interrupt Enable** This is a protected register. Please refer to open lock sequence to program it. 0 = Disabled 1 = Enabled. PD_WK_INT will be set if both PD_WK_IS and PD_WK_IE are high.
[4]	**WK_DLY**	**Wake-up Delay Counter Enable** This is a protected register. Please refer to open lock sequence to program it. When chip wakes up from Power-down mode, the clock control will delay 4096 clock cycles to wait HXT stable or 16 clock cycles to wait HIRC stable. 1 = Delay clock cycle Enabled. 0 = Delay clock cycle Disabled.
[3]	**LIRC_EN**	**LIRC Control** This is a protected register. Please refer to open lock sequence to program it. 1 = Enabled 0 = Disabled LIRC is enabled by default.
[2]	**HIRC_EN**	**HIRC Control** This is a protected register. Please refer to open lock sequence to program it. 1 = Enabled 0 = Disabled HIRC is enabled by default.
[1]	**LXT_EN**	**LXT Control** This is a protected register. Please refer to open lock sequence to program it. 1 = Enabled 0 = Disabled LXT is disabled by default.
[0]	**HXT_EN**	**HXT Control**

Bits	Description	
		This is a protected register. Please refer to open lock sequence to program it. The bit default value is set by flash controller user configuration register config0 [26]. 1 = Enabled 0 = Disabled HXT is disabled by default.

Mode	PD_EN	CPU run WFI instruction	Clock Gating
Normal Mode	0	NO	Depending on S/W setting
Idle Mode (CPU entry sleep mode)	0	YES	Only CPU clock gating
Power-down Mode	1	YES	Most Clocks are gating except LXT or LIRC and IP adopting LXT or LIRC. S/W can turn off LXT and LIRC before chip enters Power-down mode.

Table 5.5-2 Power Modes and Clocks

AHB Devices Clock Enable Control Register (AHBCLK)

These register bits are used to enable/disable AHB IP HCLK and engine clocks

Register	Offset	R/W	Description	Reset Value
AHBCLK	CLK_BA+0x04	R/W	AHB Devices Clock Enable Control Register	0x0000_0035

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**TICK_EN**	**SRAM_EN**	**EBI_EN**	**ISP_EN**	**DMA_EN**	**GPIO_EN**

Bits	Description	
[31:6]	**Reserved**	**Reserved**
[5]	**TICK_EN**	**System Tick Clock Enable** 1 = Enabled 0 = Disabled
[4]	**SRAM_EN**	**SRAM Controller Clock Enable** 1 = Enabled 0 = Disabled
[3]	**EBI_EN**	**EBI Controller Clock Enable** 1 = Enabled 0 = Disabled
[2]	**ISP_EN**	**Flash ISP Controller Clock Enable** 1 = Enabled 0 = Disabled
[1]	**DMA_EN**	**DMA Controller Clock Enable** 1 = Enabled 0 = Disabled
[0]	**GPIO_EN**	**GPIO Controller Clock Enable** 1 = Enabled 0 = Disabled

APB Devices Clock Enable Control Register (APBCLK)

These register bits are used to enable/disable APB IP PCLK and engine clocks.

Register	Offset	R/W	Description	Reset Value
APBCLK	CLK_BA+0x08	R/W	APB Devices Clock Enable Control Register	0x0000_0001

31	30	29	28	27	26	25	24
SC1_EN	SC0_EN	I2S_EN	ADC_EN	USBD_EN	LCD_EN	DAC_EN	Reserved
23	22	21	20	19	18	17	16
PWM1_CH23_EN	PWM1_CH01_EN	PWM1_CH23_EN	PWM0_CH01_EN	Reserved		UART1_EN	UART0_EN
15	14	13	12	11	10	9	8
Reserved	SPI2_EN	SPI1_EN	SPI0_EN	Reserved		I2C1_EN	I2C0_EN
7	6	5	4	3	2	1	0
SC2_EN-	**FDIV_EN**	**TMR3_EN**	**TMR2_EN**	**TMR1_EN**	**TMR0_EN**	**RTC_EN**	**WDT_EN**

Bits	Description	
[31]	**SC1_EN**	**SmartCard 1 Clock Enable Control.** 1 = Enabled 0 = Disabled
[30]	**SC0_EN**	**SmartCard 0 Clock Enable Control.** 1 = Enabled 0 = Disabled
[29]	**I2S_EN**	**I2S Clock Enable Control.** 1 = Enabled 0 = Disabled
[28]	**ADC_EN**	**Analog-Digital-Converter (ADC) Clock Enable Control.** 1 = Enabled 0 = Disabled
[27]	**USBD_EN**	**USB FS Device Controller Clock Enable Control** 1 = Enabled 0 = Disabled
[26]	**LCD_EN**	**LCD controller Clock Enable Control** 1 = Enabled 0 = Disabled
[25]	**DAC_EN**	**12-bit DAC Clock Enable Control** 1 = Enabled 0 = Disabled
[24]	**Reserved**	**Reserved**

Bits	Description	
[23]	**PWM1_CH23_EN**	**PWM1 Channel 2 and Channel 3 Clock Enable Control.** 1 = Enabled 0 = Disabled
[22]	**PWM1_CH01_EN**	**PWM1 Channel 0 and Channel 1 Clock Enable Control.** 1 = Enabled 0 = Disabled
[21]	**PWM0_CH23_EN**	**PWM0 Channel 2 and Channel 3 Clock Enable Control.** 1 = Enabled 0 = Disabled
[20]	**PWM0_CH01_EN**	**PWM0 Channel 0 and Channel 1Clock Enable Control.** 1 = Enabled 0 = Disabled
[19:18]	**Reserved**	**Reserved**
[17]	**UART1_EN**	**UART1 Clock Enable Control.** 1 = Enabled 0 = Disabled
[16]	**UART0_EN**	**UART0 Clock Enable Control.** 1 = Enabled 0 = Disabled
[15]	**Reserved**	**Reserved**
[14]	**SPI2_EN**	**SPI2 Clock Enable Control.** 1 = Enabled 0 = Disabled
[13]	**SPI1_EN**	**SPI1 Clock Enable Control.** 1 = Enabled 0 = Disabled
[12]	**SPI0_EN**	**SPI0 Clock Enable Control.** 1 = Enabled 0 = Disabled
[11:10]	**Reserved**	**Reserved**
[9]	**I2C1_EN**	**I2C1 Clock Enable Control.** 1 = Enabled 0 = Disabled
[8]	**I2C0_EN**	**I2C0 Clock Enable Control.** 1 = Enabled 0 = Disabled
[7]	**SC2_EN**	**SmartCard 2 Clock Enable Control.** 1 = Enabled

Bits	Description	
		0 = Disabled
[6]	**FDIV_EN**	**Frequency Divider Output Clock Enable Control** 1 = Enabled 0 = Disabled
[5]	**TMR3_EN**	**Timer3 Clock Enable Control** 1 = Enabled 0 = Disabled
[4]	**TMR2_EN**	**Timer2 Clock Enable Control** 1 = Enabled 0 = Disabled
[3]	**TMR1_EN**	**Timer1 Clock Enable Control** 1 = Enabled 0 = Disabled
[2]	**TMR0_EN**	**Timer0 Clock Enable Control** 1 = Enabled 0 = Disabled
[1]	**RTC_EN**	**Real-Time-Clock Clock Enable Control.** This bit is used to control the RTC APB clock only, The RTC engine Clock Source is from LXT. 1 = Enabled 0 = Disabled
[0]	**WDT_EN**	**Watchdog Timer Clock Enable Control.** This is a protected register. Please refer to open lock sequence to program it. This bit is used to control the WDT APB clock only, The WDT engine Clock Source is from LIRC. 1 = Enabled 0 = Disabled

Clock Status Register (CLKSTATUS)

These register bits are used to monitor if the chip Clock Source stable or not, and if clock switch is fail

Register	Offset	R/W	Description	Reset Value
CLKSTATUS	CLK_BA+0x0C	R	Clock status monitor Register	0x0000_001x

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CLK_SW_FAIL	Reserved		HIRC_STB	LIRC_STB	PLL_STB	LXT_STB	HXT_STB

Bits	Description	
[31:8]	**Reserved**	**Reserved**
[7]	**CLK_SW_FAIL**	**Clock Switch Fail Flag** 1 = Clock switch fail 0 = Clock switch success This bit will be set when target switch Clock Source is not stable. This bit is write 1 clear
[6:5]	**Reserved**	**Reserved**
[4]	**HIRC_STB**	**HIRC Clock Source Stable Flag** 1 = HIRC clock is stable 0 = HIRC clock is not stable or not enable
[3]	**LIRC_STB**	**LIRC Clock Source Stable Flag** 1 = LIRC clock is stable 0 = LIRC clock is not stable or not enable
[2]	**PLL_STB**	**PLL Clock Source Stable Flag** 1 = PLL clock is stable 0 = PLL clock is not stable or not enable
[1]	**LXT_STB**	**LXT Clock Source Stable Flag** 1 = LXT clock is stable 0 = LXT clock is not stable or not enable
[0]	**HXT_STB**	**HXT Clock Source Stable Flag** 1 = HXT clock is stable 0 = HXT clock is not stable or not enable

Clock Source Select Control Register 0 (CLKSEL0)

Register	Offset	R/W	Description	Reset Value
CLKSEL0	CLK_BA+0x10	R/W	Clock Source Select Control Register 0	0x0000_000x

31	30	29	28	27	26	25	24
Reserved							
20	**20**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
12	**12**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**HCLK_S**		

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:3]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[2:0]</td><td>HCLK_S</td><td>HCLK Clock Source Selection.
This is a protected register. Please refer to open lock sequence to program it.
Note:
Before Clock Source switches, the related clock sources (pre-select and new-select) must be turn on
The 3-bit default value is reloaded with the value of CFOSC (Config0[26:24]) in user configuration register in Flash controller by any reset. Therefore the default value is either 000b or 111b.
<table>
<tr><th>HCLK_S</th><th>Clock Source</th></tr>
<tr><td>000</td><td>HXT</td></tr>
<tr><td>001</td><td>LXT</td></tr>
<tr><td>010</td><td>PLL clock</td></tr>
<tr><td>011</td><td>LIRC</td></tr>
<tr><td>111</td><td>HIRC</td></tr>
</table></td></tr>
</table>

Clock Source Select Control Register 1 (CLKSEL1)

Register	Offset	R/W	Description	Reset Value
CLKSEL1	CLK_BA+0x14	R/W	Clock Source Select Control Register 1	0x0007_FFFF

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					LCD_S	Reserved	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	TMR1_S			Reserved	TMR0_S		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PWM0_CH23_S		PWM0_CH01_S		ADC_S		UART_S	

Bits	Description	
[31:19]	Reserved	**Reserved**
[18]	**LCD_S**	**LCD Clock Source Selection** 0 = Clock Source from LXT 1 = Reserved
[17:15]	**Reserved**	**Reserved**
[14:12]	**TMR1_S**	**Timer1 Clock Source Selection** <table><tr><th>TMR1_S</th><th>Clock Source</th></tr><tr><td>000</td><td>HXT</td></tr><tr><td>001</td><td>LXT</td></tr><tr><td>010</td><td>LIRC</td></tr><tr><td>011</td><td>external pin</td></tr><tr><td>1xx</td><td>HIRC</td></tr></table>
[11]	Reserved	**Reserved**
[10:8]	**TMR0_S**	**Timer0 Clock Source Selection** <table><tr><th>TMR0_S</th><th>Clock Source</th></tr><tr><td>000</td><td>HXT</td></tr><tr><td>001</td><td>LXT</td></tr><tr><td>010</td><td>LIRC</td></tr><tr><td>011</td><td>external pin</td></tr><tr><td>1xx</td><td>HIRC</td></tr></table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[7:6]</td><td>PWM0_CH23_S</td><td>PWM0 channel 2 and channel 3 Clock Source Selection
PWM0 channel 2 and channel 3 use the same Engine clock source, both of them with the same prescaler
<table>
<tr><th>PWM0_CH23_S</th><th>Clock Source</th></tr>
<tr><td>00</td><td>HXT</td></tr>
<tr><td>01</td><td>LXT</td></tr>
<tr><td>10</td><td>HCLK</td></tr>
<tr><td>11</td><td>HIRC</td></tr>
</table></td></tr>
<tr><td>[5:4]</td><td>PWM0_CH01_S</td><td>PWM0 channel 0 and channel 1 Clock Source Selection
PWM0 channel 0 and channel 1 use the same Engine clock source, both of them with the same prescaler
<table>
<tr><th>PWM0_CH01_S</th><th>Clock Source</th></tr>
<tr><td>00</td><td>HXT</td></tr>
<tr><td>01</td><td>LXT</td></tr>
<tr><td>10</td><td>HCLK</td></tr>
<tr><td>11</td><td>HIRC</td></tr>
</table></td></tr>
<tr><td>[3:2]</td><td>ADC_S</td><td>ADC Clock Source Selection
<table>
<tr><th>ADC_S</th><th>Clock Source</th></tr>
<tr><td>00</td><td>HXT</td></tr>
<tr><td>01</td><td>LXT</td></tr>
<tr><td>10</td><td>PLL clock</td></tr>
<tr><td>11</td><td>HIRC</td></tr>
</table></td></tr>
<tr><td>[1:0]</td><td>UART_S</td><td>UART 0/1 Clock Source Selection (UART0 and UART1 Use the Same Clock Source Selection)
<table>
<tr><th>UART_S</th><th>Clock Source</th></tr>
<tr><td>00</td><td>HXT</td></tr>
<tr><td>01</td><td>LXT</td></tr>
<tr><td>10</td><td>PLL clock</td></tr>
<tr><td>11</td><td>HIRC</td></tr>
</table></td></tr>
</table>

Clock Source Select Control Register 2 (CLKSEL2)

Before clock switch the related clock sources (pre-select and new-select) must be turn on

Register	Offset	R/W	Description	Reset Value
CLKSEL2	CLK_BA+0x18	R/W	Clock Source Select Control Register 2	0x000F_FFFF

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved	SPI2_S	SPI1_S	SPI0_S	SC_S		I2S_S	
15	14	13	12	11	10	9	8
Reserved	TMR3_S			Reserved	TMR2_S		
7	6	5	4	3	2	1	0
PWM1_CH23_S		PWM1_CH01_S		FRQDIV_S		Reserved	

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:23]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[22]</td><td>SPI2_S</td><td>SPI2 Clock Source Selection
0 = PLL
1 = HCLK</td></tr>
<tr><td>[21]</td><td>SPI1_S</td><td>SPI1 Clock Source Selection
0 = PLL
1 = HCLK</td></tr>
<tr><td>[20]</td><td>SPI0_S</td><td>SPI0 Clock Source Selection
0 = PLL
1 = HCLK</td></tr>
<tr><td>[19:18]</td><td>SC_S</td><td>SC Clock Source Selection
<table>
<tr><th>SC_S</th><th>Clock Source</th></tr>
<tr><td>00</td><td>HXT</td></tr>
<tr><td>01</td><td>PLL output</td></tr>
<tr><td>10</td><td>HIRC</td></tr>
<tr><td>11</td><td>HIRC</td></tr>
</table>
Note: SC0,SC1 and SC2 use the same Clock Source selection but they have different clock divider number.</td></tr>
<tr><td>[17:16]</td><td>I2S_S</td><td>I²S Clock Source Selection
<table>
<tr><th>I2S_S</th><th>Clock Source</th></tr>
<tr><td>00</td><td>HXT</td></tr>
</table></td></tr>
</table>

Bits		Description
		01 — PLL output 10 — HIRC 11 — HIRC
[15]	Reserved	Reserved
[14:12]	TMR3_S	**Timer3 Clock Source Selection** **TMR3_S** — **Clock source** 000 — HXT 001 — LXT 010 — LIRC 011 — external pin 1xx — HIRC
[11]	Reserved	Reserved
[10:8]	TMR2_S	**Timer2 Clock Source Selection** **TMR2_S** — **Clock Source** 000 — HXT 001 — LXT 010 — LIRC 011 — external pin 1xx — HIRC
[7:6]	PWM1_CH23_S	**PWM1 channel 2 and channel 2 Clock Source Selection** PWM1 channel 2 and channel 3 use the same Engine clock source, both of them with the same pre-scale **PWM1_CH23_S** — **Clock Source** 00 — HXT 01 — LXT 10 — HCLK 11 — HIRC
[5:4]	PWM1_CH01_S	**PWM1 channel 0 and channel 1 Clock Source Selection** PWM1 channel 0 and channel 1 use the same Engine clock source, both of them with the same pre-scale **PWM1_CH01_S** — **Clock Source** 00 — HXT 01 — LXT 10 — HCLK 11 — HIRC
[3:2]	FRQDIV_S	**Clock Divider Clock Source Selection**

Bits	Description		
		FRQDIV_S	**Clock Source**
		00	HXT
		01	LXT
		10	HCLK
		11	HIRC
[1:0]	**Reserved**	**Reserved**	

Clock Divider 0 Register (CLKDIV0)

Register	Offset	R/W	Description	Reset Value
CLKDIV0	CLK_BA+0x1C	R/W	Clock Divider Number Register 0	0x0000_0000

31	30	29	28	27	26	25	24
SC0_N				Reserved			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ADC_N							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
I2S_N				UART_N			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
USB_N				HCLK_N			

Bits	Description	
[31:28]	**SC0_N**	**SC 0 clock divide number from SC 0 clock source** The SC 0 clock frequency = (SC0 Clock Source frequency) / (SC0_N + 1)
[27:24]	**Reserved**	**Reserved**
[23:16]	**ADC_N**	**ADC clock divide number from ADC clock source** The ADC clock frequency = (ADC Clock Source frequency) / (ADC_N + 1)
[15:12]	**I2S_N**	**I^2S clock divide number from I^2S clock source** The I^2S clock frequency = (I^2S Clock Source frequency) / (I2S_N + 1)
[11:8]	**UART_N**	**UART clock divide number from UART clock source** The UART clock frequency = (UART Clock Source frequency) / (UART_N + 1)
[7:4]	**USB_N**	**USB clock divide number from PLL clock** The USB clock frequency = (PLL frequency) / (USB_N + 1)
[3:0]	**HCLK_N**	**HCLK clock divide number from HCLK clock source** The HCLK clock frequency = (HCLK Clock Source frequency) / (HCLK_N + 1)

Clock Divider 1 Register (CLKDIV1)

Register	Offset	R/W	Description	Reset Value
CLKDIV1	CLK_BA+0x20	R/W	Clock Divider Number Register 1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SC2_N				SC1_N			

Bits	Description	
[31:4]	**Reserved**	**Reserved**
[7:4]	**SC2_N**	**SC 2 clock divide number from SC2 clock source** The SC 2 clock frequency = (SC 2 Clock Source frequency) / (SC2_N + 1)
[3:0]	**SC1_N**	**SC 1 clock divide number from SC 1 clock source** The SC 1 clock frequency = (SC 1 Clock Source frequency) / (SC1_N + 1)

PLL Control Register (PLLCTL)

The PLL reference clock input is from HXT or HIRC. This register is used to control PLL output frequency and PLL operating mode

Register	Offset	R/W	Description	Reset Value
PLLCTL	CLK_BA+0x24	R/W	PLL Control Register	0x0003_1120

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						PLL_SRC	PD
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			OUT_DV	Reserved		IN_DV	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		FB_DV					

Bits	Description	
[17]	**PLL_SRC**	**PLL Source Clock Select** 1 = PLL source clock from HIRC 0 = PLL source clock from HXT
[16]	**PD**	**Power-down mode.** If set the PD_EN bit “1” in PWR_CTL register, the PLL will enter Power-down mode too 0 = PLL is in normal mode 1 = PLL is in power-down mode (default)
[15:13]	**Reserved**	**Reserved**
[12]	**OUT_DV**	**PLL Output Divider Control Pins** Refer to the formulas below the table. This bit MUST be 0 for PLL output low deviation.
[11:10]	**Reserved**	**Reserved**
[9:8]	**IN_DV**	**PLL Input Divider Control Pins** Refer to the formulas below the table.
[7:6]	**Reserved**	**Reserved**
[4:0]	**FB_DV**	**PLL Feedback Divider Control Pins** Refer to the formulas below the table. The range of FB_DV is from 0 to 63.

Output Clock Frequency Setting

$$FOUT = FIN \times \frac{NF}{NR} \times \frac{1}{NO}$$

Symbol	Description
FOUT	Output Clock Frequency. FOUT frequency must be greater than 48 MHz and less than 120 MHz. 48 MHz < FOUT < 120 MHz
FIN	Input (Reference) Clock Frequency
NR	IN_DV = “00” : NR = 2 IN_DV = “01” : NR = 4 IN_DV = “10” : NR = 8 IN_DV = “11” : NR = 16
NF	Feedback Divider (FB_DV + 32)
NO	OUT_DV = 0 →NO =1 OUT_DV = 1 →NO = 2 (Not recommended)

Some recommended setting

PLL clock input source	PLL clock input frequency (MHz)	PLL clock output frequency (MHz)	PLLCTL setting
HXT	4	120	0x1C
		96	0x10
		48	0x110
	8	120	0x120
		96	0x110
		48	0x210
	12	120	0x108
		96	0x220
		48	0x320
	16	120	0x220
		96	0x210
		48	0x310
	24	120	0x208
		96	0x200
		48	0x300
HIRC	12	120	0x20108
		96	0x20220
		48	0x20320

Frequency Divider Control Register (FRQDIV)

Register	Offset	R/W	Description	Reset Value
FRQDIV	CLK_BA+0x28	R/W	Frequency Divider Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**FDIV_EN**	**FSEL**			

Bits	Description	
[31:5]	**Reserved**	**Reserved**
[4]	**FDIV_EN**	**Frequency Divider Enable Bit** 0 = Frequency Divider Disabled 1 = Frequency Divider Enabled
[3:0]	**FSEL**	**Divider Output Frequency Selection Bits** The formula of output frequency is Fout = Fin/2^(N+1), Where Fin is the input clock frequency, Fout is the frequency of divider output clock and N is the 4-bit value of FSEL[3:0].

Module Clock Output Register (MCLKO)

Register	Offset	R/W	Description	Reset Value
MCLKO	CLK_BA+ 0x2C	R/W	Module Clock Output Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
20	**20**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
12	**12**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MCLK_EN	**Reserved**	**MCLK_SEL**					

Bits	Description	
[31:5]	Reserved	**Reserved**
[7]	**MCLK_EN**	**Module Clock Output Enable** User can get the module clock output from PC.0 pin via choosing the clock source in the MCLK_SEL bit field and then setting MCLK_EN bit to 1. 0 = Module clock output Disabled. 1 = Module clock output Enabled. **Note:** If this bit is enabled, PC.0 will be configured to module clock output and the setting of PC0_MFP will be ineffective.
[6]	Reserved	**Reserved**
[5:0]	**MCLK_SEL[5:0]**	**Module Clock Output Source Selection (PC.0)**

MCLK_SEL	Clock Source
00_0000	ISP_CLK
00_0001	HIRC
00_0010	HXT
00_0011	LXT
00_0100	LIRC
00_0101	PLL output
00_0110	PLL input
00_0111	System Tick
00_1000	HCLK clock
00_1010	PCLK clock
10_0000	TMR0_CLK

Bits	Description		
		10_0001	TMR1_CLK
		10_0010	UART0_CLK
		10_0011	USB_CLK
		10_0100	ADC_CLK
		10_0101	WDT_CLK
		10_0110	PWM0_CH01_CLK
		10_0111	PWM0_CH23_CLK
		10_1001	LCD_CLK
		11_1000	TMR2_CLK
		11_1001	TMR3_CLK
		11_1010	UART1_CLK
		11_1011	PWM1_CH01_CLK
		11_1100	PWM1_CH23_CLK
		11_1101	I2S_CLK
		11_1110	SC0_CLK
		11_1111	SC1_CLK

Power Down Wake-up Interrupts Status (PD_WK_IS)

Register	Offset	R/W	Description	Reset Value
WK_INTSTS	CLK_BA+0x30	R/W	Wake-up interrupt status	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
20	**20**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
12	**12**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**PD_WK_IS**

Bits	Description	
[31:1]	**Reserved**	**Reserved**
[0]	**PD_WK_IS**	**Wake-up Interrupt Sstatus in chip Power-down Mode** This bit indicates that some event resumes chip from Power-down mode The status is set if external interrupts, UART, GPIO, RTC, USB, SPI, Timer, WDT, and BOD wake-up occurred. Write 1 to clear this bit.

5.6 FLASH Memory Controller (FMC)

5.6.1 Overview

This chip is equipped with 32K/64K/123K bytes on-chip embedded Flash EPROM for application program memory (APROM) that can be updated through ISP/IAP procedure. In System Programming (ISP) function enables user to update program memory when chip is soldered on PCB. After chip powered on Cortex-M0 CPU fetches code from APROM or LDROM decided by boot select (CBS) in Config0. By the way, this chip also provides DATA Flash Region, the data flash is shared with original program memory and its start address is configurable and defined by user in Config1. The data flash size is defined by user application request.

5.6.2 Features

- AHB interface compatible
- Run up to 42 MHz with zero wait state for discontinuous address read access
- 32/64/123 KB application program memory (APROM)
- 4 KB in system programming (ISP) loader program memory (LDROM)
- Programmable data flash start address and memory size with 512 bytes page erase unit
- In System Program (ISP)/In Application Program (IAP) to update on chip Flash EPROM

5.6.3 Block Diagram

The flash memory controller consists of AHB slave interface, ISP control logic, writer interface and flash macro interface timing control logic. The block diagram of flash memory controller is shown as follows.

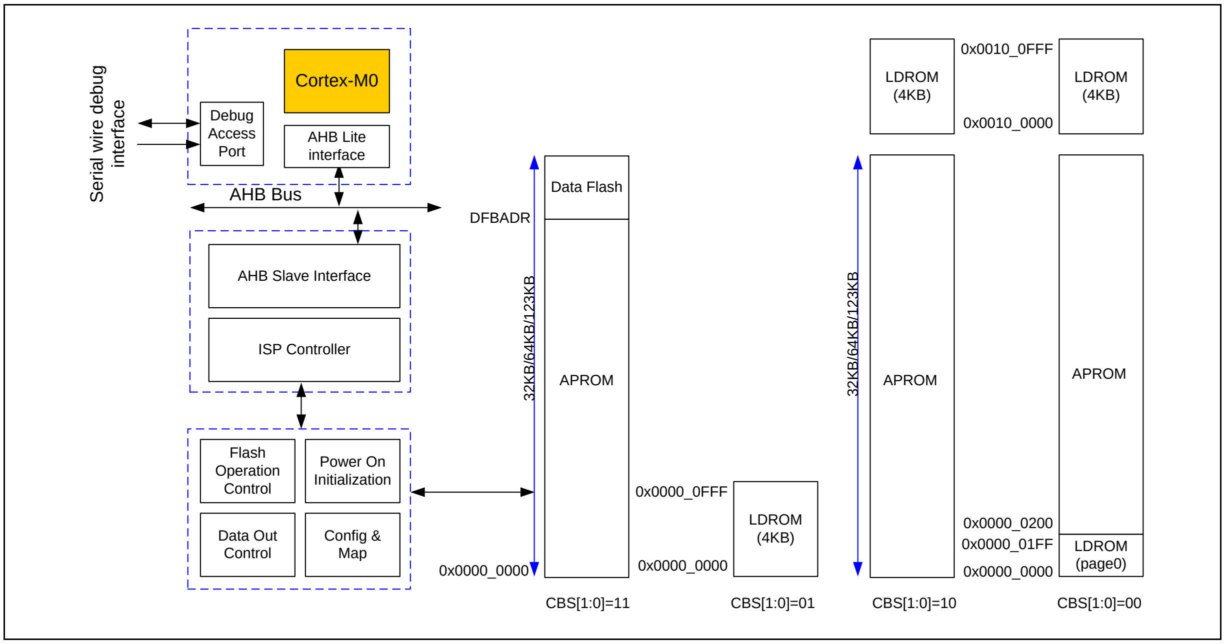


Figure 5.6-1 Block Diagram of Flash Memory Controller

5.6.4 Functional Description

5.6.4.1 Flash Memory Organization

The flash memory consists of application program memory (32 KB/64 KB/123 KB), data flash, ISP loader program memory, user configuration. User configuration block provides several bytes to control system logic, like flash security lock, boot select, Brown-out voltage level, data flash base address, and so on. It works like a fuse for power on setting. It is loaded from flash memory to its corresponding control registers during chip powered on. User can set these bits according to application request by writer before chip is mounted on PCB. The data flash start address and its size can defined by user application.

Block Name	Size	Start Address	End Address
APROM	(32-0.5*N) KB / (64-0.5*N) KB / (123-0.5*N) KB	0x0000_0000	DFBADR-1 (if DFEN=0)
Data Flash	0.5*N KB	DFBADR	0x0000_7FFF / 0x0000_FFFF / 0x0001_EBFF
Reserved for future use	901 KB	0x0001_EC00	0x000F_FFFF
LDROM	4 KB	0x0010_0000	0x0010_0FFF
User Configuration	2 words	0x0030_0000	0x0030_0004

Table 5.6-1 Memory Access Map

Part Number	NANO1XX-XCXBN	NANO1XX-XDXBN	NANO1XX-XEXBN
	Flash ROM:32 KB	Flash ROM:64 KB	Flash ROM:128 KB
APROM size	(32-0.5*N) KB	(64-0.5*N) KB	(123-0.5*N) KB
Data Flash size	(0.5*N) KB	(0.5*N) KB	(0.5*N) KB
LDROM	4 KB	4 KB	4 KB

Table 5.6-2 Flash Size

The Flash memory organization is shown below.

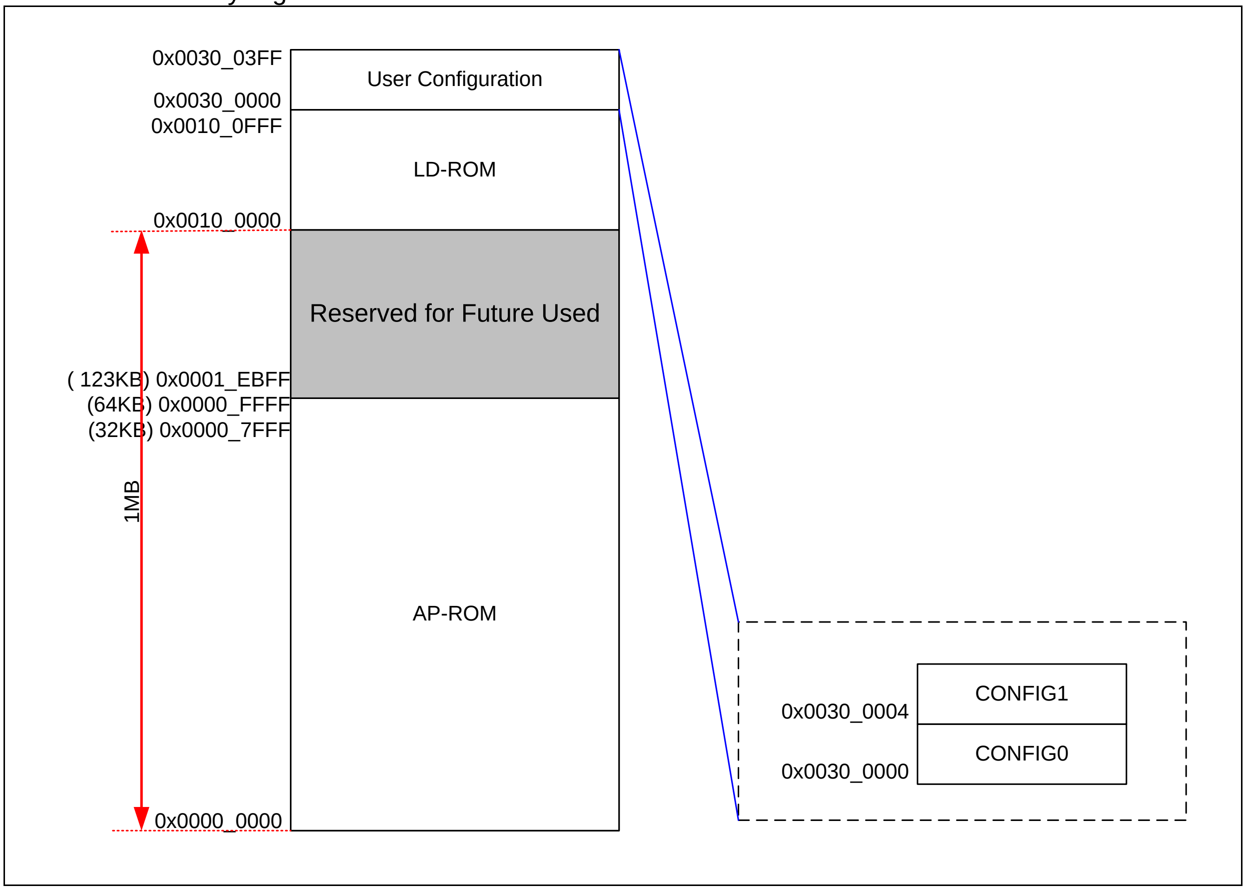


Figure 5.6-2 Flash Memory Organization

5.6.4.2 Boot Selection

This chip provides in system programming (ISP) feature for user to update program memory when chip is mounted on PCB. A dedicated 4 KB program memory (LDROM) is used to store ISP firmware. Users can select to start program from APROM or LDROM by (CBS) in Config0.

CBS[1:0]	Boot Selection
11	CPU booting from APROM, flash access range including APROM and Data Flash; LDROM can not be accessed directly, except by through ISP. APROM is write-protected in this mode.
01	CPU booting from LDROM, flash access range only 4KB LDROM; APROM can not be accessed directly, except by through ISP. APROM can be updated in this mode.
10	CPU booting from APROM, flash access range including LDROM and APROM APROM can be updated in this mode. LDROM address is mapping to 0x0010_0000~0x0010_0FFF The address 0x0000_0000 ~ 0x0000_01FF mapping can be changed to LDROM by though ISP command.
00	CPU booting from LDROM, flash access range including LDROM and most of APROM (all except page0, because the address is mapping to LDROM) APROM can be updated in this mode. LDROM address is mapping to 0x0010_0000 ~ 0x0010_0FFF, and also the first 512 bytes of LDROM is mapping to the address 0x0000_0000 ~ 0x0000_01FF. The address 0x0000_0000 ~ 0x0000_01FF mapping can be changed to APROM by though ISP command.

Table 5.6-3 Boot Selection

CBS[1:0]	Boot from	Vector Re-map	Run in LDROM Write to APROM	Run in APROM Write to LDROM	Run in LDROM Write to LDROM	Run in APROM Write to APROM
11	APROM	-	-	Yes	-	-
01	LDROM	-	Yes	-	-	-
10	APROM	Yes	-	Yes	-	Yes
00	LDROM	Yes	Yes	-	Yes	Yes

Table 5.6-4 Boot Selection and Supports Function

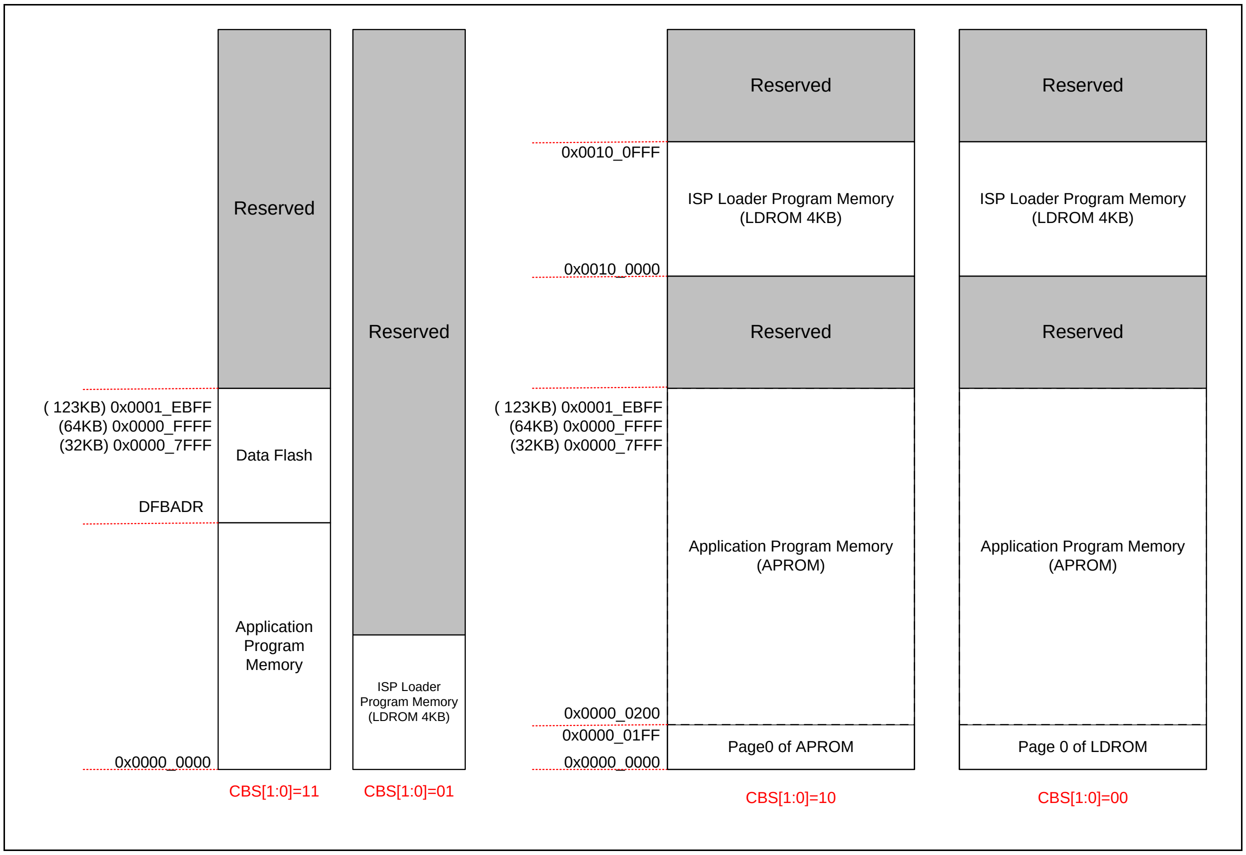


Figure 5.6-3 Flash Memory Mapping of CBS in CONFIG0

5.6.4.3 Data Flash

This chip provides data flash for user to store data. It is read/written through ISP procedure. The size of each erase unit is 512 bytes. When a word will be changed, all 128 words need to be copied to another page or SRAM in advance. The data flash base address is defined by DFBADR if DFEN bit in Config0 is enabled and application program memory size is (64-0.5*N)KB for 64KB flash, (32-0.5*N)KB for 32KB flash and data flash size is 0.5*N KB.

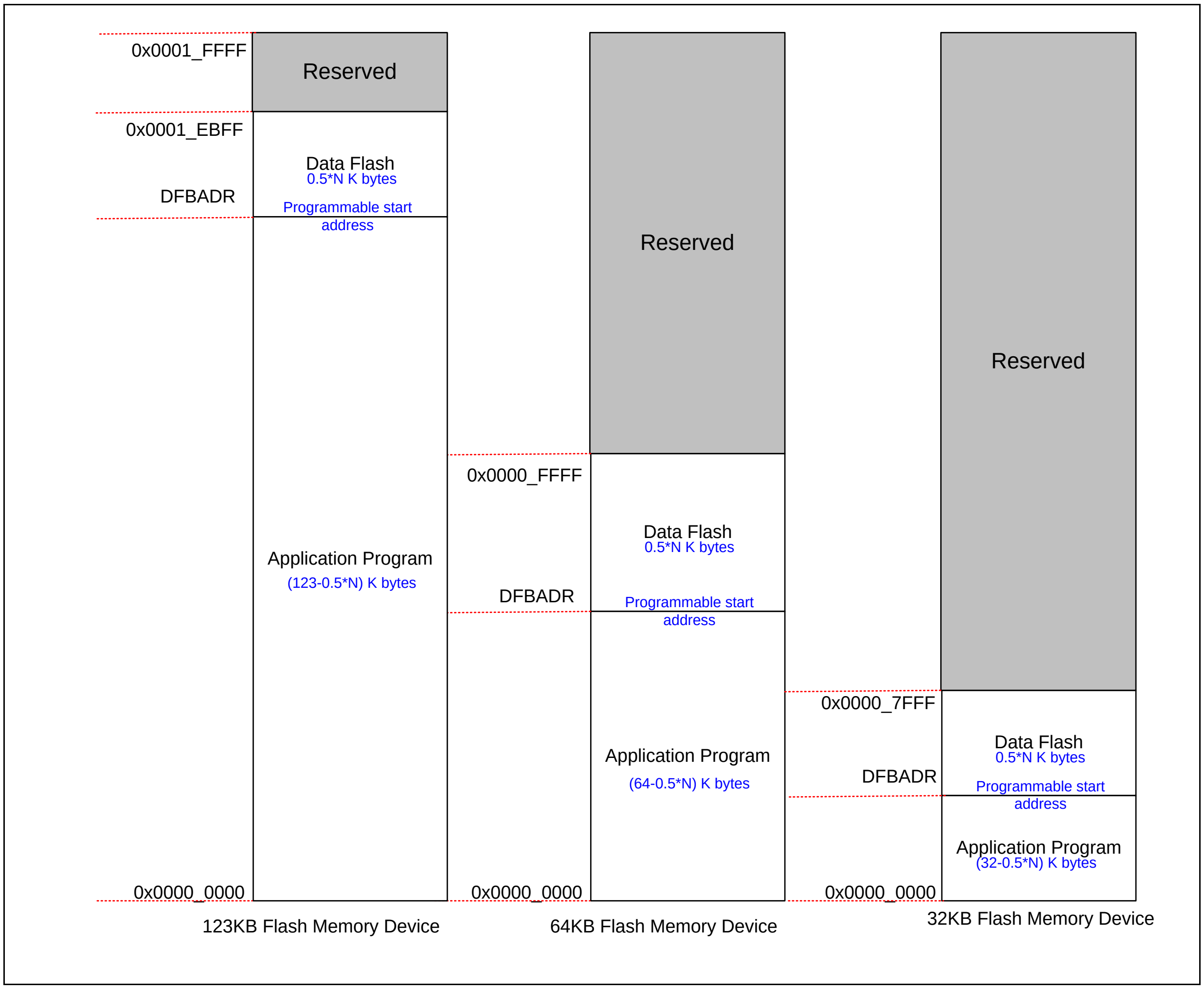


Figure 5.6-4 32/64/123 KB Flash Memory Structure

5.6.4.4 User Configuration

Config0 (Address = 0x0030_0000)

31	30	29	28	27	26	25	24
CWDT_EN	Reserved				CFOSC	Reserved	
23	22	21	20	19	18	17	16
Reserved			CBORST		Reserved		
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
CBS		Reserved				LOCK	DFEN

Config0	Address = 0x0030_0000	
Bits	**Description**	
[31]	**CWDT_EN**	**Force Watchdog Timer Clock On** 0 = Forcing the clock of Watchdog Timer to be enabled even if WDT_CTL[WTE] is set to 0. 1 = The clock of Watchdog Timer can be disabled by setting WDT_CTL[WTE] to 0.
[30:27]	**Reserved**	**Reserved**
[26]	**CFOSC**	**CPU Clock Source Selection After Reset** **CFOSC** / **Clock Source** 0 / External 12 MHz crystal clock (HXT) 1 / Internal RC 12 MHz oscillator clock (HIRC) The value of CFOSC will be load to CLKSEL0.HCLK_S[2] in system register after any reset occurs.
[25:21]	**Reserved**	**Reserved**
[20:19]	**CBORST**	**Brown-out Reset Enable Selection** **CBORST[1:0]** / **Brown-out Reset Selection** 00 / BOD17 reset enable 01 / BOD20 reset enable 10 / BOD25 reset enable 11 / Disable all BOD function
[18:8]	**Reserved**	**Reserved**

<table>
<tr><td>[7:6]</td><td>CBS</td><td>CONFIG Boot Selection
<table>
<tr><th>CBS[1:0]</th><th>Boot Selection</th></tr>
<tr><td>11</td><td>APROM(LDROM invisible)
CPU booting from APROM, flash access range including APROM and Data Flash; LDROM cannot be access directly, except by through ISP.
APROM is write-protected in this mode.</td></tr>
<tr><td>01</td><td>LDROM(APROM invisible)
CPU booting from LDROM, flash access range only LDROM 4KB; APROM cannot be access directly, except by through ISP.
APROM can be updated in this mode.</td></tr>
<tr><td>10</td><td>APROM
CPU booting from APROM, flash access range including LDROM and APROM
APROM can be updated in this mode.
LDROM address is mapping to 0x0010_0000~0x0010_0FFF
The address 0x0000_0000 ~ 0x0000_01FF mapping can be change to LDROM by though ISP command.</td></tr>
<tr><td>00</td><td>LDROM
CPU booting from LDROM, flash access range including LDROM and most of APROM (all but except page0, because the address is mapping to LDROM)
APROM can be updated in this mode.
LDROM address is mapping to 0x0010_0000 ~ 0x0010_0FFF, and also the first 512 bytes of LDROM is mapping to the address 0x0000_0000 ~ 0x0000_01FF.
The address 0x0000_0000 ~ 0x0000_01FF mapping can be change to APROM by though ISP command.</td></tr>
</table></td></tr>
<tr><td>[5:2]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[1]</td><td>LOCK</td><td>Security Lock
0 = Flash data is locked
1 = Flash data is not locked.
When flash data is locked, only Device ID, Config0 and Config1 can be read by writer and ICP through serial debug interface. Others data is locked as 0xFFFFFFFF. ISP can read data anywhere regardless of LOCK bit value.</td></tr>
<tr><td>[0]</td><td>DFEN</td><td>Data Flash Enable
0 = Data flash Enabled.
1 = Data flash Disabled.
This bit is valid when CBS[1:0] = 11 or 10.</td></tr>
</table>

Config1 (Address = 0x0030_0004)

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved				DFBA			
15	14	13	12	11	10	9	8
DFBA							
7	6	5	4	3	2	1	0
DFBA							

Config1	Address = 0x0030_0004	
Bits	**Description**	
[31:20]	Reserved	**Reserved** It is mandatory to program 0x00 to these Reserved bits
[19:0]	DFBA	**Data Flash Base Address** The data flash base address is defined by user. Since on chip flash erase unit is 512-byte, it is mandatory to keep bit 8-0 as 0.

5.6.4.5 In System Program (ISP)

The application program memory and data flash supports both hardware programming mode and In System Programming (ISP) mode. Hardware programming mode uses gang-writers to reduce programming costs and time to market while the products enter into the mass production state. However, if the product is just under development or the end product needs firmware updating in the hand of an end user, the hardware programming mode will make repeated programming difficult and inconvenient. ISP method makes it easy and possible. This chip supports ISP mode allowing a device to be reprogrammed under software control. Furthermore, the capability to update the application firmware makes wide range of applications possible.

ISP is performed without removing the microcontroller from the system. Various interfaces enable LDROM firmware to get new program code easily. The most common method to perform ISP is via UART along with the firmware in LDROM. General speaking, PC transfers the new APROM code through serial port. Then LDROM firmware receives it and re-programs into APROM through ISP commands. Nuvoton provides ISP firmware and PC application program for this chip. It makes users quite easy to perform ISP through Nuvoton ISP tool.

ISP Procedure

This chip supports booting from APROM or LDROM initially defined by user configuration bit (CBS). If user wants to update application program in APROM, user can write BS=1 and starts software reset to make chip boot from LDROM. The first step to start ISP function is to write ISPEN bit to 1. S/W is required to write RegLockAddr register in Global Control Register (GCR, 0x5000_0100) with 0x59, 0x16 and 0x88 before writing ISPCON register. This procedure is used to protect flash memory from destroying owning to unintended write during power on/off duration.

Several error conditions are checked after software writes ISPGO bit. If error condition occurs, ISP operation is not been started and ISP fail flag will be set instead of. ISPFF flag is cleared by S/W but it will not be overwritten in next ISP operation. The next ISP procedure can be started even ISPFF bit keeps at 1. It is recommended that s/w to check ISPFF bit and clear it after each ISP operation if it is set to 1.

When ISPGO bit is set, CPU will wait for ISP operation finish, during this period; peripheral still keeps working as usual. If any interrupt request occur, CPU will not service it till ISP operation finish.

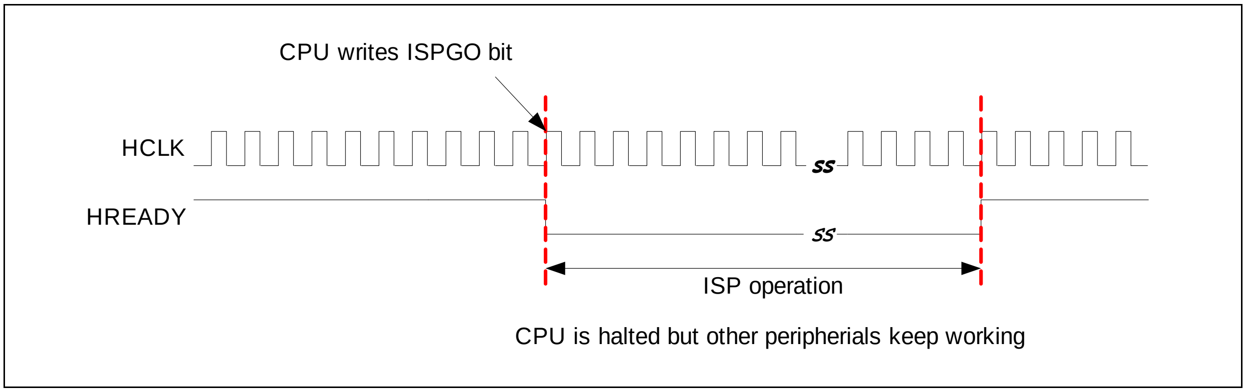


Figure 5.6-5 CPU Halt during ISP Operation

Note that this chip allows user to update CONFIG value by ISP, but for application program code security issue, s/w is required to erase APROM by page erase before erase CONFIG. Otherwise, erase CONFIG will not be allowed.

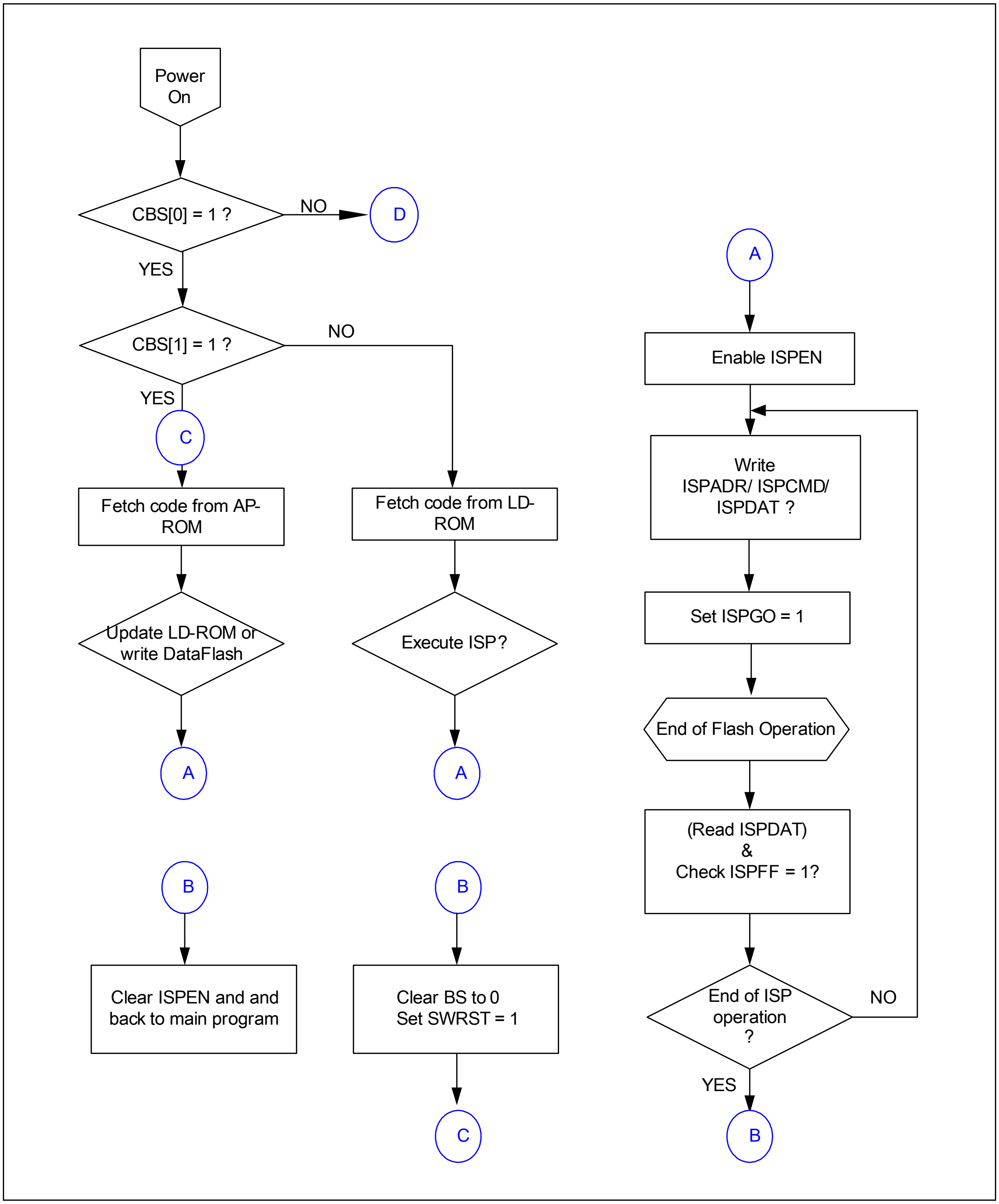


Figure 5.6-6 ISP Operation Flow

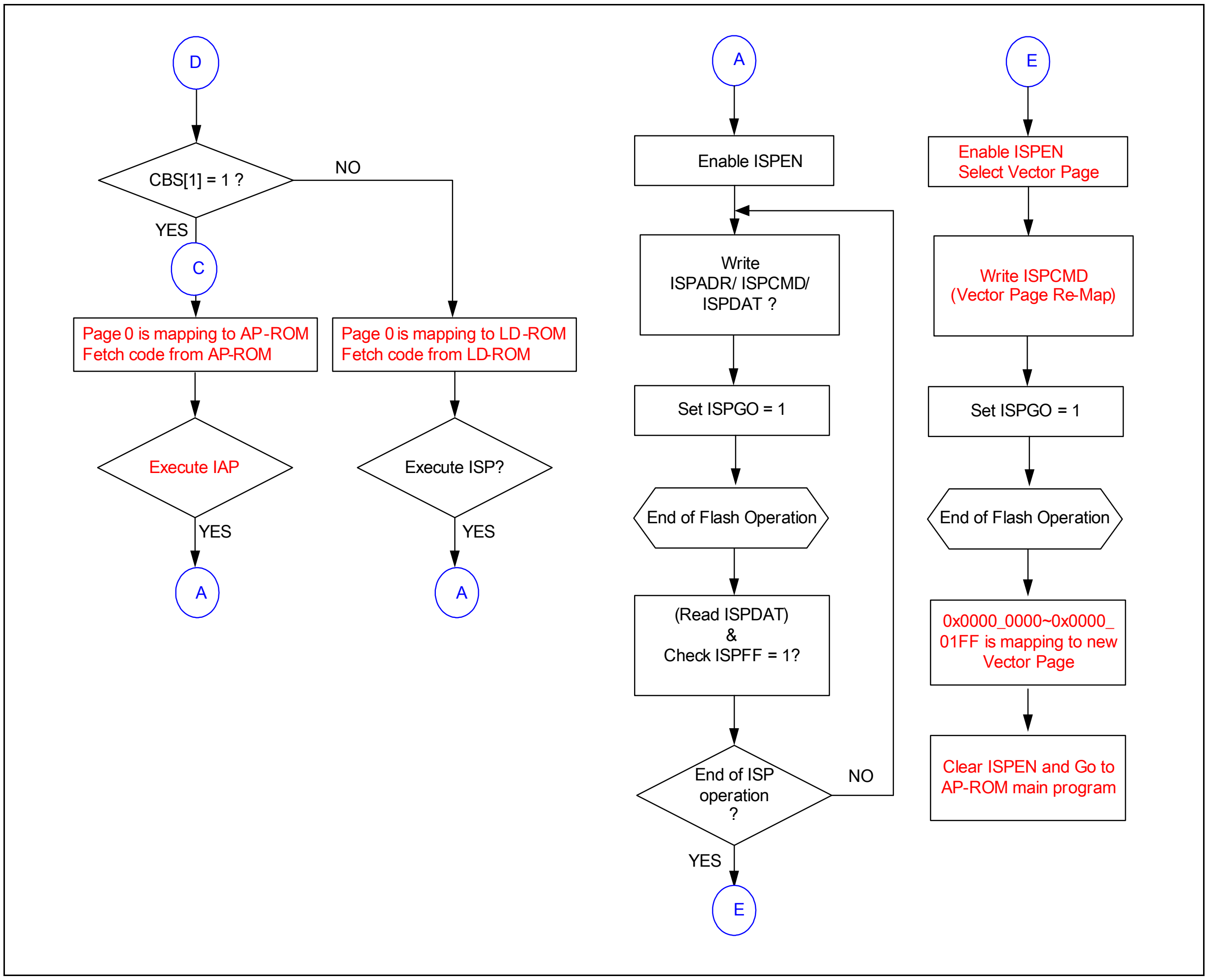


Figure 5-38 ISP Operation Flow (Continued)

ISP Mode	ISPCMD			ISPADR			ISPDAT
	FOEN	FCEN	FCTRL[3:0]	A21	A20	A[19:0]	D[31:0]
Standby	1	1	X	x	x	x	x
Read Company ID	0	0	1011	x	x	x	Data out D[31:0] = 0x0000_00DA
Read Device ID	0	0	1100	x	x	Address in A[19:0] = 0x00000	Data out D[31:0]= Device ID
Read Unique ID	0	0	0100	x	x	Address in A[19:0] = 0x00000 0x00004 0x00008	Data out D[31:0]= Unique ID
*Read Unique Customer ID	0	0	0100	x	x	Address in A[19:0] = 0x00010 0x00014 0x00018 0x0001C	Data out D[31:0]= Unique Customer ID
Vector Page Re-Map	1	0	1110	0	A20	Address in A[19:0]	x
FLASH Page Erase	1	0	0010	0	A20	Address in A[19:0]	x
FLASH Program	1	0	0001	0	A20	Address in A[19:0]	Data in D[31:0]
FLASH Read	0	0	0000	0	A20	Address in A[19:0]	Data out D[31:0]
CONFIG Page Erase	1	0	0010	1	1	Address in A[19:0]	x
CONFIG Program	1	0	0001	1	1	Address in A[19:0]	Data in D[31:0]
CONFIG Read	0	0	0000	1	1	Address in A[19:0]	Data out D[31:0]

Table 5.6-5 ISP Operation Command

***** The default value of “Unique Customer ID” is 0xFFFF which is from address 0x00010 to 0x0001C. “Unique Customer ID” only can be configured by Nuvoton, please contact Nuvoton or agent to deal with specific customer ID.

5.6.5 Register and Memory Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
FMC Base Address: **FMC_BA = 0x5000_C000**				
ISPCON	FMC_BA+0x00	R/W	ISP Control Register	0x0000_0000
ISPADR	FMC_BA+0x04	R/W	ISP Address Register	0x0000_0000
ISPDAT	FMC_BA+0x08	R/W	ISP Data Register	0x0000_0000
ISPCMD	FMC_BA+0x0C	R/W	ISP Command Register	0x0000_0000
ISPTRG	FMC_BA+0x10	R/W	ISP Trigger Register	0x0000_0000
DFBADR	FMC_BA+0x14	R	Data Flash Base Address	0x0001_F000
ISPSTA	FMC_BA+0x40	R/W	ISP Status Register	0x0000_0000

5.6.6 Register Description

ISP Control Register (ISPCON)

Register	Offset	R/W	Description	Reset Value
ISPCON	FMC_BA+0x00	R/W	ISP Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	ISPFF	LDUEN	CFGUEN	APUEN		BS	ISPEN

Bits	Description	
[31:7]	**Reserved**	**Reserved**
[6]	**ISPFF**	**ISP Fail Flag (Write-protection Bit)** This bit is set by hardware when a triggered ISP meets any of the following conditions: (1) APROM writes to itself (2) LDROM writes to itself (3) CONFIG is erased/programmed if CFGUEN is set to 0 (4) Destination address is illegal, such as over an available range Write 1 to clear.
[5]	**LDUEN**	**LDROM Update Enable (Write-protection Bit)** LDROM update enable bit. 1 = LDROM can be updated when the chip runs in APROM 0 = LDROM cannot be updated
[4]	**CFGUEN**	**Enable Config-bits Update by ISP (Write-protection Bit)** 1 = Enabling ISP can update config-bits 0 = Disabling ISP can update config-bits
[3]	**APUEN**	**APROM Update Enable (Write-protection Bit)** APROM update enable bit. 1 = APROM can be updated when the MCU runs in APROM. 0 = APROM can not be updated
[2]	**Reserved**	**Reserved**

[1]	**BS**	**Boot Select (Write-protection Bit)** Set/clear this bit to select next booting from LDROM/APROM, respectively. This bit also functions as chip booting status flag, which can be used to check where chip booted from. This bit is initiated with the inversed value of CBS in Config0 after power-on reset; It keeps the same value at other reset. 1 = boot from LDROM 0 = boot from APROM
[0]	**ISPEN**	**ISP Enable (Erite-protection Bit)** ISP function enable bit. Set this bit to enable ISP function. 1 = ISP function Enabled. 0 = ISP function Disabled.

ISP Address (ISPADR)

Register	Offset	R/W	Description	Reset Value
ISPADR	FMC_BA+0x04	R/W	ISP Address Register	0x0000_0000

31	30	29	28	27	26	25	24
ISPADR[31:24]							
23	22	21	20	19	18	17	16
ISPADR[23:16]							
15	14	13	12	11	10	9	8
ISPADR[15:8]							
7	6	5	4	3	2	1	0
ISPADR[7:0]							

Bits	Description	
[31:0]	**ISPADR**	**ISP Address** This chip supports word program only. ISPADR[1:0] must be kept 00b for ISP operation, and ISPADR[8:0] must be kept 0_0000_0000b for Vector Page Re-map Command

ISPDAT (ISP Data Register)

Register	Offset	R/W	Description	Reset Value
ISPDAT	FMC_BA+0x08	R/W	ISP Data Register	0x0000_0000

31	30	29	28	27	26	25	24
ISPDAT[31:24]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ISPDAT [23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ISPDAT [15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ISPDAT [7:0]							

Bits	Description	
[31:0]	**ISPDAT**	**ISP Data** Write data to this register before ISP program operation Read data from this register after ISP read operation

ISP Command (ISPCMD)

Register	Offset	R/W	Description	Reset Value
ISPCMD	FMC_BA+0x0C	R/W	ISP Command Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		FOEN	FCEN	FCTRL			

Bits	Description	
[31:6]	**Reserved**	**Reserved**
[5]	**FOEN**	**ISP Command** The ISP command table is shown as follows.
[4]	**FCEN**	**ISP Command** The ISP command table is shown as follows.
[3:0]	**FCTRL**	**ISP Command** The ISP command table is shown as follows.

Operation Mode	FOEN	FCEN	FCTRL[3:0]			
Read	0	0	0	0	0	0
Vector Page Re-Map	1	0	1	1	1	0
Program	1	0	0	0	0	1
Page Erase	1	0	0	0	1	0
Read CID	0	0	1	0	1	1
Read DID	0	0	1	1	0	0
Read UID	0	0	0	1	0	0

ISP Trigger Control Register (ISPTRG)

Register	Offset	R/W	Description	Reset Value
ISPTRG	FMC_BA+0x10	R/W	ISP Trigger Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**ISPGO**

Bits	Description	
[31:1]	**Reserved**	**Reserved**
[0]	**ISPGO**	**ISP Start Trigger** Write 1 to start ISP operation and this bit will be cleared to 0 by hardware automatically when ISP operation is finished. 1 = ISP is progressing. 0 = ISP operation is finished.

Data Flash Base Address Register (DFBADR)

Register	Offset	R/W	Description	Reset Value
DFBADR	FMC_BA+0x14	R	Data Flash Base Address	0x0001_F000

31	30	29	28	27	26	25	24
DFBA[31:23]							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DFBA[23:16]							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DFBA[15:8]							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DFBA[7:0]							

Bits	Description	
[31:0]	**DFBA**	**Data Flash Base Address** This register indicates data flash start address. It is a read only register. The data flash start address is defined by user. Since on chip flash erase unit is 512 bytes, it is mandatory to keep bit 8-0 as 0.

ISP Status Register (ISPSTA)

Register	Offset	R/W	Description	Reset Value
ISPSTA	FMC_BA+0x40	R/W	ISP Status Register	0x0000_0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="3">Reserved</td><td colspan="5">VECMAP</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td colspan="7">VECMAP</td><td></td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>Reserved</td><td>ISPFF</td><td colspan="3">Reserved</td><td colspan="2">CBS</td><td>ISPBUSY</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:21]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[20:9]</td><td>VECMAP</td><td>Vector Page Mapping Address
The current flash address space 0x0000_0000~0x0000_01FF is mapping to address {VEC}AP[11:0], “000000000b”} ~ {VEC}AP[11:0], “111111111b”}
Read Only</td></tr>
<tr><td>[8:7]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[6]</td><td>ISPFF</td><td>ISP Fail Flag
This bit is set by hardware when a triggered ISP meets any of the following conditions:
(1) APROM writes to itself.
(2) LDROM writes to itself.
(3) CONFIG is erased/programmed when the MCU is running in APROM.
(4) Destination address is illegal, such as over an available range.
Write 1 to clear.</td></tr>
<tr><td>[5:3]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[2:1]</td><td>CBS</td><td>Config Boot Selection Status
<table>
<tr><th>CBS[1:0]</th><th>Booting Selection</th></tr>
<tr><td>01</td><td>Chip boot from LDROM; APROM is unreadable</td></tr>
<tr><td>11</td><td>Chip boot from APROM; LDROM is unreadable</td></tr>
<tr><td>00</td><td>Chip boot from page0 of LDROM;
Both LDROM and APROM are readable</td></tr>
<tr><td>10</td><td>Chip boot from page0 of APROM;
Both LDROM and APROM are readable</td></tr>
</table></td></tr>
</table>

[0]	**ISPBUSY**	**ISP BUSY** 1 = ISP operation is busy 0 = ISP operation is finished Read Only

5.7 General Purpose I/O Controller

5.7.1 Overview

Up to 86 General Purpose I/O pins can be shared with other function pins; it depends on the chip configuration. These 86 pins are arranged in 6 ports named with GPIOA, GPIOB, GPIOC, GPIOD, GPIOE and GPIOF. Ports A ~ E have the maximum of 16 pins while port F have 6 pins. Each one of the 86 pins is independent and has the corresponding register bits to control the pin mode function and data.

The I/O type of each of I/O pins can be independently software configured as input, output, and open-drain mode. Each I/O pin has a very weak individual pull-up resistor which is about 110 KΩ~300 KΩ for V_{DD} from 1.8 V to 3.6 V.

5.7.2 Features

- Up to 86 general purpose I/O pins
- Supports Input, Output, Open-drain Operation mode
- Programmable de-bounce timing
- Each I/O pin can be programmed as either edge-trigger or level-sensitive
- Each I/O pin can be programmed as either low-level active or high-level active
- Each I/O pin can be programmed as either falling-edge trigger or rising-edge trigger

5.7.3 Block Diagram

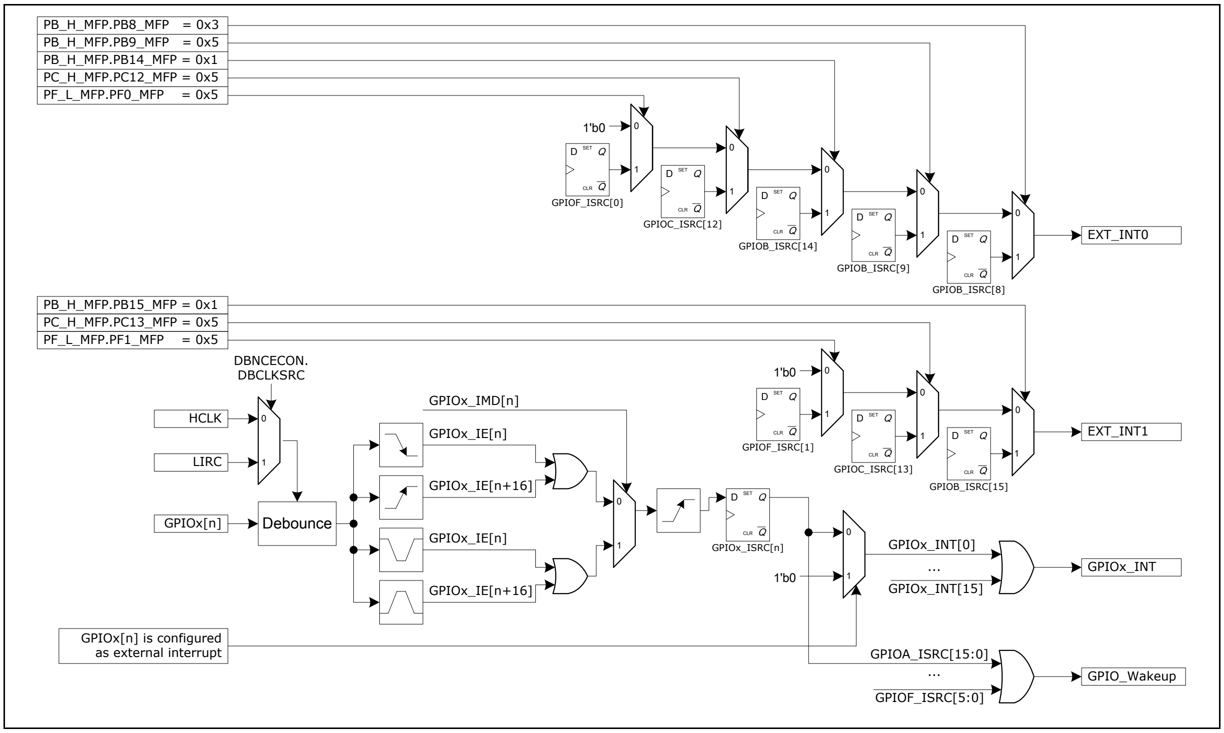


Figure 5.7-1 GPIO Block Diagram

5.7.4 Functional Description

5.7.4.1 Input Mode Explanation

Set GPIOx_PMD (PMDn [1:0]) to 00 the GPIOx port [n] pin is in Input mode and the I/O pin is in tri-state (high impedance) without output drive capability. The GPIOx_PIN value reflects the status of the corresponding port pins.

5.7.4.2 Output Mode Explanation

Set GPIOx_PMD (PMDn [1:0]) to 01 the GPIOx port [n] pin is in Output mode and the I/O pin supports digital output function with source/sink current capability. The bit value in the corresponding bit [n] of GPIO_DOUT is driven on the pin.

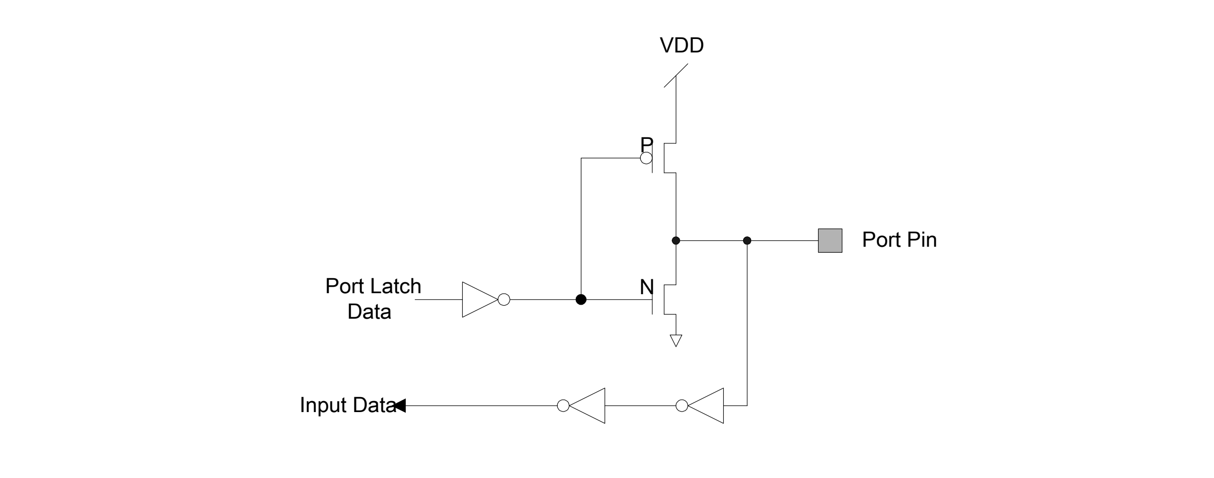


Figure 5.7-2 Push-Pull Output

5.7.4.3 Open-Drain Mode Explanation

Set GPIOx_PMD (PMDn [1:0]) to 10 the GPIOx port [n] pin is in Open-Drain mode and the I/O pin supports digital output function but only with sink current capability, an additional pull-up resister is needed for driving high state. If the bit value in the corresponding bit [n] of GPIOx_DOUT is "0", the pin drive a "low" output on the pin. If the bit value in the corresponding bit [n] of GPIOx_DOUT is "1", the pin output drives high that is controlled by the internal pull-up resistor or the external pull high resistor.

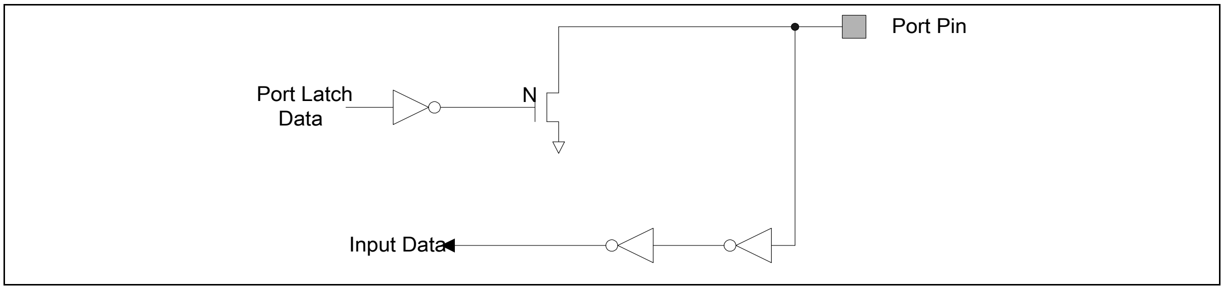


Figure 5.7-3 Open-Drain Output

5.7.5 Register and Memory Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
GP Base Address: **GP_BA =0x5000_4000**				
GPIOA_PMD	GP_BA+0x000	R/W	GPIO Port A Pin I/O Mode Control Register	0x0000_0000
GPIOA_OFFD	GP_BA+0x004	R/W	GPIO Port A Pin OFF Digital Enable Register	0x0000_0000
GPIOA_DOUT	GP_BA+0x008	R/W	GPIO Port A Data Output Value Register	0x0000_FFFF
GPIOA_DMASK	GP_BA+0x00C	R/W	GPIO Port A Data Output Write Mask Register	0x0000_0000
GPIOA_PIN	GP_BA+0x010	R	GPIO Port A Pin Value Register	0x0000_XXXX
GPIOA_DBEN	GP_BA+0x014	R/W	GPIO Port A De-bounce Enable Register	0x0000_0000
GPIOA_IMD	GP_BA+0x018	R/W	GPIO Port A Interrupt Mode Control Register	0x0000_0000
GPIOA_IER	GP_BA+0x01C	R/W	GPIO Port A Interrupt Enable Register	0x0000_0000
GPIOA_ISRC	GP_BA+0x020	R/W	GPIO Port A Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOA_PUEN	GP_BA+0x024	R/W	GPIO Port A Pull-Up Enable Register	0x0000_0000
GPIOB_PMD	GP_BA+0x040	R/W	GPIO Port B Pin I/O Mode Control Register	0x0000_0000
GPIOB_OFFD	GP_BA+0x044	R/W	GPIO Port B Pin OFF Digital Enable Register	0x0000_0000
GPIOB_DOUT	GP_BA+0x048	R/W	GPIO Port B Data Output Value Register	0x0000_FFFF
GPIOB_DMASK	GP_BA+0x04C	R/W	GPIO Port B Data Output Write Mask Register	0x0000_0000
GPIOB_PIN	GP_BA+0x050	R	GPIO Port B Pin Value Register	0x0000_XXXX
GPIOB_DBEN	GP_BA+0x054	R/W	GPIO Port B De-bounce Enable Register	0x0000_0000
GPIOB_IMD	GP_BA+0x058	R/W	GPIO Port B Interrupt Mode Control Register	0x0000_0000
GPIOB_IER	GP_BA+0x05C	R/W	GPIO Port B Interrupt Enable Register	0x0000_0000
GPIOB_ISRC	GP_BA+0x060	R/W	GPIO Port B Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOB_PUEN	GP_BA+0x064	R/W	GPIO Port B Pull-Up Enable Register	0x0000_0000
GPIOC_PMD	GP_BA+0x080	R/W	GPIO Port C Pin I/O Mode Control Register	0x0000_0000
GPIOC_OFFD	GP_BA+0x084	R/W	GPIO Port C Pin OFF Digital Enable Register	0x0000_0000
GPIOC_DOUT	GP_BA+0x088	R/W	GPIO Port C Data Output Value Register	0x0000_FFFF
GPIOC_DMASK	GP_BA+0x08C	R/W	GPIO Port C Data Output Write Mask Register	0x0000_0000
GPIOC_PIN	GP_BA+0x090	R	GPIO Port C Pin Value Register	0x0000_XXXX
GPIOC_DBEN	GP_BA+0x094	R/W	GPIO Port C De-bounce Enable Register	0x0000_0000

GPIOC_IMD	GP_BA+0x098	R/W	GPIO Port C Interrupt Mode Control Register	0x0000_0000
GPIOC_IER	GP_BA+0x09C	R/W	GPIO Port C Interrupt Enable Register	0x0000_0000
GPIOC_ISRC	GP_BA+0x0A0	R/W	GPIO Port C Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOC_PUEN	GP_BA+0x0A4	R/W	GPIO Port C Pull-Up Enable Register	0x0000_0000
GPIOD_PMD	GP_BA+0x0C0	R/W	GPIO Port D Pin I/O Mode Control Register	0x0000_0000
GPIOD_OFFD	GP_BA+0x0C4	R/W	GPIO Port D Pin OFF Digital Enable Register	0x0000_0000
GPIOD_DOUT	GP_BA+0x0C8	R/W	GPIO Port D Data Output Value Register	0x0000_FFFF
GPIOD_DMASK	GP_BA+0x0CC	R/W	GPIO Port D Data Output Write Mask Register	0x0000_0000
GPIOD_PIN	GP_BA+0x0D0	R	GPIO Port D Pin Value Register	0x0000_XXXX
GPIOD_DBEN	GP_BA+0x0D4	R/W	GPIO Port D De-bounce Enable Register	0x0000_0000
GPIOD_IMD	GP_BA+0x0D8	R/W	GPIO Port D Interrupt Mode Control Register	0x0000_0000
GPIOD_IER	GP_BA+0x0DC	R/W	GPIO Port D Interrupt Enable Register	0x0000_0000
GPIOD_ISRC	GP_BA+0x0E0	R/W	GPIO Port D Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOD_PUEN	GP_BA+0x0E4	R/W	GPIO Port D Pull-Up Enable Register	0x0000_0000
GPIOE_PMD	GP_BA+0x100	R/W	GPIO Port E Pin I/O Mode Control Register	0x0000_0000
GPIOE_OFFD	GP_BA+0x104	R/W	GPIO Port E Pin OFF Digital Enable Register	0x0000_0000
GPIOE_DOUT	GP_BA+0x108	R/W	GPIO Port E Data Output Value Register	0x0000_FFFF
GPIOE_DMASK	GP_BA+0x10C	R/W	GPIO Port E Data Output Write Mask Register	0x0000_0000
GPIOE_PIN	GP_BA+0x110	R	GPIO Port E Pin Value Register	0x0000_XXXX
GPIOE_DBEN	GP_BA+0x114	R/W	GPIO Port E De-bounce Enable Register	0x0000_0000
GPIOE_IMD	GP_BA+0x118	R/W	GPIO Port E Interrupt Mode Control Register	0x0000_0000
GPIOE_IER	GP_BA+0x11C	R/W	GPIO Port E Interrupt Enable Register	0x0000_0000
GPIOE_ISRC	GP_BA+0x120	R/W	GPIO Port E Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOE_PUEN	GP_BA+0x124	R/W	GPIO Port E Pull-Up Enable Register	0x0000_0000
GPIOF_PMD	GP_BA+0x140	R/W	GPIO Port F Pin I/O Mode Control Register	0x0000_0000
GPIOF_OFFD	GP_BA+0x144	R/W	GPIO Port F Pin OFF Digital Enable Register	0x0000_0000
GPIOF_DOUT	GP_BA+0x148	R/W	GPIO Port F Data Output Value Register	0x0000_003F
GPIOF_DMASK	GP_BA+0x14C	R/W	GPIO Port F Data Output Write Mask Register	0x0000_0000
GPIOF_PIN	GP_BA+0x150	R	GPIO Port F Pin Value Register	0x0000_00XX
GPIOF_DBEN	GP_BA+0x154	R/W	GPIO Port F De-bounce Enable Register	0x0000_0000
GPIOF_IMD	GP_BA+0x158	R/W	GPIO Port F Interrupt Mode Control Register	0x0000_0000

GPIOF_IER	GP_BA+0x15C	R/W	GPIO Port F Interrupt Enable Register	0x0000_0000
GPIOF_ISRC	GP_BA+0x160	R/W	GPIO Port F Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOF_PUEN	GP_BA+0x164	R/W	GPIO Port F Pull-Up Enable Register	0x0000_0000
DBNCECON	GP_BA+0x180	R/W	De-bounce Cycle Control Register	0x0000_0000
GPIOA0	GP_BA+0x200	R/W	GPIO Port A Bit 0 Data Register	0x0000_000X
GPIOA1	GP_BA+0x204	R/W	GPIO Port A Bit 1 Data Register	0x0000_000X
GPIOA2	GP_BA+0x208	R/W	GPIO Port A Bit 2 Data Register	0x0000_000X
GPIOA3	GP_BA+0x20C	R/W	GPIO Port A Bit 3 Data Register	0x0000_000X
GPIOA4	GP_BA+0x210	R/W	GPIO Port A Bit 4 Data Register	0x0000_000X
GPIOA5	GP_BA+0x214	R/W	GPIO Port A Bit 5 Data Register	0x0000_000X
GPIOA6	GP_BA+0x218	R/W	GPIO Port A Bit 6 Data Register	0x0000_000X
GPIOA7	GP_BA+0x21C	R/W	GPIO Port A Bit 7 Data Register	0x0000_000X
GPIOA8	GP_BA+0x220	R/W	GPIO Port A Bit 8 Data Register	0x0000_000X
GPIOA9	GP_BA+0x224	R/W	GPIO Port A Bit 9 Data Register	0x0000_000X
GPIOA10	GP_BA+0x228	R/W	GPIO Port A Bit 10 Data Register	0x0000_000X
GPIOA11	GP_BA+0x22C	R/W	GPIO Port A Bit 11 Data Register	0x0000_000X
GPIOA12	GP_BA+0x230	R/W	GPIO Port A Bit 12 Data Register	0x0000_000X
GPIOA13	GP_BA+0x234	R/W	GPIO Port A Bit 13 Data Register	0x0000_000X
GPIOA14	GP_BA+0x238	R/W	GPIO Port A Bit 14 Data Register	0x0000_000X
GPIOA15	GP_BA+0x23C	R/W	GPIO Port A Bit 15 Data Register	0x0000_000X
GPIOB0	GP_BA+0x240	R/W	GPIO Port B Bit 0 Data Register	0x0000_000X
GPIOB1	GP_BA+0x244	R/W	GPIO Port B Bit 1 Data Register	0x0000_000X
GPIOB2	GP_BA+0x248	R/W	GPIO Port B Bit 2 Data Register	0x0000_000X
GPIOB3	GP_BA+0x24C	R/W	GPIO Port B Bit 3 Data Register	0x0000_000X
GPIOB4	GP_BA+0x250	R/W	GPIO Port B Bit 4 Data Register	0x0000_000X
GPIOB5	GP_BA+0x254	R/W	GPIO Port B Bit 5 Data Register	0x0000_000X
GPIOB6	GP_BA+0x258	R/W	GPIO Port B Bit 6 Data Register	0x0000_000X
GPIOB7	GP_BA+0x25C	R/W	GPIO Port B Bit 7 Data Register	0x0000_000X
GPIOB8	GP_BA+0x260	R/W	GPIO Port B Bit 8 Data Register	0x0000_000X
GPIOB9	GP_BA+0x264	R/W	GPIO Port B Bit 9 Data Register	0x0000_000X
GPIOB10	GP_BA+0x268	R/W	GPIO Port B Bit 10 Data Register	0x0000_000X

GPIOB11	GP_BA+0x26C	R/W	GPIO Port B Bit 11 Data Register	0x0000_000X
GPIOB12	GP_BA+0x270	R/W	GPIO Port B Bit 12 Data Register	0x0000_000X
GPIOB13	GP_BA+0x274	R/W	GPIO Port B Bit 13 Data Register	0x0000_000X
GPIOB14	GP_BA+0x278	R/W	GPIO Port B Bit 14 Data Register	0x0000_000X
GPIOB15	GP_BA+0x27C	R/W	GPIO Port B Bit 15 Data Register	0x0000_000X
GPIOC0	GP_BA+0x280	R/W	GPIO Port C Bit 0 Data Register	0x0000_000X
GPIOC1	GP_BA+0x284	R/W	GPIO Port C Bit 1 Data Register	0x0000_000X
GPIOC2	GP_BA+0x288	R/W	GPIO Port C Bit 2 Data Register	0x0000_000X
GPIOC3	GP_BA+0x28C	R/W	GPIO Port C Bit 3 Data Register	0x0000_000X
GPIOC4	GP_BA+0x290	R/W	GPIO Port C Bit 4 Data Register	0x0000_000X
GPIOC5	GP_BA+0x294	R/W	GPIO Port C Bit 5 Data Register	0x0000_000X
GPIOC6	GP_BA+0x298	R/W	GPIO Port C Bit 6 Data Register	0x0000_000X
GPIOC7	GP_BA+0x29C	R/W	GPIO Port C Bit 7 Data Register	0x0000_000X
GPIOC8	GP_BA+0x2A0	R/W	GPIO Port C Bit 8 Data Register	0x0000_000X
GPIOC9	GP_BA+0x2A4	R/W	GPIO Port C Bit 9 Data Register	0x0000_000X
GPIOC10	GP_BA+0x2A8	R/W	GPIO Port C Bit 10 Data Register	0x0000_000X
GPIOC11	GP_BA+0x2AC	R/W	GPIO Port C Bit 11 Data Register	0x0000_000X
GPIOC12	GP_BA+0x2B0	R/W	GPIO Port C Bit 12 Data Register	0x0000_000X
GPIOC13	GP_BA+0x2B4	R/W	GPIO Port C Bit 13 Data Register	0x0000_000X
GPIOC14	GP_BA+0x2B8	R/W	GPIO Port C Bit 14 Data Register	0x0000_000X
GPIOC15	GP_BA+0x2BC	R/W	GPIO Port C Bit 15 Data Register	0x0000_000X
GPIOD0	GP_BA+0x2C0	R/W	GPIO Port D Bit 0 Data Register	0x0000_000X
GPIOD1	GP_BA+0x2C4	R/W	GPIO Port D Bit 1 Data Register	0x0000_000X
GPIOD2	GP_BA+0x2C8	R/W	GPIO Port D Bit 2 Data Register	0x0000_000X
GPIOD3	GP_BA+0x2CC	R/W	GPIO Port D Bit 3 Data Register	0x0000_000X
GPIOD4	GP_BA+0x2D0	R/W	GPIO Port D Bit 4 Data Register	0x0000_000X
GPIOD5	GP_BA+0x2D4	R/W	GPIO Port D Bit 5 Data Register	0x0000_000X
GPIOD6	GP_BA+0x2D8	R/W	GPIO Port D Bit 6 Data Register	0x0000_000X
GPIOD7	GP_BA+0x2DC	R/W	GPIO Port D Bit 7 Data Register	0x0000_000X
GPIOD8	GP_BA+0x2E0	R/W	GPIO Port D Bit 8 Data Register	0x0000_000X
GPIOD9	GP_BA+0x2E4	R/W	GPIO Port D Bit 9 Data Register	0x0000_000X

GPIOD10	GP_BA+0x2E8	R/W	GPIO Port D Bit 10 Data Register	0x0000_000X
GPIOD11	GP_BA+0x2EC	R/W	GPIO Port D Bit 11 Data Register	0x0000_000X
GPIOD12	GP_BA+0x2F0	R/W	GPIO Port D Bit 12 Data Register	0x0000_000X
GPIOD13	GP_BA+0x2F4	R/W	GPIO Port D Bit 13 Data Register	0x0000_000X
GPIOD14	GP_BA+0x2F8	R/W	GPIO Port D Bit 14 Data Register	0x0000_000X
GPIOD15	GP_BA+0x2FC	R/W	GPIO Port D Bit 15 Data Register	0x0000_000X
GPIOE0	GP_BA+0x300	R/W	GPIO Port E Bit 0 Data Register	0x0000_000X
GPIOE1	GP_BA+0x304	R/W	GPIO Port E Bit 1 Data Register	0x0000_000X
GPIOE2	GP_BA+0x308	R/W	GPIO Port E Bit 2 Data Register	0x0000_000X
GPIOE3	GP_BA+0x30C	R/W	GPIO Port E Bit 3 Data Register	0x0000_000X
GPIOE4	GP_BA+0x310	R/W	GPIO Port E Bit 4 Data Register	0x0000_000X
GPIOE5	GP_BA+0x314	R/W	GPIO Port E Bit 5 Data Register	0x0000_000X
GPIOE6	GP_BA+0x318	R/W	GPIO Port E Bit 6 Data Register	0x0000_000X
GPIOE7	GP_BA+0x31C	R/W	GPIO Port E Bit 7 Data Register	0x0000_000X
GPIOE8	GP_BA+0x320	R/W	GPIO Port E Bit 8 Data Register	0x0000_000X
GPIOE9	GP_BA+0x324	R/W	GPIO Port E Bit 9 Data Register	0x0000_000X
GPIOE10	GP_BA+0x328	R/W	GPIO Port E Bit 10 Data Register	0x0000_000X
GPIOE11	GP_BA+0x32C	R/W	GPIO Port E Bit 11 Data Register	0x0000_000X
GPIOE12	GP_BA+0x330	R/W	GPIO Port E Bit 12 Data Register	0x0000_000X
GPIOE13	GP_BA+0x334	R/W	GPIO Port E Bit 13 Data Register	0x0000_000X
GPIOE14	GP_BA+0x338	R/W	GPIO Port E Bit 14 Data Register	0x0000_000X
GPIOE15	GP_BA+0x33C	R/W	GPIO Port E Bit 15 Data Register	0x0000_000X
GPIOF0	GP_BA+0x340	R/W	GPIO Port F Bit 0 Data Register	0x0000_000X
GPIOF1	GP_BA+0x344	R/W	GPIO Port F Bit 1 Data Register	0x0000_000X
GPIOF2	GP_BA+0x348	R/W	GPIO Port F Bit 2 Data Register	0x0000_000X
GPIOF3	GP_BA+0x34C	R/W	GPIO Port F Bit 3 Data Register	0x0000_000X
GPIOF4	GP_BA+0x350	R/W	GPIO Port F Bit 4 Data Register	0x0000_000X
GPIOF5	GP_BA+0x354	R/W	GPIO Port F Bit 5 Data Register	0x0000_000X

5.7.6 Register Description

GPIO Port [A/B/C/D/E/F] Pin I/O Mode Control Register (GPIOx_PMD)

Register	Offset	R/W	Description	Reset Value
GPIOA_PMD	GP_BA+0x000	R/W	GPIO Port A Pin I/O Mode Control Register	0x0000_0000
GPIOB_PMD	GP_BA+0x040	R/W	GPIO Port B Pin I/O Mode Control Register	0x0000_0000
GPIOC_PMD	GP_BA+0x080	R/W	GPIO Port C Pin I/O Mode Control Register	0x0000_0000
GPIOD_PMD	GP_BA+0x0C0	R/W	GPIO Port D Pin I/O Mode Control Register	0x0000_0000
GPIOE_PMD	GP_BA+0x100	R/W	GPIO Port E Pin I/O Mode Control Register	0x0000_0000
GPIOF_PMD	GP_BA+0x140	R/W	GPIO Port F Pin I/O Mode Control Register	0x0000_0000

31	30	29	28	27	26	25	24
PMD15		PMD14		PMD13		PMD12	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PMD11		PMD10		PMD9		PMD8	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PMD7		PMD6		PMD5		PMD4	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PMD3		PMD2		PMD1		PMD0	

Bits	Description	
[2n+1:2n] n = 0,1..15	**PMDn**	**GPIO Port [x] Pin [n] Mode Control** Determine the I/O type of GPIO port [x] pin [n] 00 = GPIO port [x] pin [n] is in INPUT mode. 01 = GPIO port [x] pin [n] is in OUTPUT mode. 10 = GPIO port [x] pin [n] is in Open-Drain mode. 11 = Reserved. **Note:** For GPIOF_PMD, PMD6 ~ PMD15 are reserved.

GPIO Port [A/B/C/D/E/F] Pin OFF Digital Enable Resistor (GPIOx_OFFD)

Register	Offset	R/W	Description	Reset Value
GPIOA_OFFD	GP_BA+0x004	R/W	GPIO Port A Pin OFF Digital Enable Register	0x0000_0000
GPIOB_OFFD	GP_BA+0x044	R/W	GPIO Port B Pin OFF Digital Enable Register	0x0000_0000
GPIOC_OFFD	GP_BA+0x084	R/W	GPIO Port C Pin OFF Digital Enable Register	0x0000_0000
GPIOD_OFFD	GP_BA+0x0C4	R/W	GPIO Port D Pin OFF Digital Enable Register	0x0000_0000
GPIOE_OFFD	GP_BA+0x104	R/W	GPIO Port E Pin OFF Digital Enable Register	0x0000_0000
GPIOF_OFFD	GP_BA+0x144	R/W	GPIO Port F Pin OFF Digital Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
OFFD							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
OFFD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[n+16] n = 0,1..15	**OFFD**	**GPIO Port [x] Pin [n] Digital Input Path Disable** Determine if the digital input path of GPIO port [x] pin [n] is disabled. 0 = Digital input path of GPIO port [x] pin [n] Enabled. 1 = Digital input path of GPIO port [x] pin [n] Disabled (tied digital input to low) **Note:** For GPIOF_OFFD, bits [31:22] are reserved.
[15:0]	**Reserved**	**Reserved**

GPIO Port [A/B/C/D/E/F] Data Output Value Register (GPIOx_DOUT)

Register	Offset	R/W	Description	Reset Value
GPIOA_DOUT	GP_BA+0x008	R/W	GPIO Port A Data Output Value Register	0x0000_FFFF
GPIOB_DOUT	GP_BA+0x048	R/W	GPIO Port B Data Output Value Register	0x0000_FFFF
GPIOC_DOUT	GP_BA+0x088	R/W	GPIO Port C Data Output Value Register	0x0000_FFFF
GPIOD_DOUT	GP_BA+0x0C8	R/W	GPIO Port D Data Output Value Register	0x0000_FFFF
GPIOE_DOUT	GP_BA+0x108	R/W	GPIO Port E Data Output Value Register	0x0000_FFFF
GPIOF_DOUT	GP_BA+0x148	R/W	GPIO Port F Data Output Value Register	0x0000_003F

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DOUT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DOUT							

Bits	Description	
[31:16]	**Reserved**	**Reserved**
[n] n = 0,1..15	**DOUT**	**GPIO Port [x] Pin [n] Output Value** Each of these bits controls the status of a GPIO port [x] pin [n] when the GPI/O pin is configures as output or open-drain mode 0 = GPIO port [x] Pin [n] will drive Low if the corresponding output mode enabling bit is set. 1 = GPIO port [x] Pin [n] will drive High if the corresponding output mode enabling bit is set. **Note:** For GPIOF_DOUT, bits [15:6] are reserved.

GPIO Port [A/B/C/D/E/F] Data Output Write Mask Register (GPIOx_DMASK)

Register	Offset	R/W	Description	Reset Value
GPIOA_DMASK	GP_BA+0x00C	R/W	GPIO Port A Data Output Write Mask Register	0x0000_0000
GPIOB_DMASK	GP_BA+0x04C	R/W	GPIO Port B Data Output Write Mask Register	0x0000_0000
GPIOC_DMASK	GP_BA+0x08C	R/W	GPIO Port C Data Output Write Mask Register	0x0000_0000
GPIOD_DMASK	GP_BA+0x0CC	R/W	GPIO Port D Data Output Write Mask Register	0x0000_0000
GPIOE_DMASK	GP_BA+0x10C	R/W	GPIO Port E Data Output Write Mask Register	0x0000_0000
GPIOF_DMASK	GP_BA+0x14C	R/W	GPIO Port F Data Output Write Mask Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DMASK							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMASK							

Bits	Description	
[31:16]	**Reserved**	**Reserved**
[n] n = 0,1..15	**DMASK**	**GPIO Port [x] Pin [n] Data Output Write Mask** These bits are used to protect the corresponding register of GPIOx_DOUT bit [n]. When set the DMASK[n] to “1”, the corresponding DOUT[n] bit is protected. The write signal is masked, write data to the protect bit is ignored 0 = The corresponding GPIO_DOUT bit [n] can be updated 1 = The corresponding GPIO_DOUT bit [n] is protected **Note:** For GPIOF_DMASK, bits [15:6] are reserved. **Note:** These mask bits only take effect while CPU is doing write operation to register GPIOx_DOUT. If CPU is doing write operation to register GPIO[x][n], these mask bits will not take effect.

GPIO Port [A/B/C/D/E/F] Pin Value Register (GPIOx_PIN)

Register	Offset	R/W	Description	Reset Value
GPIOA_PIN	GP_BA+0x010	R	GPIO Port A Pin Value Register	0x0000_XXXX
GPIOB_PIN	GP_BA+0x050	R	GPIO Port B Pin Value Register	0x0000_XXXX
GPIOC_PIN	GP_BA+0x090	R	GPIO Port C Pin Value Register	0x0000_XXXX
GPIOD_PIN	GP_BA+0x0D0	R	GPIO Port D Pin Value Register	0x0000_XXXX
GPIOE_PIN	GP_BA+0x110	R	GPIO Port E Pin Value Register	0x0000_XXXX
GPIOF_PIN	GP_BA+0x150	R	GPIO Port F Pin Value Register	0x0000_00XX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PIN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PIN							

Bits	Description	
[31:16]	**Reserved**	**Reserved**
[n] n = 0,1..15	**PIN**	**GPIO Port [x] Pin [n] Value** The value read from each of these bit reflects the actual status of the respective GPI/O pin **Note:** For GPIOF_PIN, bits [15:6] are reserved.

GPIO Port [A/B/C/D/E/F] De-bounce Enable Register (GPIOx_DBEN)

Register	Offset	R/W	Description	Reset Value
GPIOA_DBEN	GP_BA+0x014	R/W	GPIO Port A De-bounce Enable Register	0x0000_0000
GPIOB_DBEN	GP_BA+0x054	R/W	GPIO Port B De-bounce Enable Register	0x0000_0000
GPIOC_DBEN	GP_BA+0x094	R/W	GPIO Port C De-bounce Enable Register	0x0000_0000
GPIOD_DBEN	GP_BA+0x0D4	R/W	GPIO Port D De-bounce Enable Register	0x0000_0000
GPIOE_DBEN	GP_BA+0x114	R/W	GPIO Port E De-bounce Enable Register	0x0000_0000
GPIOF_DBEN	GP_BA+0x154	R/W	GPIO Port F De-bounce Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DBEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DBEN							

Bits	Description	
[31:16]	**Reserved**	**Reserved**
[n] n = 0,1..15	**DBEN**	**GPIO Port [x] Pin [n] Input Signal De-bounce Enable** DBEN[n] used to enable the de-bounce function for each corresponding bit. If the input signal pulse width cannot be sampled by continuous two de-bounce sample cycle the input signal transition is seen as the signal bounce and will not trigger the interrupt. DBEN[n] is used for “edge-trigger” interrupt only, and ignored for “level trigger” interrupt 0 = The GPIO port [x] Pin [n] input signal de-bounce function is disabled 1 = The GPIO port [x] Pin [n] input signal de-bounce function is enabled The de-bounce function is valid for edge triggered interrupt. If the interrupt mode is level triggered, the de-bounce enable bit is ignored. **Note:** For GPIOF_DBEN, bits [15:6] are reserved.

GPIO Port [A/B/C/D/E/F] Interrupt Mode Control Register (GPIOx_IMD)

Register	Offset	R/W	Description	Reset Value
GPIOA_IMD	GP_BA+0x018	R/W	GPIO Port A Interrupt Mode Control Register	0x0000_0000
GPIOB_IMD	GP_BA+0x058	R/W	GPIO Port B Interrupt Mode Control Register	0x0000_0000
GPIOC_IMD	GP_BA+0x098	R/W	GPIO Port C Interrupt Mode Control Register	0x0000_0000
GPIOD_IMD	GP_BA+0x0D8	R/W	GPIO Port D Interrupt Mode Control Register	0x0000_0000
GPIOE_IMD	GP_BA+0x118	R/W	GPIO Port E Interrupt Mode Control Register	0x0000_0000
GPIOF_IMD	GP_BA+0x158	R/W	GPIO Port F Interrupt Mode Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
IMD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
IMD							

Bits	Description	
[31:16]	**Reserved**	**Reserved**
[n] n = 0,1..15	**IMD**	**GPIO Port [x] Pin [n] Edge or Level Detection Interrupt Control** IMD[n] used to control the interrupt is by level trigger or by edge trigger. If the interrupt is by edge trigger, the trigger source is control de-bounce. If the interrupt is by level trigger, the input source is sampled by one clock and the generate the interrupt 0 = Edge trigger interrupt 1 = Level trigger interrupt If set pin as the level trigger interrupt, then only one level can be set on the registers GPIOX_IER. If set both the level to trigger interrupt, the setting is ignored and no interrupt will occur The de-bounce function is valid for edge triggered interrupt. If the interrupt mode is level triggered, the de-bounce enable bit is ignored. **Note:** For GPIOF_IMD, bits [15:6] are reserved.

GPIO Port [A/B/C/D/E/F] Interrupt Enable Register (GPIOx_IER)

Register	Offset	R/W	Description	Reset Value
GPIOA_IER	GP_BA+0x01C	R/W	GPIO Port A Interrupt Enable Register	0x0000_0000
GPIOB_IER	GP_BA+0x05C	R/W	GPIO Port B Interrupt Enable Register	0x0000_0000
GPIOC_IER	GP_BA+0x09C	R/W	GPIO Port C Interrupt Enable Register	0x0000_0000
GPIOD_IER	GP_BA+0x0DC	R/W	GPIO Port D Interrupt Enable Register	0x0000_0000
GPIOE_IER	GP_BA+0x11C	R/W	GPIO Port E Interrupt Enable Register	0x0000_0000
GPIOF_IER	GP_BA+0x15C	R/W	GPIO Port F Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
RIER							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RIER							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
FIER							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FIER							

Bits	Description	
[n+16] n = 0,1..15	**RIER**	**GPIO Port [x] Pin [n] Interrupt Enable by Input Rising Edge or Input Level High** RIER[x] used to enable the interrupt for each of the corresponding input GPIO_PIN[x]. Set bit “1” also enable the pin wake-up function When set the RIER[x] bit “1”: If the interrupt is level mode trigger, the input PIN[x] state at level “high” will generate the interrupt. If the interrupt is edge mode trigger, the input PIN[x] state change from “low-to-high” will generate the interrupt. 1 = PIN[x] level-high or low-to-high interrupt Enabled 0 = PIN[x] level-high or low-to-high interrupt Disabled **Note:** For GPIOF_IE, bits [31:22] are reserved.

Bits	Description	
[n] n = 0,1..15	**FIER**	**GPIO Port [x] Pin [n] Interrupt Enable by Input Falling Edge or Input Level Low** FIER[n] used to enable the interrupt for each of the corresponding input GPIO_PIN[n]. Set bit "1" also enable the pin wake-up function When set the FIER[n] bit "1": If the interrupt is level mode trigger, the input PIN[n] state at level "low" will generate the interrupt. If the interrupt is edge mode trigger, the input PIN[n] state change from "high-to-low" will generate the interrupt. 1 = PIN[n] state low-level or high-to-low change interrupt Enabled 0 = PIN[n] state low-level or high-to-low change interrupt Disabled **Note:** For GPIOF_IER, bits [15:6] are reserved.

GPIO Port [A/B/C/D/E/F] Interrupt Trigger Source Status Register (GPIOx_ISRC)

Register	Offset	R/W	Description	Reset Value
GPIOA_ISRC	GP_BA+0x020	R/W	GPIO Port A Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOB_ISRC	GP_BA+0x060	R/W	GPIO Port B Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOC_ISRC	GP_BA+0x0A0	R/W	GPIO Port C Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOD_ISRC	GP_BA+0x0E0	R/W	GPIO Port D Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOE_ISRC	GP_BA+0x120	R/W	GPIO Port E Interrupt Trigger Source Status Register	0xXXXX_XXXX
GPIOF_ISRC	GP_BA+0x160	R/W	GPIO Port F Interrupt Trigger Source Status Register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ISRC							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ISRC							

Bits	Description	
[31:16]	**Reserved**	**Reserved**
[n] n = 0,1..15	**ISRC**	**GPIO Port [x] Pin [n] Interrupt Trigger Source Indicator** Read : 1 = Port x[n] generate an interrupt 0 = No interrupt at Port x[n] Write: 1 = Clear the correspond pending interrupt. 0 = No action. **Note:** For GPIOF_ISRC, bits [15:6] are reserved.

GPIO Port [A/B/C/D/E/F] Pull-up Enable Register (GPIOx_PUEN)

Register	Offset	R/W	Description	Reset Value
GPIOA_PUEN	GP_BA+0x024	R/W	GPIO Port A Pull-Up Enable Register	0x0000_0000
GPIOB_PUEN	GP_BA+0x064	R/W	GPIO Port B Pull-Up Enable Register	0x0000_0000
GPIOC_PUEN	GP_BA+0x0A4	R/W	GPIO Port C Pull-Up Enable Register	0x0000_0000
GPIOD_PUEN	GP_BA+0x0E4	R/W	GPIO Port D Pull-Up Enable Register	0x0000_0000
GPIOE_PUEN	GP_BA+0x124	R/W	GPIO Port E Pull-Up Enable Register	0x0000_0000
GPIOF_PUEN	GP_BA+0x164	R/W	GPIO Port F Pull-Up Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PUEN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PUEN							

Bits	Description	
[31:16]	**Reserved**	**Reserved**
[n] n = 0,1..15	**PUEN**	**GPIO Port [x] Pin [n] Pull-Up Enable Register** Read : 1 = GPIO port [A/B/C/D/E/F] bit [n] pull-up resistor Enabled. 0 = GPIO port [A/B/C/D/E/F] bit [n] pull-up resistor Disabled. **Note:** For GPIOF_PUEN, bits [15:6] are reserved.

De-bounce Cycle Control Register (DBNCECON)

Register	Offset	R/W	Description	Reset Value
DBNCECON	GP_BA+0x180	R/W	De-bounce Cycle Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		DBCLK_ON	DBCLKSRC	DBCLKSEL			

Bits	Description	
[31:6]	**Reserved**	**Reserved**
[5]	**DBCLK_ON**	**De-bounce Clock Enable** This bit controls if the de-bounce clock is enabled. However, if GPI/O pin's interrupt is enabled, the de-bounce clock will be enabled automatically no matter what the DBCLK_ON value is. If CPU is in sleep mode, this bit didn't take effect. And only the GPI/O pin with interrupt enable could get de-bounce clock. 1 = De-bounce clock Enabled. 0 = De-bounce clock Disabled.
[4]	**DBCLKSRC**	**De-bounce Counter Clock Source Selection** 1 = De-bounce counter Clock Source is the internal 10 kHz clock 0 = De-bounce counter Clock Source is the HCLK

<table>
<tr><th>Bits</th><th colspan="3">Description</th></tr>
<tr><td rowspan="18">[n]</td><td rowspan="18">PUEN</td><td colspan="2">De-bounce Sampling Cycle Selection</td></tr>
<tr><td>DBCLKSEL</td><td>Description</td></tr>
<tr><td>0x0</td><td>Sample interrupt input once per 1 clock</td></tr>
<tr><td>0x1</td><td>Sample interrupt input once per 2 clocks</td></tr>
<tr><td>0x2</td><td>Sample interrupt input once per 4 clocks</td></tr>
<tr><td>0x3</td><td>Sample interrupt input once per 8 clocks</td></tr>
<tr><td>0x4</td><td>Sample interrupt input once per 16 clocks</td></tr>
<tr><td>0x5</td><td>Sample interrupt input once per 32 clocks</td></tr>
<tr><td>0x6</td><td>Sample interrupt input once per 64 clocks</td></tr>
<tr><td>0x7</td><td>Sample interrupt input once per 128 clocks</td></tr>
<tr><td>0x8</td><td>Sample interrupt input once per 256 clocks</td></tr>
<tr><td>0x9</td><td>Sample interrupt input once per 2*256 clocks</td></tr>
<tr><td>0xA</td><td>Sample interrupt input once per 4*256clocks</td></tr>
<tr><td>0xB</td><td>Sample interrupt input once per 8*256 clocks</td></tr>
<tr><td>0xC</td><td>Sample interrupt input once per 16*256 clocks</td></tr>
<tr><td>0xD</td><td>Sample interrupt input once per 32*256 clocks</td></tr>
<tr><td>0xE</td><td>Sample interrupt input once per 64*256 clocks</td></tr>
<tr><td>0xF</td><td>Sample interrupt input once per 128*256 clocks</td></tr>
</table>

GPIO Port [A/B/C/D/E/F] Bit [n] Data Register (GPIO[A/B/C/D/E/F][n])

Register	Offset	R/W	Description	Reset Value
GPIOA0	GP_BA+0x200	R/W	GPIO Port A Bit 0 Data Register	0x0000_000X
GPIOA1	GP_BA+0x204	R/W	GPIO Port A Bit 1 Data Register	0x0000_000X
GPIOA2	GP_BA+0x208	R/W	GPIO Port A Bit 2 Data Register	0x0000_000X
GPIOA3	GP_BA+0x20C	R/W	GPIO Port A Bit 3 Data Register	0x0000_000X
GPIOA4	GP_BA+0x210	R/W	GPIO Port A Bit 4 Data Register	0x0000_000X
GPIOA5	GP_BA+0x214	R/W	GPIO Port A Bit 5 Data Register	0x0000_000X
GPIOA6	GP_BA+0x218	R/W	GPIO Port A Bit 6 Data Register	0x0000_000X
GPIOA7	GP_BA+0x21C	R/W	GPIO Port A Bit 7 Data Register	0x0000_000X
GPIOA8	GP_BA+0x220	R/W	GPIO Port A Bit 8 Data Register	0x0000_000X
GPIOA9	GP_BA+0x224	R/W	GPIO Port A Bit 9 Data Register	0x0000_000X
GPIOA10	GP_BA+0x228	R/W	GPIO Port A Bit 10 Data Register	0x0000_000X
GPIOA11	GP_BA+0x22C	R/W	GPIO Port A Bit 11 Data Register	0x0000_000X
GPIOA12	GP_BA+0x230	R/W	GPIO Port A Bit 12 Data Register	0x0000_000X
GPIOA13	GP_BA+0x234	R/W	GPIO Port A Bit 13 Data Register	0x0000_000X
GPIOA14	GP_BA+0x238	R/W	GPIO Port A Bit 14 Data Register	0x0000_000X
GPIOA15	GP_BA+0x23C	R/W	GPIO Port A Bit 15 Data Register	0x0000_000X
GPIOB0	GP_BA+0x240	R/W	GPIO Port B Bit 0 Data Register	0x0000_000X
GPIOB1	GP_BA+0x244	R/W	GPIO Port B Bit 1 Data Register	0x0000_000X
GPIOB2	GP_BA+0x248	R/W	GPIO Port B Bit 2 Data Register	0x0000_000X
GPIOB3	GP_BA+0x24C	R/W	GPIO Port B Bit 3 Data Register	0x0000_000X
GPIOB4	GP_BA+0x250	R/W	GPIO Port B Bit 4 Data Register	0x0000_000X
GPIOB5	GP_BA+0x254	R/W	GPIO Port B Bit 5 Data Register	0x0000_000X
GPIOB6	GP_BA+0x258	R/W	GPIO Port B Bit 6 Data Register	0x0000_000X
GPIOB7	GP_BA+0x25C	R/W	GPIO Port B Bit 7 Data Register	0x0000_000X
GPIOB8	GP_BA+0x260	R/W	GPIO Port B Bit 8 Data Register	0x0000_000X
GPIOB9	GP_BA+0x264	R/W	GPIO Port B Bit 9 Data Register	0x0000_000X
GPIOB10	GP_BA+0x268	R/W	GPIO Port B Bit 10 Data Register	0x0000_000X
GPIOB11	GP_BA+0x26C	R/W	GPIO Port B Bit 11 Data Register	0x0000_000X
GPIOB12	GP_BA+0x270	R/W	GPIO Port B Bit 12 Data Register	0x0000_000X

GPIOB13	GP_BA+0x274	R/W	GPIO Port B Bit 13 Data Register	0x0000_000X
GPIOB14	GP_BA+0x278	R/W	GPIO Port B Bit 14 Data Register	0x0000_000X
GPIOB15	GP_BA+0x27C	R/W	GPIO Port B Bit 15 Data Register	0x0000_000X
GPIOC0	GP_BA+0x280	R/W	GPIO Port C Bit 0 Data Register	0x0000_000X
GPIOC1	GP_BA+0x284	R/W	GPIO Port C Bit 1 Data Register	0x0000_000X
GPIOC2	GP_BA+0x288	R/W	GPIO Port C Bit 2 Data Register	0x0000_000X
GPIOC3	GP_BA+0x28C	R/W	GPIO Port C Bit 3 Data Register	0x0000_000X
GPIOC4	GP_BA+0x290	R/W	GPIO Port C Bit 4 Data Register	0x0000_000X
GPIOC5	GP_BA+0x294	R/W	GPIO Port C Bit 5 Data Register	0x0000_000X
GPIOC6	GP_BA+0x298	R/W	GPIO Port C Bit 6 Data Register	0x0000_000X
GPIOC7	GP_BA+0x29C	R/W	GPIO Port C Bit 7 Data Register	0x0000_000X
GPIOC8	GP_BA+0x2A0	R/W	GPIO Port C Bit 8 Data Register	0x0000_000X
GPIOC9	GP_BA+0x2A4	R/W	GPIO Port C Bit 9 Data Register	0x0000_000X
GPIOC10	GP_BA+0x2A8	R/W	GPIO Port C Bit 10 Data Register	0x0000_000X
GPIOC11	GP_BA+0x2AC	R/W	GPIO Port C Bit 11 Data Register	0x0000_000X
GPIOC12	GP_BA+0x2B0	R/W	GPIO Port C Bit 12 Data Register	0x0000_000X
GPIOC13	GP_BA+0x2B4	R/W	GPIO Port C Bit 13 Data Register	0x0000_000X
GPIOC14	GP_BA+0x2B8	R/W	GPIO Port C Bit 14 Data Register	0x0000_000X
GPIOC15	GP_BA+0x2BC	R/W	GPIO Port C Bit 15 Data Register	0x0000_000X
GPIOD0	GP_BA+0x2C0	R/W	GPIO Port D Bit 0 Data Register	0x0000_000X
GPIOD1	GP_BA+0x2C4	R/W	GPIO Port D Bit 1 Data Register	0x0000_000X
GPIOD2	GP_BA+0x2C8	R/W	GPIO Port D Bit 2 Data Register	0x0000_000X
GPIOD3	GP_BA+0x2CC	R/W	GPIO Port D Bit 3 Data Register	0x0000_000X
GPIOD4	GP_BA+0x2D0	R/W	GPIO Port D Bit 4 Data Register	0x0000_000X
GPIOD5	GP_BA+0x2D4	R/W	GPIO Port D Bit 5 Data Register	0x0000_000X
GPIOD6	GP_BA+0x2D8	R/W	GPIO Port D Bit 6 Data Register	0x0000_000X
GPIOD7	GP_BA+0x2DC	R/W	GPIO Port D Bit 7 Data Register	0x0000_000X
GPIOD8	GP_BA+0x2E0	R/W	GPIO Port D Bit 8 Data Register	0x0000_000X
GPIOD9	GP_BA+0x2E4	R/W	GPIO Port D Bit 9 Data Register	0x0000_000X
GPIOD10	GP_BA+0x2E8	R/W	GPIO Port D Bit 10 Data Register	0x0000_000X
GPIOD11	GP_BA+0x2EC	R/W	GPIO Port D Bit 11 Data Register	0x0000_000X

GPIOD12	GP_BA+0x2F0	R/W	GPIO Port D Bit 12 Data Register	0x0000_000X
GPIOD13	GP_BA+0x2F4	R/W	GPIO Port D Bit 13 Data Register	0x0000_000X
GPIOD14	GP_BA+0x2F8	R/W	GPIO Port D Bit 14 Data Register	0x0000_000X
GPIOD15	GP_BA+0x2FC	R/W	GPIO Port D Bit 15 Data Register	0x0000_000X
GPIOE0	GP_BA+0x300	R/W	GPIO Port E Bit 0 Data Register	0x0000_000X
GPIOE1	GP_BA+0x304	R/W	GPIO Port E Bit 1 Data Register	0x0000_000X
GPIOE2	GP_BA+0x308	R/W	GPIO Port E Bit 2 Data Register	0x0000_000X
GPIOE3	GP_BA+0x30C	R/W	GPIO Port E Bit 3 Data Register	0x0000_000X
GPIOE4	GP_BA+0x310	R/W	GPIO Port E Bit 4 Data Register	0x0000_000X
GPIOE5	GP_BA+0x314	R/W	GPIO Port E Bit 5 Data Register	0x0000_000X
GPIOE6	GP_BA+0x318	R/W	GPIO Port E Bit 6 Data Register	0x0000_000X
GPIOE7	GP_BA+0x31C	R/W	GPIO Port E Bit 7 Data Register	0x0000_000X
GPIOE8	GP_BA+0x320	R/W	GPIO Port E Bit 8 Data Register	0x0000_000X
GPIOE9	GP_BA+0x324	R/W	GPIO Port E Bit 9 Data Register	0x0000_000X
GPIOE10	GP_BA+0x328	R/W	GPIO Port E Bit 10 Data Register	0x0000_000X
GPIOE11	GP_BA+0x32C	R/W	GPIO Port E Bit 11 Data Register	0x0000_000X
GPIOE12	GP_BA+0x330	R/W	GPIO Port E Bit 12 Data Register	0x0000_000X
GPIOE13	GP_BA+0x334	R/W	GPIO Port E Bit 13 Data Register	0x0000_000X
GPIOE14	GP_BA+0x338	R/W	GPIO Port E Bit 14 Data Register	0x0000_000X
GPIOE15	GP_BA+0x33C	R/W	GPIO Port E Bit 15 Data Register	0x0000_000X
GPIOF0	GP_BA+0x340	R/W	GPIO Port F Bit 0 Data Register	0x0000_000X
GPIOF1	GP_BA+0x344	R/W	GPIO Port F Bit 1 Data Register	0x0000_000X
GPIOF2	GP_BA+0x348	R/W	GPIO Port F Bit 2 Data Register	0x0000_000X
GPIOF3	GP_BA+0x34C	R/W	GPIO Port F Bit 3 Data Register	0x0000_000X
GPIOF4	GP_BA+0x350	R/W	GPIO Port F Bit 4 Data Register	0x0000_000X
GPIOF5	GP_BA+0x354	R/W	GPIO Port F Bit 5 Data Register	0x0000_000X

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							GPIO[x][n]

Bits	Description	
[31:1]	**Reserved**	**Reserved**
[0]	**GPIO[x][n]**	**GPIO Port [x] Pin [n] I/O Data** This field supports the bit operation mode on related GPIO port [x] pin [n]. Writing this filed to set the corresponding GPIO port [x] pin [n] output value while reading this field to get the corresponding GPIO port [x] pin [n] value. Read: 1 = The corresponding GPIO port [x] pin [n] value is high. 0 = The corresponding GPIO port [x] pin [n] value is low. Write: 1 = Set corresponding GPIO port [x] pin [n] to high. 0 = Set corresponding GPIO port [x] pin [n] to low. **Note:** The write operation will not be affected by register GPIOx_DMASK.

5.8 DMA Controller

5.8.1 Overview

The DMA controller contains six channel peripheral direct memory access (PDMA) controllers, a video direct memory access (VDMA) controller and a cyclic redundancy check (CRC) generator. The PDMA controller can transfer data to and from memory or transfer data to and from APB devices. The DMA has eight channels of DMA including one channel VDMA (Memory-to-Memory) and six channels PDMA (Peripheral-to-Memory or Memory-to-Peripheral or Memory-to-Memory) and a CRC controller. For channel0 VDMA, it supports block transfer from memory to memory. For PDMA channel (DMA CH1~CH6), there is one-word buffer as transfer buffer between the Peripherals APB devices and Memory. And for channel 0 VDMA, there is a two-word buffer.

Software can stop the DMA operation by disable PDMA [PDMACEN]/VDMA [VDMACEN]. Software can recognize the completion of a DMA operation by software polling or when it receives an internal DMA interrupt. The DMA controller can increase source or destination address, fixed or wrap around them as well.

The DMA controller also contains a cyclic redundancy check (CRC) generator that can perform CRC calculation with programmable polynomial settings. The CRC engine support CPU PIO mode and DMA transfer mode.

5.8.2 Features

Seven DMA channels and a CRC generator: 1 VDMA channel and 6 PDMA channels. Each channel can support a unidirectional transfer.

AMBA AHB master/slave interface compatible, for data transfer and register read/write.

Hardware round robin priority scheme.

- VDMA
  - Memory-to-memory transfer
  - Supports block transfer with stride
  - Supports word/half-word/byte boundary address
  - Supports address direction: increment and decrement
- PDMA
  - Peripheral-to-memory, memory-to-peripheral, and memory-to-memory transfer
  - Supports word boundary address
  - Supports word alignment transfer length in memory-to-memory mode
  - Supports word/half-word/byte alignment transfer length in peripheral-to-memory and memory-to-peripheral mode
  - Supports word/half-word/byte transfer data width from/to peripheral
  - Supports address direction: increment, fixed, and wrap around
- Cyclic Redundancy Check (CRC)
  - Supports four common polynomials CRC-CCITT, CRC-8, CRC-16, and CRC-32
    - CRC-CCITT: $X^{16} + X^{12} + X^{5} + 1$
    - CRC-8: $X^{8} + X^{2} + X + 1$
    - CRC-16: $X^{16} + X^{15} + X^{2} + 1$

- CRC-32: $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$
- Programmable seed value
- Supports programmable order reverse setting for input data and CRC checksum
- Supports programmable 1's complement setting for input data and CRC checksum
- Supports CPU PIO mode or DMA transfer mode
- Supports 8/16/32-bit of data width in CPU PIO mode
  - 8-bit write mode: 1-AHB clock cycle operation
  - 16-bit write mode: 2-AHB clock cycle operation
  - 32-bit write mode: 4-AHB clock cycle operation
- Supports byte alignment transfer length in CRC DMA mode

5.8.3 Block Diagram

The DMA clock control and block diagram are shown as follows.

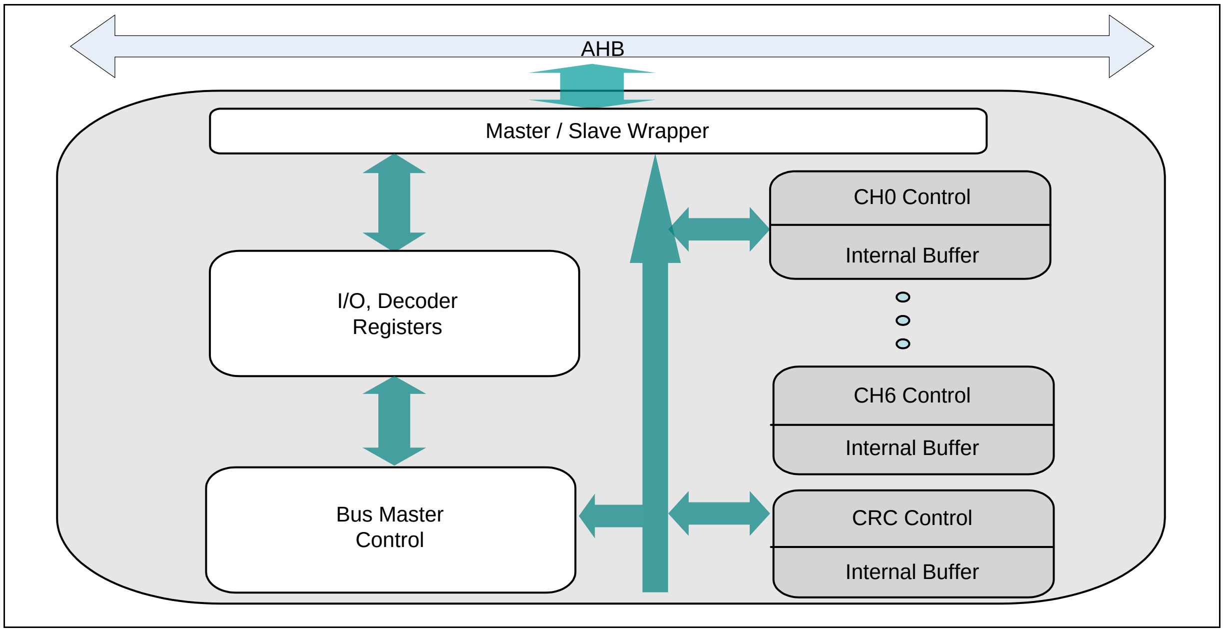


Figure 5.8-1 DMA Controller Block Diagram

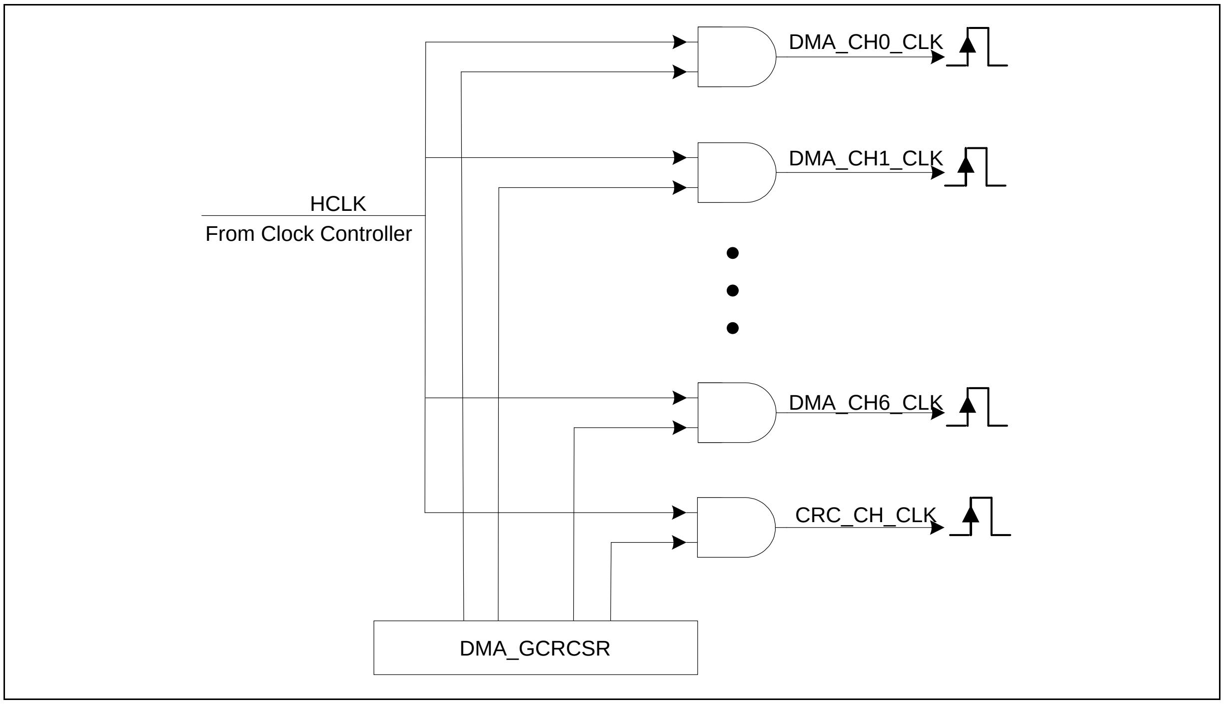


Figure 5.8-2 DMA Clock Control Diagram

NANO100 SERIES TECHNICAL REFERENCE MANUAL

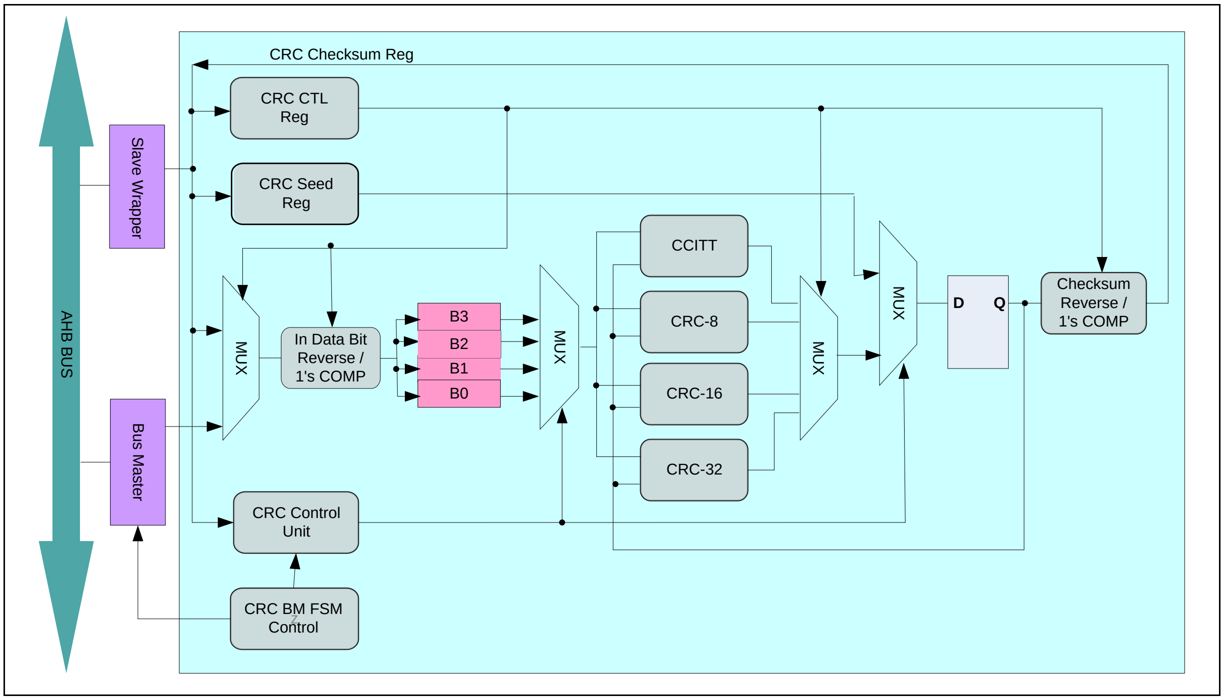


Figure 5.8-3 CRC Generator Block Diagram

5.8.4 Functional Description

The direct memory access (DMA) controller module transfers data from one address to another address, without CPU intervention. The DMA controller contains eight channels that including one channel VDMA (Memory-to-Memory) and six channels PDMA (Peripheral-to-Memory or Memory-to-Peripheral or Memory-to-Memory) and one CRC channel.

The CPU can recognize the completion of a DMA operation by software polling or when it receives an internal DMA interrupt. As to the source and destination address, the DMA controller has three different modes: increased, fixed and wrap around operation mode.

5.8.4.1 VDMA

The DMA controller contains one channel VDMA (Memory-to-Memory). The VDMA module can transfers data from one address to another address and can support block transfer with stride. When operating in VDMA mode, the transfer address can incremented successively or decremented successively.

In Without Block Transfer mode (VDMA_CSR [STRIDE_EN] Disabled), software must enable DMA channel VDMA [VDMACEN] and then write a valid source address to the VDMA_SAR register, a destination address to the VDMA_DAR register, and a transfer count to the VDMA_BCR register. Next, trigger the VDMA_CSR [TRIG_EN]. The DMA will continue the transfer until VDMA_CBCR comes down to zero

In Block Transfer mode (VDMA_CSR [STRIDE_EN] Enabled), software must enable DMA channel VDMA [VDMACEN] and then write a valid source address to the VDMA_SAR register and a source address offset count to VDMA_SASOCR [SASTOBL] register, a destination address to the VDMA_DAR register and a destination address offset count to VDMA_DASOCR [DASTOBL], and a transfer count to the VDMA_BCR register and a block byte count to VDMA_SASOCR [STBC]. Next, trigger the VDMA_CSR [TRIG_EN]. The DMA will continue the transfer until VDMA_CBCR comes down to zero. The following figure shows the block transfer relationship between source memory and destination memory.

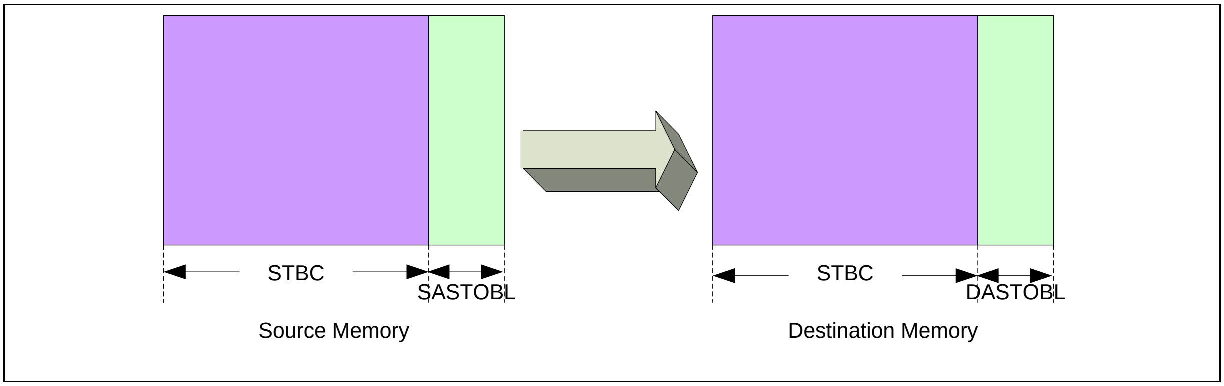


Figure 5.8-4 VDMA Block Transfer

If an error occurs during the VDMA operation, the channel stops unless software clears the error condition and sets the VDMA_CSR [SW_RST] to reset the VDMA channel and set VDMA_CSR [VDMACEN] and [TRIG_EN] bits field to start again.

5.8.4.2 PDMA

The DMA controller contains six channels PDMA (Peripheral-to-Memory or Memory-to-Peripheral or Memory-to-Memory). As to the source and destination address, the DMA controller has three different modes: increased, fixed and wrap around operation mode.

Every PDMA channel behavior is not pre-defined, so users must configure the channel service settings of DMA_DSSR0 and DMA_DSSR1 before start the related PDMA channel.

Software must enable DMA channel PDMA [PDMACEN] and then write a valid source address to the PDMA_SARx register, a destination address to the PDMA_DARx register, and a transfer count to the PDMA_BCRx register. Next, trigger the DMA_CSRx [TRIG_EN]. If the source address and destination is not in wrap around mode, the PDMA will continue the transfer until PDMA_CBCRx comes down to zero (in wrap around mode, when PDMA_CBCRx is equal to zero, the PDMA will reload PDMA_CBCRx and work around until software disables PDMA_CSRx[DMACEN]), If an error occurs during the PDMA operation, the channel stops unless software clears the error condition and sets the PDMA_CSRx [SW_RST] to reset the PDMA channel and set PDMA_CSRx [PDMACEN] and [TRIG_EN] bits field to start again.

In PDMA (Peripheral-to-Memory or Memory-to-Peripheral) mode, DMA can transfer data between the Peripherals APB IP (ex: UART, SPI, ADC….) and Memory.

5.8.4.3 CRC

The DMA controller contains a cyclic redundancy check (CRC) generator that can perform CRC calculation with programmable polynomial settings. The operation polynomial includes CRC-CCITT, CRC-8, CRC-16 and CRC-32; Software can choose the operation polynomial mode by setting CRC_MODE fields in CRC_CTL register.

The CRC engine support CPU PIO mode (CRC_CTL [CRCCEN] = 1, CRC_CTL [TRIG_EN] = 0) and DMA transfer mode (CRC_CTL [CRCCEN] = 1, CRC_CTL [TRIG_EN] = 1). The following sequence is a program sequence example.

Procedure When Operating in CPU PIO Mode:

1. Enable CRC engine by setting CRCCEN bit in CRC_CTL register.
2. Initial Setting. Setting the data format (WDATA_RVS, CHECKSUM_RVS, WDATA_COM and CHECKSUM_COM by setting CRC_CTL register), initial seed value (CRC_SEED) and select the data length by setting CRC_CTL [CPU_WDLEN] register.
3. Setting CRC reset to load the initial seed value to CRC circuit by setting CRC_RST bit in CRC_CTL register.
4. Write data to CRC_WDATA to perform CRC calculation.
5. Get the CRC checksum result by reading CRC_CHECKSUM register.

Procedure When Operating in CRC DMA Mode:

1. Enable CRC engine by setting CRCCEN bit in CRC_CTL register.
2. Initial Setting. Setting the data format (WDATA_RVS, CHECKSUM_RVS, WDATA_COM and CHECKSUM_COM by setting CRC_CTL register), initial seed value (CRC_SEED).
3. Give a valid source address and transfer count by setting CRC_DMASAR and CRC_DMABCR.
4. Enable CRC_CTL [TRIG_EN] and then hardware will reset the seed value and then read memory data to perform CRC calculation.
5. Wait CRC DMA transfer and CRC calculation done and then get the CRC checksum result by reading CRC_CHECKSUM register.

5.8.5 Register Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
VDMA Base Address: **VDMA_BA = 0x5000_8000**				
VDMA_CSR	VDMA_BA+0x00	R/W	VDMA Control Register	0x0000_0000
VDMA_SAR	VDMA_BA+0x04	R/W	VDMA Source Address Register	0x0000_0000
VDMA_DAR	VDMA_BA+0x08	R/W	VDMA Destination Address Register	0x0000_0000
VDMA_BCR	VDMA_BA+0x0C	R/W	VDMA Transfer Byte Count Register	0x0000_0000
VDMA_CSAR	VDMA_BA+0x14	R	VDMA Current Source Address Register	0x0000_0000
VDMA_CDAR	VDMA_BA+0x18	R	VDMA Current Destination Address Register	0x0000_0000
VDMA_CBCR	VDMA_BA+0x1C	R	VDMA Current Transfer Byte Count Register	0x0000_0000
VDMA_IER	VDMA_BA+0x20	R/W	VDMA Interrupt Enable Register	0x0000_0001
VDMA_ISR	VDMA_BA+0x24	R/W	VDMA Interrupt Status Register	0x0000_0000
VDMA_SASOCR	VDMA_BA+0x2C	R/W	VDMA Source Address Stride Offset Register	0x0000_0000
VDMA_DASOCR	VDMA_BA+0x30	R/W	VDMA Destination Address Stride Offset Register	0x0000_0000
PDMA CHn Base Address : **PDMA_BA_CHn = 0x5000_8000 + 0x100*n** **n=1,2..6**				
PDMA_CSRx	PDMA_BA_CHx + 0x00	R/W	PDMA Control and Status Register	0x0000_0000
PDMA_SARx	PDMA_BA_CHx + 0x04	R/W	PDMA Transfer Source Address Register	0x0000_0000
PDMA_DARx	PDMA_BA_CHx + 0x08	R/W	PDMA Transfer Destination Address Register	0x0000_0000
PDMA_BCRx	PDMA_BA_CHx + 0x0C	R/W	PDMA Transfer Byte Count Register	0x0000_0000
PDMA_CSARx	PDMA_BA_CHx + 0x14	R	PDMA Current Source Address Register	0x0000_0000
PDMA_CDARx	PDMA_BA_CHx + 0x18	R	PDMA Current Destination Address Register	0x0000_0000
PDMA_CBCRx	PDMA_BA_CHx + 0x1C	R	PDMA Current Byte Count Register	0x0000_0000
PDMA_IERx	PDMA_BA_CHx + 0x20	R/W	PDMA Interrupt Enable Control Register	0x0000_0001
PDMA_ISRx	PDMA_BA_CHx + 0x24	R/W	PDMA Interrupt Status Register	0x0000_0000
PDMA_TCRx	PDMA_BA_CHx + 0x28	R/W	PDMA Timer Count Setting Register	0x0000_0000
CRC Base Address : **CRC_BA = 0x5000_8E00**				
CRC_CTL	CRC_BA+0x00	R/W	CRC Control Register	0x2000_0000
CRC_DMASAR	CRC_BA+0x04	R/W	CRC DMA Transfer Source Address Register	0x0000_0000

CRC_DMABCR	CRC_BA +0x0C	R/W	CRC DMA Transfer Byte Count Register	0x0000_0000
CRC_DMACSAR	CRC_BA+0x14	R/W	CRC DMA Current Source Address Register	0x0000_0000
CRC_DMACBCR	CRC_BA+0x1C	R/W	CRC DMA Current Byte Count Register	0x0000_0000
CRC_DMAIER	CRC_BA+0x20	R/W	CRC DMA Interrupt Enable Control Register	0x0000_0001
CRC_DMAISR	CRC_BA+0x24	R/W	CRC DMA Interrupt Status Register	0x0000_0000
CRC_WDATA	CRC_BA+0x80	R/W	CRC Write Data Register	0x0000_0000
CRC_SEED	CRC_BA+0x84	R/W	CRC Seed Register	0xFFFF_FFFF
CRC_CHECKSUM	CRC_BA+0x88	R	CRC Checksum Register	0x0000_0000
DMA Base Address: **DMA_BA_GCR = 0x5000_8F00**				
DMA_GCRCSR	DMA_BA_GCR+0x00	R/W	DMA Global Control and Status Register	0x0000_0000
DMA_DSSR0	DMA_BA_GCR+0x04	R/W	DMA Service Selection Control Register 0	0x1F1F_1F00
DMA_DSSR1	DMA_BA_GCR+0x08	R/W	DMA Service Selection Control Register 1	0x001F_1F1F
DMA_GCRISR	DMA_BA_GCR+0x0C	R	DMA Global Interrupt Status Register	0x0000_0000

5.8.6 Register Description

VDMA Control and Status Register (VDMA_CSR)

Register	Offset	R/W	Description	Reset Value
VDMA_CSR	VDMA_BA+0x00	R/W	VDMA Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TRIG_EN	Reserved						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				DIR_SEL	STRIDE_EN	Reserved	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						SW_RST	VDMACEN

Bits	Description	
[31:24]	**Reserved**	**Reserved.**
[23]	**TRIG_EN**	**TRIG_EN** 1 = VDMA data read or write transfer Enabled. 0 = No effect. **Note1:** When VDMA transfer is completed, this bit will be cleared automatically. **Note2:** If the bus error occurs, all VDMA transfer will be stopped. Software must reset all VDMA channel, and then trig again.
[22:12]	**Reserved**	**Reserved.**
[11]	**DIR_SEL**	**Transfer Source/Destination Address Direction Select** 1 = Transfer address is decremented successively. 0 = Transfer address is incremented successively.
[10]	**STRIDE_EN**	**Stride Mode Enable** 1 = Stride transfer mode Enabled. 0 = Stride transfer mode Disabled.
[9:2]	**Reserved**	**Reserved.**
[1]	**SW_RST**	**Software Engine Reset** 1 = Reset the internal state machine and pointers. The contents of control register will not be cleared. This bit will be auto cleared after few clock cycles. 0 = No effect.
[0]	**VDMACEN**	**VDMA Channel Enable** Setting this bit to “1” enables VDMA’s operation. If this bit is cleared, VDMA will ignore all VDMA request and force Bus Master into IDLE state. **Note:** SW_RST will clear this bit.

VDMA Transfer Source Address Register (VDMA_SAR)

Register	Offset	R/W	Description	Reset Value
VDMA_SAR	VDMA_BA+0x04	R/W	VDMA Source Address Register	0x0000_0000

31	30	29	28	27	26	25	24
VDMA_SAR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VDMA_SAR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VDMA_SAR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VDMA_SAR							

Bits	Description	
[31:0]	**VDMA_SAR**	**VDMA Transfer Source Address Register** This field indicates a 32-bit source address of VDMA.

VDMA Transfer Destination Address Register (VDMA_DAR)

Register	Offset	R/W	Description	Reset Value
VDMA_DAR	VDMA_BA+0x08	R/W	VDMA Destination Address Register	0x0000_0000

31	30	29	28	27	26	25	24
VDMA_DAR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VDMA_DAR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VDMA_DAR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VDMA_DAR							

Bits	Description	
[31:0]	**VDMA_DAR**	**VDMA Transfer Destination Address Register** This field indicates a 32-bit destination address of VDMA.

VDMA Transfer Byte Count Register (VDMA_BCR)

Register	Offset	R/W	Description	Reset Value
VDMA_BCR	VDMA_BA+0x0C	R/W	VDMA Transfer Byte Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VDMA_BCR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VDMA_BCR							

Bits	Description	
[31:16]	**Reserved**	**Reserved.**
[15:0]	**VDMA_BCR**	**VDMA Transfer Byte Count Register** This field indicates a 16-bit transfer byte count of VDMA. **Note:** In Stride Enable mode (VDMA_CSR [10] = "0"]), the transfer byte count (VDMA_BCR) must be an integer multiple of STBC (VDMA_SASOCR [31:16]).

VDMA Current Source Address Register (VDMA_CSAR)

Register	Offset	R/W	Description	Reset Value
VDMA_CSAR	VDMA_BA+0x14	R	VDMA Current Source Address Register	0x0000_0000

31	30	29	28	27	26	25	24
VDMA_CSAR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VDMA_CSAR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VDMA_CSAR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VDMA_CSAR							

Bits	Description	
[31:0]	**VDMA_CSAR**	**VDMA Current Source Address Register (Read Only)** This field indicates the source address where the VDMA transfer is just occurring.

VDMA Current Destination Address Register (VDMA_CDAR)

Register	Offset	R/W	Description	Reset Value
VDMA_CDAR	VDMA_BA+0x18	R	VDMA Current Destination Address Register	0x0000_0000

31	30	29	28	27	26	25	24
VDMA_CDAR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VDMA_CDAR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VDMA_CDAR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VDMA_CDAR							

Bits	Description	
[31:0]	**VDMA_CDAR**	**VDMA Current Destination Address Register (Read Only)** This field indicates the destination address where the VDMA transfer is just occurring.

VDMA Current Byte Count Register (VDMA_CBCR)

Register	Offset	R/W	Description	Reset Value
VDMA_CBCR	VDMA_BA+0x1C	R	VDMA Current Transfer Byte Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VDMA_CBCR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VDMA_CBCR							

Bits	Description	
[31:16]	**Reserved**	**Reserved.**
[15:0]	**VDMA_CBCR**	**VDMA Current Byte Count Register (Read Only)** This field indicates the current remained byte count of VDMA.

VDMA Interrupt Enable Control Register (VDMA_IER)

Register	Offset	R/W	Description	Reset Value
VDMA_IER	VDMA_BA+0x20	R/W	VDMA Interrupt Enable Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**TD_IE**	**TABORT_IE**

Bits	Description	
[31:2]	**Reserved**	**Reserved**
[1]	**TD_IE**	**VDMA Transfer Done Interrupt Enable** 1 = Enabled interrupt generator during VDMA transfer done. 0 = Disabled interrupt generator during VDMA transfer done.
[0]	**TABORT_IE**	**VDMA Read/Write Target Abort Interrupt Enable** 1 = Enabled target abort interrupt generation during VDMA transfer. 0 = Disabled target abort interrupt generation during VDMA transfer.

VDMA Interrupt Status Register (VDMA_ISR)

Register	Offset	R/W	Description	Reset Value
VDMA_ISR	VDMA_BA+0x24	R/W	VDMA Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**TD_IS**	**TABORT_IS**

Bits	Description	
[31:2]	**Reserved**	**Reserved**
[1]	**TD_IS**	**Transfer Done Interrupt Status Flag** This bit indicates that VDMA has finished all transfer. 1 = Done. 0 = Not finished yet. **Note:** This bit is cleared by writing “1” to itself.
[0]	**TABORT_IS**	**VDMA Read/Write Target Abort Interrupt Status Flag** 1 = Bus ERROR response received. 0 = No bus ERROR response received. **Note1:** This bit is cleared by writing “1” to itself. **Note2:** The VDMA_ISR [TABORT_IF] indicate bus master received ERROR response or not, if bus master received occur it means that target abort is happened. VDMA controller will stop transfer and respond this event to software then go to IDLE state. When target abort occurred, software must reset VDMA controller, and then transfer those data again.

VDMA Source Address Stride Offset Control Register (VDMA_SASOCR)

Register	Offset	R/W	Description	Reset Value
VDMA_SASOCR	VDMA_BA+0x2C	R/W	VDMA Source Address Stride Offset Register	0x0000_0000

31	30	29	28	27	26	25	24
STBC							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
STBC							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SASTOBL							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SASTOBL							

Bits	Description	
[31:16]	**STBC**	**VDMA Stride Transfer Byte Count** The 16-bit register defines the stride transfer byte count of each row.
[15:0]	**SASTOBL**	**VDMA Source Address Stride Offset Byte Length** The 16-bit register defines the source address stride transfer offset count of each row.

VDMA Destination Address Stride Offset Control Register (VDMA_DASOCR)

Register	Offset	R/W	Description	Reset Value
VDMA_DASOCR	VDMA_BA+0x30	R/W	VDMA Destination Address Stride Offset Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DASTOBL							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DASTOBL							

Bits	Description	
[31:16]	**Reserved**	**Reserved.**
[15:0]	**DASTOBL**	**VDMA Destination Address Stride Offset Byte Length** The 16-bit register defines the destination address stride transfer offset count of each row.

PDMA Control and Status Register (PDMA_CSRx)

Register	Offset	R/W	Description	Reset Value
PDMA_CSR1	PDMA_BA_CH1+0x00	R/W	PDMA Control Register	0x0000_0000
PDMA_CSR2	PDMA_BA_CH2+0x00	R/W	PDMA Control Register	0x0000_0000
PDMA_CSR3	PDMA_BA_CH3+0x00	R/W	PDMA Control Register	0x0000_0000
PDMA_CSR4	PDMA_BA_CH4+0x00	R/W	PDMA Control Register	0x0000_0000
PDMA_CSR5	PDMA_BA_CH5+0x00	R/W	PDMA Control Register	0x0000_0000
PDMA_CSR6	PDMA_BA_CH6+0x00	R/W	PDMA Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TRIG_EN	Reserved		APB_TWS		Reserved		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			TO_EN	Reserved			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DAD_SEL		SAD_SEL		MODE_SEL		SW_RST	PDMACEN

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:24]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[23]</td><td>TRIG_EN</td><td>TRIG_EN
1 = PDMA data read or write transfer Enabled.
0 = No effect.
Note1: When PDMA transfer completed, this bit will be cleared automatically.
Note2: If the bus error occurs, all PDMA transfer will be stopped. Software must reset all PDMA channel, and then trig again.</td></tr>
<tr><td>[22:21]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[20:19]</td><td>APB_TWS</td><td>Peripheral Transfer Width Selection
<table>
<tr><th>APB_TWS</th><th>Description</th></tr>
<tr><td>00</td><td>One word (32 bits) is transferred for every PDMA operation.</td></tr>
<tr><td>01</td><td>One byte (8 bits) is transferred for every PDMA operation.</td></tr>
<tr><td>10</td><td>One half-word (16 bits) is transferred for every PDMA operation.</td></tr>
<tr><td>11</td><td>Reserved.</td></tr>
</table>
Note: This field is meaningful only when MODE_SEL is IP to Memory mode (APB-to-Memory) or Memory to IP mode (Memory-to-APB).</td></tr>
</table>

Bits	Description	
[18:13]	**Reserved**	**Reserved.**
[12]	**TO_EN**	**Time-out Enable** This bit will enable PDMA internal counter. While this counter counts to zero, the TO_IS will be set. 1 = PDMA internal counter Enabled. 0 = PDMA internal counter Disabled.
[11:8]	**Reserved**	**Reserved.**
[7:6]	**DAD_SEL**	**Transfer Destination Address Direction Selection** **DAD_SEL** / **Description** 00: Transfer Destination address is incremented successively 01: Reserved. 10: Transfer Destination address is fixed (This feature can be used when data where transferred from multiple sources to a single destination) 11: Transfer Destination address is wrapped around (When the PDMA_CBCR is equal to zero, the PDMA_CDAR and PDMA_CBCR register will be updated by PDMA_DAR and PDMA_BCR automatically. PDMA will start another transfer without software trigger until PDMA_EN disabled. When the PDMA_EN is disabled, the PDMA will complete the active transfer but the remained data which in the PDMA_BUF will not transfer to destination address).
[5:4]	**SAD_SEL**	**Transfer Source Address Direction Selection** **SAD_SEL** / **Description** 00: Transfer Source address is incremented successively. 01: Reserved. 10: Transfer Source address is fixed (This feature can be used when data where transferred from a single source to multiple destinations). 11: Transfer Source address is wrap around (When the PDMA_CBCR is equal to zero, the PDMA_CSAR and PDMA_CBCR register will be updated by PDMA_SAR and PDMA_BCR automatically. PDMA will start another transfer without software trigger until PDMA_EN disabled. When the PDMA_EN is disabled, the PDMA will complete the active transfer but the remained data which in the PDMA_BUF will not transfer to destination address).
[3:2]	**MODE_SEL**	**PDMA Mode Select** **MODE_SEL** / **Description** 00: Memory to Memory mode (Memory-to-Memory). 01: IP to Memory mode (APB-to-Memory) 10: Memory to IP mode (Memory-to-APB). 11: Reserved.
[1]	**SW_RST**	**Software Engine Reset** 1 = Reset the internal state machine and pointers. The contents of control register will not be cleared. This bit will be auto cleared after few clock cycles. 0 = No effect.

Bits	Description	
[0]	**PDMACEN**	**PDMA Channel Enable** Setting this bit to “1” enables PDMA’s operation. If this bit is cleared, PDMA will ignore all PDMA request and force Bus Master into IDLE state. **Note:** SW_RST will clear this bit.

PDMA Transfer Source Address Register (PDMA_SARx)

Register	Offset	R/W	Description	Reset Value
PDMA_SAR1	PDMA_BA_CH1+0x04	R/W	PDMA Source Address Register	0x0000_0000
PDMA_SAR2	PDMA_BA_CH2+0x04	R/W	PDMA Source Address Register	0x0000_0000
PDMA_SAR3	PDMA_BA_CH3+0x04	R/W	PDMA Source Address Register	0x0000_0000
PDMA_SAR4	PDMA_BA_CH4+0x04	R/W	PDMA Source Address Register	0x0000_0000
PDMA_SAR5	PDMA_BA_CH5+0x04	R/W	PDMA Source Address Register	0x0000_0000
PDMA_SAR6	PDMA_BA_CH6+0x04	R/W	PDMA Source Address Register	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_SAR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PDMA_SAR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMA_SAR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMA_SAR							

Bits	Description	
[31:0]	**PDMA_SAR**	**PDMA Transfer Source Address Register** This field indicates a 32-bit source address of PDMA. **Note:** The source address must be word alignment.

PDMA Transfer Destination Address Register (PDMA_DARx)

Register	Offset	R/W	Description	Reset Value
PDMA_DAR1	PDMA_BA_CH1+0x08	R/W	PDMA Destination Address Register	0x0000_0000
PDMA_DAR2	PDMA_BA_CH2+0x08	R/W	PDMA Destination Address Register	0x0000_0000
PDMA_DAR3	PDMA_BA_CH3+0x08	R/W	PDMA Destination Address Register	0x0000_0000
PDMA_DAR4	PDMA_BA_CH4+0x08	R/W	PDMA Destination Address Register	0x0000_0000
PDMA_DAR5	PDMA_BA_CH5+0x08	R/W	PDMA Destination Address Register	0x0000_0000
PDMA_DAR6	PDMA_BA_CH6+0x08	R/W	PDMA Destination Address Register	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_DAR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PDMA_DAR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMA_DAR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMA_DAR							

Bits	Description	
[31:0]	**PDMA_DAR**	**PDMA Transfer Destination Address Register** This field indicates a 32-bit destination address of PDMA. Note : The destination address must be word alignment

PDMA Transfer Byte Count Register (PDMA_BCRx)

Register	Offset	R/W	Description	Reset Value
PDMA_BCR1	PDMA_BA_CH1+0x0C	R/W	PDMA Transfer Byte Count Register	0x0000_0000
PDMA_BCR2	PDMA_BA_CH2+0x0C	R/W	PDMA Transfer Byte Count Register	0x0000_0000
PDMA_BCR3	PDMA_BA_CH3+0x0C	R/W	PDMA Transfer Byte Count Register	0x0000_0000
PDMA_BCR4	PDMA_BA_CH4+0x0C	R/W	PDMA Transfer Byte Count Register	0x0000_0000
PDMA_BCR5	PDMA_BA_CH5+0x0C	R/W	PDMA Transfer Byte Count Register	0x0000_0000
PDMA_BCR6	PDMA_BA_CH6+0x0C	R/W	PDMA Transfer Byte Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMA_BCR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMA_BCR							

Bits	Description	
[31:24]	**Reserved**	**Reserved.**
[15:0]	**PDMA_BCR**	**PDMA Transfer Byte Count Register** This field indicates a 16-bit transfer byte count of PDMA. **Note:** In Memory-to-memory (PDMA_CSR [MODE_SEL] = 00) mode, the transfer byte count must be word alignment.

PDMA Current Source Address Register (PDMA_CSARx)

Register	Offset	R/W	Description	Reset Value
PDMA_CSAR1	PDMA_BA_CH1+0x14	R	PDMA Current Source Address Register	0x0000_0000
PDMA_CSAR2	PDMA_BA_CH2+0x14	R	PDMA Current Source Address Register	0x0000_0000
PDMA_CSAR3	PDMA_BA_CH3+0x14	R	PDMA Current Source Address Register	0x0000_0000
PDMA_CSAR4	PDMA_BA_CH4+0x14	R	PDMA Current Source Address Register	0x0000_0000
PDMA_CSAR5	PDMA_BA_CH5+0x14	R	PDMA Current Source Address Register	0x0000_0000
PDMA_CSAR6	PDMA_BA_CH6+0x14	R	PDMA Current Source Address Register	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_CSAR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PDMA_CSAR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMA_CSAR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMA_CSAR							

Bits	Description	
[31:0]	**PDMA_CSAR**	**PDMA Current Source Address Register (Read Only)** This field indicates the source address where the PDMA transfer is just occurring.

PDMA Current Destination Address Register (PDMA_CDARx)

Register	Offset	R/W	Description	Reset Value
PDMA_CDAR1	PDMA_BA_CH1+0x18	R	PDMA Current Destination Address Register	0x0000_0000
PDMA_CDAR2	PDMA_BA_CH2+0x18	R	PDMA Current Destination Address Register	0x0000_0000
PDMA_CDAR3	PDMA_BA_CH3+0x18	R	PDMA Current Destination Address Register	0x0000_0000
PDMA_CDAR4	PDMA_BA_CH4+0x18	R	PDMA Current Destination Address Register	0x0000_0000
PDMA_CDAR5	PDMA_BA_CH5+0x18	R	PDMA Current Destination Address Register	0x0000_0000
PDMA_CDAR6	PDMA_BA_CH6+0x18	R	PDMA Current Destination Address Register	0x0000_0000

31	30	29	28	27	26	25	24
PDMA_CDAR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PDMA_CDAR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMA_CDAR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMA_CDAR							

Bits	Description	
[31:0]	**PDMA_CDAR**	**PDMA Current Destination Address Register (Read Only)** This field indicates the destination address where the PDMA transfer is just occurring.

PDMA Current Byte Count Register (PDMA_CBCRx)

Register	Offset	R/W	Description	Reset Value
PDMA_CBCR1	PDMA_BA_CH1+0x1C	R	PDMA Current Transfer Byte Count Register	0x0000_0000
PDMA_CBCR2	PDMA_BA_CH2+0x1C	R	PDMA Current Transfer Byte Count Register	0x0000_0000
PDMA_CBCR3	PDMA_BA_CH3+0x1C	R	PDMA Current Transfer Byte Count Register	0x0000_0000
PDMA_CBCR4	PDMA_BA_CH4+0x1C	R	PDMA Current Transfer Byte Count Register	0x0000_0000
PDMA_CBCR5	PDMA_BA_CH5+0x1C	R	PDMA Current Transfer Byte Count Register	0x0000_0000
PDMA_CBCR6	PDMA_BA_CH6+0x1C	R	PDMA Current Transfer Byte Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PDMA_CBCR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMA_CBCR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMA_CBCR							

Bits	Description	
[31:24]	**Reserved**	**Reserved.**
[23:0]	**PDMA_CBCR**	**PDMA Current Byte Count Register (Read Only)** This field indicates the current remained byte count of PDMA. **Note:** These fields will be changed when PDMA finish data transfer (data transfer to destination address),

PDMA Interrupt Enable Control Register (PDMA_IERx)

Register	Offset	R/W	Description	Reset Value
PDMA_IER1	PDMA_BA_CH1+0x20	R/W	PDMA Interrupt Enable Register	0x0000_0001
PDMA_IER2	PDMA_BA_CH2+0x20	R/W	PDMA Interrupt Enable Register	0x0000_0001
PDMA_IER3	PDMA_BA_CH3+0x20	R/W	PDMA Interrupt Enable Register	0x0000_0001
PDMA_IER4	PDMA_BA_CH4+0x20	R/W	PDMA Interrupt Enable Register	0x0000_0001
PDMA_IER5	PDMA_BA_CH5+0x20	R/W	PDMA Interrupt Enable Register	0x0000_0001
PDMA_IER6	PDMA_BA_CH6+0x20	R/W	PDMA Interrupt Enable Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**TO_IE**	**WRA_BCR_IE**				**TD_IE**	**TABORT_IE**

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:7]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[6]</td><td>TO_IE</td><td>Time-Out Interrupt Enable
1 = Time-out interrupt Enabled.
0 = Time-out interrupt Disabled.</td></tr>
<tr><td>[5:2]</td><td>WRA_BCR_IE</td><td>Wrap Around Byte Count Interrupt Enable
<table><tr><th>WRA_BCR_IE</th><th>Description</th></tr><tr><td>0001</td><td>Interrupt enable of PDMA_CBCR equals 0</td></tr><tr><td>0100</td><td>Interrupt enable of PDMA_CBCR equals 1/2 PDMA_BCR.</td></tr><tr><td>Others</td><td>Reserved.</td></tr></table></td></tr>
<tr><td>[1]</td><td>TD_IE</td><td>PDMA Transfer Done Interrupt Enable
1 = Interrupt generator Enabled when PDMA transfer is done.
0 = Interrupt generator Disabled when PDMA transfer is done.</td></tr>
<tr><td>[0]</td><td>TABORT_IE</td><td>PDMA Read/Write Target Abort Interrupt Enable
1 = Target abort interrupt generation Enabled during PDMA transfer.
0 = Target abort interrupt generation Disabled during PDMA transfer.</td></tr>
</table>

PDMA Interrupt Status Register (PDMA_ISRx)

Register	Offset	R/W	Description	Reset Value
PDMA_ISR1	PDMA_BA_CH1+0x24	R/W	PDMA Interrupt Status Register	0x0000_0000
PDMA_ISR2	PDMA_BA_CH2+0x24	R/W	PDMA Interrupt Status Register	0x0000_0000
PDMA_ISR3	PDMA_BA_CH3+0x24	R/W	PDMA Interrupt Status Register	0x0000_0000
PDMA_ISR4	PDMA_BA_CH4+0x24	R/W	PDMA Interrupt Status Register	0x0000_0000
PDMA_ISR5	PDMA_BA_CH5+0x24	R/W	PDMA Interrupt Status Register	0x0000_0000
PDMA_ISR6	PDMA_BA_CH6+0x24	R/W	PDMA Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	TO_IS	WRA_BCR_IS				TD_IS	TABORT_IS

Bits	Description	
[31:7]	**Reserved**	**Reserved**
[6]	**TO_IS**	**Time-Out Interrupt Status Flag** This flag indicated that PDMA has waited peripheral request for a period defined by PDMA_TCR. 1 = Time-out flag. 0 = No time-out flag. **Note:** This bit is cleared by writing "1" to itself.
[5:2]	**WRA_BCR_IS**	**Wrap Around Transfer Byte Count Interrupt Status Flag** WAR_)CR_IS [0] (xxx1) = PDMA_CBCR equal 0 flag. WAR_BCR_IS [2] (x1xx) = PDMA_CBCR equal 1/2 PDMA_BCR flag. **Note:** Each bit is cleared by writing "1" to itself. This field is only valid in wrap around mode. (PDMA_CSR[DAD_SEL] =11 or PDMA_CSR[SAD_SEL] =11)
[1]	**TD_IS**	**Transfer Done Interrupt Status Flag** This bit indicates that PDMA has finished all transfer. 1 = Done. 0 = Not finished yet. **Note:** This bit is cleared by writing "1" to itself.
[0]	**TABORT_IS**	**PDMA Read/Write Target Abort Interrupt Status Flag**

Bits	Description	
		1 = Bus ERROR response received. 0 = No bus ERROR response received. **Note1:** This bit is cleared by writing “1” to itself. **Note2:** The PDMA_ISR [TABORT_IF] indicate bus master received ERROR response or not, if bus master received occur it means that target abort is happened. PDMA controller will stop transfer and respond this event to software then go to IDLE state. When target abort occurred, software must reset PDMA controller, and then transfer those data again.

PDMA Timer Count Setting Register (PDMA_TCRx)

Register	Offset	R/W	Description	Reset Value
PDMA_TCR1	PDMA_BA_CH1+0x28	R/W	PDMA Timer Counter Setting Register	0x0000_0000
PDMA_TCR2	PDMA_BA_CH2+0x28	R/W	PDMA Timer Counter Setting Register	0x0000_0000
PDMA_TCR3	PDMA_BA_CH3+0x28	R/W	PDMA Timer Counter Setting Register	0x0000_0000
PDMA_TCR4	PDMA_BA_CH4+0x28	R/W	PDMA Timer Counter Setting Register	0x0000_0000
PDMA_TCR5	PDMA_BA_CH5+0x28	R/W	PDMA Timer Counter Setting Register	0x0000_0000
PDMA_TCR6	PDMA_BA_CH6+0x28	R/W	PDMA Timer Counter Setting Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMA_TCR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMA_TCR							

Bits	Description	
[31:16]	**Reserved**	**Reserved.**
[15:0]	**PDMA_TCR**	**PDMA Timer Count Setting Register** Each PDMA channel contains an internal counter. The internal counter loads the value of PDAM_TCR and starts counting down when setting PDMA_CSRx [TO_EN] register. PDMA will request interrupt when this internal counter reaches zero and PDMA_IERx[TO_IE] is high. This internal counter will reload and start counting when completing each peripheral request service.

CRC Control Register (CRC_CTL)

Register	Offset	R/W	Description	Reset Value
CRC_CTL	CRC_BA+0x00	R/W	CRC Control Register	0x2000_0000

31	30	29	28	27	26	25	24
CRC_MODE		CPU_WDLEN		CHECKSUM_COM	WDATA_COM	CHECKSUM_RVS	WDATA_RVS
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TRIG_EN	Reserved						
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						CRC_RST	CRCEN

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:30]</td><td>CRC_MODE</td><td>CRC Polynomial Mode
<table><tr><th>CRC_MODE</th><th>Description</th></tr><tr><td>00</td><td>CRC-CCITT Polynomial Mode</td></tr><tr><td>01</td><td>CRC-8 Polynomial Mode</td></tr><tr><td>10</td><td>CRC-16 Polynomial Mode</td></tr><tr><td>11</td><td>CRC-32 Polynomial Mode</td></tr></table></td></tr>
<tr><td>[29:28]</td><td>CPU_WDLEN</td><td>CPU Write Data Length
When operating in CPU PIO mode (CRCCEN= 1, TRIG_EN = 0), this field indicates the write data length
<table><tr><th>CPU_WDLEN</th><th>Description</th></tr><tr><td>00</td><td>The data length is 8-bit mode</td></tr><tr><td>01</td><td>The data length is 16-bit mode</td></tr><tr><td>10</td><td>The data length is 32-bit mode</td></tr><tr><td>11</td><td>Reserved</td></tr></table>Note1: This field is only used for CPU PIO mode.
Note2: When the data length is 8-bit mode, the valid data is CRC_WDATA [7:0], and if the data length is 16 bit mode, the valid data is CRC_WDATA [15:0].</td></tr>
<tr><td>[27]</td><td>CHECKSUM_COM</td><td>Checksum Complement
1 = 1's complement for CRC checksum
0 = No bit order reverse for CRC checksum</td></tr>
<tr><td>[26]</td><td>WDATA_COM</td><td>Write Data Complement
1 = 1's complement for CRC write data in</td></tr>
</table>

Bits	Description	
		0 = No bit order reverse for CRC write data in.
[25]	**CHECKSUM_RVS**	**Checksum Reverse** 1 = Bit order reverse for CRC checksum 0 = No bit order reverse for CRC checksum **Note:** If the checksum data is 0XDD7B0F2E, the bit order reverse for CRC checksum is 0x74F0DEBB
[24]	**WDATA_RVS**	**Write Data Order Reverse** 1 = Bit order reverse for CRC write data in (per byre) 0 = No bit order reverse for CRC write data in. **Note:** If the write data is 0xAABBCCDD, the bit order reverse for CRC write data in is 0x55DD33BB
[23]	**TRIG_EN**	**Trigger Enable** 1 = CRC DMA data read or write transfer Enabled. 0 = No effect **Note1:** If this bit assert that indicates the CRC engine operation in CRC DMA mode, so don't filled any data in CRC_WDATA register. **Note2:** When CRC DMA transfer completed, this bit will be cleared automatically. **Note3:** If the bus error occurs, all CRC DMA transfer will be stopped. Software must reset all DMA channel, and then trigger again.
[22:2]	**Reserved**	**Reserved.**
[1]	**CRC_RST**	**CRC Engine Reset** 0 = Writing 0 to this bit has no effect. 1 = Writing 1 to this bit will reset the internal CRC state machine and internal buffer. The contents of control register will not be cleared. This bit will be auto cleared after few clock cycles. **Note:** When operating in CPU PIO mode, setting this bit will reload the initial seed value
[0]	**CRCCEN**	**CRC Channel Enable** Setting this bit to 1 enables CRC's operation. When operating in CRC DMA mode (TRIG_EN = 1), if user clear this bit, the DMA operation will be continuous until all CRC DMA operation done, and the TRIG_EN bit will asserted until all CRC DMA operation done. But in this case, the CRC_DMAISR [BLKD_IF] flag will inactive, user can read CRC result by reading CRC_CHECKSUM register when TRIG_EN = 0 When operating in CRC DMA mode (TRIG_EN = 1), if user want to stop the transfer immediately, user can write 1 to CRC_RST bit to stop the transmission.

CRC DMA Transfer Source Address Register (CRC_DMASAR)

Register	Offset	R/W	Description	Reset Value
CRC_DMASAR	CRC_BA+0x04	R/W	CRC DMA Source Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CRC_DMASAR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CRC_DMASAR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRC_DMASAR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRC_DMASAR							

Bits	Description	
[31:0]	**CRC_DMASAR**	**CRC DMA Transfer Source Address Register** This field indicates a 32-bit source address of CRC DMA. Note : The source address must be word alignment

CRC DMA Transfer Byte Count Register (CRC_DMABCR)

Register	Offset	R/W	Description	Reset Value
CRC_DMABCR	CRC_BA+0x0C	R/W	CRC Transfer Byte Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRC_DMABCR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRC_DMABCR							

Bits	Description	
[31:0]	**Reserved**	**Reserved**
[15:0]	**CRC_DMABCR**	**CRC DMA Transfer Byte Count Register** This field indicates a 16-bit transfer byte count number of CRC DMA

CRC DMA Current Source Address Register (CRC_DMACSAR)

Register	Offset	R/W	Description	Reset Value
CRC_DMACSAR	CRC_BA+0x14	R/W	CRC Current Source Address Register	0x0000_0000

31	30	29	28	27	26	25	24
CRC_DMACSAR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CRC_DMACSAR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRC_DMACSAR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRC_DMACSAR							

Bits	Description	
[31:0]	**CRC_DMACSAR**	**CRC DMA Current Source Address Register (Read Only)** This field indicates the source address where the CRC DMA transfer is just occurring.

CRC DMA Current Byte Count Register (CRC_DMACBCR)

Register	Offset	R/W	Description	Reset Value
CRC_DMACBCR	CRC_BA+0x1C	R/W	CRC Current Transfer Byte Count Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRC_DMACBCR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRC_DMACBCR							

Bits	Description	
[31:16]	**Reserved**	**Reserved**
[15:0]	**CRC_DMACBCR**	**CRC DMA Current Byte Count Register (Read Only)** This field indicates the current remained byte count of CRC_DMA. **Note:** CRC_RST will clear this register value.

CRC DMA Interrupt Enable Control Register (CRC_DMAIER)

Register	Offset	R/W	Description	Reset Value
CRC_DMAIER	CRC_BA+0x20	R/W	CRC Interrupt Enable Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**BLKD_IE**	**TABORT_IE**

Bits	Description	
[31:2]	**Reserved**	**Reserved.**
[1]	**BLKD_IE**	**CRC DMA Transfer Done Interrupt Enable** 1 = Interrupt generator Enabled during CRC DMA transfer done. 0 = Interrupt generator Disabled during CRC DMA transfer done.
[0]	**TABORT_IE**	**CRC DMA Read/Write Target Abort Interrupt Enable** 1 = Target abort interrupt generation Enabled during CRC DMA transfer. 0 = Target abort interrupt generation Disabled during CRC DMA transfer.

CRC DMA Interrupt Status Register (CRC_DMAISR)

Register	Offset	R/W	Description	Reset Value
CRC_DMAISR	CRC_BA+0x24	R/W	CRC Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**BLKD_IF**	**TABORT_IF**

Bits	Description	
[31:2]	**Reserved**	**Reserved.**
[1]	**BLKD_IF**	**Block Transfer Done Interrupt Flag** This bit indicates that CRC DMA has finished all transfer. 1 = Done 0 = Not finished yet Software can write 1 to clear this bit to zero
[0]	**TABORT_IF**	**CRC DMA Read/Write Target Abort Interrupt Flag** 1 = Bus ERROR response received 0 = No bus ERROR response received Software can write 1 to clear this bit to zero **Note:** The CRC_DMAISR [TABORT_IF] indicate bus master received ERROR response or not. If bus master received ERROR response, it means that target abort is happened. DMA will stop transfer and respond this event to software then go to IDLE state. When target abort occurred, software must reset DMA, and then transfer those data again

CRC Write Data Register (CRC_WDATA)

Register	Offset	R/W	Description	Reset Value
CRC_WDATA	CRC_BA+0x80	R/W	CRC Write Data Register	0x0000_0000

31	30	29	28	27	26	25	24
CRC_WDATA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CRC_WDATA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRC_WDATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRC_WDATA							

Bits	Description	
[31:0]	**CRC_WDATA**	**CRC Write Data Register** When operating in CPU PIO (CRC_CTL [CRCCEN] = 1, CRC_CTL [TRIG_EN] = 0) mode, software can write data to this field to perform CRC operation; When operating in CRC DMA mode (CRC_CTL [CRCCEN] = 1, CRC_CTL [TRIG_EN] = 0), this field will be used for DMA internal buffer. **Note1:** When operating in CRC DMA mode, so don't filled any data in this field. **Note2:**The CRC_CTL [WDATA_COM] and CRC_CTL [WDATA_RVS] bit setting will affected this field; For example, if WDATA_RVS = 1, if the write data in CRC_WDATA register is 0xAABBCCDD, the read data from CRC_WDATA register will be 0x55DD33BB.

CRC Seed Register (CRC_SEED)

Register	Offset	R/W	Description	Reset Value
CRC_SEED	CRC_BA+0x84	R/W	CRC Seed Register	0xFFFF_FFFF

31	30	29	28	27	26	25	24
CRC_SEED							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CRC_SEED							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRC_SEED							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRC_SEED							

Bits	Description	
[31:0]	**CRC_SEED**	**CRC Seed Register** This field indicates the CRC seed value.

CRC Checksum Register (CRC_CHECKSUM)

Register	Offset	R/W	Description	Reset Value
CRC_CHECKSUM	CRC_BA+0x88	R	CRC Check Sum Register	0x0000_0000

31	30	29	28	27	26	25	24
CRC_CHECKSUM							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CRC_CHECKSUM							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRC_CHECKSUM							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRC_CHECKSUM							

Bits	Description	
[31:0]	**CRC_CHECKSUM**	**CRC Checksum Register** This field indicates the CRC checksum

PDMA Global Control and Status Register (DMA_GCRCSR)

Register	Offset	R/W	Description	Reset Value
DMA_GCRCSR	DMA_BA_GCR+0x00	R/W	DMA Global Control and Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							CRC_CLK_EN
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	CLK6_EN	CLK5_EN	CLK4_EN	CLK3_EN	CLK2_EN	CLK1_EN	CLK0_EN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:25]	**Reserved**	**Reserved.**
[24]	**CRC_CLK_EN**	**CRC Controller Clock Enable Control** 1 = Enabled 0 = Disabled
[23:15]	**Reserved**	**Reserved**
[14]	**CLK6_EN**	**DMA Controller Channel 6 Clock Enable Control** 1 = Enabled 0 = Disabled
[13]	**CLK5_EN**	**DMA Controller Channel 5 Clock Enable Control** 1 = Enabled 0 = Disabled
[12]	**CLK4_EN**	**DMA Controller Channel 4 Clock Enable Control** 1 = Enabled 0 = Disabled
[11]	**CLK3_EN**	**DMA Controller Channel 3 Clock Enable Control** 1 = Enabled 0 = Disabled
[10]	**CLK2_EN**	**DMA Controller Channel 2 Clock Enable Control** 1 = Enabled 0 = Disabled
[9]	**CLK1_EN**	**DMA Controller Channel 1 Clock Enable Control** 1 = Enabled 0 = Disabled

Bits	Description	
[8]	**CLK0_EN**	**DMA Controller Channel 0 Clock Enable Control** 1 = Enabled 0 = Disabled
[7:0]	**Reserved**	**Reserved.**

DMA Service Selection Control Register 0 (DMA_DSSR0)

Register	Offset	R/W	Description	Reset Value
DMA_DSSR0	DMA_BA_GCR+0x04	R/W	DMA Service Selection Control Register 0	0x1F1F_1F00

31	30	29	28	27	26	25	24
Reserved			**CH3_SEL**				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**CH2_SEL**				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**CH1_SEL**				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							

Bits	Description	
[31:29]	**Reserved**	**Reserved.**
[28:24]	**CH3_SEL**	**Channel 3 Selection** This filed defines which peripheral is connected to PDMA channel 3. Software can configure the peripheral setting by CH3_SEL. The channel configuration is the same as CH1_SEL field. Please refer to the explanation of CH1_SEL.
[23:21]	**Reserved**	**Reserved.**
[20:16]	**CH2_SEL**	**Channel 2 Selection** This filed defines which peripheral is connected to PDMA channel 2. Software can configure the peripheral setting by CH2_SEL. The channel configuration is the same as CH1_SEL field. Please refer to the explanation of CH1_SEL.
[15:13]	**Reserved**	**Reserved.**
[12:8]	**CH1_SEL**	**Channel 1 Selection** This filed defines which peripheral is connected to PDMA channel 1. Software can configure the peripheral by setting CH1_SEL.

CH1_SEL	CONNECTION	CH1_SEL	CONNECTION
00000	Connect to SPI0_TX.	10000	Connect to SPI0_RX.
00001	Connect to SPI1_TX.	10001	Connect to SPI1_RX.
00010	Connect to UART0_TX.	10010	Connect to UART0_RX.
00011	Connect to UART1_TX.	10011	Connect to UART1_RX.
00100	Connect to USB_TX.	10100	Connect to USB_RX.
00101	Connect to I^{2}S_TX.	10101	Connect to I^{2}S_RX.
00110	Connect to DAC0_TX.	10110	Connect to ADC.
00111	Connect to DAC1_TX.	10111	Reserved.

Bits	Description	
		<table><tr><td>01000</td><td>Connect to SPI2_TX.</td><td>11000</td><td>Connect to SPI2_RX.</td></tr><tr><td>01001</td><td>Connect to TMR0.</td><td>11001</td><td>Connect to PWM0_CH0.</td></tr><tr><td>01010</td><td>Connect to TMR1.</td><td>11010</td><td>Connect to PWM0_CH2.</td></tr><tr><td>01011</td><td>Connect to TMR2.</td><td>11011</td><td>Connect to PWM1_CH0.</td></tr><tr><td>01100</td><td>Connect to TMR3.</td><td>11100</td><td>Connect to PWM1_CH2.</td></tr><tr><td colspan="4">Others : Disable to connected any peripheral</td></tr></table>
[7:0]	**Reserved**	**Reserved.**

DMA Service Selection Control Register 1 (DMA_DSSR1)

Register	Offset	R/W	Description	Reset Value
DMA_DSSR1	DMA_BA_GCR+0x08	R/W	DMA Service Selection Control Register 1	0x001F_1F1F

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			CH6_SEL				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			CH5_SEL				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			CH4_SEL				

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:21]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[20:16]</td><td>CH6_SEL</td><td>Channel 6 Selection
This filed defines which peripheral is connected to PDMA channel 6. Software can configure the peripheral setting by CH6_SEL. The channel configuration is the same as CH4_SEL field. Please refer to the explanation of CH4_SEL.</td></tr>
<tr><td>[15:13]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[12:8]</td><td>CH5_SEL</td><td>Channel 5 Selection
This filed defines which peripheral is connected to PDMA channel 5. Software can configure the peripheral setting by CH5_SEL. The channel configuration is the same as CH4_SEL field. Please refer to the explanation of CH4_SEL.</td></tr>
<tr><td>[7:5]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[4:0]</td><td>CH4_SEL</td><td>Channel 4 Selection
This filed defines which peripheral is connected to PDMA channel 4. Software can configure the peripheral by setting CH4_SEL. The channel configuration is the same as CH4_SEL field4’b0000: CH4
<table>
<tr><th>CH4_SEL</th><th>Connection</th><th>CH4_SEL</th><th>Connection</th></tr>
<tr><td>00000</td><td>Connect to SPI0_TX.</td><td>10000</td><td>Connect to SPI0_RX.</td></tr>
<tr><td>00001</td><td>Connect to SPI1_TX.</td><td>10001</td><td>Connect to SPI1_RX.</td></tr>
<tr><td>00010</td><td>Connect to UART0_TX.</td><td>10010</td><td>Connect to UART0_RX.</td></tr>
<tr><td>00011</td><td>Connect to UART1_TX.</td><td>10011</td><td>Connect to UART1_RX.</td></tr>
<tr><td>00100</td><td>Connect to USB_TX.</td><td>10100</td><td>Connect to USB_RX.</td></tr>
<tr><td>00101</td><td>Connect to I^2S_TX.</td><td>10101</td><td>Connect to I^2S_RX.</td></tr>
<tr><td>00110</td><td>Connect to DAC0_TX.</td><td>10110</td><td>Connect to ADC.</td></tr>
<tr><td>00111</td><td>Connect to DAC1_TX.</td><td>10111</td><td>Reserved.</td></tr>
</table></td></tr>
</table>

Bits	Description				
		01000	Connect to SPI2_TX.	11000	Connect to SPI2_RX.
		01001	Connect to TMR0.	11001	Connect to PWM0_CH0.
		01010	Connect to TMR1.	11010	Connect to PWM0_CH2.
		01011	Connect to TMR2.	11011	Connect to PWM1_CH0.
		01100	Connect to TMR3.	11100	Connect to PWM1_CH2.
		Others : Disable to connected any peripheral			

DMA Global Interrupt Status Register (DMA_GCRISR)

Register	Offset	R/W	Description	Reset Value
DMA_GCRISR	DMA_BA_GCR+0x0C	R	DMA Global Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**CRC_INTR**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**INTR6**	**INTR5**	**INTR4**	**INTR3**	**INTR2**	**INTR1**	**INTR0**

Bits	Description	
[31:17]	**Reserved**	**Reserved**
[16]	**CRC_INTR**	**Interrupt Pin Status of CRC Controller** This bit is the Interrupt status of CRC controller **Note:** This bit is read only
[15:7]	**Reserved**	**Reserved**
[6]	**INTR6**	**Interrupt Pin Status Of Channel 6 (Read Only)** This bit is the Interrupt pin status of DMA channel4. **Note:** This bit is read only
[5]	**INTR5**	**Interrupt Pin Status Of Channel 5 (Read Only)** This bit is the Interrupt pin status of DMA channel4. **Note:** This bit is read only
[4]	**INTR4**	**Interrupt Pin Status Of Channel 4 (Read Only)** This bit is the Interrupt pin status of DMA channel4. **Note:** This bit is read only
[3]	**INTR3**	**Interrupt Pin Status Of Channel 3 (Read Only)** This bit is the Interrupt pin status of DMA channel3. **Note:** This bit is read only
[2]	**INTR2**	**Interrupt Pin Status Of Channel 2 (Read Only)** This bit is the Interrupt pin status of DMA channel2. **Note:** This bit is read only
[1]	**INTR1**	**Interrupt Pin Status Of Channel 1 (Read Only)** This bit is the Interrupt pin status of DMA channel1. **Note:** This bit is read only

Bits	Description	
[0]	**INTR0**	**Interrupt Pin Status Of Channel 0 (Read Only)** This bit is the Interrupt pin status of DMA channel0. **Note:** This bit is read only

5.9 Timer Controller

5.9.1 Overview

This chip is equipped with four timer modules including TIMER0, TIMER1, TIMER2 and TIMER3 (TIMER0/1 is at APB1 and TIMER2/3 is at APB2), which allow user to easily implement a counting scheme or timing control for applications. The timer can perform functions like frequency measurement, event counting, interval measurement, clock generation, delay timing, and so on. The timer can generate an interrupt signal upon timeout, or provide the current value of count during operation.

5.9.2 Features

- Independent Clock Source for each Timer (TMRx_CLK, x= 0, 1,2,3)
- Time-out period = (Period of timer clock input) * (8-bit pre-scale counter + 1) * (24-bit TCMP)
- Counting cycle time = (1 / TMRx_CLK) * (2^8) * (2^24)
- Internal 8-bit pre-scale counter
- Internal 24-bit up counter is readable through TDR (Timer Data Register)
- Supports One-shot, Periodic,Output Toggle and Countinuous Counting Operation mode
- Supports external pin capture for interval measurement
- Supports external pin capture for timer counter reset
- Supports Inter-Timer trigger
- Supports Internal trigger event to ADC, DAC and PDMA

5.9.3 Block Diagram

Each timer is equipped with an 8-bit pre-scale counter, a 24-bit up-counter, a 24-bit compare register and an interrupt request signal. Refer to Figure 5.9-1 for the timer controller block diagram. There are five options of clock sources for each timer, Figure 5.9-2 illustrate the Clock Source control function.

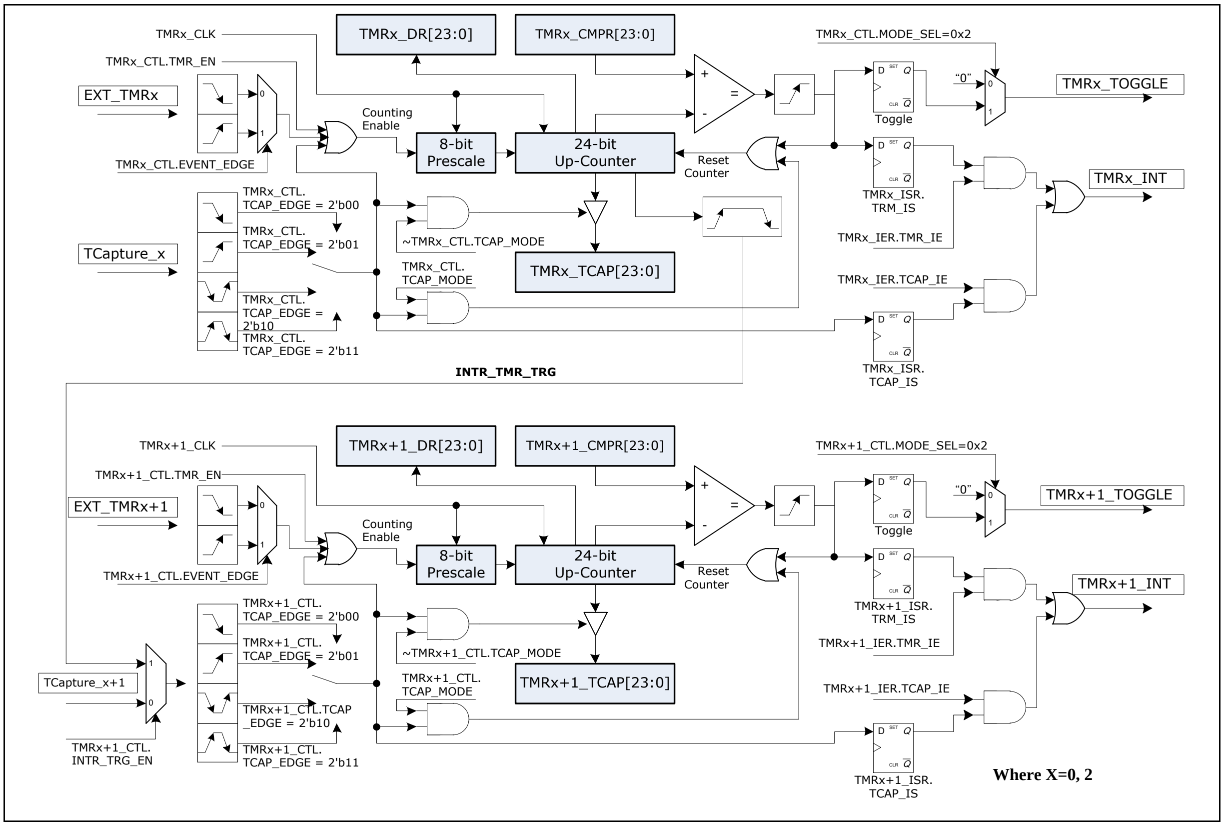


Figure 5.9-1 Timer Controller Block Diagram

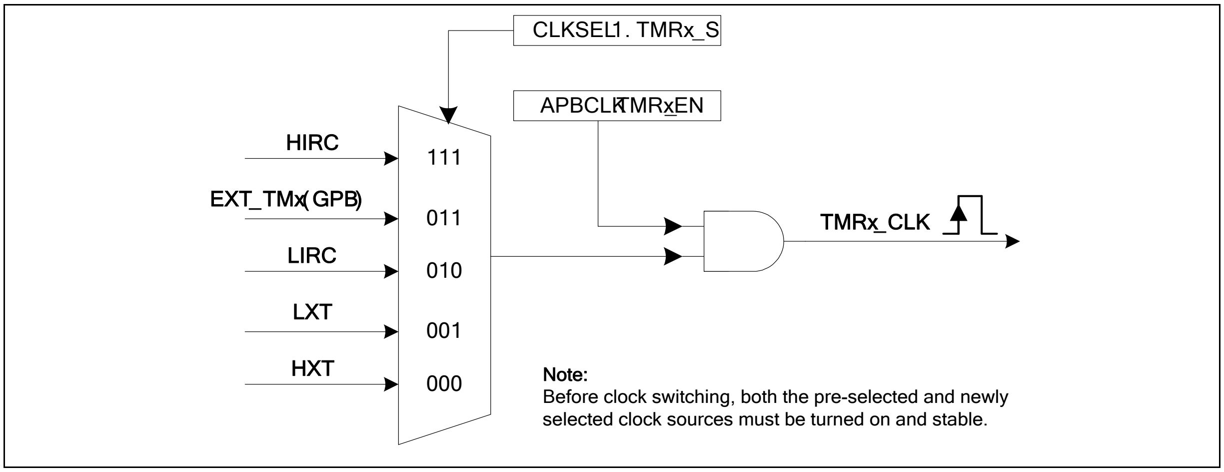


Figure 5.9-2 Timer Clock Controller Diagram

5.9.4 Functional Description

Timer controller provides One-shot, Period, Toggle and Continuous Counting operation modes. The event counting function is also provided to count the events/counts from external pin and external pin capture function for interval measurement or reset timer counter. In addition, timer controller provides the Inter-Timer Trigger Mode to measure input frequency precisely. Each operating function mode is shown as follows:

5.9.4.1 One-shot Mode

If the timer is operated in One-shot mode (MODE_SEL[1:0] is 0x0) and TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1, the timer counter starts up counting. Once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, the TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 1, and TMR_IS (TMRx_ISR[0] timer interrupt status) is 1 then the interrupt signal is generated and sent to NVIC to inform CPU for indicating that the timer counting overflow happens. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 0, no interrupt signal is generated.

In this operating mode, once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1, timer counting operation stops and the timer counter value (TMRx_DR value) goes back to counting initial value then TMR_EN (TMRx_CTL[0] timer counter enable bit) is cleared to 0 by timer controller automatically. That is to say, timer operates timer counting and compares with TMRx_CMPR value function only one time after programming the timer compare register (TMRx_CMPR) value and TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1. So, this operating mode is called One-Shot mode.

5.9.4.2 Periodic Mode

If the timer is operated in Period mode (MODE_SEL[1:0] is 0x1) and TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1, the timer counter starts up counting. Once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, the TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 1, and TMR_IS (TMRx_ISR[0] timer interrupt status) is 1 then the interrupt signal is generated and sent to NVIC to inform CPU for indicating that the timer counting overflow happens. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 0, no interrupt signal is generated.

In this operating mode, once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1, the timer counter value (TMRx_DR value) goes back to counting initial value and TMR_EN (TMRx_CTL[0] timer counter enable bit) is kept at 1 (counting enable continuously) and timer counter operates up counting again. If TMR_IS (TMRx_ISR[0] timer interrupt status) is cleared by software, once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value again, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 also. That is to say, timer operates timer counting and compares with TMRx_CMPR value function periodically. The timer counting operation does not stop until the TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 0. The interrupt signal is also generated periodically. So, this operating mode is called Periodic mode.

5.9.4.3 Toggle Mode

If the timer is operated in Toggle mode (MODE_SEL[1:0] is 0x2) and TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1, the timer counter starts up counting. Once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, the TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 1, and TMR_IS (TMRx_ISR[0] timer interrupt status) is 1 then the interrupt signal is generated and sent to NVIC to inform CPU for indicating that the timer counting overflow happens. If TMR_IE (TMRx_IER[0] timer interrupt enable bit) is set to 0, no interrupt signal is generated.

In this operating mode, once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1, toggle out signal is set to 1, the timer counter value (TMRx_DR value) goes back to counting initial value and TMR_EN (TMRx_CTL[0] timer counter enable bit) is kept at 1 (counting enable continuously) and timer counter operates up counting again. If TMR_IS (TMRx_ISR[0] timer interrupt status) is cleared by software, once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value again, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 also and toggle out signal is set to 0. The timer counting operation does not stop until the TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 0. Thus, the toggle output signal is changing back and forth with 50% duty cycle. So, this operating mode is called Toggle mode.

5.9.4.4 Continuous Counting Mode

If the timer is operated in Continuous Counting mode (MODE_SEL[1:0] is 0x3) and TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1, the timer counter starts up counting. Once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, the TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1. If TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 1, and TMR_IS (TMRx_ISR[0] timer interrupt status) is 1 then the interrupt signal is generated and sent to NVIC to inform CPU for indicating that the timer counting overflow happens. If TMR_EN (TMRx_CTL[0] timer counter enable bit) is set to 0, no interrupt signal is generated.

In this operating mode, once the timer counter value (TMRx_DR value) reaches timer compare register (TMRx_CMPR) value, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 and TMR_EN (TMRx_CTL[0] timer counter enable bit) is kept at 1 (counting enable continuously) and timer counter continuous counting without reload the timer counter value (TMRx_DR value) to counting initial value. User can change different timer compare register (TMRx_CMPR) value immediately without disabling timer counter and restarting timer counter counting.

For example, the timer compare register (TMRx_CMPR) value is set as 80, first. (The timer compare register (TMRx_CMPR) should be less than 2^24 and be greater than 1). Once the timer counter value (TMRx_DR value) reaches to 80, TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 and TMR_EN (TMRx_CTL[0] timer counter enable bit) is kept at 1 (counting enable continuously) and timer counter value (TMRx_DR value) will not goes back to 0, it continues counting to 81, 82, 83,⋯ to (2^24 -1) then 0, 1, 2, 3, ⋯ to 2^24 -1 again and again. Next, if user programs timer compare register (TMRx_CMPR) value as 200 and the TMR_IS (TMRx_ISR[0] timer interrupt status) is cleared to 0, then TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 again when timer counter value (TMRx_DR value) reaches to 200. At last, user programs timer compare register (TMRx_CMPR) value as 500 and clears TMR_IS (TMRx_ISR[0] timer interrupt status) to 0, then TMR_IS (TMRx_ISR[0] timer interrupt status) will set to 1 again when timer counter value (TMRx_DR value) reaches to 500. In this mode, the timer counter value (TMRx_DR value) is keeping up counting always even if TMR_IS (TMRx_ISR[0] timer interrupt status) is 1. So, this operation mode is called as Continuous Counting mode.

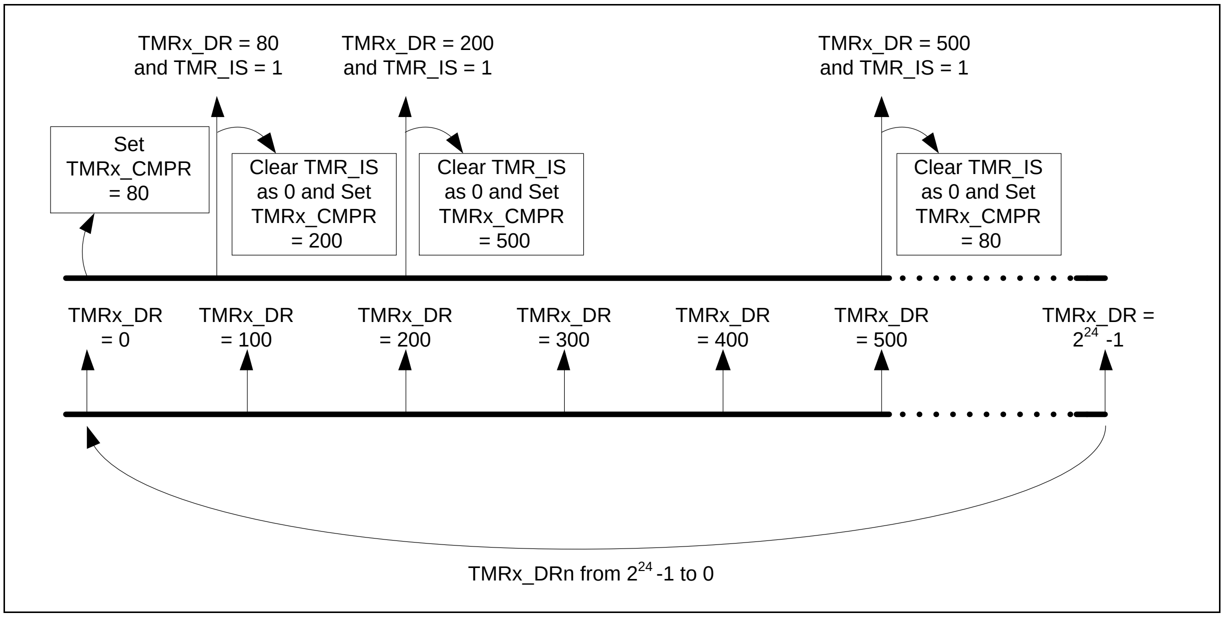


Figure 5.9-3 Timer Clock Controller Diagram

5.9.4.5 Event Counting Mode

An application which can count the events/counts from external event input pin is called as event counting function. In this mode, most of the timer control registers are the same with the timer operating function mode except EVENT_EN (TMRx_CTL[12] event counting mode enable) has to set to 1. When status transition on external event input pin, the event counter value (TMRx_DR value) will be counted according to EVENT_EDGE (TMRx_CTL[13] event counting mode edge selection) setting. EVNT_DEB_EN (TMRx_CTL[14] external event de-bounce enable) bit is for enabled or disabled edge detection de-bounce circuit of external event input pin. The max frequency of event counting source on external event input pin should be less than 1/4 TRMx_CLK if EVNT_DEB_EN (TMRx_CTL[14] external event de-bounce enable) is 0 or less than 1/10 TRMx_CLK if EVNT_DEB_EN (TMRx_CTL[14] external event de-bounce enable) is 1. Otherwise, the event counter value (TMRx_DR value) will not be counted normally.

5.9.4.6 Timer Counter Capture/Reset Function

In this mode, Timer will monitor the transition of external pin to save the 24-bit counter value or reset the 24-bit counter.

If TCAP_MODE is 0, the transition on external pin is used as timer counter capture function. In this mode, if CAP_CNT_MOD is 0, the free-counting mode, 24-bit up-counting timer will keep counting continuously. And when the transition of external pin matches the TCAP_EDGE setting, the value of 24-bit up-counting timer will be saved into register TMRx_TCAP. If CAP_CNT_MOD is 1, the trigger-counting mode, 24-bit up-counting timer will keep its value at zero. Once the transition of external pin matches the 1st transition of TCAP_EDGE setting, the 24-bit up-counting timer will start counting. And then if the transition of external pin matc^{he}s the 2nd transition of TCAP_EDGE setting, the 24-bit up-counting timer will stop counting. And its value will be saved into register TMRx_TCAP.

If TCAP_MODE is 1, the transition on external pin is used as timer counter reset function. In this mode, once the transition of external pin matches the TCAP_EDGE setting, the 24-bit up-counting timer will be reset.

To detect the transition of external pin, the timer circuit implements the de-bounce for external pin. Based on the result of de-bounce circuit and external pin, the rising-edge or falling-edge could be detected. The reset value of de-bounce circuit is “0” and the de-bounce would only active when both

TCAP_DEB_EN and TCAP_EN are enabled. So, if the external pin level is “1” and TCAP_EDGE is set to detect rising-edge of external pin, then after de-bounce circuit active (TCAP_DEB_EN is “1” and TCAP_EN is “1”), a false rising-edge would be detected. This would result in the incorrect capture data (TMRx_TCAP) while 1st time the TCAP_IS is set. To avoid this incorrect capture data to affect the capture application, discard this 1st capture data is necessary and recommended.

5.9.4.7 Inter-Timer Trigger Mode

In this mode, the TMRx (where x=0 or 2), will be forced in counter mode, counting with external event, and will generate a signal (INTR_TMR_TRG) to trigger TMRx+1 (where x=0 or 2). The TMRx+1 will be forced in trigger-counting mode of capture function.

While INTR_TRG_EN is set, the TMRx will make a rising-edge transition of trigger signal (INTR_TMR_TRG) to TMRx+1 while 24-bit counter is counting from 0x0 to 0x1. And when 24-bit counter reaches the 24-bit TCMPR value, the TMRx will make a falling-edge transition of trigger signal (INTR_TMR_TRG) to TMRx+1.

When INTR_TMR_TRG transited from low to high (rising-edge), the 24-bit counter of TMRx+1 will start to count. Also, when INTR_TMR_TRG transited from high to low (falling-edge), the 24-bit counter of TMRx+1 will stop counting. At the same time, the value of 24-bit counter will be saved into TMRx+1_TCAP.

The example listed below is to show how inter-timer trigger mode work. When inter-timer trigger mode is enabled (TMRx_CTL.INTR_TRG_EN = 1), the TMRx_Counter starts to up-counting in each external event (TMRx) detected. When TRMx_Counter counted from 0x0 to 0x1, the INTR_TMR_TRG is set high, and TMRx+1_Counter is then start counting in each TMRx+1_CLK. When TMRx_Counter reaches the value of TMRx_CMPR, it stops counting and ITNR_TMR_TRG is set low. And then, the TMRx+1_Counter stops counting, too. In the same time, the value of TMRx+1_Counter is sampled into TMRx+1_TCAP and interrupt status TMRx_ISR.TMR_IS is also set. In addition, the TRMx_CTL.INTR_TRG_EN is also automatically cleared by H/W.

By using Inter-timer Trigger mode, the period of external event (TMRx) could be measured more precisely.

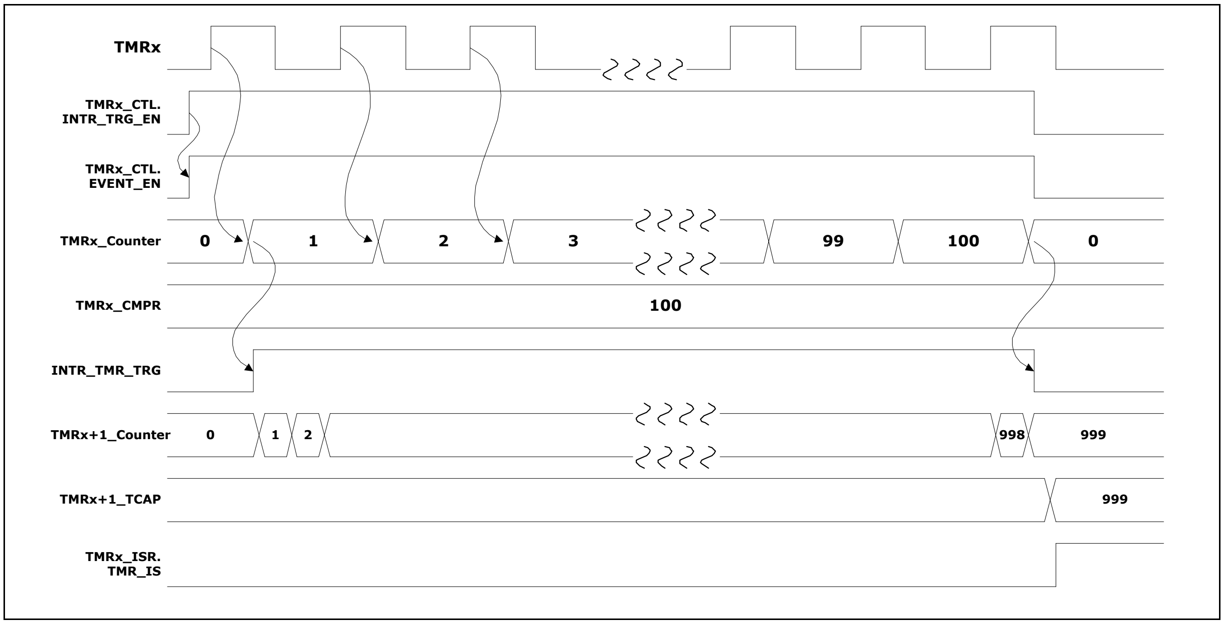


Figure 5.9-4 Inter-Timer Trigger Mode

5.9.5 Register and Memory Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
TMR Base Address: **TMR0_BA = 0x4001_0000** **TMR1_BA = 0x4001_0100** **TMR2_BA = 0x4011_0000** **TMR3_BA = 0x4011_0100**				
TMR_CTL **x=0,1,2,3**	TMRx_BA+0x000	R/W	Timer x Control Register	0x0000_0000
TMR_PRECNT **x=0,1,2,3**	TMRx_BA+0x004	R/W	Timer x Pre-Scale Counter Register	0x0000_0000
TMR_CMPR **x=0,1,2,3**	TMRx_BA+0x008	R/W	Timer x Compare Register	0x0000_0000
TMR_IER **x=0,1,2,3**	TMRx_BA+0x00C	R/W	Timer x Interrupt Enable Register	0x0000_0000
TMR_ISR **x=0,1,2,3**	TMRx_BA+0x010	R/W	Timer x Interrupt Status Register	0x0000_0000
TMR_DR **x=0,1,2,3**	TMRx_BA+0x014	R	Timer x Data Register	0x0000_0000
TMR_TCAP **x=0,1,2,3**	TMRx_BA+0x018	R	Timer x Capture Data Register	0x0000_0000
TMR Base Address: **TMR0_BA = 0x4001_0000**				
GPA_SHADOW	TMR0_BA+0x200	R	GPIO Port A Pin Value Shadow Register	0x0000_XXXX
GPB_SHADOW	TMR0_BA+0x204	R	GPIO Port B Pin Value Shadow Register	0x0000_XXXX
GPC_SHADOW	TMR0_BA+0x208	R	GPIO Port C Pin Value Shadow Register	0x0000_XXXX
GPD_SHADOW	TMR0_BA+0x20C	R	GPIO Port D Pin Value Shadow Register	0x0000_XXXX
GPE_SHADOW	TMR0_BA+0x210	R	GPIO Port E Pin Value Shadow Register	0x0000_XXXX
GPF_SHADOW	TMR0_BA+0x214	R	GPIO Port F Pin Value Shadow Register	0x0000_XXXX

5.9.6 Register Description

Timer x Control Register (TMRx_CTL)

Register	Offset	R/W	Description	Reset Value
TMR_CTL x=0,1,2,3	TMRx_BA+0x000	R/W	Timer x Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							INTR_TRG_EN
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved	TCAP_DEB_EN	Reserved	CAP_CNT_MOD	TCAP_EDGE		TCAP_MODE	TCAP_EN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
-	EVNT_DEB_EN	EVENT_EDGE	EVENT_EN	CAP_TRG_EN	PDMA_TEEN	DAC_TEEN	ADC_TEEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TMR_ACT	Reserved	MODE_SEL		DBGACK_EN	WAKE_EN	SW_RST	TMR_EN

Bits	Description	
[31:25]	**Reserved**	**Reserved**
[24]	**INTR_TRG_EN**	**Inter-Timer Trigger Mode Enable** This bit controls if Inter-timer Trigger mode is enabled. If Inter-timer Trigger mode is enabled, the TMRx will be in counter mode and counting with external Clock Source or event. And, TMRx+1 will be in trigger-counting mode of capture function. 0 = Inter-timer trigger mode Disabled. 1 = Inter-timer trigger mode Enabled. **Note:** For TMRx+1_CTL, this bit is ignored and the read back value is always 1'b0.
[23]	**Reserved**	**Reserved**
[22]	**CAP_DEB_EN**	**Tcapture Pin De-bounce Enable** When CAP_DEB_EN is set, the Tcapture pin de-bounce circuit will be enabled to eliminate the bouncing of the signal. In de-bounce circuit the Tcapture pin signal will be sampled 4 times by TMRx_CLK. 0 = De-bounce circuit Disabled. 1 = De-bounce circuit Enabled. **Note:** When TCAP_EN is enabled, enable this bit is recommended. And, while TCAP_EN is disabled, disable this bit is recommended to save power consumption.
[21]	**Reserved**	**Reserved**

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[20]</td><td>CAP_CNT_MOD</td><td>Timer Capture Counting Mode Selection
This bit indicates the behavior of 24-bit up-counting timer while TCAP_EN is set to high.
If this bit is 0, the free-counting mode, the behavior of 24-bit up-counting timer is defined by MODE_SEL field. When TCAP_EN is set, TCAP_MODE is 0, and the transition of Tcapture pin matches the TCAP_EDGE setting, the value of 24-bit up-counting timer will be saved into register TMRx_TCAPn.
If this bit is 1, Trigger-counting mode, 24-bit up-counting timer will be not counting and keep its value at zero. When TCAP_EN is set, TCAP_MODE is 0, and once the transition of external pin matches the 1st transition of TCAP_EDGE setting, the 24-bit up-counting timer will start counting. And then if the transition of external pin matches the 2nd transition of TCAP_EDGE setting, the 24-bit up-counting timer will stop counting. And its value will be saved into register TMRx_TCAPn.
0 = Capture with free-counting timer mode.
1 = Capture with trigger-counting timer mode.
Note: For TMRx+1_CTL, if INTR_TRG_EN is set, the CAP_CNT_MOD will be forced to high, the capture with Trigger-counting Timer mode.</td></tr>
<tr><td>[19:18]</td><td>TCAP_EDGE</td><td>Tcapture Pin Edge Detect Selection
This field defines that active transition of Tcapture pin is for timer counter reset function or for timer capture function.
For timer counter reset function and free-counting mode of timer capture function, the configurations are:
<table>
<tr><th>TCAP_EDGE</th><th>Output Clock (MCLK)</th></tr>
<tr><td>0x0</td><td>A falling edge (1 to 0 transition) on Tcapture pin is an active transition.</td></tr>
<tr><td>0x1</td><td>A rising edge (0 to 1 transition) on Tcapture pin is an active transition.</td></tr>
<tr><td>0x2</td><td>Both falling edge (1 to 0 transition) and rising edge (0 to 1 transition) on Tcapture pin are active transitions.</td></tr>
<tr><td>0x3</td><td>Both falling edge (1 to 0 transition) and rising edge (0 to 1 transition) on Tcapture pin are active transitions.</td></tr>
</table>
For trigger-counting mode of timer capture function, the configurations are:
<table>
<tr><th>TCAP_EDGE</th><th>Output clock (MCLK)</th></tr>
<tr><td>0x0</td><td>1st falling edge on Tcapture pin triggers 24-bit timer to start counting, while 2nd falling edge triggers 24-bit timer to stop counting.</td></tr>
<tr><td>0x1</td><td>1st rising edge on Tcapture pin triggers 24-bit timer to start counting, while 2nd rising edge triggers 24-bit timer to stop counting.</td></tr>
<tr><td>0x2</td><td>Falling edge on Tcapture pin triggers 24-bit timer to start counting, while rising edge triggers 24-bit timer to stop counting.</td></tr>
<tr><td>0x3</td><td>Rising edge on Tcapture pin triggers 24-bit timer to start counting, while falling edge triggers 24-bit timer to stop counting.</td></tr>
</table>
Note: For TMRx+1_CTL, if INTR_TRG_EN is set, the TCAP_EDGE will be forced to 11.</td></tr>
</table>

Bits	Description	
[17]	TCAP_MODE	**Tcapture Pin Function Mode Selection** This bit indicates if the transition on Tcapture pin is used as timer counter reset function or timer capture function. 0 = The transition on Tcapture pin is used as timer capture function. 1 = The transition on Tcapture pin is used as timer counter reset function. **Note:** For TMRx+1_CTL, if INTR_TRG_EN is set, the TCAP_MODE will be forced to low.
[16]	TCAP_EN	**Tcapture Pin Functional Enable** This bit controls if the transition on Tcapture pin could be used as timer counter reset function or timer capture function. 0 = The transition on Tcapture pin is ignored. 1 = The transition on Tcapture pin will result in the capture or reset of 24-bit timer counter. **Note:** For TMRx_CTL, if INTR_TRG_EN is set, the TCAP_EN will be forced to low and the Tcapture pin transition is ignored. **Note:** For TMRx+1_CTL, if INTR_TRG_EN is set, the TCAP_EN will be forced to high.
[15]	-	**Reserved**
[14]	EVNT_DEB_EN	**External Event De-bounce Enable** When EVNT_DEB_EN is set, the external event pin de-bounce circuit will be enabled to eliminate the bouncing of the signal. In de-bounce circuit the external event pin will be sampled 4 times by TMRx_CLK. 0 = De-bounce circuit Disabled. 1 = De-bounce circuit Enabled. **Note:** When EVENT_EN is enabled, enable this bit is recommended. And, while EVENT_EN is disabled, disable this bit is recommended to save power consumption.
[13]	EVENT_EDGE	**Event Counting Mode Edge Selection** This bit indicates which edge of external event pin enabling the timer to increase 1. 0 = A falling edge of external event enabling the timer to increase 1. 1 = A rising edge of external event enabling the timer to increase 1.
[12]	EVENT_EN	**Event Counting Mode Enable** When EVENT_EN is set, the increase of 24-bit up-counting timer is controlled by external event pin. While the transition of external event pin matches the definition of EVENT_EDGE, the 24-bit up-counting timer increases by 1. Or, the 24-bit up-counting timer will keep its value unchanged. 0 = Timer counting is not controlled by external event pin. 1 = Timer counting is controlled by external event pin. **Note:** When EVENT_EN is enabled, user can not choose EXT_TMx(GPB) as clock source. However, the speed of chosen clock must 3 times greater than the speed of EXT_TMx(GPB).

Bits	Description	
[11]	**CAP_TRG_EN**	**TCAP_IS Trigger Mode Enable** This bit controls if the TMR_IS or TCAP_IS is used to trigger PDMA, DAC and ADC while TMR_IS or TCAP_IS is set. If this bit is low and TMR_IS is set, timer will generate an internal trigger event to PDMA, DAC or ADC while related trigger enable bit (PDMA_TEEN, DAC_TEEN or ADC_TEEN) is also set. If this bit is set high and TCAP_IS is set, timer will generate an internal trigger event to PDMA, DAC or ADC while related trigger enable bit (PDMA_TEEN, DAC_TEEN or ADC_TEEN) is also set. 0 = TMR_IS is used to trigger PDMA, DAC and ADC. 1 = TCAP_IS is used to trigger PDMA, DAC and ADC.
[10]	**PDMA_TEEN**	**TMR_IS or TCAP_IS Trigger PDMA Enable** This bit controls if TMR_IS or TCAP_IS could trigger PDMA. When PDMA_TEEN is set, TMR_IS is set and the CAP_TRG_EN is low, the timer controller will generate an internal trigger event to PDMA controller. When PDMA_TEEN is set, TCAP_IS is set and the CAP_TRG_EN is high, the timer controller will generate an internal trigger event to PDMA controller. 0 = TMR_IS or TCAP_IS trigger PDMA Disabled. 1 = TMR_IS or TCAP_IS trigger PDMA Enabled.
[9]	**DAC_TEEN**	**TMR_IS or TCAP_IS Trigger DAC Enable** This bit controls if TMR_IS or TCAP_IS could trigger DAC. When DAC_TEEN is set, TMR_IS is set and the CAP_TRG_EN is low, the timer controller will generate an internal trigger event to DAC controller. When DAC_TEEN is set, TCAP_IS is set and the CAP_TRG_EN is high, the timer controller will generate an internal trigger event to DAC controller. 0 = TMR_IS or TCAP_IS trigger DAC Disabled. 1 = TMR_IS or TCAP_IS trigger DAC Enabled.
[8]	**ADC_TEEN**	**TMR_IS or TCAP_IS Trigger ADC Enable** This bit controls if TMR_IS or TCAP_IS could trigger ADC. When ADC_TEEN is set, TMR_IS is set and the CAP_TRG_EN is low, the timer controller will generate an internal trigger event to ADC controller. When ADC_TEEN is set, TCAP_IS is set and the CAP_TRG_EN is high, the timer controller will generate an internal trigger event to ADC controller. 0 = TMR_IS or TCAP_IS trigger ADC Disabled. 1 = TMR_IS or TCAP_IS trigger ADC Enabled.
[7]	**TMR_ACT**	**Timer Active Status Bit (Read Only)** This bit indicates the timer counter status of timer. 0 = Timer is not active. 1 = Timer is in active.
[6]	**Reserved**	**Reserved**

<table>
<tr><th>Bits</th><th colspan="3">Description</th></tr>
<tr><td rowspan="5">[5:4]</td><td rowspan="5">MODE_SEL</td><td colspan="2">Timer Operating Mode Select</td></tr>
<tr><td>MODE_SEL</td><td>Output clock (MCLK)</td></tr>
<tr><td>0x0</td><td>The timer is operating in the one-shot mode.
In this mode, the associated interrupt signal is generated (if TMR_IER [TMR_IE] is enabled) once the value of 24-bit up counter equals the TMRx_CMPR. And TMR_CTL [TMR_EN] is automatically cleared by hardware.</td></tr>
<tr><td>0x1</td><td>The timer is operating in the periodic mode.
In this mode, the associated interrupt signal is generated periodically (if TMR_IER [TMR_IE] is enabled) while the value of 24-bit up counter equals the TMRx_CMPR. After that, the 24-bit counter will be reset and starts counting from zero again.</td></tr>
<tr><td>0x2</td><td>The timer is operating in the periodic mode with output toggling.
In this mode, the associated interrupt signal is generated periodically (if TMR_IER [TMR_IE] is enabled) while the value of 24-bit up counter equals the TMRx_CMPR. After that, the 24-bit counter will be reset and starts counting from zero again.
At the same time, timer controller will also toggle the output pin TMRx_TOG_OUT to its inverse level (from low to high or from high to low).
Note: The default level of TMRx_TOG_OUT after reset is low.</td></tr>
<tr><td></td><td></td><td>0x3</td><td>The timer is operating in continuous counting mode.
In this mode, the associated interrupt signal is generated when TMR_DR = TMR_CMPR (if TMR_IER [TMR_IE] is enabled). However, the 24-bit up-counter counts continuously without reset.</td></tr>
<tr><td>[3]</td><td>DBGACK_EN</td><td colspan="2">ICE Debug Mode Acknowledge Ineffective Enable
0 = ICE debug mode acknowledgement effects TIMER counting and TIMER counter will be held while ICE debug mode acknowledged.
1 = ICE debug mode acknowledgement is ineffective and TIMER counter will keep going no matter ICE debug mode acknowledged or not.</td></tr>
<tr><td>[2]</td><td>WAKE_EN</td><td colspan="2">Wake-up Enable
When WAKE_EN is set and the TMR_IS or TCAP_IS is set, the timer controller will generate a wake-up trigger event to CPU.
0 = Wake-up trigger event Disabled.
1 = Wake-up trigger event Enabled.</td></tr>
<tr><td>[1]</td><td>SW_RST</td><td colspan="2">Software Reset
Set this bit will reset the timer counter, pre-scale counter and also force TMR_CTL [TMR_EN] to 0.
0 = No effect.
1 = Reset Timer’s pre-scale counter, internal 24-bit up-counter and TMR_CTL [TMR_EN] bit.
Note: This bit will be auto cleared and takes at least 3 TMRx_CLK clock cycles.</td></tr>
</table>

Bits	Description	
[0]	**TMR_EN**	**Timer Counter Enable Bit** 0 = Stops/Suspends counting 1 = Starts counting **Note1:** Set TMR_EN to 1 enables 24-bit counter keeps up counting from the last stop counting value. **Note2:** This bit is auto-cleared by hardware in one-shot mode (MODE_SEL [5:4] =2'b00) once the value of 24-bit up counter equals the TMRx_CMPR.

Timer x Pre-scale Counter Register (TMRx_PRECNT)

Register	Offset	R/W	Description	Reset Value
TMR_PRECNT x=0,1,2,3	TMRx_BA+0x004	R/W	Timer x Pre-Scale Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PRESCALE_CNT							

Bits	Description	
[31:8]	**Reserved**	**Reserved**
[7:0]	**PRESCALE_CNT**	**Pre-scale Counter** Clock input is divided by PRESCALE_CNT + 1 before it is fed to the counter. If PRESCALE_CNT =0, then there is no scaling.

Timer x Compare Register (TMRx_CMPR)

Register	Offset	R/W	Description	Reset Value
TMR_CMPR **x=0,1,2,3**	TMRx_BA+0x008	R/W	Timer x Compare Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TMR_CMP							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TMR_CMP							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TMR_CMP							

Bits	Description	
[31:24]	**Reserved**	**Reserved**
[23:0]	**TMR_CMP**	**Timer Compared Value** TMR_CMP is a 24-bit compared register. When the internal 24-bit up-counter counts and its value is equal to TMR_CMP value, a Timer Interrupt is requested if the timer interrupt is enabled with TMR_IER [TMR_IE] is enabled. The TMR_CMP value defines the timer counting cycle time. Time-out period = (Period of timer clock input) * (8-bit PRESCALE_CNT + 1) * (24-bit TMR_CMP) **Note1:** Never write 0x0 or 0x1 in TMR_CMP, or the core will run into unknown state. **Note2:** No matter TMR_CTL [TMR_EN] is 0 or 1, whenever software write a new value into this register, TIMER will restart counting using this new value and abort previous count.

Timer x Interrupt Enable Register (TMRx_IER)

Register	Offset	R/W	Description	Reset Value
TMR_IER x=0,1,2,3	TMRx_BA+0x00C	R/W	Timer x Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved						TCAP_IE	TMR_IE

Bits	Description	
[31:2]	**Reserved**	**Reserved**
[1]	**TCAP_IE**	**Timer Capture Function Interrupt Enable** 0 = Timer External Pin Function Interrupt Disabled 1 = Timer External Pin Function Interrupt Enabled **Note:** If timer external pin function interrupt is enabled, the timer asserts its interrupt signal when the TCAP_EN is set and the transition of external pin matches the TCAP_EDGE setting
[0]	**TMR_IE**	**Timer Interrupt Enable** 0 = Timer Interrupt Disabled. 1 = Timer Interrupt Enabled. **Note:** If timer interrupt is enabled, the timer asserts its interrupt signal when the associated counter is equal to TMR_CMPR.

Timer x Interrupt Status Register (TMRx_ISR)

Register	Offset	R/W	Description	Reset Value
TMR_ISR x=0,1,2,3	TMRx_BA+0x010	R/W	Timer x Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		NCAP_DET_STS	TMR_Wake_STS	Reserved		TCAP_IS	TMR_IS

Bits	Description	
[31:6]	**Reserved**	**Reserved**
[5]	**NCAP_DET_STS**	**New Capture Detected Status** This status is to indicate there is a new incoming capture event detected before CPU clearing the TCAP_IS status. If the above condition occurred, the Timer will keep register TMRx_CAP unchanged and drop the new capture value. This bit is also cleared to 0 while TCAP_IS is cleared. 0 = New incoming capture event didn't detect before CPU clearing TCAP_IS status. 1 = New incoming capture event detected before CPU clearing TCAP_IS status.
[4]	**TMR_Wake_STS**	**Timer Wake-up Status** If timer causes CPU wakes up from power-down mode, this bit will be set to high. It must be cleared by software with a write 1 to this bit. 0 = Timer does not cause system wake-up. 1 = Wakes system up from power-down mode by Timer timeout.
[3:2]	**-**	**Reserved**
[1]	**TCAP_IS**	**Timer Capture Function Interrupt Status** This bit indicates the external pin function interrupt status of Timer. This bit is set by hardware when TCAP_EN is set high, and the transition of external pin matches the TCAP_EDGE setting. Write 1 to clear this bit to zero. If this bit is active and TCAP_IE is enabled, Timer will trigger an interrupt to CPU.

Bits	Description	
[0]	**TMR_IS**	**Timer Interrupt Status** This bit indicates the interrupt status of Timer. This bit is set by hardware when the up counting value of internal 24-bit counter matches the timer compared value (TMR_CMPR). Write 1 to clear this bit to 0. If this bit is active and TMR_IE is enabled, Timer will trigger an interrupt to CPU.

Timer x Data Register (TMRx_DR)

Register	Offset	R/W	Description	Reset Value
TMR_DR **x=0,1,2,3**	TMRx_BA+0x014	R	Timer x Data Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TDR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TDR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TDR							

Bits	Description	
[31:24]	**Reserved**	**Reserved**
[23:0]	**TDR**	**Timer Data Register** User can read this register for internal 24-bit timer up-counter value.

Timer x Capture Data Register (TMRx_TCAP)

Register	Offset	R/W	Description	Reset Value
TMR_TCAP **x=0,1,2,3**	TMRx_BA+0x018	R	Timer x Capture Data Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CAP							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CAP							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CAP							

Bits	Description	
[31:24]	**Reserved**	**Reserved**
[23:0]	**CAP**	**Timer Capture Data Register** When TCAP_EN is set, TCAP_MODE is 0, and the transition of external pin matches the TCAP_EDGE setting, the value of 24-bit up-counting timer will be saved into register TMRx_TCAP. User can read this register for the counter value.

GPIO Port x Pin Value Shadow Register (GPx_SHADOW)

Register	Offset	R/W	Description	Reset Value
GPA_SHADOW	TMR0_BA+0x200	R	GPIO Port A Pin Value Shadow Register	0x0000_XXXX
GPB_SHADOW	TMR0_BA+0x204	R	GPIO Port B Pin Value Shadow Register	0x0000_XXXX
GPC_SHADOW	TMR0_BA+0x208	R	GPIO Port C Pin Value Shadow Register	0x0000_XXXX
GPD_SHADOW	TMR0_BA+0x20C	R	GPIO Port D Pin Value Shadow Register	0x0000_XXXX
GPE_SHADOW	TMR0_BA+0x210	R	GPIO Port E Pin Value Shadow Register	0x0000_XXXX
GPF_SHADOW	TMR0_BA+0x214	R	GPIO Port F Pin Value Shadow Register	0x0000_XXXX

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PIN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PIN							

Bits	Description	
[31:16]	**Reserved**	**Reserved**
[n] n = 0,1..15	**PIN**	**GPIO Port [A/B/C/D/E/F] Pin Values** The value read from each of these bit reflects the actual status of the respective GPI/O pin. These registers are shadow registers of GPIOx_PIN register. **Note:** For GPF_SHADOW, bits [15:9] are reserved.

5.10 Pulse Width Modulation (PWM)

5.10.1 Overview

This chip has two PWM controllers, each controller has 4 independent PWM outputs, CH0~CH3, or as 2 complementary PWM pairs, (CH0, CH1), (CH2, CH3) with 2 programmable dead-zone generators.

Each two PWM outputs, (CH0, CH1), (CH2, CH3), share the same 8-bit prescaler, clock divider providing 5 divided frequencies (1, 1/2, 1/4, 1/8, 1/16). Each PWM output has independent 16-bit PWM down-count counter for PWM period control, and 16-bit comparators for PWM duty control. Each dead-zone generator has two outputs. The first dead-zone generator output is CH0 and CH1, and for the second dead-zone generator, the output is CH2 and CH3. The 2 sets of PWM controller total provide eight independent PWM interrupt flags which are set by hardware when the corresponding PWM period down counter reaches zero. PWM interrupt will be asserted when both PWM interrupt source and its corresponding enable bit are active. Each PWM output can be configured as one-shot mode to produce only one PWM cycle signal or continuous mode to output PWM waveform continuously.

When DZEN01 of PWMx_CTL is set, CH0 and CH1 perform complementary PWM paired function; the paired PWM timing, period, duty and dead-time are determined by PWM channel 0 timer and Dead-zone generator 0. Similarly, When DZEN23 of PWMx_CTL is set the complementary PWM pair of (CH2, CH3) is controlled by PWM channel 2.

To prevent PWM driving output pin with unsteady waveform, the 16-bit period down counter and 16-bit comparator are implemented with double buffer. When user writes data to counter/comparator buffer registers the updated value will be loaded into the 16-bit down counter/ comparator at the time down counter reaching zero. The double buffering feature avoids glitch at PWM outputs.

When the 16-bit period down counter reaches zero, the interrupt request is generated. If PWM output is set as continuous mode, when the down counter reaches zero, it is reloaded with CN of PWMx_DUTYy(y=0~3) Register automatically then start decreases, repeatedly. If the PWM output is set as one-shot mode, the down counter will stop and generate one interrupt request when it reaches zero.

The value of PWM counter comparator is used for pulse width modulation. The counter control logic changes the output level when down-counter value matches the value of compare register.

The alternate feature of the PWM is digital input capture function. If capture function is enabled the PWM output pin is switched as capture input pin. The capture channel 0 and PWM CH0 share one timer; and the capture channel 1 and PWM CH1 share one timer, and etc. Therefore user must setup the PWM timer before enabling capture feature. After capture feature is enabled, the capture always latches PWM timer to Capture Rising Latch Register (PWMx_CRLy) where y=0~3, when input channel has a rising transition and latches PWM timer to Capture Falling Latch Register (PWMx_CFLy) where y=0~3, when input channel has a falling transition. Capture channel 0 interrupt is programmable by setting PWMx_CAPINTEN. Whenever Capture event latched for channel 0/1/2/3, the PWM timer 0/1/2/3 will be reload at this moment if the corresponding reload enable bit specified in CAPCTL are set.

The maximum captured frequency that PWM can capture is dominated by the capture interrupt latency. When capture interrupt occurs, software will do at least three steps, they are: Read PWMINTSTS to tell it from interrupt source and Read PWMx_CRLy/PWMx_CFLy(y=0~3) to get capture value and finally write 1 to clear PWMx_INTSTS. If interrupt latency will take time T0 to finish, the capture signal mustn't transient during this interval. In this case, the maximum capture frequency will be 1/T0.

5.10.2 Features

5.10.2.1 PWM Function:

- Two PWM controllers, each controller having 4 independent PWM outputs, CH0~CH3, or as 2 complementary PWM pairs, (CH0, CH1), (CH2, CH3) with 2 programmable dead-zone generators
- Up to 8 PWM channels or 4 PWM paired channels
- Up to 16 bits PWM counter width
- PWM Interrupt request synchronous with PWM period
- Single-shot or Continuous mode
- Four Dead-Zone generators

5.10.2.2 Capture Function:

- Timing control logic shared with PWM timer.
- 8 Capture input channels shared with 8 PWM output channels.
- Each channel supports one rising latch register (PWMx_CRLy), one falling latch register (PWMx_CFLy) and Capture interrupt flag (CAPIFy) where x=0~1,y=0~3.
- Eight 16-bit counters for eight capture channels or four 32-bit counter for four capture channels when cascade is enabled: when CH01CASKEN is set, the original 16-bit counter of channel 1 will combine with channel 0's 16 bit counter for channel 0 input capture counting and so does CH23CASKEN for channel 2, 3
- Supports PDMA transfer function for PWMx channel 0, 2

5.10.3 Block Diagram

The following figures illustrate the architecture of PWM in groups. (Timer 0&1 are in one group and timer 2&3 are in another, and so on.)

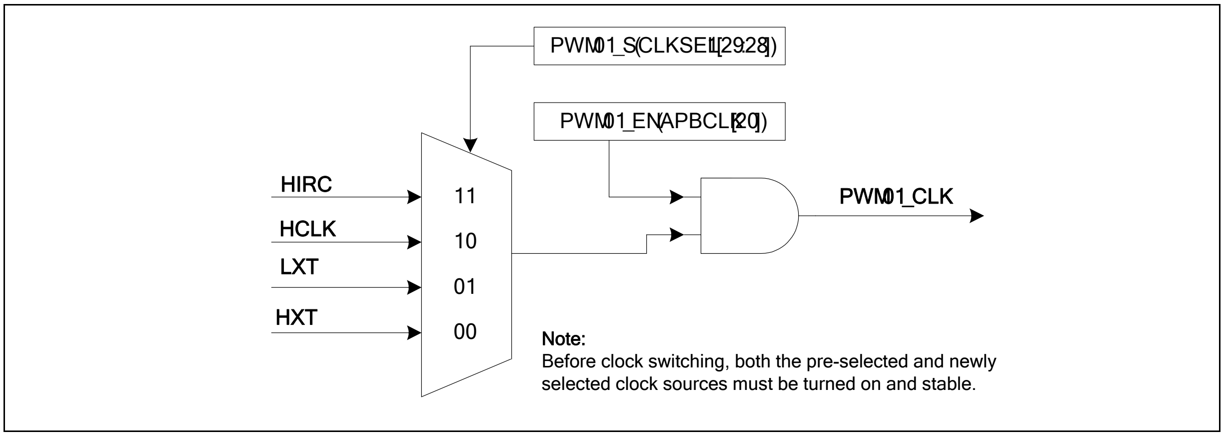


Figure 5.10-1 PWM0 Clock

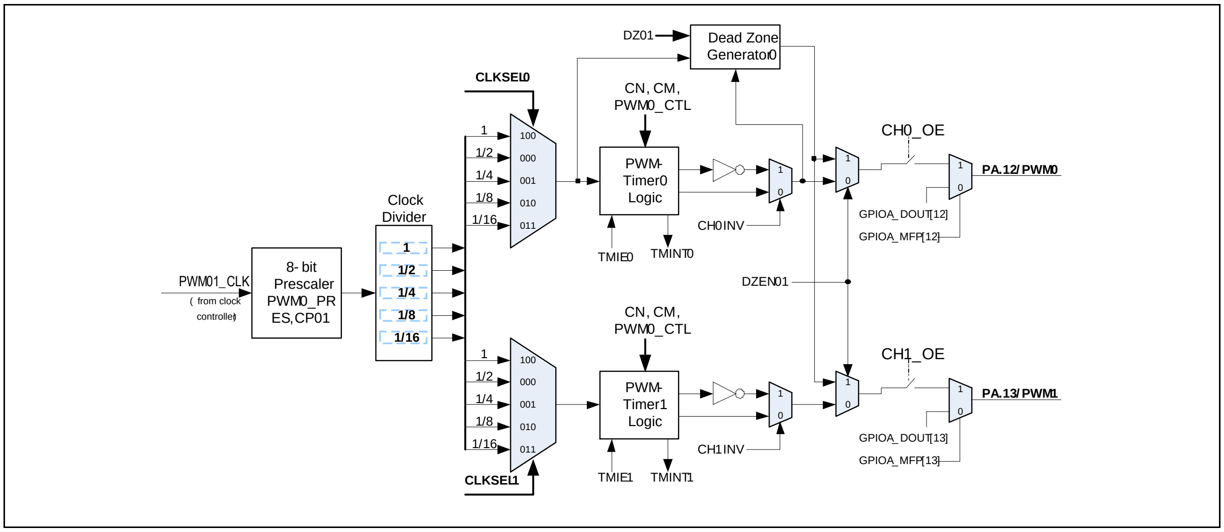


Figure 5.10-2 PWM0 Generator for Channel 0, 1

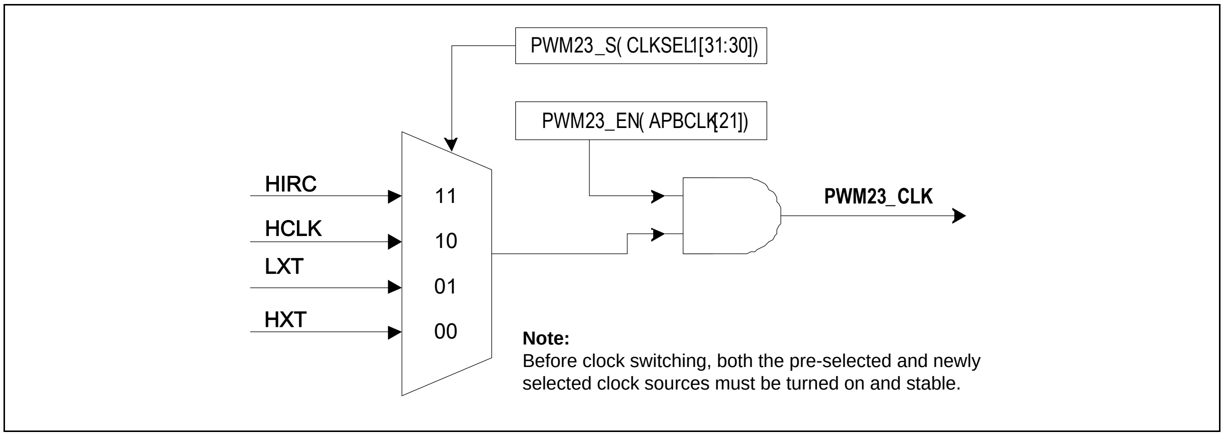


Figure 5.10-3 PWM1 Clock

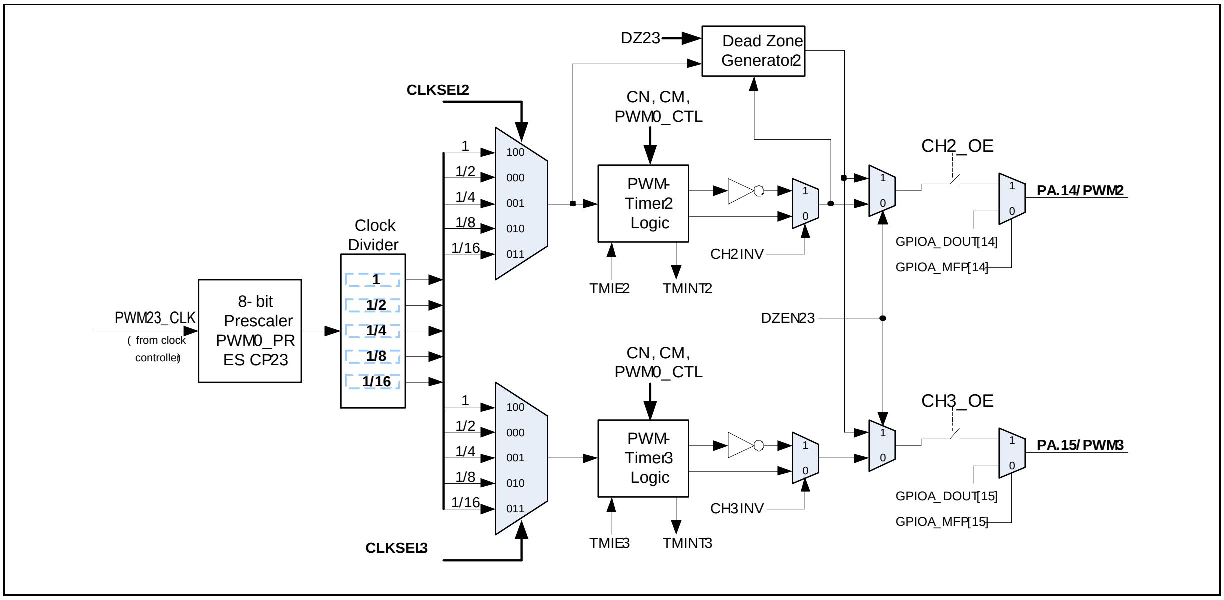


Figure 5.10-4 PWM Generator for Channel 2, 3

5.10.4 Functional Description

5.10.4.1 PWM-Timer Operation

The PWM period and duty control are decided by PWMx_DUTYy (y =0~3) register CN (PWMx_DUTYy[15:0]) and CM (PWMx_DUTYy[31:16]). The PWM-timer timing operation is shown in Figure 5.10-5. The pulse width modulation follows the formula below and the legend of PWM-Timer Comparator is shown in Figure 5.10-6. Note that the corresponding I/O pins must be configured as output type before PWM function is enabled.

PWM frequency = PWMxy_CLK/(prescale+1)*(clock divider)/(CN+1); where xy, could be 01, 23, depending on selected PWM channel.

Duty ratio = (CM+1)/(CN+1).

CM >= CN: PWM output is always high.

CM < CN: PWM low width= (CN-CM) unit1; PWM high width = (CM+1) unit.

If CM = 0: PWM low width = (CN) unit; PWM high width = 1 unit

Note: 1. Unit = one PWM clock cycle.

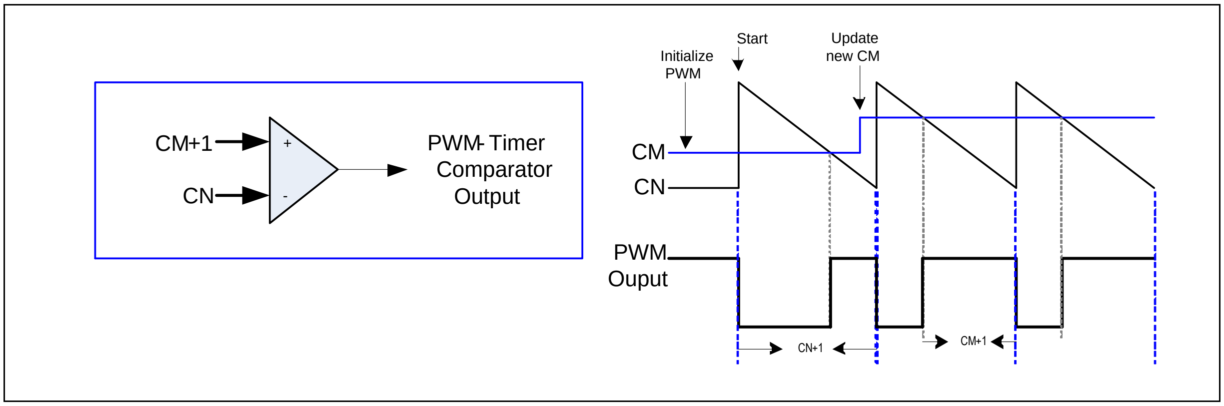


Figure 5.10-5 Legend of Internal Comparator Output of PWM-Timer

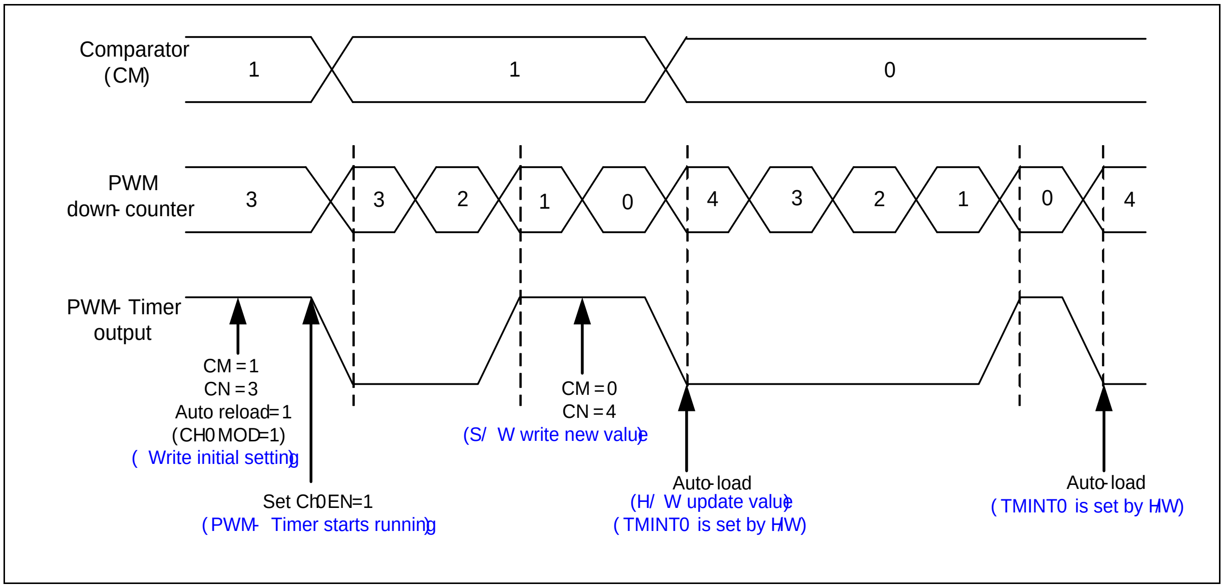


Figure 5.10-6 PWM-Timer Operation Timing for Channel 0

5.10.4.2 PWM Double Buffering, Continuous and One-shot Operation

The PWM has double buffering function; the reload value is updated at the start of next period without affecting current timer operation. The PWM counter value can be written into bit [15:0] of PWMx_DUTY0~3.

The bit CH0MOD in PWM Control Register (PWMx_CTL) defines PWM operation in Continuous or One-shot mode If CH0MOD is set to one (continuous mode), the controller loads CN to PWM counter when PWM counter reaches zero. If CN is set to zero, PWM counter will be halt when PWM counter counts to zero.

In one-shot mode (CH0MOD=0), the corresponding channel will output only one duty waveform and counter will be stopped if no further corresponding duty register updated. When PWM counter is running, updating corresponding duty register will engage the next duty waveform.

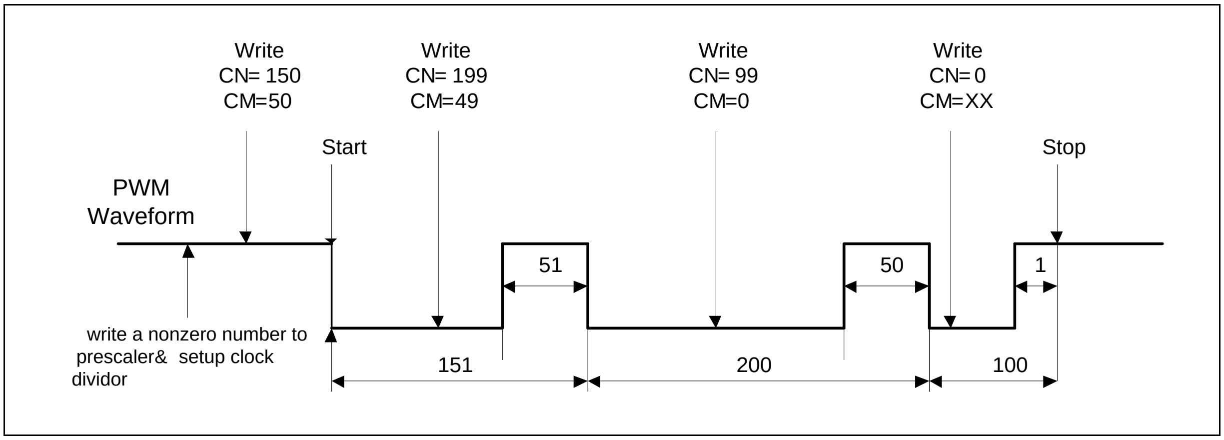


Figure 5.10-7 PWM Double Buffer Illustration

5.10.4.3 Modulate Duty Ratio

The double buffering function allows CM to be written at any point in current cycle. The loaded value will take effect from next cycle.

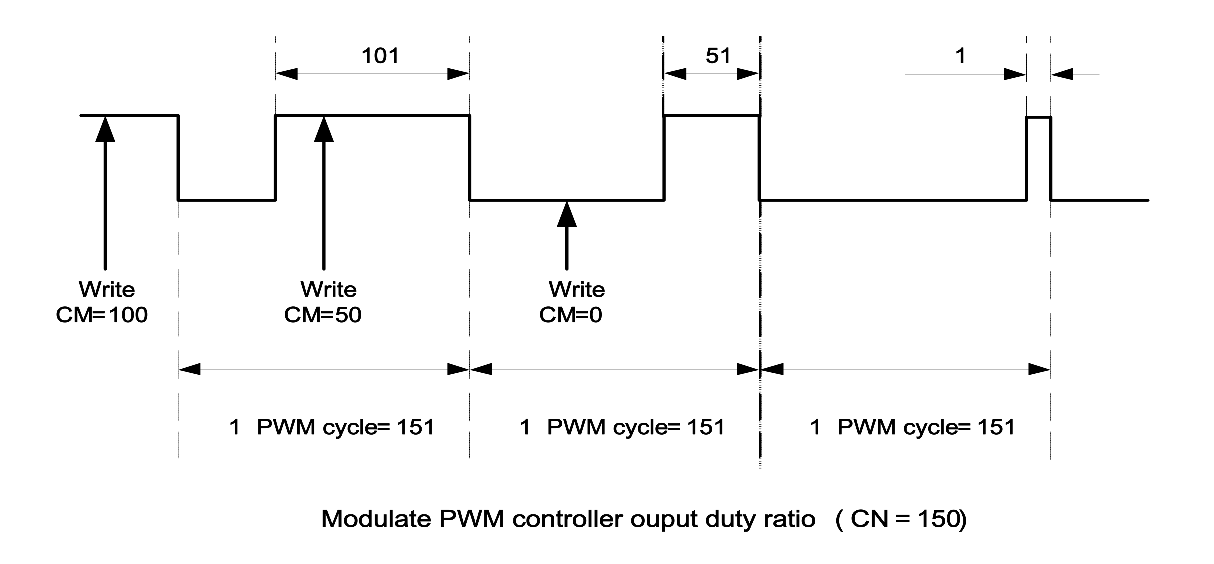


Figure 5.10-8 PWM Controller Output Duty Ratio

5.10.4.4 Dead-Zone Generator

PWM implements Dead Zone generator. They are built for power device protection. This function generates a programmable time gap to delay PWM rising output. User can program Dead-Zone counter to determine the Dead Zone interval.

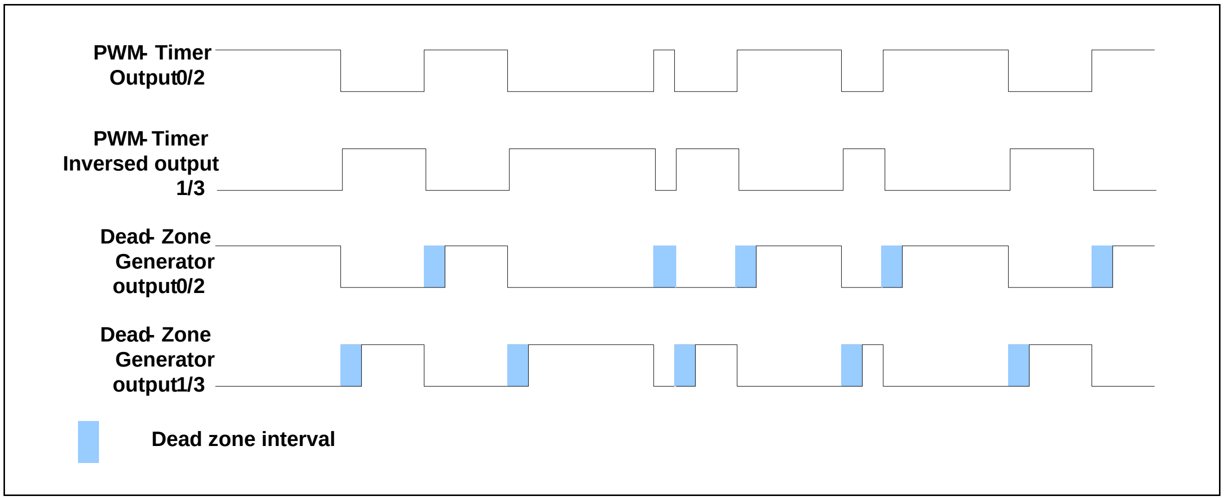


Figure 5.10-9 Paired PWM Output with Dead Zone Generation Operation

5.10.4.5 Capture Operation

The Capture channel 0 and PWM channel 0 share one timer ; and the Capture channel1 and PWM channel 1 share another timer, and etc. The capture always latches PWM-timer to PWMx_CRL0 when input channel has a rising transition and latches PWM-timer to PWMx_CFL0 when input channel has a falling transition. Capture channel 0 interrupt is programmable by setting PWMx_CAPINTEN[0] (Rising latch Interrupt enable) and PWM_CAPINTEN[1] (Falling latch Interrupt enable) to decide the condition of interrupt occur. Whenever the Capture module issues a capturing flag (rising latched or falling latched) that are defined in CAPUNTSTS, and the reload enable bit(defined in CAPCTL) is also set the corresponding PWM timer will be reloaded with CN at this moment. Note that the corresponding I/O pins must be configured as input type before Capture function is enabled.

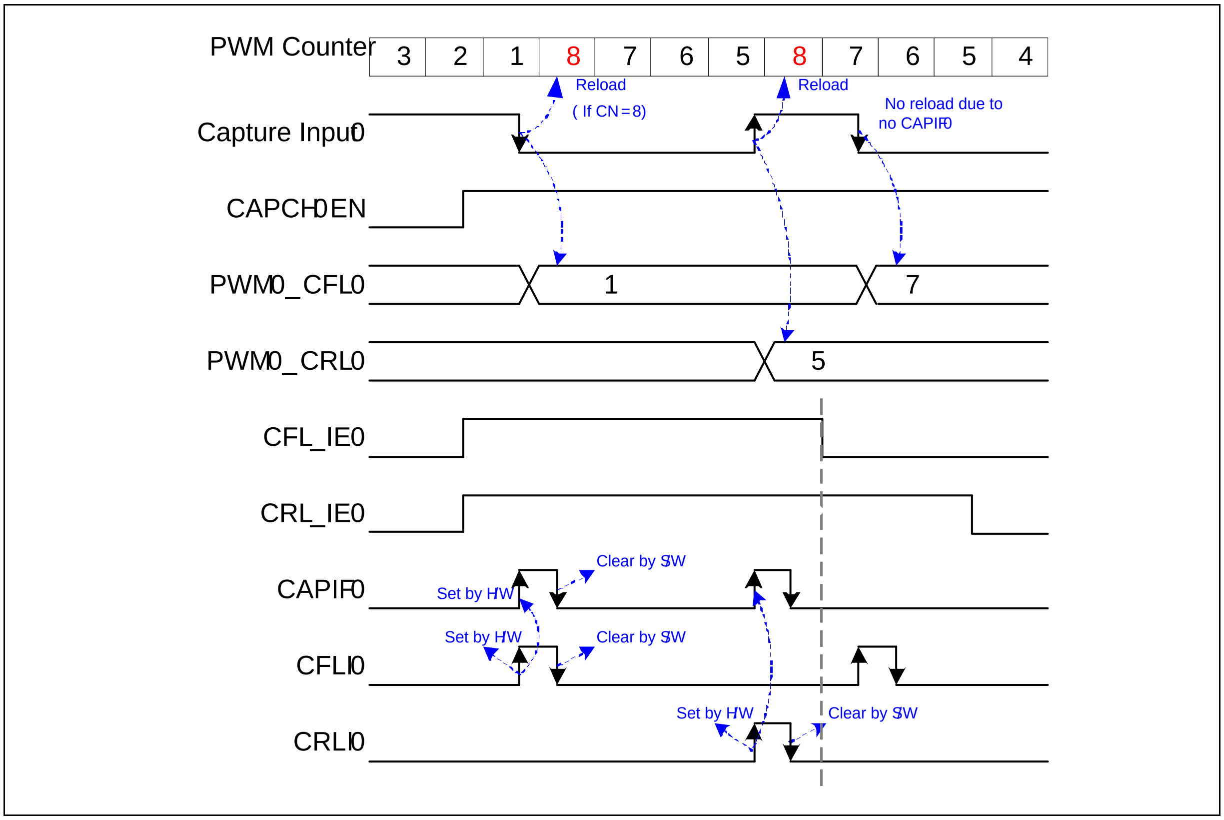


Figure 5.10-10 PWM Capture Operation Timing

At this example, the CN is 8:

The PWM timer will be reloaded with CN when a capture interrupt flag (CAPIF0) is set.

The channel low pulse widt– is (CN – CRL).

The channel high pulse widt– is (CN – CFL).

In some case that need wider counter, user can cascade two 16 bit counters to 32 bit counter for capturing

The cascade method is depicted below

When enabling CH01CASK in bit[13] of PWMx_CAPCTL, the internal cascade logic will combine CH0's 16 bit counter with CH1's 16 bit counter to become 32 bit counter for CH0 and the same for enabling CH23CASK in bit[29] of PWMx_CAPCTL for CH2 and CH3.CNR0 /CNR2 and CNR1/CNR3 are also cascaded. At this case, the capturing function for CH1 and CH3 are useless.

When capturing flag is setup (rising for CAPINTSTS[1],and falling for CAPINTSTS[2]), the capture data for rising latched is stored in CRLR0 and CRLR1 and CFLR1 and CFLR0 for falling latched.

CRLR1 is located in the upper half word and CRLR0 is in lower half word and the same for CFLR1 and CFLR0.

User can also read CRLR0 or CFLR0 register directly to get 32 bit capturing data when cascade is enabled

Note: Cascade function is only for PWM capture function.

5.10.4.6 PWM PDMA Function

PWM support PDMA transfer function when operating in capture mode and is only for specified channel (channel 0,2),when the corresponding PDMA enable bit(defined in CAPCTL register) is set, capture module will issue a request to PDMA controller when the preceding capture event happened. PDMA controller will issue ACK to capture module and read back PDMACH0 register to memory. By setting PDMACAPMOD0 and PDMACAPMOD2, PDMA can transfer rising latched data or falling latched data or both of them to memory. When using PDMA to transfer both falling and rising data, remember to set CHxRFORDER in PWMx_CAPCTL to decide the order of transferring data (falling edge latched is first or rising edged latched first)

5.10.4.7 PWM-Timer Interrupt Architecture

There are eight PWM, interrupts, PWM0CH0_INT ~ PWMCH3_INT, PWM1CH0_INT ~ PWM1CH3_INT which are OR into PWM0_INT and PWM1_INT. PWM CH0 and Capture channel 0 share one interrupt, PWM CH1 and Capture channel 1 share the same interrupt and so on. Therefore, PWM function and Capture function in the same channel cannot be used at the same time.

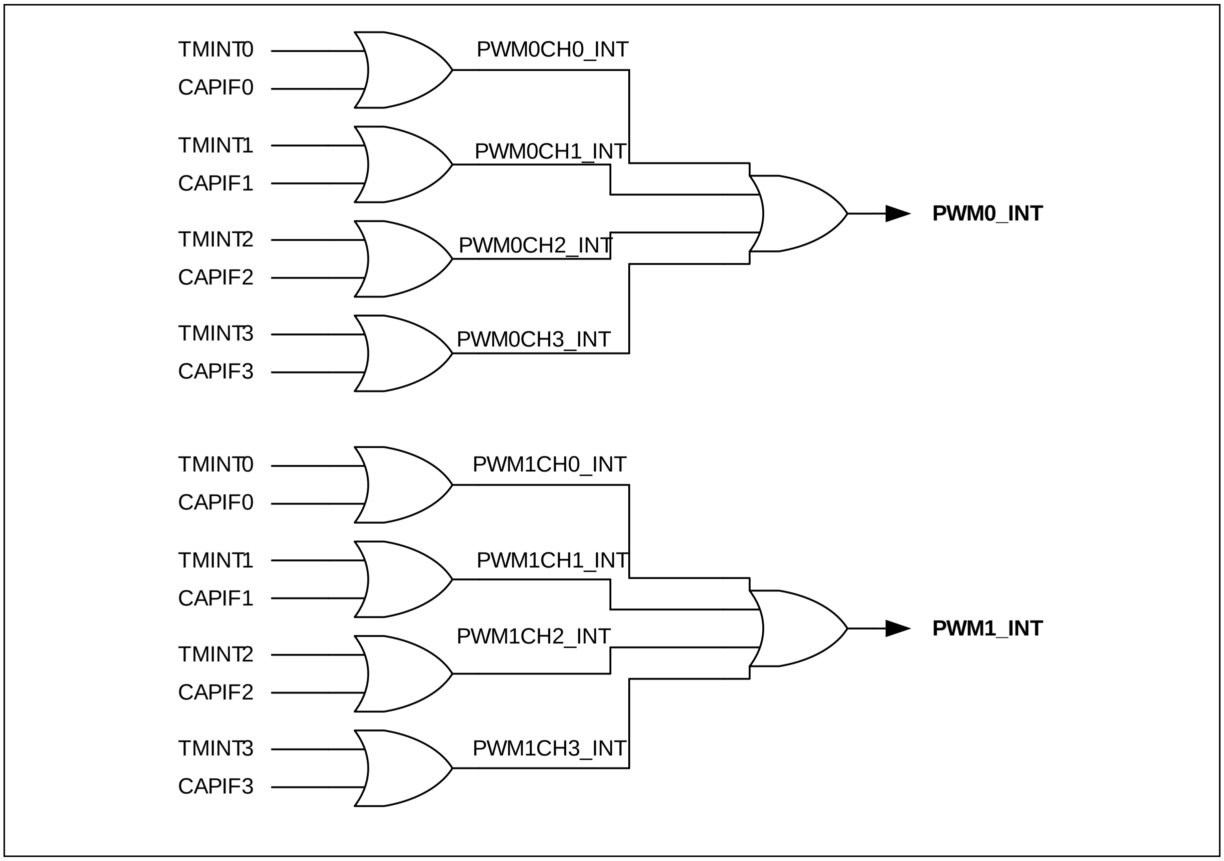


Figure 5.10-11 PWM-Timer Interrupt

5.10.4.8 PWM-Timer Start Procedure

The following procedure is for starting a PWM drive.

- Setup clock selector (PWMx_CLKSEL),x=0~1
- Setup prescaler (PWMx_PRES),x=0~1
- Setup inverter on/off, dead zone generator on/off, auto-reload/one-shot mode and Stop PWM-timer (PWMx_CTL, x=0~1)
- Setup interrupt enable register (PWMx_INTEN),x=0~1
- Setup PWM output enable (PWMx_OE),x=0~1
- Setup the corresponding GPI/O pins to PWM function
- Setup the corresponding GPI/O pins to output type
- Enable PWM down-counter start running (Set ChxEN = 1 in PWMx_CTL, x=0~1)
- Setup CM and CN of PWMx_DUTYy register for setting PWM duty, x=0~1,y=0~3.(When cascade is enabled the CM is used for the upper half word of the 32 bit CN)
- The procedure 1~8 mentioned above may be set up not in the order and PWM Timer can still work fine

5.10.4.9 PWM-Timer Stop Procedure

Take PWM0, Channel 0 for example:

Method 1:

Set 32 bit PWMx_DUTY0 register to 0, and wait for PWM timeout interrupt (if bit 0 of PWMx_IE is set) occurring or polling the corresponding time-out flag. When PWM timeout interrupt occurred or the flag is set, disable PWM-Timer (CH0EN in PWMx_CTL). (Recommended)

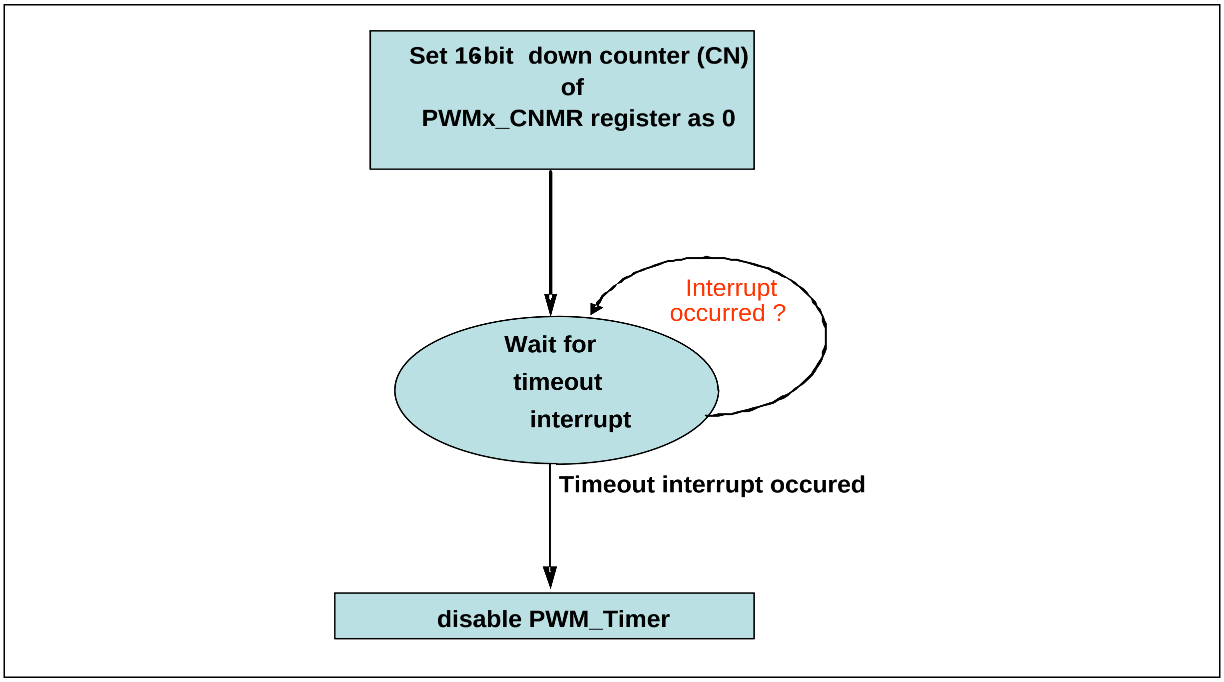


Figure 5.10-12 PWM-Timer Stop Method 1

Method 2:

Disable PWM-Timer directly ((ChxEN in PWMx_CTL). (Not recommended)

The reason is that disabling ChxEN will immediately stop PWM output signal and lead to change the duty of the PWM output, this may cause damage to the control circuit of motor.

5.10.4.10 Capture Start Procedure

- Setup clock selector (PWMx_CLKSEL),x=0~1
- Setup prescaler (PWMx_PRES),x=0~1
- Setup channel enabled, rising/falling interrupt enable and input signal inverter on/off (PWMx_CAPCTL,PWMx_CAPINTEN),x=0~1
- Setup PWM down-counter (CN) of PWMx_DUTYy, x= 0~1, y=0~3 register
- Set Capture Input Enable Register (PWMx_CAPCTL),x=0~1
- Setup the corresponding GPI/O pins to PWM function
- Setup the corresponding GPI/O pins to input type
- Enable PWM down-counter start running (Set ChyEN = 1 in PWMx_CTL), x=0~1,y=0~3

5.10.5 Register and Memory Map

R: read only, **W**: write only, **R/W**: both read and written

Register	Offset	R/W	Description	Reset Value
PWM Base Address: **PWM0_BA = 0x4004_0000** **PWM1_BA = 0x4014_0000**				
PWM_PRES **x=0,1**	PWMx_BA+0x000	R/W	PWM Prescaler Register	0x0000_0000
PWM_CLKSEL **x=0,1**	PWMx_BA+0x004	R/W	PWM Clock Select Register	0x0000_0000
PWM_CTL **x=0,1**	PWMx_BA+0x008	R/W	PWM Control Register	0x0000_0000
PWM_INTEN **x=0,1**	PWMx_BA+0x00C	R/W	PWM Interrupt Enable Register	0x0000_0000
PWM_INTSTS **x=0,1**	PWMx_BA+0x010	R/W	PWM Interrupt Indication Register	0x0000_0010
PWM_OE **x=0,1**	PWMx_BA+0x014	R/W	PWM Output Enable for PWM0~PWM3	0x0000_0000
PWM_DUTY0 **x=0,1**	PWMx_BA+0x01C	R/W	PWM Counter/Comparator Register 0	0x0000_0000
PWM_DATA0 **x=0,1**	PWMx_BA+0x020	R	PWM Data Register 0	0x0000_0000
PWM_DUTY1 **x=0,1**	PWMx_BA+0x028	R/W	PWM Counter/Comparator Register 1	0x0000_0000
PWM_DATA1 **x=0,1**	PWMx_BA+0x02C	R	PWM Data Register 1	0x0000_0000
PWM_DUTY2 **x=0,1**	PWMx_BA+0x034	R/W	PWM Counter/Comparator Register 2	0x0000_0000
PWM_DATA2 **x=0,1**	PWMx_BA+0x038	R	PWM Data Register 2	0x0000_0000
PWM_DUTY3 **x=0,1**	PWMx_BA+0x040	R/W	PWM Counter/Comparator Register 3	0x0000_0000
PWM_DATA3 **x=0,1**	PWMx_BA+0x044	R	PWM Data Register 3	0x0000_0000
PWM_CAPCTL **x=0,1**	PWMx_BA+0x054	R/W	Capture Control Register	0x0000_0000
PWM_CAPINTEN **x=0,1**	PWMx_BA+0x058	R/W	Capture interrupt enable Register	0x0000_0000

Register	Offset	R/W	Description	Reset Value
PWM_CAPINTSTS **x=0,1**	PWMx_BA+0x05C	R/W	Capture Interrupt Indication Register	0x0000_0000
PWM_CRL0 **x=0,1**	PWMx_BA+0x060	R	Capture Rising Latch Register (Channel 0)	0x0000_0000
PWM_CFL0 **x=0,1**	PWMx_BA+0x064	R	Capture Falling Latch Register (Channel 0)	0x0000_0000
PWM_CRL1 **x=0,1**	PWMx_BA+0x068	R	Capture Rising Latch Register (Channel 1)	0x0000_0000
PWM_CFL1 **x=0,1**	PWMx_BA+0x06C	R	Capture Falling Latch Register (Channel 1)	0x0000_0000
PWM_CRL2 **x=0,1**	PWMx_BA+0x070	R	Capture Rising Latch Register (Channel 2)	0x0000_0000
PWM_CFL2 **x=0,1**	PWMx_BA+0x074	R	Capture Falling Latch Register (Channel 2)	0x0000_0000
PWM_CRL3 **x=0,1**	PWMx_BA+0x078	R	Capture Rising Latch Register (Channel 3)	0x0000_0000
PWM_CFL3 **x=0,1**	PWMx_BA+0x07C	R	Capture Falling Latch Register (Channel 3)	0x0000_0000
PWM_PDMACH0 **x=0,1**	PWMx_BA+0x080	R	PDMA channel 0 captured data	0x0000_0000
PWM_PDMACH2 **x=0,1**	PWMx_BA+0x084	R	PDMA channel 2 captured data	0x0000_0000

5.10.6 Register Description

PWM Pre-scale Register

Register	Offset	R/W	Description	Reset Value
PWM_PRES x=0,1	PWMx_BA+0x000	R/W	PWM Prescaler Register	0x0000_0000

31	30	29	28	27	26	25	24
DZ23							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DZ01							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CP23							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CP01							

Bits		Description
[31:24]	**DZ23**	**Dead Zone Interval Register for CH2 and CH3 Pair** These 8 bits determine dead zone length. The unit time of dead zone length is received from clock selector 2.
[23:16]	**DZ01**	**Dead Zone Interval Register for CH0 and CH1 Pair** These 8 bits determine dead zone length. The unit time of dead zone length is received from clock selector 0.
[15:8]	**CP23**	**Clock Prescaler 2 for PWM Timer 2 & 3** Clock input is divided by (CP23 + 1) before it is fed to the counter 2 & 3 If CP23=0, the prescaler 2 output clock will be stopped. So PWM counter2 and 3 will be stopped also.
[7:0]	**CP01**	**Clock Prescaler 0 for PWM Timer 0 & 1** Clock input is divided by (CP01 + 1) before it is fed to the counter 0 & 1 If CP01 =0, the prescaler 0 output clock will be stopped. So PWM counter 0 and 1 will be stopped also.

PWM Clock Selector Register (PWMx_CLKSEL)

Register	Offset	R/W	Description	Reset Value
PWM_CLKSEL **x=0,1**	PWMx_BA+0x004	R/W	PWM Clock Select Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	CLKSEL3			Reserved	CLKSEL2		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	CLKSEL1			Reserved	CLKSEL0		

Bits	Description	
[31:15]	**Reserved**	**Reserved**
[14:12]	**CLKSEL3**	**Timer 3 Clock Source Selection** **Select clock input for timer 3.** **CLKSEL3** = **Input Clock Divided by**: 100 = 1 011 = 16 010 = 8 001 = 4 000 = 2
[10:8]	**CLKSEL2**	**Timer 2Clock Source Selection** Select clock input for timer 2. (Table is the same as CLKSEL3)
[6:4]	**CLKSEL1**	**Timer 1 Clock Source Selection** Select clock input for timer 1. (Table is the same as CLKSEL3)
[2:0]	**CLKSEL0**	**Timer 0 Clock Source Selection** Select clock input for timer 0. (Table is the same as CLKSEL3)

PWM Control Register (PWMx_CTL)

Register	Offset	R/W	Description	Reset Value
PWM_CTL **x=0,1**	PWMx_BA+0x008	R/W	PWM Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				CH3MOD	CH3INV	Reserved	CH3EN
23	22	21	20	19	18	17	16
Reserved				CH2MOD	CH2INV	Reserved	CH2EN
15	14	13	12	11	10	9	8
Reserved				CH1MOD	CH1INV	Reserved	CH1EN
7	6	5	4	3	2	1	0
Reserved		DZEN23	DZEN01	CH0MOD	CH0INV	Reserved	CH0EN

Bits	Description	
[27]	**CH3MOD**	**PWM-Timer 3 Continuous/One-shot Mode** 1 = Continuous Mode 0 = One-Shot Mode **Note:** If there is a rising transition at this bit, it will cause CN and CM of PWM0_DUTY3 to be cleared.
[26]	**CH3INV**	**PWM-Timer 3 Output Inverter ON/OFF** 1 = Inverter ON 0 = Inverter OFF
[24]	**CH3EN**	**PWM-Timer 3 Enable/Disable Start Run** 1 = PWM-Timer 3 Start Run Enabled. 0 = PWM-Timer 3 Running Stopped.
[19]	**CH2MOD**	**PWM-Timer 2 Continuous/One-shot Mode** 1 = Continuous Mode 0 = One-Shot Mode **Note:** If there is a rising transition at this bit, it will cause CN and CM of PWM0_DUTY2 be cleared.
[18]	**CH2INV**	**PWM-Timer 2 Output Inverter ON/OFF** 1 = Inverter ON 0 = Inverter OFF
[16]	**CH2EN**	**PWM-Timer 2 Enable/Disable Start Run** 1 = PWM-Timer 2 Start Run Enabled. 0 = PWM-Timer 2 Running Stopped.
[11]	**CH1MOD**	**PWM-Timer 1 Continuous/One-shot Mode**

Bits	Description	
		1 = Continuous Mode 0 = One-Shot Mode **Note:** If there is a rising transition at this bit, it will cause CN and CM of PWM0_DUTY1 to be cleared.
[10]	**CH1INV**	**PWM-Timer 1 Output Inverter ON/OFF** 1 = Inverter ON 0 = Inverter OFF
[8]	**CH1EN**	**PWM-Timer 1 Enable/Disable Start Run** 1 = PWM-Timer 1 Start Run Enabled. 0 = PWM-Timer 1 Running Stopped.
[5]	**DZEN23**	**Dead-Zone 2 Generator Enable/Disable** 1 = Enabled 0 = Disabled **Note:** When Dead-Zone Generator is enabled, the pair of PWM2 and PWM3 becomes a complementary pair.
[4]	**DZEN01**	**Dead-Zone 0 Generator Enable/Disable** 1 = Enabled 0 = Disabled **Note:** When Dead-Zone Generator is enabled, the pair of PWM0 and PWM1 becomes a complementary pair.
[3]	**CH0MOD**	**PWM-Timer 0 Continuous/One-Shot Mode** 1 = Continuous Mode 0 = One-Shot Mode **Note:** If there is a rising transition at this bit, it will cause CN and CM of PWM0_DUTY0 to be cleared.
[2]	**CH0INV**	**PWM-Timer 0 Output Inverter ON/OFF** 1 = Inverter ON 0 = Inverter OFF
[0]	**CH0EN**	**PWM-Timer 0 Enable/Disable Start Run** 1 = PWM-Timer 0 Start Run Enabled. 0 = PWM-Timer 0 Running Stopped.

PWM Interrupt Enable Register (PWMx_INTEN)

Register	Offset	R/W	Description	Reset Value
PWM_INTEN **x=0,1**	PWMx_BA+0x00C	R/W	PWM Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				TMIE3	TMIE2	TMIE1	TMIE0

Bits	Description	
[31:4]	**Reserved**	**Reserved.**
[3]	**TMIE3**	**PWM Timer 3 Interrupt Enable** 1 = Enabled 0 = Disabled
[2]	**TMIE2**	**PWM Timer 2 Interrupt Enable** 1 = Enabled 0 = Disabled
[1]	**TMIE1**	**PWM Timer 1 Interrupt Enable** 1 = Enabled 0 = Disabled
[0]	**TMIE0**	**PWM Timer 0 Interrupt Enable** 1 = Enabled 0 = Disabled

PWM Interrupt Flag Register (PWMx_INTSTS)

Register	Offset	R/W	Description	Reset Value
PWM_INTSTS x=0,1	PWMx_BA+0x010	R/W	PWM Interrupt Indication Register	0x0000_0010

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							PresSyncFlag
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Duty3Syncflag	Duty2Syncflag	Duty1Syncflag	Duty0Syncflag	TMINT3	TMINT2	TMINT1	TMINT0

Bits	Description	
[8]	**PresSyncFlag**	**Prescale Synchronize Flag** 1 = Prescale is synchronizing to ECLK domain 0 = Prescale has been synchronized to ECLK domain **Note:** software should check this flag when writing Prescale, if this flag is set, and user ignore this flag and change Prescale, the Prescale may be wrong for one prescale cycle
[7]	**Duty3Syncflag**	**Duty3 Synchronize Flag** 1 = Duty3 is synchronizing to ECLK domain 0 = Duty3 has been synchronized to ECLK domain **Note:** software should check this flag when writing duty3, if this flag is set, and user ignore this flag and change duty3, the corresponding CNR and CMR may be wrong for one duty cycle
[6]	**Duty2Syncflag**	**Duty2 Synchronize Flag** 1 = Duty2 is synchronizing to ECLK domain 0 = Duty2 has been synchronized to ECLK domain **Note:** software should check this flag when writing duty2, if this flag is set, and user ignore this flag and change duty2, the corresponding CNR and CMR may be wrong for one duty cycle
[5]	**Duty1Syncflag**	**Duty1 Synchronize Flag** 1 = Duty1 is synchronizing to ECLK domain 0 = Duty1 has been synchronized to ECLK domain **Note:** software should check this flag when writing duty1, if this flag is set, and user ignore this flag and change duty1, the corresponding CNR and CMR may be wrong for one duty cycle
[4]	**Duty0Syncflag**	**Duty0 Synchronize Flag** 1 = Duty0 is synchronizing to ECLK domain

Bits	Description	
		0 = Duty0 has been synchronized to ECLK domain **Note:** software should check this flag when writing duty0, if this flag is set, and user ignore this flag and change duty0, the corresponding CNR and CMR may be wrong for one duty cycle
[3]	**TMINT3**	**PWM Timer 3 Interrupt Flag** Flag is set by hardware when PWM3 down counter reaches zero, software can clear this bit by writing a one to it.
[2]	**TMINT2**	**PWM Timer 2 Interrupt Flag** Flag is set by hardware when PWM2 down counter reaches zero, software can clear this bit by writing a one to it.
[1]	**TMINT1**	**PWM Timer 1 Interrupt Flag** Flag is set by hardware when PWM1 down counter reaches zero, software can clear this bit by writing a one to it.
[0]	**TMINT0**	**PWM Timer 0 Interrupt Flag** Flag is set by hardware when PWM0 down counter reaches zero, software can clear this bit by writing a one to it.

Note: User can clear each interrupt flag by writing a one to corresponding bit in PWM_IS.

PWM Output Enable Register (PWMx_OE) for PWM0

Register	Offset	R/W	Description	Reset Value
PWM_OE **x=0,1**	PWMx_BA+0x014	R/W	PWM Output Enable for PWM0~PWM3	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**CH3_OE**	**CH2_OE**	**CH1_OE**	**CH0_OE**

Bits	Description	
[3]	**CH3_OE**	**PWM CH3 Output Enable Register** 1 = PWM CH3 output to pin Enabled. 0 = PWM CH3 output to pin Disabled. **Note:** The corresponding GPI/O pin also must be switched to PWM function (refer to GPx_MFP)
[2]	**CH2_OE**	**PWM CH2 Output Enable Register** 1 = PWM CH2 output to pin Enabled. 0 = PWM CH2 output to pin Disabled. **Note:** The corresponding GPI/O pin also must be switched to PWM function (refer to GPx_MFP)
[1]	**CH1_OE**	**PWM CH1 Output Enable Register** 1 = PWM CH1 output to pin Enabled. 0 = PWM CH1 output to pin Disabled. **Note:** The corresponding GPI/O pin also must be switched to PWM function (refer to GPx_MFP)
[0]	**CH0_OE**	**PWM CH0 Output Enable Register** 1 = PWM CH0 output to pin Enabled. 0 = PWM CH0 output to pin Disabled. **Note:** The corresponding GPI/O pin also must be switched to PWM function (refer to GPx_MFP)

PWM DUTY Register 3-0 (PWMx_DUTY3~0)

Register	Offset	R/W	Description	Reset Value
PWM_DUTY0 **x=0,1**	PWMx_BA+0x01C	R/W	PWM Counter/Comparator Register 0	0x0000_0000
PWM_DUTY1 **x=0,1**	PWMx_BA+0x028	R/W	PWM Counter/Comparator Register 1	0x0000_0000
PWM_DUTY2 **x=0,1**	PWMx_BA+0x034	R/W	PWM Counter/Comparator Register 2	0x0000_0000
PWM_DUTY3 **x=0,1**	PWMx_BA+0x040	R/W	PWM Counter/Comparator Register 3	0x0000_0000

31	30	29	28	27	26	25	24
CM							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CM							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CN							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CN							

Bits	Description	
[31:16]	**CM**	**PWM Comparator Register** CM determines the PWM duty. PWM frequency = PWMxy_CLK/(prescale+1)*(clock divider)/(CN+1); where xy, could be 01, 23, depending on the selected PWM channel. Duty ratio = (CM+1)/(CN+1). CM >= CN: PWM output is always high. CM < CN: PWM low width = (CN-CM) unit; PWM high width = (CM+1) unit. CM = 0: PWM low width = (CN) unit; PWM high width = 1 unit (Unit = one PWM clock cycle) **Note:** Any write to CM will take effect in next PWM cycle.
[15:0]	**CN**	**PWM Counter/Timer Loaded Value** CN determines the PWM period. PWM frequency = PWMxy_CLK/(prescale+1)*(clock divider)/(CN+1); where xy, could be 01, 23, depends on selected PWM channel. Duty ratio = (CM+1)/(CN+1). CM >= CN: PWM output is always high.

Bits	Description	
		CM < CN: PWM low width = (CN-CM) unit; PWM high width = (CM+1) unit. CM = 0: PWM low width = (CN) unit; PWM high width = 1 unit (Unit = one PWM clock cycle) **Note:** Any write to CN will take effect in next PWM cycle.

PWM Data Register (PWMx_DATA)

Register	Offset	R/W	Description	Reset Value
PWM_DATA0 **x=0,1**	PWMx_BA+0x020	R	PWM Data Register 0	0x0000_0000
PWM_DATA1 **x=0,1**	PWMx_BA+0x02C	R	PWM Data Register 1	0x0000_0000
PWM_DATA2 **x=0,1**	PWMx_BA+0x038	R	PWM Data Register 2	0x0000_0000
PWM_DATA3 **x=0,1**	PWMx_BA+0x044	R	PWM Data Register 3	0x0000_0000

31	30	29	28	27	26	25	24
sync	DATA (only for the corresponding cascade enable is set for channel 0, 2)						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DATA (only for the corresponding cascade enable is set for channel 0, 2)							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA							

Bits	Description	
[31]	**sync**	**Indicate that CNR value is sync to PWM counter** 1 = CNR value is not sync to PWM counter 0 = CNR value is sync to PWM counter **Note:** when the corresponding cascade enable .bit is set is bit will not appear in the corresponding channel
[30:16]	**PWMx_DATAy[30:16]**	**PWM Data Register** User can monitor PWMx_DATAy to know the current value in 32-bit down count counter **Notes:**This will be valid only for the corresponding cascade enable .bit is set
[15:0]	**PWMx_DATAy[15:0]**	**PWM Data Register** User can monitor PWMx_DATAy to know the current value in 16-bit down count counter.

Capture Control Register (PWMx_CAPCTL)

Register	Offset	R/W	Description	Reset Value
PWM_CAPCTL x=0,1	PWMx_BA+0x054	R/W	Capture Control Register	0x0000_0000

31	30	29	28	27	26	25	24
CAPRELOADFEN3	CAPRELOADREN3	CH23CASK	CH2RFORDER	Reserved	CAPCH3PADEN	CAPCH3EN	INV3
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CAPRELOADFEN2	CAPRELOADREN2	PDMACAPMOD2		CH2PDMAEN	CAPCH2PADEN	CAPCH2EN	INV2
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CAPRELOADFEN1	CAPRELOADREN1	CH01CASK	CH0RFORDER	Reserved	CAPCH1PADEN	CAPCH1EN	INV1
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CAPRELOADFEN0	CAPRELOADREN0	PDMACAPMOD0		CH0PDMAEN	CAPCH0PADEN	CAPCH0EN	INV0

Bits	Description	
[31]	**CAPRELOADFEN3**	**Reload CNR3 when CH3 falling capture Event Comes** 1 = Falling capture reload for CH3 Enabled. 0 = Falling capture reload for CH3 Disabled.
[30]	**CAPRELOADREN3**	**Reload CNR3 when CH3 Rising Capture Event Comes** 1 = Rising capture reload for CH3 Enabled. 0 = Rising capture reload for CH3 Disabled.
[29]	**CH23CASK**	**Cascade channel 2 and channel 3 PWM counter for capturing usage**
[28]	**CH2RFORDER**	**Set this bit to determine whether the PWM_CRL2 or PWM_CFL2 is the first captured data transferred to memory through PDMA when PDMACAPMOD2 = 2'b11** 1 = PWM_CRL2 is the first captured data to memory 0 = PWM_CFL2 is the first captured data to memory
[27]	**Reserved**	**Reserved**
[26]	**CAPCH3PADEN**	**Capture Input Enable Register** 0 = OFF 1 = ON
[25]	**CAPCH3EN**	**Capture Channel 3 transition Enable/Disable** 1 = Capture function on channel 3 Enabled. 0 = Capture function on channel 3 Disabled When Enabled, Capture latched the PMW-timer and saved to PWM_CRL3 (Rising latch) and PWM_CFL3 (Falling latch). When Disabled, Capture does not update PWM_CRL3 and PWM_CFL3, and disable Channel 3 Interrupt.
[24]	**INV3**	**Channel 3 Inverter ON/OFF**

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>1 = Inverter ON. Reverse the input signal from GPIO before fed to Capture timer
0 = Inverter OFF</td></tr>
<tr><td>[23]</td><td>CAPRELOADFEN2</td><td>Reload CNR2 when CH2 capture failing event coming
1 = Failing capture reload for CH2 Enabled
0 = Failing capture reload for CH2 Disabled</td></tr>
<tr><td>[22]</td><td>CAPRELOADREN2</td><td>Reload CNR2 when CH2 capture rising event coming
1 = Rising capture reload for CH2 Enabled
0 = Rising capture reload for CH2 Disabled</td></tr>
<tr><td>[21:20]</td><td>PDMACAPMOD2</td><td>Select CRL2 or CFL2 for PDMA Transfer<table><tr><td>00</td><td>reserved</td></tr><tr><td>01</td><td>CRL2</td></tr><tr><td>10</td><td>CFL2</td></tr><tr><td>11</td><td>both CRL2 and CFL2</td></tr></table></td></tr>
<tr><td>[19]</td><td>CH2PDMAEN</td><td>Channel 2 PDMA Enable
1 = Channel 2 PDMA function Enabled for the channel 2 captured data and transfer to memory
0 = Channel 2 PDMA function Disabled.</td></tr>
<tr><td>[18]</td><td>CAPCH2PADEN</td><td>Capture Input Enable Register
0 = OFF
1 = ON</td></tr>
<tr><td>[17]</td><td>CAPCH2EN</td><td>Capture Channel 2 transition Enable/Disable
1 = Capture function on channel 2 Enabled.
0 = Capture function on channel 2 Disabled
When Enabled, Capture latched the PWM-timer value and saved to PWM_CRL2 (Rising latch) and PWM_CFL2 (Falling latch).
When Disabled, Capture does not update PWM_CRL2 and PWM_CFL2, and disable Channel 2 Interrupt.</td></tr>
<tr><td>[16]</td><td>INV2</td><td>Channel 2 Inverter ON/OFF
1 = Inverter ON. Reverse the input signal from GPIO before fed to Capture timer
0 = Inverter OFF</td></tr>
<tr><td>[15]</td><td>CAPRELOADFEN1</td><td>Reload CNR1 when CH1 capture falling event coming
1 = Capture falling reload for CH1 Enabled.
0 = Capture falling reload for CH1 Disabled.</td></tr>
<tr><td>[14]</td><td>CAPRELOADREN1</td><td>Reload CNR1 when CH1 Capture Rising Event Comes
1 = Rising capture reload for CH1 Enabled
0 = Rising capture reload for CH1 Disabled.</td></tr>
</table>

Bits	Description	
[13]	**CH01CASK**	**Cascade channel 0 and channel 1 PWM timer for capturing usage**
[12]	**CH0RFORDER**	**Set this bit to determine whether the PWM_CRL0 or PWM_CFL0 is the first captured data transferred to memory through PDMA when PDMACAPMOD0 =2'b11** 1 = PWM_CRL0 is the first captured data to memory 0 = PWM_CFL0 is the first captured data to memory
[11]	**Reserved**	**Reserved**
[10]	**CAPCH1PADEN**	**Capture Input Enable Register** 0 = OFF 1 = ON
[9]	**CAPCH1EN**	**Capture Channel 1 transition Enable/Disable** 1 = Capture function on channel 1 Enabled. 0 = Capture function on channel 1 Disabled. When Enabled, Capture latched the PMW-counter and saved to PWM_CRL1 (Rising latch) and PWM_CFL1 (Falling latch). When Disabled, Capture does not update PWM_CRL1 and PWM_CFL1, and disable Channel 1 Interrupt.
[8]	**INV1**	**Channel 1 Inverter ON/OFF** 1 = Inverter ON. Reverse the input signal from GPIO before fed to Capture timer 0 = Inverter OFF
[7]	**CAPRELOADFEN0**	**Reload CNR0 when CH0 Capture Falling Event Comes** 1 = Falling capture reload for CH0 Enabled 0 = Falling capture reload for CH0 Disabled
[6]	**CAPRELOADREN0**	**Reload CNR0 when CH0 Capture Rising Event Comes** 1 = Rising capture reload for CH0 Enabled 0 = Rising capture reload for CH0 Disabled
[5:4]	**PDMACAPMOD0**	**Select CRL0 or CFL0 for PDMA Transfer** 00 : reserved 01 : CRL0 10 : CFL0 11 : both CRL0 and CFL0
[3]	**CH0PDMAEN**	**Channel 0 PDMA Enable** 1 = Channel 0 PDMA function Enabled for the channel 0 captured data and transfer to memory. 0 = Channel 0 PDMA function Disabled.
[2]	**CAPCH0PADEN**	**Capture Input Enable Register** 0 = OFF 1 = ON
[1]	**CAPCH0EN**	**Capture Channel 0 transition Enable/Disable** 1 = Capture function on channel 0 Enabled.

Bits	Description	
		0 = Capture function on channel 0 Disabled. When Enabled, Capture latched the PWM-timer value and saved to PWM_CRL0 (Rising latch) and PWM_CFL0 (Falling latch). When Disabled, Capture does not update PWM_CRL0 and PWM_CFL0, and disable Channel 0 Interrupt.
[0]	**INV0**	**Channel 0 Inverter ON/OFF** 1 = Inverter ON. Reverse the input signal from GPIO before fed to Capture timer 0 = Inverter OFF

Capture Interrupt Enable Register (PWMx_CAPINTEN)

Register	Offset	R/W	Description	Reset Value
PWM_CAPINTEN x=0,1	PWMx_BA+0x058	R/W	Capture interrupt enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						**CFL_IE3**	**CRL_IE3**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved						**CFL_IE2**	**CRL_IE2**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						**CFL_IE1**	**CRL_IE1**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**CFL_IE0**	**CRL_IE0**

Bits	Description	
[31:26]	**Reserved**	**Reserved**
[25]	**CFL_IE3**	**Channel 3 Falling Latch Interrupt Enable ON/OFF** 1 = Falling latch interrupt Enabled. 0 = Falling latch interrupt Disabled. When Enabled, if Capture detects Channel 3 has falling transition, Capture issues an Interrupt.
[24]	**CRL_IE3**	**Channel 3 Rising Latch Interrupt Enable ON/OFF** 1 = Rising latch interrupt Enabled. 0 = Rising latch interrupt Disabled. When Enabled, if Capture detects Channel 3 has rising transition, Capture issues an Interrupt.
[23:18]	**Reserved**	**Reserved**
[17]	**CFL_IE2**	**Channel 2 Falling Latch Interrupt Enable ON/OFF** 1 = Falling latch interrupt Enabled. 0 = Falling latch interrupt Disabled. When Enabled, if Capture detects Channel 2 has falling transition, Capture issues an Interrupt.
[16]	**CRL_IE2**	**Channel 2 Rising Latch Interrupt Enable ON/OFF** 1 = Rising latch interrupt Enabled. 0 = Rising latch interrupt Disabled. When Enabled, if Capture detects Channel 2 has rising transition, Capture issues an Interrupt.
[15:10]	**Reserved**	**Reserved**

Bits	Description	
[9]	**CFL_IE1**	**Channel 1 Falling Latch Interrupt Enable** 1 = Falling latch interrupt Enabled. 0 = Falling latch interrupt Disabled. When Enabled, if Capture detects Channel 1 has falling transition, Capture issues an Interrupt.
[8]	**CRL_IE1**	**Channel 1 Rising Latch Interrupt Enable** 1 = Rising latch introll Enabled. 0 = Rising latch interrupt Disabled. When Enabled, if Capture detects Channel 1 has rising transition, Capture issues an Interrupt.
[7:2]	**Reserved**	reserved
[1]	**CFL_IE0**	**Channel 0 Falling Latch Interrupt Enable ON/OFF** 1 = Falling latch interrupt Enabled. 0 = Falling latch interrupt Disabled. When Enabled, if Capture detects Channel 0 has falling transition, Capture issues an Interrupt.
[0]	**CRL_IE0**	**Channel 0 Rising Latch Interrupt Enable ON/OFF** 1 = Rising latch interrupt Enabled. 0 = Rising latch interrupt Disabled. When Enabled, if Capture detects Channel 0 has rising transition, Capture issues an Interrupt.

Capture Interrupt Status Register (PWMx_CAPINTSTS)

Register	Offset	R/W	Description	Reset Value
PWM_CAPINTSTS **x=0,1**	PWMx_BA+0x05C	R/W	Capture Interrupt Indication Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved			**CAPOVF3**	**CAPOVR3**	**CFLI3**	**CRLI3**	**CAPIF3**
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			**CAPOVF2**	**CAPOVR2**	**CFLI2**	**CRLI2**	**CAPIF2**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**CAPOVF1**	**CAPOVR1**	**CFLI1**	**CRLI1**	**CAPIF1**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**CAPOVF0**	**CAPOVR0**	**CFLI0**	**CRLI0**	**CAPIF0**

Bits	Description	
[31:29]	**Reserved**	**Reserved**
[28]	**CAPOVF3**	**Capture Falling Flag Over Run for Channel 3** This flag indicate CFL3 update faster than software reading it when it is set This bit will be cleared automatically when user clear CFLI3 bit 26 of PWM_CAPINTSTS
[27]	**CAPOVR3**	**Capture Rising Flag Over Run for Channel 3** This flag indicate CRL3update faster than software reading it when it is set This bit will be cleared automatically when user clear CRLI3 bit 25 of PWM_CAPINTSTS
[26]	**CFLI3**	**PWM_CFL3 Latched Indicator Bit** When input channel 3 has a falling transition, PWM_CFL3 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1 to it.
[25]	**CRLI3**	**PWM_CRL3 Latched Indicator Bit** When input channel 3 has a rising transition, PWM_CRL3 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1 to it.
[24]	**CAPIF3**	**Capture3 Interrupt Indication Flag** If channel 3 rising latch interrupt is enabled (CRL_IE3 =1), a rising transition occurs at input channel 3 will result in CAPIF3 to high; Similarly, a falling transition will cause CAPIF3 to be set high if channel 3 falling latch interrupt is enabled (CFL_IE3=1). This flag is cleared by software with a write 1 on it.
[23:21]	**Reserved**	**Reserved**
[20]	**CAPOVF2**	**Capture Falling Flag Over Run for Channel 2** This flag indicate CFL2 update faster than software reading it when it is set This bit will be cleared automatically when user clear CFLI2 bit 18 of PWM_CAPINTSTS
[19]	**CAPOVR2**	**Capture Rising Flag Over Run for Channel 2**

Bits	Description	
		This flag indicate CRL2 update faster than software reading it when it is set This bit will be cleared automatically when user clear CRLI2 bit 17 of PWM_CAPINTSTS
[18]	**CFLI2**	**PWM_CFL2 Latched Indicator Bit** When input channel 2 has a falling transition, PWM0_CFL2 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1 to it.
[17]	**CRLI2**	**PWM_CRL2 Latched Indicator Bit** When input channel 2 has a rising transition, PWM0_CRL2 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1 to it.
[16]	**CAPIF2**	**Capture2 Interrupt Indication Flag** If channel 2 rising latch interrupt is enabled (CRL_IE2=1), a rising transition occurs at input channel 2 will result in CAPIF2 to high; Similarly, a falling transition will cause CAPIF2 to be set high if channel 2 falling latch interrupt is enabled (CFL_IE2=1). This flag is cleared by software with a write 1 on it.
[15:13]	**Reserved**	**Reserved**
[12]	**CAPOVF1**	**Capture Falling Flag Over Run for Channel 1** This flag indicate CFL1 update faster than software reading it when it is set This bit will be cleared automatically when user clear CFLI1 bit 10 of PWM_CAPINTSTS
[11]	**CAPOVR1**	**Capture Rising Flag Over Run for Channel 1** This flag indicate CRL1 update faster than software reading it when it is set This bit will be cleared automatically when user clear CRLI1 bit 9 of PWM_CAPINTSTS
[10]	**CFLI1**	**PWM_CFL1 Latched Indicator Bit** When input channel 1 has a falling transition, PWM_CFL1 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1 to it.
[9]	**CRLI1**	**PWM_CRL1 Latched Indicator Bit** When input channel 1 has a rising transition, PWM_CRL1 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1 to it.
[8]	**CAPIF1**	**Capture1 Interrupt Indication Flag** If channel 1 rising latch interrupt is enabled (CRL_IE1 =1), a rising transition occurs at input channel 1 will result in CAPIF1 to high; Similarly, a falling transition will cause CAPIF1 to be set high if channel 1 falling latch interrupt is enabled (CFL_IE1 =1). This flag is cleared by software with a write 1 on it.
[7:5]	**Reserved**	**Reserved**
[4]	**CAPOVF0**	**Capture Falling Flag Over Run for Channel 0** This flag indicate CFL0 update faster than software read it when it is set This bit will be cleared automatically when user clear CFLI0 bit 2 of PWM_CAPINTSTS
[3]	**CAPOVR0**	**Capture Rising Flag Over Run for Channel 0** This flag indicate CRL0 update faster than software reading it when it is set This bit will be cleared automatically when user clears CRLI0 bit 1 of PWM_CAPINTSTS.
[2]	**CFLRI0**	**PWM_CFL0 Latched Indicator Bit** When input channel 0 has a falling transition, PWM0_CFL0 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1

Bits	Description	
		to it.
[1]	**CRLI0**	**PWM_CRL0 Latched Indicator Bit** When input channel 0 has a rising transition, PWM0_CRL0 was latched with the value of PWM down-counter and this bit is set by hardware, software can clear this bit by writing 1 to it.
[0]	**CAPIF0**	**Capture0 Interrupt Indication Flag** If channel 0 rising latch interrupt is enabled (CRL_IE0 =1), a rising transition occurs at input channel 0 will result in CAPIF0 to high; Similarly, a falling transition will cause CAPIF0 to be set high if channel 0 falling latch interrupt is enabled (CFL_IE0 =1). This flag is cleared by software with a write 1 on it.

Capture Rising Latch Register3-0 (PWMx_CRL3-0)

Register	Offset	R/W	Description	Reset Value
PWM_CRL0 **x=0,1**	PWMx_BA+0x060	R	Capture Rising Latch Register (Channel 0)	0x0000_0000
PWM_CRL1 **x=0,1**	PWMx_BA+0x068	R	Capture Rising Latch Register (Channel 1)	0x0000_0000
PWM_CRL2 **x=0,1**	PWMx_BA+0x070	R	Capture Rising Latch Register (Channel 2)	0x0000_0000
PWM_CRL3 **x=0,1**	PWMx_BA+0x078	R	Capture Rising Latch Register (Channel 3)	0x0000_0000

31	30	29	28	27	26	25	24
CRL (only for channel 0 and channel 2 when corresponding cascade enable bit is set)							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CRL (only for channel 0 and channel 2 when corresponding cascade enable bit is set)							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CRL							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CRL							

Bits	Description	
[31:16]	**CRL[31:16]**	**Upper Half Word of 32-bit Capture Data when Cascade Enabled** When cascade is enabled for capture channel 0, 2,the original 16 bit counter extend to 32 bit, and capture result CRL0 and CRL2 are also extend to 32 bit,
[15:0]	**CRL[15:0]**	**Capture Rising Latch Register** Latch the PWM counter when Channel 0/1/2/3 has rising transition.

PWM Capture Falling Latch Register3-0 (PWMx_CFL3-0)

Register	Offset	R/W	Description	Reset Value
PWM_CFL0 x=0,1	PWMx_BA+0x064	R	Capture Falling Latch Register (Channel 0)	0x0000_0000
PWM_CFL1 x=0,1	PWMx_BA+0x06C	R	Capture Falling Latch Register (Channel 1)	0x0000_0000
PWM_CFL2 x=0,1	PWMx_BA+0x074	R	Capture Falling Latch Register (Channel 2)	0x0000_0000
PWM_CFL3 x=0,1	PWMx_BA+0x07C	R	Capture Falling Latch Register (Channel 3)	0x0000_0000

31	30	29	28	27	26	25	24
CFL (only for channel 0 and channel 2 when corresponding cascade enable bit is set)							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CFL (only for channel 0 and channel 2 when corresponding cascade enable bit is set)							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CFL							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CFL							

Bits	Description	
[31:16]	**CFL[31:16]**	**Upper Half Word of 32-bit Capture Data When Cascade Enabled** When cascade is enabled for capture channel 0, 2, the original 16 bit counter extend to 32 bit, and capture result CFL0 and CFL2 are also extend to 32 bit,
[15:0]	**CFL[15:0]**	**Capture Falling Latch Register** Latch the PWM counter when Channel 01/2/3 has Falling transition.

PWMx_PDMACH0 (PDMA Data Register for Capture Channel 0)

Register	Offset	R/W	Description	Reset Value
PWM_PDMACH0 **x=0,1**	PWMx_BA+0x080	R	PDMA channel 0 captured data	0x0000_0000

31	30	29	28	27	26	25	24
PDMACH0							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PDMACH0							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMACH0							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMACH0							

Bits	Description	
[31:24]	**Captured data[31:24]**	**PDMACH0** When CH01CASK is disabled, this byte is 0 When CH01CASK is enabled, It's the 4th byte of 32 bit capturing data for channel 0
[23:16]	**Captured data[23:16]**	**PDMACH0** When CH01CASK is disabled, this byte is 0 When CH01CASK is enabled, It is the third byte of 32 bit capturing data for channel 0
[15:8]	**Captured data[15:8]**	**PDMACH0** When CH01CASK is disabled, it is the capturing value(CFL0/CRL0) for channel 0 When CH01CASK is enabled, It is the second byte of 32 bit capturing data for channel 0
[7:0]	**Captured data[7:0]**	**PDMACH0** When CH01CASK is disabled, it is the capturing value(CFL0/CRL0) for channel 0 When CH01CASK is enabled, It is the for the first byte of 32 bit capturing data for channel 0

PWMx_PDMACH2 (PDMA Data Register for Capture Channel 2)

Register	Offset	R/W	Description	Reset Value
PWM_PDMACH2 **x=0,1**	PWMx_BA+0x084	R	PDMA channel 2 captured data	0x0000_0000

31	30	29	28	27	26	25	24
PDMACH2							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
PDMACH2							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
PDMACH2							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
PDMACH2							

Bits	Description	
[31:24]	**Captured data[31:24]**	**PDMACH0** When CH23CASK is disabled, this byte is 0 When CH23CASK is enabled, It's the 4th byte of 32 bit capturing data for channel 2
[23:16]	**Captured data[23:16]**	**PDMACH0** When CH23CASK is disabled, this byte is 0 When CH23CASK is enabled, It is the third byte of 32 bit capturing data for channel 2
[15:8]	**Captured data[15:8]**	**PDMACH0** When CH23CASK is disabled, it is the capturing value(CFL2/CRL2) for channel 2 When CH23CASK is enabled, It is the second byte of 32 bit capturing data for channel 2
[7:0]	**Captured data[7:0]**	**PDMACH0** When CH23CASK is disabled, it is the capturing value(CFL2/CRL2) for channel 2 When CH23CASK is enabled, It is the for the first byte of 32 bit capturing data for channel 2

5.11 Watchdog Timer Controller

5.11.1 Overview

The purpose of Watchdog Timer is to perform a system reset after the software running into a problem. This prevents system from hanging for an infinite period of time. Besides, this Watchdog Timer supports the function to wake-up CPU from power-down mode. The watchdog timer includes an 18-bit free running counter with programmable time-out intervals.

5.11.2 Features

- 18-bit free running WDT counter for Watchdog timer time-out interval.
- Selectable time-out interval (2^4 ~ 2^18) and the time-out interval is 104 ms ~ 26.316 s (if WDT_CLK = 10 kHz).
- Reset period = (1 / 10 kHz) * 63, if WDT_CLK = 10 kHz.

5.11.3 Block Diagram

The Watchdog Timer clock control and block diagram are shown as follows.

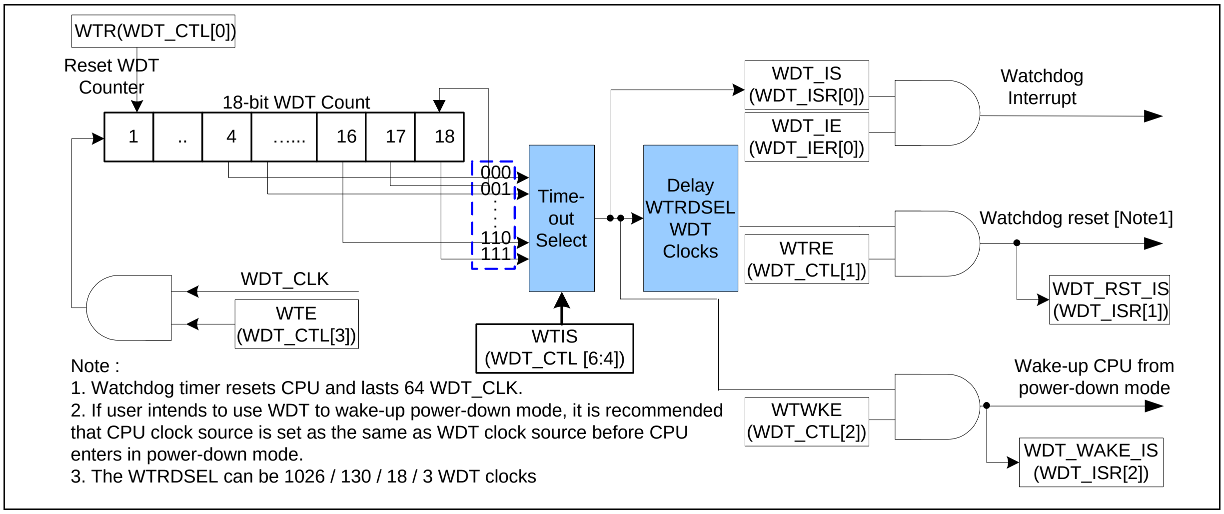


Figure 5.11-1 Watchdog Controller Block Diagram

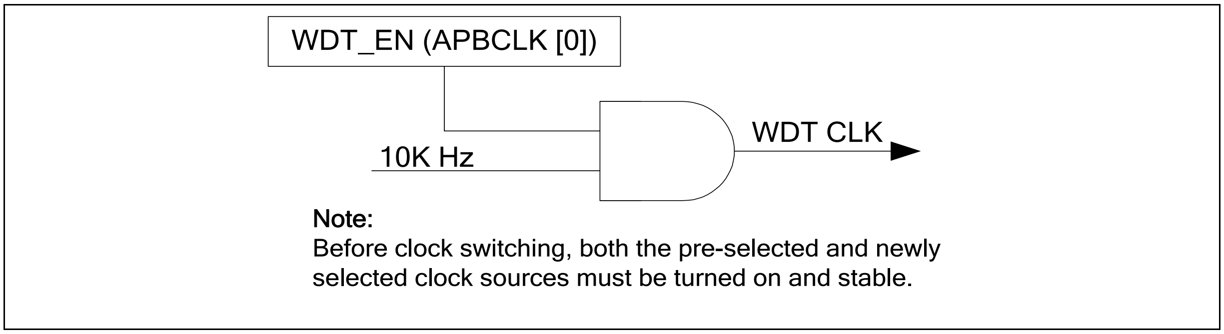


Figure 5.11-2 Watchdog Timer Clock Control Diagram

5.11.4 Functional Description

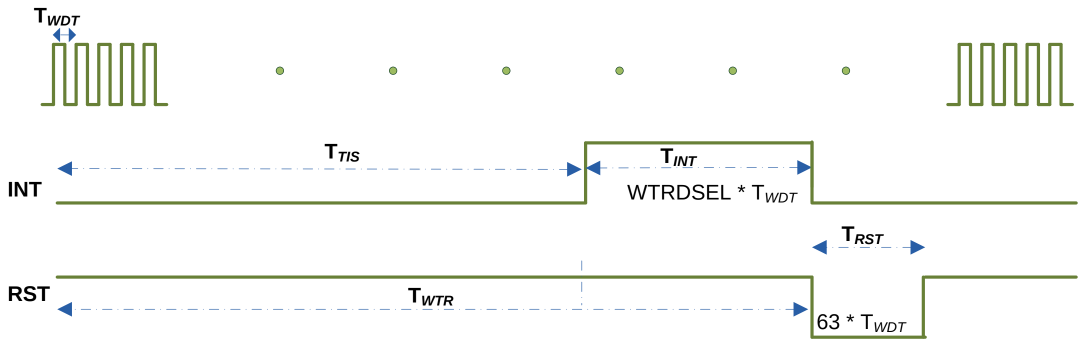
The purpose of Watchdog Timer is to perform a system reset after the software running into a problem. This prevents system from hanging for an infinite period of time. Besides, this Watchdog Timer supports the function to wake-up the chip from Power-down mode. Moreover, the Watchdog counter will be automatically reset when the chip is entering Power-down mode. The watchdog timer includes an 18-bit free running counter with programmable time-out intervals. The following figure show the watchdog time-out interval selection and the next figure show the timing of watchdog interrupt signal and reset signal.

Setting WTE (WDT_CTL [3]) enables the watchdog timer and the WDT counter starts counting up. When the counter reaches the selected time-out interval, Watchdog timer interrupt flag WDT_IS will be set immediately to request a WDT interrupt if the watchdog timer interrupt enable bit WDT_IE is set, in the meanwhile, a specified delay time (WTRDSEL * T^{WDT}) follows the time-out event. User must set WTR (WDT_CTL [0]) (Watchdog timer reset) high to reset the 18-bit WDT counter to avoid CPU from Watchdog timer reset before the delay time expires. WTR bit is auto cleared by hardware after WDT counter is reset. There are eight time-out intervals with specific delay time which are selected by Watchdog timer interval select bits WTIS (WDTCR [6:4]). If the WDT counter has not been cleared after the specific delay time expires, the watchdog timer will set Watchdog Timer Reset Flag

(WDT_RST_IS) high and reset CPU. This reset will last 63 * WDT clocks (T^{RST}) then CPU restarts executing program from reset vector (0x0000 0000). WDT_RST_IS will not be cleared by Watchdog reset. User may poll WDT_RST_IS by software to recognize the reset source.

WTIS	WTR Time-out Interval	Interrupt Period T_{INT}	Time-out Interval WDT_CLK = 10 KHz T_{TIS}	Reset Internal WDT_CLK = 10 KHz T_{WTR}
000	$2^4 * T_{WDT}$	$1024 * T_{WDT}$	1.6 ms	104 ms
001	$2^6 * T_{WDT}$	$1024 * T_{WDT}$	6.4 ms	108.8 ms
010	$2^8 * T_{WDT}$	$1024 * T_{WDT}$	25.6 ms	128 ms
011	$2^{10} * T_{WDT}$	$1024 * T_{WDT}$	102.4 ms	204.8 ms
100	$2^{12} * T_{WDT}$	$1024 * T_{WDT}$	407 ms	512 ms
101	$2^{14} * T_{WDT}$	$1024 * T_{WDT}$	1.638 s	1.741 s
110	$2^{16} * T_{WDT}$	$1024 * T_{WDT}$	6.553 s	6.656 s
111	$2^{18} * T_{WDT}$	$1024 * T_{WDT}$	26.214 s	26.316 s

Table 5.11-1 Watchdog Time-out Interval Selection



- ***T_{WDT} : Watchdog Engine Clock Time Period***
- ***T_{TIS} : Watchdog Timeout Interval Selection Period***
- ***T_{INT} : Watchdog Interrupt Period***
- ***T_{RST} : Watchdog Reset Period***
- ***T_{WTR} : Watchdog Reset Interval Period***

Figure 5.11-3 Watchdog Timing of Interrupt and Reset Signal

5.11.5 Register and Memory Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
WDT Base Address: **WDT_BA = 0x4000_4000**				
WDT_CTL	WDT_BA+0x00	R/W	Watchdog Timer Control Register	0x0000_0070
WDT_IER	WDT_BA+0x04	R/W	Watchdog Timer Interrupt Enable Register	0x0000_0000
WDT_ISR	WDT_BA+0x08	R/W	Watchdog Timer Interrupt Status Register	0x0000_0000

5.11.6 Register Description

Watchdog Timer Control Register (WDT_CTL)

Register	Offset	R/W	Description	Reset Value
WDT_CTL	WDT_BA+0x00	R/W	Watchdog Timer Control Register	0x0000_0070

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						WTRDSEL	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	WTIS			WTE	WTWKE	WTRE	WTR

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:10]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[9:8]</td><td>WTRDSEL</td><td>Watchdog Timer Reset Delay Select
When watchdog timeout happened, software has a time named watchdog reset delay period to clear watchdog timer to prevent watchdog reset happened. Software can select a suitable value of watchdog reset delay period for different watchdog timeout period.
<table><tr><th>WTRDSEL</th><th>Description</th></tr><tr><td>00</td><td>Watchdog reset delay period is 1026 watchdog clock</td></tr><tr><td>01</td><td>Watchdog reset delay period is 130 watchdog clock</td></tr><tr><td>10</td><td>Watchdog reset delay period is 18 watchdog clock</td></tr><tr><td>11</td><td>Watchdog reset delay period is 3 watchdog clock</td></tr></table>
This register will be reset if watchdog reset happened</td></tr>
<tr><td>[7]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[6:4]</td><td>WTIS</td><td>Watchdog Timer Interval Selection
This is a protected register. Please refer to open lock sequence to program it.
These three bits select the time-out interval for the Watchdog timer. This count is free running counter.
Please refer to the Table 5.11-1.</td></tr>
<tr><td>[3]</td><td>WTE</td><td>Watchdog Timer Enable
This is a protected register. Please refer to open lock sequence to program it.
1 = Watchdog timer Enabled.
0 = Watchdog timer Disabled (this action will reset the internal counter).</td></tr>
</table>

[2]	**WTWKE**	**Watchdog Timer Wake-Up Function Enable** This is a protected register. Please refer to open lock sequence to program it. 1 = Wake-up function Enabled so that Watchdog timer time-out can wake up CPU from power-down mode. 0 = Watchdog timer Wake-up CPU function Disabled.
[1]	**WTRE**	**Watchdog Timer Reset Function Enable** This is a protected register. Please refer to open lock sequence to program it. Setting this bit will enable the Watchdog timer reset function. 1 = Watchdog timer reset function Enabled. 0 = Watchdog timer reset function Disabled.
[0]	**WTR**	**Clear Watchdog Timer** This is a protected register. Please refer to open lock sequence to program it. Set this bit will clear the Watchdog timer. 1 = Reset the contents of the Watchdog timer. 0 = No effect. **Note:** This bit will be auto cleared after few clock cycles.

Watchdog Timer Interrupt Enable Register (WDT_IER)

Register	Offset	R/W	Description	Reset Value
WDT_IER	WDT_BA+0x04	R/W	Watchdog Timer Interrupt Enable Register	0x0000_0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="7">Reserved</td><td>WDT_IE</td></tr>
</table>

Bits	Description	
[31:1]	**Reserved**	**Reserved.**
[0]	**WDT_IE**	**Watchdog Timer Interrupt Enable** 1 = Watchdog timer interrupt Enabled 0 = Watchdog timer interrupt Disabled

Watchdog Timer Interrupt Status Register (WDT_ISR)

Register	Offset	R/W	Description	Reset Value
WDT_ISR	WDT_BA+0x08	R/W	Watchdog Timer Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**WDT_WAKE_IS**	**WDT_RST_IS**	**WDT_IS**

Bits	Description	
[31:3]	**Reserved**	**Reserved.**
[2]	**WDT_WAKE_IS**	**Watchdog Timer Wake-up Status** If Watchdog timer causes system to wake up from power-down mode, this bit will be set to high. It must be cleared by software with a write “1” to this bit. 1 = Wake system up from power-down mode by Watchdog time-out. 0 = Watchdog timer does not cause system wake-up. **Note1:** When system in power-down mode and watchdog time-out, hardware will set WDT_WAKE_IS and WDT_IS. **Note2:** After one engine clock, this bit can be cleared by writing “1” to it
[1]	**WDT_RST_IS**	**Watchdog Timer Reset Status** When the Watchdog timer initiates a reset, the hardware will set this bit. This flag can be read by software to determine the source of reset. Software is responsible to clear it manually by writing “1” to it. If WTRE is disabled, then the Watchdog timer has no effect on this bit. 1 = Watchdog timer reset occurs 0 = Watchdog timer reset did not occur **Note:** This bit is read only, but can be cleared by writing “1” to it.
[0]	**WDT_IS**	**Watchdog Timer Interrupt Status** If the Watchdog timer interrupt is enabled, then the hardware will set this bit to indicate that the Watchdog timer interrupt has occurred. If the Watchdog timer interrupt is not enabled, then this bit indicates that a time-out period has elapsed. 1 = Watchdog timer interrupt occurs 0 = Watchdog timer interrupt did not occur **Note:** This bit is read only, but can be cleared by writing “1” to it.

5.12 Window Watchdog Timer Controller

5.12.1 Overview

The purpose of Window Watchdog Timer is to perform a system reset within a specified window period to prevent software run to uncontrollable status by any unpredictable condition.

5.12.2 Features

- 6-bit down counter and 6-bit compare value to make the window period flexible
- Selectable WWDT clock pre-scale counter to make WWDT time-out interval variable

5.12.3 Block Diagram

The Window Watchdog Timer block diagram is shown as follows.

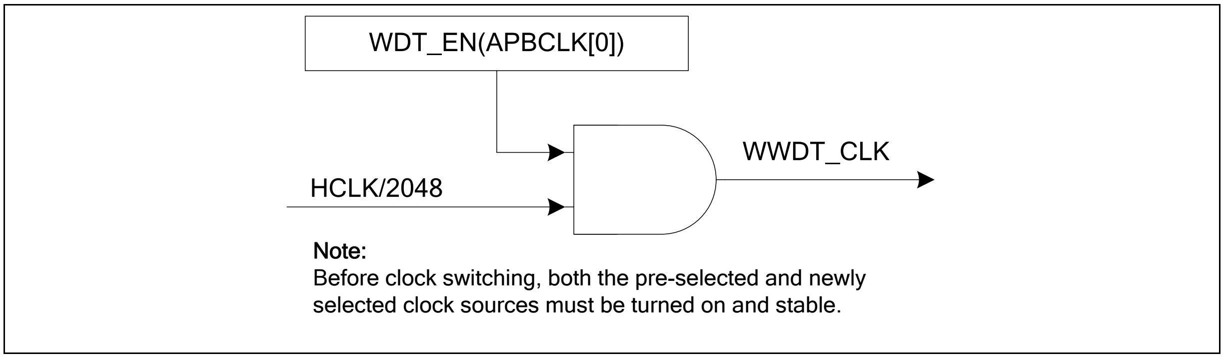


Figure 5.12-1 Window Watchdog Controller Block Diagram

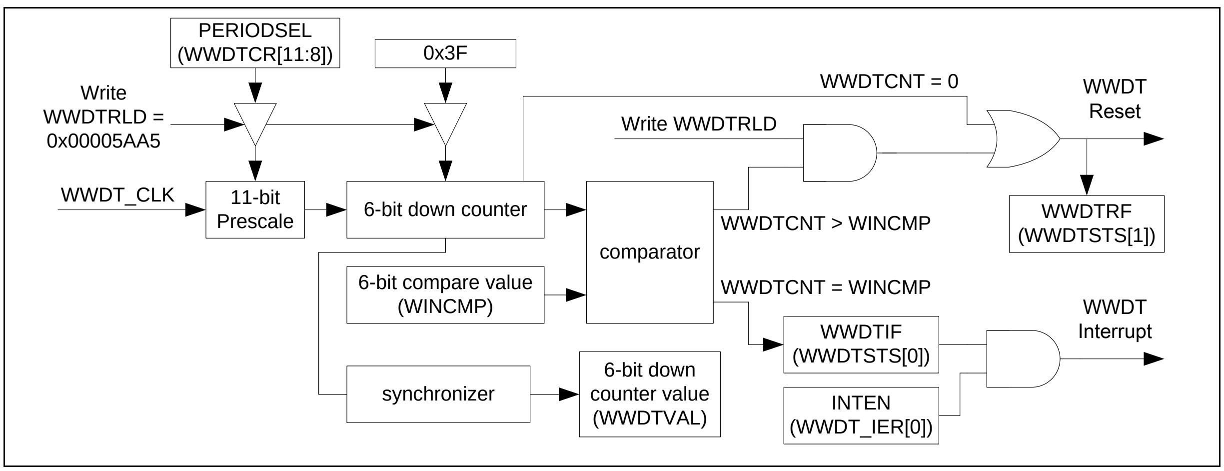


Figure 5.12-2 Watchdog Controller Block Diagram

5.12.4 Functional Description

The window watchdog timer includes a 6-bit down counter with programmable prescaler to define different time-out intervals.

The clock source of 6-bit window watchdog timer is based on system clock divide 2048 with a programmable 11-bit prescaler. The programmable 11-bit prescaler is controlled by register PERIODSEL (WWDTCR[11:8]) and the correlate of PERIODSEL and prescaler value is list in Table 5.12-1.

PERIODSEL	Prescaler Value	Timeout Period	Timeout Interval 12 M/2048 = 5.859 kHz WWDT_CLK=5.859 kHz
0000	1	1 * 64 * T_{WWDT}	10.9 ms
0001	2	2 * 64 * T_{WWDT}	21.8 ms
0010	4	4 * 64 * T_{WWDT}	43.7 ms
0011	8	8 * 64 * T_{WWDT}	87.4 ms
0100	16	16 * 64 * T_{WWDT}	174.7 ms
0101	32	32 * 64 * T_{WWDT}	349.5 ms
0110	64	64 * 64 * T_{WWDT}	699.1 ms
0111	128	128 * 64 * T_{WWDT}	1.3981 s
1000	192	192 * 64 * T_{WWDT}	2.0971 s
1001	256	256 * 64 * T_{WWDT}	2.7962 s
1010	384	384 * 64 * T_{WWDT}	4.1943 s
1011	512	512 * 64 * T_{WWDT}	5.5924 s
1100	768	768 * 64 * T_{WWDT}	8.3886 s
1101	1024	1024 * 64 * T_{WWDT}	11.1848 s
1110	1536	1536 * 64 * T_{WWDT}	16.7772 s
1111	2048	2048 * 64 * T_{WWDT}	22.3696 s

Table 5.12-1 Window Watchdog Prescaler Value Selection

The window watchdog timer can be enabled by software setting WWDTEN (WWDTCR[0]) to 1. As window watchdog timer is enabled, the down counter will start counting from 0x3F and cannot be stopped by software.

During WWDT down counting, the WWDT interrupt will happen if the counter value is equal to window watchdog timer compare value WINCMP (WWDTCR[21:16]) and INTEN(WWDT_IER[0]) is set to 1. The WWDT reset will happen if the WWDT counter value reaches to 0. Before WWDT counter down to 0, software can write certain value (0x00005AA5) to register WWDTRLD to reload 0x3F to WWDT counter to prevent WWDT reset happen and this reload action only active when WWDT counter value is equal or smaller than WINCMP. If software writes WWDTRLD during the period that WWDT counter larger than WINCMP, additional WWDT reset will happen to cause chip be reset.

When software writes certain value (0x00005AA5) to register WWDTRLD to reload WWDT counter, it need 3 window watchdog clocks to sync reload command to actually perform reload action. It means if software set window watchdog clock prescaler as divide 1, the compare value WINCMP (WWDTCR[21:16]) should larger than 2 or software will not able to reload WWDT counter before WWDT reset happened.

To prevent program run to unexpected code to disable window watchdog, the control register WWDTCR and WWDT_IER can only be write 1 time after chip power on or reset. Software can not to disable window watchdog, change pre-scale period or change window compare value as window watchdog is enabled by software unless chip is reset. And when CPU in sleep mode (WFI or WFE), the CPU will be disabled but system clock will keep, so the window watchdog timer will still counting; but when system in Power-down mode, the system clock will also be disabled, so the window watchdog timer will be stopped.

5.12.5 Register and Memory Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
WWDT Base Address: **WWDT_BA = 0x4000_4100**				
WWDTRLD	WWDT_BA+0x00	W	Window Watchdog Timer Reload Counter Register	0x0000_0000
WWDTCR	WWDT_BA+0x04	R/W	Window Watchdog Timer Control Register	0x003F_0800
WWDT_IER	WWDT_BA+0x08	R/W	Window Watchdog Timer Interrupt Enable Register	0x0000_0000
WWDTSTS	WWDT_BA+0x0C	R/W	Window Watchdog Timer Status Register	0x0000_0000
WWDTVAL	WWDT_BA+0x10	R	Window Watchdog Timer Counter Value Register	0x0000_003F

5.12.6 Register Description

Window Watchdog Timer Reload Counter Register (WWDTRLD)

Register	Offset	R/W	Description	Reset Value
WWDTRLD	WWDT_BA+0x00	W	Window Watchdog Timer Reload Counter Register	0x0000_0000

31	30	29	28	27	26	25	24
WWDTRLD							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
WWDTRLD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
WWDTRLD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WWDTRLD							

Bits	Description	
[31:0]	**WWDTRLD**	**Window Watchdog Timer Reload Counter Register** Writing 0x00005AA5 to this register will reload the Window Watchdog Timer counter value to 0x3F. **Note:** SW only can write WWDTRLD when WWDT counter value between 0 and WINCMP. If SW writes WWDTRLD when WWDT counter value larger than WINCMP, WWDT will generate RESET signal.

Window Watchdog Timer Control Register (WWDTCR)

Register	Offset	R/W	Description	Reset Value
WWDTCR	WWDT_BA+0x04	R/W	Window Watchdog Timer Control Register	0x003F_0800

Note: This register can be write only one time after chip power on or reset.

31	30	29	28	27	26	25	24
DBGEN	Reserved						
23	22	21	20	19	18	17	16
Reserved		WINCMP					
15	14	13	12	11	10	9	8
Reserved				PERIODSEL			
7	6	5	4	3	2	1	0
Reserved							WWDTEN

Bits	Description	
[31]	**DBGEN**	**WWDT Debug Enable** 1 = WWDT still counted even system is in Debug mode. 0 = WWDT stopped count if system is in Debug mode.
[30:22]	**Reserved**	**Reserved.**
[21:16]	**WINCMP**	**WWDT Window Compare Register** Set this register to adjust the valid reload window. **Note:** SW only can write WWDTRLD when WWDT counter value between 0 and WINCMP. If SW writes WWDTRLD when WWDT counter value larger than WWCMP, WWDT will generate RESET signal.
[15:12]	**Reserved**	**Reserved.**
[11:8]	**PERIODSEL**	**WWDT Pre-scale Period Select** These three bits select the pre-scale for the WWDT counter period. Please refer to Table 5.12-1.
[7:1]	**Reserved**	**Reserved.**
[0]	**WWDTEN**	**Window Watchdog Enable** Set this bit to enable Window Watchdog timer. 1 = Window Watchdog timer function Enabled 0 = Window Watchdog timer function Disabled

Window Watchdog Timer Interrupt Enable Register (WWDT_IER)

Register	Offset	R/W	Description	Reset Value
WWDT_IER	WWDT_BA+0x08	R/W	Window Watchdog Timer Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**WWDTIE**

Note: This register can be write only one time after chip power on or reset.

Bits	Description	
[31:1]	**Reserved**	**Reserved.**
[0]	**WWDTIE**	**WWDT Interrupt Enable** Setting this bit will enable the Watchdog timer interrupt function. 1 = Watchdog timer interrupt function Enabled 0 = Watchdog timer interrupt function Disabled

Window Watchdog Timer Status Register (WWDTSTS)

Register	Offset	R/W	Description	Reset Value
WWDTSTS	WWDT_BA+0x0C	R/W	Window Watchdog Timer Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**WWDTRF**	**WWDTIF**

Bits	Description	
[31:2]	**Reserved**	**Reserved.**
[1]	**WWDTRF**	**WWDT Reset Flag** When WWDT counter down count to 0 or write WWDTRLD during WWDT counter larger than WINCMP, chip will be reset and this bit is set to 1. Software can write 1 to clear this bit to 0.
[0]	**WWDTIF**	**WWDT Compare Match Interrupt Flag** When WWCMP match the WWDT counter, then this bit is set to 1. This bit will be cleared by software write 1 to this bit.

Window Watchdog Timer Counter Value Register (WWDTVAL)

Register	Offset	R/W	Description	Reset Value
WWDTVAL	WWDT_BA+0x10	R	Window Watchdog Timer Counter Value Register	0x0000_003F

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**WWDTVAL**					

Bits	Description	
[31:6]	**Reserved**	**Reserved.**
[5:0]	**WWDTVAL**	**WWDT Counter Value** This register reflects the counter value of window watchdog. This register is read only

5.13 RTC

5.13.1 Overview

Real Time Clock (RTC) unit provides user the real time and calendar message. The Clock Source of RTC is from an external 32.768 kHz crystal connected at pins X32I and X32O (reference to pin Description) or from an external 32.768 kHz oscillator output fed at pin X32I. The RTC unit provides the time message (second, minute, hour) in Time Loading Register (TLR) as well as calendar message (day, month, year) in Calendar Loading Register (CLR). The data message is expressed in BCD format. This unit offers alarm function that user can preset the alarm time in Time Alarm Register (TAR) and alarm calendar in Calendar Alarm Register (CAR).

The RTC unit supports periodic Time Tick and Alarm Match interrupts. The periodic interrupt has 8 period options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second which are selected by TTR (TTR[2:0]). When RTC counter in TLR and CLR is equal to alarm setting time registers TAR and CAR, the alarm interrupt status (RIIR.AIS) is set and the alarm interrupt is requested if the alarm interrupt is enabled (RIER.AIER=1). The RTC Time Tick (if wake-up CPU function is enabled, RTC_TTR[TWKE] high) and Alarm Match can cause CPU wake-up from idle or Power-down mode.

5.13.2 Features

- One time counter (second, minute, hour) and calendar counter (day, month, year) for user to check the time
- Alarm register (second, minute, hour, day, month, year)
- 12-hour or 24-hour mode is selectable
- Leap year compensation automatically
- Day of week counter
- Frequency compensate register (FCR)
- All time and calendar message is expressed in BCD code
- Supports periodic time tick interrupt with 8 period options 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second
- Supports RTC Time Tick and Alarm Match interrupt
- Supports wake-up CPU from Power-down mode
- Supports 80 bytes spare registers and a snoop pin to clear the content of these spare registers

5.13.3 Block Diagram

The block diagram of Real Time Clock is depicted as follows.

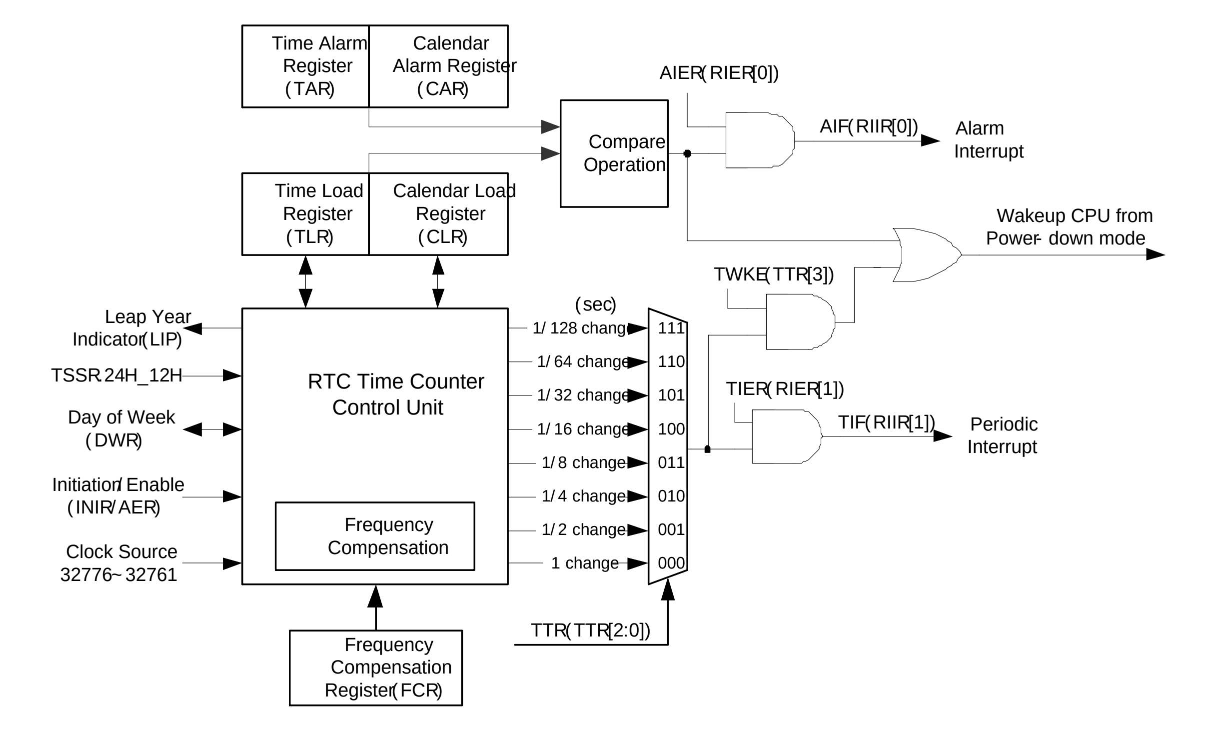


Figure 5.13-1 RTC Block Diagram

5.13.4 Functional Description

5.13.4.1 Access to RTC Register

Due to clock difference between RTC clock and system clock, when user write new data to any one of the registers, the register will not be updated until 2 RTC clocks later (60us).

In addition, user must be aware that RTC block does not check whether loaded data is out of bounds or not. RTC does not check rationality between DWR and CLR either.

5.13.4.2 RTC Initiation

When RTC block is powered on, user has to write a number (0xa5eb1357) to INIR to reset all logic. INIR acts as hardware reset circuit. Once INIR has been set as 0xa5eb1357, there is no action for RTC if any value is programmed into INIR register.

5.13.4.3 RTC Read/Write Enable

Register AER bit 15~0 is served as RTC read/write password to protect RTC registers. AER bit 15~0 has to be set as 0xa965 to enable access restriction. Once it is set, it will take effect at least 512 RTC clocks (about 15ms). Programmer can read RTC enabled status flag in AER.ENF to check whether if RTC unit start operating or not.

5.13.4.4 Frequency Compensation

The RTC FCR allows software to make digital compensation to a clock input. The frequency of clock input must be in the range from 32776 Hz to 32761 Hz. The cycle of RTC frequency compensation is 60 seconds. It will compensate the input crystal via writing RTC_FCR register and the precision is ± 0.4768 ppm (±0.0412 sec/day). Please follow the example and formula below to write the actual frequency of 32k crystal to RTC_FCR register. Following are the compensation examples for higher or lower than 32768 Hz.

Example 1:

Frequency counter measurement : 32773.65 Hz (> 32768 Hz)

Integer part: 32773

FCR.Integer = (32773 – 32768) + 7 = 12 = 0x0c

Fraction part: 0.65

FCR.Fraction = 0.65 x 64 = 41.6 => 0x29

Example 2:

Frequency counter measurement : 32763.25 Hz (< 32768 Hz)

Integer part: 32763

FCR.Integer = (32763 – 32768) + 7 = 2 = 0x02

Fraction part: 0.25

FCR.Fraction = 0.25 x 64 = 16 => 0x10

Note:

The value of RTC_FCR register will be as default value (0x0000_0700) while the compensation is not executed. User can utilize a frequency counter to measure RTC clock source via RTC time ticks interrupt event on one of GPI/O pin in manufacturing, and store the value in Flash memory for retrieval when the product is first power on. In the meanwhile, user can use RTC time ticks interrupt event to check the result of RTC frequency compensation.

5.13.4.5 Time and Calendar Counter

TLR and CLR are used to load the time and calendar. TAR and CAR are used for alarm. They are all represented by BCD.

5.13.4.6 12/24 Hour Time Scale Selection

The 12/24 hour time scale selection depends on TSSR bit 0.

5.13.4.7 Day of the Week Counter

The RTC unit provides day of week in Day of the Week Register (DWR). The value is defined from 0 to 6 to represent Sunday to Saturday respectively.

5.13.4.8 Periodic Time Tick Interrupt

The periodic interrupt has 8 period option 1/128, 1/64, 1/32, 1/16, 1/8, 1/4, 1/2 and 1 second which are selected by TTR.TTR[2:0]. When periodic time tick interrupt is enabled by setting RIER.TIER to 1, the

Periodic Time Tick Interrupt is requested periodically in the period selected by TTR register.

5.13.4.9 Alarm Interrupt

When RTC counter in TLR and CLR is equal to alarm setting time TAR and CAR the alarm interrupt status (RIIR.AIS) is set and the alarm interrupt is requested if the alarm interrupt is enabled (RIER.AIER=1).

Application Note:

TAR, CAR, TLR and CLR registers are all BCD counter.

Programmer has to make sure that the loaded values are reasonable. For example, Load CLR as 201a (year), 13 (month), 00 (day), or CLR does not match with DWR, etc.

Reset state :

Register	Reset State
AER	0
CLR	05/1/1 (year/month/day)
TLR	00:00:00 (hour : minute : second)
CAR	00/00/00 (year/month/day)
TAR	00:00:00 (hour : minute : second)
TSSR	1 (24 hr mode)
DWR	6 (Saturday)
RIER	0
RIIR	0
LIR	0
TTR	0

In TLR and TAR, only 2 BCD digits are used to express “year”. It is assumed that 2 BCD digits of xY denote 20xY, but not 19xY or 21xY.

5.13.4.10 Spare Registers and Snoop Pin

The RTC is equipped with 80 bytes spare registers to store important user information, and also has a snoop function to detect the transition of snoop pin. Once the transition defined in register RTC_SPRCTL[1] is detected in snoop pin, the 80 bytes spare registers will be cleared by RTC automatically.

As these 80 bytes spare registers locates in LXT (32.768 kHz) clock domain (it’s asynchronous with system clock domain), a synchronization latency is necessary when writing data to these 80 bytes spare registers. Once CPU writes one of 20 spare registers (RTC_SPR0 ~ RTC_SPR19), it’s necessary to polling bit SPRRDY (RTC_SPRCTL[7]) to check if data is written into registers. CPU could only access (read or write) the spare registers again once SPRRDY is high. Any access (read or write) to spare registers while SPRRDY low is undefined.

5.13.5 Register and Memory Map

R: read only, **W:** write only, **R/W:** both read and write, **C:** write 1 to clear

Register	Offset	R/W	Description	Reset Value
RTC Base Address: **RTC_BA = 0x4000_8000**				
RTC_INIR	RTC_BA+0x00	R/W	RTC Initiation Register	0x0000_0000
RTC_AER	RTC_BA+0x04	R/W	RTC Access Enable Register	0x0000_0000
RTC_FCR	RTC_BA+0x08	R/W	RTC Frequency Compensation Register	0x0000_0700
RTC_TLR	RTC_BA+0x0C	R/W	Time Loading Register	0x0000_0000
RTC_CLR	RTC_BA+0x10	R/W	Calendar Loading Register	0x0005_0101
RTC_TSSR	RTC_BA+0x14	R/W	Time Scale Selection Register	0x0000_0001
RTC_DWR	RTC_BA+0x18	R/W	Day of the Week Register	0x0000_0006
RTC_TAR	RTC_BA+0x1C	R/W	Time Alarm Register	0x0000_0000
RTC_CAR	RTC_BA+0x20	R/W	Calendar Alarm Register	0x0000_0000
RTC_LIR	RTC_BA+0x24	R	Leap Year Indicator Register	0x0000_0000
RTC_RIER	RTC_BA+0x28	R/W	RTC Interrupt Enable Register	0x0000_0000
RTC_RIIR	RTC_BA+0x2C	R/W	RTC Interrupt Indication Register	0x0000_0000
RTC_TTR	RTC_BA+0x30	R/W	RTC Time Tick Register	0x0000_0000
RTC_SPRCTL	RTC_BA+0x3C	R/W	RTC Spare Functional Control Register	0x0000_0080
RTC_SPR0	RTC_BA+0x40	R/W	RTC Spare Register 0	0x0000_0000
RTC_SPR1	RTC_BA+0x44	R/W	RTC Spare Register 1	0x0000_0000
RTC_SPR2	RTC_BA+0x48	R/W	RTC Spare Register 2	0x0000_0000
RTC_SPR3	RTC_BA+0x4C	R/W	RTC Spare Register 3	0x0000_0000
RTC_SPR4	RTC_BA+0x50	R/W	RTC Spare Register 4	0x0000_0000
RTC_SPR5	RTC_BA+0x54	R/W	RTC Spare Register 5	0x0000_0000
RTC_SPR6	RTC_BA+0x58	R/W	RTC Spare Register 6	0x0000_0000
RTC_SPR7	RTC_BA+0x5C	R/W	RTC Spare Register 7	0x0000_0000
RTC_SPR8	RTC_BA+0x60	R/W	RTC Spare Register 8	0x0000_0000
RTC_SPR9	RTC_BA+0x64	R/W	RTC Spare Register 9	0x0000_0000
RTC_SPR10	RTC_BA+0x68	R/W	RTC Spare Register 10	0x0000_0000
RTC_SPR11	RTC_BA+0x6C	R/W	RTC Spare Register 11	0x0000_0000
RTC_SPR12	RTC_BA+0x70	R/W	RTC Spare Register 12	0x0000_0000
RTC_SPR13	RTC_BA+0x74	R/W	RTC Spare Register 13	0x0000_0000

Register	Offset	R/W	Description	Reset Value
RTC_SPR14	RTC_BA+0x78	R/W	RTC Spare Register 14	0x0000_0000
RTC_SPR15	RTC_BA+0x7C	R/W	RTC Spare Register 15	0x0000_0000
RTC_SPR16	RTC_BA+0x80	R/W	RTC Spare Register 16	0x0000_0000
RTC_SPR17	RTC_BA+0x84	R/W	RTC Spare Register 17	0x0000_0000
RTC_SPR18	RTC_BA+0x88	R/W	RTC Spare Register 18	0x0000_0000
RTC_SPR19	RTC_BA+0x8C	R/W	RTC Spare Register 19	0x0000_0000

5.13.6 Register Description

RTC Initiation Register (RTC_INIR)

Register	Offset	R/W	Description	Reset Value
RTC_INIR	RTC_BA+0x00	R/W	RTC Initiation Register	0x0000_0000

31	30	29	28	27	26	25	24
INIR							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
INIR							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
INIR							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INIR							**INIR[0]/ ACTIVE**

Bits	Description	
[31:1]	**INIR**	**RTC Initiation (Write Only)** When RTC block is powered on, RTC is at reset state. User has to write a number (0x a5eb1357) to INIR to make RTC leaving reset state. Once the INIR is written as 0xa5eb1357, the RTC will be in un-reset state permanently. The INIR is a write-only field and read value will be always “0”.
[0]	**ACTIVE**	**RTC Active Status (Read Only)** 1 = RTC is at normal active state. 0 = RTC is at reset state

RTC Access Enable Register (RTC_AER)

Register	Offset	R/W	Description	Reset Value
RTC_AER	RTC_BA+0x04	R/W	RTC Access Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**ENF**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
AER							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AER							

Bits	Description	
[31:17]	**Reserved**	**RESERVED**
[16]	**ENF**	**RTC Register Access Enable Flag (Read Only)** 1 = RTC register read/write Enabled 0 = RTC register read/write Disabled This bit will be set after AER[15:0] register is load a 0xA965, and be cleared automatically 512 RTC clocks or AER[15:0] is not 0xA965.

Register	AER[16] = 1 (Access Enabled)	AER[16] = 0 (Access Disabled)
INIR	R/W	R/W
AER	R/W	R/W
FCR	R/W	-
TLR	R/W	R
CLR	R/W	R
TSSR	R/W	R/W
DWR	R/W	R
TAR	R/W	-
CAR	R/W	-
LIR	R	R
RIER	R/W	R/W
RIIR	R/W	R/W
TTR	R/W	-
SPRCTL	R/W	-

		SPR0~SPR19	R/W	-
[15:0]	**AER**	**RTC Register Access Enable Password (Write Only)** 0xa965 = RTC access Enabled Others = RTC access Disabled		

RTC Frequency Compensation Register (RTC_FCR)

Register	Offset	R/W	Description	Reset Value
RTC_FCR	RTC_BA+0x08	R/W	RTC Frequency Compensation Register	0x0000_0700

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**INTEGER**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**FRACTION**					

Bits	Description	
[31:12]	**Reserved**	**Reserved**
[11:8]	**INTEGER**	**Integer Part** <table><tr><th>Integer Part of Detected Value</th><th>FCR[11:8]</th><th>Integer Part of Detected Value</th><th>FCR[11:8]</th></tr><tr><td>32776</td><td>1111</td><td>32768</td><td>0111</td></tr><tr><td>32775</td><td>1110</td><td>32767</td><td>0110</td></tr><tr><td>32774</td><td>1101</td><td>32766</td><td>0101</td></tr><tr><td>32773</td><td>1100</td><td>32765</td><td>0100</td></tr><tr><td>32772</td><td>1011</td><td>32764</td><td>0011</td></tr><tr><td>32771</td><td>1010</td><td>32763</td><td>0010</td></tr><tr><td>32770</td><td>1001</td><td>32762</td><td>0001</td></tr><tr><td>32769</td><td>1000</td><td>32761</td><td>0000</td></tr></table>
[7:6]	**Reserved**	**Reserved**
[5:0]	**FRACTION**	**Fraction Part** Formula = (fraction part of detected value) x 64 **Note:** Digit in FCR must be expressed as hexadecimal number.

Note: This register can be read back after the RTC enable is active by AER.

RTC Time Loading Register (RTC_TLR)

Register	Offset	R/W	Description	Reset Value
RTC_TLR	RTC_BA+0x0C	R/W	Time Loading Register	0x0000_0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="2">Reserved</td><td colspan="2">10HR</td><td colspan="4">1HR</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td>Reserved</td><td colspan="3">10MIN</td><td colspan="4">1MIN</td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>Reserved</td><td colspan="3">10SEC</td><td colspan="4">1SEC</td></tr>
</table>

Bits	Description	
[31:22]	**Reserved**	Reserved
[21:20]	**10HR**	10 Hour Time Digit (0~2)
[19:16]	**1HR**	1 Hour Time Digit (0~9)
[15]	**Reserved**	Reserved
[14:12]	**10MIN**	10 Min Time Digit (0~5)
[11:8]	**1MIN**	1 Min Time Digit (0~9)
[7]	**Reserved**	Reserved
[6:4]	**10SEC**	10 Sec Time Digit (0~5)
[3:0]	**1SEC**	1 Sec Time Digit (0~9)

Note: TLR is a BCD digit counter and RTC will not check the loaded data.

The reasonable value range is listed in the parenthesis.

RTC Calendar Loading Register (RTC_CLR)

Register	Offset	R/W	Description	Reset Value
RTC_CLR	RTC_BA+0x10	R/W	Calendar Loading Register	0x0005_0101

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="4">10YEAR</td><td colspan="4">1YEAR</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td colspan="3">Reserved</td><td>10MON</td><td colspan="4">1MON</td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="2">Reserved</td><td colspan="2">10DAY</td><td colspan="4">1DAY</td></tr>
</table>

Bits	Description	
[31:24]	**Reserved**	Reserved
[23:20]	**10YEAR**	10 Year Calendar Digit (0~9)
[19:16]	**1YEAR**	1 Year Calendar Digit (0~9)
[12]	**10MON**	10 Month Calendar Digit (0~1)
[11:8]	**1MON**	1 Month Calendar Digit (0~9)
[5:4]	**10DAY**	10 Day Calendar Digit (0~3)
[3:0]	**1DAY**	1 Day Calendar Digit (0~9)

Note: CLR is a BCD digit counter and RTC will not check loaded data.

The reasonable value range is listed in the parenthesis.

RTC Time Scale Selection Register (RTC_TSSR)

Register	Offset	R/W	Description	Reset Value
RTC_TSSR	RTC_BA+0x14	R/W	Time Scale Selection Register	0x0000_0001

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**24hr_12hr**

Bits	Description	
[31:1]	**Reserved**	**Reserved**
[0]	**24hr_12hr**	**24-Hour / 12-Hour Mode Selection** It indicates that TLR and TAR are in 24-hour mode or 12-hour mode 1 = select 24-hour time scale 0 = select 12-hour time scale with AM and PM indication

24-hour Time Scale	12-hour Time Scale	24-hour Time Scale	12-hour Time Scale (PM Time + 20)
00	12(AM12)	12	32(PM12)
01	01 (AM01)	13	21 (PM01)
02	02(AM02)	14	22(PM02)
03	03(AM03)	15	23(PM03)
04	04 (AM04)	16	24 (PM04)
05	05(AM05)	17	25(PM05)
06	06(AM06)	18	26(PM06)
07	07(AM07)	19	27(PM07)
08	08(AM08)	20	28(PM08)
09	09(AM09)	21	29(PM09)
10	10 (AM10)	22	30 (PM10)
11	11 (AM11)	23	31 (PM11)

RTC Day of the Week Register (RTC_DWR)

Register	Offset	R/W	Description	Reset Value
RTC_DWR	RTC_BA+0x18	R/W	Day of the Week Register	0x0000_0006

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**DWR**		

Bits	Description	
[31:3]	**Reserved**	**Reserved**
[2:0]	**DWR**	**Day of the Week Register** **DWR** / **Day of the Week** 0 – Sunday 1 – Monday 2 – Tuesday 3 – Wednesday 4 – Thursday 5 – Friday 6 – Saturday

DWR	Day of the Week
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

RTC Time Alarm Register (RTC_TAR)

Register	Offset	R/W	Description	Reset Value
RTC_TAR	RTC_BA+0x1C	R/W	Time Alarm Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved		**10HR**		**1HR**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	**10MIN**			**1MIN**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**10SEC**			**1SEC**			

Bits	Description	
[31:22]	**Reserved**	Reserved
[21:20]	**10HR**	10 Hour Time Digit of Alarm Setting (0~2)
[19:16]	**1HR**	1 Hour Time Digit of Alarm Setting (0~9)
[15]	**Reserved**	Reserved
[14:12]	**10MIN**	10 Min Time Digit of Alarm Setting (0~5)
[11:8]	**1MIN**	1 Min Time Digit of Alarm Setting (0~9)
[7]	**Reserved**	Reserved
[6:4]	**10SEC**	10 Sec Time Digit of Alarm Setting (0~5)
[3:0]	**1SEC**	1 Sec Time Digit of Alarm Setting (0~9)

Note:

TAR is a BCD digit counter and RTC will not check loaded data.

The reasonable value range is listed in the parenthesis.

The register can be read back after the RTC unit is active by AER

RTC Calendar Alarm Register (RTC_CAR)

Register	Offset	R/W	Description	Reset Value
RTC_CAR	RTC_BA+0x20	R/W	Calendar Alarm Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
10YEAR				**1YEAR**			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**10MON**	**1MON**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**10DAY**		**1DAY**			

Bits	Description	
[31:24]	**Reserved**	Reserved
[23:20]	**10YEAR**	10 Year Calendar Digit of Alarm Setting (0~9)
[19:16]	**1YEAR**	1 Year Calendar Digit of Alarm Setting (0~9)
[12]	**10MON**	10 Month Calendar Digit of Alarm Setting (0~1)
[11:8]	**1MON**	1 Month Calendar Digit of Alarm Setting (0~9)
[5:4]	**10DAY**	10 Day Calendar Digit of Alarm Setting (0~3)
[3:0]	**1DAY**	1 Day Calendar Digit of Alarm Setting (0~9)

Note:

CAR is a BCD digit counter and RTC will not check loaded data.

The reasonable value range is listed in the parenthesis.

The register can be read back after the RTC unit is active by AER

RTC Leap Year Indication Register (RTC_LIR)

Register	Offset	R/W	Description	Reset Value
RTC_LIR	RTC_BA+0x24	R	Leap Year Indicator Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**LIR**

Bits	Description	
[31:1]	**Reserved**	**Reserved**
[0]	**LIR**	**Leap Year Indication REGISTER (Read Only).** 1 = This year is leap year 0 = This year is not a leap year

RTC Interrupt Enable Register (RTC_RIER)

Register	Offset	R/W	Description	Reset Value
RTC_RIER	RTC_BA+0x28	R/W	RTC Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**SNOOPIER**	**TIER**	**AIER**

Bits	Description	
[31:3]	**Reserved**	**Reserved**
[2]	**SNOOPIER**	**Snooper Pin Event Detection Interrupt Enable** 1 = Snooper Pin Event Detection Interrupt is enabled 0 = Snooper Pin Event Detection Interrupt is disabled.
[1]	**TIER**	**Time Tick Interrupt and Wake-up by Tick Enable** 1 = RTC Time Tick Interrupt is enabled 0 = RTC Time Tick Interrupt is disabled.
[0]	**AIER**	**Alarm Interrupt Enable** 1 = RTC Alarm Interrupt is enabled 0 = RTC Alarm Interrupt is disabled

RTC Interrupt Indication Register (RTC_RIIR)

Register	Offset	R/W	Description	Reset Value
RTC_RIIR	RTC_BA+0x2C	R/W	RTC Interrupt Indication Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					SNOOPIS	TIS	AIS

Bits	Description	
[31:3]	**Reserved**	**Reserved**
[2]	**SNOOPIS**	**Snooper Pin Event Detection Interrupt Status** When SNOOPEN is high and an event defined by SNOOPEDGE detected in snooper pin, this flag will be set. While this bit is set and SNOOPIER is also high, RTC will generate an interrupt to CPU. Write "1" to clear this bit to "0". 1 = Snooper pin event defined by SNOOPEDGE detected. 0 = Snooper pin event defined by SNOOPEDGE never detected.
[1]	**TIS**	**RTC Time Tick Interrupt Status** RTC unit will set TIF to high periodically in the period selected by TTR[2:0]. When this bit is set and TIER is also high, RTC will generate an interrupt to CPU. This bit is cleared by writing "1" to it through software. 1 = RTC Time Tick Interrupt is requested 0 = RCT Time Tick Interrupt condition never occurred.
[0]	**AIS**	**RTC Alarm Interrupt Status** RTC unit will set AIS to high once the RTC real time counters TLR and CLR reach the alarm setting time registers TAR and CAR. When this bit is set and AIER is also high, RTC will generate an interrupt to CPU. This bit is cleared by writing "1" to it through software. 1 = RTC Alarm Interrupt is requested if RIER.AIER=1 0 = RCT Alarm Interrupt condition never occurred.

RTC Time Tick Registeter (RTC_TTR)

Register	Offset	R/W	Description	Reset Value
RTC_TTR	RTC_BA+0x30	R/W	RTC Time Tick Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**TWKE**	**TTR**		

Bits	Description	
[31:1]	**Reserved**	**Reserved**
[3]	**TWKE**	**RTC Timer Wake-up CPU Function Enable Bit** If TWKE is set before CPU enters power-down mode, when a RTC Time Tick, CPU will be wakened up by RTC unit. 1 = Wake-up function Enabled so that CPU can be waken up from Power-down mode by Time Tick. 0 = Time Tick wake-up CPU function Disabled. **Note:** Tick timer setting follows the TTR description.
[2:0]	**TTR**	**Time Tick Register** The RTC time tick period for Periodic Time Tick Interrupt request. TTR: 0 = 1; 1 = 1/2; 2 = 1/4; 3 = 1/8; 4 = 1/16; 5 = 1/32; 6 = 1/64; 7 = 1/128 (see table below) **Note:** This register can be read back after the RTC is active by AER.

TTR	Time tick (second)
0	1
1	1/2
2	1/4
3	1/8
4	1/16
5	1/32
6	1/64
7	1/128

RTC Spare Function Controller Register (RTC_SPRCTL)

Register	Offset	R/W	Description	Reset Value
RTC_SPRCTL	RTC_BA+0x3C	R/W	RTC Spare Functional Control Register	0x0000_0080

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPRRDY	Reserved					SNOOPEDGE	SNOOPEN

Bits	Description	
[31:8]	**Reserved**	**Reserved**
[7]	**SPRRDY**	**SPR Register Ready** This bit indicates if the registers SPR0 ~ SPR19 are ready to read. After CPU writing registers SPR0 ~ SPR19, polling this bit to check if SP0 ~ SPR19 are updated done is necessary. This it is read only and any write to this bit won't take any effect. 1 = SPR0 ~ SPR19 are updated done and ready to read. 0 = SPR0 ~ SPR19 updating is in progress.
[6:2]	**Reserved**	**Reserved**
[1]	**SNOOPEDGE**	**Snooper Active Edge Selection** This bit defines which edge of snooper pin will generate a snooper pin detected event to clear the 20 spare registers. 1 = Falling edge of snooper pin generates snooper pin detected event. 0 = Rising edge of snooper pin generates snooper pin detected event.
[0]	**SNOOPEN**	**Snooper Pin Event Detection Enable** This bit enables the snooper pin event detection. When this bit is set high and an event defined by SNOOPEDGE detected, the 20 spare registers will be cleared to "0" by hardware automatically. And, the SNOOPIF will also be set. In addition, RTC will also generate wake-up event to wake system up. 1 = Snooper pin event detection function Enabled. 0 = Snooper pin event detection function Disabled.

RTC Spare Register X (RTC_SPRx)

Register	Offset	R/W	Description	Reset Value
RTC_SPR0	RTC_BA+0x40	R/W	RTC Spare Register 0	0x0000_0000
RTC_SPR1	RTC_BA+0x44	R/W	RTC Spare Register 1	0x0000_0000
RTC_SPR2	RTC_BA+0x48	R/W	RTC Spare Register 2	0x0000_0000
RTC_SPR3	RTC_BA+0x4C	R/W	RTC Spare Register 3	0x0000_0000
RTC_SPR4	RTC_BA+0x50	R/W	RTC Spare Register 4	0x0000_0000
RTC_SPR5	RTC_BA+0x54	R/W	RTC Spare Register 5	0x0000_0000
RTC_SPR6	RTC_BA+0x58	R/W	RTC Spare Register 6	0x0000_0000
RTC_SPR7	RTC_BA+0x5C	R/W	RTC Spare Register 7	0x0000_0000
RTC_SPR8	RTC_BA+0x60	R/W	RTC Spare Register 8	0x0000_0000
RTC_SPR9	RTC_BA+0x64	R/W	RTC Spare Register 9	0x0000_0000
RTC_SPR10	RTC_BA+0x68	R/W	RTC Spare Register 10	0x0000_0000
RTC_SPR11	RTC_BA+0x6C	R/W	RTC Spare Register 11	0x0000_0000
RTC_SPR12	RTC_BA+0x70	R/W	RTC Spare Register 12	0x0000_0000
RTC_SPR13	RTC_BA+0x74	R/W	RTC Spare Register 13	0x0000_0000
RTC_SPR14	RTC_BA+0x78	R/W	RTC Spare Register 14	0x0000_0000
RTC_SPR15	RTC_BA+0x7C	R/W	RTC Spare Register 15	0x0000_0000
RTC_SPR16	RTC_BA+0x80	R/W	RTC Spare Register 16	0x0000_0000
RTC_SPR17	RTC_BA+0x84	R/W	RTC Spare Register 17	0x0000_0000
RTC_SPR18	RTC_BA+0x88	R/W	RTC Spare Register 18	0x0000_0000
RTC_SPR19	RTC_BA+0x8C	R/W	RTC Spare Register 19	0x0000_0000

31	30	29	28	27	26	25	24
SPARE							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
SPARE							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SPARE							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SPARE							

Bits	Description	
[31:0]	**SPARE**	**SPARE** This field is used to store back-up information defined by software. This field will be cleared by hardware automatically once a snooper pin event is detected.

5.14 UART Controller

5.14.1 Overview

The UART controllers provides up to two channels of Universal Asynchronous Receiver/Transmitter (UART) modules that are UART0 and UART1. (UART0 is at APB1 and UART1 is at APB2).

The Universal Asynchronous Receiver/Transmitter (UART) performs a serial-to-parallel conversion on data received from the peripheral, and a parallel-to-serial conversion on data transmitted from the CPU. The UART controller also supports IrDA (SIR) function mode, LIN Master/Slave function mode and RS-485 function mode. Each UART channel supports nine types of interrupts including receiver threshold level reaching interrupt (INT_RDA), transmitter FIFO empty interrupt (INT_THRE), line status interrupt (break error, parity error, framing error or RS-485 interrupt) (INT_RLS), time-out interrupt (INT_TOUT), MODEM status interrupt (INT_MODEM), Buffer error interrupt (INT_BUF_ERR), wake-up interrupt (INT_WAKE), auto-baud rate detect or auto-baud rate counter overflow flag (INT_ABAUD) and LIN function interrupt (INT_LIN).

The UART0 and UART1 are built-in with a 16-byte transmitter FIFO (TX_FIFO) and a 16-byte receiver FIFO (RX_FIFO) that reduces the number of interrupts presented to the CPU. The CPU can read the status of the UART at any time during the operation. The reported status information includes the type and condition of the transfer operations being performed by the UART, as well as 3 error conditions (parity error, framing error or break interrupt) occur while receiving data. The UART controller supports auto-baud rate detection. The auto-baud rate detection controls the process of measuring the incoming clock/data rate for the baud rate generation and can be read and written at user discretion. The UART controller also support incoming data or CTSn wake-up function. When the system is in power-down mode, an incoming data or CTSn signal will wake-up CPU from power-down mode. The UART includes a programmable baud rate generator that is capable of dividing crystal clock input by divisors to produce the clock that transmitter and receiver need. The baud rate equation is Baud Rate = UART_CLK / [BRD + 1], where BRD are defined in UART Baud Rate Divider Register (UARTx_BAUD). Below table lists the equations in the various conditions and the UART baud rate setting table.

DIV_16_EN	**BRD**	**Baud Rate Equation**
Disable (Mode 0)	A	UART_CLK / (A+1), A must >8
Enable (Mode 1)	A	UART_CLK / [16 * (A+1)]

Table 5.14-1 UART Baud Rate Equation

System clock =12 MHz		
Baud rate	Mode 0	Mode 1
921600	A=12	Not Supported
460800	A=25	Not Supported
230400	A=51	A=2
115200	A=103	A=6
57600	A=207	A=12
38400	A=311	A=19

19200	A=624	A=38
9600	A=1249	A=77
4800	A=2499	A=155

Table 5.14-2 UART Baud Rate Setting

5.14.1.1 Auto-Flow Control

The UART0 and UART1 controllers support auto-flow control function that uses two low-level signals, CTSn (clear-to-send) and RTSn (request-to-send) to control the flow of data transfer between the UART and external devices (ex: Modem). When auto-flow is enabled, the UART is not allowed to receive data until the UART asserts RTSn (RTSn high) to external device. When the number of bytes in the RX-FIFO equals the value of UART_TLCTL [RTS_TRI_LEV], the RTSn is de-asserted. The UART sends data out when UART controller detects CTSn is asserted (CTSn high) from external device. If a valid asserted CTSn is not detected the UART controller will not send data out.

5.14.1.2 Auto-Baud Rate Detection

The UART0 and UART1 controllers support auto-baud rate detection. The auto-baud rate function can be used to measure the receiver incoming data baud rate. If enabled the auto-baud feature, UART controller will measure the bit time of the received data stream and set the divisor latch registers UART_BARD. Auto-baud rate detection is started by setting the UART_CTL [ABAUD_EN].

5.14.1.3 UART Wake-Up Function

The UART0 and UART1 controllers support wake-up system function. The wake-up function includes CTSn wake-up function (UART_CTL [WAKE_CTS_EN]) and data wake-up function (UART_CTL [WAKE_DATA_EN]). When the system is operation in power-down mode, the UART can wake-up system by CTSn pin or by incoming data.

5.14.1.4 IrDA Function Mode

The UART controllers also provides Serial IrDA (SIR, Serial Infrared) function (User must set UART_FUN_SEL to select IrDA function). The SIR specification defines a short-range infrared asynchronous serial transmission mode with one start bit, 8 data bits, and 1 stop bit. The maximum data rate is 115.2 Kbps (half duplex). The IrDA SIR block contains an IrDA SIR Protocol encoder/decoder. The IrDA SIR protocol is half-duplex only. So it cannot transmit and receive data at the same time. The IrDA SIR physical layer specifies a minimum 10 ms transfer delay between transmission and reception, and in IrDA Operation mode the UART_BAUD setting must be mode1 (UART_BAUD [DIV_16_EN] = “1”).

5.14.1.5 RS-485 Function Mode

Another alternate function of UART controllers is RS-485 9 bit mode function whose direction control can be controlled by RTSn pin or GPIO. The RS-485 function mode is selected by setting the UART_FUN_SEL register to select RS-485 function. The RS-485 driver control is implemented by using the RTSn control signal from an asynchronous serial port to enable the RS-485 driver. In RS-485 mode, many characteristics of the RX and TX are same as UART.

5.14.1.6 LIN Function Mode

The LIN mode is selected by setting the LIN_EN bit in UART_FUN_SEL register. In LIN mode, one start bit and 8-bit data format with 1-bit stop bit are required in accordance with the LIN standard.

5.14.2 Features

- Full duplex, asynchronous communications.
- Separate receiving / transmitting 16 bytes entry FIFO for data payloads.
- Supports hardware auto-flow control/flow control function (CTSn, RTSn) and programmable (CTSn, RTSn) flow control trigger level.
- Supports programmable baud rate generator for each channel.
- Supports auto-baud rate detect function.
- Supports programmable receiver buffer trigger level.
- Supports incoming data or CTSn to wake-up function.
- Supports 9 bit receiver buffer time-out detection function.
- All UART channels can be served by the PDMA controller.
- Programmable transmitting data delay time between the last stop bit leaving the TX-FIFO and the de-assertion by setting UART_TMCTL [DLY] register.
- Supports break error, frame error, parity error and receiving / transmitting buffer overflow detect function.
- Fully programmable serial-interface characteristics:
  - Programmable number of data bit, 5, 6, 7, 8 character.
  - Programmable parity bit, even, odd, no parity or stick parity bit generation and detection.
  - Programmable stop bit, 1, 1.5, or 2 stop bit generation.
- Supports IrDA SIR function mode
  - Supports 3/16 bit period modulation.
- Supports LIN function mode.
  - Supports LIN Master/Slave mode
  - Supports programmable break generation function for transmitter.
  - Supports break detect function for receiver.
- Supports RS-485 function mode.
  - Supports RS-485 9bit mode.
  - Supports hardware or software controls RTSn or software control GPIO to control transfer direction.

5.14.3 Block Diagram

The UART clock control and block diagram are shown as follows. The UART controller is completely asynchronous design with two clock domains, PCLK and engine clock. Note that the PCLK should be higher than or equal to the frequency of engine clock.

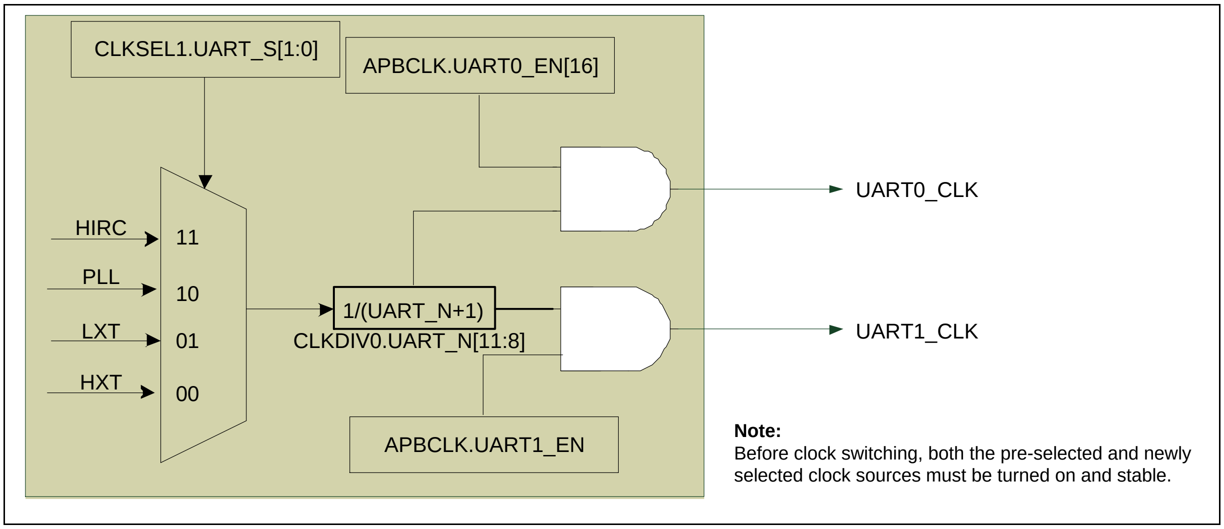


Figure 5.14-1 UART Clock Control Diagram

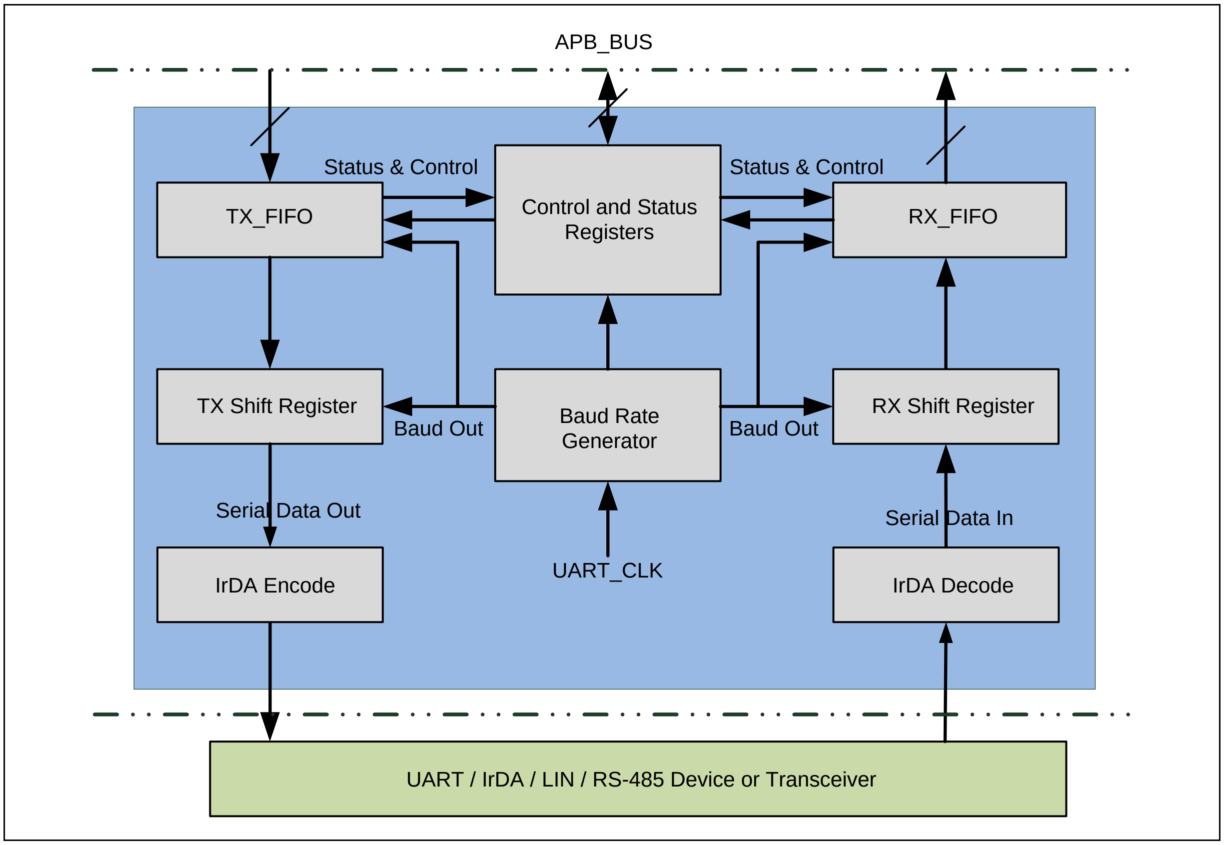


Figure 5.14-2 UART Block Diagram

TX_FIFO

The transmitter buffered is a 16 byte FIFO to reduce the number of interrupts presented to the CPU.

RX_FIFO

The receiver buffered is a 16 byte FIFO (plus three error bits per byte) to reduce the number of interrupts presented to the CPU.

TX Shift Register

The block shifts the transmitting data out serially control block.

RX Shift Register

The block shifts the receiving data in serially control block.

Baud Rate Generator

Divide the external clock or internal clock by the divisor to get the desired baud rate clock. Refer to for baud rate equation.

IrDA Encode

This block is the IrDA encode control block.

IrDA Decode

This block is the IrDA decode control block.

Control and Status Register

This is a register set, including the transfer line control registers (UART_TLCTL), transfer status registers (UART_TRSR), and control register (UART_CTL) for transmitter and receiver. The time-out control register (UART_TMCTL) identifies the condition of time-out interrupt. This register set also includes the interrupt enable register (UART_IER) and interrupt status register (UART_ISR) to enable or disable the responding interrupt and to identify the occurrence of the responding interrupt. There are nine types of interrupts including receiver threshold level reaching interrupt (INT_RDA), transmitter FIFO empty interrupt (INT_THRE), line status interrupt (break error, parity error, framing error or RS-485 interrupt) (INT_RLS), time-out interrupt (INT_TOUT), MODEM status interrupt (INT_MODEM), Buffer error interrupt (INT_BUF_ERR), wake-up interrupt (INT_WAKE), auto-baud rate detect or auto-baud rate counter overflow flag (INT_ABAUD) of LIN function interrupt (INT_LIN).

5.14.4 Functional Description

5.14.4.1 Auto-Flow Control

The following diagram demonstrates the auto-flow control block diagram.

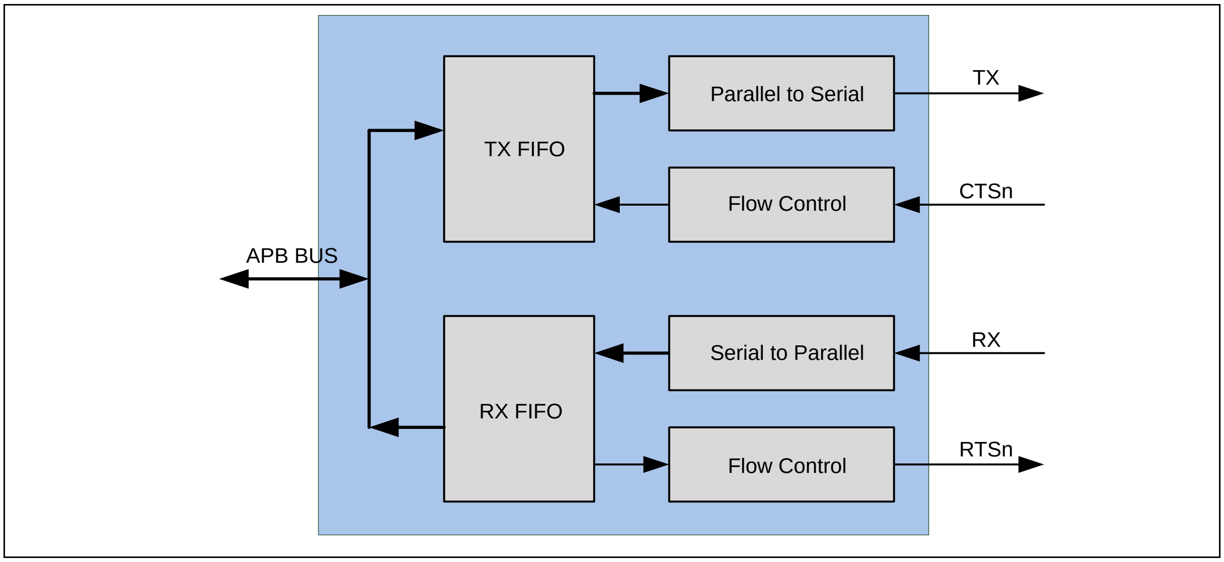


Figure 5.14-3 UART Auto-Flow Control Block Diagram

5.14.4.2 Auto-Baud Rate Detect

The UART supports auto-baud rate detection. If auto-baud feature enabled, controller will measure the bit time of the received data stream (LSB must be “1”) and set the divisor latch registers UART_BARD. Auto-baud rate detection is started by setting the UART_CTL [ABAUD_EN]. When the auto-baud rate detection flow finishes, the ABAUD_EN bit will be cleared automatically, and the UART_ISR [ABAUD_IS] and UART_TRSR [ABAUD_F] will be setting. If have time-out occurs (baud rate counter overflow), the UART_ISR [ABAUD_IS] and UART_TRSR [ABAUD_TOUT_F] will be setting. The following diagram demonstrates the auto-baud rate detection function.

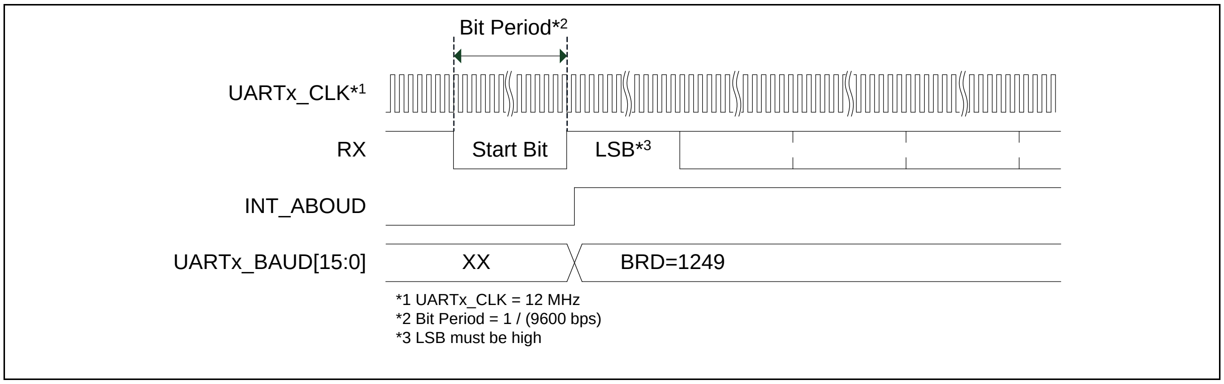


Figure 5.14-4 UART Auto-Baud Rate Block Diagram

5.14.4.3 Wake-Up Function

The UART0 and UART1 controllers support wake-up system function. The wake-up function includes CTSn wake-up function (UART_CTL [WAKE_CTS_EN]) and data wake-up function (UART_CTL [WAKE_DATA_EN]). When the system is in power-down, the UART can wake-up system by CTSn pin or by incoming data. When incoming data wakes system up, the incoming data will be received and stored in FIFO, and controller will clear the UART_CTL [WAKE_DATA_EN] register automatically. The following diagram demonstrates the wake-up function.

CTSn Wake-Up Case 1

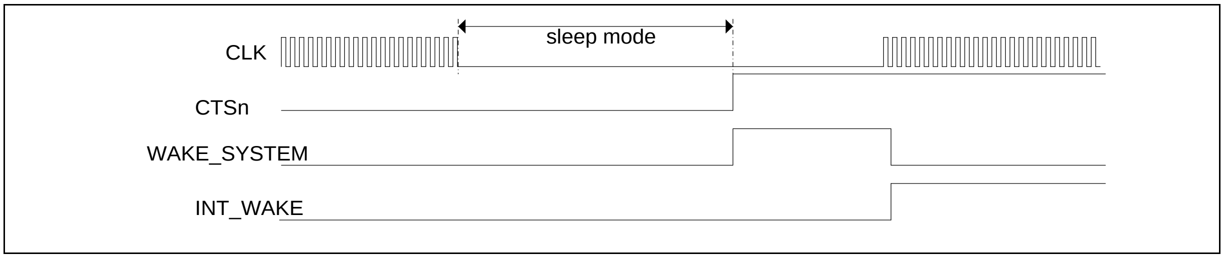


Figure 5.14-5 UART CTSn Wake-Up Case 1

CTSn Wake-Up Case 2

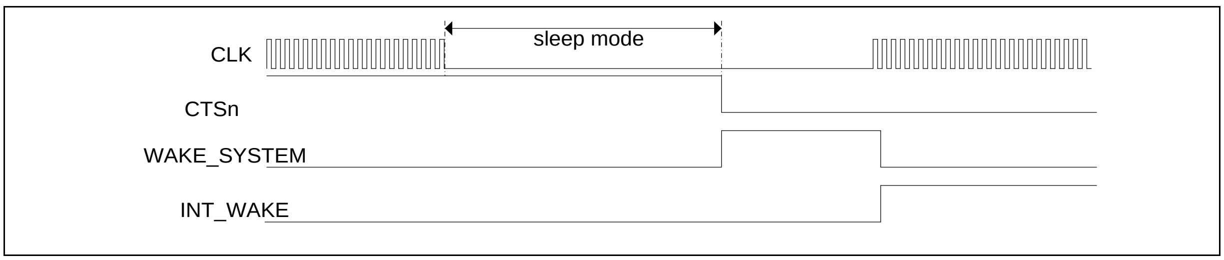


Figure 5.14-6 UART CTSn Wake-Up Case 2

Data Wake-Up

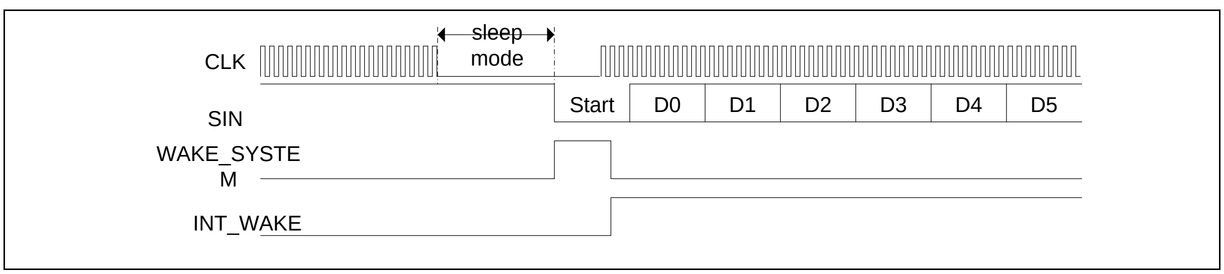


Figure 5.14-7 UART DATA Wake-Up

5.14.4.4 IrDA Function Mode

The UART support IrDA SIR (Serial Infrared) Transmit Encoder and Receive Decoder. IrDA mode is selected by setting the FUN_SEL bit in UART_FUN_SEL register to select IrDA mode and when operating in IrDA mode, the receive FIFO trigger level must be “1” by setting UART_TLCTL [RFITL] = “0”.

The UART_BAUD [DIV_16_EN] bit must be enabled in IrDA mode operation.

Baud Rate = Clock / (16 * (BRD + 1)), where BRD is Baud Rate Divider in UART_BAUD register.

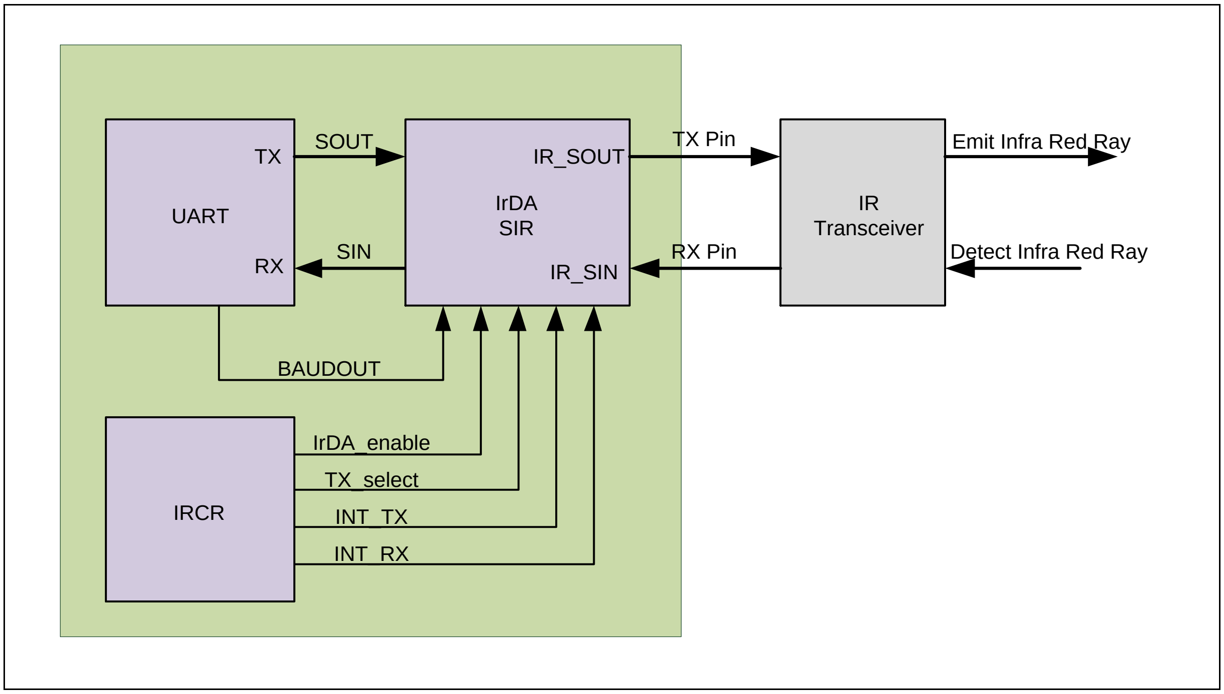


Figure 5.14-8 IrDA Block Diagram

IrDA SIR Transmit Encoder

The IrDA SIR Transmit Encoder modulates Non-Return-to Zero (NRZ) transmit bit stream output from UART. The IrDA SIR physical layer specifies the usage of Return-to-Zero, Inverted (RZI) mlation scheme which represent logic 0 as an infra light pulse. The modulated output pulse stream is transmitted to an external output driver and infrared Light Emitting Diode.

The transmitted pulse width is specified as 3/16 period of baud rate.

IrDA SIR Receive Decoder

The IrDA SIR Receive Decoder demodulates the return-to-zero bit stream from the input detector and outputs the NRZ serial bits stream to the UART received data input. The decoder input is normally high in the idle state. (Because of this, IRCR bit 6 should be set as “1” by default)

A start bit is detected when the decoder input is LOW

IrDA SIR Operation

The IrDA SIR Encoder/decoder provides functionality which converts between UART data stream and half duplex serial SIR interface. The following diagram is IrDA encoder/decoder waveform:

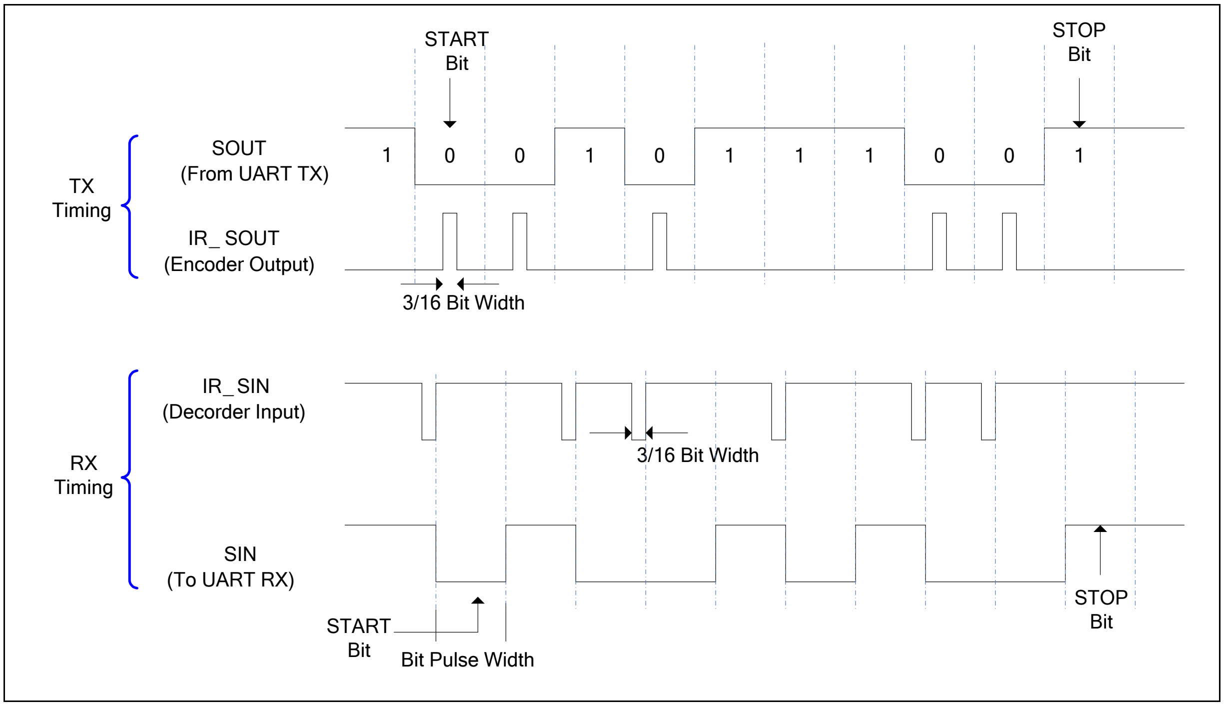


Figure 5.14-9 IrDA TX/RX Timing Diagram

5.14.4.5 RS-485 Function Mode

The UART supports RS-485 9-bit mode function. The RS-485 mode is selected by setting the UART_FUN_SEL register to select RS-485 function and when operating in RS-485 mode, the receive FIFO trigger level must be "1" by setting UART_TLCTL [RFITL] = "0". The RS-485 driver control is implemented by the RTSn control signal from an asynchronous serial port to enable the RS-485 driver. In RS-485 mode, many characteristics of the RX and TX are same as UART.

In RS-485 function mode, the bit 9 will be configured as address bit. For data characters, the bit 9 is set to "0". Software can program UART_TLCTL register to control the 9-th bit (When the PBE, EPE and SPE are set, the 9-th bit is transmitted as "0" and when PBE and SPE are set and EPE is cleared, the 9-th bit is transmitted as "1"). The Controller supports three operation modes that are RS-485 Normal Multi-drop Operation Mode (RS-485 NMM Mode), RS-485 Auto Address Detection Operation Mode (RS-485 AAD Mode) and RS-485 Auto Direction Control Operation Mode (RS-485 AUD Mode). One of the three operation modes can be selected by programming UART_ALT_CTL register, and software can driving the transfer delay time between the last stop bit leaving the TX-FIFO and the de-assertion by setting UART_TMCTL [DLY] register.

RS-485 Normal Multi-drop Operation Mode (RS-485 NMM Mode)

In RS-485 Normal Multi-drop operation mode, software can decide whether receiver will ignore data before an address byte is detected (bit 9 = "1"). When an address byte be detected (bit 9 = "1") by hardware, the address byte data will be stored in the RX-FIFO. Software can decide whether to enable or disable receiver to accept the following data byte by setting UART_CTL [RX_DIS]. If the receiver is be enabled (UART_CTL[RX_DIS] is low), all received byte data will be accepted and stored in the RX-FIFO, and if the receiver is disabled(UART_CTL[RX_DIS] is high), all received byte data will be ignore until the next address byte be detected. If software disable receiver by setting UART_CTL [RX_DIS] register high, when a next address byte is detected, the controller will clear the UART_CTL [RX_DIS] bit and the address byte data will be stored in the RX-FIFO.

Program Sequence Example:

1. Program FUN_SEL in UART_FUN_SEL to select RS-485 function.

2. Program the RX_DIS bit in UART_CTL register to determine whether to store the received data before an address byte is detected (bit 9 = “1”).

3. Program the RS-485_NMM by setting UART_ALT_CTL register.

4. When an address byte is detected (bit 9 = “1”), hardware will set UART_ISR [RLS_IS] and UART_TRSR [RS-485_ADDET_F] flag.

5. Software can decide whether to accept the following data byte by setting UART_CTL [RX_DIS].

6. Repeat step 4 and step 5.

RS-485 Auto Address Detection Operation Mode (RS-485 AAD Mode)

In RS-485 Auto Address Detection Operation Mode, the receiver will ignore any data until an address byte is detected (bit 9 = “1”) and the address byte data match the UART_ALT_CTL [ADDR_MATCH] value. The address byte data will be stored in the RX-FIFO. The following all data will be accepted and stored in the RX-FIFO until an address byte not match the UART_ALT_CTL [ADDR_MATCH] value. In RS-485 AAD mode, don’t fill any value to UART_CTL [RX_DIS] bit.

Program Sequence example:

1. Program FUN_SEL in UART_FUN_SEL to select RS-485 function.
2. Program the RS-485_AAD by setting UART_ALT_CTL register.
3. When an address byte is detected (bit9 = “1”), hardware will compare the address byte and the UART_ALT_CTL [ADDR_MATCH] value.
4. If the address byte matches the UART_ALT_CTL [ADDR_MATCH] value, hardware will set UART_ISR [RLS_IS] and UART_TRSR [RS-485_ADDET_F] flag. And the receiver will sorted address byte to FIFO and accept the following data transfer and stored data in FIFO until next address byte be detected.

   However if the address byte does not match the UART_ALT_CTL [ADDR_MATCH] value, hardware will ignored the address byte data and ignored the following data transfer
5. Respect step 3 and step 4.

RS-485 Auto Direction Mode (RS-485 AUD Mode)

Another option function of RS-485 controllers is RS-485 auto direction control function. The RS-485 driver control is implemented by using the RTSn control signal from an asynchronous serial port to enable the RS-485 driver. The RTSn line is connected to the RS-485 driver enable such that setting the RTSn line to high (logic “1”) will enable the RS-485 driver. Setting the RTSn line to low (logic “0”) will put the driver into the tri-state condition. User can setting LEV_RTS in UART_MCSR register to change the RTSn driving level.

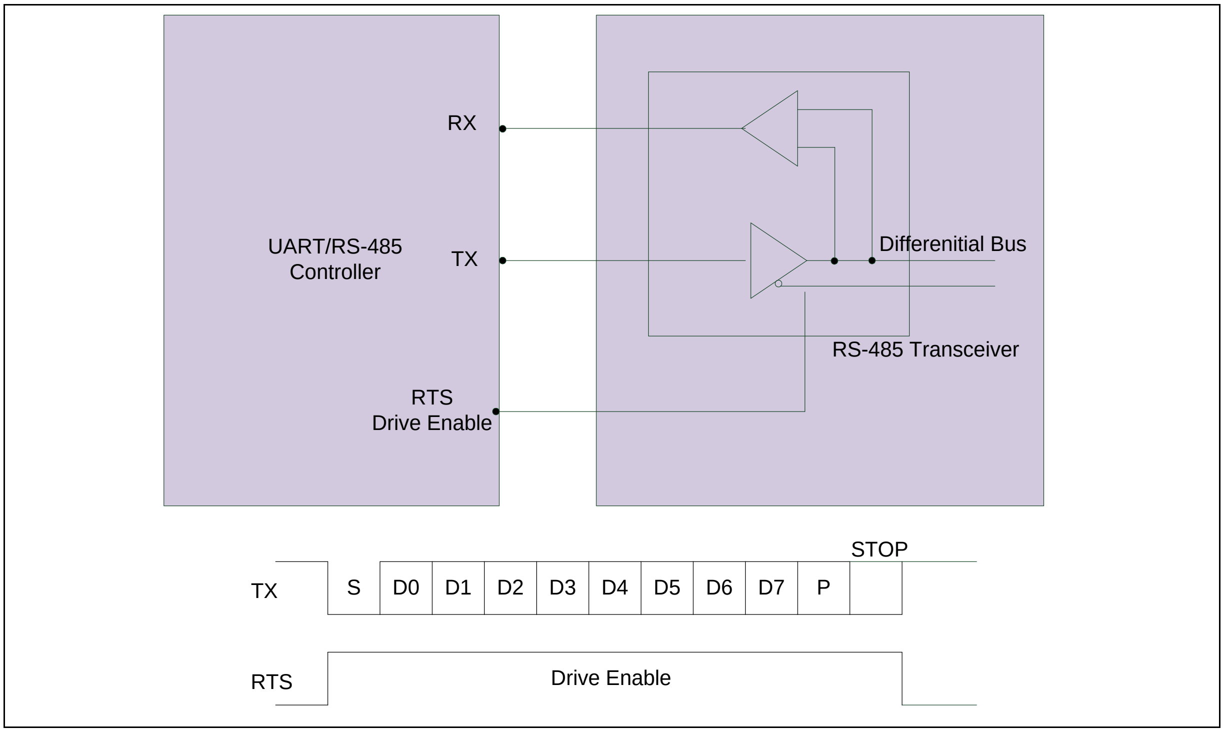


Figure 5.14-10 Structure of RS-485 Frame

5.14.4.6 LIN (Local Interconnection Network) Function Mode

The UART support LIN function, and LIN mode is selected by setting the UART_FUN_SEL register to select LIN function. In LIN function mode, each byte field is initialed by a start bit with value 0 (dominant), followed by 8 data bits (LSB is first) and ended by 1 stop bit with value one (recessive) in accordance with the LIN standard.

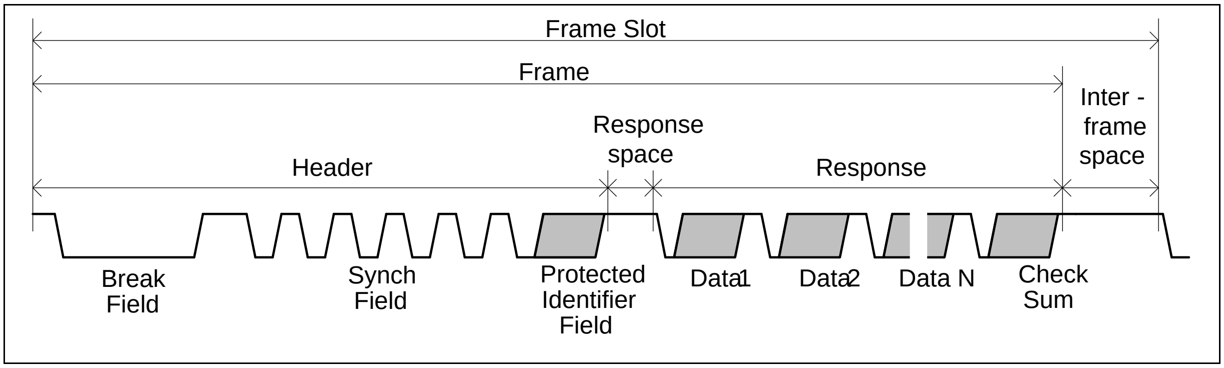


Figure 5.14-11 Structure of LIN Frame

Program Flow of LIN Bus Transmit Transfer (TX)

The program flow of LIN Bus Transmit transfer (TX) is shown as follows

Case 1: The Header Field Select as "Break"

1. Set LIN function mode by setting UART_FUN_SEL register.
2. Choose the data header to "break field" by setting UART_ALT_CTL [LIN_HEAD_SEL]).
3. Enable UART_ALT_CTL [BIT_ERR_EN] bit, and when the SIN pin is not equal to SOUT pin that the hardware will generator an interrupt to CPU). If user wants to receive data at the same time, user must enable UART_ALT_CTL [LIN_RX_EN] bit.
4. Fill LIN_TX_BCNT to choose break field length. (The break field length is LIN_TX_BCNT + 8).
5. Set the LIN_TX_EN bit in UART_ALT_CTL register to start transmission break field, and when break filed operation is finished, hardware will set UART_ISR [LIN_IS] and UART_TRSR [LIN_TX_F] flag and UART_ALT_CTL[LIN_TX_EN will be cleared automatically.
6. Fill 0x55 to UART_THR to request synch field transmission.
7. Fill the protected identifier value (PID) in the UART_THR
8. Fill N bytes data and Checksum to UART_THR then repeat step 4 ~ step 8 to transmit the data.

Case 2: The Header Field Select as "Break + Sync"

1. Set LIN function mode by setting UART_FUN_SEL register.
2. Choose the data header to "break + sync" field by setting UART_ALT_CTL [LIN_HEAD_SEL]).
3. Enable UART_ALT_CTL [BIT_ERR_EN] bit, and when the SIN pin is not equal to SOUT pin that the hardware will generator an interrupt to CPU). If user wants to receive data at the same time, user must enable UART_ALT_CTL [LIN_RX_EN] bit.
4. Fill LIN_TX_BCNT to choose break field length. (The break field length is LIN_TX_BCNT + 8).
5. Set the LIN_TX_EN bit in UART_ALT_CTL register to start transmission break and sync field, and when break and sync filed operation is finished, hardware will set UART_ISR [LIN_IS] and UART_TRSR [LIN_TX_F] flag and UART_ALT_CTL[LIN_TX_EN will be cleared automatically.
6. Fill the protected identifier value (PID) in the UART_THR
7. Fill N bytes data and Checksum to UART_THR then repeat step 4 ~ step 7 to transmit the data.

Case 3: The Header Field Select as "Break + Sync + PID"

1. Set LIN function mode by setting UART_FUN_SEL register.
2. Choose the data header to "break + sync + PID field" by setting UART_ALT_CTL [LIN_HEAD_SEL]).
3. Enable UART_ALT_CTL [BIT_ERR_EN] bit, and when the SIN pin is not equal to SOUT pin that the hardware will generator an interrupt to CPU). If user wants to receive data at the same time, user must enable UART_ALT_CTL [LIN_RX_EN] bit.
4. Fill LIN_TX_BCNT to choose break field length. (The break field length is LIN_TX_BCNT + 8).
5. Set the LIN_TX_EN bit in UART_ALT_CTL register to start transmission break, sync and PID field, and when break, sync and PID filed operation is finished, hardware will set UART_ISR [LIN_IS] and UART_TRSR [LIN_TX_F] flag and UART_ALT_CTL[LIN_TX_EN will be cleared

automatically.

6. Fill N bytes data and Checksum to UART_THR then repeat step 4 ~ step 6 to transmit the data.

Program Flow of LIN Bus Receiver transfer (RX)

The program flow of LIN Bus Receiver transfer (RX) is show as follows.

Case 1: The header Field Select as "Break"

1. Set LIN function mode by setting UART_FUN_SEL register.
2. Choose the data header to "break field" by setting UART_ALT_CTL [LIN_HEAD_SEL]).
3. Set the LIN_RX_EN bit in UART_ALT_CTL register to enable LIN RX mode.
4. Wait for the flag LIN_RX_F in UART_TRSR to check RX received break field or not. (The break field will not be stored in FIFO)
5. Wait for the flag RDA_IF in UART_ISR and read back the UART_RBR register.

Case 2: The Header Field Select as "Break + Sync"

1. Set LIN function mode by setting UART_FUN_SEL register.
2. Choose the data header to "break + sync field" by setting UART_ALT_CTL [LIN_HEAD_SEL]).
3. Set the LIN_RX_EN bit in UART_ALT_CTL register to enable LIN RX mode.
4. Wait for the flag LIN_RX_F in UART_TRSR to check RX received break field and sync field. If the break and sync field is received, hardware will set UART_TRSR [LIN_RX_F] flag If the break be received but the sync field not equal 0x55, hardware will set UART_TRSR [LIN_RX_F] and UART_TRSR [LIN_RX_SYNC_ERR_F] flag. The break and sync data (equal 0x55 or not) will not be stored in FIFO.
5. Wait for the flag RDA_IF in UART_ISR and read back the UART_RBR register.

Case 3: The Header Field Select as "Break + Sync + PID"

1. Set LIN function mode by setting UART_FUN_SEL register.
2. Choose the data header to "break + sync + PID field" by setting UART_ALT_CTL [LIN_HEAD_SEL]).
3. Set the LIN_RX_EN bit in UART_ALT_CTL register to enable LIN RX mode.
4. In this operation mode, hardware will control data automatically. Hardware will ignore any data until received break + sync (0x55) + PID value match the UART_ALT_CTL [ADDR_MATCH] value (break + sync + PID will not be stored in FIFO). When received break + sync (0x55) + PID value match the UART_ALT_CTL [ADDR_MATCH] value, hardware will set UART_TRSR [LIN_RX_F] and the following all data will be accepted and stored in the RX-FIFO until detect next break field. If the receiver received break + wrong sync (not equal 0x55) + PID value, that hardware will set UART_TRSR [LIN_RX_F] and UART_TRSR [LIN_RX_SYNC_ERR_F] flag and the receiver will be disabled. If the receiver received break + sync (0x55) + wrong PID value, that hardware will set UART_TRSR [LIN_RX_F] flag and the receiver will be disabled.
5. Wait for the flag RDA_IF in UART_ISR and read back the UART_RBR register.

5.14.5 Register and Memory Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
UART Base Address: **UART0_BA = 0x4005_0000** **UART1_BA = 0x4015_0000**				
UART_RBR **x=0,1**	UARTx_BA+0x00	R	UART Receive Buffer Register.	Undefined
UART_THR **x=0,1**	UARTx_BA+0x00	W	UART Transmit Holding Register.	Undefined
UART_CTL **x=0,1**	UARTx_BA+0x04	R/W	UART Control State Register.	0x0000_0000
UART_TLCTL **x=0,1**	UARTx_BA+0x08	R/W	UART Transfer Line Control Register.	0x0000_0000
UART_IER **x=0,1**	UARTx_BA+0x0C	R/W	UART Interrupt Enable Register.	0x0000_0000
UART_ISR **x=0,1**	UARTx_BA+0x10	R/W	UART Interrupt Status Register.	0x0000_0002
UART_TRSR **x=0,1**	UARTx_BA+0x14	R/W	UART Transfer State Status Register.	0x0000_0000
UART_FSR **x=0,1**	UARTx_BA+0x18	R/W	UART FIFO State Status Register.	0x0000_0A02
UART_MCSR **x=0,1**	UARTx_BA+0x1C	R/W	UART Modem State Status Register.	0x0002_0002
UART_TMCTL **x=0,1**	UARTx_BA+0x20	R/W	UART Time-Out Control State Register.	0x0000_01FF
UART_BAUD **x=0,1**	UARTx_BA+0x24	R/W	UART Baud Rate Divisor Register	0x0000_0000
UART_IRCR **x=0,1**	UARTx_BA+0x30	R/W	UART IrDA Control Register.	0x0000_0040
UART_ALT_CSR **x=0,1**	UARTx_BA+0x34	R/W	UART Alternate Control State Register.	0x0000_0000
UART_FUN_SEL **x=0,1**	UARTx_BA+0x38	R/W	UART Function Select Register.	0x0000_0000

Note: The x of the UARTx_REG represents the UART channel.

5.14.6 Registers Description

UART Receive Buffer Register (UARTx_RBR)

Register	Offset	R/W	Description	Reset Value
UART_RBR **x=0,1**	UARTx_BA+0x00	R	UART Receive Buffer Register.	Undefined

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RBR							

Bits	Description	
[31:8]	**Reserved**	**Reserved.**
[7:0]	**RBR**	**Receive Buffer Register** By reading this register, the UART will return an 8-bit data received from RX pin (LSB first).

UART Transmit Holding Register (UARTx_THR)

Register	Offset	R/W	Description	Reset Value
UART_THR **x=0,1**	UARTx_BA+0x00	W	UART Transmit Holding Register.	Undefined

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
THR							

Bits	Description	
[31:8]	**Reserved**	**Reserved.**
[7:0]	**THR**	**Transmit Holding Register** By writing to this register, the UART will send out an 8-bit data through the TX pin (LSB first).

UART Control Register (UARTx_CTL)

Register	Offset	R/W	Description	Reset Value
UART_CTL **x=0,1**	UARTx_BA+0x04	R/W	UART Control State Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**ABAUD_EN**	**Reserved**		**WAKE_DATA_EN**	**WAKE_CTS_EN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DMA_TX_EN	**DMA_RX_EN**	**AUTO_CTS_EN**	**AUTO_RTS_EN**	**TX_DIS**	**RX_DIS**	**TX_RST**	**RX_RST**

Bits	Description	
[31:13]	**Reserved**	**Reserved.**
[12]	**ABAUD_EN**	**Auto-Baud Rate Detect Enable** 1 = Auto-baud rate detect function Enabled. 0 = Auto-baud rate detect function Disabled. **Note:** When the auto-baud rate detect operation finishes, hardware will clear this bit and the associated interrupt (INT_ABAUD) will be generated (If UART_IER [ABAUD_IE] be enabled).
[11:10]	**Reserved**	**Reserved.**
[9]	**WAKE_DATA_EN**	**Incoming Data Wake-up Function Enable** 1 = Incoming data wake-up function Enabled when the system is in power-down mode, incoming data will wake-up system from power-down mode. 0 = Incoming data wake-up system Disabled **Note:** Hardware will clear this bit when the incoming data wake-up operation finishes and “system clock” work stable.
[8]	**WAKE_CTS_EN**	**CTSn Wake-Up Function Enable** 1 = Wake-up function Enabled when the system is in power-down mode, an external CTSn change will wake-up system from power-down mode. 0 = CTSn wake-up system function Disabled
[7]	**DMA_TX_EN**	**TX DMA Enable** This bit can enable or disable TX PDMA service. 1 = TX PDMA service function Enabled. 0 = TX PDMA service function Disabled.
[6]	**DMA_RX_EN**	**RX DMA Enable** This bit can enable or disable RX PDMA service.

Bits	Description	
		1 = RX PDMA service function Enabled. 0 = RX PDMA service function Disabled.
[5]	**AUTO_CTS_EN**	**CTSn Auto-Flow Control Enable** 1 = CTSn auto-flow control Enabled. 0 = CTSn auto-flow control. Disabled **Note:** When CTSn auto-flow is enabled, the UART will send data to external device when CTSn input assert (UART will not send data to device until CTSn is asserted).
[4]	**AUTO_RTS_EN**	**RTSn Auto-Flow Control Enable** 1 = RTSn auto-flow control Enabled. 0 = RTSn auto-flow control. Disabled. **Note:** When RTSn auto-flow is enabled, if the number of bytes in the RX-FIFO equals the UART_FCR [RTS_Tri_Lev], the UART will reassert RTSn signal.
[3]	**TX_DIS**	**Transfer Disable Register.** The transceiver is disabled or not (set "1" to disable transceiver) 1 = Transfer Disabled. 0 = Transfer Enabled.
[2]	**RX_DIS**	**Receiver Disable Register.** The receiver is disabled or not (set "1" to disable receiver) 1 = Receiver Disabled. 0 = Receiver Enabled. **Note1:** When used for RS-485 NMM mode, user can set this bit to receive data before detecting address byte. **Note2:** In RS-485 AAD mode, this bit will be setting to "1" automatically. **Note3:** In RS-485 AUD mode and LIN "break + sync +PID" header mode, hardware will control data automatically, so don't fill any value to this bit.
[1]	**TX_RST**	**TX Software Reset** When TX_RST is set, all the bytes in the transmitting FIFO and TX internal state machine are cleared. 1 = Reset the TX internal state machine and pointers. 0 = No effect. **Note:** This bit will be auto cleared and take at least 3 UART engine clock cycles.
[0]	**RX_RST**	**RX Software Reset** When RX_RST is set, all the bytes in the receiving FIFO and RX internal state machine are cleared. 1 = Reset the RX internal state machine and pointers. 0 = No effect. **Note:** This bit will be auto cleared and take at least 3 UART engine clock cycles.

UART Transfer Line Control Register (UARTx_TLCTL)

Register	Offset	R/W	Description	Reset Value
UART_TLCTL **x=0,1**	UARTx_BA+0x08	R/W	UART Transfer Line Control Register. ***Note**	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		RTS_TRI_LEV		Reserved		RFITL	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	BCB	SPE	EPE	PBE	NSB	DATA_LEN	

Bits	Description	
[31:14]	**Reserved**	**Reserved.**
[13:12]	**RTS_TRI_LEV**	**RTSn Trigger Level (For Auto-flow Control Use)** <table><tr><th>RTS_TRI_LEV</th><th>Trigger Level (Bytes)</th></tr><tr><td>00</td><td>01</td></tr><tr><td>01</td><td>04</td></tr><tr><td>10</td><td>08</td></tr><tr><td>11</td><td>14</td></tr></table> **Note:** This field is used for auto RTSn flow control.
[11:10]	**Reserved**	**Reserved.**
[9:8]	**RFITL**	**RX-FIFO Interrupt (INT_RDA) Trigger Level** When the number of bytes in the receiving FIFO is equal to the RFITL then the RDA_IF will be set (if IER [RDA_IEN] is enabled, an interrupt will be generated) <table><tr><th>RFITL</th><th>INTR_RDA Trigger Level (Bytes)</th></tr><tr><td>00</td><td>01</td></tr><tr><td>01</td><td>04</td></tr><tr><td>10</td><td>08</td></tr><tr><td>11</td><td>14</td></tr></table> **Note:** When operating in IrDA mode or RS-485 mode, the RFITL must be set to “0”.

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[7]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[6]</td><td>BCB</td><td>Break Control Bit
When this bit is set to logic “1”, the serial data output (TX) is forced to the Spacing State (logic “0”). This bit acts only on TX pin and has no effect on the transmitter logic.</td></tr>
<tr><td>[5]</td><td>SPE</td><td>Stick Parity Enable
1 = When bits PBE, EPE and SPE are set, the parity bit is transmitted and checked as “0”. When PBE and SPE are set and EPE is cleared, the parity bit is transmitted and checked as “1”. In RS-485 mode, PBE, EPE and SPE can control bit 9, the bit 9 setting are shown as follows.
0 = Stick parity Disabled
<table><tr><th rowspan="3">RS-485
Mode</th><th>SPE</th><th>EPE</th><th>PBE</th><th>Bit9</th></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>[4]</td><td>EPE</td><td>Even Parity Enable
1 = Even number of logic 1’s are transmitted or check the data word and parity bits in receiving mode.
0 = Odd number of logic 1’s are transmitted or check the data word and parity bits in receiving mode.
Note: This bit has effect only when PBE bit (parity bit enable) is set.</td></tr>
<tr><td>[3]</td><td>PBE</td><td>Parity Bit Enable
1 = Parity bit is generated or checked bet“een the “last data”word “it” and ”stop bit” of the serial data.
0 = Parity bit is not generated (transmitting data) or checked (receiving data) during transfer.</td></tr>
<tr><td>[2]</td><td>NSB</td><td>Number of STOP Bit Length
1 = 1.5 “STOP bit” is generated in the transmitted data when 5-bit word length is selected, and 2 STOP bit” is generated when 6, 7 and 8 bits data length is selected.
0 = 1 “ STOP bit” is generated in the transmitted data</td></tr>
<tr><td>[1:0]</td><td>DATA_LEN</td><td>Data Length
<table><tr><th>DATA_LEN</th><th>Character Length</th></tr><tr><td>00</td><td>5 bits</td></tr><tr><td>01</td><td>6 bits</td></tr><tr><td>10</td><td>7 bits</td></tr><tr><td>11</td><td>8 bits</td></tr></table></td></tr>
</table>

UART Interrupt Enable Register (UARTx_IER)

Register	Offset	R/W	Description	Reset Value
UART_IER **x=0,1**	UARTx_BA+0x0C	R/W	UART Interrupt Enable Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							LIN_IE
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ABAUD_IE	WAKE_IE	BUF_ERR_IE	RTO_IE	MODEM_IE	RLS_IE	THRE_IE	RDA_IE

Bits	Description	
[31:9]	**Reserved**	**Reserved.**
[8]	**LIN_IE**	**LIN Interrupt Enable** 1 = INT_LIN Enabled 0 = INT_LIN Masked off
[7]	**ABAUD_IE**	**Auto-Baud Rate Interrupt Enable** 1 = INT_ABAUD Enabled 0 = INT_ABAUD Masked off
[6]	**WAKE_IE**	**Wake-Up Interrupt Enable** 1 = INT_WAKE Enabled 0 = INT_WAKE Masked off
[5]	**BUF_ERR_IE**	**Buffer Error Interrupt Enable** 1 = INT_BUF_ERR Enabled 0 = INT_BUT_ERR Masked off
[4]	**RTO_IE**	**RX Time-Out Interrupt Enable** 1 = INT_TOUT Enabled 0 = INT_TOUT Masked off
[3]	**MODEM_IE**	**Modem Status Interrupt Enable** 1 = INT_MOS Enabled 0 = INT_MOS Masked off
[2]	**RLS_IE**	**Receive Line Status Interrupt Enable** 1 = INT_RLS Enabled 0 = INT_RLS Masked off

Bits	Description	
[1]	**THRE_IE**	**Transmit Holding Register Empty Interrupt Enable** 1 = INT_THRE Enabled 0 = INT_THRE Masked off
[0]	**RDA_IE**	**Receive Data Available Interrupt Enable** 1 = INT_RDA Enabled 0 = INT_RDA Masked off

UART Interrupt Status Control Register (UARTx_ISR)

Register	Offset	R/W	Description	Reset Value
UART_ISR **x=0,1**	UARTx_BA+0x10	R/W	UART Interrupt Status Register.	0x0000_0002

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**LIN_IS**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ABAUD_IS	**WAKE_IS**	**BUF_ERR_IS**	**RTO_IS**	**MODEM_IS**	**RLS_IS**	**THRE_IS**	**RDA_IS**

Bits	Description	
[31:9]	**Reserved**	**Reserved.**
[8]	**LIN_IS**	**LIN Interrupt Status Flag (Read Only)** This bit is set when the LIN TX header transmitted, RX header received or the SIN does not equal SOUT and if IER [LIN_IE] is set then the LIN interrupt will be generated. **Note1:** This bit is read only, but can be cleared by it by writing “1” to UART_TRSR [BIT_ERR_F], UART_TRSR [BIT_TX_F] or UART_TRSR [LIN_RX_F]. **Note2:** This bit is cleared when both the BIT_ERR_F, BIT_TX_F and LIN_RX_F are cleared.
[7]	**ABAUD_IS**	**Auto-Baud Rate Interrupt Status Flag (Read Only)** This bit is set when auto-baud rate detection function finished or the auto-baud rate counter was overflow and if IER [ABAUD_IE] is set then the auto-baud rate interrupt will be generated. **Note1:** This bit is read only, but can be cleared by it by writing “1” to UART_TRSR [ABAUD_TOUT_F] or UART_TRSR [ABAUD_F]. **Note2:** This bit is cleared when both the ABAUD_TOUT_F and ABAUD_F are cleared.
[6]	**WAKE_IS**	**Wake-Up Interrupt Status Flag (Read Only)** This bit is set in Power-down mode, the receiver received data or CTSn signal. If IER [WAKE_IE] is set then the wake-up interrupt will be generated. **Note:** This bit is read only, but can be cleared by it by writing “1” to it.
[5]	**BUF_ERR_IS**	**Buffer Error Interrupt Status Flag (Read Only)** This bit is set when the TX or RX-FIFO overflowed. When BUF_ERR_IS is set, the transfer maybe not correct. If IER [BUF_ER_IEN] is set then the buffer error interrupt will be generated. **Note1:** This bit is read only, but can be cleared by it by writing “1” to UART_FSR [TX_OVER_F] or UART_FSR [RX_OVER_F]. **Note2:** This bit is cleared when both the TX_OVER_F and RX_OVER_F are cleared.

Bits	Description	
[4]	**RTO_IS**	**RX Time-Out Interrupt Status Flag (Read Only)** This bit is set when the RX-FIFO is not empty and no activities occur in the RX-FIFO and the time-out counter equal to TOIC. If IER [Tout_IEN] is set then the tout interrupt will be generated. **Note:** This bit is read only and user can read UART_RBR (RX is in active) to clear it.
[3]	**MODEM_IS**	**MODEM Interrupt Status Flag (Read Only)** This bit is set when the CTSn pin has state change (DCTSF = “1”). If IER [MODEM_IEN] is set then the modem interrupt will be generated. **Note:** This bit is read only, but can be cleared by it by writing “1” to UART_MCSR [DCT_F].
[2]	**RLS_IS**	**Receive Line Interrupt Status Flag (Read Only).** This bit is set when the RX received data has parity error (UART_FSR [PE_F]), framing error (UART_FSR [FE_F]), break error (UART_FSR [BI_F]) or RS-485 detect address byte (UART_TRSR [RS-485_ADDET_F]).If IER [RLS_IEN] is set then the RLS interrupt will be generated. **Note1:** This bit is read only, but can be cleared by it by writing “1” to UART_FSR [BI_F], UART_FSR [FE_F], UART_FSR [PE_F] or UART_TRSR [RS-485_ADDET_F]. **Note2:** This bit is cleared when all the BI_F, FE_F, PE_F and RS-485_ADDET_F are cleared.
[1]	**THRE_IS**	**Transmit Holding Register Empty Interrupt Flag (Read Only).** This bit is set when the last data of TX-FIFO is transferred to Transmitter Shift Register. If IER [THRE_IEN] is set that the THRE interrupt will be generated. **Note:** This bit is read only and it will be cleared when writing data into THR (TX-FIFO not empty).
[0]	**RDA_IS**	**Receive Data Available Interrupt Flag (Read Only).** When the number of bytes in the RX-FIFO equals the RFITL then the RDA_IF will be set. If IER [RDA_IEN] is set then the RDA interrupt will be generated. **Note:** This bit is read only and it will be cleared when the number of unread bytes of RX-FIFO drops below the threshold level (RFITL).

Interrupt Indicator	Interrupt Source	Interrupt Enable	Interrupt Flag	Flag Clear
INT_LIN	LIN Function Interrupt	BUF_ERR_IE	LIN_IS = UART_TRSR [BIR_ERR_F] or UART_TRSR [LIN_RX_F] or UART_TRSR [LIN_TX_F]	Write “1” to UART_TRSR [BIR_ERR_F] or UART_TRSR [LIN_RX_F] or UART_TRSR [LIN_TX_F].
INT_ABAUD	Auto-Baud Rate Interrupt	ABAUD_IE	ABAUD_IS = UART_TRSR [ABAUD_TOUT_F] or UART_TRSR [ABAUD_F]	Write “1” to UART_TRSR [ABAUD_TOUT_F] UART_TRSR [ABAUD_F].
INT_WAKE	Wake-Up Interrupt	WAKE_IE	WAKE_IS	Write “1” to UART_ISR[WAKE_IS]

INT_BUF_ERR	Buffer Error Interrupt	BUF_ERR_IE	BUF_ERR_IS = UART_FSR [RX_OVER_F] or UART_FSR [TX_OVER_F].	Write “1” to UART_FSR [RX_OVER_F] or UART_FSR [TX_OVER_F].
INT_RTO	RX Time-Out Interrupt	RTO_IE	RTO_IS	Read UART_RBR
INT_MODEM	Modem Status Interrupt	MODEM_IE	MODME_IS = UART_MCSR[DCT_F]	Write “1” to UART_MCSR[DCT_F]
INT_RLS	Receive Line Status Interrupt	RLS_IE	RLS_IS = UART_FSR [BI_F] or UART_FSR [FE_F] or UART_FSR [PE_F] or UART_TRSR [RS-485_ADDDET_F]	Write “1” to UART_FSR [BI_F] or UART_FSR [FE_F] or UART_FSR [PE_F] or UART_TRSR [RS-485_ADDDET_F]
INT_THRE	Transmit Holding Register Empty Interrupt	THRE_IE	THRE_IS	Write UART_THR
INT_RDA	Receive Data Available Interrupt	RDA_IE	RDA_IS	Read UART_RBR

Table 5.14-3 UART Interrupt Sources and Flags

UART Transfer Status Register (UARTx_TRSR)

Register	Offset	R/W	Description	Reset Value
UART_TRSR **x=0,1**	UARTx_BA+0x14	R/W	UART Transfer State Status Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**LIN_RX_SYNC_ERR_F**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		**BIT_ERR_F**	**LIN_RX_F**	**LIN_TX_F**	**ABAUD_TOUT_F**	**ABAUD_F**	**RS-485_ ADDET_F**

Bits	Description	
[31:9]	**Reserved**	**Reserved.**
[8]	**LIN_RX_SYNC_ERR_F**	**LIN RX SYNC Error Flag (Read Only)** This bit is set to logic "1" when LIN received incorrect SYNC field. User can choose the header by setting UART_ALT_CTL [LIN_HEAD_SEL] register. If the field includes "break field + sync field" and if the sync data does not equal 0x55, the LIN_RX_F and LIN_RX_SYNC_ERR_F will be set and the wrong sync data will be ignored. The controller will receive next data and put it in FIFO. If the field includes "break field + sync field + PID field" and if the sync data does not equal 0x55, the LIN_RX_F and LIN_RX_SYNC_ERR_F will be set and the wrong sync data will be ignored. The controller will receive next data and put it in FIFO. **Note:** This bit is read only, but can be cleared by writing "1" to LIN_RX_F.
[7:6]	**Reserved**	**Reserved.**
[5]	**BIT_ERR_F**	**Bit Error Detect Status Flag (Read Only)** At TX transfer state, hardware will monitoring the bus state, if the input pin (SIN) state is not equal to the output pin (SOUT) state, BIT_ERR_F will be set. When occur bit error, hardware will generate an interrupt to CPU (INT_LIN). **Note1:** This bit is read only, but it can be cleared by writing "1" to it. **Note2:** This bit is only valid when enabling the bit error detection function (UART_ALT_CTL [BIT_ERR_EN] = "1").
[4]	**LIN_RX_F**	**LIN RX Interrupt Flag (Read Only)** This bit is set to logic "1" when received LIN header field. The header may be "break field" or "break field + sync field" or "break field + sync field + PID field", and it can be choose by setting UART_ALT_CTL [LIN_HEAD_SEL] register. If the field includes "break field", when the receiver received break field then the LIN_RX_F will be set. The controller will receive next data and put it in FIFO. If the field includes "break field + sync field", hardware will wait for the flag LIN_RX_F

Bits	Description	
		in UART_TRSR to check RX received break field and sync field. If the break and sync field is received, hardware will set UART_TRSR [LIN_RX_F] flag, and if the break is received but the sync field does not equal 0x55, then hardware will set UART_TRSR [LIN_RX_F] and UART_TRSR [LIN_RX_SYNC_ERR_F] flag. The break and sync data (equals 0x55 or not) will not be stored in FIFO. If the field includes "break field + sync field + PID field", In this operation mode, hardware will control data automatically. Hardware will ignore any data until received break + sync (0x55) + PID value match the UART_ALT_CTL [ADDR_MATCH] value (break + sync + PID will not be stored in FIFO). When received break + sync (0x55) + PID value match the UART_ALT_CTL [ADDR_MATCH] value, hardware will set UART_TRSR [LIN_RX_F] and the following all data will be accepted and stored in the RX-FIFO until detect next break field. If the receiver received break + wrong sync (not equal 0x55) + PID value, hardware will set UART_TRSR [LIN_RX_F] and UART_TRSR [LIN_RX_SYNC_ERR_F] flag and the receiver will be disabled. If the receiver received break + sync (0x55) + wrong PID value, hardware will set UART_TRSR [LIN_RX_F] flag and the receiver will be disabled. **Note:** This bit is read only, but can be cleared by writing "1" to it.
[3]	**LIN_TX_F**	**LIN TX Interrupt Flag (Read Only)** This bit is set to logic "1" when LIN transmitted header field. The header may be "break field" or "break field + sync field" or "break field + sync field + PID field", it can be choose by setting UART_ALT_CTL[LIN_HEAD_SEL] register. **Note:** This bit is read only, but can be cleared by writing "1" to it.
[2]	**ABAUD_TOUT_F**	**Auto-Baud Rate Time-Out Interrupt (Read Only)** This bit is set to logic "1" in Auto-baud Rate Detect mode and the baud rate counter is overflow. **Note:** This bit is read only, but can be cleared by writing "1" to it.
[1]	**ABAUD_F**	**Auto-Baud Rate Interrupt (Read Only)** This bit is set to logic "1" when auto-baud rate detect function finished. **Note:** This bit is read only, but can be cleared by writing "1" to it.
[0]	**RS-485_** **ADDET_F**	**RS-485 Address Byte Detection Status Flag (Read Only)** This bit is set to logic "1" and set UART_ALT_CTL [RS-485_ADD_EN] whenever in RS-485 mode the receiver detected any address byte character (bit 9 ='1') bit". This bit is reset whenever the CPU writes "1" to this bit. **Note1:** This field is used for RS-485 mode. **Note2:** This bit is read only, but can be cleared by writing "1" to it.

UART FIFO Status Register (UART_FSR)

Register	Offset	R/W	Description	Reset Value
UART_FSR **x=0,1**	UARTx_BA+0x18	R/W	UART FIFO State Status Register.	0x0000_0A02

31	30	29	28	27	26	25	24
Reserved			TX_POINTER_F				
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved			RX_POINTER_F				
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				TE_F	TX_FULL_F	TX_EMPTY_F	TX_OVER_F
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	BI_F	FE_F	PE_F	Reserved	RX_FULL_F	RX_EMPTY_F	RX_OVER_F

Bits	Description	
[31:28]	**Reserved**	**Reserved.**
[28:24]	**TX_POINTER_F**	**TX-FIFO Pointer (Read Only)** This field indicates the TX-FIFO Buffer Pointer. When CPU writes one byte data into UART_THR, TX_POINTER_F increases one. When one byte of TX-FIFO is transferred to Transmitter Shift Register, TX_POINTER_F decreases one.
[23:21]	**Reserved**	**Reserved.**
[20:16]	**RX_POINTER_F**	**RX-FIFO Pointer (Read Only)** This field indicates the RX-FIFO Buffer Pointer. When UART receives one byte from external device, RX_POINTER_F increases one. When one byte of RX-FIFO is read by CPU, RX_POINTER_F decreases one.
[15:12]	**Reserved**	**Reserved.**
[11]	**TE_F**	**Transmitter Empty Status Flag (Read Only)** Bit is set by hardware when TX is inactive. (TX shift register does not have data) Bit is cleared automatically when TX-FIFO is transfer data to TX shift register or TX is empty but the transfer does not finish.
[10]	**TX_FULL_F**	**Transmitter FIFO Full (Read Only)** This bit indicates TX-FIFO full or not. This bit is set when TX_POINTER_F is equal to 16, otherwise is cleared by hardware.
[9]	**TX_EMPTY_F**	**Transmitter FIFO Empty (Read Only)** This bit indicates TX-FIFO empty or not. When the last byte of TX-FIFO has been transferred to Transmitter Shift Register, hardware sets this bit high. It will be cleared when writing data into THR (TX-FIFO not empty).
[8]	**TX_OVER_F**	**TX Overflow Error Interrupt Status Flag (Read Only)** If TX-FIFO (UART_THR) is full, an additional write to UART_THR will cause this bit to

Bits	Description	
		logic "1". **Note:** This bit is read only, but it can be cleared by writing "1" to it.
[7]	**Reserved**	**Reserved.**
[6]	**BI_F**	**Break Status Flag (Read Only)** This bit is set to a logic "1" whenever the received data input(RX) is held in the "spacing state" (logic "0") for longer than a full word transmission time (that is, the total time of "start bit" + data bits + parity + stop bits) and it is reset whenever the CPU writes "1" to this bit. **Note:** This bit is read only, but it can be cleared by writing "1" to it.
[5]	**FE_F**	**Framing Error Status Flag (Read Only)** This bit is set to logic "1" whenever the received character does not have a valid "stop bit" (that is, the stop bit following the last data bit or parity bit is detected as a logic "0"), and it is reset whenever the CPU writes "1" to this bit. **Note:** This bit is read only, but it can be cleared by writing "1" to it.
[4]	**PE_F**	**Parity Error State Status Flag (Read Only)** This bit is set to logic "1" whenever the received character does not have a valid "parity bit", and it is reset whenever the CPU writes "1" to this bit. **Note:** This bit is read only, but it can be cleared by writing "1" to it.
[3]	**Reserved**	**Reserved.**
[2]	**RX_FULL_F**	**Receiver FIFO Full (Read Only)** This bit initiates RX-FIFO full or not. This bit is set when RX_POINTER_F is equal to 16, otherwise is cleared by hardware.
[1]	**RX_EMPTY_F**	**Receiver FIFO Empty (Read Only)** This bit initiate RX-FIFO empty or not. When the last byte of RX-FIFO has been read by CPU, hardware sets this bit high. It will be cleared when UART receives any new data.
[0]	**RX_OVER_F**	**RX Overflow Error Status Flag (Read Only)** This bit is set when RX-FIFO overflow. If the number of bytes of received data is greater than RX-FIFO (UART_RBR) size, 16 bytes of UART0/UART1, this bit will be set. **Note:** This bit is read only, but it can be cleared by writing "1" to it.

UART MODEM Control Register (UARTx_MCSR)

Register	Offset	R/W	Description	Reset Value
UART_MCSR **x=0,1**	UARTx_BA+0x1C	R/W	UART Modem State Status Register.	0x0002_0002

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					DCT_F	CTS_ST	LEV_CTS
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						RTS_ST	LEV_RTS

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:19]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[18]</td><td>DCT_F</td><td>Detect CTSn State Change Status Flag (Read Only)
This bit is set whenever CTSn input has change state, and it will generate Modem interrupt to CPU when UART_IER [Modem_IEN].
Note: This bit is read only, but it can be cleared by writing “1” to it.</td></tr>
<tr><td>[17]</td><td>CTS_ST</td><td>CTSn Pin Status (Read Only)
This bit is the pin status of CTSn.</td></tr>
<tr><td>[16]</td><td>LEV_CTS</td><td>CTSn Trigger Level
This bit can change the CTSn trigger level.
1 = High level triggered
0 = Low level triggered
<table>
<tr><th>Operation Mode</th><th>LEV_CTS</th><th>CTSn
Pin Input</th><th>CTS_ST</th><th>Transmitter
State</th></tr>
<tr><td rowspan="4">CTS Auto-Flow
Control Mode</td><td>0</td><td>0</td><td>0</td><td>STOP</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>ACTIVE</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>ACTIVE</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>STOP</td></tr>
</table></td></tr>
<tr><td>[15:12]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[1]</td><td>RTS_ST</td><td>RTSn Pin State (Read Only)
This bit is the pin status of RTSn.</td></tr>
<tr><td>[0]</td><td>LEV_RTS</td><td>RTSn Trigger Level
This bit can change the RTSn trigger level.</td></tr>
</table>

Bits	Description

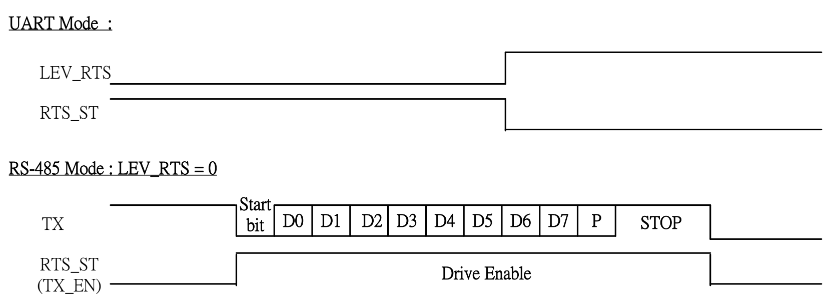
1 = high level triggered

0 = low level triggered

For example, the relation waveform between LEV_RTS and RTSn shown as follows.

Operation Mode	LEV_RTS	RTS_ST (Default Output State)
RS-485 AUD Mode (Note)	0	0
	1	1
RTS Auto-Flow Control Mode (Note)	0	1
	1	0
Normal mode	0	1
	1	0

Note: In RS-485 AUD mode and RTS Auto-flow control mode, hardware will control the output RTS pin automatically, so the table indicates the default value.

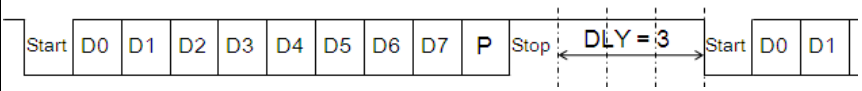


Note: The default setting in UART mode is LEV_RTS = “0” and RTS_ST = “1”.

UART TIME-oUT Register (UARTx_TMCTL)

Register	Offset	R/W	Description	Reset Value
UART_TMCTL **x=0,1**	UARTx_BA+0x20	R/W	UART Time-Out Control State Register.	0x0000_01FF

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DLY							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**TOIC**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TOIC							

Bits	Description	
[31:24]	**Reserved**	**Reserved.**
[23:16]	**DLY**	**TX Delay Time Value** This field is use to program the transfer delay time between the last stop bit leaving the TX-FIFO and the de-assertion of by setting UART_TMCTL [DLY] register.  **Note1:** Fill all "0" to this field indicates to disable this function. **Note2:** The real delay value is DLY. **Note3:** The counting clock is baud rate clock.
[15:9]	**Reserved**	**Reserved.**
[8:0]	**TOIC**	**Time-Out Comparator** The time-out counter resets and starts counting (the counting clock = baud rate) whenever the RX-FIFO receives a new data word. Once the content of time-out counter (TOUT_CNT) is equal to time-out interrupt comparator (TOIC), a receiver time-out interrupt (INT_TOUT) is generated if UART_IER [RTO_IEN]. A new incoming data word or RX-FIFO empty clears INT_TOUT. **Note1:** Fill all "0" to this field indicates to disable this function. **Note2:** The real time-out value is TOIC + 1. **Note3:** The counting clock is baud rate clock. **Note4:** The UART data format is start bit + 8 data bits + parity bit + stop bit, although software can configure this field by any value but it is recommend to filled this field great than 0xA.

UART Baud Rate Divider Register (UARTx_BAUD)

Register	Offset	R/W	Description	Reset Value
UART_BAUD **x=0,1**	UARTx_BA+0x24	R/W	UART Baud Rate Divisor Register	0x0000_0000

31	30	29	28	27	26	25	24
DIV_16_EN	Reserved						
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BRD							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BRD							

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31]</td><td>DIV_16_EN</td><td>Divider 16 Enable
The BRD = Baud Rate Divider, and the baud rate equation is Baud Rate = UART_CLK/ [16 * (BRD + 1)]; The default value of M is 16.
1 = The equation of baud rate is UART_CLK / [16 * (BRD+1)]
0 = The equation of baud rate is UART_CLK / [(BRD+1)]
Note: In IrDA mode, this bit must disable.</td></tr>
<tr><td>[30:16]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[15:0]</td><td>BRD</td><td>Baud Rate Divider
The low byte of the baud rate divider
<table>
<tr><th>DIV_16_EN</th><th>BRD</th><th>Baud Rate Equation</th></tr>
<tr><td>Disable
(Mode 0)</td><td>A</td><td>UART_CLK / (A+1), A must >8</td></tr>
<tr><td>Enable
(Mode 1)</td><td>A</td><td>UART_CLK / [16 * (A+1)]</td></tr>
</table></td></tr>
</table>

UART IrDA Control Register (UARTx_IRCR)

Register	Offset	R/W	Description	Reset Value
UART_IRCR **x=0,1**	UARTx_BA+0x30	R/W	UART IrDA Control Register.	0x0000_0040

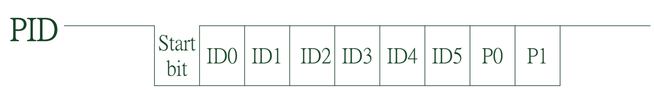
31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**INV_RX**	**INV_TX**	**Reserved**			**TX_SELECT**	**Reserved**

Bits	Description	
[31:7]	**Reserved**	**Reserved.**
[6]	**INV_RX**	**INV_RX** 1 = Inverse RX input signal 0 = No inversion
[5]	**INV_TX**	**INV_TX** 1 = Inverse TX output signal 0 = No inversion
[4:2]	**Reserved**	**Reserved.**
[1]	**TX_SELECT**	**TX_SELECT** 1 = IrDA transmitter Enabled. 0 = IrDA receiver Enabled. **Note:** In IrDA mode, the UART_BAUD [DIV_16_EN) register must be set (the baud equation must be Clock / 16 * (BRD)
[0]	**Reserved**	**Reserved.**

UART ALT Control State Register (UARTx_ALT_CTL)

Register	Offset	R/W	Description	Reset Value
UART_ALT_CSR x=0,1	UARTx_BA+0x34	R/W	UART Alternate Control State Register.	0x0000_0000

31	30	29	28	27	26	25	24
ADDR_PID_MATCH							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				**RS-485_ ADD_EN**	**RS-485_AUD**	**RS-485_AAD**	**RS-485_NMM**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**BIT_ERR_EN**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
LIN_TX_EN	**LIN_RX_EN**	**LIN_HEAD_SEL**		**Reserved**	**LIN_TX_BCNT**		

Bits	Description	
[31:24]	**ADDR_PID_MATCH**	**Address / PID Match Value Register** This field contains the RS-485 address match values in RS-485 Function mode. This field contains the LIN protected identifier field n LIN Function mode, software fills ID0~ID5 (ADDR_PID_MATCH [5:0]), hardware will calculate P0 and P1.  P0 = ID0 xor ID1 xor ID2 xor ID4 P1 = ~(ID1 xor ID3 xor ID4 xor ID5) **Note:** This field is used for RS-485 auto address detection mode or used for LIN protected identifier field (PID).
[23:20]	**Reserved**	**Reserved.**
[19]	**RS-485_ ADD_EN**	**RS-485 Address Detection Enable** This bit is used to enable RS-485 hardware address detection mode. If hardware detects address byte, and then the controller will set UART_TRSR [RS_485_ADDET_F] = “1”. 1 = Address detection mode Enabled. 0 = Address detection mode Disabled. **Note:** This field is used for RS-485 any operation mode.
[18]	**RS-485_AUD**	**RS-485 Auto Direction Mode (RS-485 AUD Mode)** 1 = RS-485 Auto Direction mode (AUD) Enabled. 0 = RS-485 Auto Direction mode (AUD) Disabled. **Note:** It can be active in RS-485_AAD or RS-485_NMM operation mode.

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[17]</td><td>RS-485_AAD</td><td>RS-485 Auto Address Detection Operation Mode (RS-485 AAD Mode)
1 = RS-485 Auto Address Detection Operation mode (AAD) Enabled.
0 = RS-485 Auto Address Detection Operation mode (AAD) Disabled.
Note: It can’t be active in RS-485_NMM Operation mode.</td></tr>
<tr><td>[16]</td><td>RS-485_NMM</td><td>RS-485 Normal Multi-Drop Operation Mode (RS-485 NMM Mode)
1 = RS-485 Normal Multi-drop Operation mode (NMM) Enabled.
0 = RS-485 Normal Multi-drop Operation mode (NMM) Disabled.
Note: It can’t be active in RS-485_AAD Operation mode.</td></tr>
<tr><td>[15:9]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[8]</td><td>Bit_ERR_EN</td><td>Bit Error Detect Enable
1 = Bit error detection Enabled.
0 = Bit error detection function Disabled.
Note: In LIN function mode, when bit error occurs, hardware will generate an interrupt to CPU (INT_LIN).</td></tr>
<tr><td>[7]</td><td>LIN_TX_EN</td><td>LIN TX Header Trigger Enable
1 = LIN TX Header Trigger Enabled.
0 = LIN TX Header Trigger Disabled.
Note1: When TX header field (break field or break and sync field or break, sync and PID field) transfer operation finished, this bit will be cleared automatically and generate a interrupt to CPU (INT_LIN).
Note2: If user wants to receive transmit data, it recommended to enable LIN_RX_EN bit.</td></tr>
<tr><td>[6]</td><td>LIN_RX_EN</td><td>LIN RX Enable
When LIN RX mode enabled and received a break field or sync field or PID field (Select by LIN_Header_SEL), the controller will generator a interrupt to CPU (INT_LIN)
1 = LIN RX mode Enabled.
0 = LIN RX mode Disabled.</td></tr>
<tr><td>[5:4]</td><td>LIN_HEAD_SEL</td><td>LIN Header Selection
<table><tr><th>LIN_HEAD_SEL</th><th>Description</th></tr><tr><td>00</td><td>The LIN header includes “break field”.</td></tr><tr><td>01</td><td>The LIN header includes “break field + sync field”.</td></tr><tr><td>10</td><td>The LIN header includes “break field + sync field + PID field”.</td></tr><tr><td>11</td><td>Reserved.</td></tr></table></td></tr>
<tr><td>[3]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[2:0]</td><td>LIN_TX_BCNT</td><td>LIN TX Break Field Count Register
The field contains 3-bit LIN TX break field count.
Note: The break field length is LIN_TX_BCNT + 8.</td></tr>
</table>

UART Function Select Register (UARTx_FUN_SEL)

Register	Offset	R/W	Description	Reset Value
UART_FUN_SEL **x=0,1**	UARTx_BA+0x38	R/W	UART Function Select Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**FUN_SEL**	

Bits	Description	
[31:2]	**Reserved**	**Reserved.**
[1:0]	**FUN_SEL**	**Function Select Enable** <table><tr><th>FUN_SEL</th><th>Description</th></tr><tr><td>00</td><td>UART function mode.</td></tr><tr><td>01</td><td>LIN function mode.</td></tr><tr><td>10</td><td>IrDA Function.</td></tr><tr><td>11</td><td>RS-485 Function.</td></tr></table>

5.15 Smart Card Host Interface (SC)

5.15.1 Overview

The Smart Card Interface controller (SC controller) is based on ISO/IEC 7816-3 standard and fully compliant with PC/SC Specifications. It also provides status of card insertion/removal.

5.15.2 Features

- ISO-7816-3 T = 0, T = 1 compliant
- EMV2000 compliant
- Supports up to three ISO-7816-3 ports
- Separates receive / transmit 4 byte entry buffer for data payloads
- Programmable transmission clock frequency
- Programmable receiver buffer trigger level
- Programmable guard time selection (11 ETU ~ 266 ETU)
- A 24-bit and two 8-bit counters for Answer to Reset (ATR) and waiting times processing
- Supports auto inverse convention function
- Supports stop clock level and clock stop (clock keep) function
- Supports transmitter and receiver error retry and error number limitation function
- Supports hardware activation sequence process
- Supports hardware warm reset sequence process
- Supports hardware deactivation sequence process
- Supports hardware auto deactivation sequence when detected the card removal.
- Support UART mode
  - Half duplex, asynchronous communications
  - Separate receiving / transmitting 4 bytes entry FIFO for data payloads
  - Support programmable baud rate generator for each channel
  - Support programmable receiver buffer trigger level
  - Programmable transmitting data delay time between the last stop bit leaving the TX-FIFO and the de-assertion by setting SCx_EGTR [EGT] register
  - Programmable even, odd or no parity bit generation and detection
  - Programmable stop bit, 1 or 2 stop bit generation

5.15.3 Block Diagram

The SC clock control and block diagram are shown as follows. The SC controller is completely asynchronous design with to clock domains, PCLK and engine clock, note that the PCLK should be higher than or equal to the frequency of engine clock.

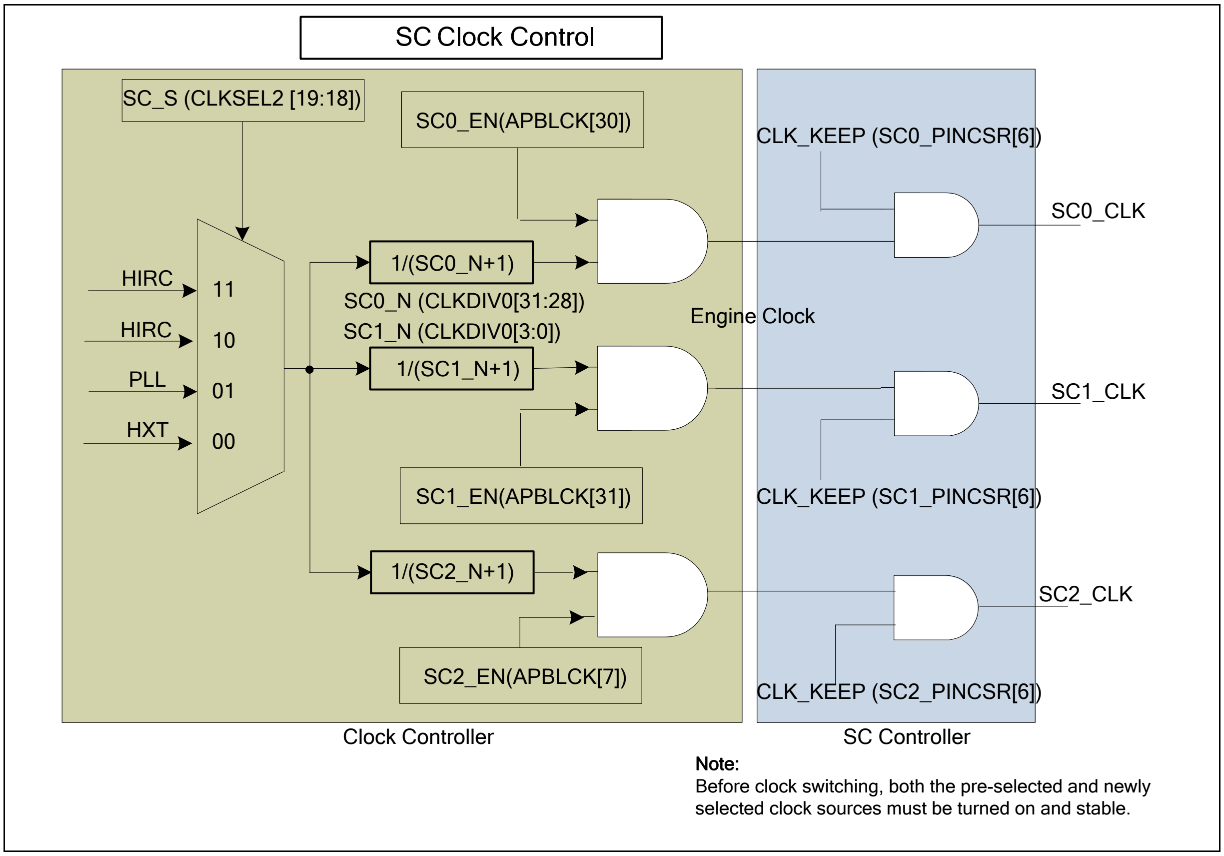


Figure 5.15-1 SC Clock Control Diagram (4-bit Pre-Scale Counter in Clock Controller)

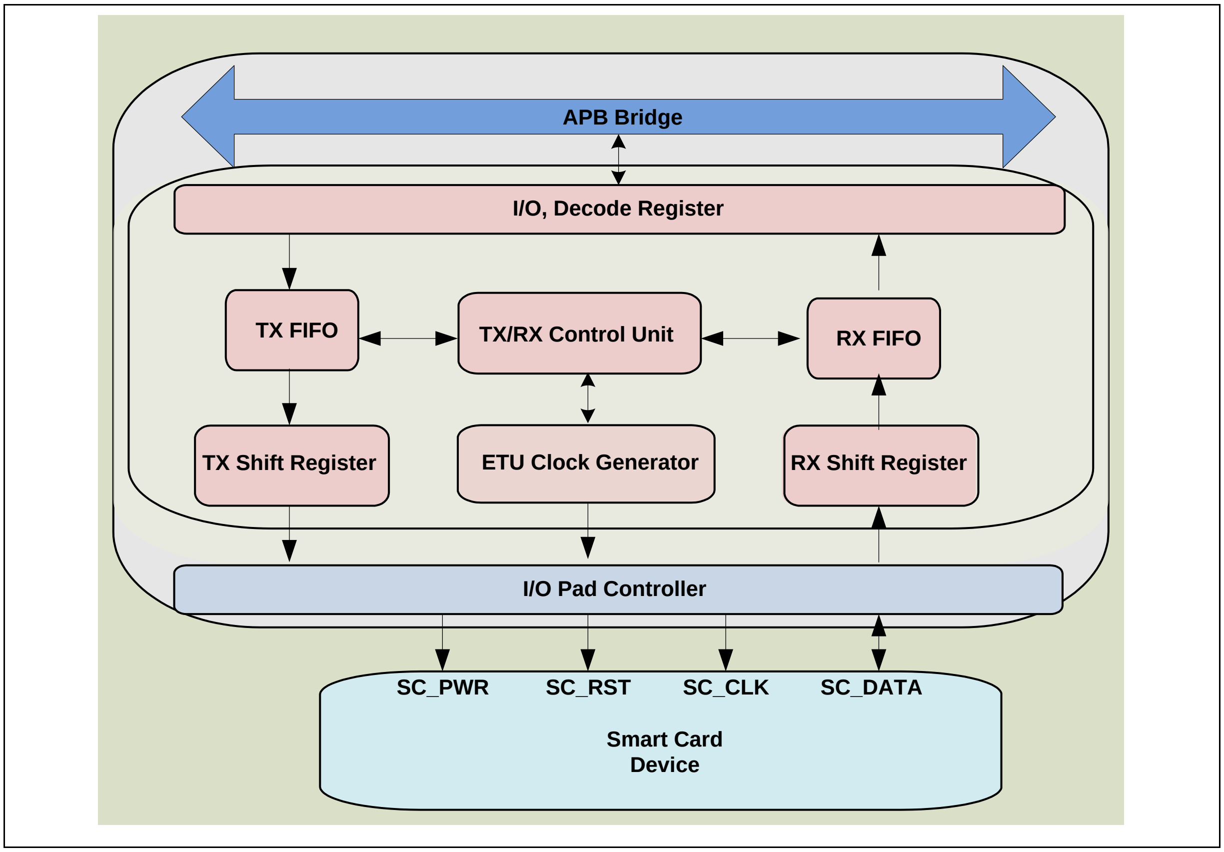


Figure 5.15-2 SC Controller Block Diagram

5.15.4 Functional Description

Basically, the smart card interface acts as a half-duplex asynchronous communication port and its data format is composed of ten consecutive bits which is shown as follows.

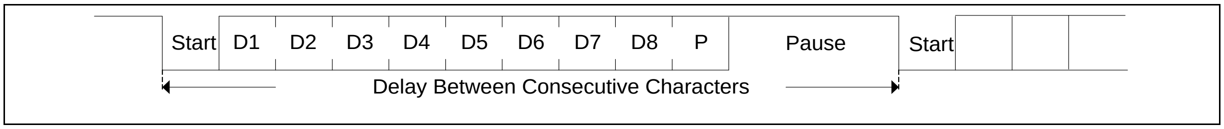


Figure 5.15-3 SC Data Character

5.15.4.1 Activation, Warm Reset and Deactivation Sequence

Activation

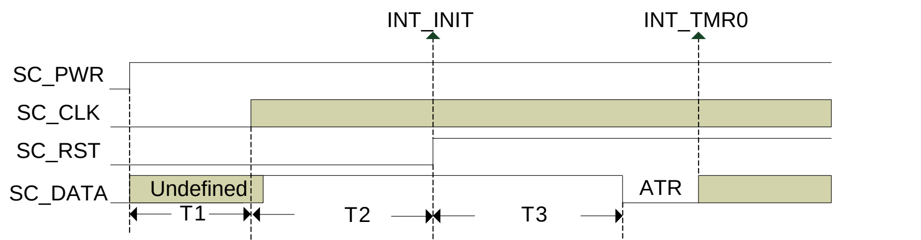
The Smart Card Interface controller supports hardware activation, warm reset and deactivation sequence. The activation sequence is shown as follows.

- Set SC_RST to low
- Set SC_PWR at high level and SC_DATA in reception mode.
- Enable SC_CLK clock
- De-assert SC_RST to high

The activation sequence can be controlled by software or hardware. If software wants to control it, software can control SC_PINCSR and SC_TMRx register to process the activation sequence or setting SC_ALTCTL [ACT_EN] register, and then the interface will perform the hardware activation sequence.

Following is activation control sequence in hardware activation mode:

- Set activation timing by setting SC_ALTCTL [INIT_SEL].
- TMR0 can be selected when SC_CTL [TMR_SEL] is 01, 10 or 11.
- Set operation mode SC_TMR0 [MODE] to 0011 and give an Answer to Reset value by setting SC_TMR0 [CNT] register.
- When hardware de-asserts SC_RST to high, hardware will generator an interrupt INT_INIT to CPU at the same time (SC_IER[INIT_IE] = “1”)
- If the TMR0 decreases the counter to “1” (start from SC_RST) and the card does not response ATR before that time, hardware will generate interrupt INT_TMR0 to CPU.



Time	Comment
T1	SC_PWR to SC_CLK Start
T2	SC_CLK Start to SC_RST Assert
T3	SC_CLK Start to ART Appear

INIT_SEL	T1	T2
00	85	489
01	133	537
10	165	569
11	165	42060

Unit : SC Clock

Note : These value are measured by chip IO pin, the real value will depended on system design

Figure 5.15-4 SC Activation Sequence

Warm Reset

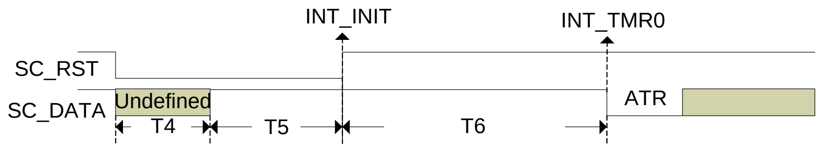
The warm reset sequence is shown as follows.

The warm reset sequence can be controlled by software or hardware. If software wants to control it, software can control SC_PINCSR and SC_TMRx register to process the warm reset sequence or set SC_ALTCTL [WARST_EN] register, and then the interface will perform the hardware warm reset sequence.

Following is warm reset control sequence in hardware warm reset mode

- Set warm reset timing by setting SC_ALTCTL [INIT_SEL].
- Select TMR0 by setting SC_CTL [TMR_SEL] register (TMR_SEL can be 01, 10, or 11).
- Set operation mode SC_TMR0 [MODE]) to 011 and give an Answer to Reset value by setting SC_TMR0 [CNT] register.
- Set TMR0_SEN and WARST_EN to start counting by. SC_ALTCTL register.
- When hardware de-asserts SM_RST to high, hardware will generate an interrupt INT_INIT to CPU at the same time (SC_IER[INIT_IE] = “1”)

- If the TMR0 decrease the counter to “1” (start from SC_RST) and the card does not response ATR before that time, hardware will generate interrupt INT_TMR0 to CPU.



Time	Comment
T4	SC_RST to SC_DATA Reception Mode
T5	SC_DATA Reception Mode to SC_RST Assert
T6	SC_CLK Start to ATR Appear

INIT_SEL	T4	T5
00	81	483
01	129	531
10	161	563
11	161	42106

Unit:SC Clock

Note: This value is measured by chip IO pin, the real value will depended on system design

Figure 5.15-5 SC Warm Reset Sequence

Deactivation

The deactivation sequence is shown as follows:

- Set SC_RST to low
- Stop SC_CLK
- Set SC_DATA to state low
- Deactivated SC_PWR

The deactivation sequence can be controlled by software or hardware. If software wants to control it, software can control SC_PINCSR and SC_TMR0 register to process the deactivation sequence or set SC_ALTCTL [DACT_EN] register, and then the interface will perform the hardware deactivation sequence.

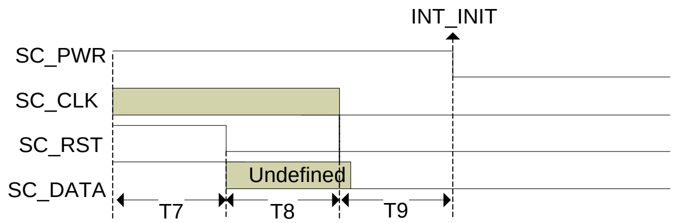
The SC controller also supports auto deactivation sequence when the card removal detection is set (SC_PINCSR [A)AC_CDEN)).

Following is deactivation control sequence in hardware deactivation mode:

Set deactivation timing by setting SC_ALTCTL [INIT_SEL].

Set DACT_EN to start counting by. SC_ALTCTL register.

When hardware de-asserts SC_PWR to low, controller will generate an interrupt INT_INIT to CPU at the same time (SC_IER[INIT_IE] = “1”).



Time	Comment
T7	Deactivation Trigger to SC_RST Low
T8	SMC_RST Low to Stop SC_CLK
T9	Stop SC_CLK to Stop SC_PWR

INIT_SEL	T7	T8	T9
00	97	83	87
01	145	131	135
10	177	163	167
11	177	163	167

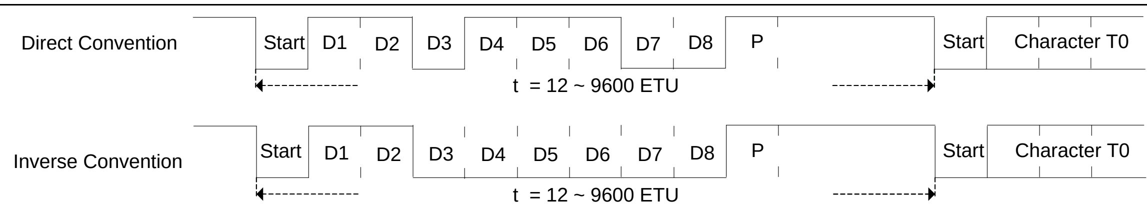
Unit: SC Clock

Figure 5.15-6 SC Deactivation Sequence

5.15.4.2 Initial Character TS

According to 7816-3, the initial character TS of answer to request (ATR) has two possible patterns (as shown in the following figure). If the TS pattern is 0_1100_0000_1, it is inverse convention. When decoded by inverse convention, the conveyed byte is equal to '3F'. If the TS pattern is 0_1101_1100_1, it is direct convention. When decoded by direct convention, the conveyed byte is equal to '3B'.Software can set SC_CTL [AUTO_CON_EN] and then the operating convention will be decided by hardware. Software can also set the SC_CTL [CON_SEL] register (set to 00 or 11) to change the operating convention after SC received TS of answer to request (ATR).

If software enables auto convention function by setting SC_CTL [AUTO_CON_EN] register, the setting step must be done before Answer to Reset state and the first data must be 0x3B or 0x3F. After hardware received first data and stored it at buffer, the hardware will decided the convention and change the SC_CTL [CON_SEL] register automatically. If the first data is neither 0x3B nor 0x3F, the hardware will generator an interrupt INT_ACON_ERR (if SC_IER [ACON_ERR_IE] = "1") to CPU.



Direct Convention	0_1101_1100_1 (0x3B)
Inverse Convention	0_1100_0000_1 (0x3F)

Figure 5.15-7 Initial Character TS

5.15.4.3 Error Signal and Character Repetition

According to 7816-3 T=0 mode description, as shown in following, if the receiver receives a wrong parity bit, it will pull the SC_DATA to low one to two bit period to inform the transmitter parity error. Then the transmitter will retransmit the character. The SC interface controller supports hardware error detection function in receiver and supports hardware re-transmit function in transmitter. Software can enable re-transmit function by setting SC_CTL [TX_ERETRY_EN]. Software can also define the retry (re-transmit) number limitation in SC_CTL [TX_ERETRY] register. If the re-transmit number is greater than SC_CTL [TX_ERETRY], transmitter will transfer the next new data to device and generate an interrupt INT_TERR (if SC_IER [TERR_IE) = "1") to CPU. If the number of received errors by receiver is greater than SC_CTL [RX_ERETRY], receiver will receive this error data to buffer and generate an interrupt INT_TERR(if SC_IER [TERR_IE] = "1") to CPU.

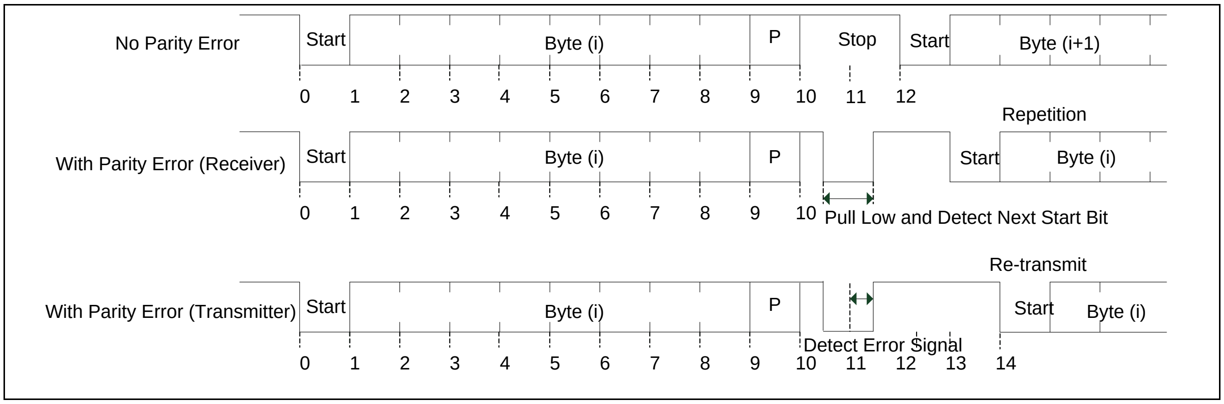


Figure 5.15-8 SC Error Signal

5.15.4.4 Internal Time-Out Counter

The smart card interface includes a 24-bit time-out counter and two 8-bit time-out counters. These counters help the controller in processing different real-time interval (ATR, WWT, BWT, etc.). Each counter can be set to start counting once the trigger enable bit has been written or a START bit has been detected.

The following is the programming flow:

- Enable / Disable counter by setting SC_CTL[TMR_SEL]
- Select operation mode ([MODE]) and give a count value ([CNT]) by setting SC_TMRx register.
- Set TMRx_SEN to start counting.

5.15.4.5 UART Mode

When the SCx_UACTL [UA_MODE_EN] bit is set, the Smart Card Interface controller can also be used as UART function without flow control. In UART mode, SC data (SCx_DATA) pin will be used as UART RXD and SC clock (SCx_CLK) pin will be used as UART TXD. The following is the programming example for UART mode.

Programming example:

1. Enter UART mode by setting the SCx_UACTL[UA_MODE_EN] bit .
2. Do software reset by setting the SCx_ALTCTL[RX_RST] and SCx_ALTCTL[TX_RST] bit to ensure that all state machines return to idle state.
3. Fill "0" to the SCx_CTL [CON_SEL] and SCx_CTL [AUTO_CON_EN] field. (in UART mode, those fields must be "0")
4. Select the UART baud rate by setting SCx_ETUCR [ETU_RDIV] fields.
   - Baud rate = f / (ETU_RDIV+1), where f is SMC engine clock frequence (SCx_CLK), the effective ETU_RDIV is between 0x04 to 0xFFF. The value that less than 0x04 will be regarded as 0x04.

- For example, if user wants to set the baud rate as 115200, and SC clock frequency is 12 MHz, ETU_RDIV should be set to 0x67 and the error rate is around 0.16%.

5. Select the data format including data length (by setting SCx_UA_CTL [DATA_LEN]), parity format (by setting SCx_UA_CTL [OPB] and SCx_UA_CTL [PBDIS] bit) and stop bit length (by setting SCx_CTL [SLEN] or SCx_EGTR [EGT]).
6. Select the receiver buffer trigger level by setting the SCx_CTL [RX_FTRI_LEV] field and select the receiver buffer time-out value by setting the SCx_RFTMR [RFTMR] field.
7. Write the SCx_THR (TX) register or read the SC_RBR (RX) register to perform UART function.

5.15.5 Register and Memory Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
SC Base Address: **SC0_BA = 0x4019_0000** **SC1_BA = 0x401B_0000** **SC2_BA = 0x401C_0000**				
SCx_RBR **x=0,1,2**	SCx_BA+0x00	R	SC Receiving Buffer Register (Read Only).	Undefined
SCx_THR **x=0,1,2**	SCx_BA+0x00	W	SC Transmit Holding Register.	Undefined
SCx_CTL **x=0,1,2**	SCx_BA+0x04	R/W	SC Control Register.	0x0000_0000
SCx_ALTCTL **x=0,1,2**	SCx_BA+0x08	R/W	SC Alternate Control Register.	0x0000_0000
SCx_EGTR **x=0,1,2**	SCx_BA+0x0C	R/W	SC Extend Guard Time Register.	0x0000_0000
SCx_RFTMR **x=0,1,2**	SCx_BA+0x10	R/W	SC Receive Buffer Time-Out Register.	0x0000_0000
SCx_ETUCR **x=0,1,2**	SCx_BA+0x14	R/W	SC ETU Control Register.	0x0000_0173
SCx_IER **x=0,1,2**	SCx_BA+0x18	R/W	SC Interrupt Enable Register.	0x0000_0000
SCx_ISR **x=0,1,2**	SCx_BA+0x1C	R/W	SC Interrupt Status Register.	0x0000_0002
SCx_TRSR **x=0,1,2**	SCx_BA+0x20	R/W	SC Transfer Status Register.	0x0000_0202
SCx_PINCSR **x=0,1,2**	SCx_BA+0x24	R/W	SC Pin Control State Register.	0x0000_00x0
SCx_TMR0 **x=0,1,2**	SCx_BA+0x28	R/W	SC Internal Timer Control Register 0.	0x0000_0000
SCx_TMR1 **x=0,1,2**	SCx_BA+0x2C	R/W	SC Internal Timer Control Register 1.	0x0000_0000
SCx_TMR2 **x=0,1,2**	SCx_BA+0x30	R/W	SC Internal Timer Control Register 2.	0x0000_0000
SCx_UACTL **x=0,1,2**	SCx_BA+0x34	R/W	SC UART Mode Control Register.	0x0000_0000

SCx_TDRA **x=0,1,2**	SCx_BA+0x38	R	SC Timer Current Data Register A.	0x0000_07FF
SCx_TDRB **x=0,1,2**	SCx_BA+0x3C	R	SC Timer Current Data Register B.	0x0000_7F7F

Note: The x of the SCx_REG represents the SC channel.

5.15.6 Register Description

SC Receiving Buffer Register (SCx_RBR)

Register	Offset	R/W	Description	Reset Value
SC_RBR **x=0,1,2**	SCx_BA+0x00	R	SC Receiving Buffer Register (Read Only).	Undefined

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RBR							

Bits	Description	
[31:8]	**Reserved**	**Reserved.**
[7:0]	**RBR**	**Receive Buffer Register** By reading this register, the SC will return an 8-bit received data.

SC Transmit Holding Register (SCx_THR)

Register	Offset	R/W	Description	Reset Value
SC_THR **x=0,1,2**	SCx_BA+0x00	W	SC Transmit Holding Register.	Undefined

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
THR							

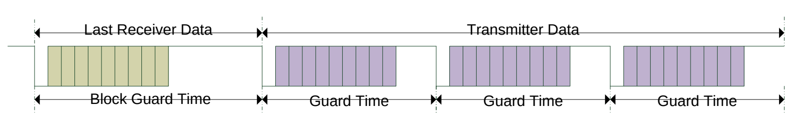
Bits	Description	
[31:8]	**Reserved**	**Reserved.**
[7:0]	**THR**	**Transmit Holding Register** By writing to this register, the SC will send out an 8-bit data. **Note:** If SC_CTL [SC_CEN] not enabled, this register can not be programmed.

SC Control Register (SCx_CTL)

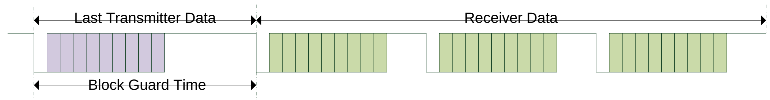
Register	Offset	R/W	Description	Reset Value
SC_CTL **x=0,1,2**	SCx_BA+0x04	R/W	SC Control Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved						**CD_DEB_SEL**	
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TX_ERETRY_EN	**TX_ERETRY**			**RX_ERETRY_EN**	**RX_ERETRY**		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
SLEN	**TMR_SEL**		**BGT**				
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RX_FTRI_LEV		**CON_SEL**		**AUTO_CON_EN**	**DIS_TX**	**DIS_RX**	**SC_CEN**

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[30:26]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[25:24]</td><td>CD_DEB_SEL</td><td>Card Detect De-Bounce Select Register
This field indicates the card detect de-bounce selection.
<table>
<tr><th>CD_DEB_SEL</th><th>Description</th></tr>
<tr><td>00</td><td>De-bounce sample card insert once per 384 (128 * 3) engine clocks and de-bounce sample card removal once per 128 engine clocks.</td></tr>
<tr><td>01</td><td>De-bounce sample card insert once per 192 (64 * 3) engine clocks and de-bounce sample card removal once per 64 engine clocks.</td></tr>
<tr><td>10</td><td>De-bounce sample card insert once per 96 (32 * 3) engine clocks and de-bounce sample card removal once per 32 engine clocks.</td></tr>
<tr><td>11</td><td>De-bounce sample card insert once per 48 (16 * 3) engine clocks and de-bounce sample card removal once per 16 engine clocks.</td></tr>
</table></td></tr>
<tr><td>[23]</td><td>TX_ERETRY_EN</td><td>TX Error Retry Enable Register
This bit enables transmitter retry function when parity error has occurred.
1 = TX error retry function Enabled.
0 = TX error retry function Disabled.
Note: User must fill TX_ERETRY value before enabling this bit.</td></tr>
<tr><td>[22:20]</td><td>TX_ERETRY</td><td>TX Error Retry Register
This field indicates the maximum number of transmitter retries that are allowed when parity error has occurred.
Note1: The real retry number is TX_ERETRY + 1, so 8 is the maximum retry number.
Note2: This field can not be changed when TX_ERETRY_EN enabled. The change flow</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>is to disable TX_ETRTRY_EN first and then fill new retry value.</td></tr>
<tr><td>[19]</td><td>RX_ERETRY_EN</td><td>RX Error Retry Enable Register
This bit enables receiver retry function when parity error has occurred.
1 = RX error retry function Enabled
0 = RX error retry function Disabled.
Note: User must fill RX_ERETRY value before enabling this bit.</td></tr>
<tr><td>[18:16]</td><td>RX_ERETRY</td><td>RX Error Retry Register
This field indicates the maximum number of receiver retries that are allowed when parity error has occurred.
Note1: The real maximum retry number is RX_ERETRY + 1, so 8 is the maximum retry number.
Note2: This field can not be changed when RX_ERETRY_EN enabled. The change flow is to disable RX_ETRTRY_EN first and then fill new retry value.</td></tr>
<tr><td>[15]</td><td>SLEN</td><td>Stop Bit Length
This field indicates the length of stop bit.
1 = The stop bit length is 1 ETU.
0 = The stop bit length is 2 ETU.
Note: The default stop bit length is 2.</td></tr>
<tr><td>[14:13]</td><td>TMR_SEL</td><td>Timer Selection
<table>
<tr><th>TMR_SEL</th><th>Description</th></tr>
<tr><td>00</td><td>Disable all internal timer function.</td></tr>
<tr><td>01</td><td>Enable internal 24 bit timer. Software can configure it by setting SC_TMR0 [23:0]. SC_TMR1 and SC_TMR2 will be ignored in this mode.</td></tr>
<tr><td>10</td><td>Enable internal 24-bit timer and 8-bit internal timer. Software can configure the 24-bit timer by setting SC_TMR0 [23:0] and configure the 8-bit timer by setting SC_TMR1 [7:0]. SC_TMR2 will be ignored in this mode.</td></tr>
<tr><td>11</td><td>Enable internal 24 bit timer and two 8-bit timers. Software can configure them by setting SC_TMR0 [23:0], SC_TMR1 [7:0] and SC_TMR2 [7:0].</td></tr>
</table></td></tr>
<tr><td>[12:8]</td><td>BGT</td><td>Block Guard Time (BGT)
This field indicates the counter for block guard time. According to ISO7816-3, in T=0 mode, software must fill 15 (real block guard time = 16) to this field and in T=1 mode software must fill 21 (real block guard time = 22) to it.
In TX mode, hardware will auto hold off first character until BGT has elapsed regardless of the TX data.


Note1 : Hardware will control the transmit block guard time by SC_CTL [BGT] register setting.
Note2 : Hardware will control the transmit guard time by SC_EGTR [EGT] register setting.
In RX mode, software can enable SC_ALTCTL [RX_BGT_EN] to detect the first coming character timing. If the incoming data timing less than BGT, an interrupt will be generated.</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>

Note : If the incoming data timing less than SC_CTL [BGT], an interrupt will be generated (SC_ALTCR [RX_BGT_EN] enable)

Note: The real block guard time is BGT + 1.</td></tr>
<tr><td>[7:6]</td><td>RX_FTRI_LEV</td><td>RX Buffer Trigger Level
When the number of bytes in the receiving buffer equals the RX_FTRI_LEV, the RDA_IF will be set (if IER [RDA_IEN] is enabled, an interrupt will be generated).
<table>
<tr><th>RX_FTRI_LEV</th><th>INTR_RDA Trigger Level (Bytes)</th></tr>
<tr><td>00</td><td>01</td></tr>
<tr><td>01</td><td>02</td></tr>
<tr><td>10</td><td>03</td></tr>
<tr><td>11</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>[5:4]</td><td>CON_SEL</td><td>Convention Selection
<table>
<tr><th>CON_SEL</th><th>INTR_RDA Trigger Level (Bytes)</th></tr>
<tr><td>00</td><td>Direct convention.</td></tr>
<tr><td>01</td><td>Reserved.</td></tr>
<tr><td>10</td><td>Reserved.</td></tr>
<tr><td>11</td><td>Inverse convention.</td></tr>
</table>
Note: If AUTO_CON_EN is enabled, this field must be ignored.</td></tr>
<tr><td>[3]</td><td>AUTO_CON_EN</td><td>Auto Convention Enable
0 = Auto-convention Disabled.
1 = Auto-convention Enabled. When hardware receives TS in answer to reset state and the TS is direct convention, CON_SEL will be set to 00 automatically, otherwise if the TS is inverse convention, CON_SEL will be set to 11.
If software enables auto convention function, the setting step must be done before Answer to Reset state and the first data must be 0x3B or 0x3F. After hardware received first data and stored it at buffer, hardware will decided the convention and change the SC_CTL[CON_SEL] register automatically. If the first data is not 0x3B or 0x3F, hardware will generate an interrupt INT_ACON_ERR(if SC_IER [ACON_ERR_IE = “1”] to CPU.</td></tr>
<tr><td>[2]</td><td>DIS_TX</td><td>TX Transition Disable
1 = Transceiver Disabled.
0 = Transceiver Enabled.</td></tr>
<tr><td>[1]</td><td>DIS_RX</td><td>RX Transition Disable
1 = Receiver Disabled.
0 = Receiver Enabled.</td></tr>
<tr><td>[0]</td><td>SC_CEN</td><td>SC Engine Enable
Set this bit to “1” to enable SC operation. If this bit is cleared, SC will force all transition to IDLE state.</td></tr>
</table>

SC Alternate Control Register (SCx_ALTCTL)

Register	Offset	R/W	Description	Reset Value
SC_ALTCTL **x=0,1,2**	SCx_BA+0x08	R/W	SC Alternate Control Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TMR2_ATV	TMR1_ATV	TMR0_ATV	RX_BGT_EN	Reserved		INIT_SEL	
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TMR2_SEN	TMR1_SEN	TMR0_SEN	WARST_EN	ACT_EN	DACT_EN	RX_RST	TX_RST

Bits	Description	
[31:16]	**Reserved**	**Reserved.**
[15]	**TMR2_ATV**	**Internal Timer2 Active State (Read Only)** This bit indicates the timer counter status of timer2. 1 = Timer2 is active. 0 = Timer2 is not active.
[14]	**TMR1_ATV**	**Internal Timer1 Active State (Read Only)** This bit indicates the timer counter status of timer1. 1 = Timer1 is active 0 = Timer1 is not active.
[13]	**TMR0_ATV**	**Internal Timer0 Active State (Read Only)** This bit indicates the timer counter status of timer0. 1 = Timer0 is active. 0 = Timer0 is not active.
[12]	**RX_BGT_EN**	**Receiver Block Guard Time Function Enable** 1 = Receiver block guard time function Enabled. 0 = Receiver block guard time function Disabled.
[11:10]	**Reserved**	**Reserved.**
[9:8]	**INIT_SEL**	**Initial Timing Selection** This field indicates the timing of hardware initial state (activation or warm-reset or deactivation). Unit: SC clock Activation: Refer to the activation figure.

<table>
<tr><th>Bits</th><th>Description</th><th></th></tr>
<tr><td></td><td></td><td>

INIT_SEL	T1	T2
00	85	489
01	133	537
10	165	569
11	165	42060

Warm-reset: Refer to the warm-reset figure

INIT_SEL	T4	T5
00	81	483
01	129	531
10	161	563
11	161	42106

Deactivation: Refer to the deactivation figure

INTI_SEL	T7	T8	T9
00	97	83	87
01	145	131	135
10	177	163	167
11	177	163	167

</td></tr>
<tr><td>[7]</td><td>TMR2_SEN</td><td>

Internal Timer2 Start Enable

This bit enables Timer2 to start counting. Software can fill “0” to stop it and set “1” to reload and count.

1 = Starts counting.

0 = Stops counting.

Note1: This field is used for internal 8-bit timer when SC_CTL [TMR_SEL] == 11. Don’t filled TMR2_SEN when SC_CTL [TMR_SEL] == 00 or 01 or 10.

Note2: If the operation mode is not in auto-reload mode (SC_TMR2 [26] = “0”), this bit will be auto-cleared by hardware.

Note3: This field will be cleared by TX_RST and RX_RST. So don’t fill this bit, TX_RST, and RX_RST at the same time.

Note4: If SC_CTL [SC_CEN] is not enabled, this filed can not be programmed.

</td></tr>
<tr><td>[6]</td><td>TMR1_SEN</td><td>

Internal Timer1 Start Enable

This bit enables Timer “1” to start counting. Software can fill 0 to stop it and set “1” to reload and count.

1 = Starts counting.

0 = Stops counting.

Note1: This field is used for internal 8-bit timer when SC_CTL [TMR_SEL] = 01 or 10. Don’t filled TMR1_SEN when SC_CTL [TMR_SEL] = 00 or 11.

Note2: If the operation mode is not in auto-reload mode (SC_TMR1 [26] = “0”), this bit will be auto-cleared by hardware.

Note3: This field will be cleared by TX_RST and RX_RST, so don’t fill this bit, TX_RST,

</td></tr>
</table>

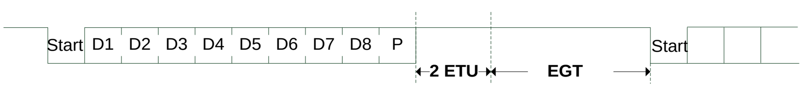
Bits	Description	
		and RX_RST at the same time. **Note4:** If SC_CTL [SC_CEN] is not enabled, this filed can not be programmed.
[5]	**TMR0_SEN**	**Internal Timer0 Start Enable** This bit enables Timer0 to start counting. Software can fill "0" to stop it and set "1" to reload and count. 1 = Starts counting. 0 = Stops counting. **Note1:** This field is used for internal 24 bit timer when SC_CTL [TMR_SEL] = 01. **Note2:** If the operation mode is not in auto-reload mode (SC_TMR0 [26] = "0"), this bit will be auto-cleared by hardware. **Note3:** This field will be cleared by TX_RST and RX_RST. So don't fill this bit, TX_RST and RX_RST at the same time. **Note4:** If SC_CTL [SC_CEN] is not enabled, this filed can not be programmed.
[4]	**WARST_EN**	**Warm Reset Sequence Generator Enable** This bit enables SC controller to initiate the card by warm reset sequence 1 = Warm reset sequence generator Enabled. 0 = No effect. **Note1:** When the warm reset sequence completed, this bit will be cleared automatically and the SC_ISR [INIT_IS] will be set to "1". **Note2:** This field will be cleared by TX_RST and RX_RST, so don't fill this bit, TX_RST, and RX_RST at the same time. **Note3:** If SC_CTL [SC_CEN] is not enabled, this filed can not be programmed.
[3]	**ACT_EN**	**Activation Sequence Generator Enable** This bit enables SC controller to initiate the card by activation sequence 1 = Activation sequence generator Enabled. 0 = No effect. **Note1:** When the activation sequence completed, this bit will be cleared automatically and the SC_IS [INIT_IS] will be set to "1". **Note2:** This field will be cleared by TX_RST and RX_RST, so don't fill this bit, TX_RST, and RX_RST at the same time. **Note3:** If SC_CTL [SC_CEN] is not enabled, this filed can not be programmed.
[2]	**DACT_EN**	**Deactivation Sequence Generator Enable** This bit enables SC controller to initiate the card by deactivation sequence 1 = Deactivation sequence generator Enabled. 0 = No effect. **Note1:** When the deactivation sequence completed, this bit will be cleared automatically and the SC_ISR [INIT_IS] will be set to "1". **Note2:** This field will be cleared by TX_RST and RX_RST. So don't fill this bit, TX_RST, and RX_RST at the same time. **Note3:** If SC_CTL [SC_CEN] is not enabled, this filed can not be programmed.
[1]	**RX_RST**	**RX Software Reset** When RX_RST is set, all the bytes in the receiver buffer and RX internal state machine will be cleared. 1 = Reset the RX internal state machine and pointers.

Bits	Description	
		0 = No effect. **Note:** This bit will be auto cleared and needs at least 3 SC engine clock cycles.
[0]	**TX_RST**	**TX Software Reset** When TX_RST is set, all the bytes in the transmit buffer and TX internal state machine will be cleared. 1 = Reset the TX internal state machine and pointers. 0 = No effect. **Note:** This bit will be auto cleared and needs at least 3 SC engine clock cycles.

SC Extend Guard Time Register (SCx_EGTR)

Register	Offset	R/W	Description	Reset Value
SC_EGTR **x=0,1,2**	SCx_BA+0x0C	R/W	SC Extend Guard Time Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EGT							

Bits	Description	
[31:8]	**Reserved**	**Reserved.**
[7:0]	**EGT**	**Extended Guard Time** This field indicates the extended guard timer value.  **Note:** The counter is ETU based and the real extended guard time is EGT.

SC Receiver Buffer Time-out Register (SCx_RFTMR)

Register	Offset	R/W	Description	Reset Value
SC_RFTMR **x=0,1,2**	SCx_BA+0x10	R/W	SC Receive Buffer Time-Out Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							RFTM
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RFTM							

Bits	Description	
[31:9]	**Reserved**	**Reserved.**
[8:0]	**RFTM**	**SC Receiver Buffer Time-Out Register (ETU Based)** The time-out counter resets and starts counting whenever the RX buffer received a new data word. Once the counter decrease to “1” and no new data is received or CPU does not read data by reading SC_RBR register, a receiver time-out interrupt INT_RTMR will be generated(if SC_IER[RTMR_IE] is high). **Note1:** The counter is ETU based and the real count value is RFTM + 1 **Note2:** Fill all “0” to this field to disable this function.

SC Clock Divider Control Register (SCx_ETUCR)

Register	Offset	R/W	Description	Reset Value
SC_ETUCR **x=0,1,2**	SCx_BA+0x14	R/W	SC ETU Control Register.	0x0000_0173

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
COMPEN_EN	**Reserved**			**ETU_RDIV**			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ETU_RDIV							

Bits	Description	
[31:16]	**Reserved**	**Reserved.**
[15]	**COMPEN_EN**	**Compensation Mode Enable** This bit enables clock compensation function. When this bit enabled, hardware will alternate between n clock cycles and (n-1) clock cycles, where n is the value to be written into the ETU_RDIV register. 1 = Compensation function Enabled. 0 = Compensation function Disabled.
[14:12]	**Reserved**	**Reserved.**
[11:0]	**ETU_RDIV**	**ETU Rate Divider** The field indicates the clock rate divider. The real ETU is ETU_RDIV + 1. **Note1:** Software can configure this field, but this field must be greater than 0x04. **Note2:** Software can configure this field, but if the error rate is equal to 2%, this field must be greater than 0x040.

SC Interrupt Control Register (SCx_IER)

Register	Offset	R/W	Description	Reset Value
SC_IER x=0,1,2	SCx_BA+0x18	R/W	SC Interrupt Enable Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					ACON_ERR_IE	RTMR_IE	INIT_IE
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CD_IE	BGT_IE	TMR2_IE	TMR1_IE	TMR0_IE	TERR_IE	TXBE_IE	RDA_IE

Bits	Description	
[31:11]	**Reserved**	**Reserved**
[10]	**ACON_ERR_IE**	**Auto convention Error Interrupt Enable** This field is used for auto convention error interrupt enable. 1 = INT_ACON_ERR Enabled. 0 = INT_ACON_ERR Disabled.
[9]	**RTMR_IE**	**Receiver Buffer Time-Out Interrupt Enable** This field is used for receiver buffer time-out interrupt enable. 1 = INT_RTMR Enabled 0 = INT_RTMR Disabled
[8]	**INIT_IE**	**Initial End Interrupt Enable** This field is used for activation (SC_ALTCTL [ACT_EN]), deactivation (SC_ALTCTL [DACT_EN]) and warm reset (SC_ALTCTL [WARST_EN]) sequence interrupt enable. 1 = INT_INIT Enabled 0 = INT_INIT Disabled
[7]	**CD_IE**	**Card Detect Interrupt Enable** This field is used for card detect interrupt enable. The card detect status register is SC_PINCSR [CD_CH] and SC_PINCSR[CD_CL]. 1 = INT_CD Enabled 0 = INT_CD Disabled.
[6]	**BGT_IE**	**Block Guard Time Interrupt Enable** This field is used for block guard time interrupt enable. 1 = INT_BGT Enabled 0 = INT_BGT Disabled

Bits	Description	
[5]	**TMR2_IE**	**Timer2 Interrupt Enable** This field is used for TMR2 interrupt enable. 1 = INT_TMR2 Enabled 0 = INT_TMR2 Disabled
[4]	**TMR1_IE**	**Timer1 Interrupt Enable** This field is used for TMR1 interrupt enable. 1 = INT_TMR1 Enabled. 0 = INT_TMR1 Disabled.
[3]	**TMR0_IE**	**Timer0 Interrupt Enable** This field is used for TMR0 interrupt enable. 1 = INT_TMR0 Enabled 0 = INT_TMR0 Disabled
[2]	**TERR_IE**	**Transfer Error Interrupt Enable** This field is used for transfer error interrupt enable. The transfer error states is at SC_TRSR register which includes receiver break error (RX_EBR_F), frame error (RX_EFR_F), parity error (RX_EPA_F), receiver buffer overflow error (RX_OVER_F), transmit buffer overflow error (TX_OVER_F), receiver retry over limit error (RX_OVER_ERETRY) and transmitter retry over limit error (TX_OVER_ERETRY). 1 = INT_TERR Enabled 0 = INT_TERR Disabled
[1]	**TBE_IE**	**Transmit Buffer Empty Interrupt Enable** This field is used for transmit buffer empty interrupt enable. 1 = INT_THRE Enabled. 0 = INT_THRE Disabled.
[0]	**RDA_IE**	**Receive Data Reach Interrupt Enable** This field is used for received data reaching trigger level (SC_CTL [RX_FTRI_LEV]) interrupt enable. 1 = INT_RDR Enabled. 0 = INT_RDR Disabled.

SC Interrupt Status Register (SCx_ISR)

Register	Offset	R/W	Description	Reset Value
SC_ISR **x=0,1,2**	SCx_BA+0x1C	R/W	SC Interrupt Status Register.	0x0000_0002

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**ACON_ERR_IS**	**RTMR_IS**	**INIT_IS**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CD_IS	**BGT_IS**	**TMR2_IS**	**TMR1_IS**	**TMR0_IS**	**TERR_IS**	**TBE_IS**	**RDA_IS**

Bits	Description	
[31:11]	**Reserved**	**Reserved.**
[10]	**ACON_ERR_IS**	**Auto Convention Error Interrupt Status Flag (Read Only)** This field indicates auto convention sequence error. If the received TS at ATR state is not 0x3B or 0x3F, this bit will be set. **Note:** This bit is read only, but can be cleared by writing “1” to it.
[9]	**RTMR_IS**	**Receiver buffer Time-Out Interrupt Status Flag (Read Only)** This field is used for receiver buffer time-out interrupt status flag. **Note:** This field is the status flag of receiver buffer time-out state. If software wants to clear this bit, software must read the receiver buffer remaining data by reading SC_RBR register,
[8]	**INIT_IS**	**Initial End Interrupt Status Flag (Read Only)** This field is used for activation (SC_ALTCTL [ACT_EN]), deactivation (SC_ALTCTL [DACT_EN]) and warm reset (SC_ALTCTL [WARST_EN]) sequence interrupt status flag. **Note:** This bit is read only, but it can be cleared by writing “1” to it.
[7]	**CD_IS**	**Card Detect Interrupt Status Flag (Read Only)** This field is used for card detect interrupt status flag. The card detect status register is SC_PINCSR [CD_INS_F] and SC_PINCSR [CD_REM_F]. **Note:** This field is the status flag of SC_PINCSR [CD_INS_F] or SC_PINCSR [CD_REM_F]. So if software wants to clear this bit, software must write “1” to this field.
[6]	**BGT_IS**	**Block Guard Time Interrupt Status Flag (Read Only)** This field is used for block guard time interrupt status flag. **Note:** This bit is read only, but it can be cleared by writing “1” to it.
[5]	**TMR2_IS**	**Timer2 Interrupt Status Flag (Read Only)** This field is used for TMR2 interrupt status flag. **Note:** This bit is read only, but it can be cleared by writing “1” to it.

Bits	Description	
[4]	**TMR1_IS**	**Timer1 Interrupt Status Flag (Read Only)** This field is used for TMR1 interrupt status flag. **Note:** This bit is read only, but it can be cleared by writing “1” to it.
[3]	**TMR0_IS**	**Timer0 Interrupt Status Flag (Read Only)** This field is used for TMR0 interrupt status flag. **Note:** This bit is read only, but it can be cleared by writing “1” to it.
[2]	**TERR_IS**	**Transfer Error Interrupt Status Flag (Read Only)** This field is used for transfer error interrupt status flag. The transfer error states is at SC_TRSR register which includes receiver break error (RX_EBR_F), frame error (RX_EFR_F), parity error (RX_EPA_F) and receiver buffer overflow error (RX_OVER_F), transmit buffer overflow error (TX_OVER_F), receiver retry over limit error (RX_OVER_ERETRY) and transmitter retry over limit error (TX_OVER_ERETRY). **Note:** This field is the status flag of SC_TRSR [RX_EBR_F], SC_TRSR [RX_EFR_F], SC_TRSR [RX_EPA_F], SC_TRSR [RX_OVER_F], SC_TRSR [TX_OVER_F], SC_TRSR [RX_OVER_ERETRY] or SC_TRSR [TX_OVER_ERETRY]. So if software wants to clear this bit, software must write “1” to each field.
[1]	**TBE_IS**	**Transmit Buffer Empty Interrupt Status Flag (Read Only)** This field is used for transmit buffer empty interrupt status flag. This bit is different with SC_TRSR [TX_EMPTY_F] flag and SC_TRSR [TX_ATV] flag; The TX_EMPTY_F will be set when the last byte data be read to shift register and TX_ATV flag indicates the transmitter is in active or not (the last data has been transmitted or not), but the TBE_IS may be set when the last byte data be read to shift register or the last data has been transmitted. When this bit assert, software can write 1~4 byte data to SC_THR register. **Note:** If software wants to clear this bit, software must write data to SC_THR register and then this bit will be cleared automatically.
[0]	**RDA_IS**	**Receive Data Reach Interrupt Status Flag (Read Only)** This field is used for received data reaching trigger level (SC_CTL [RX_FTRI_LEV]) interrupt status flag. **Note:** This field is the status flag of received data reaching SC_CTL [RX_FTRI_LEV]. If software reads data from SC_RBR and receiver pointer is less than SC_CTL [RX_FTRI_LEV], this bit will be cleared automatically.

SC Transfer Status Register (SCx_TRSR)

Register	Offset	R/W	Description	Reset Value
SC_TRSR **x=0,1,2**	SCx_BA+0x20	R/W	SC Transfer Status Register.	0x0000_0202

31	30	29	28	27	26	25	24
TX_ATV	TX_OVER_ERETRY	TX_ERETRY_F	Reserved		TX_POINT_F		
23	22	21	20	19	18	17	16
RX_ATV	RX_OVER_ERETRY	RX_ERETRY_F	Reserved		RX_POINT_F		
15	14	13	12	11	10	9	8
Reserved					TX_FULL_F	TX_EMPTY_F	TX_OVER_F
7	6	5	4	3	2	1	0
Reserved	RX_EBR_F	RX_EFR_F	RX_EPA_F	Reserved	RX_FULL_F	RX_EMPTY_F	RX_OVER_F

Bits	Description	
[31]	**TX_ATV**	**Transmit In Active Status Flag (Read Only)** This bit is set by hardware when TX transfer is in active or the last byte transmission has not completed. This bit is cleared automatically when TX transfer is finished and the STOP bit (include guard time) has been transmitted.
[30]	**TX_OVER_ERETRY**	**Transmitter Over Retry Error (Read Only)** This bit is set by hardware when transmitter re-transmits over retry number limitation. **Note:** This bit is read only, but it can be cleared by writing “1” to it.
[29]	**TX_ERETRY_F**	**Transmitter Retry Error (Read Only)** This bit is set by hardware when transmitter re-transmits. **Note1:** This bit is read only, but it can be cleared by writing “1” to it. Note2 This bit is a flag and can not generate any interrupt to CPU.
[28:27]	**Reserved**	**Reserved.**
[26:24]	**TX_POINT_F**	**Transmit Buffer Pointer Status Flag (Read Only)** This field indicates the TX buffer pointer status flag. When CPU writes data into SC_THR, TX_POINT_F increases one. When one byte of TX Buffer is transferred to transmitter shift register, TX_POINT_F decreases one.
[23]	**RX_ATV**	**Receiver In Active Status Flag (Read Only)** This bit is set by hardware when RX transfer is in active. This bit is cleared automatically when RX transfer is finished.
[22]	**RX_OVER_ERETRY**	**Receiver Over Retry Error (Read Only)** This bit is set by hardware when RX transfer error retry over retry number limit. **Note1:** This bit is read only, but it can be cleared by writing “1” to it. **Note2:** If CPU enables receiver retries function by setting SC_CTL [RX_ERETRY_EN]

Bits	Description	
		register, the RX_EPA_F flag will be ignored (hardware will not set RX_EPA_F).
[21]	RX_ERETRY_F	**Receiver Retry Error (Read Only)** This bit is set by hardware when RX has any error and retries transfer. **Note1:** This bit is read only, but it can be cleared by writing "1" to it. Note2 This bit is a flag and can not generate any interrupt to CPU. **Note3:** If CPU enables receiver retry function by setting SC_CTL [RX_ERETRY_EN] register, the RX_EPA_F flag will be ignored (hardware will not set RX_EPA_F).
[20:19]	Reserved	Reserved.
[18:16]	RX_POINT_F	**Receiver Buffer Pointer Status Flag (Read Only)** This field indicates the RX buffer pointer status flag. When SC receives one byte from external device, RX_POINT_F increases one. When one byte of RX buffer is read by CPU, RX_POINT_F decreases one.
[15:11]	Reserved	Reserved.
[10]	TX_FULL_F	**Transmit buffer Full Status flag (Read Only)** This bit indicates TX buffer full or not. This bit is set when TX pointer is equal to 4, otherwise is cleared by hardware.
[9]	TX_EMPTY_F	**Transmit buffer Empty Status Flag (Read Only)** This bit indicates TX buffer empty or not. When the last byte of TX buffer has been transferred to Transmitter Shift Register, hardware sets this bit high. It will be cleared when writing data into SC_THR (TX buffer not empty).
[8]	TX_OVER_F	**TX Overflow Error Interrupt Status Flag (Read Only)** If TX buffer is full (TX_FULL_F = "1"), an additional write data to SC_THR will cause this bit to logic "1". **Note1:** This bit is read only, but it can be cleared by writing "1" to it. **Note2:** The additional write data will be ignored.
[7]	Reserved	Reserved.
[6]	RX_EBR_F	**Receiver Break Error Status Flag (Read Only)** This bit is set to a logic "1" whenever the received data input (RX) held in the "spacing state" (logic "0") is longer than a full word transmission time (that is, the total time of "start bit" + data bits + parity + stop bits). . **Note1:** This bit is read only, but it can be cleared by writing "1" to it. **Note2:** If CPU sets receiver retries function by setting SC_CTL [RX_ERETRY_EN] register, hardware will not set this flag.
[5]	RX_EFR_F	**Receiver Frame Error Status Flag (Read Only)** This bit is set to logic "1" whenever the received character does not have a valid "stop bit" (that is, the stop bit following the last data bit or parity bit is detected as a logic "0"). **Note1:** This bit is read only, but can be cleared by writing "1" to it. **Note2:** If CPI sets receiver retries function by setting SC_CTL [RX_ERETRY_EN] register, hardware will not set this flag.
[4]	RX_EPA_F	**Receiver Parity Error Status Flag (Read Only)** This bit is set to logic "1" whenever the received character does not have a valid "parity bit". **Note1:** This bit is read only, but it can be cleared by writing "1" to it.

Bits	Description	
		Note2: If CPU sets receiver retries function by setting SC_CTL [RX_ERETRY_EN] register, hardware will not set this flag.
[3]	**Reserved**	**Reserved.**
[2]	**RX_FULL_F**	**Receiver Buffer Full Status Flag (Read Only)** This bit indicates RX buffer full or not. This bit is set when RX pointer is equal to 4, otherwise it is cleared by hardware.
[1]	**RX_EMPTY_F**	**Receiver Buffer Empty Status Flag(Read Only)** This bit indicates RX buffer empty or not. When the last byte of RX buffer has been read by CPU, hardware sets this bit high. It will be cleared when SC receives any new data.
[0]	**RX_OVER_F**	**RX Overflow Error Status Flag (Read Only)** This bit is set when RX buffer overflow. If the number of received bytes is greater than RX Buffer (SC_RBR) size, 4 bytes of SC, this bit will be set. **Note1:** This bit is read only, but it can be cleared by writing “1” to it. **Note2:** The overwrite data will be ignored.

SC Pin Control State Register (SCx_PINCSR)

Register	Offset	R/W	Description	Reset Value
SC_PINCSR **x=0,1,2**	SCx_BA+0x24	R/W	SC Pin Control State Register.	0x0000_00x0

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							**SC_DATA_I_ST**
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**CD_LEV**	**SC_DATA_O**	**SC_OEN_ST**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ADAC_CD_EN	**CLK_KEEP**	**CLK_STOP_LEV**	**CD_PIN_ST**	**CD_INS_F**	**CD_REM_F**	**SC_RST**	**POW_EN**

Bits	Description	
[31:17]	**Reserved**	**Reserved.**
[16]	**SC_DATA_I_ST**	**SC Data Input Pin Status (Read Only)** This bit is the pin status of SC_DATA_I 1 = The SC_DATA_I pin is high. 0 = The SC_DATA_I pin is low.
[15:11]	**Reserved**	**Reserved.**
[10]	**CD_LEV**	**Card Detect Level** 1 = When hardware detects the card detect pin from low to high, it indicates a card is detected. 0 = When hardware detects the card detect pin from high to low, it indicates a card is detected.  **Note:** Software must select card detect level before Smart Card engine enable
[9]	**SC_DATA_O**	**Output of SC Data Pin** This bit is the pin status of SC data output but user can drive this pin to high or low by setting this bit. 1 = Drive SC data output pin to high. 0 = Drive SC data output pin to low. **Note:** When SC is at activation, warm re set or deactivation mode, this bit will be changed automatically. So don't fill this field when SC is in these modes.
[8]	**SC_OEN_ST**	**SC Data Pin Output Enable Status (Read Only)**

Bits	Description	
		1 = SC data output enable pin status is at high 0 = SC data output enable pin status is at low
[7]	**ADAC_CD_EN**	**Auto Deactivation When Card Removal** 1 = Auto deactivation Enabled when hardware detected the card is removal 0 = Auto deactivation Disabled when hardware detected the card is removal. **Note1:** When the card is removal, hardware will stop any process and then do deactivation sequence (if this bit be setting). If this process completes. Hardware will generate an interrupt INT_INIT to CPU.
[6]	**CLK_KEEP**	**SC Clock Enable** 1 = SC clock always keeps free running. 0 = SC clock generation Disabled. **Note:** When operation at activation, warm reset or deactivation mode, this bit will be changed automatically. So don't fill this field When operating in these modes.
[5]	**CLK_STOP_LEV**	**SC Clock Stop Level** This field indicates the clock polarity control in clock stop mode. 1 = SC_CLK stopped in high level. 0 = SC_CLK stopped in low level.
[4]	**CD_PIN_ST**	**Card Detect Status Of SC_CD Pin Status (Read Only)** This bit is the pin status flag of SC_CD 1 = SC_CD pin state at high. 0 = SC_CD pin state at low.
[3]	**CD_INS_F**	**Card Detect Insert Status Of SC_CD Pin (Read Only)** This bit is set whenever card has been inserted. 1 = Card insert. 0 = No effect. **Note1:** This bit is read only, but it can be cleared by writing "1" to it. **Note2:** Card detect engine will start after SC_CTL [SC_CEN] set.
[2]	**CD_REM_F**	**Card Detect Removal Status Of SC_CD Pin (Read Only)** This bit is set whenever card has been removal. 1 = Card Removal. 0 = No effect. **Note1:** This bit is read only, but it can be cleared by writing "1" to it. **Note2:** Card detect engine will start after SC_CTL [SC_CEN] set.
[1]	**SC_RST**	**SC_RST Pin Signal** This bit is the pin status of SC_RST but user can drive SC_RST pin to high or low by setting this bit. 1 = Drive SC_RST pin to high. 0 = Drive SC_RST pin to low. **Note:** When operation at activation, warm reset or deactivation mode, this bit will be changed automatically. So don't fill this field When operating in these modes.
[0]	**POW_EN**	**SC_POW_EN Pin Signal** This bit is the pin status of SC_POW_EN but user can drive SC_POW_EN pin to high or

Bits	Description	
		low by setting this bit. 1 = Drive SC_POW_EN pin to high. 0 = Drive SC_POW_EN pin to low. **Note:** When operation at activation, warm reset or deactivation mode, this bit will be changed automatically. So don't fill this field When operating in these modes.

SC Timer Control Register 0 (SC_TMR0)

Register	Offset	R/W	Description	Reset Value
SC_TMR0 **x=0,1,2**	SCx_BA+0x28	R/W	SC Internal Timer Control Register 0.	0x0000_0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="4">Reserved</td><td colspan="4">MODE</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="8">CNT</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td colspan="8">CNT</td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">CNT</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:28]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[27:24]</td><td>MODE</td><td>Timer 0 Operation Mode Selection
This field indicates the internal 24 bit timer operation selection.
<table>
<tr><th>TMR0_SEL</th><th colspan="2">Operation Description</th></tr>
<tr><td rowspan="3">0000</td><td colspan="2">Down counter starts when SC_ALTCTL [TMR0_SEN] enabled and ends when counter time-out. The time-out value will be CNT+1</td></tr>
<tr><td>Start</td><td>Start count when SC_ALTCTL [TMR0_SEN] enabled</td></tr>
<tr><td>End</td><td>When the down counter is equal to “0”, hardware will set TMR0_IS and clear SC_ALTCTL [TMR0_SEN] automatically.</td></tr>
<tr><td rowspan="3">0001</td><td colspan="2">Down counter starts when first START bit detected and ends when counter time-out. The time-out value will be CNT+1.</td></tr>
<tr><td>Start</td><td>Start</td></tr>
<tr><td>End</td><td>End</td></tr>
<tr><td rowspan="3">0010</td><td colspan="2">Down counter starts when the first START bit detected (reception) and ends when counter time-out occur. The time-out value will be CNT+1.</td></tr>
<tr><td>Start</td><td>Start count when the first START detected bit (reception) after SC_ALTCTL [TMR0_SEN] set to “1”.</td></tr>
<tr><td>End</td><td>When the down counter is equal to “0”, hardware will set TMR0_IS and clear SC_ALTCTL [TMR0_SEN] automatically.</td></tr>
<tr><td>0011</td><td colspan="2">Down counter is only used for hardware activation, warm reset sequence to measure ATR timing.
The timing starts when SC_RST de-assertion and ends when ATR response received or time-out.
If the counter decreases to “0” before ATR response received, hardware will generate an interrupt to CPU. The time-out value will be</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td></td><td>
<table>
<tr><td rowspan="3"></td><td colspan="2">CNT+1.</td></tr>
<tr><td>Start</td><td>Start count when SC_RST de-assertion after SC_ALTCTL [TMR0_SEN] set to “1”.
It is used for hardware activation, warm reset mode.</td></tr>
<tr><td>End</td><td>When the down counter is equal to “0” before ATR response received, hardware will set TMR0_IS and clear SC_ALTCTL [TMR0_SEN] automatically.
When ATR received and down counter does not equal to “0”, hardware will clear SC_ALTCTL [TMR0_SEN] automatically.</td></tr>
<tr><td>0100</td><td colspan="2">Same as mode 0000, but when the down counter is equal to “0”, hardware will set TMR0_IS and counter will re-load the SC_TMR0 [CNT] value and re-count until software clears SC_ALTCTL [TMR0_SEN].
When SC_ALTCTL [TMR0_ATV] = “1”, software can change SC_TMR0 [CNT] value at any time. When the down counter is equal to “0”, counter will reload the new value of SC_TMR0 [CNT] and re-count.
The time-out value will be CNT+1.</td></tr>
<tr><td>0101</td><td colspan="2">Same as mode 0001, but when the down counter is equal to “0”, hardware will set TMR0_IS and counter will re-load the SC_TMR0 [CNT] value. When the next START bit is detected, counter will re-count until software clears SC_ALTCTL [TMR0_SEN].
When SC_ALTCTL [TMR0_ATV] = “1” software can change SC_TMR0 [CNT] value at any time, when the down counter equal to “0”, it will reload the new value of SC_TMR0 [CNT] and re-counting.
The time-out value will be CNT+1.</td></tr>
<tr><td>0110</td><td colspan="2">Same as mode 0010, but when the down counter is equal to “0”, it will set TMR0_IS and counter will re-load the SC_TMR0 [CNT] value. When the next START bit is detected, counter will re-count until software clears SC_ALTCTL [TMR0_SEN].
When SC_ALTCTL [TMR0_ATV] = “1”, software can change SC_TMR0 [CNT] value at any time. When the down counter is equal to “0”, counter will reload the new value of SC_TMR0 [CNT] and re-count.
The time-out value will be CNT+1.</td></tr>
<tr><td rowspan="3">0111</td><td colspan="2">Down counter starts when first START bit detected and ends when software clears SC_ALTCTL [TMR0_SEN] bit. If next START bit detected, counter will reload the new value of SC_TMR0 [CNT] and re-counting.
If the counter decreases to “0” before the next START bit detected, hardware will generate an interrupt to CPU. The time-out value will be CNT+1.</td></tr>
<tr><td>Start</td><td>Start count when the first START bit detected after SC_ALTCTL [TMR0_SEN] set to “1”.</td></tr>
<tr><td>End</td><td>Stop count after SC_ALTCTL [TMR0_SEN] set to “0”.</td></tr>
<tr><td rowspan="2">1000</td><td colspan="2">Up counter starts when SC_ALTCTL [TMR0_SEN] enabled and ends when SC_ALTCTL [TMR0_SEN] disabled. This count value will be stored in SC_TDRA [23:0]. In this mode, hardware can not generate any interrupt to CPU. The real count value will be SC_TDRA [23:0] +1.</td></tr>
<tr><td>Start</td><td>Start count after SC_ALTCTL [TMR0_SEN] set to “1”, and the start count value is “0” (hardware will ignore CNT value).</td></tr>
</table>
</td></tr>
</table>

Bits	Description		
		End	Stop count after SC_ALTCTL [TMR0_SEN] set to "0" and store the value to SC_TDRA [23:0] register.
[23:0]	**CNT**	**Timer 0 Counter Value Register (ETU Base)** This field indicates the internal timer operation values.	

SC Timer Control Register 1 (SCx_TMR1)

Register	Offset	R/W	Description	Reset Value
SC_TMR1 **x=0,1,2**	SCx_BA+0x2C	R/W	SC Internal Timer Control Register 1.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				**MODE**			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT							

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:28]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[27:24]</td><td>MODE</td><td>Timer 1 Operation Mode Selection
This field indicates the internal 8-bit timer operation selection.
<table>
<tr><th>TMR1_SEL</th><th colspan="2">Operation Description</th></tr>
<tr><td rowspan="3">0000</td><td colspan="2">Down counter starts when SC_ALTSCR [TMR1_SEN] enabled and ends when counter time-out. The time-out value will be CNT+1</td></tr>
<tr><td>Start</td><td>Start count when SC_ALTCTL [TMR1_SEN] enabled</td></tr>
<tr><td>End</td><td>When the down counter is equal to “0”, hardware will set TMR1_IS and clear SC_ALTCTL [TMR1_SEN] automatically.</td></tr>
<tr><td rowspan="3">0001</td><td colspan="2">Down counter starts when the first START bit detected and ends when counter time-out. The time-out value will be CNT+1.</td></tr>
<tr><td>Start</td><td>Start count when the first START bit (reception or transmission) detected after SC_ALTCTL [TMR1_SEN] set to “1”.</td></tr>
<tr><td>End</td><td>When the down counter is equal to “0”, hardware will set TMR1_IS and clear SC_ALTCTL [TMR1_SEN] automatically.</td></tr>
<tr><td rowspan="3">0010</td><td colspan="2">Down counter starts when the first START bit detected (reception) and ends when counter time-out. The time-out value will be CNT+1.</td></tr>
<tr><td>Start</td><td>Start count when the first START bit detected (reception) after SC_ALTCTL [TMR1_SEN] set to “1”.</td></tr>
<tr><td>End</td><td>When the down counter is equal to “0”, hardware will set TMR1_IS and clear SC_ALTCTL [TMR1_SEN] automatically.</td></tr>
<tr><td>0100</td><td colspan="2">Same as mode 0000, but when the down counter is equal to “0”, hardware will set TMR1_IS and counter will re-load the SC_TMR1 [CNT] value and re-count until software clears SC_ALTCTL [TMR1_SEN].</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td colspan="2">When SC_ALTCTL [TMR1_ATV] = “1”, software can change SC_TMR1 [CNT] value at any time. When the down counter is equal to “0”, the counter will reload the new value of SC_TMR1 [CNT] and re-count.

The time-out value will be CNT+1.</td></tr>
<tr><td>0101</td><td colspan="2">Same as mode 0001, but when the down counter is equal to “0”, hardware will set TMR1_IS and counter will re-load the SC_TMR1 [CNT] value. When the next START bit is detected, counter will re-count until software clears SC_ALTCTL [TMR1_SEN].

When SC_ALTCTL [TMR1_ATV] = “1” software can change SC_TMR1 [CNT] value at any time. When the down counter is equal to “0”, the counter will reload the new value of SC_TMR1 [CNT] and re-count.

The time-out value will be CNT+1.</td></tr>
<tr><td>0110</td><td colspan="2">Same as mode 0010, but when the down counter is equal to “0”, hardware will set TMR1_IS and counter will re-load the SC_TMR1 [CNT] value. When the next START bit is detected, counter will re-count until software clears SC_ALTCTL [TMR1_SEN].

When SC_ALTCTL [TMR1_ATV] = “1”, software can change SC_TMR1 [CNT] value at any time. When the down counter is equal to “0”, the counter will reload the new value of SC_TMR1 [CNT] and re-count.

The time-out value will be CNT+1.</td></tr>
<tr><td rowspan="3">0111</td><td colspan="2">Down counter starts when first START bit detected and ends when software clears SC_ALTCTL [TMR1_SEN] bit. If next START bit detected, counter will reload the new value of SC_TMR1 [CNT] and re-count.

If the counter decreases to “0” before the next START bit detected, hardware will generate an interrupt to CPU. The time-out value will be CNT+1.</td></tr>
<tr><td>Start</td><td>Start count when the first START bit detected after SC_ALTCTL [TMR1_SEN] set to “1”.</td></tr>
<tr><td>End</td><td>Stop count after SC_ALTCTL [TMR1_SEN] set to “0”.</td></tr>
<tr><td rowspan="3">1000</td><td colspan="2">Up counter starts when SC_ALTCTL [TMR1_SEN] enabled and ends when SC_ALTCTL [TMR1_SEN] disabled. This count value will be stored in SC_TDRB [7:0]. In this mode, hardware can not generate any interrupt to CPU. The real count value will be SC_TDRB [7:0] +1.</td></tr>
<tr><td>Start</td><td>Start count after SC_ALTCTL [TMR1_SEN] set to “1”, and the start count value is “0” (hardware will ignore CNT value).</td></tr>
<tr><td>End</td><td>Stop count after SC_ALTCTL [TMR1_SEN] set to “0” and store the value to SC_TDRB [7:0] register.</td></tr>
</table>
</td></tr>
<tr><td>[23:8]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[7:0]</td><td>CNT</td><td>Timer 1 Counter Value Register (ETU Base)

This field indicates the internal timer operation values.</td></tr>
</table>

SC Timer Control Register 2 (SCx_TMR2)

Register	Offset	R/W	Description	Reset Value
SC_TMR2 **x=0,1,2**	SCx_BA+0x30	R/W	SC Internal Timer Control Register 2.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				MODE			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CNT							

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:28]</td><td>Reserved</td><td>Reserved.</td></tr>
<tr><td>[27:24]</td><td>MODE</td><td>Timer 2 Operation Mode Selection
This field indicates the internal 8-bit timer operation selection.
<table>
<tr><th>TMR2_SEL</th><th colspan="2">Operation Description</th></tr>
<tr><td rowspan="3">0000</td><td colspan="2">Down counter starts when SC_ALTCTL [TMR2_SEN] enabled and ends when counter time-out. The time-out value will be CNT+1</td></tr>
<tr><td>Start</td><td>Start count when SC_ALTCTL [TMR2_SEN] enabled</td></tr>
<tr><td>End</td><td>When the down counter is equal to “0”, the controller will set TMR2_IS and clear SC_ALTCTL [TMR2_SEN] automatically.</td></tr>
<tr><td rowspan="3">0001</td><td colspan="2">Down counter starts when the first START bit detected and ends when counter time-out. The time-out value will be CNT+1.</td></tr>
<tr><td>Start</td><td>Start count when the first START bit (reception or transmission) detected after SC_ALTCTL [TMR2_SEN] set to “1”.</td></tr>
<tr><td>End</td><td>When the down counter is equal to “0”, hardware will set TMR2_IS and clear SC_ALTCTL [TMR2_SEN] automatically.</td></tr>
<tr><td rowspan="2">0010</td><td colspan="2">Down counter starts when first START bit detected (reception) and ends when counter time-out. The time-out value will be CNT+1.</td></tr>
<tr><td>Start</td><td>Start count when the first START bit detected (reception) after SC_ALTCTL [TMR2_SEN] set to “1”.</td></tr>
</table></td></tr>
</table>

Bits	Description		
			End: When the down counter is equal to "0", hardware will set TMR2_IS and clear SC_ALTCTL [TMR2_SEN] automatically.
		0100	Same as mode 0000, but when the down counter is equal to "0", hardware will set TMR2_IS and counter will re-load the SC_TMR2 [CNT] value and re-count until software clears SC_ALTCTL [TMR2_SEN]. When SC_ALTCTL [TMR2_ATV] = "1" software can change SC_TMR2 [CNT] value at any time. When the down counter is equal to "0", the counter will reload the new value of SC_TMR2 [CNT] and re-count. The time-out value will be CNT+1.
		0101	Same as mode 0001, but when the down counter is equal to "0", hardware will set TMR2_IS and counter will re-load the SC_TMR2 [CNT] value. When the next START bit is detected counter will re-count until software clears SC_ALTCTL [TMR2_SEN]. When SC_ALTCTL [TMR2_ATV] = "1", software can change SC_TMR2 [CNT] value at any time. When the down counter is equal to "0", the counter will reload the new value of SC_TMR2 [CNT] and re-count. The time-out value will be CNT+1.
		0100	Same as mode 0000, but when the down counter is equal to "0", hardware will set TMR2_IS and counter will re-load the SC_TMR2 [CNT] value and re-count until software clears SC_ALTCTL [TMR2_SEN]. When SC_ALTCTL [TMR2_ATV] = "1" software can change SC_TMR2 [CNT] value at any time. When the down counter is equal to "0", the counter will reload the new value of SC_TMR2 [CNT] and re-count. The time-out value will be CNT+1.
		0101	Same as mode 0001, but when the down counter is equal to "0", hardware will set TMR2_IS and counter will re-load the SC_TMR2 [CNT] value. When the next START bit is detected counter will re-count until software clears SC_ALTCTL [TMR2_SEN]. When SC_ALTCTL [TMR2_ATV] = "1", software can change SC_TMR2 [CNT] value at any time. When the down counter is equal to "0", the counter will reload the new value of SC_TMR2 [CNT] and re-count. The time-out value will be CNT+1.
		0110	Same as mode 0010, but when the down counter is equal to "0", it will set TMR2_IS and re-load the SC_TMR2 [CNT] value. When the next START bit is detected it will re-count until software clears SC_ALTCTL [TMR2_SEN]. When SC_ALTCTL [TMR2_ATV] = "1" software can change SC_TMR2 [CNT] value at any time. When the down counter is equal to "0", it will reload the new value of SC_TMR2 [CNT] and re-count. The time-out value will be CNT+1.
		0111	Down counter starts from first START bit and ends after software clears SC_ALTCTL [TMR2_SEN] bit. If counter detects next START bit, it will reload the new value of SC_TMR2 [CNT] and re-count. If the counter decreases to "0" before detection the next START bit, it will generate an interrupt to CPU. The time-out value will be CNT+1. Start: Start count on the first START bit after SC_ALTCTL

Bits	Description			
				[TMR2_SEN] set to "1".
			End	Stop count after SC_ALTCTL [TMR2_SEN] set to "0".
		1000	Up counter starts from SC_ALTCTL [TMR2_SEN] enabled and end after SC_ALTCTL [TMR2_SEN] disabled. This count value will be stored in SC_TDRB [15:8]. In this mode, it can not generator any interrupt to CPU. The real count value will be SC_TDRB [15:8] +1.	
			Start	Start count after SC_ALTCTL [TMR1_SEN] set to "1", and the start count value is "0" (hardware will ignore CNT value).
			End	Stop count after SC_ALTCTL [TMR2_SEN] set to "0" and store the value to SC_TDRB [15:8] register.
[23:8]	**Reserved**	**Reserved.**		
[7:0]	**CNT**	**Timer 2 Counter Value Register (ETU Base)** This field indicates the internal timer operation values.		

SC UART Mode Control Register (SCx_UACTL)

Register	Offset	R/W	Description	Reset Value
SCx_UACTL **x=0,1,2**	SCx_BA+0x34	R/W	SC UART Mode Control Register.	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OPE	PBDIS	DATA_LEN		Reserved			UA_MODE_EN

Bits	Description	
[31:8]	-	**Reserved.**
[7]	**OPE**	**Odd Parity Enable** 1 = Odd number of logic 1's are transmitted or check the data word and parity bits in receiving mode. 0 = Even number of logic 1's are transmitted or check the data word and parity bits in receiving mode. **Note:** This bit has effect only when PBDIS bit is '0'.
[6]	**PBDIS**	**Parity Bit Disable** 1 = Parity bit is not generated (transmitting data) or checked (receiving data) during transfer. 0 = Parity bit is generated or checked between the "last data word bit" and "stop bit" of the serial data. **Note:** In Smart Card mode, this field must be '0' (default setting is with parity bit)
[5:4]	**DATA_LEN**	**Data Length** DATA_LEN = 00: 8 bits; 01: 7 bits; 10: 6 bits; 11: 5 bits **Note:** In Smart Card mode, this field must be '00'
[3:1]	**Reserved**	**Reserved**
[0]	**UA_MODE_EN**	**UART Mode Enable** 1 = UART mode.

DATA_LEN (bits [5:4]):

DATA_LEN	Character Length
00	8 bits
01	7 bits
10	6 bits
11	5 bits

Bits	Description	
		0 = Smart Card mode. **Note1:** When operating in UART mode, user must set SCx_CTL [CON_SEL] and SCx_CTL [AUTO_CON_EN] to “0”. **Note2:** When operating in smart card mode, user must set SCx_UACTL [7:0] register to “0”. **Note3:** When UART is enabled, hardware will generate a reset to reset internal buffer and internal state machine.

SC Timer Current Data Register A (SCx_TDRA)

Register	Offset	R/W	Description	Reset Value
SC_TDRA **x=0,1,2**	SCx_BA+0x38	R	SC Timer Current Data Register A.	0x0000_07FF

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TDR0							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TDR0							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TDR0							

Bits	Description	
[31:24]	**Reserved**	**Reserved.**
[23:0]	**TDR0**	**Timer0 Current Data Register (Read Only)** This field indicates the current count values of timer0.

SC Timer Current Data Register B (SCx_TDRB)

Register	Offset	R/W	Description	Reset Value
SC_TDRB **x=0,1,2**	SCx_BA+0x3C	R	SC Timer Current Data Register B.	0x0000_7F7F

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TDR2							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TDR1							

Bits	Description	
[31:16]	**Reserved**	**Reserved.**
[15:8]	**TDR2**	**Timer2 Current Data Register (Read Only)** This field indicates the current count values of timer2.
[7:0]	**TDR1**	**Timer1 Current Data Register (Read Only)** This field indicates the current count values of timer1.

5.16 I²C

5.16.1 Overview

I^2C is a two-wire, bi-directional serial bus that provides a simple and efficient method of data exchange between devices. The I^2C standard is a true multi-master bus including collision detection and arbitration that prevents data corruption if two or more masters attempt to control the bus simultaneously. Serial, 8-bit oriented bi-directional data transfers can be made up to 1.0 Mbps.

Data is transferred between a Master and a Slave synchronously to SCL on the SDA line on a byte-by-byte basis. Each data byte is 8-bit long. There is one SCL clock pulse for each data bit with the MSB being transmitted first. An acknowledge bit follows each transferred byte.

A transition on the SDA line while SCL is high is interpreted as a command (START or STOP). Each bit is sampled during the high period of SCL; therefore, the SDA line may be changed only during the low period of SCL and must be held stable during the high period of SCL.

The controller's on-chip I^2C logic provides the serial interface that meets the I^2C bus standard mode specification. The I^2C controller handles byte transfers autonomously. Pull up resistor is needed for I^2C operation as these are open drain pins.

The I^2C controller is equipped with two slave address registers. The contents of the registers are irrelevant when I^2C is in Master mode. In the Slave mode, the seven most significant bits must be loaded with the user's own slave address. The I^2C hardware will react if the contents of I2CADDR are matched with the received slave address.

This controller supports the "General Call (GC)" function. If the GC bit is set this controller will respond to General Call address (00H). Clear GC bit to disable general call function. When GC bit is set and the I^2C is in Slave mode, it can receive the general call address which is equal to 00H after master sends general call address to the I^2C bus, then it will follow status of GC mode. If it is in Master mode, the ACK bit must be cleared when it sends general call address of 00H to the I^2C bus.

The I^2C-bus controller supports multiple address recognition with two address mask register. When the bit in the address mask register is set to one, it means the received corresponding address bit is don't-care. If the bit is set to zero, that means the received corresponding register bit should be exact the same as address register.

5.16.2 Features

- Acts as Master or Slave mode
- Bidirectional data transfer between masters and slaves
- Multi-master bus (no central master)
- Arbitration between simultaneously transmitting masters without corruption of serial data on the bus
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus
- Serial clock synchronization can be used as a handshake mechanism to suspend and resume serial transfer
- One built-in 14-bit time-out counter requesting the I^2C interrupt if the I^2C bus hangs up and timer-out counter overflows.
- Programmable clock divider allows versatile rate control
- Supports 7-bit addressing mode
- Supports multiple address recognition (Two slave addresses with mask option)
- Supports Power-down wake-up function

5.16.3 Functional Description

5.16.3.1 I²C Protocol

The following figure shows the typical I²C protocol. Normally, a standard communication consists of four parts:

- START or Repeated START signal generation
- STOP signal generation
- Slave address transfer
- Data transfer

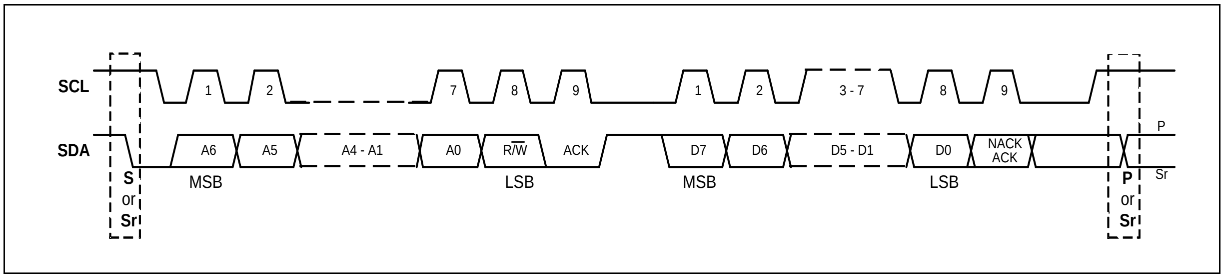


Figure 5.16-1 I²C Protocol

5.16.3.2 START or Repeated START Signal

When the bus is free/idle, meaning no master device is engaging the bus (both SCL and SDA lines are high), a master can initiate a transfer by sending a START signal. A START signal, usually referred to as the S-bit, is defined as a HIGH to LOW transition on the SDA line while SCL is HIGH. The START signal denotes the beginning of a new data transfer.

A Repeated START (Sr) is a START signal without first generating a STOP signal. The master uses this method to communicate with another slave or the same slave in a different transfer direction (e.g. from writing to a device to reading from a device) without releasing the bus.

5.16.3.3 STOP Signal

The master can terminate the communication by generating a STOP signal. A STOP signal, usually referred to as the P-bit, is defined as a LOW to HIGH transition on the SDA line while SCL is HIGH.

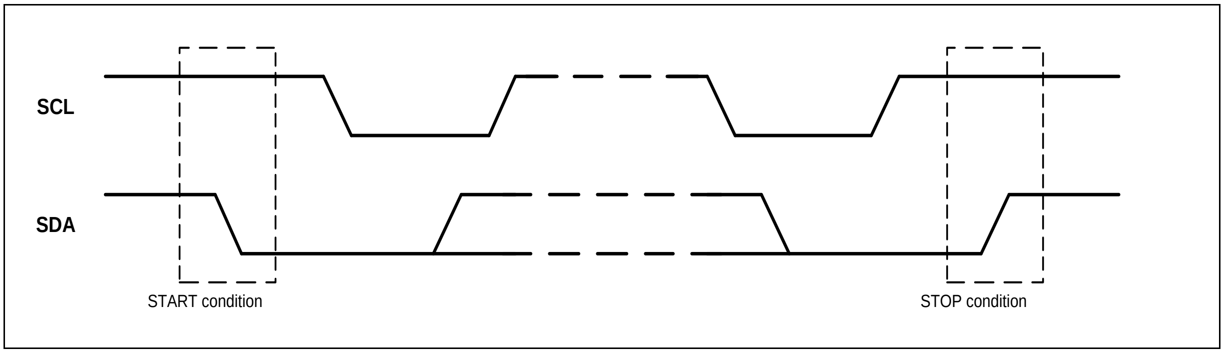


Figure 5.16-2 I²C START and STOP Conditions

5.16.3.4 Slave Address Transfer

The first byte of data transferred by the master immediately after the START signal is the slave address. This is a 7-bits calling address followed by the R/W bit. The R/W bit signals the slave the data transfer direction. No two slaves in the system can have the same address. Only the slave with an address that matches the one transmitted by the master will respond by returning an acknowledge bit by pulling the SDA low at the 9th SCL clock cycle.

5.16.3.5 Data Transfer

Once successful slave addressing has been achieved, the data transfer can proceed on a byte-by-byte basis in the direction specified by the RW bit sent by the master. Each transferred byte is followed by an acknowledge bit on the 9th SCL clock cycle. If the slave signals a Not Acknowledge (NACK), the master can generate a STOP signal to abort the data transfer or generate a Repeated START signal and start a new transfer cycle.

If the master, as the receiving device, does Not Acknowledge (NACK) to the slave, the slave releases the SDA line for the master to generate a STOP or Repeated START signal.

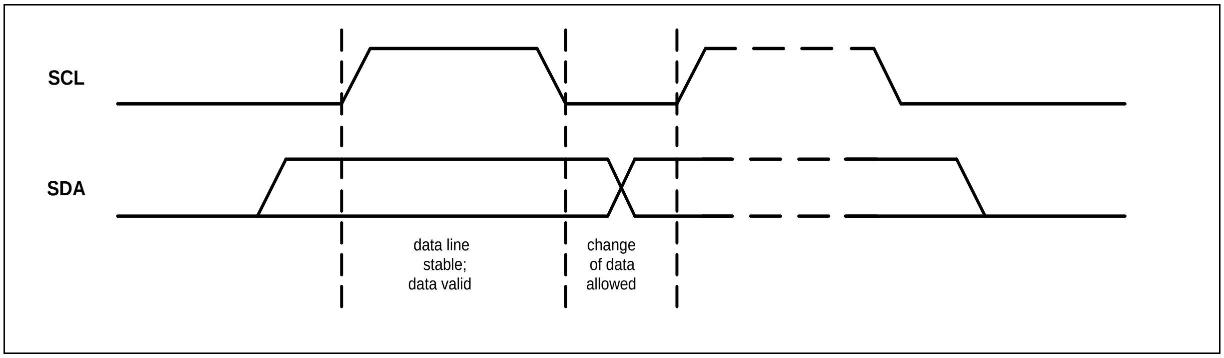


Figure 5.16-3 Bit Transfer on I^2C Bus

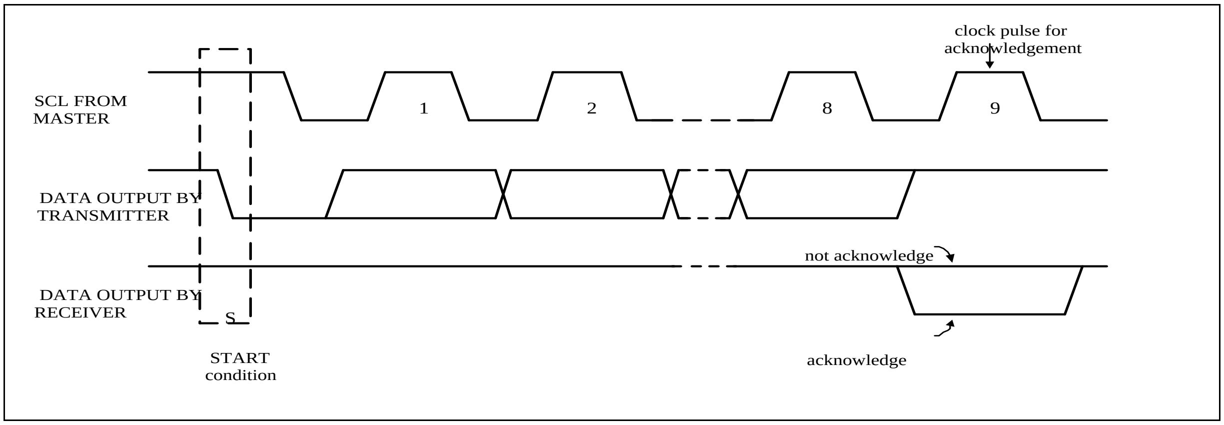


Figure 5.16-4 Acknowledge on I^2C Bus

5.16.3.6 I2C Protocol Register

To control I^2C port through the following special function registers: I2CON (control register), I2CINTSTS (interrupt status register), I2CSTATUS (status register), I2DIV (clock rate register), I2CTOUT (time-out counter register), I2CDAT (data register), I2CSADDRn (address registers, n=0~1), I2CSAMASKn (address mask registers, n=0~1).

5.16.3.7 Control Register (I2CON)

The CPU can read from and write to I2CON[7:0] directly. When the I^2C port is enabled by setting IPEN (I2CON [0]) to high, the internal states will be controlled by I2CON and I^2C logic hardware.

There are two bits are affected by hardware: the INTSTS (I2CINTSTS[0]) bit is set when the I^2C hardware requests a serial interrupt, and the STOP (I2CON[2]) bit is cleared when a STOP condition is present on the bus. The STOP bit is also cleared when IPEN = 0.

Once a new status code is generated and stored in I2CSTATUS, the I^2C Interrupt Flag bit INTSTS (I2CINTSTS [0]) will be set automatically. If the Enable Interrupt bit INTEN (I2CON [7]) is set at this time, the I^2C interrupt will be generated. The bit field I2CSTATUS[7:0] stores the internal state code, the content keeps stable until INTSTS is cleared by software.

5.16.3.8 Interrupt Status Register (I2CINTSTS)

There are 2 interrupt status.

(1). INTSTS: When a new state is present in the I2CSTATUS register, this bit will be set automatically, and if INTEN bit is set, the I^2C interrupt is requested.

(2). TIF: Refer to the section I^2C time out counter.

5.16.3.9 Status Register (I2CSTATUS)

I2CSTATUS[7:0] is an 8-bit read-only register. The bit field I2CSTATUS[7:0] contain the status code. There are 26 possible status codes. When I2CSTATUS[7:0] is F8H, no serial interrupt is requested. All other I2CSTATUS[7:0] values correspond to defined I^2C states. When each of these states is entered, a status interrupt is requested (INTSTS (I2CINTSTS[0]) = 1). A valid status code is present in I2CSTATUS[7:0] one cycle after INTSTS is set by hardware and is still present one cycle after INTSTS has been reset by software.

In addition, the state 00H stands for a Bus Error, which occurs when a START or STOP condition is present at an incorrect position in the I^2C format frame. A Bus Error may occur during the serial transfer of an address byte, a data byte or an acknowledge bit. To recover I^2C from bus error, STOP (I2CON[2]) should be set and INTSTS (I2CINTSTS[0]) should be clear to enter Not Addressed Slave mode. Then STOP is cleared to release bus and to wait for a new communication. I^2C bus cannot recognize stop condition during this action when bus error occurs.

Master Mode		**Slave Mode**	
STATUS	**Description**	**STATUS**	**Description**
0x08	Start	0xA0	Slave Transmit Repeat Start or Stop
0x10	Master Repeat Start	0xA8	Slave Transmit Address ACK
0x18	Master Transmit Address ACK	0xB0	Slave Transmit Arbitration Lost
0x20	Master Transmit Address NACK	0xB8	Slave Transmit Data ACK
0x28	Master Transmit Data ACK	0xC0	Slave Transmit Data NACK
0x30	Master Transmit Data NACK	0xC8	Slave Transmit Last Data ACK
0x38	Master Arbitration Lost	0x60	Slave Receive Address ACK

0x40	Master Receive Address ACK	0x68	Slave Receive Arbitration Lost
0x48	Master Receive Address NACK	0x80	Slave Receive Data ACK
0x50	Master Receive Data ACK	0x88	Slave Receive Data NACK
0x58	Master Receive Data NACK	0x70	GC mode Address ACK
0x00	Bus error	0x78	GC mode Arbitration Lost
		0x90	GC mode Data ACK
		0x98	GC mode Data NACK
0xF8	Bus Released **Note:** Status “0xF8” exists in both master/slave modes, and it won’t raise interrupt.		

Table 5.16-1 I^2C Status Code Description

5.16.3.10 I^2C Baud Rate(I2CDIV)

The data baud rate of I^2C is determined by CLK_DIV(I2CDIV[7:0]) register when I^2C is in a Master mode. It is not necessary in Slave mode. In Slave mode, the I^2C will automatically synchronize to clock frequency from I^2C master device.

The data baud rate of I^2C setting is Data Baud Rate of I^2C = (system clock) /(4x(CLK_DIV+1)). If system clock =16 MHz, the CLK_DIV = 40 (28H), the data baud rate of I^2C = 16 MHz/(4X(40+1)) = 97.5 K bits/sec.

5.16.3.11 The I^2C Time-out Counter (I2CTOUT)

There is a 14-bit time-out counter which can be used to deal with the I^2C bus hang-up. If the time-out counter is enabled, when the bus start signal is detected, the counter starts up counting until counter overflows (TIF=1) and requests I^2C interrupt to CPU or there is stop signal being detected. User can also stop counter counting by clearing TOUTEN to 0. When time-out counter is enabled, setting flag I2C_STS and STAINTSTS to high and the falling edge of I^2C bus clock will reset counter and re-start up counting after I2C_STS is cleared or after the falling edge of bus clock. If the I^2C bus hangs up, it causes the STATUS and I2C_STS not to be updated for a period. The 14-bit time-out counter may overflow and acknowledge CPU the I^2C interrupt. Refer to the following Figure for 14-bit time-out counter. User may clear TIF by write 1 to this bit.

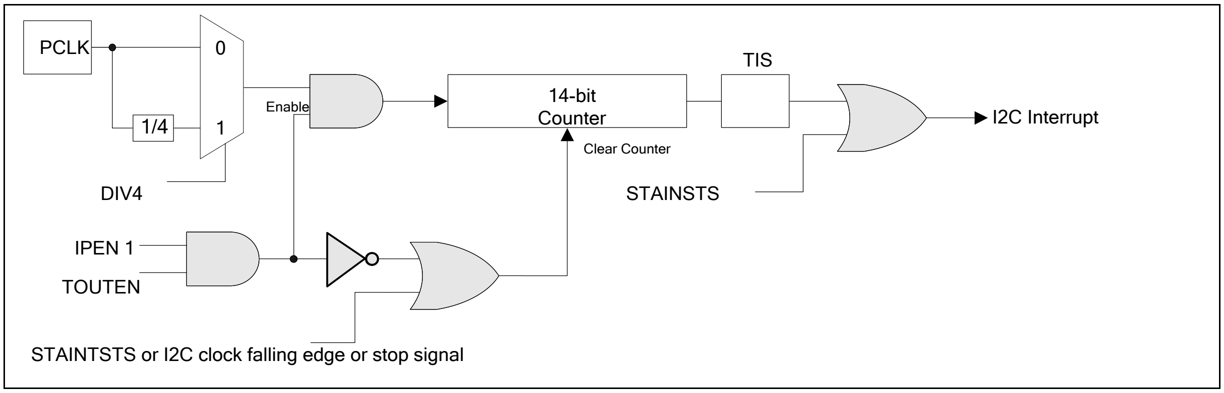


Figure 5.16-5 I^2C Time-out Block Diagram

5.16.3.12 I^2C Data Register (I2CDATA)

This register contains a byte of serial data to be transmitted or a byte which just has been received. The CPU can read from or write to this 8-bit (DATA(I2CDATA[7:0])) directly while it is not in the process of shifting a byte. When I^2C is in a defined state and the serial interrupt flag INTSTS (I2CINTSTS[0]) is set, data in DATA remains stable. While data is being shifted out, data on the bus is simultaneously being shifted in; DATA always contains the last data byte present on the bus.

The acknowledge bit is controlled by the I^2C hardware and cannot be accessed by the CPU. Serial data is shifted through into DATA on the rising edges of serial clock pulses on the SCL line. When a byte has been shifted into DATA, the serial data is available in DATA, and the acknowledge bit (ACK or NACK) is returned by the control logic during the ninth clock pulse. In order to monitor bus status while sending data, the bus data will be shifted to DATA[7:0] when sending DATA[7:0] to bus. In the case of sending data, serial data bits are shifted out from I2CDAT [7:0] on the falling edge of SCL clocks, and is shifted to DATA [7:0] on the rising edge of SCL clocks.

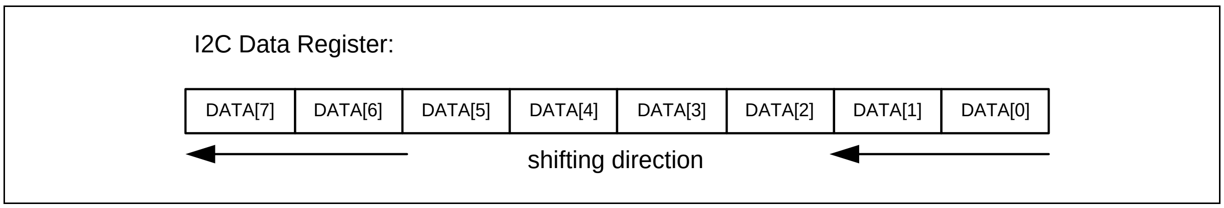


Figure 5.16-6 I^2C Data Shifting Direction

5.16.3.13 Address Registers (I2CSADDR)

The I^2C port is equipped with four slave address registers I2CSADDRn (n=0~1). The contents of the register are irrelevant when I^2C is in Master mode. In Slave mode, the bit field I2CSADDRn[7:1] must be loaded with the chip's own slave address. The I^2C hardware will react if the contents of I2CSADDRn are matched with the received slave address.

The I^2C ports support the "General Call" function. If the GCALL (I2CSADDRn[0]) bit is set, the I^2C port hardware will respond to General Call address (00H). Clearing GCALL bit will disable general call function.

When GCALL bit is set and the I^2C is in Slave mode, it can receive the general call address by 00H after Master send general call address to I^2C bus, then it will follow status of GCALL mode.

5.16.3.14 Slave Address Mask Registers (I2CSAMASK)

I^2C bus controllers support multiple address recognition with two address mask registers. When the bit in the address mask register is set to 1, it means the received corresponding address bit is don't-care. If the bit is set to 0, that means the received corresponding register bit should be exact the same as address register.

5.16.3.15 The I^2C Wake-up Control Register (I2CWKUPCON)

When entering Power-down mode, other I^2C master can wake up our chip by addressing our I^2C device. User must set I2CWKUPCON[WKUPEN] before entering Power-down mode.

5.16.3.16 The I^2C Wake-up Status Register (I2CWKUPSTS)

When system is waken up by other I^2C master device, WKUPIF is set to indicate this event

5.16.3.17 Operation Mode

The on-chip I²C ports support three operation modes, Master, Slave, and General Call Mode.

In a given application, I²C port may operate as a master or as a slave. In Slave mode, the I²C port hardware looks for its own slave address and the general call address. If one of these addresses is detected, and if the slave is willing to receive or transmit data from/to Master(by setting the ACK (I2CON[1]) bit), acknowledge pulse will be transmitted out on the 9th clock, hence an interrupt is requested on both master and slave devices if interrupt is enabled. When the microcontroller wishes to become the bus master, hardware waits until the bus is free before entering Master mode so that a possible slave action is not be interrupted. If bus arbitration is lost in Master mode, I²C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer.

To control the I²C bus transfer in each mode, user needs to set I2CON, I2CDATA registers according to current status code of I2CSTATUS register. In other words, for each I²C bus action, user needs to check current status by I2CSTATUS register, and then set I2CON, I2CDATA registers to take bus action. Finally, check the response status by I2CSTATUS.

The bits, START, STOP and ACK (I2CON[3:1]) are used to control the next state of the I²C hardware after INTSTS flag of I2CINTSTS [0] register is cleared. Upon completion of the new action, a new status code will be updated in I2CSTATUS register and the INTSTS flag (I2CINTSTS[0]) will be set. If the I²C interrupt control bit INTEN (I2CON [7]) is set, appropriate action or software branch of the new status code can be performed in the Interrupt service routine.

The following figure shows the current I²C status code is 0x08, and then set DATA=SLA+W and (STA,STO,SI,AA) = (0,0,1,x) to send the address to I²C bus. If a slave on the bus matches the address and response ACK, the I2CSTATUS will be updated by status code 0x18.

Note: (STA, STO, SI, AA) = (START, STOP,I2C_STS, ACK)

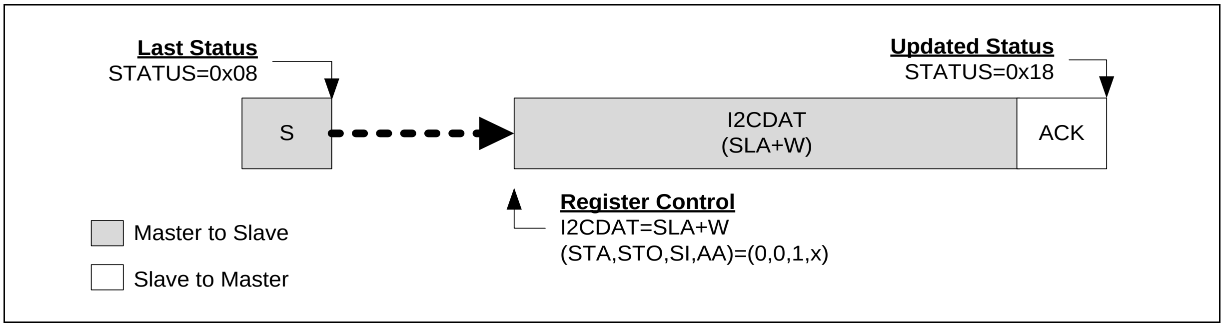


Figure 5.16-7 Control I²C Bus according to Current I²C Status

5.16.3.18 Master Mode

In the following figures, all possible protocols for I²C master are shown. User needs to follow proper path of the flow to implement required I²C protocol.

In other words, user can send a START signal to bus and I²C will be in Master Transmitter mode or Master receiver mode after START signal has been sent successfully and new status code would be 0x08. Followed by START signal, user can send slave address, read/write bit, data and Repeat START, STOP to perform I²C protocol.

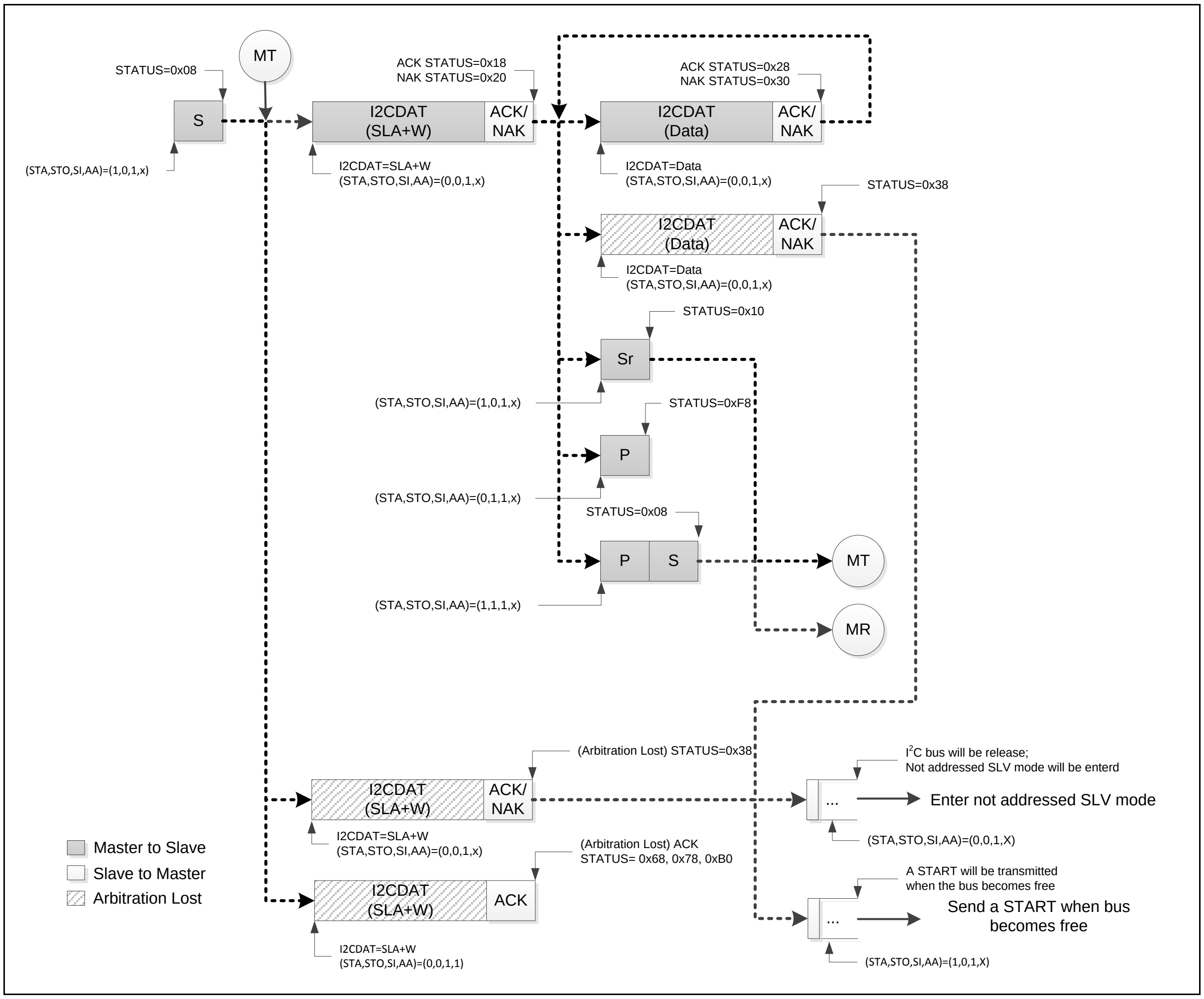


Figure 5.16-8 Master Transmitter Mode Control Flow

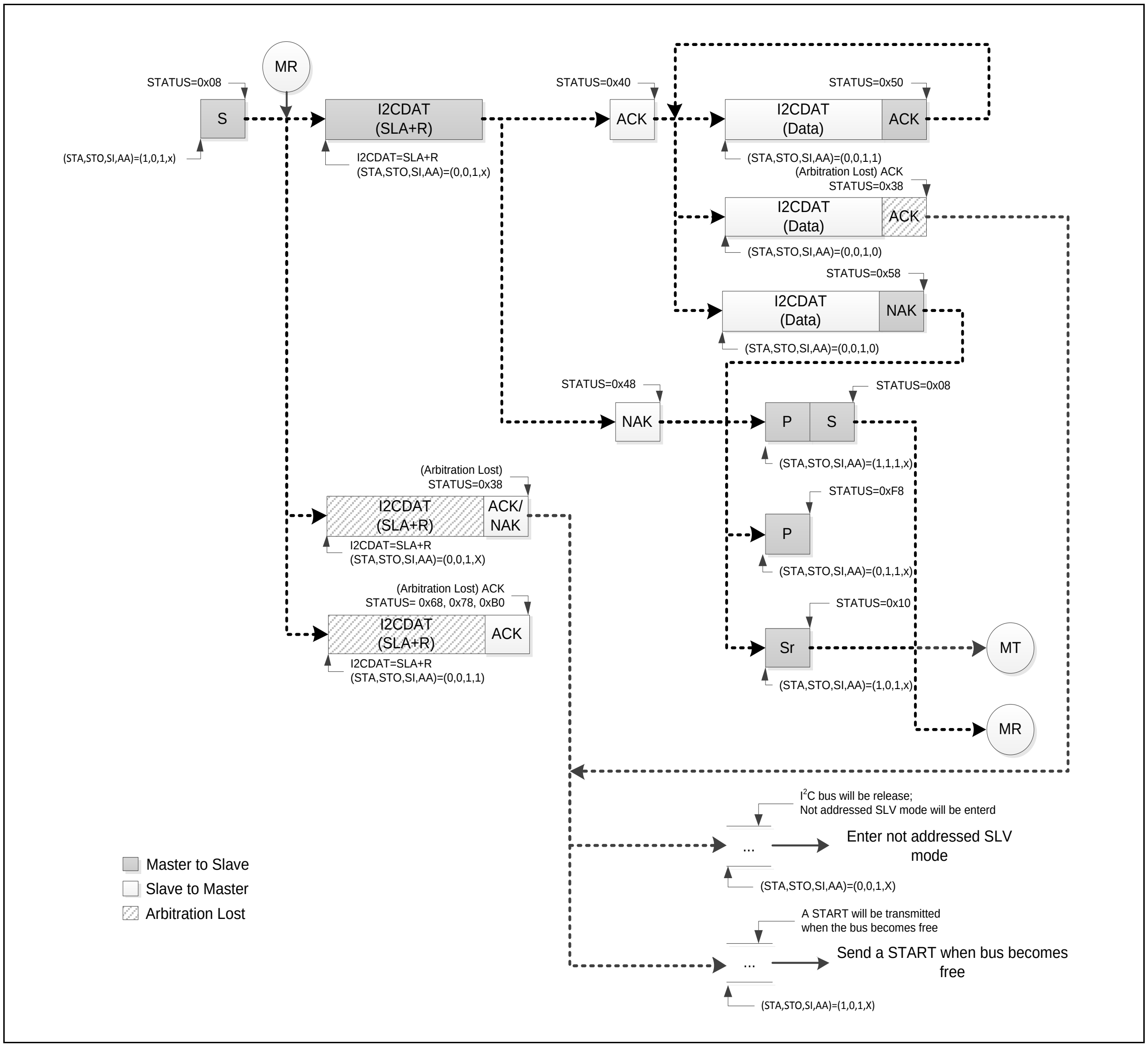


Figure 5.16-9 Master Receiver Mode Control Flow

If the I²C is in Master mode and gets arbitration lost, the status code will be 0x38. In status 0x38, user may set (START, STOP, I2C_STS, ACK) = (1, 0, 1, X) to send START to re-start Master operation when bus become free. Otherwise, user may set (START, STOP, I2C_STS, ACK) = (0, 0, 1, X) to release I²C bus and enter not addressed Slave mode.

Note: (STA, STO, SI, AA) = (START, STOP,I2C_STS, ACK)

5.16.3.19 Slave Mode

When reset default, I²C is not addressed and will not recognize the address on I²C bus. User can set slave address by I2CSADDRx and set (START, STOP, I2C_STS, ACK) = (0, 0, 1, 1) to let I²C recognize the address sent by master. The follow figure shows all the possible flow for I²C in Slave mode. Users need to follow a proper flow to implement their own I²C protocol.

If bus arbitration is lost in Master mode, I²C port switches to Slave mode immediately and can detect its own slave address in the same serial transfer. If the detected address is SLA+W (Master want to write data to Slave) after arbitration lost, the status code is 0x68. If the detected address is SLA+R

(Master want to read data from Slave) after arbitration lost, the status code is 0xB0.

Note: During I^{2}C communication, the SCL clock will be released when writing ‘1’ to clear INTSTS flag in Slave mode.

Note: (STA, STO, SI, AA) = (START, STOP, I2C_STS, ACK)

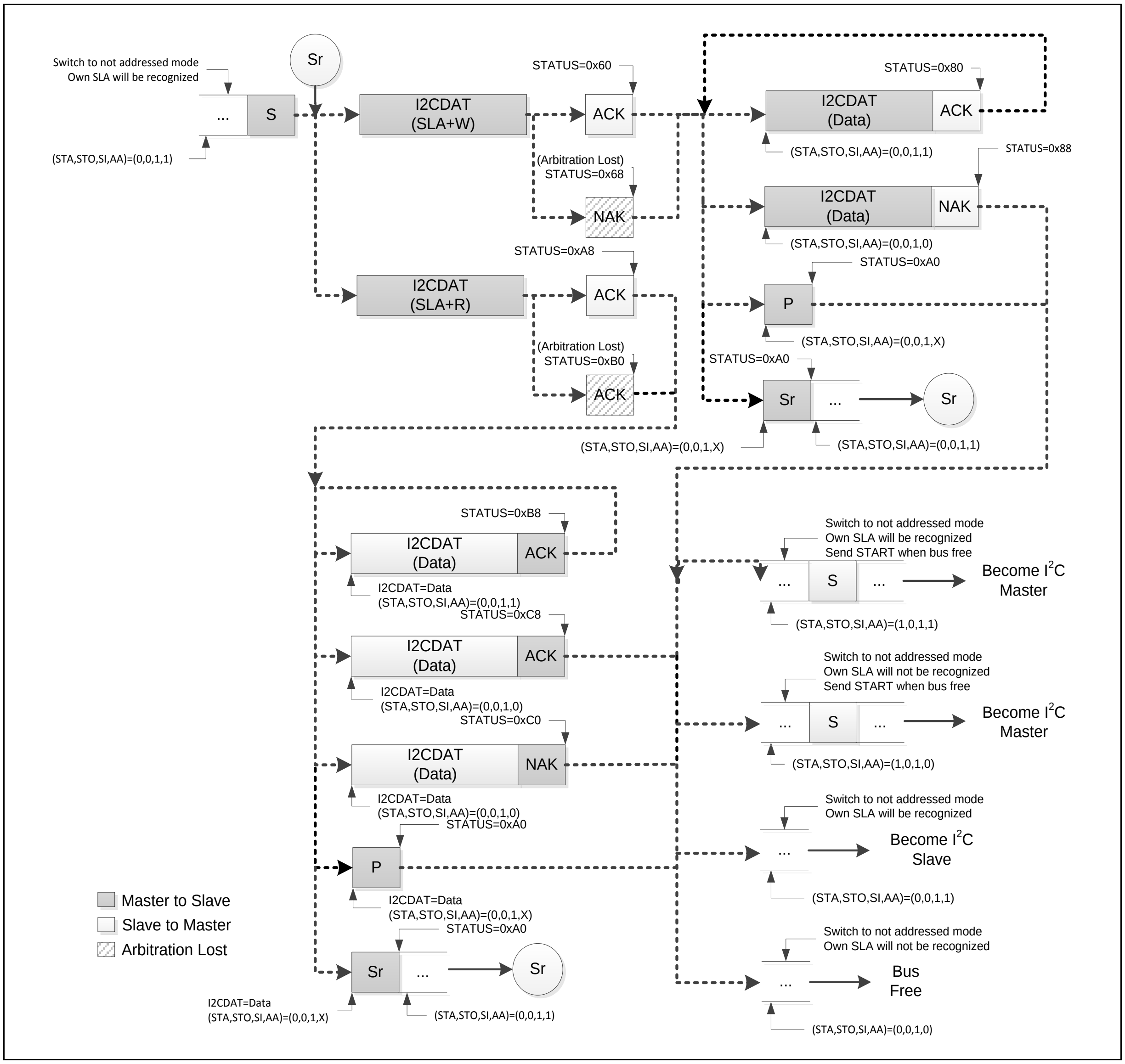


Figure 5.16-10 Slave Mode Control Flow

If I^{2}C is still receiving data in addressed Slave mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0x88 as shown in the above figure when getting 0xA0 status.

If I^{2}C is still transmitting data in addressed Slave mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0xC8 as shown in the above figure when getting 0xA0 status.

Note: After slave gets status of 0x88, 0xC8, 0xC0 and 0xA0, slave can switch to not address mode and own SLA will not be recognized. If entering this status, slave will not receive any I^{2}C signal or address from master. At this status, I^2C should be reset to leave this status.

5.16.3.20 General Call (GC) Mode

If the GCALL (I2CSADDRn [0]) bit is set to 1, the I^2C port hardware will respond to General Call address (00H). User can clear GCALL bit to disable general call function. When the GCALL bit is set and the I^2C is in Slave mode, it can receive the general call address by 0x00 after master send general call address to I^2C bus, then it will follow status of GCALL mode.

Note: (STA, STO, SI, AA) = (START, STOP, I2C_STS, ACK)

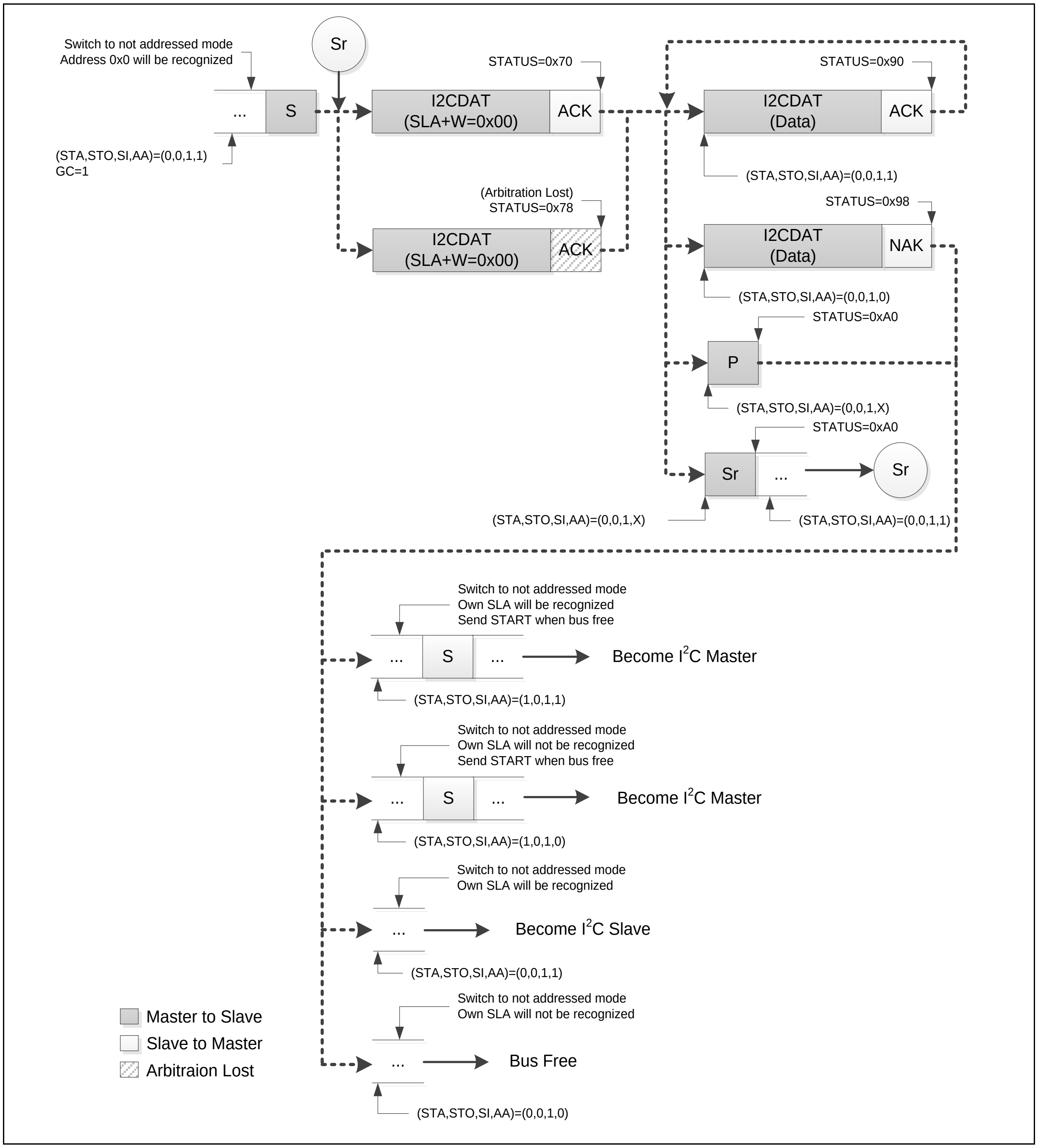


Figure 5.16-11 GC Mode

If I^2C is still receiving data in GCALL mode but got a STOP or Repeat START, the status code will be 0xA0. User could follow the action for status code 0x98 in the above figure when getting 0xA0 status.

Note: After slave gets status of 0x98 and 0xA0, slave can switch to not address mode and own SLA will not be recognized. If entering this status, slave will not receive any I^2C signal or address from master. At this time, I^2C controller should be reset to leave this status.

5.16.3.21 Multi-Master

In some applications, there are two or more masters on the same I^2C bus to access slaves, and the masters may transmit data simultaneously. The I^2C supports multi-master by including collision detection and arbitration to prevent data corruption.

- When I2CSTATUS = 0x38, an “Arbitration Lost” is received. Arbitration lost event maybe occur during the send START bit, data bits or STOP bit. User could set (START, STOP, I2C_STS, ACK) = (1, 0, 1, X) to send START again when bus free, or set (START, STOP,I2C_STS, ACK) = (0, 0, 1, X) to send STOP to back to not addressed Slave mode.
- When I2CSTATUS = 0x00, a “Bus Error” is received. To recover I^2C bus from a bus error, STO should be set and SI should be cleared, and then STO is cleared to release bus.
  - ◆ Set (START, STOP,I2C_STS, ACK) = (0, 1, 1, X) to stop current transfer
  - ◆ Set (START, STOP, I2C_STS, ACK) = (0, 0, 1, X) to release bus

5.16.4 Register and Memory Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
I^2C Base Address: I2Cx_BA = 0x4002_0000 + 0x100000 * x x = 0,1				
I2CCON	I2Cx_BA+0x00	R/W	I^2C Control Register	0x0000_0000
I2CINTSTS	I2Cx_BA+0x04	R/W	I^2C Interrupt Status Register	0x0000_0000
I2CSTATUS	I2Cx_BA+0x08	R	I^2C Status Register	0x0000_00F8
I2CDIV	I2Cx_BA+0x0C	R/W	I^2C clock divided Register	0x0000_0000
I2CTOUT	I2Cx_BA+0x10	R/W	I^2C Time-out control Register	0x0000_0000
I2CDATA	I2Cx_BA+0x14	R/W	I^2C DATA Register	0x0000_0000
I2CSADDR0	I2Cx_BA+0x18	R/W	I^2C Slave address Register0	0x0000_0000
I2CSADDR1	I2Cx_BA+0x1C	R/W	I^2C Slave address Register1	0x0000_0000
I2CSAMASK0	I2Cx_BA+0x28	R/W	I^2C Slave address Mask Register0	0x0000_0000
I2CSAMASK1	I2Cx_BA+0x2C	R/W	I^2C Slave address Mask Register1	0x0000_0000
I2CWKUPCON	I2Cx_BA+0x3C	R/W	I^2C Wake-up Control Register	0x0000_0000
I2CWKUPSTS	I2Cx_BA+0x40	R	I^2C Wake-up Status Register	0x0000_0000

5.16.5 Register Description

I²C Control Register (I2CCON)

Register	Offset	R/W	Description	Reset Value
I2CCON	I2C0_BA+0x00 I2C1_BA+0x00	R/W	I²C Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTEN	Reserved		I2C_STS	START	STOP	ACK	IPEN

Bits	Description	
[31:8]	**Reserved**	**Reserved**
[7]	**INTEN**	**Interrupt Enable.** 1 = I²C interrupt Enabled. 0 = I²C interrupt Disabled.
[6:5]	**Reserved**	**Reserved**
[4]	**I2C_STS**	**I²C Status.** When a new state is present in the I2CSTATUS register, this bit will be set automatically, and if the INTEN bit is set, the I²C interrupt is requested. It must be cleared by software by writing one to this bit and the I²C protocol function will go ahead until the STOP is active or the IPEN is disabled 1 = I²C's Status active 0 = I²C's Status disabled and the I²C protocol function will go ahead.
[3]	**START**	**I²C START Command** Setting this bit to 1 to enter Master mode, the device sends a START or repeat START condition to bus when the bus is free and it will be cleared to 0 after the START command is active and the STATUS has been updated. 1 = Sends a START or repeat START condition to bus. 0 = After START or repeat START is active.
[2]	**STOP**	**I²C STOP Control Bit.** In Master mode, set this bit to 1 to transmit a STOP condition to bus then the controller will check the bus condition if a STOP condition is detected and this bit will be cleared by hardware automatically. In Slave mode, set this bit to 1 to reset the controller to the defined "not addressed" Slave mode. This means it is NO LONGER in the slave receiver mode to receive data from the

Bits	Description	
		master transmit device. 1 = Sends a STOP condition to bus in Master mode or reset the controller to “not addressed” in Slave mode. 0 = Will be cleared by hardware automatically if a STOP condition is detected.
[1]	**ACK**	**Assert Acknowledge Control Bit** 1 = When this bit is set to 1 prior to address or data received, an acknowledged will be returned during the acknowledge clock pulse on the SCL line when a. A slave is acknowledging the address sent from master b. The receiver devices are acknowledging the data sent by transmitter. 0 =: When this bit is set to 0 prior to address or data received, a Not acknowledged (high level to SDA) will be returned during the acknowledge clock pulse.
[0]	**IPEN**	**I^2C Function Enable** When this bit is set to 1, the I^2C serial function is enabled. 1 = I^2C function Enabled. 0 = I^2C function Disabled.

I²C Interrupt Status Register (I2CINTSTS)

Register	Offset	R/W	Description	Reset Value
I2CINTSTS	I2C0_BA+0x04 I2C1_BA+0x04	R/W	I²C Interrupt Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**TIF**	**INTSTS**

Bits	Description	
[31:2]	**Reserved**	**Reserved**
[1]	**TIF**	**Time-out Status** 1 = Time-Out flag active and it is set by hardware. It can introllerrupt CPU when INTEN bit is set. 0 = No Time-out flag. Software can cleat this flag.
[0]	**INTSTS**	**I²S STATUS's Interrupt Status** 1 = New state is presented in the I2CSTATUS. Software can write 1 to cleat this bit. 0 = No bus event occurred.

I^{2}C Status Register (I2CSTATUS)

Register	Offset	R/W	Description	Reset Value
I2CSTATUS	I2C0_BA+0x08 I2C1_BA+0x08	R	I^{2}C Status Register	0x0000_00F8

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
STATUS							

Bits	Description	
[31:8]	**Reserved**	**Reserved**
[7:0]	**STATUS**	**I^{2}C Status Register** Indicates the current status code of the bus information. The detail information about the status is described in the sections of I^{2}C protocol register and operation mode.

I²C Baud Rate Control Register (I2CDIV)

Register	Offset	R/W	Description	Reset Value
I2CDIV	I2C0_BA+0x0C I2C1_BA+0x0C	R/W	I²C clock divided Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	22	21	20	19	18	17	16
Reserved							
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
CLK_DIV							

Bits	Description	
[31:8]	**Reserved**	**Reserved**
[7:0]	**CLK_DIV**	**I2C Clock Divider Conrolltrol Register** The I²C clock rate bits: Data Baud Rate of I²C = PCLK /(4 x (CLK_DIV + 1)) **Note:** the minimum value of CLK_DIV is 4.

I^2C Time-out Counter Register (I2CTOUT)

Register	Offset	R/W	Description	Reset Value
I2CTOUT	I2C0_BA+0x10 I2C1_BA+0x10	R/W	I^2C Time-out control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						DIV4	TOUTEN

Bits	Description	
[31:2]	**Reserved**	**Reserved**
[1]	**DIV4**	**Time-Out Counter Input Clock Divider by 4** 1 = Enabled 0 = Disabled When this bit is set enabled, the Time-Out period is prolonging 4 times.
[0]	**TOUTEN**	**Time-out Counter Enable/Disable** 1 = Enabled 0 = Disabled When set this bit to enable, the 14 bits time-out counter will start counting when STAINTSTS is cleared. Setting flag STAINTSTS to high or the falling edge of I^2C clock or stop signal will reset counter and re-start up counting after STAINTSTS is cleared.

I²C Data Register (I2CDATA)

Register	Offset	R/W	Description	Reset Value
I2CDATA	I2C0_BA+0x14 I2C1_BA+0x14	R/W	I²C DATA Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DATA							

Bits	Description	
[31:8]	**Reserved**	**Reserved**
[7:0]	**DATA**	**I²C Data Register** The DATA contains a byte of serial data to be transmitted or a byte which has just been received. **Note:** Refer to Data register section for more detail information.

I²C Slave Address Register (I2CSADDRx)

Register	Offset	R/W	Description	Reset Value
I2CSADDR0	I2C0_BA+0x18 I2C1_BA+0x18	R/W	I²C Slave address Register0	0x0000_0000
I2CSADDR1	I2C0_BA+0x1C I2C1_BA+0x1C	R/W	I²C Slave address Register1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SADDR							GCALL

Bits	Description	
[31:8]	**Reserved**	**Reserved**
[7:1]	**SADDR**	**I²C Salve Address Register** The content of this register is irrelevant when the device is in Master mode. In the Slave mode, the seven most significant bits must be loaded with the device's own address. The device will react if either of the address is matched.
[0]	**GCALL**	**General Call Function** 1 = General Call Function Enabled. 0 = General Call Function Disabled. **Note:** Refer to Address Register section for more detail information..

SLAVE ADDRESS MASK REGISTER (I2CAMSKx)

Register	Offset	R/W	Description	Reset Value
I2CSAMASK0	I2C0_BA+0x28 I2C1_BA+0x28	R/W	I^2C Slave address Mask Register0	0x0000_0000
I2CSAMASK1	I2C0_BA+0x2C I2C1_BA+0x2C	R/W	I^2C Slave address Mask Register1	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SAMASK							Reserved

Bits	Description	
[31:8]	**Reserved**	**Reserved**
[7:1]	**SAMASK**	**I^2C Slave Address Mask Register** 1 = Mask enable (the received corresponding address bit is don't care.) 0 = Mask disable (the received corresponding register bit should be exact the same as address register.)
[0]	**Reserved**	**Reserved**

I^2C Wake-up Control Register (I2WKUPCON)

Register	Offset	R/W	Description	Reset Value
I2CWKUPCON	I2C0_BA+0x3C I2C1_BA+0x3C	R/W	I^2C Wake-up Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**WKUPEN**

Bits	Description	
[31:1]	**Reserved**	**Reserved**
[0]	**WKUPEN**	**I^2C Wake-up Function Enable** 1 = I^2C wake-up function Enabled. 0 = I^2C wake-up function Disabled.

I²C Wake-up StatusRegister (I2CWKUPSTS)

Register	Offset	R/W	Description	Reset Value
I2CWKUPSTS	I2C0_BA+0x40 I2C1_BA+0x40	R	I²C Wake-up Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved							**WKUPIF**

Bits	Description	
[31:1]	**Reserved**	**Reserved**
[0]	**WKUPIF**	**Wake-up Interrupt Flag** 1 = Wake-up flag active. 0 = Wake-up flag inactive. Software can write 1 to clear this flag

5.17 I^2S

5.17.1 Overview

The audio controller consists of I^2S protocol to interface with external audio CODEC. Two 8 word deep FIFO for receiving path and transmitting path respectively and is capable of handling 8 ~ 32 bit word sizes. PDMA controller handles the data movement between FIFO and memory.

5.17.2 Features

- I^2S can operate as either master or Slave mode.
- Capable of handling 8, 16, 24 and 32 bits word sizes.
- Mono and stereo of audio data are supported.
- I^2S and MSB justified data format are supported.
- Two FIFO data buffers (each 32 bits) are provided, one is for transmitting and the other is for receiving.
- Generate interrupt when buffer levels cross a programmable boundary.
- Two PDMA channels request, one is for transmitting and the other is for receiving.

5.17.3 Block Diagram

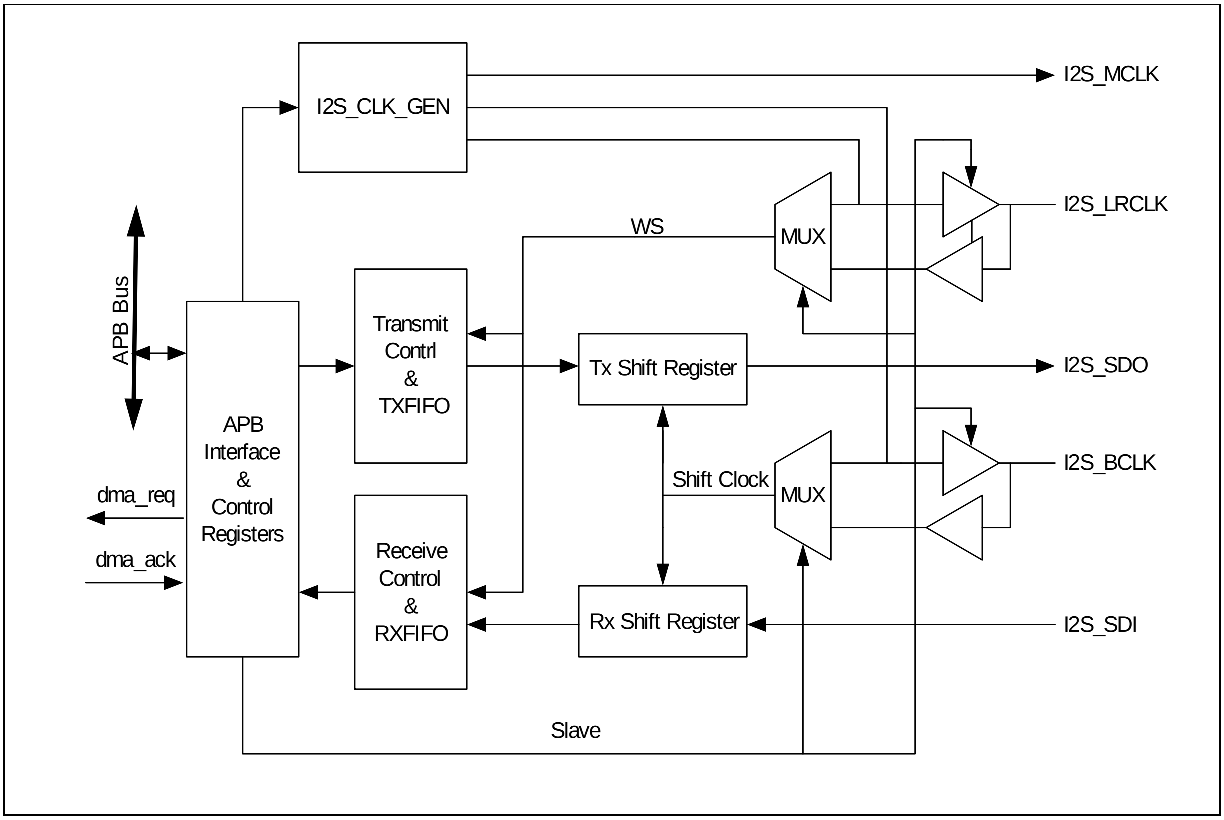


Figure 5.17-1 I^2S Controller Block Diagram

5.17.4 Functional Description

5.17.4.1 I²S Operation

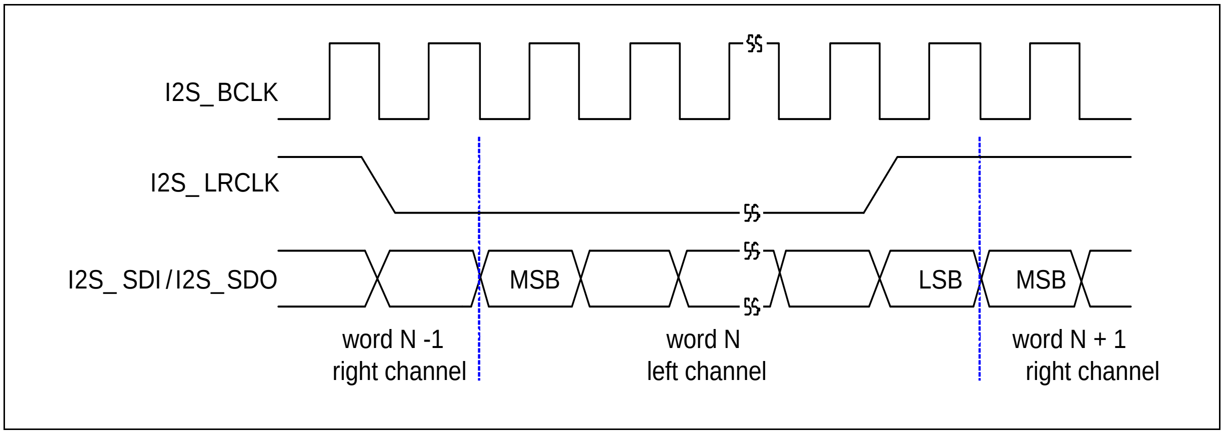


Figure 5.17-2 I²S bus timing diagram (Format = 0)

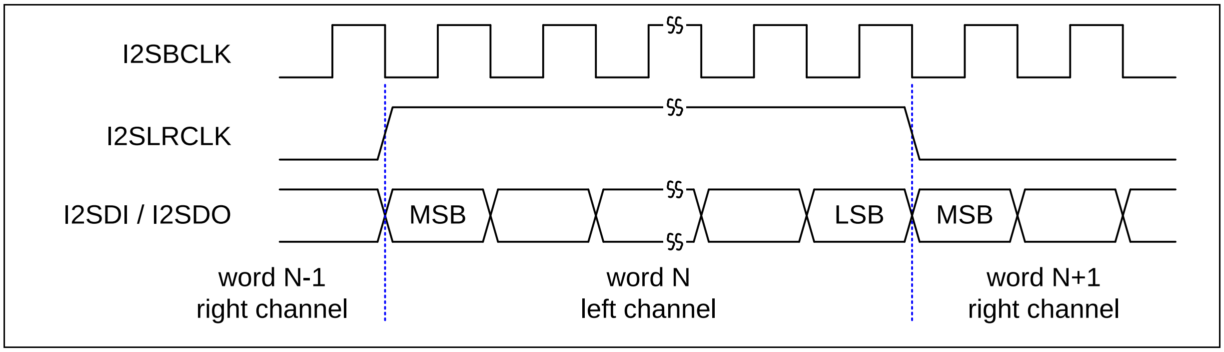


Figure 5.17-3 MSB Justified Timing Diagram (Format = 1)

5.17.4.2 I²S FIFO

The data structures of FIFO of 8/16/32 bits are described as follows.

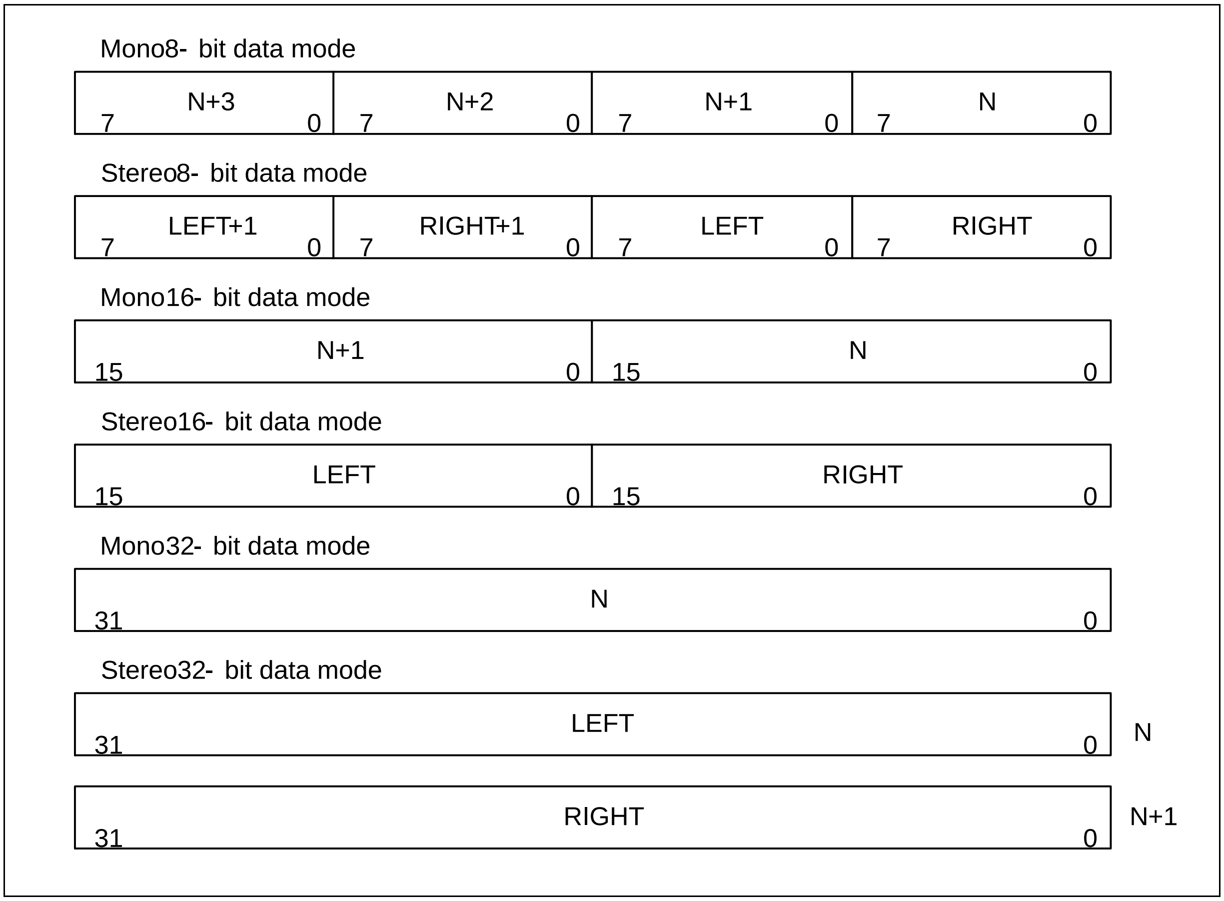


Figure 5.17-4 FIFO Contents for Various I^2S Modes

5.17.5 Register and Memory Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
I^2S Base Address: **I2S_BA = 0x401A_0000**				
I2S_CTRL	I2S_BA+0x00	R/W	I^2S Control Register	0x0000_0000
I2S_CLKDIV	I2S_BA+0x04	R/W	I^2S Clock Divider Register	0x0000_0000
I2S_INTEN	I2S_BA+0x08	R/W	I^2S Interrupt Enable Register	0x0000_0000
I2S_STATUS	I2S_BA+0x0C	R/W	I^2S Status Register	0x0014_1000
I2S_TXFIFO	I2S_BA+0x10	W	I^2S Transmit FIFO Register	0x0000_0000
I2S_RXFIFO	I2S_BA+0x14	R	I^2S Receive FIFO Register	0x0000_0000

5.17.6 Register Description

I²S Control Register (I2S_CTRL)

Register	Offset	R/W	Description	Reset Value
I2S_CTRL	I2S_BA+0x00	R/W	I²S Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RXLCH	Reserved	RXDMA	TXDMA	CLK_RXFIFO	CLR_TXFIFO	LCHZCEN	RCHZCEN
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
MCLKEN	RXTH			TXTH			SLAVE
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
FORMAT	MONO	WORDWIDTH		MUTE	RXEN	TXEN	I2SEN

Bits	Description	
[31:24]	**Reserved**	**Reserved**
[23]	**RXLCH**	**Receive Left Channel Enable** When monaural format is selected (MONO = 1), I²S will receive right channel data if RXLCH is set to 0, and receive left channel data if RXLCH is set to 1. 1 = Receives left channel data when monaural format is selected. 0 = Receives right channel data when monaural format is selected.
[22]	**Reserved**	**Reserved**
[21]	**RXDMA**	**Enable Receive DMA** When RX DMA is enabled, I²S requests PDMA to transfer data from receiving FIFO to memory if FIFO is not empty. 1 = RX DMA Enabled. 0 = RX DMA Disabled.
[20]	**TXDMA**	**Enable Transmit DMA** When TX DMA is enabled, I²S requests PDMA to transfer data from memory to transmitting FIFO if FIFO is not full 1 = TX DMA Enabled. 0 = TX DMA Disabled.
[19]	**CLR_RXFIFO**	**Clear Receiving FIFO** Write "1" to clear receiving FIFO, internal pointer is reset to FIFO start point, and RX_LEVEL[3:0] returns to zero and receiving FIFO becomes empty. This bit is cleared by hardware automatically, and read it return zero.
[18]	**CLR_TXFIFO**	**Clear Transmit FIFO** Write "1" to clear transmitting FIFO, internal pointer is reset to FIFO start point, TX_LEVEL[3:0] returns to zero and transmitting FIFO becomes empty but data in transmit

		FIFO is not changed. This bit is cleared by hardware automatically, read it to return zero.
[17]	**LCHZCEN**	**Left Channel Zero Cross Detect Enable** If this bit is set to "1", when left channel data sign bit is changed or next shift data bits are all zero then LZCF flag in I2S_STATUS register is set to "1". It works on transmitting mode only. 1 = Left channel zero cross detection Enabled 0 = Left channel zero cross detection Disabled
[16]	**RCHZCEN**	**Right Channel Zero Cross Detect Enable** If this bit is set to "1", when right channel data sign bit is changed or next shift data bits are all zero then RZCF flag in I2S_STATUS register is set to "1". It works on transmitting mode only. 1 = Right channel zero cross detection Enabled. 0 = Right channel zero cross detection Disabled.
[15]	**MCLKEN**	**Master Clock Enable** Enable master MCLK timing output to the external audio codec device. The output frequency is according to MCLK_DIV[2:0] in the I2S_CLKDIV register. 1 = Master Clock Enabled. 0 = Master Clock Disabled.
[14:12]	**RXTH**	**Receiving FIFO Threshold Level** When received data word(s) in buffer is equal to or higher than threshold level, the RXTHF flag is set. <table><tr><th>RXTH</th><th>Description</th></tr><tr><td>000</td><td>1 word data in receiving FIFO</td></tr><tr><td>001</td><td>2 word data in receiving FIFO</td></tr><tr><td>010</td><td>3 word data in receiving FIFO</td></tr><tr><td>011</td><td>4 word data in receiving FIFO</td></tr><tr><td>100</td><td>5 word data in receiving FIFO</td></tr><tr><td>101</td><td>6 word data in receiving FIFO</td></tr><tr><td>110</td><td>7 word data in receiving FIFO</td></tr><tr><td>111</td><td>8 word data in receiving FIFO</td></tr></table>
[11:9]	**TXTH**	**Transmit FIFO Threshold Level** If remain data word (32 bits) in transmitting FIFO is the same or less than threshold level then TXTHF flag is set. <table><tr><th>TXTH</th><th>Description</th></tr><tr><td>000</td><td>1 word data in receiving FIFO</td></tr><tr><td>001</td><td>2 word data in receiving FIFO</td></tr><tr><td>010</td><td>3 word data in receiving FIFO</td></tr><tr><td>011</td><td>4 word data in receiving FIFO</td></tr><tr><td>100</td><td>5 word data in receiving FIFO</td></tr><tr><td>101</td><td>6 word data in receiving FIFO</td></tr></table>

		110 — 7 word data in receiving FIFO 111 — 8 word data in receiving FIFO
[8]	**SLAVE**	**Slave Mode** I^2S can operate as master or Slave mode. For Master mode, I2S_BCLK and I2S_LRCLK pins are output mode and also outputs I2S_BCLK and I2S_LRCLK signals to the audio CODEC. When act as Slave mode, I2S_BCLK and I2S_LRCLK pins are input mode and I2S_BCLK and I2S_LRCLK signals are received from the outer audio CODEC chip. 1 = Slave mode 0 = Master mode
[7]	**FORMAT**	**Data Format** 1 = MSB justified data format 0 = I^2S data format
[6]	**MONO**	**Monaural Data** 1 = Data is monaural format and gets the right channel data from I^2S bus when this mode is enabled. 0 = Data is stereo format
[5:4]	**WORDWIDTH**	**Word width** **WORDWIDTH** — **Description** 00 — data is 8-bit 01 — data is 16-bit 10 — data is 24-bit 11 — data is 32-bit
[3]	**MUTE**	**Transmitting Mute Enable** 1 = Transmit ‘0’ to channel. 0 = Transmit data in buffer to channel.
[2]	**RXEN**	**Receive Enable** 1 = Data receiving Enabled. 0 = Data receiving Disabled.
[1]	**TXEN**	**Transmit Enable** 1 = Data transmitting Enabled. 0 = Data transmitting Disabled.
[0]	**I2SEN**	**I^2S Controller Enable** 1 = Enabled. 0 = Disabled.

I^2S Clock Divider Register (I2S_CLKDIV)

Register	Offset	R/W	Description	Reset Value
I2S_CLKDIV	I2S_BA+0x04	R/W	I^2S Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
BCLK_DIV							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**MCLK_DIV**		

Bits	Description	
[31:16]	**Reserved**	**Reserved**
[15:8]	**BCLK_DIV**	**Bit Clock Divider** If I^2S is operated in Master mode, bit clock is provided by this chip. Software can program these bits to generate sampling rate clock frequency. BCLK = I2SCLK /(2x(BCLK_DIV + 1))
[7:3]	**Reserved**	**Reserved**
[2:0]	**MCLK_DIV**	**Master Clock Divider** If the external crystal frequency is (2xMCLK_DIV)*256fs then software can program these bits to generate 256fs clock frequency to audio CODEC chip. If MCLK_DIV is set to "0", MCLK is the same as external clock input. For example, sampling rate is 48 kHz and the external crystal clock is 12.288 MHz, set MCLK_DIV=0. MCLK = I2SCLK/(2x(MCLK_DIV))

I^{2}S Interrupt Enable Register (I2S_INTEN)

Register	Offset	R/W	Description	Reset Value
I2S_INTEN	I2S_BA+0x08	R/W	I^{2}S Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			**LZCIE**	**RZCIE**	**TXTHIE**	**TXOVFIE**	**TXUDFIE**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					**RXTHIE**	**RXOVFIE**	**RXUDFIE**

Bits	Description	
[31:13]	**Reserved**	**Reserved**
[12]	**LZCIE**	**Left Channel Zero Cross Interrupt Enable** Interrupt occurs if this bit is set to "1" and left channel is zero crossing. 1 = Interrupt Enabled 0 = Interrupt Disabled
[11]	**RZCIE**	**Right Channel Zero Cross Interrupt Enable** Interrupt occurs if this bit is set to "1" and right channel is zero crossing. 1 = Interrupt Enabled. 0 = Interrupt Disabled.
[10]	**TXTHIE**	**Transmitting FIFO Threshold Level Interrupt Enable** Interrupt occurs if this bit is set to "1" and data words in transmitting FIFO is less than TXTH[2:0]. 1 = Interrupt Enabled. 0 = Interrupt Disabled.
[9]	**TXOVFIE**	**Transmitting FIFO Overflow Interrupt Enable** Interrupt occurs if this bit is set to "1" and transmitting FIFO overflow flag is set to "1" 1 = Interrupt Enabled. 0 = Interrupt Disabled.
[8]	**TXUDFIE**	**Transmitting FIFO Underflow Interrupt Enable** Interrupt occurs if this bit is set to "1" and transmitting FIFO underflow flag is set to "1". 1 = Interrupt Enabled. 0 = Interrupt Disabled.
[7:3]	**Reserved**	**Reserved**

[2]	**RXTHIE**	**Receiving FIFO Threshold Level Interrupt Enable** Interrupt occurs if this bit is set to “1” and data words in receiving FIFO is less than RXTH[2:0]. 1 = Interrupt Enabled. 0 = Interrupt Disabled.
[1]	**RXOVFIE**	**Receiving FIFO Overflow Interrupt Enable** Interrupt occurs if this bit is set to “1” and receiving FIFO overflow flag is set to “1” 1 = Interrupt Enabled. 0 = Interrupt Disabled.
[0]	**RXUDFIE**	**Receiving FIFO Underflow Interrupt Enable** Interrupt occurs if this bit is set to “1” and receiving FIFO underflow flag is set to “1”. 1 = Interrupt Enabled. 0 = Interrupt Disabled.

I^2S Status Register (I2S_STATUS)

Register	Offset	R/W	Description	Reset Value
I2S_STATUS	I2S_BA+0x0C	R/W	I^2S Status Register	0x0014_1000

31	30	29	28	27	26	25	24
TX_LEVEL				RX_LEVEL			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
LZCF	RZCF	TXBUSY	TXEMPTY	TXFULL	TXTHF	TXOVF	TXUDF
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			RXEMPTY	RXFULL	RXTHF	RXOVF	RXUDF
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				RIGHT	I2STXINT	I2SRXINT	I2SINT

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:28]</td><td>TX_LEVEL</td><td>Transmitting FIFO Level
These bits indicate the number of word(s) in the transmitting FIFO
<table><tr><th>TX_LEVEL</th><th>Description</th></tr><tr><td>0000</td><td>No data</td></tr><tr><td>0001</td><td>1 word in the transmitting FIFO</td></tr><tr><td>- - -</td><td>- - -</td></tr><tr><td>1000</td><td>8 words the in transmitting FIFO</td></tr></table></td></tr>
<tr><td>[27:24]</td><td>RX_LEVEL</td><td>Receive FIFO Level
These bits indicate the number of word(s) in the receiving FIFO
<table><tr><th>RX_LEVEL</th><th>Description</th></tr><tr><td>0000</td><td>No data</td></tr><tr><td>0001</td><td>1 word in the transmitting FIFO</td></tr><tr><td>- - -</td><td>- - -</td></tr><tr><td>1000</td><td>8 words the in transmitting FIFO</td></tr></table></td></tr>
<tr><td>[23]</td><td>LZCF</td><td>Left Channel Zero Cross Flag
It indicates the next sample data sign bit of left channel is changed or all data bits are zero.
1 = Left channel zero cross is detected
0 = No zero cross
This bit is cleared by writing 1.</td></tr>
<tr><td>[22]</td><td>RZCF</td><td>Right channel zero cross flag
It indicates the data sign of right channel next sample data is changed or all data bits are zero.
1 = Right channel zero cross is detected</td></tr>
</table>

		0 = No zero cross This bit is cleared by writing 1.
[21]	**TXBUSY**	**Transmitting Busy** This bit is cleared to 0 when all data in the transmitting FIFO and shift buffer is shifted out. Set this bit to [1] when 1^{st} data is loading to shift buffer. 1 = Transmit shift buffer is busy 0 = Transmit shift buffer is empty This bit is read only.
[20]	**TXEMPTY**	**Transmitting FIFO Empty** This bit reflect data word number in the transmitting FIFO is zero 1 = Not empty 0 = Empty This bit is read only.
[19]	**TXFULL**	**Transmitting FIFO Full** This bit reflect data word number in the transmitting FIFO is 8 1 = Not full. 0 = Full. This bit is read only
[18]	**TXTHF**	**Transmitting FIFO Threshold Flag** When data word(s) in the transmitting FIFO is equal to or lower than threshold value set in TXTH[2:0],the TXTHF bit becomes to “1”. It keeps at 1 till TX_LEVEL[3:0] is higher than TXTH[1:0] after software writes data into the TXFIFO register. 1 = Data word(s) in transmitting FIFO is equal to or lower than threshold level 0 = Data word(s) in transmitting FIFO is higher than threshold level This bit is read only
[17]	**TXOVF**	**Transmit FIFO Overflow Flag** Write data to the transmitting FIFO when it is full and this bit will set to “1” 1 = Overflow 0 = No overflow This bit is cleared by writing 1.
[16]	**TXUDF**	**Transmitting FIFO Underflow Flag** When the transmitting FIFO is empty and shift logic hardware read data from the data FIFO causes this set to “1”. 1 = Underflow 0 = No underflow This bit is cleared by writing 1.
[15:13]	**Reserved**	**Reserved**
[12]	**RXEMPTY**	**Receiving FIFO Empty** This bit reflect data word number in the receiving FIFO is zero 1 = Not empty 0 = Empty This bit is read only.

[11]	**RXFULL**	**Receiving FIFO Full** This bit reflect data word number in the receiving FIFO is 8 1 = Not full. 0 = Full. This bit is read only
[10]	**RXTHF**	**Receiving FIFO Threshold Flag** When data word(s) in the receiving FIFO is equal to or higher than threshold value set in RXTH[2:0], the RXTHF bit becomes to "1". It keeps at "1" till RX_LEVEL[3:0] less than RXTH[1:0] after software reads data from the RXFIFO register. 1 = Data word(s) in receiving FIFO is equal to or higher than threshold level 0 = Data word(s) in receiving FIFO is lower than threshold level This bit is read only
[9]	**RXOVF**	**Receiving FIFO Overflow Flag** When the receiving FIFO is full and receiving hardware attempts to write data into receiving FIFO then this bit is set to "1". Data in 1st buffer is overwritten. 1 = Overflow occurred 0 = No overflow occurred This bit is cleared by writing 1.
[8]	**RXUDF**	**Receiving FIFO Underflow Flag** Read the receiving FIFO when it is empty, this bit set to "1" indicate underflow occur. 1 = Underflow occurred 0 = No underflow occurred This bit is cleared by writing 1.
[7:4]	**Reserved**	**Reserved**
[3]	**RIGHT**	**Right Channel** This bit indicates the current transmitting data is belong to right channel 1 = Right channel 0 = Left channel This bit is read only
[2]	**I2STXINT**	**I^{2}S Transmit Interrupt** 1 = Transmit interrupt occurred 0 = No transmit interrupt occurred This bit is read only
[1]	**I2SRXINT**	**I^{2}S Receiving Interrupt** 1 = Receiving interrupt occurred 0 = No receiving interrupt occurred This bit is read only
[0]	**I2SINT**	**I^{2}S Interrupt Flag** 1 = I^{2}S interrupt occurred 0 = No I^{2}S interrupt It is wire-OR of I2STXINT and I2SRXINT bits. This bit is read only.

I^2S Transmit Register (I2S_TXFIFO)

Register	Offset	R/W	Description	Reset Value
I2S_TXFIFO	I2S_BA+0x10	W	I^2S Transmit FIFO Register	0x0000_0000

31	30	29	28	27	26	25	24
TXFIFO							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TXFIFO							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TXFIFO							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TXFIFO							

Bits	Description	
[31:0]	**TXFIFO**	**Transmitting FIFO register** I^2S contains 8 words (8x32-bit) data buffer for data transmitting. Write data to this register in order to prepare data for transmitting. The remaining word number is indicated by TX_LEVEL[3:0] in the I2S_STATUS register. This register is write only.

I^{2}S Receive Register (I2S_RXFIFO)

Register	Offset	R/W	Description	Reset Value
I2S_RXFIFO	I2S_BA+0x14	R	I^{2}S Receive FIFO Register	0x0000_0000

31	30	29	28	27	26	25	24
RXFIFO							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RXFIFO							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RXFIFO							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RXFIFO							

Bits	Description	
[31:0]	**RXFIFO**	**Receiving FIFO Register** I^{2}S contains 8 words (8x32-bit) data buffer for data receiving. Read this register to get data in FIFO. The remaining data word number is indicated by RX_LEVEL[3:0] in the I2S_STATUS register. This register is read only.

5.18 SPI

5.18.1 Overview

The Serial Peripheral Interface (SPI) is a synchronous serial data communication protocol. Devices communicate in Master/Slave mode with 4-wire bi-direction interface. It is used to perform a serial-to-parallel conversion on data received from a peripheral device, and a parallel-to-serial conversion on data transmitted to a peripheral device. The SPI controller can be configured as a master or a slave devicee.

The SPI controller supports wake-up function. When this chip stays in power-down mode, it can be waked up chip by off-chip device.

This controller supports variable serial clock function for special application and 2-bit transfer mode to connect 2 off-chip slave devices at the same time. The SPI controller also supports PDMA function to access the data buffer.

5.18.2 Features

- Supports Master (max. 32 MHz) or Slave (max. 16 MHz) mode operation
- Supports 1 bit and 2 bit transfer mode
- Support Dual IO transfer mode
- Configurable bit length of a transaction from 8 to 32-bit
- Supports MSB first or LSB first transfer sequence
- Two slave select lines supported in Master mode
- Configurable byte or word suspend mode
- Supports byte re-ordering function
- Supports variable serial clock in Master mode
- Provide separate 8-level depth transmit and receive FIFO buffer
- Supports wake-up function
- Supports PDMA transfer
- Supports three wires, no slave select signal, bi-direction interface

5.18.3 Block Diagram

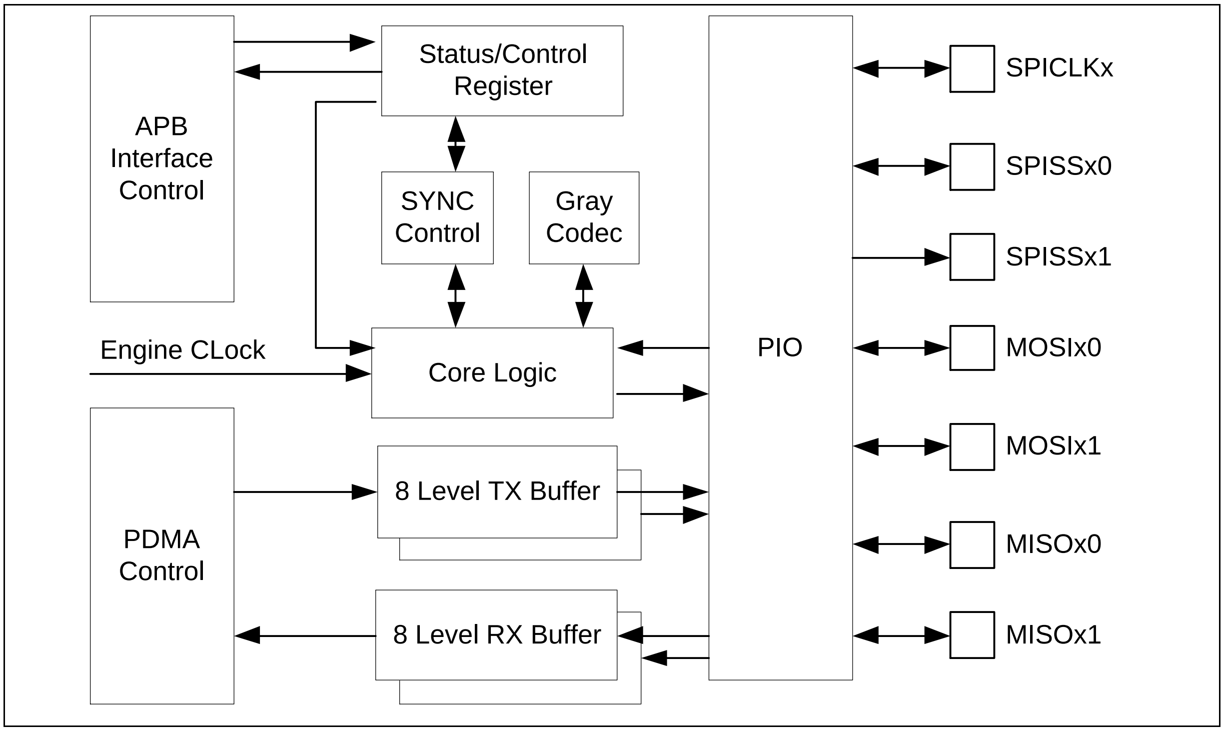


Figure 5.18-1 SPI Block Diagram

Where “x” indicates the number of SPI block. Ex: SPICLK0 'is the 1st SPI block serial clock port.

5.18.4 Functional Description

5.18.4.1 SPI Engine Clock and Serial Clock

SPI controller needs the SPI engine clock to drive the SPI logic unit to perform the data transfer. The SPI engine clock rate is determined by the settings of clock source and clock divisor. The SPIx_S bit of CLKSEL2 register determines the clock source of the SPI engine clock. The clock source can be HCLK or PLL output clock. The DIVIDER setting of SPI_DIVIDER register determines the divisor of the clock rate calculation.

In master mode, if the variable clock function is disabled, the output frequency of the serial clock output pin is equal to the SPI engine clock rate. In slave mode, the SPI serial clock is provided by an off-chip master device. The SPI engine clock rate of slave device must be faster than the serial clock rate of the master device connected together. The frequency of SPI engine clock cannot be faster than the APB clock rate regardless of master mode or slave mode.

5.18.4.2 Master/Slave Mode

This SPI controller can be set as master or Slave mode by setting the SLAVE bit (SPI_CTL[18]) to communicate with the off-chip SPI slave or master device. The application block diagrams in Master and Slave mode are shown below.

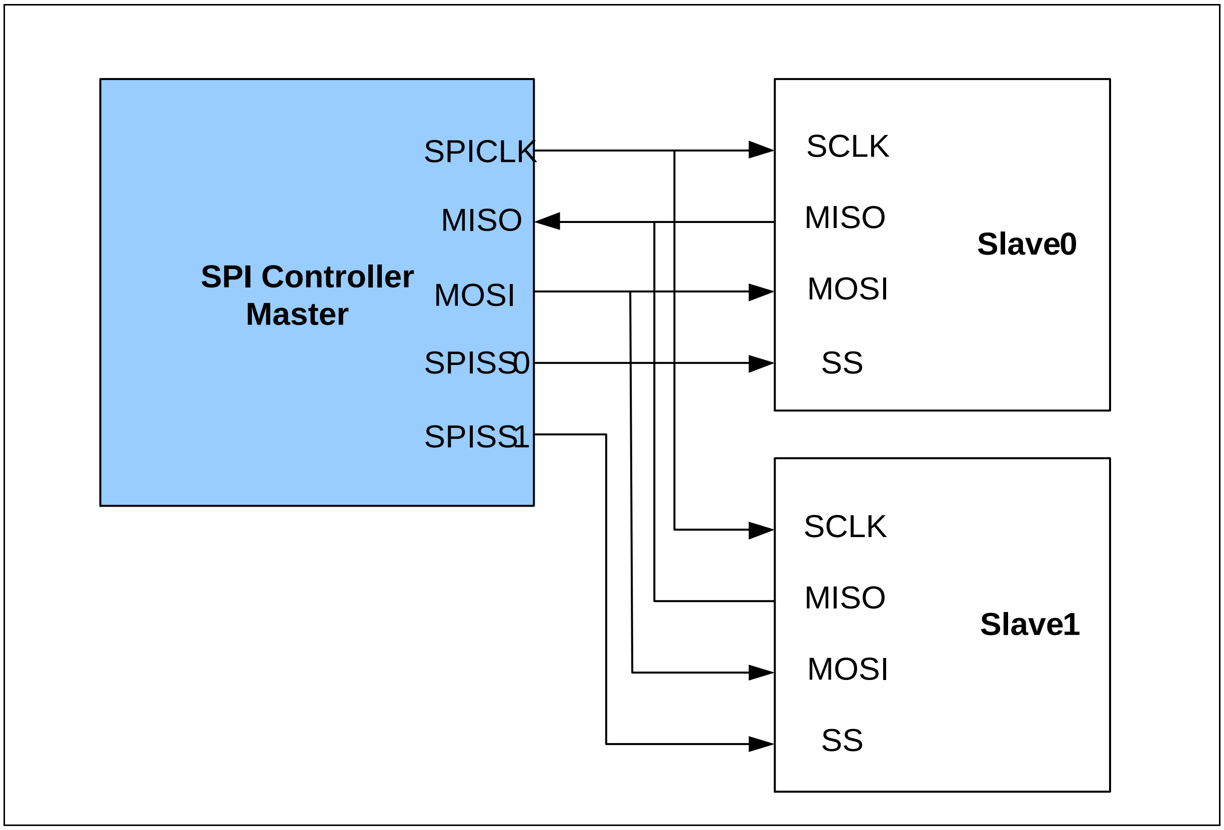


Figure 5.18-2 SPI Master Mode Application Block Diagram

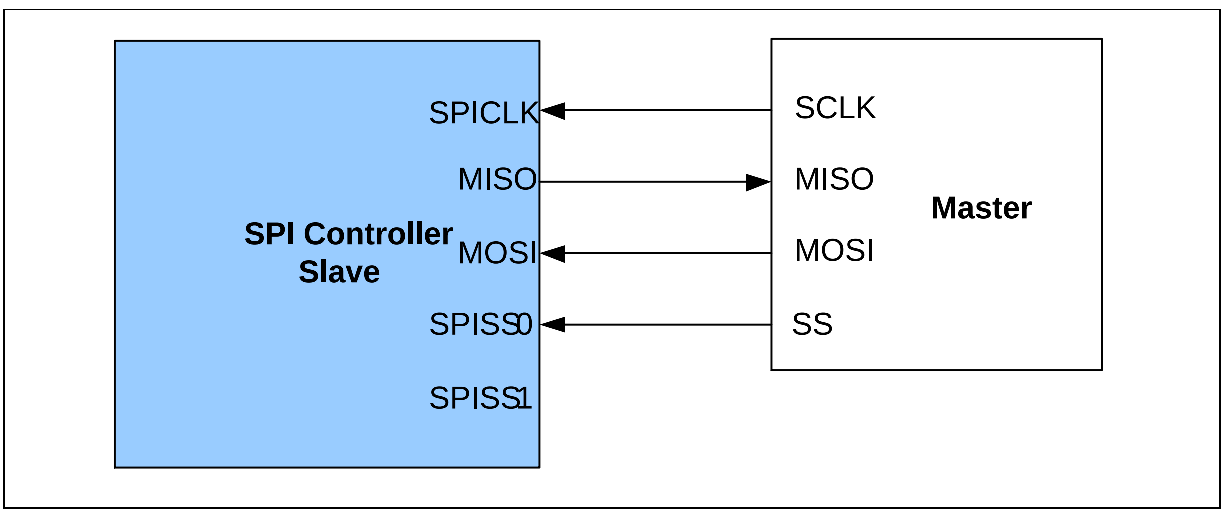


Figure 5.18-3 SPI Slave Mode Application Block Diagram

5.18.4.3 Slave Selection

In Master mode, this SPI controller can drive up to two off-chip slave devices through the slave select output signals SPIxSS0 and SPIxSS1, but it is a time-sharing operation and it can not operate with two slave devices simultaneously.

In Slave mode, the off-chip master device drives the slave select signal from the SPIxSS0 port to this

SPI controller. In Master/Slave mode, the active level of slave select signal can be programmed to low active or high active in SS_LVL bit (SPI_SSR[2]), and the SS_LTRIG bit (SPI_SSR[4]) define the slave select signal SPIxSS0/1 is level trigger or edge trigger. The selection of trigger condition depends on what type of peripheral slave/master device is connected.

In Slave mode, if the SS_LTRIG bit is configured as level trigger, the LTRIG_FLAG bit (SPI_SSR[4]) is used to indicate if the count of received bits among one transaction meets the requirement which define in TX_BIT_LEN.

5.18.4.4 Level-trigger / Edge-trigger

In Slave mode, the slave select signal can be configured as level-trigger or edge-trigger. In edge-trigger, the data transfer starts from an active edge and ends on an inactive edge. If master does not send an inactive edge to slave, the transfer procedure will not be completed and the unit transfer interrupt flag of slave will not be set. In level-trigger, the following two conditions will terminate the transfer procedure and the unit transfer interrupt flag of slave will be set. The first condition is that if the number of transferred bits matches the settings of TX_BIT_LEN, the unit transfer interrupt flag of slave will be set. The second condition, if master set the slave select pin to inactive level during the transfer is in progress, it will force slave device to terminate the current transfer no matter how many bits have been transferred and the unit transfer interrupt flag will be set. User can read the status of LTRIG_FLAG bit to check if the data has been completely transferred.

5.18.4.5 Automatic Slave Select

In Master mode, if the AUTOSS bit (SPI_SSR[3]) is set as 1, the slave select signals will be generated automatically and output to SPIxSS0 and SPIxSS1 ports according to SSR[0] (SPI_SSR[0]) and SSR[1] (SPI_SSR[1]) whether it is enabled or not. It means that the slave select signals, which is enabled in SPI_SSR[1:0] register is asserted by the SPI controller when the SPI data transfer is started by setting the GO_BUSY bit (SPI_CTL[0]) and is de-asserted after the data transfer is finished. If the AUTOSS bit is cleared to 0, the slave select output signals are asserted and de-asserted by manual setting and clearing the related bits in SPI_SSR[1:0] register. The active level of the slave select output signals is specified in SS_LVL bit (SPI_SSR[2]).

In master mode, if the value of SP_CYCLE[3:0] is less than 3 and the AUTOSS is set as 1, the slave select signal will keep at active state between two successive transactions.

In slave mode, to recognize the inactive state of the slave select signal, the inactive period of the slave select signal must be larger than or equal to 6 engine clock periods between two successive transactions.

5.18.4.6 No Slave Select Mode (3-WIRE mode)

When the software sets the NOSLVSEL bit to enable the 3-wire mode, the SPI controller can work with no slave select signal in slave mode. The NOSLVSEL bit only takes effect in slave mode. It only needs three pins, SPICLK, SPI_MISO, and SPI_MOSI, to communicate with a SPI master. The SPISS pin can be configured as a GPIO. When the NOSLVSEL bit is set to 1, the SPI slave will be ready to transmit/receive data after the GO_BUSY bit is set to 1. In 3-wire mode, the SS_LTRIG, SPI_SSR[4], shall be set as 1.

In normal operation, the interrupt flag in SLV_START_INTSTS will be set when the transfer has start and there is also interrupt event when the received data meet the required bits which define in TX_BIT_LEN. If the received bits are less than the requirement and there is no more serial clock input over the time period which is defined by the user in Slave mode with no slave select, the user can set the SLV_ABORT bit to force the current transfer done and then the user can get a transfer done interrupt event.

5.18.4.7 Variable Clock Function

In master mode, if the VARCLK_EN bit (SPI_CTL[23]) is set to 1, the output of serial clock can be programmed as variable frequency pattern. The serial clock period of each cycle depends on the setting of the SPI_VARCLK register. When the variable clock function is enabled, the TX_BIT_LEN setting must be set as 0x10 to configure the data transfer as 16-bit transfer mode. The VARCLK[31] determines the clock period of the first clock cycle. If VARCLK[31] is 0, the first clock cycle depends on the DIVIDER1 setting; if it is 1, the first clock cycle depends on the DIVIDER2 setting. Two successive bits in VARCLK[30:1] defines one clock cycle. The bit field VARCLK[30:29] defines the second clock cycle of SPI serial clock of a transaction, and the bit field VARCLK[28:27] defines the third clock cycle and so on. The VARCLK[0] is unmeaning. The following figure shows the timing relationship among the serial clock (SPICLK), the VARCLK, the DIVIDER1 and the DIVIDER2 registers.

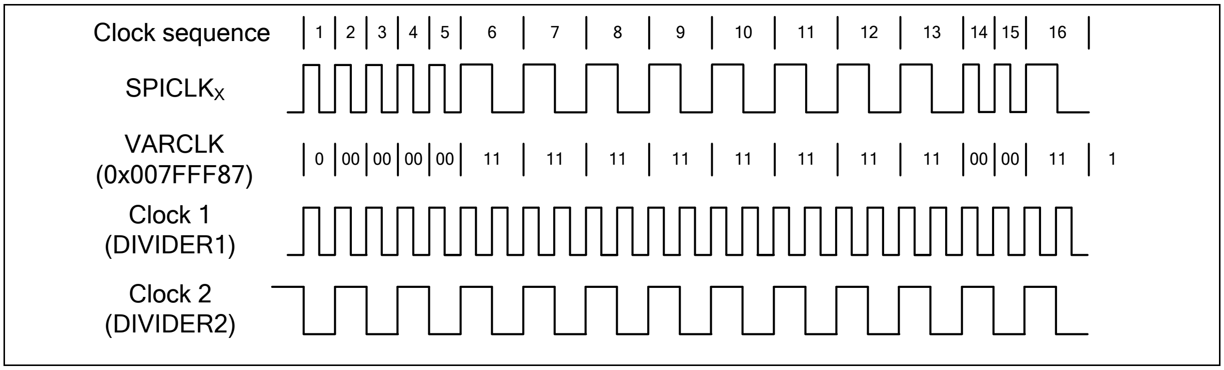


Figure 5.18-4 SPI Variable Clock Frequency

5.18.4.8 Clock Polarity

The CLKP bit (SPI_CTL[11]) defines the serial clock idle state in Master mode. If CLKP = 1, the output SPICLK is idle at high state. If CLKP = 0, it is idle at low state. For variable serial clock, it works in CLKP = 0 only.

5.18.4.9 Transmitting/Receiving Bit Length

The bit length of a transaction word is defined in TX_BIT_LEN bit (SPI_CTL[7:3]). It can be configured up to 32 bits in a transaction word for transmitting and receiving.

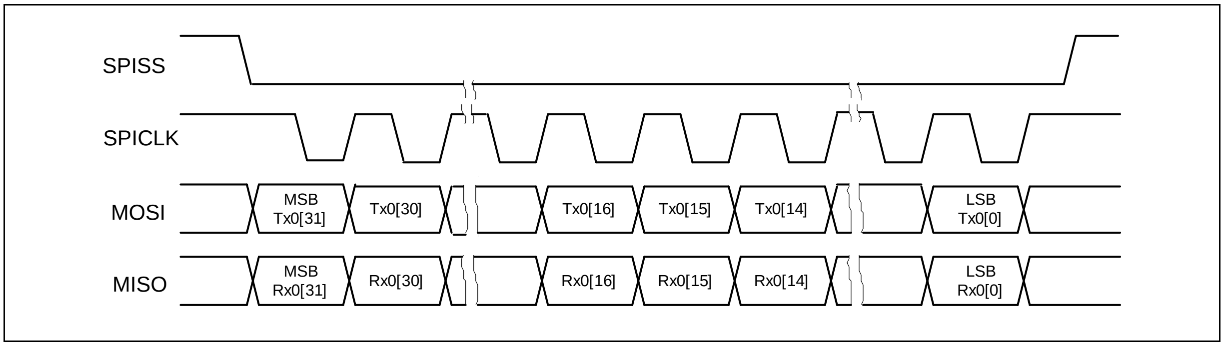


Figure 5.18-5 SPI 32-bit in One Transaction

5.18.4.10 LSB First

The LSB bit (SPI_CTL[10]) defines the bit transfer sequence in a transaction. If set the LSB bit to 1, the transfer sequence is LSB first. The bit 0 will be transferred firstly. If clear the LSB bit to 0, the transfer sequence is MSB first (refer to the figure above).

The TX_NEG bit (SPI_CTL[2]) defines the data transmitted out either at negative edge or at positive edge of serial clock SPICLK.

5.18.4.11 Receive Edge

The RX_NEG bit (SPI_CTL[1]) defines the data received in either at negative edge or at positive edge of serial clock SPICLK. Note that TX_NEG and RX_NEG must be exclusive.

5.18.4.12 Word Suspend

These four bits field of SP_CYCLE (SPI_CTL[15:12]) provide a configurable clock suspend interval 0.5 ~ 15.5 serial clock periods between two successive transaction words in Master mode. The definition of the suspend interval is the interval between the last clock edge of the preceding transaction word and the first clock edge of the following transaction word. The default value of SP_CYCLE is 0x3 (3.5 serial clock cycles). This SP_CYCLE setting will not take effect to the word suspend interval if the software disables the FIFO mode.

If both the VARCLK_EN, SPI_CNTRL[23], and the FIFO bit, SPI_CNTRL[21], are set as 1, the minimum word suspend period is (6.5 + SP_CYCLE)*SPI serial clock period.

5.18.4.13 Byte Reorder

When the transfer is set as MSB first (LSB = 0), the REORDER is enabled, the data stored in the SPI_TX FIFO and SPI_RX FIFO will be rearranged in the order as [BYTE0, BYTE1, BYTE2, BYTE3] in 32 bit transfer mode (TX_BIT_LEN = 0). The sequence of transmitted/received data will be BYTE0, BYTE1, BYTE2, and then BYTE3. If the TX_BIT_LEN is set as 24-bits transfer mode, the data in TX buffer and RX buffer will be rearranged as [unknown byte, BYTE0, BYTE1, BYTE2]. The SPI controller will transmit/receive data with the sequence of BYTE0, BYTE1 and then BYTE2. Each byte will be transmitted/received with MSB first. The rule of 16-bits mode is the same as above. Byte reorder function is only available when TX_BIT_LEN is configured as 16, 24, and 32 bits.

Note: The byte reorder function is not supported when the variable serial clock function is enabled.

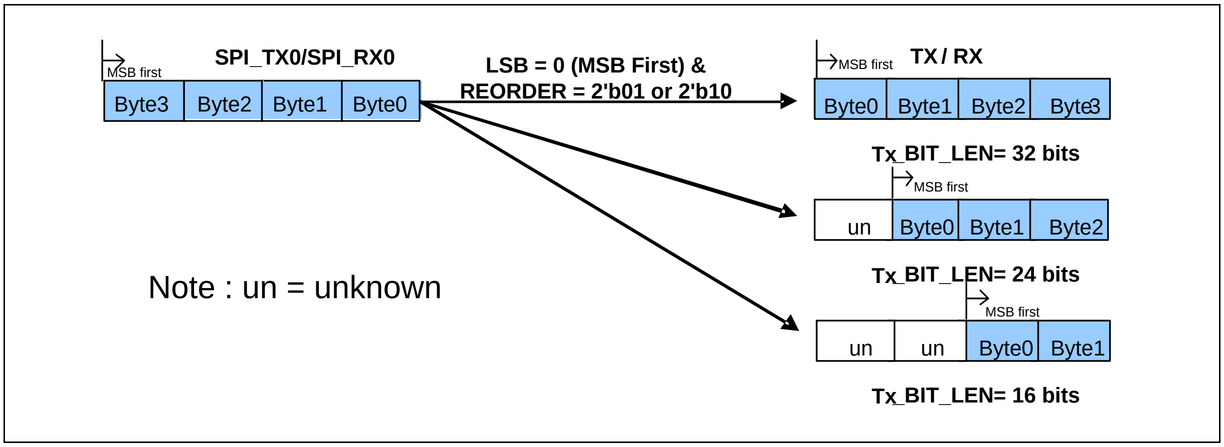


Figure 5.18-6 SPI Byte Reorder

5.18.4.14 Byte Suspend

In Master mode, if SPI_CTL[19] is set to 1, the hardware will insert a suspend interval of 0.5~15.5 serial clock periods between two successive bytes in a transfer word. Both settings of byte suspend interval and word suspend interval are configured in SP_CYCLE.

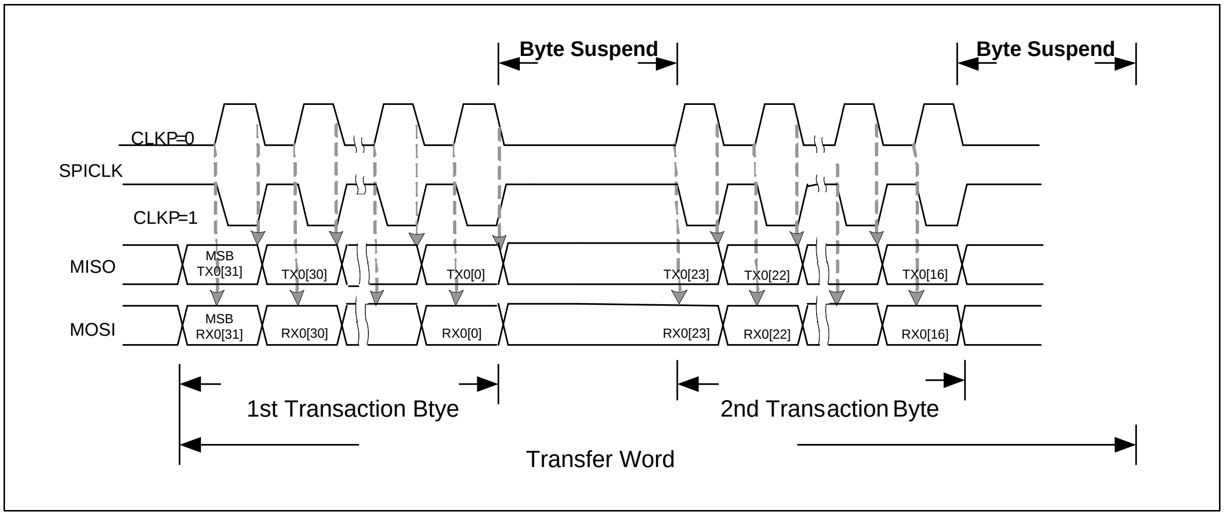


Figure 5.18-7 SPI Byte Suspend Mode

5.18.4.15 Interrupt

Each SPI controller can generates an individual interrupt when data transfer is finished or the FIFO reach the threshold level and the respective interrupt event flag INTSTS (SPI_STATUS[7], TX_INTSTS (SPI_STATUS[10]), RX_INTSTS (SPI_STATUS[8]) will be set. The interrupt event flag will generates an interrupt to CPU if the interrupt enable bit INTEN (SPI_CTL[17]) is set. The interrupt event flag INTSTS, TX_INTSTS, and RX_INTSTS can be cleared only by writing 1 to it.

As the SPI controller finishes a unit transfer, the unit transfer interrupt flag IF (SPI_STS[7]) will be set to 1. The unit transfer interrupt event will generate an interrupt to CPU if the unit transfer interrupt enable bit IE (SPI_CNTRL[17]) is set. The unit transfer interrupt flag can be cleared only by writing 1 to it.

In 3-wire mode, the slave 3-wire mode start interrupt flag, SLV_START_INTSTS, will be set to 1 when the slave senses the SPI clock signal. The SPI controller will issue an interrupt if the SSTA_INTEN is set to 1. If the count of the received bits is less than the setting of TX_BIT_LEN and there is no more serial clock input over the expected time period which is defined by the user, the user can set the SLV_ABORT bit to abort the current transfer. The unit transfer interrupt flag, IF, will be set to 1 if the software set the SLV_ABORT bit.

In FIFO mode, there is time-out function to inform user. If there is a received data in the FIFO and it does not be read by software over 64 SPI engine clock periods in master mode or over 576 SPI engine clock periods in slave mode, it will send a time-out interrupt to the system if the time-out interrupt enable bit, FIFO_CTL[7], is set to 1.

5.18.4.16 Two Bit Transfer Mode

This SPI controller also supports 2-bit transfer mode when enabling the TWOB bit, SPI_CTL[22]. When the TWOB bit is enabled, it can transmit and receive 2-bit serial transfer data simultaneousl$^{y.}$

The 1st bit is through the MOSI0 and MISO0 port to transmit the data from SPI_TX0 register and receive the data into SPI_RX0 regist^{er}. The 2nd bit is through the MOSI1and MISO1 port to transmit the data from SPI_TX1 register and receive the data into SPI_RX1 register. The system block and timing of 2-bit transfer mode are shown below.

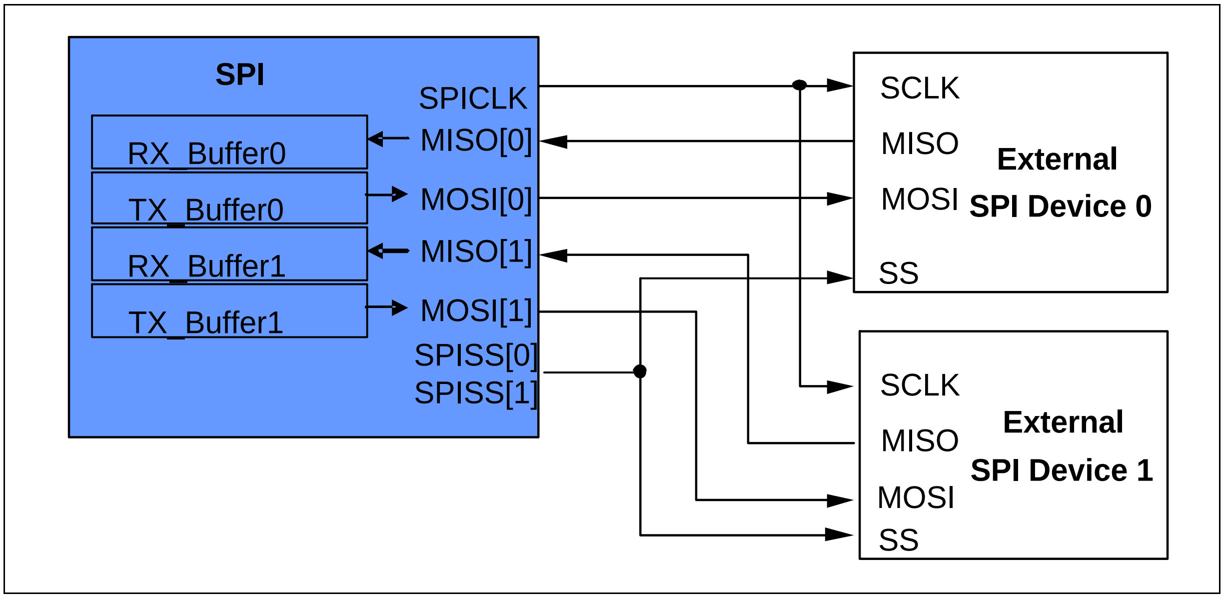


Figure 5.18-8 SPI 2-bit Transfer Mode

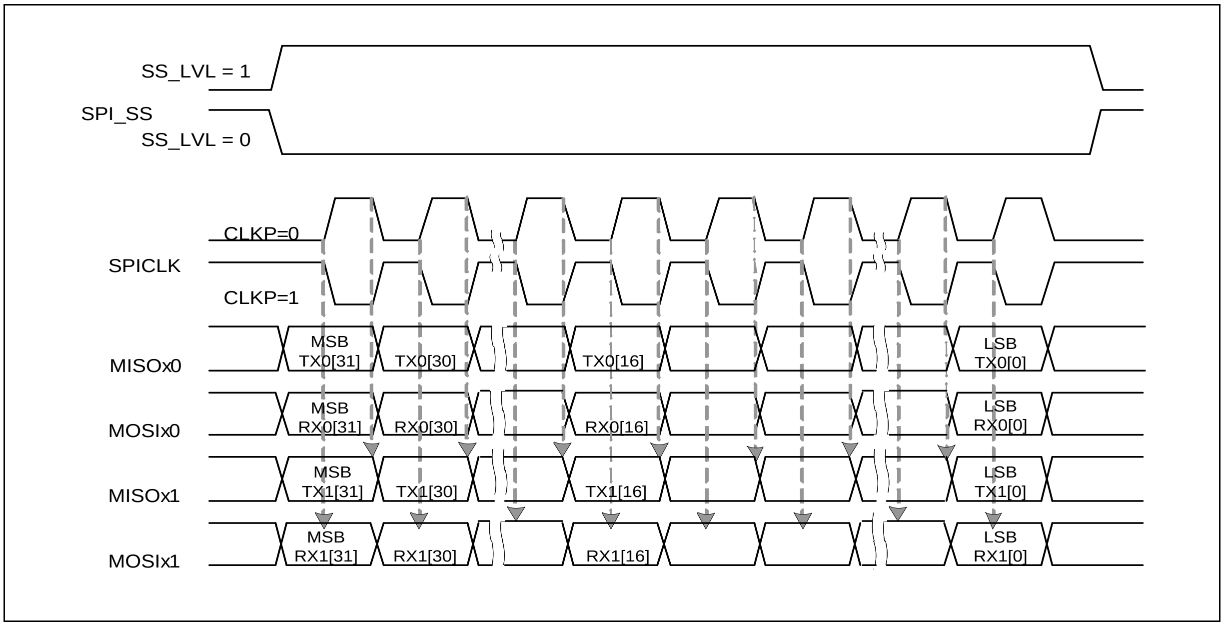


Figure 5.18-9 SPI 2-bit Transfer Mode Timing Diagram

5.18.4.17 Dual IO Mode

This SPI controller also supports dual IO transfer when set the DUAL_IO_EN bit (SPI_CTL[29]) to 1. The DUAL_IO_DIR bit (SP)_CTL2[28] is used to define the direction of the transfer data. When set the DUAL_IO_DIR bit to 1, the controller will send the data to external device. When the DUAL_IO_DIR bit set 0, the controller will read the data from the external device. This function supports 8, 16, 24, and

32-bits of bit length.

The dual IO mode is not supported when the 3-wire mode or the byte reorder function is enabled.

If both the DUAL_IO_EN and DUAL_IO_DIR bits are set as 1, the MOSI0 is the even bit data output and the MISO0 will be set as the odd bit data output. If the DUAL_IO_EN is set as 1 and DUAL_IO_DIR is set as 0, both the MISO0 and MOSI0 will be set as data input ports.

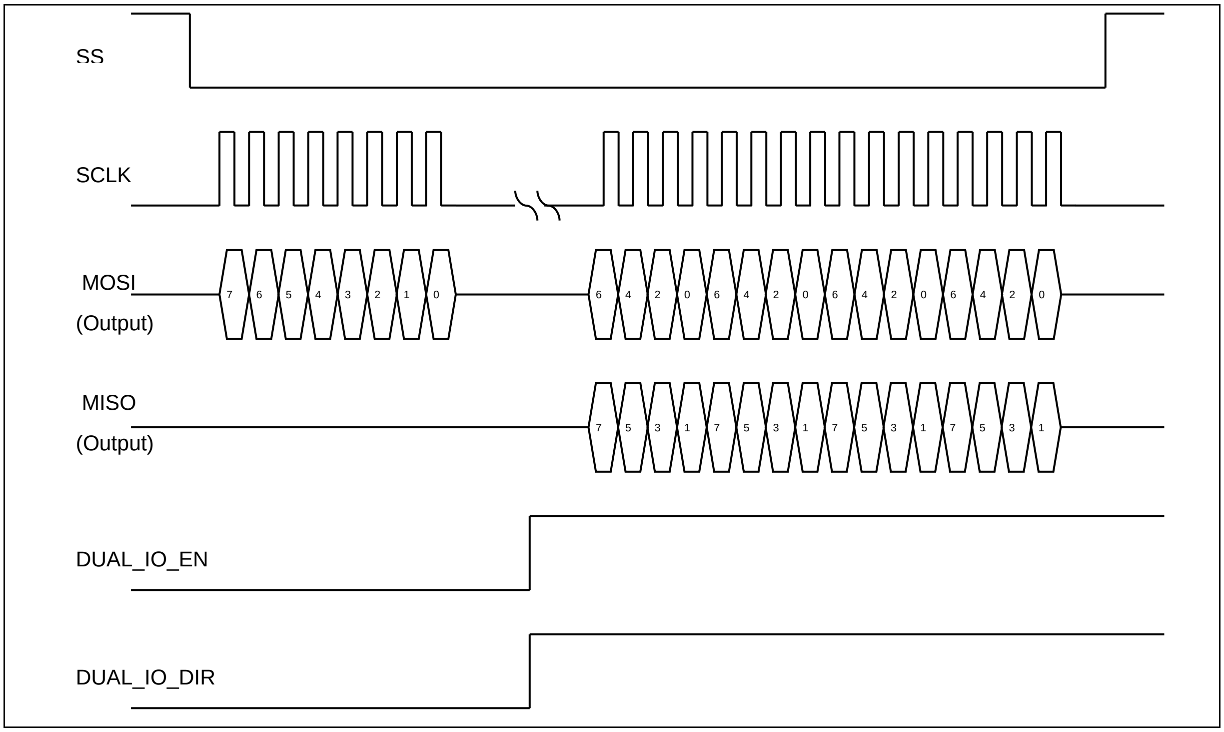


Figure 5.18-10 SPI DUAL-I/O output Sequence

Note: The byte reorder and no slave select functions are not support in DUAL I/O mode.

5.18.4.18 Time Out

In FIFO mode, there is time-out function to inform user. If there is a received data in the FIFO and it isn't read by user over 64 SPI engine clock in master mode or over 514 SPI engine clock in slave mode, it will send a time-out interrupt to the system if the time-out enable bit, FIFO_CTL[7], is set to 1.

5.18.4.19 FIFO Mode

The SPI controller supports FIFO mode when the FIFO bit, SPI_CNTRL[21], is set as 1. The SPI controllers equip with 8 32-bit wide transmit and receive FIFO buffers.

The transmit FIFO buffer is an 8-level depth, 32-bit wide, first-in, first-out register buffer. The software can write data to the transmit FIFO buffer by writing the SPI_TX0 register. The data stored in the transmit FIFO buffer will be read and sent out by the transmission control logic. If the 8-level transmit FIFO buffer is full, the TX_FULL bit will be set to 1. When the SPI transmission logic unit draws out the last datum of the transmit FIFO buffer, so that the 8-level transmit FIFO buffer is empty, the TX_EMPTY bit will be set to 1. Notice that the TX_EMPTY flag is set to 1 while the last transaction is still in progress. In master mode, the software should check both the GO_BUSY bit and TX_EMPTY bit to make sure whether the SPI is in idle or not.

The received FIFO buffer is also an 8-level depth, 32-bit wide, first-in, first-out register buffer. The receive control logic will store the received data to this buffer. The software can read the FIFO buffer data from SPI_RX0 register. There are FIFO related status bits, like RX_EMPTY and RX_FULL, to indicate the current status of FIFO buffer.

In FIFO mode, the software can set the transmitting and receiving threshold by setting the TX_THRESHOLD and RX_THRESHOLD settings. When the count of valid data stored in transmit FIFO buffer is less than or equal to TX_THRESHOLD setting, the TX_INTSTS bit will be set to 1. When the count of valid data stored in receive FIFO buffer is larger than RX_THRESHOLD setting, the RX_INTSTS bit will be set to 1.

In FIFO mode, the software can write 8 data to the SPI transmit FIFO buffer in advance. When the SPI controller operates in FIFO mode, the GO_BUSY bit of SPI_CNTRL register will be controlled by hardware, software should not modify the content of SPI_CNTRL register unless clearing the FIFO bit to disable the FIFO mode.

In Master mode transmission operation, the TX_EMPTY flag will be cleared to 0 when the FIFO bit is set to 1 and the software write the first datum to the SPI_TX0 register. The transmission starts immediately as long as the transmit FIFO buffer is not empty. User can write the next data into SPI_TX0 register immediately. The SPI controller will insert a suspend interval between two successive transactions in FIFO mode and the period of suspend interval is decided by the setting of SP_CYCLE (SPI_CNTRL [15:12]). User can write data into SPI_TX0 register as long as the TX_FULL flag is 0.

The subsequent transactions will be triggered automatically if the transmitted data are updated in time. If the SPI_TX0 register does not be updated after all data transfer are done, the transfer will stop.

In Master mode reception operation, the serial data are received from MISOx pin and stored to receive FIFO buffer. The RX_EMPTY flag will be cleared to 0 while the receive FIFO buffer contain unread data. The software can read the received data from SPI_RX0 register as long as the RX_EMPTY flag is 0. If the receive FIFO buffer contains 8 unread data, the RX_FULL flag will be set to 1. The SPI controller will stop receiving data until the software read the SPI_RX0 register.

In Slave mode, when the FIFO bit is set as 1, the GO_BUSY bit will be set as 1 by hardware automatically. If user wants to stop the slave mode SPI data transfer, both the FIFO bit and GO_BUSY bit must be cleared to 0 by software.

In Slave mode transmission operation, when the software writes data to SPI_TX0 register, the data will be loaded into transmit FIFO buffer and the TX_EMPTY flag will be set to 0. The transmission will start when the slave device receives clock signal from master. The software can write data to SPI_TX0 register as long as TX_FULL flag is 0. After all data have been drawn out by the SPI transmission logic unit and the software does not update the SPI_TX0 register, the TX_EMPTY flag will be set to 1.

In Slave mode reception operation, the serial data is received from MOSIx pin and stored to SPI_RX0 register. The reception mechanism is similar to master mode receiving operation.

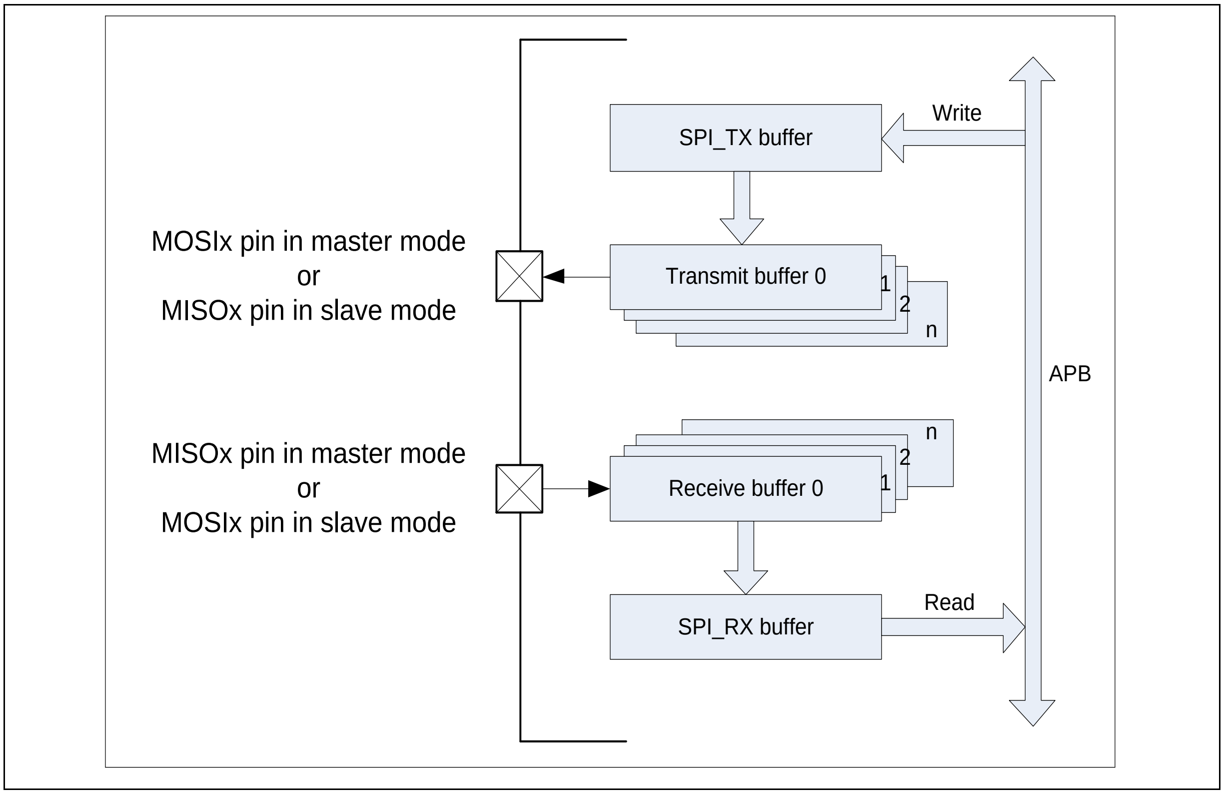


Figure 5.18-11 SPI FIFO Mode Control Timing

5.18.4.20 Wake-up

There is wake-up function in the SPI controller. When the system enter power-down mode by user, it can be waked up by receive a toggle signal of SPICLK from the external device.

5.18.4.21 SPI Programming

In Master/Slave mode, the active level of device/slave select (SPI_SS) signal can be programmed to low active or high active in SS_LVL bit (SPI_SSR[2]), but the SPISS0/1 is level trigger or edge trigger which is defined in SS_LTRIG bit (SPI_SSR[4]). The serial clock (SPICLK) idle state can be configured as high state or low state by setting the CLKP bit (SPI_CTL[11]). It also provides the bit length of a transaction in TX_BIT_LEN (SPI_CTL[7:3]), and transmit/receive data from MSB or LSB first in LSB bit (SPI_CTL[10]). Users also can select which edge of serial clock to transmit/receive data in TX_NEG/RX_NEG (SPI_CTL[2:1]) . Four SPI timing diagrams for Master/Slave operations and the related settings are shown below.

Note: Tx_NEG = TX_NEG; Rx_NEG = RX_NEG and Tx_BIT_LEN = TX_BIT_LEN.

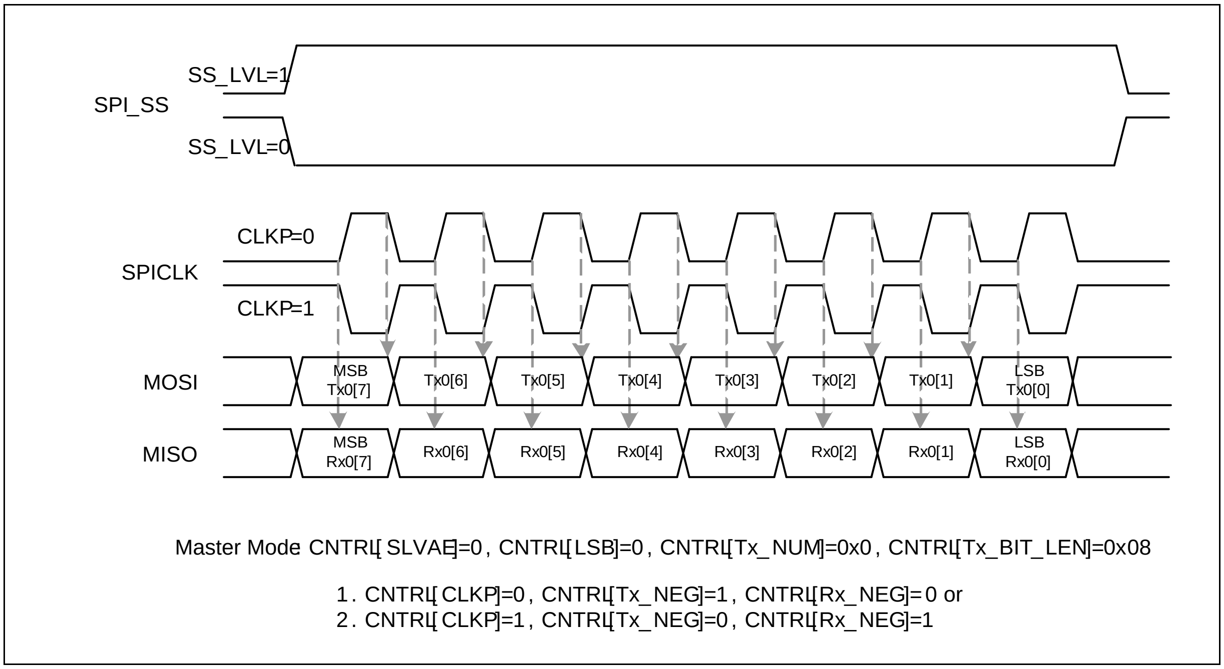


Figure 5.18-12 SPI Timing in Master Mode

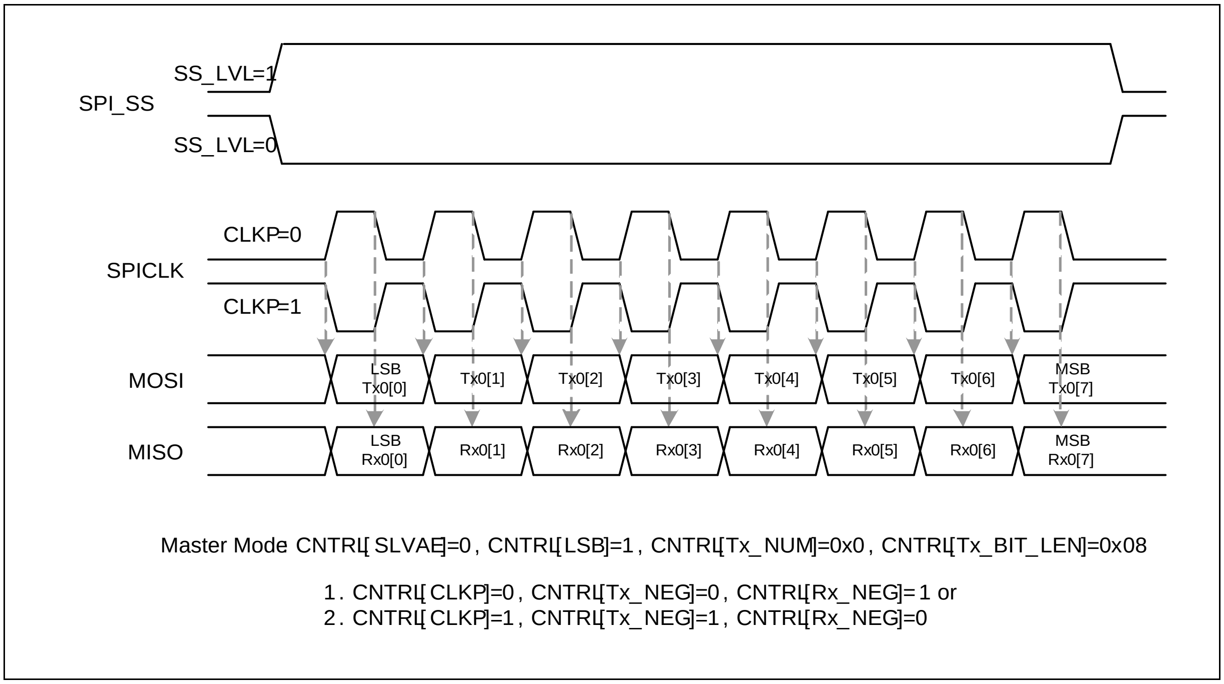


Figure 5.18-13 SPI Timing in Master Mode (Alternate Phase of SPICLK & LSB = 1)

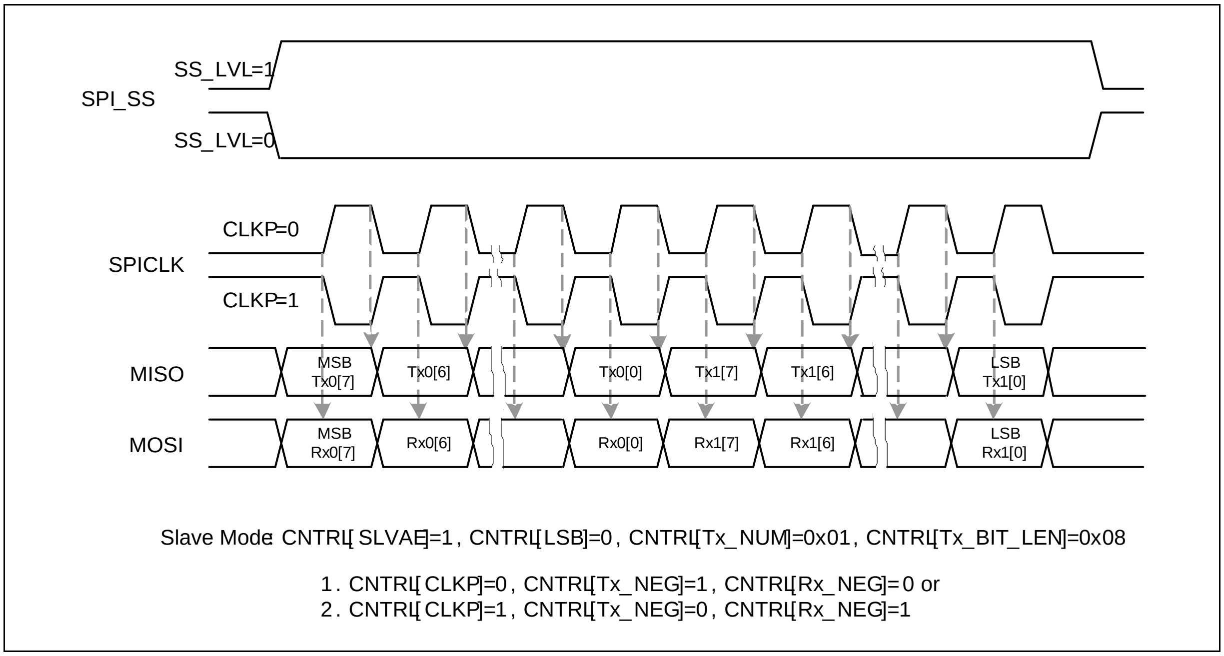


Figure 5.18-14 SPI Timing in Master Mode (Alternate Phase of SPICLK & LSB = 0)

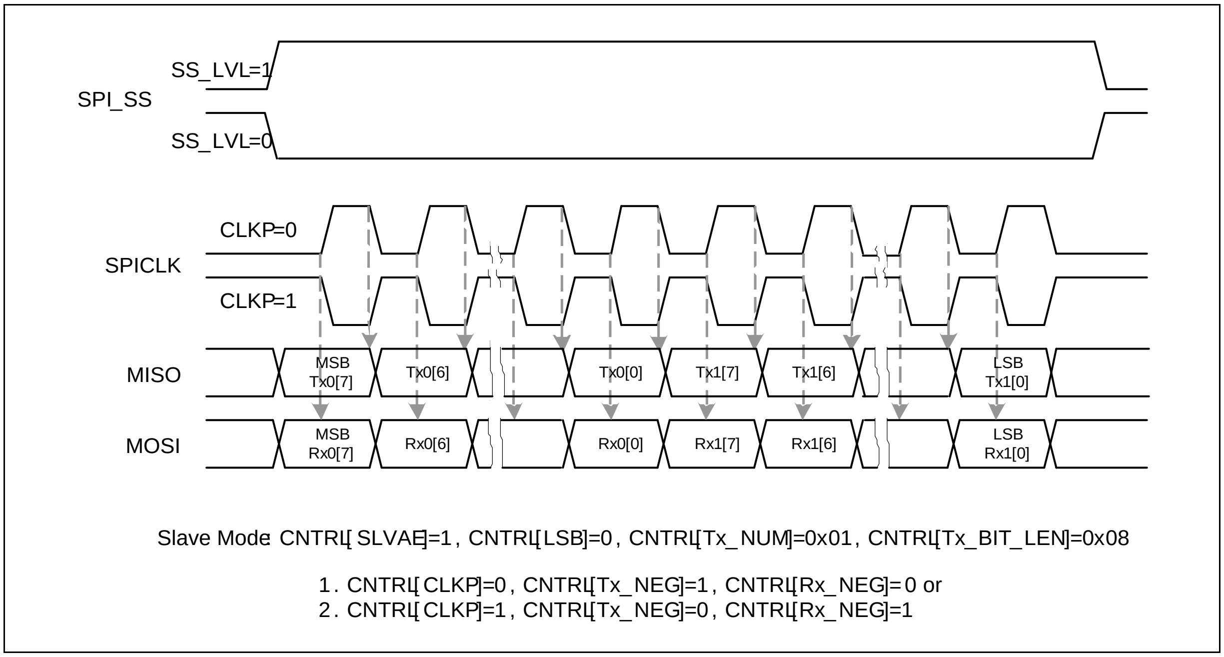


Figure 5.18-15 SPI Timing in Slave Mode

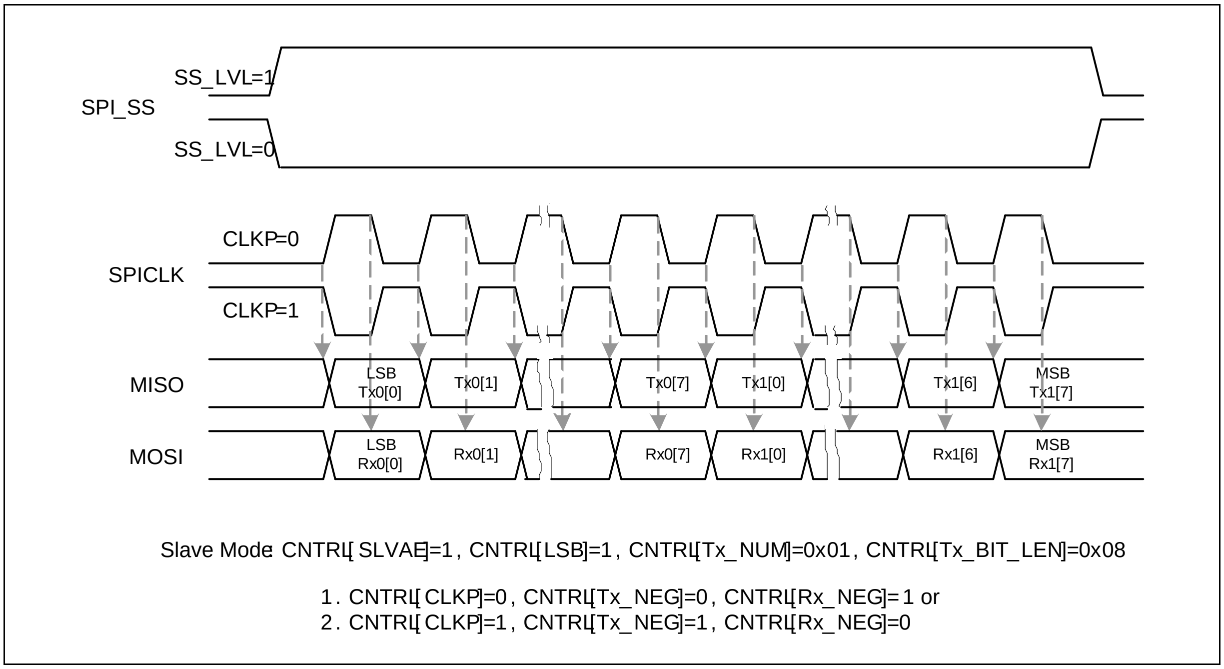


Figure 5.18-16 SPI Timing in Slave Mode (Alternate Phase of SPICLK)

5.18.4.22 SPI Programming Example

Example 1, SPI controller is set as a master to access an off-chip slave device with following specifications:

- Data bit is received on positive edge of serial clock
- Data bit is transmitted on negative edge of serial clock
- Data bit is transferred from MSB first
- SPICLK is idle at low state
- Only one byte of data to be transmitted/received in a transfer
- Slave select signal is active low

Basically, the specification of the connected off-chip slave device should be referred in details before the following steps:

1. Set the DIVIDER1 (SPI_CLKDIV[7:0]) register to determine the output frequency of serial clock.
2. Write the SPI_SSR register a proper value for the related settings of Master mode:

   (a). to disable the Automatic Slave Select bit AUTOSS (SPI_SSR[3] = 0), (b). Select low level trigger output of slave select signal in the Slave Select Active Level bit SS_LVL (SPI_SSR[2] = 0) and Slave Select Level Trigger bit SS_LTRIG. (c). Select slave select signal to be output at the I/O pin by setting the respective Slave Select Register bits SSR[0] or SSR[1] (SPI_SSR[1:0]) to active the off-chip slave devices.
3. Write the related settings into the SPI_CTL register (1). To control this SPI controller as master device in SLAVE bit (SPI_CTL[18] = 0). (2). Force the serial clock idle state at low in CLKP bit (SPI_CTL[11] = 0), (3). Select data transmitted at negative edge of serial clock in TX_NEG bit (SPI_CTL[2] = 1). (4). Select data received at positive edge of serial clock in RX_NEG bit (SPI_CTL[1] = 0). (5). Set the bit length of transaction as 8 in TX_BIT_LEN bit field (SPI_CTL[7:3] = 0x08). (6). Set LSB transfer first in LSB bit (SPI_CTL[10] = 1).

4. If this SPI master attempts to transmit (write) one byte data to the off-chip slave device, write the byte data that will be transmitted into the SPI_TX0 register.
5. If this SPI master just only attempts to receive (read) one byte data from the off-chip slave device and does not care what data will be transmitted, the software does not need to update the SPI_TX0 register.
6. Enable the GO_BUSY bit (SPI_CTL[0] = 1) to start the data transfer at the SPI interface.
7. Waiting for SPI interrupt (if the Interrupt Enable INTEN bit is set) or just polling the GO_BUSY bit till it be cleared to 0 by hardware automatically.
8. Read out the received one byte data from RDATA0[7:0) (SPI_RX0[7:0]) register.
9. Go to 4) to continue another data transfer or set SSR[0] or SSR[1] to 0 to inactivate the off-chip slave devices.

Example 2, SPI controller is set as a slave device that is controlled by an off-chip master device, and supposes the off-chip master device to access the on-chip SPI slave controller through the SPI interface with the following specifications:

- Data bit is received on positive edge of serial clock
- Data bit is transmitted on negative edge of serial clock
- Data bit is transferred from LSB first
- SPICLK is idle at high state
- Only one byte of data to be transmitted/received in a transfer
- Slave select signal is high level trigger

Basically, the specification of the connected off-chip master device should be referred in details before the following steps

1. Select high level and level trigger for the input of slave select signal in the Slave Select Active Level bit SS_LVL (SPI_SSR[2] = 1) and the Slave Select Level Trigger bit SS_LTRIG (SPI_SSR[4] = 1).
2. Write the related settings into the SPI_CTL register to control this SPI slave actions. Set this SPI controller as slave device in SLAVE bit (SPI_CTL[18] = 1). Select the serial clock idle state at high in CLKP bit (SPI_CTL[11] = 1). Select data transmitted at negative edge of serial clock in TX_NEG bit (SPI_CTL[2] = 1), Select data received at positive edge of serial clock in RX_NEG bit (SPI_CTL[1] = 0). Set the bit length of transaction as 8 bits in TX_BIT_LEN bit field (SPI_CTL[7:3] = 0x08). Set LSB transfer first in LSB bit (SPI_CTL[10] = 1), and don't care the SP_CYCLE bit field (SPI_CTL[15:12]) due to not burst mode in this case.
3. If this SPI slave attempts to transmit (be read) one byte data to the off-chip master device, write the byte data that will be transmitted into the SPI_TX0 register.
4. If this SPI slave just only attempts to receive (be written) one byte data from the off-chip master device and does not care what data will be transmitted, the software does not need to update the SPI_TX0 register.
5. Enable the GO_BUSY bit (SPI_CTL[0] = 1) to wait for the slave select trigger input and serial clock input from the off-chip master device to start the data transfer at the SPI interface.
6. Waiting for SPI interrupt (if the Interrupt Enable INTEN bit is set) or just polling the GO_BUSY bit till it be cleared to 0 by hardware automatically
7. Read out the received one byte data from (SPI_RX0[7:0])
8. Go to 3) to continue another data transfer or stop data transfer.

5.18.5 Register and Memory Map

R: read only, **W:** write only, **R/W:** both read and write, **C:** Write 1 Clear

Register	Offset	R/W	Description	Reset Value
SPI Base Address: **SPI0_BA = 0x4003_0000** **SPI1_BA = 0x4013_0000** **SPI2_BA = 0x400D_0000**				
SPI_CTL	SPI0_BA+0x00 SPI1_BA+0x00 SPI2_BA+0x00	R/W	SPI Control Register	0x0000_0004
SPI_STATUS	SPI0_BA+0x04 SPI1_BA+0x04 SPI2_BA+0x04	R/W	SPI Status Register	0x0000_0005
SPI_CLKDIV	SPI0_BA+0x08 SPI1_BA+0x08 SPI2_BA+0x08	R/W	SPI Clock Divider Register	0x0000_0000
SPI_SSR	SPI0_BA+0x0C SPI1_BA+0x0C SPI2_BA+0x0C	R/W	SPI Slave Select Register	0x0000_0000
SPI_RX0	SPI0_BA+0x10 SPI1_BA+0x10 SPI2_BA+0x10	R	SPI Receive Data FIFO Register 0	0x0000_0000
SPI_RX1	SPI0_BA+0x14 SPI1_BA+0x14 SPI2_BA+0x14	R	SPI Receive Data FIFO Register 1	0x0000_0000
SPI_TX0	SPI0_BA+0x20 SPI1_BA+0x20 SPI2_BA+0x20	W	SPI Transmit Data FIFO Register 0	0x0000_0000
SPI_TX1	SPI0_BA+0x24 SPI1_BA+0x24 SPI2_BA+0x24	W	SPI Transmit Data FIFO Register 1	0x0000_0000
SPI_VARCLK	SPI0_BA+0x34 SPI1_BA+0x34 SPI2_BA+0x34	R/W	SPI Variable Clock Pattern Flag Register	0x007F_FF87
SPI_DMA	SPI0_BA+0x38 SPI1_BA+0x38 SPI2_BA+0x38	R/W	SPI DMA Control Register	0x0000_0000

SPI_FFCTL	SPI0_BA+0x3C SPI1_BA+0x3C SPI2_BA+0x3C	R/W	SPI FIFO Control Register	0x0000_0000
SPI_INTERNAL	SPI0_BA+0x50 SPI1_BA+0x50 SPI2_BA+0x50	R/W	SPI INTERNAL Register	0x0000_0000

Note: In the following register description, the “x” indicates 0~2 for each SPI module. For example, SPIx_BA includes the SPI0_BA, SPI1_BA, and SPI2_BA.

5.18.6 Register Description

SPI Control Register (SPI_CTL)

Register	Offset	R/W	Description	Reset Value
SPI_CTL	SPI0_BA+0x00 SPI1_BA+0x00 SPI2_BA+0x00	R/W	SPI Control Register	0x0000_0004

31	30	29	28	27	26	25	24
WKEUP_EN	Reserved	DUAL_IO_EN	DUAL_IO_DIR	Reserved	Reserved		
23	22	21	20	19	18	17	16
VARCLK_EN	TWOB	FIFOM	Reserved	REORDER	SLAVE	INTEN	Reserved
15	14	13	12	11	10	9	8
SP_CYCLE				CLKP	LSB	Reserved	
7	6	5	4	3	2	1	0
TX_BIT_LEN					TX_NEG	RX_NEG	GO_BUSY

Bits	Description	
[31]	**WKEUP_EN**	**Wake-Up Enable** 1 = Wake-up function Enabled. 0 = Wake-up function Disabled when the system enters Power-down mode. When the system enters Power-down mode, the system can be wake-up from the SPI controller when this bit is enabled and if there is any toggle in the SPICLK port. After the system wake-up, this bit must be cleared by user to disable the wake-up requirement.
[30]	**Reserved**	**Reserved**
[29]	**DUAL_IO_EN**	**Dual IO Mode Enable** 1 = Dual I/O Mode function Enabled. 0 = Dual I/O Mode function Disabled.
[28]	**DUAL_IO_DIR**	**Dual IO Mode Direction** 1 = Data write in the Dual I/O Mode function 0 = Date read in the Dual I/O Mode function
[27:24]	**Reserved**	**Reserved**
[23]	**VARCLK_EN**	**Variable Clock Enable** 1 = The serial clock output frequency is variable. The output frequency is decided by the value of VARCLK (SPI_VARCLK), DIVIDER1, and DIVIDER2. 0 = The serial clock output frequency is fixed and only decided by the value of DIVIDER1 **Note:** When this VARCLK_EN bit is set to 1, the setting of TX_BIT_LEN must be programmed as 0x10 (16-bit mode).
[22]	**TWOB**	**2-bit Transfer Mode Active** 1 = 2-bit transfer mode Enabled.

<table>
<tr><td></td><td></td><td>0 = 2-bit transfer mode Disabled.

Note that when enabling TWOB, the serial transmitted 2-bits data are from SPI_TX1/0, and the received 2-bits data input are put into SPI_RX1/0.</td></tr>
<tr><td>[21]</td><td>FIFOM</td><td>FIFO Mode Enable

1 = FIFO mode.

0 = Normal mode.

Note:

1. Before enabling FIFO mode, the other related settings should be set in advance.
2. In Master mode, if the FIFO mode is enabled, the GO_BUSY bit will be set “1” automatically after the data was written into the 8-depth FIFO. The user can clear this FIFO bit after the transmit FIFO status is empty and the GO_BUSY back to 0.

3.</td></tr>
<tr><td>[19]</td><td>REORDER</td><td>Byte Reorder Function Enable

<table><tr><th>REORDER</th><th>Description</th></tr><tr><td>0</td><td>Disable byte reorderfunction.</td></tr><tr><td>1</td><td>Enable byte reorder function and insert a byte suspend interval among each byte. The setting of TX_BIT_LEN must be configured as 00b (32 bits/ word)</td></tr></table>
The suspend interval is defined in SP_CYCLE.

Note:

1. The byte reorder function is only available if TX_BIT_LEN is defined as 16, 24, and 32 bits.

2. In Slave mode with level-trigger configuration, if the byte suspend function is enabled, the slave select pin must be kept at active state during the successive four bytes transfer.

3. The byte reorder function is not supported when the variable serial clock function or the dual I/O mode is enabled.</td></tr>
<tr><td>[18]</td><td>SLAVE</td><td>Slave Mode

1 = SPI controller set as Slave mode.

0 = SPI controller set as Master mode.</td></tr>
<tr><td>[17]</td><td>INTEN</td><td>Interrupt Enable

1 = SPI Interrupt Enabled.

0 = SPI Interrupt Disabled.</td></tr>
<tr><td>[16]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[15:12]</td><td>SP_CYCLE</td><td>Suspend Interval (Master Only)

These four bits provide configurable suspend interval between two successive transmit/receive transaction in a transfer. The suspend interval is from the last falling clock edge of the current transaction to the first rising clock edge of the successive transaction if CLKP = “0”. If CLKP = “1”, the interval is from the rising clock edge to the falling clock edge.

The default value is 0x3. The desired suspend interval is obtained according to the following equation:

(SP_CYCLE[3:0) + 0.5) * period of SPICLK</td></tr>
</table>

<table>
<tr><td></td><td></td><td>Ex:
SP_CYCLE = 0x0 … 0.5 SPICLK clock cycle
SP_CYCLE = 0x1 … 1.5 SPICLK clock cycle
……
SP_CYCLE = 0xE … 14.5 SPICLK clock cycle
SP_CYCLE = 0xF … 15.5 SPICLK clock cycle
If the Variable Clock function is enabled, the minimum period of suspend interval (the transmit data in FIFO buffer is not empty) between the successive transaction is (6.5 + SP_CYCLE) * SPICLK clock cycle.</td></tr>
<tr><td>[11]</td><td>CLKP</td><td>Clock Polarity
1 = The default level of SCLK is high in idle state.
0 = The default level of SCLK is low in idle state.</td></tr>
<tr><td>[10]</td><td>LSB</td><td>Send LSB First
1 = The LSB, bit 0 of the SPI_TX0/1, is sent first to the the SPI data output pin, and the first bit received from the SPI data input pin will be put in the LSB position of the SPI_RX register (SPI_RX0/1).
0 = The MSB, which bit of transmit/receive register depends on the setting of TX_BITLEN, is transmitted/received first.</td></tr>
<tr><td>[9:8]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[7:3]</td><td>TX_BIT_LEN</td><td>Transmit Bit Length
This field specifies how many bits can be transmitted / received in one transaction. The minimum bit length is 8 bits and can be up to 32 bits.
<table>
<tr><th>TX_BIT_LEN</th><th>Description</th></tr>
<tr><td>01000</td><td>8 bits are transmitted in one transaction</td></tr>
<tr><td>01001</td><td>9 bits are transmitted in one transaction</td></tr>
<tr><td>------</td><td>----------</td></tr>
<tr><td>11111</td><td>31 bits are transmitted in one transaction</td></tr>
<tr><td>00000</td><td>32 bits are transmitted in one transaction</td></tr>
</table></td></tr>
<tr><td>[2]</td><td>TX_NEG</td><td>Transmit At Negative Edge
1 = The transmitted data output is changed on the falling edge of SPI_SCLK.
0 = The transmitted data output is changed on the rising edge of SPI_SCLK.</td></tr>
<tr><td>[1]</td><td>RX_NEG</td><td>Receive At Negative Edge
1 = The received data is latched on the falling edge of SPI_SCLK.
0 = The received data is latched on the rising edge of SPI_SCLK.</td></tr>
<tr><td>[0]</td><td>GO_BUSY</td><td>SPI Transfer Control Bit and Busy Status
1 = In Master mode, writing “1” to this bit will start the SPI data transfer; In Slave mode, writing ‘1’ to this bit indicates that the salve is ready to communicate with a master.
0 = Writing this bit “0” will stop data transfer if SPI is transferring.
If the FIFO mode is disabled, during the data transfer, this bit keeps the value of ‘1’. As the transfer is finished, this bit will be cleared automatically. Software can read this bit to check if the SPI is in busy status.
In FIFO mode, this bit will be controlled by hardware. Software should not modify this bit. In slave mode, this bit always returns 1 when software reads this register. In master mode, this bit reflects the busy or idle status of SPI.</td></tr>
</table>

		Note: 1. When FIFO mode is disabled, all configurations should be set before writing "1" to the GO_BUSY bit in the SPI_CTL register. 2. When FIFO bit is disabled and the software uses TX or RX PDMA function to transfer data, this bit will be cleared after the PDMA controller finishes the data transfer.

SPI Status Register (SPI_STATUS)

Register	Offset	R/W	Description	Reset Value
SPI_STATUS	SPI0_BA+0x04 SPI1_BA+0x04 SPI2_BA+0x04	R/W	SPI Status Register	0x0000_0005

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TX_FIFO_CNT				RX_FIFO_CNT			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved			TIME_OUT_STS	Reserved	TXINT_STS	RX_OVER_RUN	RXINT_STS
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
INTSTS	SLV_START_INTSTS	Reserved	LTRIG_FLAG	TX_FULL	TX_EMPTY	RX_FULL	RX_EMPTY

Bits	Description	
[31:24]	**Reserved**	**Reserved**
[23:20]	**TX_FIFO_CNT**	**Data counts in TX FIFO (Read Only)**
[19:16]	**RX_FIFO_CNT**	**Data counts in RX FIFO (Read Only)**
[15:13]	**Reserved**	**Reserved**
[12]	**TIME_OUT_STS**	**TIMEOUT Interrupt Flag** 1 = RX fifo is not empty and there is not be read over the 64 SPI_CLK period in master mode and over the 576 ECLK period in slave mode. When the received fifo is read by user, the timeout status will be cleared automatically. 0 = There is not timeout event on the received buffer. **Note:** This bit will be cleared by writing 1 to itself.
[11]	**Reserved**	**Reserved**
[10]	**TXINT_STS**	**TX FIFO Threshold Interrupt Status (Read Only)** 1 = TX valid data counts small or equal than **TXTHRESHOLD.** 0 = TX valid data counts bigger than **TXTHRESHOLD.**
[9]	**RX_OVER_RUN**	**RX FIFO Over Run Status** If SPI receives data when RX FIFO is full, this bit will set to 1, and the received data will dropped. **Note:** This bit will be cleared by writing 1 to itself.
[8]	**RXINT_STS**	**RX FIFO Threshold Interrupt Status (Read Only)** 1 = RX valid data counts bigger than **RXTHRESHOLD.**

		0 = RX valid data counts small or equal than **RXTHRESHOLD.** **Note:** If RXINT_EN = 1 and RX_INTSTS = 1, SPI will generate interrupt.
[7]	**INTSTS**	**Interrupt Status** 1 = Transfer is done. The interrupt is requested when the INTEN bit is enabled. 0 = Transfer is not finished yet. **Note:** This bit is read only, but can be cleared by writing "1" to this bit.
[6]	**SLV_START_INTSTS**	**Slave Start Interrupt Status** It is used to dedicate that the transfer has started in Slave mode with no slave select. 1 = Transfer has started in Slave mode with no slave select. It is auto clear by transfer done or writing one clear. 0 = Slave started transfer no active.
[4]	**LTRIG_FLAG**	**Level Trigger Accomplish Flag (INTERNAL ONLY)** In Slave mode, this bit indicates whether the received bit number meets the requirement or not after the current transaction done. 1 = The transferred bit length meets the specified requirement which defined in TX_BIT_LEN. 0 = The transferred bit length of one transaction does not meet the specified requirement. **Note:** This bit is READ only. As the software sets the GO_BUSY bit to 1, the LTRIG_FLAG will be cleared to 0 after 4 SPI engine clock periods plus 1 system clock period. In FIFO mode, this bit is unmeaning.
[3]	**TX_FULL**	**Transmitted FIFO_FULL Status** 1 = Transmitted data FIFO is full in the dual FIFO mode. 0 = Transmitted data FIFO is not full in the dual FIFO mode.
[2]	**TX_EMPTY**	**Transmitted FIFO_EMPTY Status** 1 =Transmitted data FIFO is empty in the dual FIFO mode. 0 = Transmitted data FIFO is not empty in the dual FIFO mode.
[1]	**RX_FULL**	**Received FIFO_FULL Status** 1 = Received data FIFO is full in the dual FIFO mode. 0 = Received data FIFO is not full in dual FIFO mode.
[0]	**RX_EMPTY**	**Received FIFO_EMPTY Status** 1 = Received data FIFO is empty in the dual FIFO mode. 0 = Received data FIFO is not empty in the dual FIFO mode.

SPI Serial Clock Divider Register (SPI_CLKDIV)

Register	Offset	R/W	Description	Reset Value
SPI_CLKDIV	SPI0_BA+0x08 SPI1_BA+0x08 SPI2_BA+0x08	R/W	SPI Clock Divider Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
DIVIDER2							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DIVIDER1							

Bits	Description	
[31:24]	**Reserved**	**Reserved**
[23:16]	**DIVIDER2**	**Clock Divider 2 Register** The value in this field's the 2nd frequency divider of the PCLK to generate the serial clock of SPI_SCLK. The desired frequency is obtained according to the following equation: $f_{sclk} = \frac{f_{eclk}}{(DIVIDER2+1)*2}$
[15:8]	**Reserved**	**Reserved**
[7:0]	**DIVIDER1**	**Clock Divider 1 Register** The value in this field is the 1th frequency divider of the PCLK to generate the serial clock of SPI_SCLK. The desired frequency is obtained according to the following equation: $f_{sclk} = \frac{f_{eclk}}{(DIVIDER1+1)}$ Where f_{eclk} is the SPI engine clock source. It is defined in the CLK_SEL1.

SPI Slave Select Register (SPI_SSR)

Register	Offset	R/W	Description	Reset Value
SPI_SSR	SPI0_BA+0x0C SPI1_BA+0x0C SPI2_BA+0x0C	R/W	SPI Slave Select Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							SS_INT_OPT
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						SSTA_INTEN	SLV_ABORT
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved		NOSLVSEL	SS_LTRIG	AUTOSS	SS_LVL	SSR	

Bits	Description	
[31:10]	**Reserved**	**Reserved**
[16]	**SS_INT_OPT**	**Slave Select Interrupt Option** It is used to enable the interrupt when the transfer has done in slave mode. 1 = There is interrupt event when the slave select is inactive. It is used to inform the user the transaction has finished and the slave select into the inactive state. 0 = No any interrupt, even there is slave select inactive event.
[9]	**SSTA_INTEN**	**Slave Start Interrupt Enable** It is used to enable interrupt when the transfer has started in Slave mode with no slave select. If there is no transfer done interrupt over the time period which is defined by user after the transfer start, the user can set the SLV_ABORT bit to force the transfer done. 1 = Transaction start interrupt Enabled. It is cleared when the current transfer done or the SLV_START_INTSTS bit cleared (write 1 clear). 0 = Tansfer start interrupt Disabled.
[8]	**SLV_ABORT**	**Abort in Slave Mode with No Slave Selected** In normal operation, there is interrupt event when the received data meet the required bits which define in TX_BIT_LEN. If the received bits are less than the requirement and there is no more serial clock input over the time period which is defined by user in slave mode with no slave select, the user can set this bit to force the current transfer done and then the user can get a transfer done interrupt event. **Note:** It is auto cleared to "0" by hardware when the abort event is active.
[5]	**NOSLVSEL**	**No Slave Selected in Slave Mode** This is used to ignore the slave select signal in Slave mode. The SPI controller can work on 3 wire interface including SPICLK, SPI_MISO, and SPI_MOSI when it is set as a slave device. 1 = The controller is 3-wire bi-direction interface in Slave mode. When this bit is set as 1,

		the controller start to transmit/receive data after the GO_BUSY bit active and the serial clock input. 0 = The controller is 4-wire bi-direction interface. **Note:** In no slave select signal mode, the SS_LTRIG, SPI_SSR[4], shall be set as “1”.
[4]	**SS_LTRIG**	**Slave Select Level Trigger** 1 = The slave select signal will be level-trigger. It depends on SS_LVL to decide the signal is active low or active high. 0 = The input slave select signal is edge-trigger.
[3]	**AUTOSS**	**Automatic Slave Selection (Master Only)** 1 = If this bit is set as “1”, SPISS[1:0] signals are generated automatically. It means that device/slave select signal, which is set in SSR[1:0] register is asserted by the SPI controller when transmit/receive is started, and is de-asserted after each transaction is done. 0 = If this bit is set as “0”, slave select signals are asserted and de-asserted by setting and clearing related bits in SSR[1:0] register.
[2]	**SS_LVL**	**Slave Select Active Level** It defines the active level of device/slave select signal (SPISS[1:0]). 1 = The SPI_SS slave select signal is active High. 0 = The SPI_SS slave select signal is active Low.
[1:0]	**SSR**	**Slave Select Active Register (Master Only)** If AUTOSS bit is cleared, writing “1” to SSR[0] bit sets the SPISS[0] line to an active state and writing “0” sets the line back to inactive state.(the same as SSR[1] for SPISS[1]) If AUTOSS bit is set, writing “1” to any bit location of this field will select appropriate SPISS[1:0] line to be automatically driven to active state for the duration of the transaction, and will be driven to inactive state for the rest of the time. (The active level of SPISS[1:0] is specified in SS_LVL). **Note:** 1. This interface can only drive one device/slave at a given time. Therefore, the slaves select of the selected device must be set to its active level before starting any read or write transfer. 2. SPISS[0] is also defined as device/slave select input in Slave mode. And that the slave select input must be driven by edge active trigger which level depend on the SS_LVL setting, otherwise the SPI slave core will go into dead path until the edge active triggers again or reset the SPI core by software.

SPI Receive FIFO Register (SPI_RX)

Register	Offset	R/W	Description	Reset Value
SPI_RX0	SPI0_BA+0x10 SPI1_BA+0x10 SPI2_BA+0x10	R	SPI Receive Data FIFO Register 0	0x0000_0000
SPI_RX1	SPI0_BA+0x14 SPI1_BA+0x14 SPI2_BA+0x14	R	SPI Receive Data FIFO Register 1	0x0000_0000

31	30	29	28	27	26	25	24
RXDATA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
RXDATA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
RXDATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
RXDATA							

Bits	Description	
[31:0]	**RDATA**	**Receive Data FIFO Register** The received data can be read on it. If the FIFO bit is set as 1, the user also checks the RX_EMPTY, SPI_STATUS[0], to check if there is any more received data or not. **Note:** These registers are read only.

SPI Transmit Data FIFO Register (SPI_TX)

Register	Offset	R/W	Description	Reset Value
SPI_TX0	SPI0_BA+0x20 SPI1_BA+0x20 SPI2_BA+0x20	W	SPI Transmit Data FIFO Register 0	0x0000_0000
SPI_TX1	SPI0_BA+0x24 SPI1_BA+0x24 SPI2_BA+0x24	W	SPI Transmit Data FIFO Register 1	0x0000_0000

31	30	29	28	27	26	25	24
TDATA							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
TDATA							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TDATA							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TDATA							

Bits	Description	
[31:0]	**TDATA**	**Transmit Data FIFO Register** The Data Transmit Registers hold the data to be transmitted in the next transfer. The number of valid bits depends on the setting of transmit bit length field of the SPI_CTL register. For example, if TX_BIT_LEN is set to 0x08, the bit SPI_TX[7:0] will be transmitted in next transfer. If TX_BIT_LEN is set to 0x00, the SPI controller will perform a 32-bit transfer. **Note:** When the SPI controller is configured as a slave device and the FIFO mode is disabled, if the SPI controller attempts to transmit data to a master, the software must update the transmit data register before setting the GO_BUSY bit to 1.

SPI Variable Clock Register (SPI_VARCLK)

Register	Offset	R/W	Description	Reset Value
SPI_VARCLK	SPI0_BA+0x34 SPI1_BA+0x34 SPI2_BA+0x34	R/W	SPI Variable Clock Pattern Flag Register	0x007F_FF87

31	30	29	28	27	26	25	24
VARCLK							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
VARCLK							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
VARCLK							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
VARCLK							

Bits	Description	
[31:0]	**VARCLK**	**Variable Clock Pattern Flag** The value in this field is the frequency patterns of the SPICLK. If the bit pattern of VARCLK is '0', the output frequency of SPICLK is according the value of DIVIDER1. If the bit patterns of VARCLK are '1', the output frequency of SPICLK is according the value of DIVIDER2. **Note:** It is used for CLKP = 0 only.

SPI DMA Control Register (SPI_DMA)

Register	Offset	R/W	Description	Reset Value
SPI_DMA	SPI0_BA+0x38 SPI1_BA+0x38 SPI2_BA+0x38	R/W	SPI DMA Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved					PDMA_RST	RX_DMA_EN	TX_DMA_EN

Bits	Description	
[31:3]	**Reserved**	**Reserved.**
[2]	**PDMA_RST**	**PDMA Reset** It is used to reset the SPI PDMA function into default state. 1 = Reset PDMA function. 0 = After reset PDMA function or in normal operation. **Note:** it is auto cleared to "0" after the reset function done.
[1]	**RX_DMA_EN**	**Receiving PDMA Enable(PDMA Reads SPI Data to Memory)** Set this bit to "1" will start the receive PDMA process. SPI controller will issue request to PDMA controller automatically when there is data written into the received buffer or the status of RX_EMPTY status is set to 0 in FIFO mode. If using the RX_PDMA mode to receive data but TX_DMA is disabled, the GO_BUSY bit shall be set by user. Hardware will clear this bit to 0 automatically after PDMA transfer done. In Slave mode and the FIFO bit is disabled, if the receive PDMA is enabled but the transmit PDMA is disabled, the minimal suspend interval between two successive transactions input is need to be larger than 9 SPI slave engine clock + 4 APB clock for edge mode and 9.5 SPI slave engine clock + 4 APB clock.
[0]	**TX_DMA_EN**	**Transmit PDMA Enable (PDMA Writes Data to SPI)** Set this bit to 1 will start the transmit PDMA process. SPI controller will issue request to PDMA controller automatically. If using PDMA mode to transfer data, remember not to set GO_BUSY bit of SPI_CNTRL register. The DMA controller inside SPI controller will set it automatically whenever necessary. **Note:** 1. Two transaction need minimal 18 APB clock + 8 SPI serial clocks suspend interval in master mode for edge mode and 18 APB clock + 9.5 serial clocks for level mode. 2. If the 2-bit function is enabled, the requirement timing shall append 18 APB clock

		based on the above clock period. Hardware will clear this bit to 0 automatically after PDMA transfer done.

SPI FIFO Control Register (SPI_FFCTL)

Register	Offset	R/W	Description	Reset Value
SPI_FFCTL	SPI0_BA+0x3C SPI1_BA+0x3C SPI2_BA+0x3C	R/W	SPI FIFO Control Register	0x0000_0000

31	30	29	28	27	26	25	24
	TX_THRESHOLD				RX_THRESHOLD		
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TIMEOUT_EN	Reserved		RXOVE_INTEN	TXINT_EN	RX_INTEN	TX_CLR	RX_CLR

Bits	Description	
[31]	**Reserved**	**Reserved.**
[30:28]	**TX_THRESHOLD**	**Transmit FIFO Threshold** 3-bit register, value from 0 ~7. If TX valid data counts small or equal than **TXTHRESHOLD**, **TXINT_STS** will set to 1, else **TXINT_STS** will set to 0.
[26:24]	**RX_THRESHOLD**	**Received FIFO Threshold** 3-bits register, value from 0 ~7. If RX valid data counts large than **RXTHRESHOLD**, **RXINT_STS** will set to 1, else **RXINT_STS** will set to 0.
[23:8]	**Reserved**	**Reserved.**
[7]	**TIMEOUT_EN**	**RX Read timeout function enable** 1 = RX read Timeout function Enabled. 0 = RX read Timeout function Disabled.
[6:5]	**Reserved**	**Reserved.**
[4]	**RXOVINT_EN**	**RX FIFO Over Run Interrupt Enable** 1 = RX FIFO over run interrupt Enabled. 0 = RX FIFO over run interrupt Disabled
[3]	**TXINT_EN**	**TX Threshold Interrupt Enable** 1 = TX threshold interrupt Enable 0 = Tx threshold interrupt Disabled.
[2]	**RXINT_EN**	**RX Threshold Interrupt Enable**

		1 = RX threshold interrupt Enable 0 = Rx threshold interrupt Disabled.
[1]	**TX_CLR**	**Transmitting FIFO Counter Clear** This bit is used to clear the transmit counter in FIFO Mode. This bit can be written "1" to clear the transmitting counter and this bit will be cleared to "0" automatically after clearing transmitting counter. After the clear operation, the flag of TX_EMPTY in SPI_STATUS[2] will be set to "1".
[0]	**RX_CLR**	**Receiving FIFO Counter Clear** This bit is used to clear the receiver counter in FIFO Mode. This bit can be written "1" to clear the receiver counter and this bit will be cleared to "0" automatically after clearing receiving counter. After the clear operation, the flag of RX_EMPTY in SPI_STATUS[0] will be set to "1".

5.19 External Bus Interface

5.19.1 Overview

This chip is equipped with an external bus interface (EBI) to access external device. To save the connections between external device and this chip, EBI support address bus and data bus multiplex mode. Also, address latch enable (ALE) signal is used to differentiate the address and data cycle.

5.19.2 Features

- External devices with max. 64 Kbytes size (8-bit data width)/128 Kbytes (16-bit data width) supported
- Supports variable external bus base clock (MCLK)
- Supports 8-bit or 16-bit data width
- Supports variable data access time (tACC), address latch enable time (tALE) and address hold time (tAHD)
- Address bus and data bus multiplex mode supported to save the address pins
- Configurable idle cycle supported for different access condition: Write command finish (W2X), Read-to-Read (R2R), Read-to-Write (R2W)
- Supports PDMA and VDMA transfer

5.19.3 Block Diagram

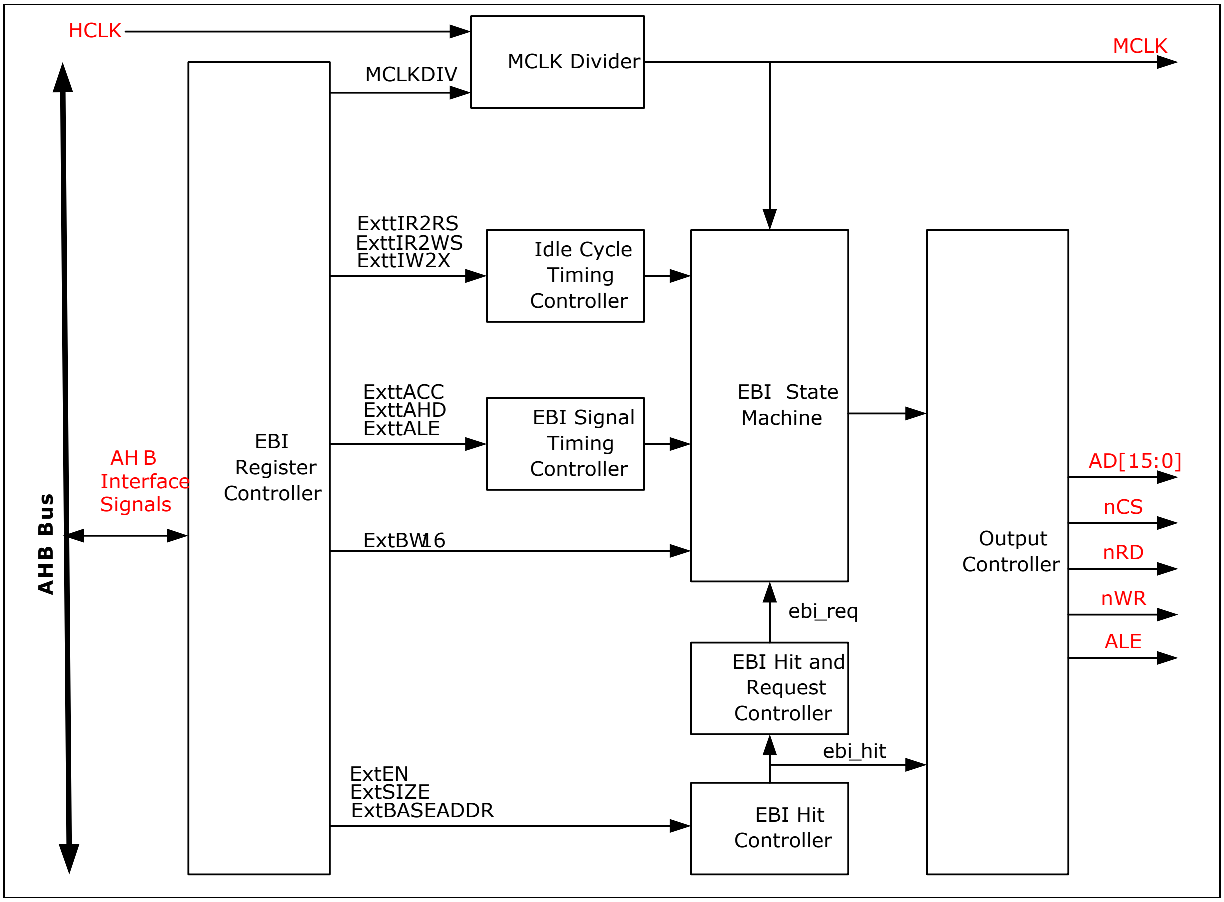


Figure 5.19-1 EBI Block Diagram

5.19.4 Functional Description

5.19.4.1 EBI Area and Address Hit

The EBI mapping address is located at 0x6000_0000 ~ 0x6001_FFFF and the total memory space is 128Kbyte. When system request address hit EBI's memory space, the corresponding EBI chip select signal is asserted and EBI state machine operates.

For an 8-bit device (64Kbyte), EBI mapped this 64Kbyte device to 0x6000_0000 ~ 0x6000_FFFF and 0x6001_0000 ~ 0x6001_FFFF simultaneously.

5.19.4.2 EBI Data Width Connection

The EBI supports devices whose address bus and data bus are multiplexed. For the external device with separated address and data bus, the connection to device needs additional logic to latch the address. In this case, pin ALE is connected to the latch device to latch the address value. Pin AD is the input of the latch device, and the output of the latch device is connected to the address of external device. For 16-bit device, the AD [15:0] shared by address and 16-bit data. For 8-bit device, only AD [7:0] shared by address and 8-bit data, AD [15:8] is dedicated for address and could be connected to 8-bit device directly.

For 8-bit data width, the system address bit [15:0] is used as the device's address [15:0]. For 16-bit data width, system address bit [16:1] is used as the device's address [15:0] and system address bit [0] is useless.

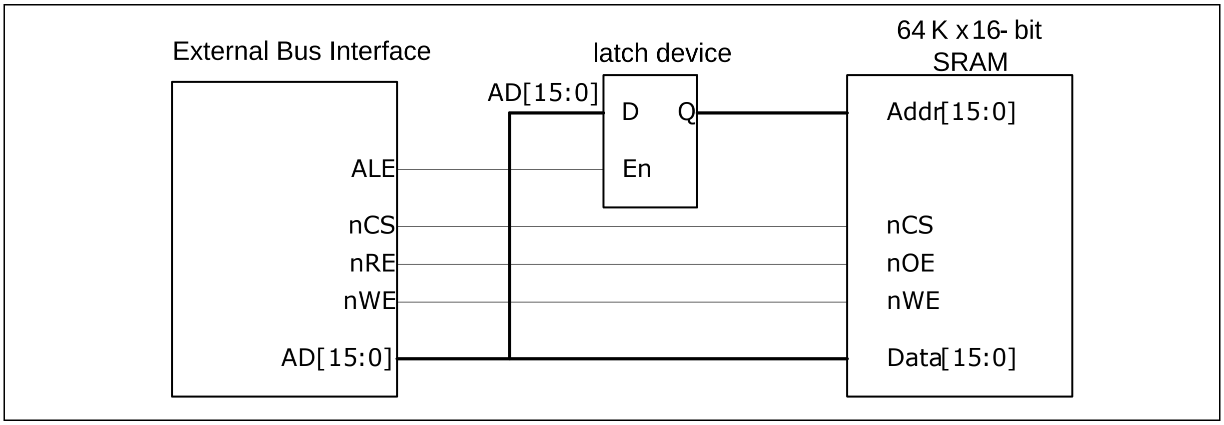


Figure 5.19-2 Connection of 16-bit EBI Data Width 16-bit Device

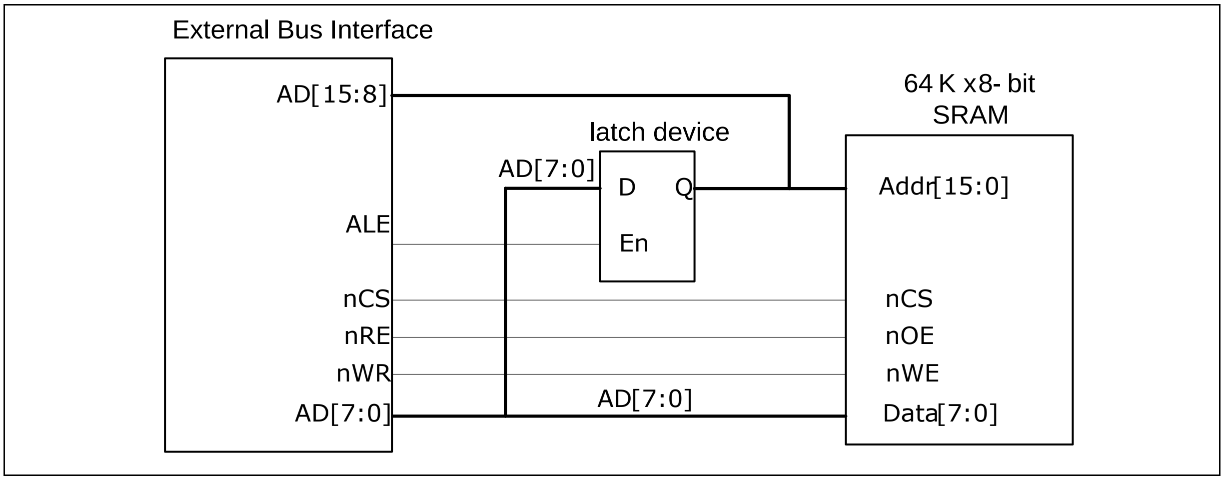


Figure 5.19-3 Connection of 8-bit EBI Data Width with 8-bit Device

When the system access data width is larger than EBI data width, EBI controller will finish a system access command by operating EBI access more than once. For example, if system requests a 32-bit data through EBI device, the EBI controller will operate accessing four times when setting EBI data width with 8-bit.

5.19.4.3 EBI Operating Control

MCLK Control

In this chip, all EBI signals will be synchronized by MCLK when EBI is operating. When this chip connects to the external device with slower operating frequency, the MCLK can divide most to 32 by setting MCLKDIV of register EBICON. Therefore, the EBI is suitable for a wide frequency range of EBI device. If MCLK is set to HCLK/1, EBI signals are synchronized by positive edge of MCLK, else by negative edge of MCLK.

Operation and Access Timing Control

In the start of access, chip select (nCS) asserts to low and wait one MCLK for address setup time (tASU) for address stable. Then ALE asserts to high after address is stable and keeps for a period of time (tALE) for address latch. After latch address, ALE asserts to low and wait one MCLK for latch hold time (tLHD) and another one MCLK cycle (tA2D) that is inserted behind address hold time to be the bus turn-around time for address change to data. Then nRD asserts to low when read access or nWR asserts to low when write access. Then nRD or nWR asserts to high after keeps access time (tACC) for reading output stable or writing finish. After that, EBI signals keep for data access hold time (tAHD) and chip select asserts to high, address is released by current access control.

EBI provides a flexible timing control for different external device. In EBI timing control, tASU, tLHD and tA2D are fixed to 1 MCLK cycle, tAHD can modulate to 1~8 MCLK cycles by setting ExttAHD of register EXTIME, and tACC can modulate to 1~32 MCLK cycles by setting ExttACC of register EXTIME, and tALE can modulate to 1~8 MCLK cycles by setting tALE of register EBICON.

Parameter	Value	Unit	Description
tASU	1	MCLK	Address Latch Setup Time.
tALE	1~8	MCLK	ALE High Period. Controlled by ExttALE of EBICON.
tLHD	1	MCLK	Address Latch Hold Time.
tA2D	1	MCLK	Address To Data Delay (Bus Turn-Around Time).
tACC	1~32	MCLK	Data Access Time. Controlled by ExttACC of EXTIME.
tAHD	1~8	MCLK	Data Access Hold Time. Controlled by ExttAHB of EXTIME.
IDLE	1~16	MCLK	Idle Cycle. Controlled by ExtIR2R, ExtIR2W and ExtIW2X of EXTIME.

Table 5.19-1 EBI timing control parameter

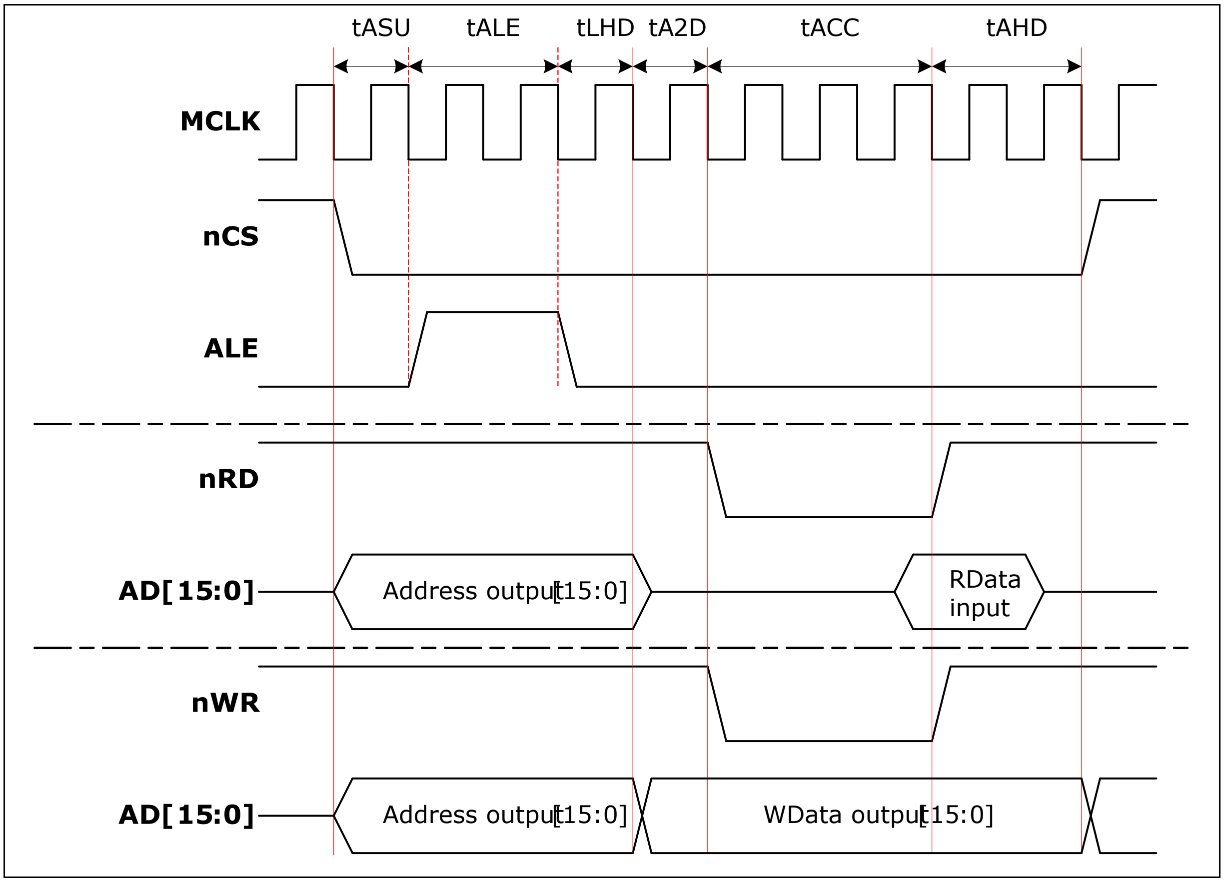


Figure 5.19-4 Timing Control Waveform for 16-bit Data Width

Figure 5.19-4 is an example of setting 16-bit data width. In this example, AD bus is used for being address [15:0] and data[15:0]. When ALE assert to high, AD is address output. After address is latched, ALE asserts to low and the AD bus change to high impedance to wait device output data in read access operation, or it is used for being write data output.

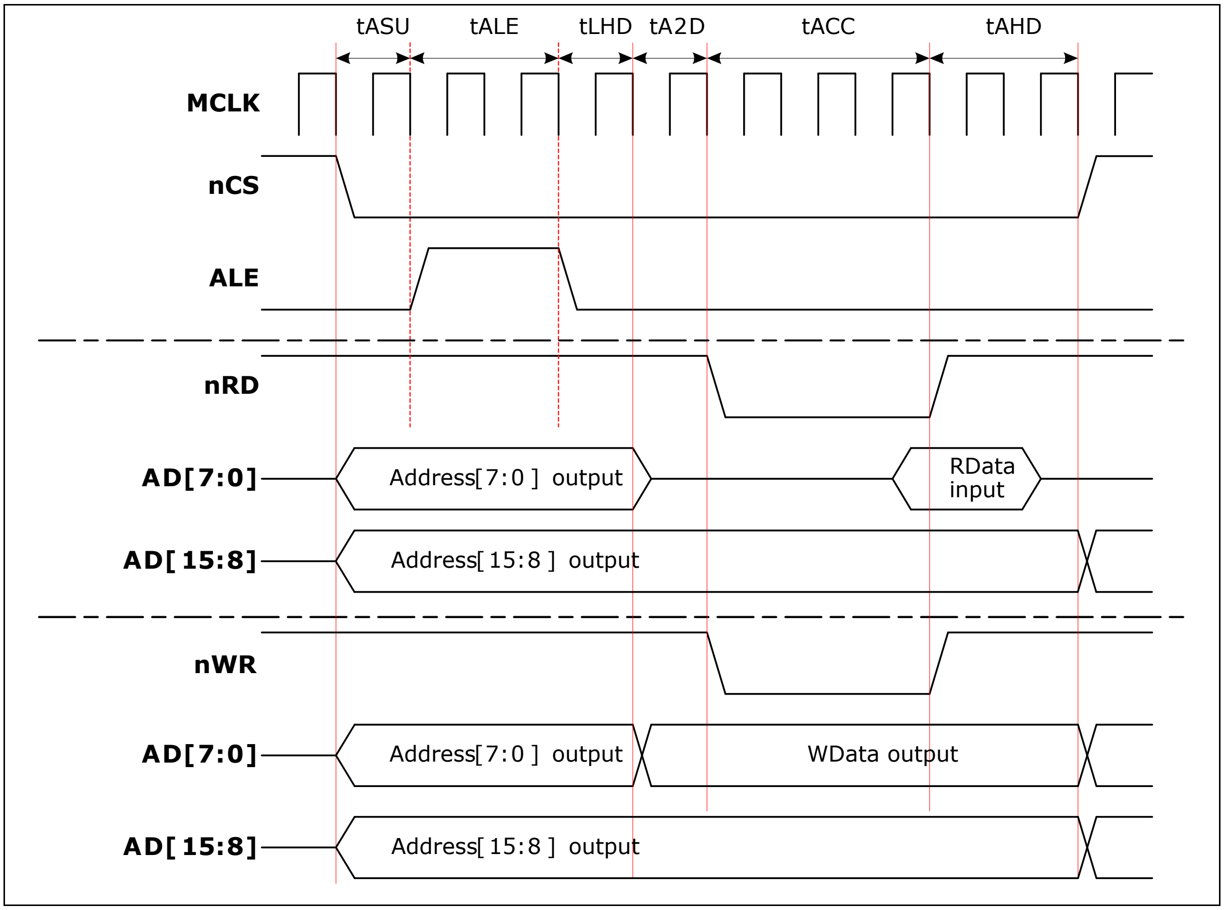


Figure 5.19-5 Timing Control Waveform for 8-bit Data Width

Figure 5.19-5 is an example of setting 8-bit data width. The difference between 8-bit and 16-bit data width is AD[15:8]. In 8-bit data width setting, AD[15:8] is always Address[15:8] output so that external latch needs only 8-bit width.

Insert Idle Cycle

When EBI accessing continuously, there may occur bus conflict if the device access time is much longer compared with system clock frequency. EBI supply additional idle cycle to solve this problem. During idle cycle, all control signals of EBI are inactive. The following figure shows the idle cycle.

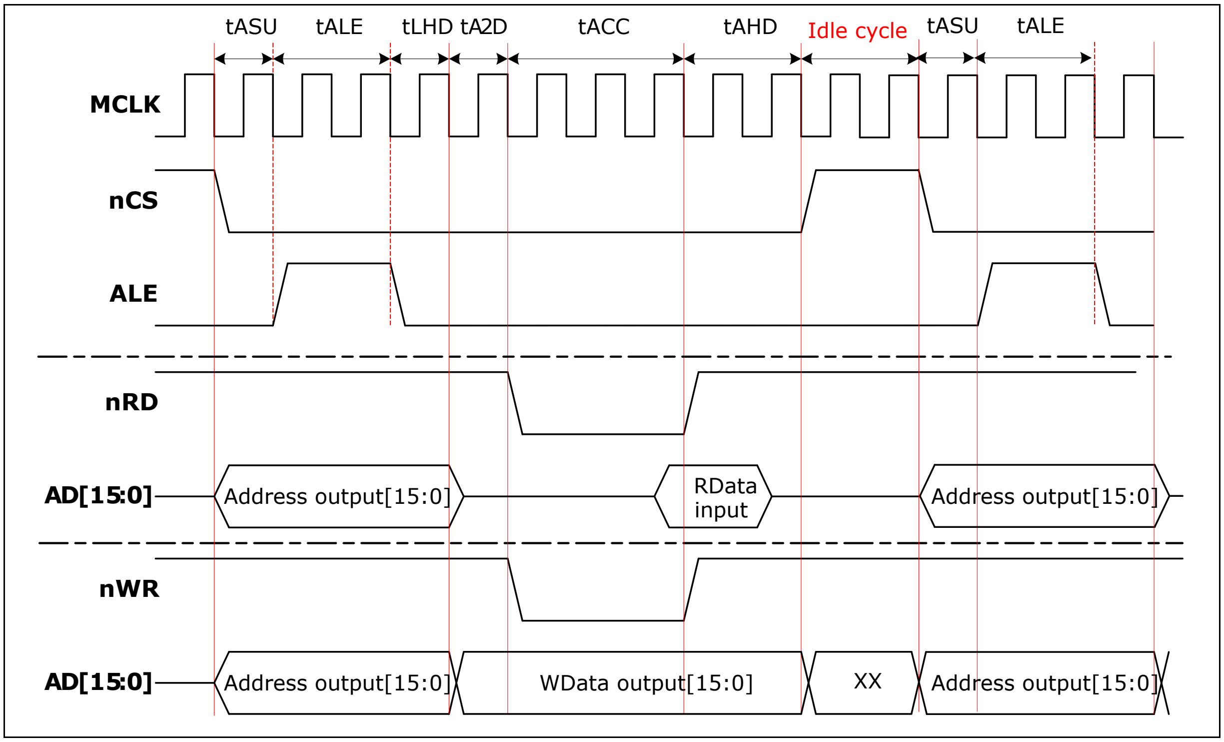


Figure 5.19-6 Timing Control Waveform for Insert Idle Cycle

There are three conditions that EBI can insert idle cycle by timing control:

- After write access
- After read access and before next read access
- After read access and before next write access

By setting ExtIW2X, ExtIR2R, and ExtIR2W of register EXTIME, the time of idle cycle can be specified from 0~15 MCLK.

5.19.5 Register and Memory Map

R: read only, **W:** write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
EBI Base Address: **EBI_BA = 0x5001_0000**				
EBICON	EBI_BA+0x00	R/W	External Bus Interface General Control Register	0x0000_0000
EXTIME	EBI_BA+0x04	R/W	External Bus Interface Timing Control Register	0x0000_0000

5.19.6 Register Description

External Bus Interface CONTROL REGISTER (EBICON)

Register	Offset	R/W	Description	Reset Value
EBICON	EBI_BA+0x00	R/W	External Bus Interface General Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved					**ExttALE**		
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved				**MCLKEN**	**MCLKDIV**		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**ExtBW16**	**ExtEN**

Bits	Description	
[31:19]	**Reserved**	**Reserved**
[18:16]	**ExttALE**	**Expand Time of ALE** The ALE width (tALE) to latch the address can be controlled by ExttALE. tALE = (ExttALE+1)*MCLK
[15:12]	**Reserved**	**Reserved**
[11]	**MCLKEN**	**External Clock Enable** This bit control if EBI generates the clock to external device. If external device is a synchronous device, it's necessary to set this bit high to enable EBI generating clock to external device. If the external device is an asynchronous device, keep this bit low is recommended to save power consumption. 0 = EBI Disabled to generate clock to external device. 1 = EBI Enabled to generate clock to external device.

Bits	Description	
[10:8]	**MCLKDIV**	**External Output Clock Divider** The frequency of EBI output clock is controlled by MCLKDIV as shown in the following table. **MCLKDIV** | **Output clock (MCLK)** 0x0 | HCLK/1 0x1 | HCLK/2 0x2 | HCLK/4 0x3 | HCLK/8 0x4 | HCLK/16 0x5 | HCLK/32 0x6 | Default 0x7 | Default Notice: Default value of output clock is HCLK/1
[7:2]	**Reserved**	**Reserved**
[1]	**ExtBW16**	**EBI Data Width 16-bit** This bit defines if the data bus is 8-bit or 16-bit. 0 = EBI data width is 8-bit 1 = EBI data width is 16-bit
[0]	**ExtEN**	**EBI Enable** This bit is the functional enable bit for EBI. 0 = EBI function is disabled 1 = EBI function is enabled

External Bus Interface Timing CONTROL REGISTER (EXTIME)

Register	Offset	R/W	Description	Reset Value
EXTIME	EBI_BA+0x04	R/W	External Bus Interface Timing Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved				ExtIR2R			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved				ExtIR2W			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ExtIW2X				Reserved	ExttAHD		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			ExttACC				

Bits	Description	
[31:28]	**Reserved**	**Reserved**
[27:24]	**ExtIR2R**	**Idle State Cycle between Read-Read** When read action is finish and next action is going to read, idle state is inserted and nCS return to high if ExtIR2R is not zero. Idle state cycle = (ExtIR2R*MCLK)
[23:20]	**Reserved**	**Reserved**
[19:16]	**ExtIR2W**	**Idle State Cycle between Read-Write** When read action is finish and next action is going to write, idle state is inserted and nCS return to high if ExtIR2W is not zero. Idle state cycle = (ExtIR2W*MCLK)
[15:12]	**ExtIW2X**	**Idle State Cycle after Write** When write action is finish, idle state is inserted and nCS return to high if ExtIW2X is not zero. Idle state cycle = (ExtIW2X*MCLK)
[11]	**Reserved**	**Reserved**
[10:8]	**ExttAHD**	**EBI Data Access Hold Time** ExttAHD define data access hold time (tAHD). tAHD = (ExttAHD +1) * MCLK
[7:5]	**Reserved**	**Reserved**
[4:0]	**ExttACC**	**EBI Data Access Time** ExttACC define data access time (tACC). tACC = (ExttACC +1) * MCLK

5.20 USB

5.20.1 Overview

The USB controller is a USB 2.0 full-speed device controller. It is compliant with USB 2.0 full speed device specification and supports control/bulk/interrupt/isochronous transfer types.

In this device controller, there are two main interfaces: the APB bus and USB bus which comes from the USB PHY transceiver. For the APB bus, the CPU can program control registers through it. There is an internal 512-byte SRAM as data buffer in this controller. For IN token or OUT token transfer, it is necessary to write data to SRAM or read data from SRAM through the APB interface. Users need to allocate the effective starting address of SRAM for each endpoint buffer through "buffer segmentation register (BUFSEG)".

This device controller contains 8 configurable endpoints. Each endpoint can be configured as IN or OUT endpoint. The function address of the device and endpoint number in each endpoint shall be configured properly in advance for receiving or transmitting a data packet correctly. The transmitting/receiving length in each endpoint is defined in maximum payload register (MXPLD) and the handshakes between Host and Device are also handled by it.

There are four different interrupt events in this controller. They are the wake-up function, device plug-in or plug-out event, USB events, like IN ACK, OUT ACK etc, and BUS events, like suspend and resume, etc. Any event will cause an interrupt, and users just need to check the related event flags in interrupt event status register (USB_INTSTS) to acknowledge what kind of events occurring, and then check the related USB Endpoint Status Register (USB_EPSTS) to acknowledge what kind of event occurring in this endpoint.

A software-disable function is also supported for this USB controller. It is used to simulate the disconnection of this device from the host. If user enables the DRVSE0 bit (USB_CTL[4]), the USB controller will force USB_DP and USB_DM to level low and USB device function is disabled (disconnected). After disable the DRVSE0 bit, USB_DP will be pulled high by internal pull-high circuit then host will enumerate the USB device connection again.

Reference: Universal Serial Bus Specification Revision 2.0

5.20.2 Features

This Universal Serial Bus (USB) performs a serial interface with a single connector type for attaching all USB peripherals to the host system. Following is the feature listing of this USB.

- Compliant with USB 2.0 Full-Speed specification.
- Provide 1 interrupt vector with 4 different interrupt events (WAKEUP, FLDET, USB and BUS).
- Supports Control/Bulk/Interrupt/Isochronous transfer type.
- Supports suspend function when no bus activity existing for 3 ms.
- Provide 8 endpoints for configurable Control/Bulk/Interrupt/Isochronous transfer types
- 512-byte SRAM buffer inside
- Provide remote wake-up capability.

5.20.3 Block Diagram

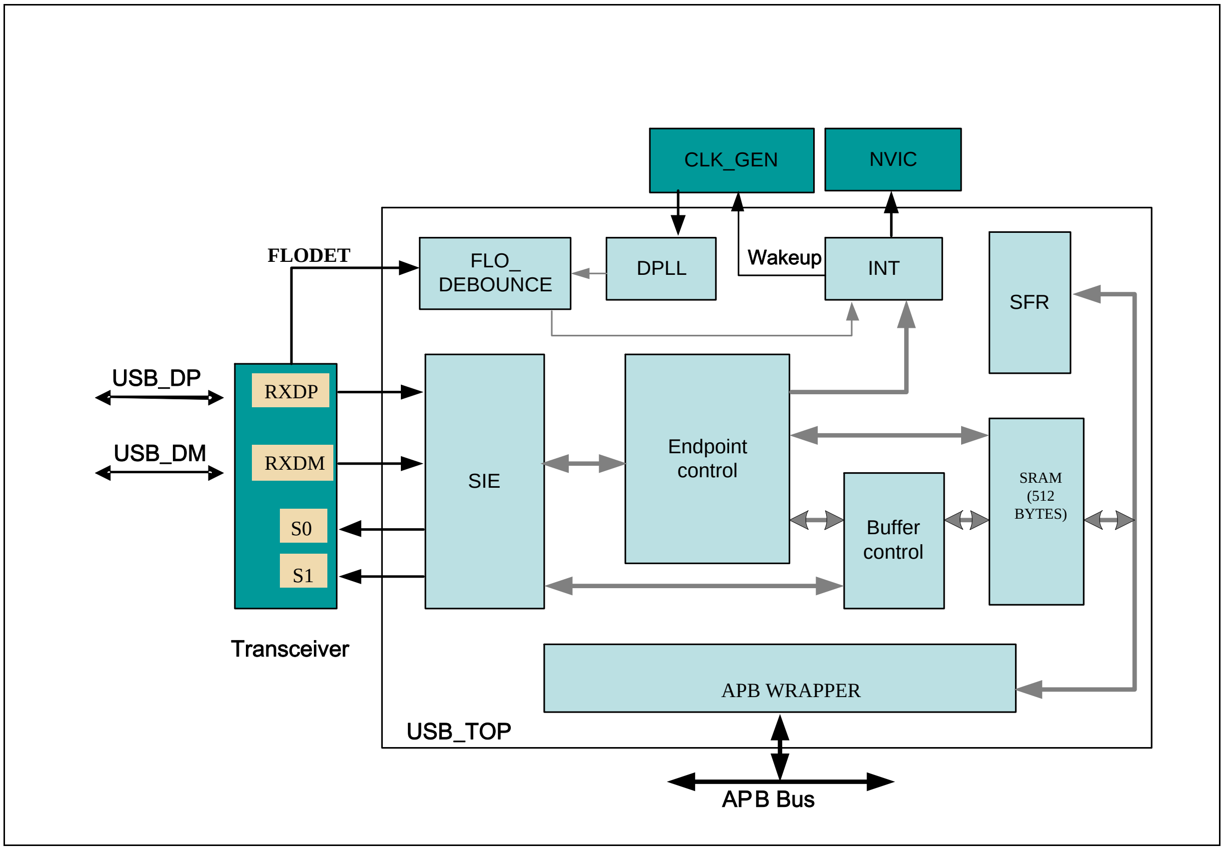


Figure 5.20-1 USB Block Diagram

5.20.4 Functional Description

5.20.4.1 SIE (Serial Interface Engine)

The SIE is the front-end of the device controller and handles most of the USB packet protocol. The SIE typically comprehends signaling up to the transaction level. The functions that it handles could include:

- Packet recognition, transaction sequencing
- SOP, EOP, RESET, RESUME signal detection/generation
- Clock/Data separation
- NRZI Data encoding/decoding and bit-stuffing
- CRC generation and checking (for Token and Data)
- Packet ID (PID) generation and checking/ decoding
- Serial-Parallel/ Parallel-Serial conversion

5.20.4.2 Endpoint Control

There are 8 endpoints in this controller. Each of the endpoint can be configured as IN or OUT endpoint. All the operations including Control, Bulk, Interrupt and Isochronous transfer are implemented in this block. The block of ENDPOINT CONTROL is also used to manage the data sequential synchronization, endpoint states, current start address, transaction status, and data buffer status for each endpoint.

5.20.4.3 Buffer Control

There is 512-byte SRAM in the controller and the 8 endpoints share this buffer. The user shall configure each endpoint's effective starting address in the buffer segmentation register before the function active. The BUFFER CONTROL block is used to control each endpoint's effective starting address and its SRAM size is defined in the MXPLD register.

The following figure describes the starting address for each endpoint according to the content of BUFSEG and MXPLD registers. If the BUFSEG0 is programmed as 0x08h and MXPLD0 is set as 0x40h, the SRAM size of endpoint 0 is start from USB_BASE + 0x108h and end in USB_BASE + 0x148h because the USB SRAM base is USB_BASE + 0x100h.

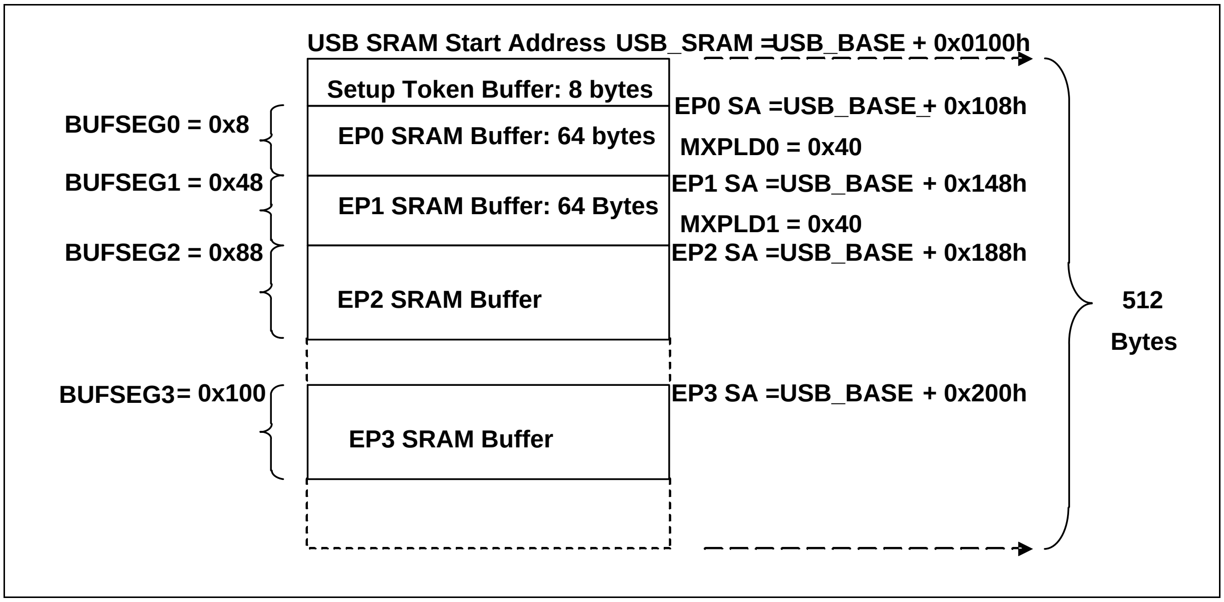


Figure 5.20-2 Endpoint allocation in SRAM

5.20.4.4 DPLL (Digital Phase Lock Loop)

The bit rate of USB data is 12 MHz. The DPLL uses the 48 MHz which comes from the clock controller to lock the input data RXDP and RXDM. The 12 MHz bit rate clock is also converted from DPLL.

5.20.4.5 Floating De-bounce

A USB device may be plug-in or plug-out from the USB. In order to monitor the state of a USB device when it is detached from the USB, the device controller provides hardware de-bounce for USB floating detect interrupt to avoid bounce problems on USB plug in and unplug. Floating detect interrupt appears about 10 ms later than USB plug-in and plug-out. A user can acknowledge USB plug-in/plug-out by reading "FLDET" in USB_BUSSTS register. The flag in "FLDET" bit represents the current state on the bus without de-bounce. If the FLDET is 1, it means the controller has plug-in the USB bus. If polling this bit to check USB state, users must add software de-bounce if necessary.

5.20.4.6 Interrupt & Wake-up

This USB provides wake-up function and 1 interrupt vector with 4 interrupt events (WAKEUP, FLDET, USB and BUS). The WAKEUP event is used to wake-up the system clock when the power-down mode is enabled. (The power-down mode function is defined in system power control register, PWRCTL). The FLDET event is used for USB plug-in or plug-out. The USB event notifies users of some USB requests, like IN ACK, OUT ACK etc., and the BUS event notifies users of some bus events, like suspend, resume, etc. User must set related bits in both NVIC and "interrupt enable register (USB_INTEN) of USB Device Controller to enable USB interrupts.

Wake-up interrupt is only present when the system entered power-down mode and then wake-up event had happened. When the system enters power-down mode, any change on USB_DP, USB_DM and device floating detect pin can wake-up this chip (provided that USB wake-up function is enabled). If this change is not intentionally, for example, a noise on floating detect pin, no interrupt but wake-up interrupt will occur. After USB wake-ups, this interrupt will occur when no other USB interrupt events are present for more than 20ms. The following is the control flow of wake-up interrupt.

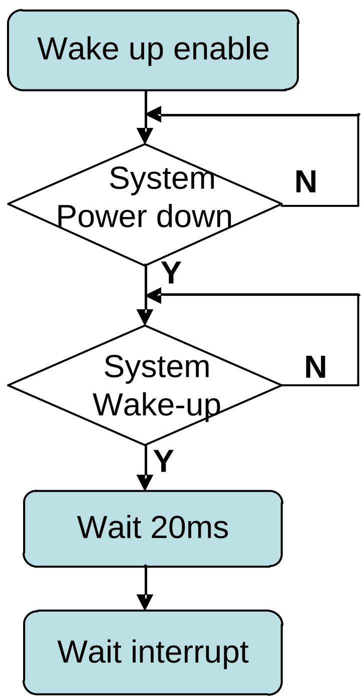


Figure 5.20-3 USB Wake-up Interrupt Operation Flow

USB interrupt is used to notify users of any USB event on the bus, and a user can read USB_EPSTS and USB_INTSTS to know what kind of request is to which endpoint and take necessary responses.

Same as USB interrupt, BUS interrupt notifies users of some bus events, like USB reset, suspend, time-out, and resume. A user can read USB_BUSSTS register to acknowledge bus events.

5.20.4.7 Power Saving

USB turns off PHY transceiver automatically to save power while this chip enters power-down mode. Furthermore, the users can write 0 into the PHY_EN bit (USB_CTL[1]), to turn off the PHY under special circumstances like suspend to save power.

Handling Transactions with USB device Peripheral

User can use interrupt or polling USB_INTSTS to monitor the USB Transactions. When a transaction occurs, the USB_INTSTS will be set by hardware and send an interrupt request to CPU (if related interrupt was enabled). Users can poll USB_INTSTS register to get these events without interrupt. The following is the control flow with interrupt enable bits active.

When USB host has requested data from device controller, users need to prepare related data into the specified endpoint buffer. After the required data is in buffer, users need to write the actual data length in the specified MAXPLD register. Once this register is written, the internal signal "In_Rdy" will be asserted and the data in buffer will be transmitted immediately after receiving associated IN token from Host. Note that after transferring the specified data, the signal "In_Rdy" will de-assert automatically by hardware.

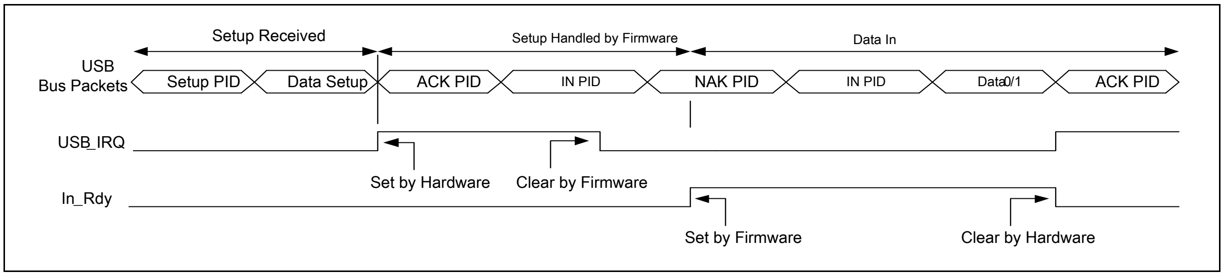


Figure 5.20-4 USB Setup Transaction Followed by Data IN Transaction

Alternatively, when USB host wants to transmit data to the OUT endpoint in the device controller, hardware will buffer these data to the specified endpoint buffer. After this transaction is completed, hardware will record the data length in related MAXPLD register and de-assert the signal "Out_Rdy". This will avoid hardware accepting next transaction until users move out current data in the related endpoint buffer. Once users have processed this transaction, the related register "MAXPLD" needs to be written by firmware to assert the signal "Out_Rdy" again to receive next transaction.

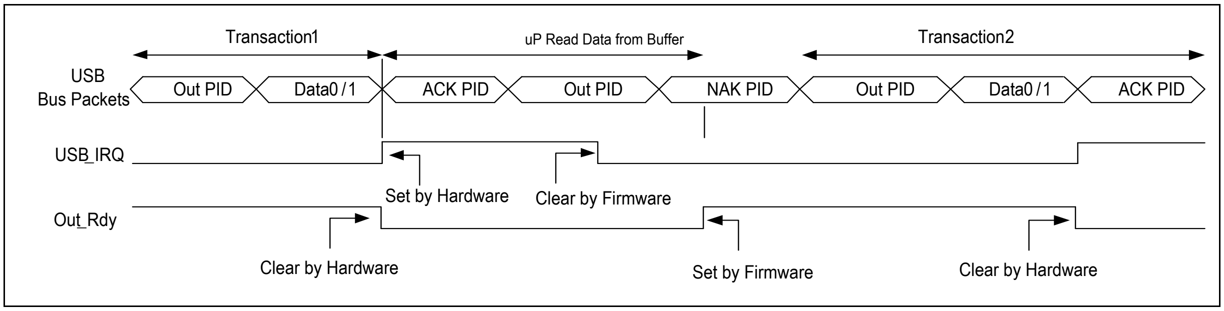


Figure 5.20-5 USB Data OUT Transaction

5.20.5 Register and Memory Map

R: read only, **W:** write only, **R/W:** both read and write, **W/C**: Write 1 Clear

Register	Offset	R/W	Description	Reset Value
USB Base Address: **USB_BA = 0x4006_0000**				
USB_CTL	USB_BA+0x000	R/W	USB Control Register	0x0000_0900
USB_BUSSTS	USB_BA+0x004	R	USB Bus Status Register	0x0000_0000
USB_INTEN	USB_BA+0x008	R/W	Interrupt Enable Register	0x0000_0000
USB_INTSTS	USB_BA+0x00C	R/W	Interrupt Event Status Register	0x0000_0000
USB_FADDR	USB_BA+0x010	R/W	Device 's Function Address Register	0x0000_0000
USB_EPSTS	USB_BA+0x014	R	Endpoint Status Register	0x0000_0000
USB_BUFSEG	USB_BA+0x018	R/W	Setup Token Buffer Segmentation Register	0x0000_0000
USB_EPSTS2	USB_BA+0x01C	R	Endpoint Bus Status	0x0000_0000
USB_BUFSEG0	USB_BA+0x020	R/W	Endpoint 0 Buffer Segmentation Register	0x0000_0000
USB_MXPLD0	USB_BA+0x024	R/W	Endpoint 0 Maximal Payload Register	0x0000_0000
USB_CFG0	USB_BA+0x028	R/W	Endpoint 0 Configuration Register	0x0000_0000
USB_BUFSEG1	USB_BA+0x030	R/W	Endpoint 1 Buffer Segmentation Register	0x0000_0000
USB_MXPLD1	USB_BA+0x034	R/W	Endpoint 1 Maximal Payload Register	0x0000_0000
USB_CFG1	USB_BA+0x038	R/W	Endpoint 1 Configuration Register	0x0000_0000
USB_BUFSEG2	USB_BA+0x040	R/W	Endpoint 2 Buffer Segmentation Register	0x0000_0000
USB_MXPLD2	USB_BA+0x044	R/W	Endpoint 2 Maximal Payload Register	0x0000_0000
USB_CFG2	USB_BA+0x048	R/W	Endpoint 2 Configuration Register	0x0000_0000
USB_BUFSEG3	USB_BA+0x050	R/W	Endpoint 3 Buffer Segmentation Register	0x0000_0000
USB_MXPLD3	USB_BA+0x054	R/W	Endpoint 3 Maximal Payload Register	0x0000_0000
USB_CFG3	USB_BA+0x058	R/W	Endpoint 3 Configuration Register	0x0000_0000
USB_BUFSEG4	USB_BA+0x060	R/W	Endpoint 4 Buffer Segmentation Register	0x0000_0000
USB_MXPLD4	USB_BA+0x064	R/W	Endpoint 4 Maximal Payload Register	0x0000_0000
USB_CFG4	USB_BA+0x068	R/W	Endpoint 4 Configuration Register	0x0000_0000
USB_BUFSEG5	USB_BA+0x070	R/W	Endpoint 5 Buffer Segmentation Register	0x0000_0000
USB_MXPLD5	USB_BA+0x074	R/W	Endpoint 5 Maximal Payload Register	0x0000_0000
USB_CFG5	USB_BA+0x078	R/W	Endpoint 5 Configuration Register	0x0000_0000
USB_BUFSEG6	USB_BA+0x080	R/W	Endpoint 6 Buffer Segmentation Register	0x0000_0000
USB_MXPLD6	USB_BA+0x084	R/W	Endpoint 6 Maximal Payload Register	0x0000_0000

USB_CFG6	USB_BA+0x088	R/W	Endpoint 6 Configuration Register	0x0000_0000
USB_BUFSEG7	USB_BA+0x090	R/W	Endpoint 7 Buffer Segmentation Register	0x0000_0000
USB_MXPLD7	USB_BA+0x094	R/W	Endpoint 7 Maximal Payload Register	0x0000_0000
USB_CFG7	USB_BA+0x098	R/W	Endpoint 7 Configuration Register	0x0000_0000
USB_BIST	USB_BA+0x0A0	R/W	USB Buffer Self Test Control Register	0x0000_0000
USB_PDMA	USB_BA+0x0A4	R/W	USB PDMA Control Register	0x0000_0000

5.20.6 Register Description

USB Controller Register (USB_CTL)

Register	Offset	R/W	Description	Reset Value
USB_CTL	USB_BA+0x000	R/W	USB Control Register	0x0000_0900

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	Reserved					WAKEUP_EN	RWAKEUP
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			DRVSE0	DPPU_EN	PWRDB	PHY_EN	USB_EN

Bits	Description	
[31:10]	**Reserved**	**Reserved**
[9]	**WAKEUP_EN**	**Wake-Up Function Enable** 1 = USB wake-up function Enabled. 0 = USB wake-up function Disabled.
[8]	**RWAKEUP**	**Remote Wake-up** 1 = Force USB bus to K (USB_DP low, USB_DM: high) state, used for remote wake-up 0 = Don't force USB bus to K state
[7:5]	**Reserved**	**Reserved**
[4]	**DRVSE0**	**Force USB PHY Transceiver to Drive SE0 (Single Ended Zero)** The Single Ended Zero is present when both lines (USB_DP, USB_DM) are being pulled low. 1 = Force USB PHY transceiver to drive SE0 0 = None The default value is "1".
[3]	**DPPU_EN**	**Pull-Up Resistor on USB_DP Enable** 1 = Pull-up resistor in USB_DP bus will be active. 0 = Pull-up resistor in USB_DP bus Disabled.
[2]	**PWRDB**	**Power down PHY Transceiver, Low Active** 1 = Turn-on related circuit of PHY transceiver. 0 = Power-down related circuit of PHY transceiver.
[1]	**PHY_EN**	**PHY Transceiver Enable** 1 = PHY transceiver Enabled.

		0 = PHY transceiver Disabled.
[0]	**USB_EN**	**USB Function Enable** 1 = USB Enabled. 0 = USB Disabled.

USB Bus Status Register (USB_BUSSTS)

Register	Offset	R/W	Description	Reset Value
USB_BUSSTS	USB_BA+0x004	R	USB Bus Status Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved			**FLDET**	**TIMEOUT**	**RESUME**	**SUSPEND**	**USBRST**

Bits	Description	
[31: 5]	**Reserved**	**Reserved**
[4]	**FLDET**	**Device Floating Detection** 1 = When the controller is attached into the USB, this bit will be set as “1” 0 = The controller didn’t attach into the USB
[3]	**TIMEOUT**	**Time-out Flag** 1 = Bus no any response more than 18 bits time. It is read only.
[2]	**RESUME**	**Resume Status** 1 = Resume from suspend. It is read only.
[1]	**SUSPEND**	**Suspend Status** 1 = Bus idle more than 3 ms, either cable is plugged off or host is sleeping. It is read only.
[0]	**USBRST**	**USB Reset Status** 1 = Bus reset when SE0 (single-ended 0) more than 2.5uS. It is read only.

USB Interrupt Enable Register (USB_INTEN)

Register	Offset	R/W	Description	Reset Value
USB_INTEN	USB_BA+0x008	R/W	Interrupt Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**WAKEUP_IE**	**FLDET_IE**	**USBEVT_IE**	**BUSEVT_IE**

Bits	Description	
[31: 4]	**Reserved**	**Reserved**
[3]	**WAKEUP_IE**	**USB Wake-up Interrupt Enable** 1 = Wake-up Interrupt Enabled. 0 = Wake-up Interrupt Disabled.
[2]	**FLDET_IE**	**Floating Detect Interrupt Enable** 1 = Floating detect Interrupt Enabled. 0 = Floating detect Interrupt Disabled.
[1]	**USBEVT_IE**	**USB Event Interrupt Enable** 1 = USB event interrupt Enabled. 0 = USB event interrupt Disabled.
[0]	**BUSEVT_IE**	**Bus Event Interrupt Enable** 1 = BUS event interrupt Enabled. 0 = BUS event interrupt Disabled.

USB Interrupt Status Register (USB_INTSTS)

Register	Offset	R/W	Description	Reset Value
USB_INTSTS	USB_BA+0x00C	R/W	Interrupt Event Status Register	0x0000_0000

31	30	29	28	27	26	25	24
SETUP	Reserved						
23	22	21	20	19	18	17	16
EPEVT7	EPEVT6	EPEVT5	EPEVT4	EPEVT3	EPEVT2	EPEVT1	EPEVT0
15	14	13	12	11	10	9	8
Reserved							
7	6	5	4	3	2	1	0
Reserved				WKEUP_STS	FLD_STS	USB_STS	BUS_STS

Bits	Description	
[31]	SETUP	**Setup Event Status** 1 = Setup event occurred, cleared by write "1" to USB_INTSTS[31]. 0 = No Setup event
[30: 24]	Reserved	Reserved
[23]	EPEVT7	**USB Event Status on EP7** 1 = USB event occurred on Endpoint 7, check USB_EPSTS2[6:4] to know which kind of USB event was occurred, cleared by write "1" to USB_INTSTS[23] or USB_INTSTS[1]. 0 = No event occurred in Endpoint 7
[22]	EPEVT6	**USB Event Status on EP6** 1 = USB event occurred on Endpoint 6, check USB_EPSTS2[2:0] to know which kind of USB event was occurred, cleared by write "1" to USB_INTSTS[22] or USB_INTSTS[1]. 0 = No event occurred in Endpoint 6
[21]	EPEVT5	**USB Event Status on EP5** 1 = USB event occurred on Endpoint 5, check USB_EPSTS[31:28] to know which kind of USB event was occurred, cleared by write "1" to USB_INTSTS[21] or USB_INTSTS[1]. 0 = No event occurred in Endpoint 5
[20]	EPEVT4	**USB Event Status on EP4** 1 = USB event occurred on Endpoint 4, check USB_EPSTS[27:24] to know which kind of USB event was occurred, cleared by write "1" to USB_INTSTS[20] or USB_INTSTS[1]. 0 = No event occurred in Endpoint 4
[19]	EPEVT3	**USB Event Status on EP3** 1 = USB event occurred on Endpoint 3, check USB_EPSTS[23:20] to know which kind of USB event was occurred, cleared by write "1" to USB_INTSTS[19] or

		USB_INTSTS[1]. 0 = No event occurred in Endpoint 3
[18]	**EPEVT2**	**USB Event Status on EP2** 1 = USB event occurred on Endpoint 2, check USB_EPSTS[19:16] to know which kind of USB event was occurred, cleared by write “1” to USB_INTSTS[18] or USB_INTSTS[1]. 0 = No event occurred in Endpoint 2
[17]	**EPEVT1**	**USB Event Status on EP1** 1 = USB event occurred on Endpoint 1, check USB_EPSTS[15:12] to know which kind of USB event was occurred, cleared by write “1” to USB_INTSTS[17] or USB_INTSTS[1]. 0 = No event occurred in Endpoint 1
[16]	**EPEVT0**	**USB Event Status on EP0** 1 = USB event occurred on Endpoint 0, check USB_EPSTS[11:8] to know which kind of USB event was occurred, cleared by write “1” to USB_INTSTS[16] or USB_INTSTS[1]. 0 = No event occurred in Endpoint 0
[15:4]	**Reserved**	**Reserved**
[3]	**WKEUP_STS**	**Wake-up Interrupt Status** 1 = Wake-up event occurred, cleared by write 1 to USB_INTSTS[3] 0 = No wake-up event is occurred
[2]	**FLD_STS**	**Floating Interrupt Status** 1 = There is attached event in the USB and it is cleared by write “1” to USB_INTSTS[2]. 0 = There is not attached event in the USB.
[1]	**USB_STS**	**USB Interrupt Status** The USB event means that there is Setup Token, IN token, OUT ACK, ISO IN, or ISO OUT event in the bus. This bit is used to indicate that there is one of events in the bus. 1 = USB event occurred, check EPSTS0~7[3:0] in USB_EPSTS[31:8] to know which kind of USB event was occurred, cleared by write “1” to USB_INTSTS[1] or USB_INTSTS[31] or EPEVT0~7. 0 = No USB event is occurred
[0]	**BUS_STS**	**BUS Interrupt Status** The BUS event means there is bus suspense or bus resume in the bus. This bit is used to indicate that there is one of events in the bus. 1 = BUS event occurred; check USB_BUSSTS[3:0] to know which kind of bus event was occurred, cleared by write “1” to USB_INTSTS[0]. 0 = No BUS event is occurred

USB Device Address Register (USB_FADDR)

A seven-bit value uses as the address of a device on the USB BUS.

Register	Offset	R/W	Description	Reset Value
USB_FADDR	USB_BA+0x010	R/W	Device 's Function Address Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	**FADDR**						

Bits	Description	
[31:7]	**Reserved**	**Reserved**
[6:0]	**FADDR**	USB device's function address

USB Endpoint Status Register (USB_EPSTS)

Register	Offset	R/W	Description	Reset Value
USB_EPSTS	USB_BA+0x014	R	Endpoint Status Register	0x0000_0000

31	30	29	28	27	26	25	24
EPSTS5				EPSTS4			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
EPSTS3				EPSTS2			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
EPSTS1				EPSTS0			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
OVERRUN	Reserved						

Bits	Description	
[31:28]	**EPSTS5**	**Endpoint 5 Bus Status** These bits are used to show the current status of this endpoint. **EPSTS5** – **Description** 000 – INACK 001 – Reserved 010 – OUT Packet Data0 ACK 110 – OUT Packet Data1 ACK 011 – Setup ACK 111 – Isochronous transfer end
[27:24]	**EPSTS4**	**Endpoint 4 Bus Status** These bits are used to show the current status of this endpoint. **EPSTS4** – **Description** 000 – INACK 001 – Reseved 010 – OUT Packet Data0 ACK 110 – OUT Packet Data1 ACK 011 – Setup ACK 111 – Isochronous transfer end
[23:20]	**EPSTS3**	**Endpoint 3 Bus Status** These bits are used to show the current status of this endpoint.

		<table><tr><th>EPSTS3</th><th>Description</th></tr><tr><td>000</td><td>INACK</td></tr><tr><td>001</td><td>Reserved</td></tr><tr><td>010</td><td>OUT Packet Data0 ACK</td></tr><tr><td>110</td><td>OUT Packet Data1 ACK</td></tr><tr><td>011</td><td>Setup ACK</td></tr><tr><td>111</td><td>Isochronous transfer end</td></tr></table>
[19:16]	**EPSTS2**	**Endpoint 2 Bus Status** These bits are used to show the current status of this endpoint. <table><tr><th>EPSTS2</th><th>Description</th></tr><tr><td>000</td><td>INACK</td></tr><tr><td>001</td><td>Reserved</td></tr><tr><td>010</td><td>OUT Packet Data0 ACK</td></tr><tr><td>110</td><td>OUT Packet Data1 ACK</td></tr><tr><td>011</td><td>Setup ACK</td></tr><tr><td>111</td><td>Isochronous transfer end</td></tr></table>
[15:12]	**EPSTS1**	**Endpoint 1 Bus Status** These bits are used to show the current status of this endpoint. <table><tr><th>EPSTS1</th><th>Description</th></tr><tr><td>000</td><td>INACK</td></tr><tr><td>001</td><td>Reserved</td></tr><tr><td>010</td><td>OUT Packet Data0 ACK</td></tr><tr><td>110</td><td>OUT Packet Data1 ACK</td></tr><tr><td>011</td><td>Setup ACK</td></tr><tr><td>111</td><td>Isochronous transfer end</td></tr></table>
[11:8]	**EPSTS0**	**Endpoint 0 Bus Status** These bits are used to show the current status of this endpoint. <table><tr><th>EPSTS0</th><th>Description</th></tr><tr><td>000</td><td>INACK</td></tr><tr><td>001</td><td>Resrved</td></tr><tr><td>010</td><td>OUT Packet Data0 ACK</td></tr><tr><td>110</td><td>OUT Packet Data1 ACK</td></tr><tr><td>011</td><td>Setup ACK</td></tr><tr><td>111</td><td>Isochronous transfer end</td></tr></table>
[7]	**OVERRUN**	**Overrun** It means the received data is over the maximum payload number or not. 1 = Out Data more than the Max Payload in MXPLD register or the Setup Data more than

		8 Bytes 0 = No overrun.
[6:0]	**Reserved**	**Reserved**

USB Buffer Segment Register (USB_BUFSEG)

For Setup Token Only

Register	Offset	R/W	Description	Reset Value
USB_BUFSEG	USB_BA+0x018	R/W	Setup Token Buffer Segmentation Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							**BUFSEG**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BUFSEG					**Reserved**		

Bits	Description	
[31:9]	**Reserved**	**Reserved**
[8:3]	**BUFSEG**	This register is used for Setup token only. It is used to define the offset address for the Setup Token with the USB SRAM starting address. Its physical address is USB_SRAM address + {BUFSEG[5:0], 000} where the USB_SRAM = USB_BASE + 0x100h
[2:0]	**Reserved**	**Reserved**

USB Endpoint Status 2 Register (USB_EPSTS2)

Register	Offset	R/W	Description	Reset Value
USB_EPSTS2	USB_BA+0x01C	R	Endpoint Bus Status	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
EPSTS7				**EPSTS6**			

Bits	Description	
[31:8]	**Reserved**	Reserved
[6:4]	**EPSTS7**	**Endpoint 7 Bus Status** These bits are used to show the current status of this endpoint. **EPSTS7** / **Description** 000: INACK 001: Reserved 010: OUT Packet Data0 ACK 110: OUT Packet Data1 ACK 011: Setup ACK 111: Isochronous transfer end
[2:0]	**EPSTS6**	**Endpoint 6 Bus Status** These bits are used to show the current status of this endpoint. **EPSTS6** / **Description** 000: INACK 001: Reserved 010: OUT Packet Data0 ACK 110: OUT Packet Data1 ACK 011: Setup ACK 111: Isochronous transfer end

USB Buffer Segment Register (USB_BUFSEGx) x = 0~7

Register	Offset	R/W	Description	Reset Value
USB_BUFSEG0	USB_BA+0x020	R/W	Endpoint 0 Buffer Segmentation Register	0x0000_0000
USB_BUFSEG1	USB_BA+0x030	R/W	Endpoint 1 Buffer Segmentation Register	0x0000_0000
USB_BUFSEG2	USB_BA+0x040	R/W	Endpoint 2 Buffer Segmentation Register	0x0000_0000
USB_BUFSEG3	USB_BA+0x050	R/W	Endpoint 3 Buffer Segmentation Register	0x0000_0000
USB_BUFSEG4	USB_BA+0x060	R/W	Endpoint 4 Buffer Segmentation Register	0x0000_0000
USB_BUFSEG5	USB_BA+0x070	R/W	Endpoint 5 Buffer Segmentation Register	0x0000_0000
USB_BUFSEG6	USB_BA+0x080	R/W	Endpoint 6 Buffer Segmentation Register	0x0000_0000
USB_BUFSEG7	USB_BA+0x090	R/W	Endpoint 7 Buffer Segmentation Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							BUFSEG
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BUFSEG					Reserved		

Bits	Description	
[31:9]	**Reserved**	**Reserved**
[8:3]	**BUFSEG**	It is used to define the offset address for each Endpoint with the USB SRAM starting address. Its physical address is USB_SRAM address + {BUFSEG[5:0], 000}; where the USB_SRAM = USB_BASE + 0x100h. Refer to the section *5.4.3.3* for the endpoint SRAM structure and its description.
[2:0]	**Reserved**	**Reserved**

USB Maximal Payload Register (USB_MXPLDx) x = 0~7

Register	Offset	R/W	Description	Reset Value
USB_MXPLD0	USB_BA+0x024	R/W	Endpoint 0 Maximal Payload Register	0x0000_0000
USB_MXPLD1	USB_BA+0x034	R/W	Endpoint 1 Maximal Payload Register	0x0000_0000
USB_MXPLD2	USB_BA+0x044	R/W	Endpoint 2 Maximal Payload Register	0x0000_0000
USB_MXPLD3	USB_BA+0x054	R/W	Endpoint 3 Maximal Payload Register	0x0000_0000
USB_MXPLD4	USB_BA+0x064	R/W	Endpoint 4 Maximal Payload Register	0x0000_0000
USB_MXPLD5	USB_BA+0x074	R/W	Endpoint 5 Maximal Payload Register	0x0000_0000
USB_MXPLD6	USB_BA+0x084	R/W	Endpoint 6 Maximal Payload Register	0x0000_0000
USB_MXPLD7	USB_BA+0x094	R/W	Endpoint 7 Maximal Payload Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
							MXPLD
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
MXPLD							

Bits	Description	
[31:9]	**Reserved**	**Reserved**
[8:0]	**MXPLD**	**Maximal Payload** It is used to define the length of data which is transmitted to host (IN token) or the actual length of data receiving from host (OUT token). It also used to indicate that the endpoint is ready to be transmitted in IN token or received in OUT token. (1). When the register is written by CPU, For IN token, the value of MXPLD is used to define the length of data to be transmitted and indicate the data buffer is ready. For OUT token, it means that the controller is ready to receive data from host and the value of MXPLD is the maximal data length comes from host. (2). When the register is read by CPU, For IN token, the value of MXPLD is indicated the length of data be transmitted to host For OUT token, the value of MXPLD is indicated the actual length of data receiving from host. **Note:** Once MXPLD is written, the data packets will be transmitted/received immediately after IN/OUT token arrived.

USB Configuration Register (USB_CFGx) x = 0~7

Register	Offset	R/W	Description	Reset Value
USB_CFG0	USB_BA+0x028	R/W	Endpoint 0 Configuration Register	0x0000_0000
USB_CFG1	USB_BA+0x038	R/W	Endpoint 1 Configuration Register	0x0000_0000
USB_CFG2	USB_BA+0x048	R/W	Endpoint 2 Configuration Register	0x0000_0000
USB_CFG3	USB_BA+0x058	R/W	Endpoint 3 Configuration Register	0x0000_0000
USB_CFG4	USB_BA+0x068	R/W	Endpoint 4 Configuration Register	0x0000_0000
USB_CFG5	USB_BA+0x078	R/W	Endpoint 5 Configuration Register	0x0000_0000
USB_CFG6	USB_BA+0x088	R/W	Endpoint 6 Configuration Register	0x0000_0000
USB_CFG7	USB_BA+0x098	R/W	Endpoint 7 Configuration Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved	Reserved					SSTALL	CSTALL
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
DSQ_SYNC	EPMODE		ISOCH	EP_NUM			

Bits	Description	
[31:10]	**Reserved**	**Reserved**
[9]	**SSTALL**	**Set STALL Response** 1 = Set the device to respond STALL automatically 0 = Disable the device to response STALL
[8]	**CSTALL**	**Clear STALL Response** 1 = Clear the device to response STALL handshake in setup stage 0 = Disable the device to clear the STALL handshake in setup stage
[7]	**DSQ_SYNC**	**Data Sequence Synchronization** 1 = DATA1 PID 0 = DATA0 PID It is used to specify the DATA0 or DATA1 PID in the current transaction. It will toggle automatically in IN token after host response ACK. In the other tokens, the user shall take care of it to confirm the right PID in its transaction.
[6:5]	**EPMODE**	**Endpoint Mode**

		<table><tr><th>EPMODE</th><th>Description</th></tr><tr><td>00</td><td>Endpoint Disabled</td></tr><tr><td>01</td><td>Out endpoint</td></tr><tr><td>10</td><td>IN endpoint</td></tr><tr><td>11</td><td>Undefined</td></tr></table>
[4]	**ISOCH**	**Isochronous Endpoint** This bit is used to set the endpoint as Isochronous endpoint, no handshake.
[3:0]	**EP_NUM**	**Endpoint Number** These bits are used to define the endpoint number of the current endpoint

USB PDMA Controller Register (USB_PDMA)

Register	Offset	R/W	Description	Reset Value
USB_PDMA	USB_BA+0x0A4	R/W	USB PDMA Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved				**PDMA_RST**	**BYTEM**	**PDMA_TRG**	**PDMA_RW**

Bits	Description	
[31:2]	**Reserved**	**Reserved**
[3]	**PDMA_RST**	**PDMA Reset** It is used to reset the USB PDMA function into default state. 1 = Reset the PDMA function in this controller. 0 = No Reset PDMA Reset Disable **Note:** it is auto cleared to 0 after the reset function done.
[2]	**BYTEM**	**CPU access USB SRAM Size Mode Select** 1 = Byte Mode: The size of the transfer from CPU to USB SRAM is Byte order 0 = Word Mode: The size of the transfer from CPU to USB SRAM is Word order
[1]	**PDMA_TRG**	**Active PDMA Function** 1 = The PDMA function in USB is active 0 = The PDMA function is not active This bit will be automatically cleared after PDMA transfer done.
[0]	**PDMA_RW**	**PDMA_RW** 1 = The PDMA will read data from USB buffer to memory 0 = The PDMA will read data from memory to USB buffer

5.21 LCD Display Driver

5.21.1 Overview

The LCD driver can directly drive a LCD glass by creating the ac segment and common voltage signals automatically. It can support static, 1/2 duty, 1/3 duty, 1/4 duty, 1/5 duty and 1/6 duty LCD glass with up to 38 segments with 6 COM (segment 0 is used as LCD_COM4 and segment 1 is used as LCD_COM5) or 40 segments with 4 COM (LCD_COM0 ~ LCD_COM3).

A built-in charge pump function can be enabled to provide the LCD glass with higher voltage than the system voltage. The LCD driver would generate voltage higher than the threshold voltage in older to darken a segment and a voltage lower than threshold to make a segment clear. However, the LCD display segment will degrade if the applied voltage has a DC-component. To avoid this, the generated waveform by LCD driver are arranged such that average voltage of each segment is zero and the RMS(root-mean-square) voltage applied on a LCD segment lower than the segment threshold making LCD clear and RMS voltage higher than the segment threshold making LCD dark.

Note : Output voltage for ADC/LCD shared pins cannot be higher than V_{DD} because these pins are without 5V tolerance.

(LQFP64 : LCD_SEG17, LCD_SEG19, LCD_SEG20, LCD_SEG21, LCD_SEG22, LCD_SEG23)

(LQFP128 : LCD_SEG36, LCD_SEG37, LCD_SEG38, LCD_SEG39)

5.21.2 Features

- Supports up to 174 dots (6x29) or 124 dots (4x31) in LQFP64 package and 228 dots (6x38) or 160 dots (4x40) in LQFP100/LQFP128 package Segment/Com pins:
- Common 0-5 multiplexing functions with GPI/O pins
- Segment 0-39 multiplexing function with GPI/O pins
- Supports Static,1/2 bias and 1/3 bias voltage
- Six display modes: Static,1/2 duty, 1/3 duty, 1/4 duty, 1/5 duty or 1/6 duty Selectable LCD frequency by frequency divider
- Configurable frame frequency
- Internal Charge pump, adjustable contrast adjustment
- Embedded LCD bias reference ladder (R-Type, 200 kΩ resisters)
- Configurable Charge pump frequency
- Blinking capability
- Supports R/C-type method
- LCD frame interrupt

5.21.3 Block Diagram

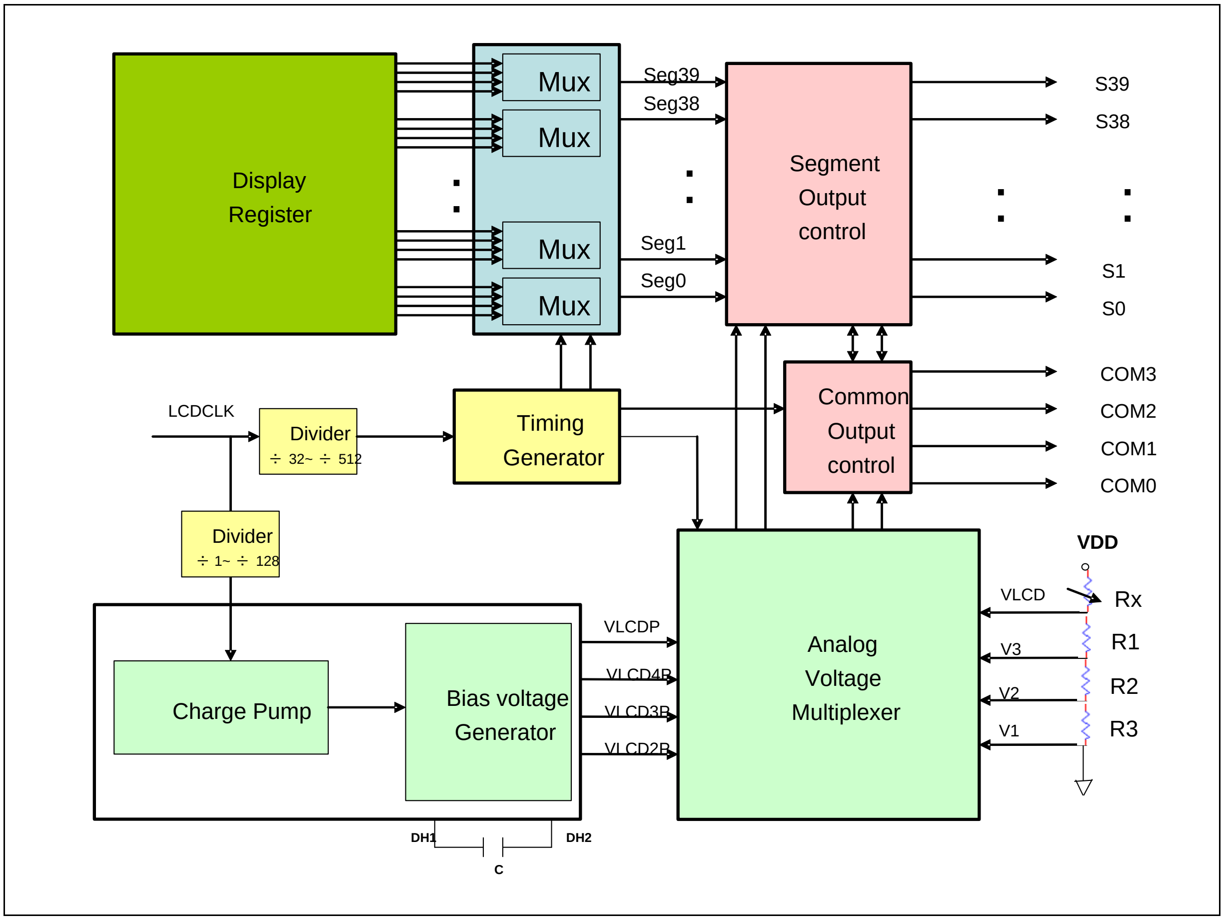


Figure 5.21-1 LCD Driver Block Diagram

5.21.4 Functional Description

The LCD driver consists of display memory register, segment output control, common output, timing generator, charge pump and analog voltage multiplexer blocks. The display memory register stores LCD segment darkened or cleared data. The data bit that is stored in display memory register with “1” makes the LCD segment be darkened and the data bit that is stored in display memory register with “0” makes the LCD segment be cleared. The display memory register is organized with LCD_MEM_0 ~ LCD_MEM_9 registers. Programming the data bits in LCD_MEM_0 ~ LCD_MEM_9 registers can make the corresponding LCD segment be darkened or cleared. The data stored in display memory register is multiplexed to segment output block sequentially with clock generated by timing generator block. The segment output block is in charged of producing SEG 0 ~ SEG 39 driving line and the common output block is in charged of producing COM0 ~ COM3 driving line. Charge pump block provides boosting voltage function for LCD glass. The charge pump input voltage range is from 1.8V to 3.6V. The multi-levels bias voltage can be programmed by CPUMP_VOL_SET bits of LCD_DISPCTL register and the multi-levels bias voltage from 2.6V to 3.3 V can be generated by charge pump block. The analog voltage multiplexer can generates static, 1/2 bias and 1/3 bias voltage output by setting BIAS_SEL bits of LCD_DISPCTL registers. User can program the BIAS_SEL bits to generate different bias voltage for COM and SEG driving line to drive LCD glass. Each common signal is selected sequentially according to the specified number of time slices of its frame period. For example, in 1/3 duty, COM0 to COM2 will output waveforms, COM3 will be tied to low. Whereas for 1/6 duty, COM0 to COM5 will output waveforms. COM signal waveform is shown in Figure 5.21-3.

5.21.4.1 LCD Display Memory MAP

DISPLAY Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COM	X	X	COM5	COM4	COM3	COM2	OCM1	COM0	X	X	COM5	COM4	COM3	COM2	OCM1	COM0	X	X	COM5	COM4	COM3	COM2	OCM1	COM0	X	X	COM5	COM4	COM3	COM2	OCM1	COM0
LCD_MEM_9	X	X	SEG39	SEG39	SEG39	SEG39	SEG39	SEG39	X	X	SEG38	SEG38	SEG38	SEG38	SEG38	SEG38	X	X	SEG37	SEG37	SEG37	SEG37	SEG37	SEG37	X	X	SEG36	SEG36	SEG36	SEG36	SEG36	SEG36
LCD_MEM_8	X	X	SEG35	SEG35	SEG35	SEG35	SEG35	SEG35	X	X	SEG34	SEG34	SEG34	SEG34	SEG34	SEG34	X	X	SEG33	SEG33	SEG33	SEG33	SEG33	SEG33	X	X	SEG32	SEG32	SEG32	SEG32	SEG32	SEG32
LCD_MEM_7	X	X	SEG31	SEG31	SEG31	SEG31	SEG31	SEG31	X	X	SEG30	SEG30	SEG30	SEG30	SEG30	SEG30	X	X	SEG29	SEG29	SEG29	SEG29	SEG29	SEG29	X	X	SEG28	SEG28	SEG28	SEG28	SEG28	SEG28
LCD_MEM_6	X	X	SEG27	SEG27	SEG27	SEG27	SEG27	SEG27	X	X	SEG26	SEG26	SEG26	SEG26	SEG26	SEG26	X	X	SEG25	SEG25	SEG25	SEG25	SEG25	SEG25	X	X	SEG24	SEG24	SEG24	SEG24	SEG24	SEG24
LCD_MEM_5	X	X	SEG23	SEG23	SEG23	SEG23	SEG23	SEG23	X	X	SEG22	SEG22	SEG22	SEG22	SEG22	SEG22	X	X	SEG21	SEG21	SEG21	SEG21	SEG21	SEG21	X	X	SEG20	SEG20	SEG20	SEG20	SEG20	SEG20
LCD_MEM_4	X	X	SEG19	SEG19	SEG19	SEG19	SEG19	SEG19	X	X	SEG18	SEG18	SEG18	SEG18	SEG18	SEG18	X	X	SEG17	SEG17	SEG17	SEG17	SEG17	SEG17	X	X	SEG16	SEG16	SEG16	SEG16	SEG16	SEG16
LCD_MEM_3	X	X	SEG15	SEG15	SEG15	SEG15	SEG15	SEG15	X	X	SEG14	SEG14	SEG14	SEG14	SEG14	SEG14	X	X	SEG13	SEG13	SEG13	SEG13	SEG13	SEG13	X	X	SEG12	SEG12	SEG12	SEG12	SEG12	SEG12
LCD_MEM_2	X	X	SEG11	SEG11	SEG11	SEG11	SEG11	SEG11	X	X	SEG10	SEG10	SEG10	SEG10	SEG10	SEG10	X	X	SEG09	SEG09	SEG09	SEG09	SEG09	SEG09	X	X	SEG08	SEG08	SEG08	SEG08	SEG08	SEG08
LCD_MEM_1	X	X	SEG07	SEG07	SEG07	SEG07	SEG07	SEG07	X	X	SEG06	SEG06	SEG06	SEG06	SEG06	SEG06	X	X	SEG05	SEG05	SEG05	SEG05	SEG05	SEG05	X	X	SEG04	SEG04	SEG04	SEG04	SEG04	SEG04
LCD_MEM_0	X	X	SEG03	SEG03	SEG03	SEG03	SEG03	SEG03	X	X	SEG02	SEG02	SEG02	SEG02	SEG02	SEG02	X	X	SEG01	SEG01	SEG01	SEG01	SEG01	SEG01	X	X	SEG00	SEG00	SEG00	SEG00	SEG00	SEG00

Figure 5.21-2 LCD Memory Map

5.21.4.2 Frame Counter (FC) and Blinking Display

In 6-mux configuration, COM0, COM1, COM2, COM3, COM4 and COM5 organize one frame. In 5-mux configuration, COM0, COM1, COM2, COM3 and COM4 organize one frame. In 4-mux configuration, COM0, COM1, COM2 and COM3 organize one frame. In 3-mux configuration, COM0, COM1 and COM2 organize one frame. In 2-mux configuration, COM0 and COM1 organize one frame. In static configuration, COM0 organizes one frame. The frame counter can be pre-scaled by programming pre-scale counter (LCD_FCR[PRESCL]). The pre-scale counter can be divided by 1, 2, 4 and 8. The frame counter is counted down from FCV (LCD_FCR[9:4]) to zero. FCV is the top value of frame counter. If FCEN (LCD_FCR[0]) is set to 1 and FCINTEN (LCD_FCR[1]) is set to 1, once frame counter is counted down to zero, the frame counter overflow interrupt is generated. At the same time, the frame counter is reloaded with FCV automatically. The LCD blinking display is controlled with frame counter overflow time. In blinking configuration, the segments are turned on and turned off alternately by frame counter overflow time. The frame counter overflowing interrupt can be also used as synchronization for filling data to LCD display memory register.

5.21.4.3 LCD Display Power Down

If the power-down request is triggered from system manager, LCD controller will execute the frame completely to avoid the DC component. When the frame is executed completely, the LCD power down interrupt signal is generated to inform system manager the LCD controller is ready to enter power down state, if PDINT_EN (LCD_CTL[9]) is enabled.Otherwise, if PDINT_EN (LCD_CTL[9]) is disabled, the LCD power down interrupt signal is blocked and the interrupt is disabled. If the PDDSIP_EN (LCD_CTL[8]) is set to 1, the LCD display is operated in Power-down mode. Otherwise, if PDDSIP_EN (LCD_CTL[8]) is cleared to 0, the LCD display is off in Power-down mode.

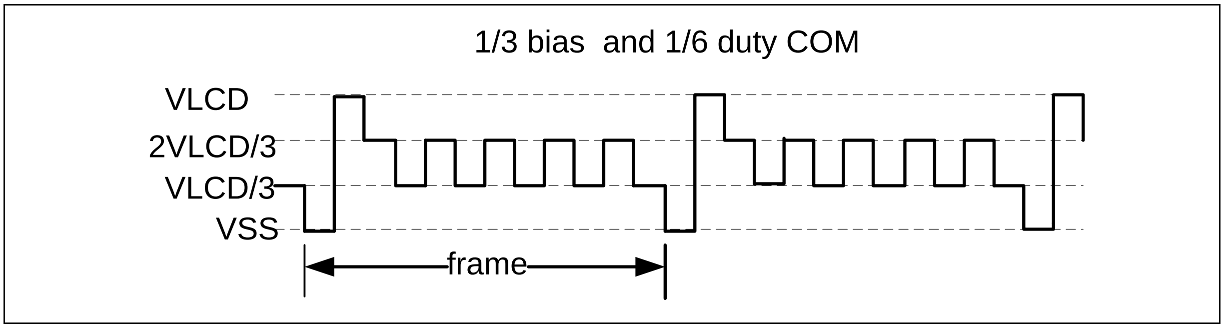


Figure 5.21-3 COM Signal Waveform

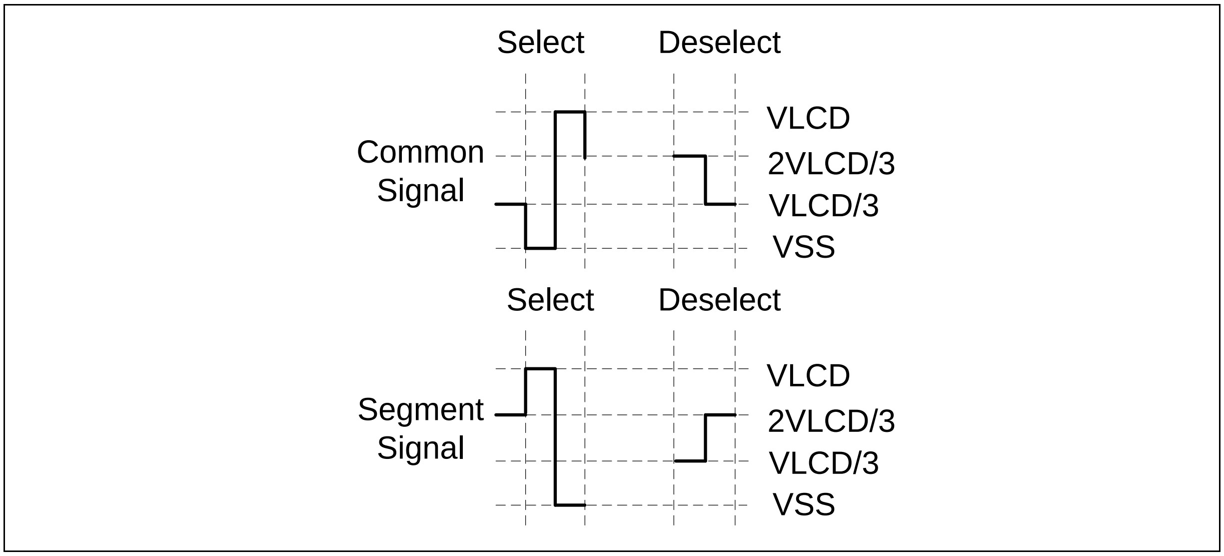


Figure 5.21-4 SEG Signal Waveform

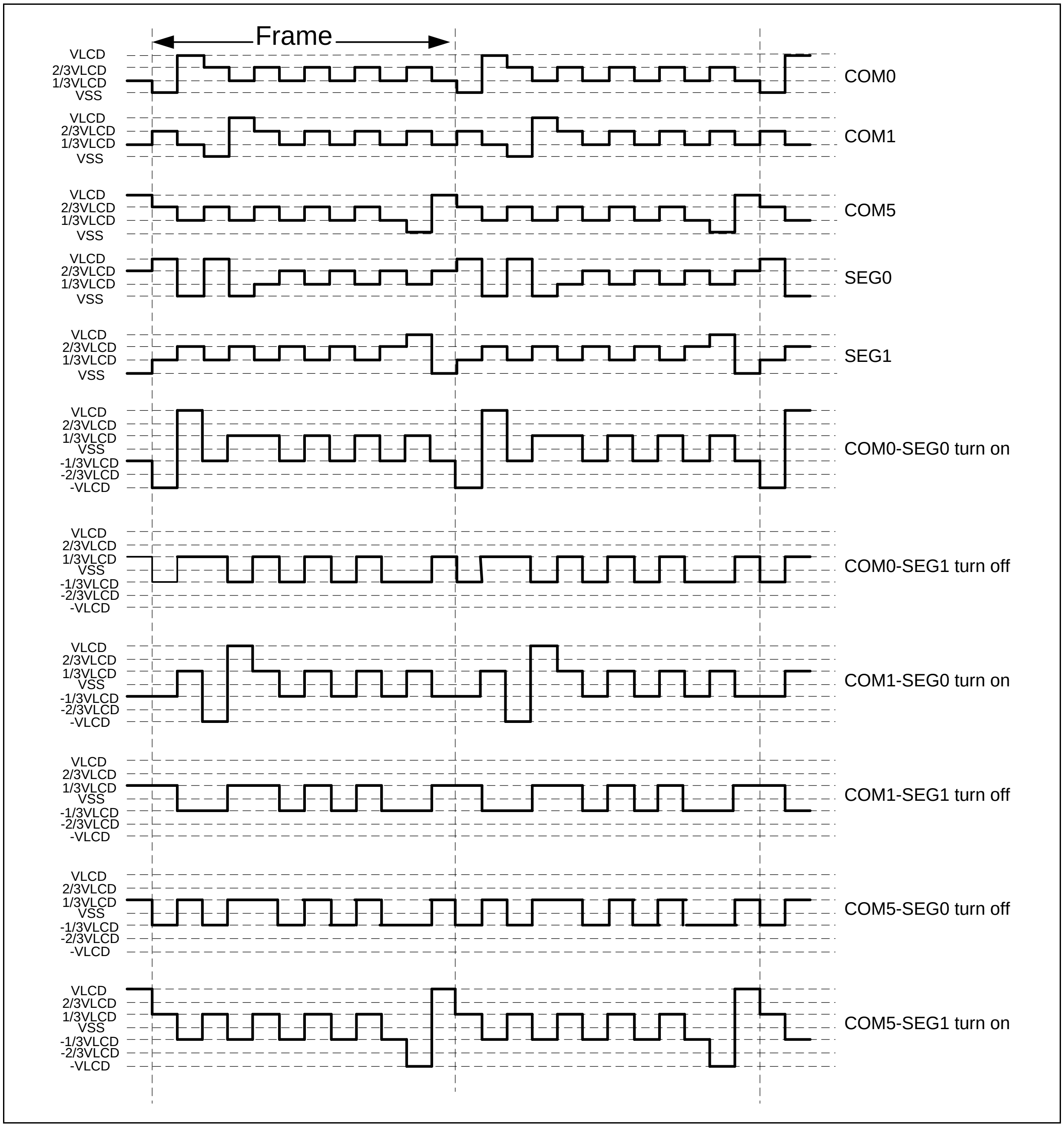


Figure 5.21-5 COM-SEG Signal Waveform by 1/6 Duty with 1/3 Bias

5.21.5 Register and Memory Map

R: read only, **W**: write only, **R/W**: both read and write

Register	Offset	R/W	Description	Reset Value
LCD Base Address: **LCD_BA = 0x400B_0000**				
LCD_CTL	LCD_BA+0x00	R/W	LCD Control Register	0x0000_0000
LCD_DISPCTL	LCD_BA+0x04	R/W	LCD Display Control Register	0x0000_0000
LCD_MEM_0	LCD_BA+0x08	R/W	LCD SEG3 ~ SEG0 data	0x0000_0000
LCD_MEM_1	LCD_BA+0x0C	R/W	LCD SEG7 ~ SEG4 data	0x0000_0000
LCD_MEM_2	LCD_BA+0x10	R/W	LCD SEG11 ~ SEG8 data	0x0000_0000
LCD_MEM_3	LCD_BA+0x14	R/W	LCD SEG15 ~ SEG12 data	0x0000_0000
LCD_MEM_4	LCD_BA+0x18	R/W	LCD SEG19 ~ SEG16 data	0x0000_0000
LCD_MEM_5	LCD_BA+0x1C	R/W	LCD SEG23 ~ SEG20 data	0x0000_0000
LCD_MEM_6	LCD_BA+0x20	R/W	LCD SEG27 ~ SEG24 data	0x0000_0000
LCD_MEM_7	LCD_BA+0x24	R/W	LCD SEG31 ~ SEG28 data	0x0000_0000
LCD_MEM_8	LCD_BA+0x28	R/W	LCD SEG35 ~ SEG32 data	0x0000_0000
LCD_MEM_9	LCD_BA+0x2C	R/W	LCD SEG39 ~ SEG36 data	0x0000_0000
LCD_FCR	LCD_BA+0x30	R/W	LCD frame counter control register	0x0000_0000
LCD_FCSTS	LCD_BA+0x34	R/W	LCD frame counter status	0x0000_0000

5.21.6 Register Description

LCD Control Register (LCD_CTL)

Register	Offset	R/W	Description	Reset Value
LCD_CTL	LCD_BA+0x00	R/W	LCD Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved						PDINT_EN	PDDISP_EN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
BLINK	FREQ			MUX			EN

Bits	Description	
[31:10]	**Reserved**	**Reserved**
[9]	**PDINT_EN**	**Power Down Interrupt Enable** If the power down request is triggered from system management, LCD controller will execute the frame completely to avoid the DC component. When the frame is executed completely, the LCD power down interrupt signal is generated to inform system management that LCD controller is ready to enter power down state, if PDINT_EN is set to 1. Otherwise, if PDINT_EN is set to 0, the LCD power down interrupt signal is blocked and the interrupt is disabled to send to system management. 0 = Power Down Interrupt Disabled 1 = Power Down Interrupt Enabled
[8]	**PDDISP_EN**	**Power Down Display Enable** The LCD can be programmed to be displayed or not be displayed at power down state by PDDISP_EN setting. 0 = LCD display Disabled (LCD is put out) at power down state 1 = LCD display Enabled (LCD keeps the display) at power down state
[7]	**BLINK**	**LCD Blinking Enable** 0 = Blinking Disabled. 1 = Blinking Enabled.
[6:4]	**FREQ**	**LCD Frequency Selection** 000 = LCDCLK Divided by 32 001 = LCDCLK Divided by 64 010 = LCDCLK Divided by 96 011 = LCDCLK Divided by 128 100 = LCDCLK Divided by 192

		101 = LCDCLK Divided by 256 110 = LCDCLK Divided by 384 111 = LCDCLK Divided by 512
[3:1]	**MUX**	**Mux select** 000 = Static 001 = 1/2 duty 010 = 1/3 duty 011 = 1/4 duty 100 = 1/5 duty 101 = 1/6 duty 110 = Reserved 111 = Reserved **Note :** User does not need to set PD_H_MFP bit field, but only to set the MUX bit field to switch LCD_SEG0 and LCD_SEG1 to LCD_COM4 and LCD_COM5 for Nano110 and Nano130 series.

LCD_SEG0(LCD_COM4) and LCD_SEG1(LCD_COM5) Pins Definition
for Setting MUX Bit Field for Nano110 and Nano130 Series

MUX	**LQFP128**		**LQFP64**	
	SEG0(COM4) for PD.15	SEG1(COM5) for PD.14	SEG0(COM4) for PB.1	SEG1(COM5) for PB.0
0b000, 0b001 0b010, 0b011	LCD_SEG0	LCD_SEG1	LCD_SEG0	LCD_SEG1
0b100	LCD_COM4	LCD_SEG1	LCD_COM4	LCD_SEG1
0b101	LCD_COM4	LCD_COM5	LCD_COM4	LCD_COM5

[0]	**EN**	**LCD Enable** 0 = LCD controller operation Disabled. 1 = LCD controller operation Enabled.

LCD Display Control Register (LCD_DISPCTL)

Register	Offset	R/W	Description	Reset Value
LCD_DISPCTL	LCD_BA+0x04	R/W	LCD Display Control Register	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved		CPUMP_FREQ			CPUMP_VOL_SET		
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved	BV_SEL	Reserved	IBRL_EN	Reserved	BIAS_SEL		CPUMP_EN

Bits	Description	
[31:14]	Reserved	Reserved
[13:11]	CPUMP_FREQ	**Charge Pump Frequency Selection** 000 = LCDCLK 001 = LCDCLK/2 010 = LCDCLK/4 011 = LCDCLK/8 100 = LCDCLK/16 101 = LCDCLK/32 110 = LCDCLK/64 111 = LCDCLK/128
[10:8]	CPUMP_VOL_SET	**Charge Pump Voltage Selection** 000 = 2.7V 001 = 2.8V 010 = 2.9V 011 = 3.0V 100 = 3.1V 101 = 3.2V 110 = 3.3V 111 = 3.4V
[7]	Reserved	Reserved
[6]	BV_SEL	**Bias Voltage Type Selection** 0 = C-Type bias mode. Bias voltage source from internal bias generator. 1 = R-Type bias mode. Bias voltage source from external bias generator. **Note:** The external resistor ladder should be connected to the V1 pin, V2 pin, V3 pin and

		V_{SS}. The V_{LCD} pin should also be connected to V_{DD}.
[5]	**Reserved**	**Reserved**
[4]	**IBRL_EN**	**Internal Bias Reference ladder Enable** 0 = Bias reference ladder Disabled. 1 = Bias reference ladder Dnabled.
[3]	**Reserved**	**Reserved**
[2:1]	**BIAS_SEL**	**Bias Selection** 00 = Static 01 = 1/2 Bias 10 = 1/3 Bias 11 = Reserved
[0]	**CPUMP_EN**	**Charge Pump Enable** 0 = Disabled. 1 = Enabled.

LCD MEM_x Register (LCD_MEMORY_x) (x= 0 ~ 9)

Register	Offset	R/W	Description	Reset Value
LCD_MEM_0	LCD_BA+0x08	R/W	LCD SEG3 ~ SEG0 data	0x0000_0000
LCD_MEM_1	LCD_BA+0x0C	R/W	LCD SEG7 ~ SEG4 data	0x0000_0000
LCD_MEM_2	LCD_BA+0x10	R/W	LCD SEG11 ~ SEG8 data	0x0000_0000
LCD_MEM_3	LCD_BA+0x14	R/W	LCD SEG15 ~ SEG12 data	0x0000_0000
LCD_MEM_4	LCD_BA+0x18	R/W	LCD SEG19 ~ SEG16 data	0x0000_0000
LCD_MEM_5	LCD_BA+0x1C	R/W	LCD SEG23 ~ SEG20 data	0x0000_0000
LCD_MEM_6	LCD_BA+0x20	R/W	LCD SEG27 ~ SEG24 data	0x0000_0000
LCD_MEM_7	LCD_BA+0x24	R/W	LCD SEG31 ~ SEG28 data	0x0000_0000
LCD_MEM_8	LCD_BA+0x28	R/W	LCD SEG35 ~ SEG32 data	0x0000_0000
LCD_MEM_9	LCD_BA+0x2C	R/W	LCD SEG39 ~ SEG36 data	0x0000_0000

31	30	29	28	27	26	25	24
Reserved		SEG_3_4x	SEG_3_4x	SEG_3_4x	SEG_3_4x	SEG_3_4x	SEG_3_4x
23	22	21	20	19	18	17	16
Reserved		SEG_2_4x	SEG_2_4x	SEG_2_4x	SEG_2_4x	SEG_2_4x	SEG_2_4x
15	14	13	12	11	10	9	8
Reserved		SEG_1_4x	SEG_1_4x	SEG_1_4x	SEG_1_4x	SEG_1_4x	SEG_1_4x
7	6	5	4	3	2	1	0
Reserved		SEG_0_4x	SEG_0_4x	SEG_0_4x	SEG_0_4x	SEG_0_4x	SEG_0_4x

Bits	Description	
[31:30]	Reserved	Reserved
[29:24]	SEG_3_4x data	SEG_3_4x DATA for COM (x = 0 ~ 9)
[23:22]	Reserved	Reserved
[21:16]	SEG_2_4x data	SEG_2_4x DATA for COM(x= 0 ~ 9)
[17:15]	Reserved	Reserved
[14:8]	SEG_1_4x data	SEG_1_4x DATA for COM(x= 0 ~ 9)
[7:6]	Reserved	Reserved
[5:0]	SEG_0_4x data	SEG_0_4x DATA for COM(x= 0 ~ 9)

LCD Frame Counter Register (LCD_FCR)

Register	Offset	R/W	Description	Reset Value
LCD_FCR	LCD_BA+0x30	R/W	LCD frame counter control register	0x0000_0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td colspan="6">Reserved</td><td colspan="2">FCV</td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="4">FCV</td><td colspan="2">PRESCL</td><td>FCINTEN</td><td>FCEN</td></tr>
</table>

Bits	Description	
[31:10]	**Reserved**	**Reserved**
[9:4]	**FCV**	**Frame Counter Top Value** These 6 bits contain the top value of the Frame counter.
[3:2]	**PRESCL**	**Frame Counter Pre-scaler Value** 00 = CLKframe/1 01 = CLKframe/2 10 = CLKframe/4 11 = CLKframe/8
[1]	**FCINTEN**	**LCD Frame Counter Interrupt Enable** 0 = Frame counter interrupt Disabled. 1 = Frame counter interrupt Enabled.
[0]	**FCEN**	**LCD Frame Counter Enable** 0 = Disabled. 1 = Enabled

LCD Frame Counter Status Register (LCD_INTSTS)

Register	Offset	R/W	Description	Reset Value
LCD_FCSTS	LCD_BA+0x34	R/W	LCD frame counter status	0x0000_0000

31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Reserved						**PDSTS**	**FCSTS**

Bits	Description	
[31:2]	**Reserved**	**Reserved**
[1]	**PDSTS**	**Power-down Interrupt Status** 0 = Inform system manager that LCD controller is not ready to enter power-down state until this bit becomes 1 if power down is set and one frame is not executed completely. 1 = Inform system manager that LCD controller is ready to enter power-down state if power down is set and one frame is executed completely.
[0]	**FCSTS**	**LCD Frame Counter Status** 0 = Frame counter value does not reach FCV (Frame Count TOP value). 1 = Frame counter value reaches FCV (Frame Count TOP value). If the FCINTEN is s enabled, the frame counter overflow Interrupt is generated.

5.21.7 Application Circuit

External Resister ladder

1. Most commonly used for high V_{DD} voltages.
2. Uses inexpensive resistors to create the multilevel LCD voltages. Regardless of the number of pixels that are energized the current remains constant. The voltage at point V_{LCD} is typically tied to V_{DD}, either internally or externally
3. The resister values are determined by two factors.
   a. Display quality
   b. Power consumption

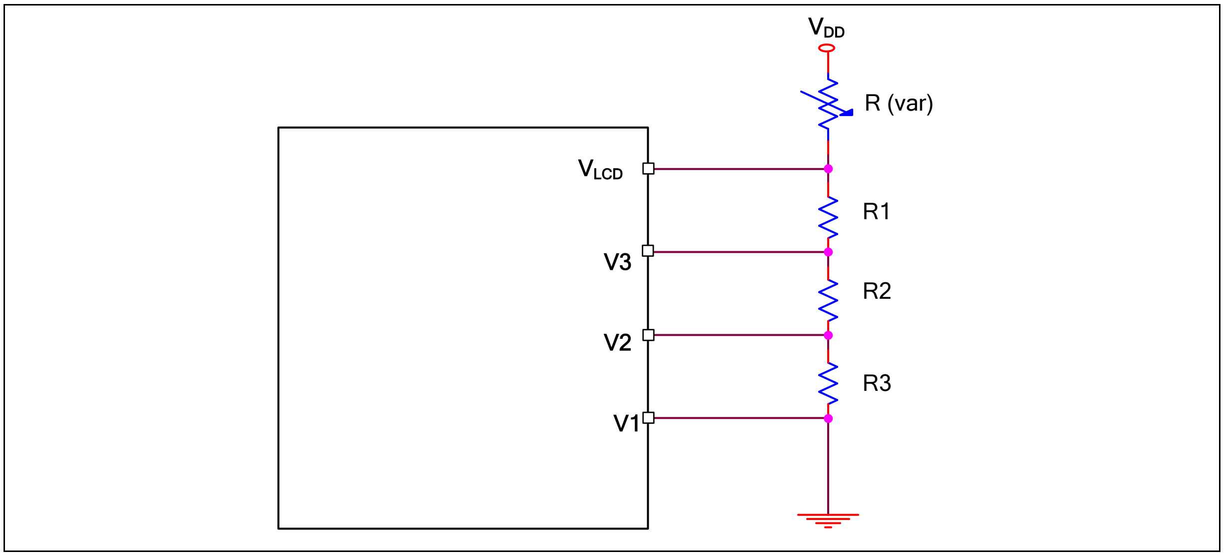


Figure 5.21-6 1/3 Bias (External Resistor Ladder)

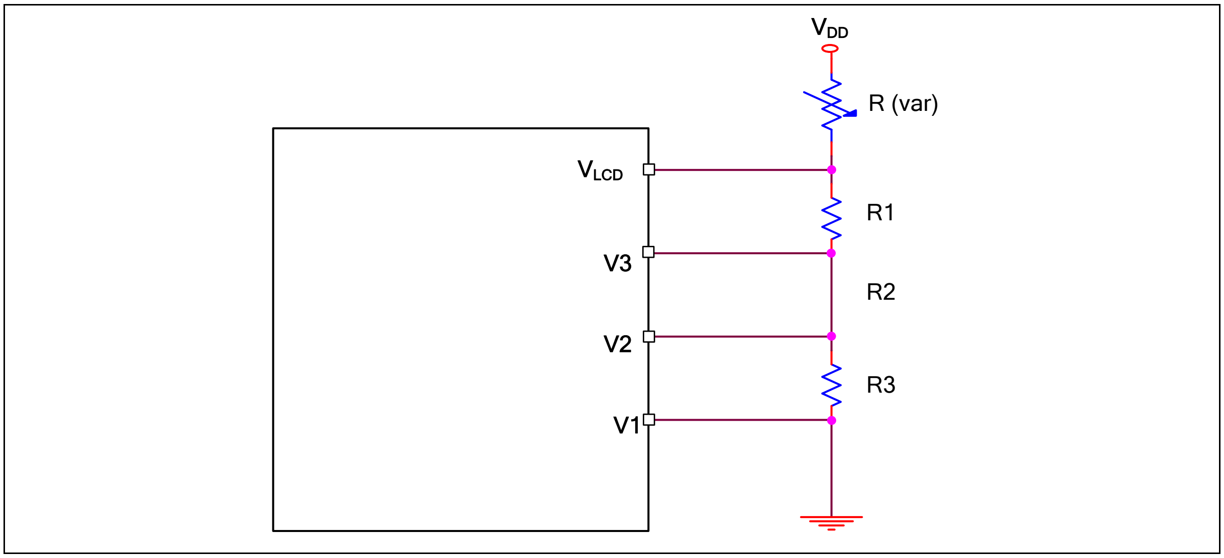


Figure 5.21-7 1/2 Bias (External Resistor Ladder)

Resistor ladder with capacitors

Sometimes the addition of parallel capacitors to the resistance can reduce the distortion caused by charging/discharging currents. This effect is limited since at some point a large resistor and large capacitor cause a voltage level shift which negatively impacts the display quality.

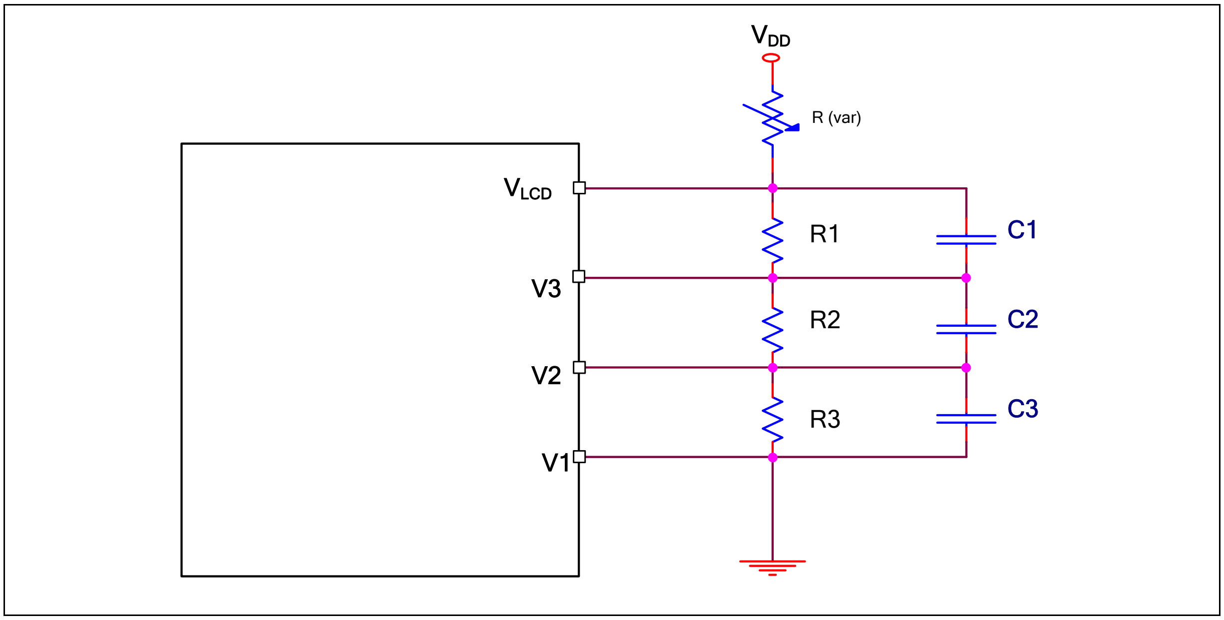


Figure 5.21-8 1/3 Bias (Resistor Ladder with Capacitor)

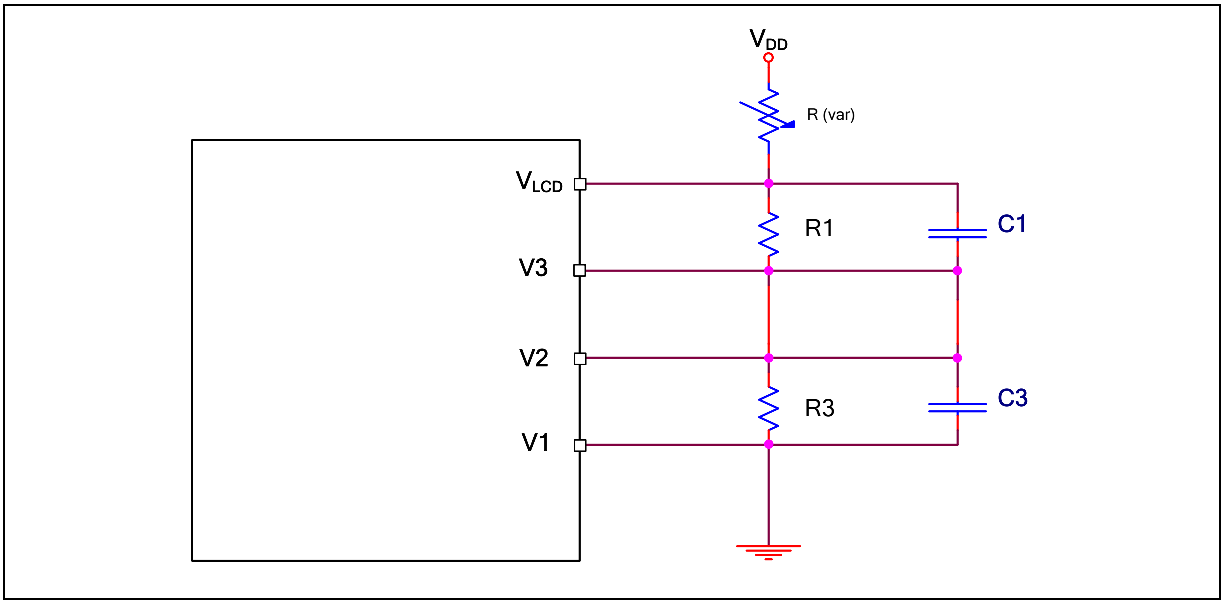


Figure 5.21-9 1/2 Bias (Resistor Ladder with Capacitor)

Charge Pump

1. Ideal for low voltage battery operation because the V_{DD} voltage can be boosted up to drive the LCD panel.

2. The charge pump requires a charging capacitor and filter capacitor for each of the LCD voltages.

3. Another feature that makes the charge pump ideal for battery applications is that the current consumption is proportional to the number of pixels that are energized.

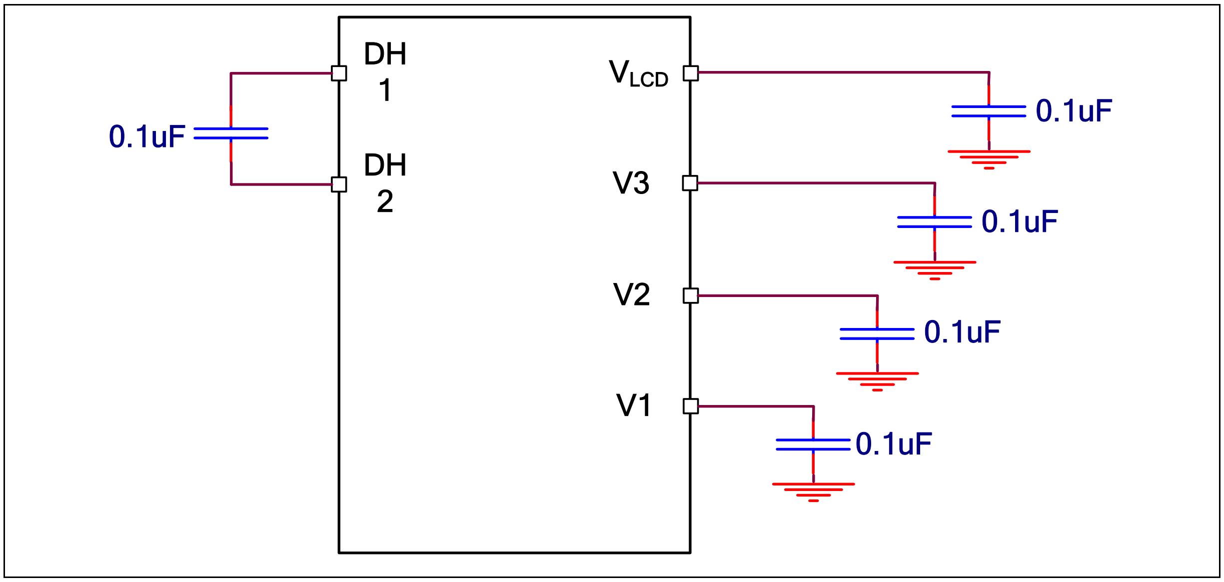


Figure 5.21-10 1/3 Bias (Charge Pump)

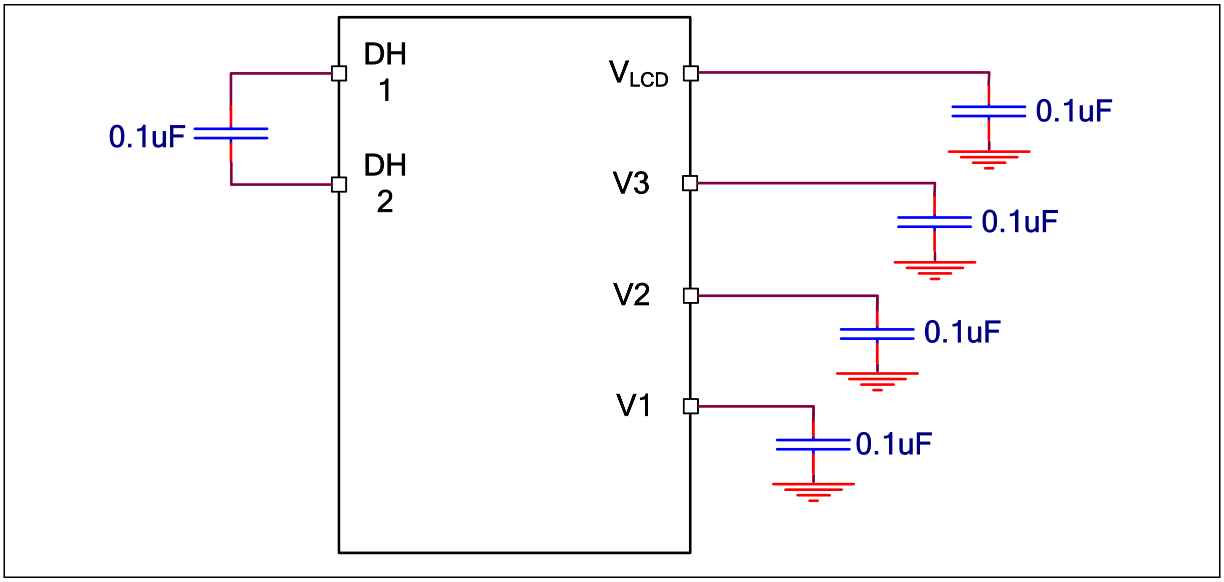


Figure 5.21-11 1/2 Bias (Charge Pump)

5.22 Analog to Digital Converter (ADC)

5.22.1 Overview

This chip contains one 12-bit successive approximation analog-to-digital converter (SAR A/D converter) with 12 external input channels and 6 internal channels. The A/D converter supports three operation modes: Single, Single-cycle Scan and Continuous Scan mode, and can be started by software and external STADC/PB.8 pin and timer event start.

Note that the I/O pins used as ADC analog input pins must be configured as input type and off digital function (GPIOA_OFFD) should be turned on before ADC function is enabled.

5.22.2 Features

- Analog input voltage range: 0~V_{REF} (Max to 3.6V)
- Selectable 12-bits, 10-bits, 8-bits and 6-bits resolution
- Supports sampling time settings (in ADC_CLK unit) for channel 0~11 individually and channel 12~17 share the same one sampling time setting
- Supports two power-down modes:
  - Power-down mode
  - Standby mode
- Up to 12 external analog input channels (channel0 ~ channel11), and 6 internal channels (channel12~channel17) converting six voltage sources, including DAC0, DAC1, internal band-gap voltage, internal temperature sensor output, AV_{DD}, and AV_{SS}.
- Maximum ADC clock frequency is 42 MHz and each conversion is 19 clocks+ sampling time depending on the input resistance.
- Three operating modes
  - Single mode: A/D conversion is performed one time on a specified channel.
  - Single-cycle Scan mode: A/D conversion is performed one cycle on all specified channels with the sequence from the lowest numbered channel to the highest numbered channel.
  - Continuous Scan mode: A/D converter continuously performs Single-cycle scan mode until software stops A/D conversion.
- An A/D conversion can be started by:
  - Software write 1 to ADST bit
  - External pin STADC
  - Selects one from four timer events (TMR0, TMR1, TMR2 and TMR3) that enable ADC and transfer AD results by PDMA
- Conversion results held in data registers for each channel
- Conversion result can be compared with a specified value and user can select whether to generate an interrupt when conversion result is equal to the compare register setting.
- Supports Calibration and load Calibration words capability.

5.22.3 Block Diagram

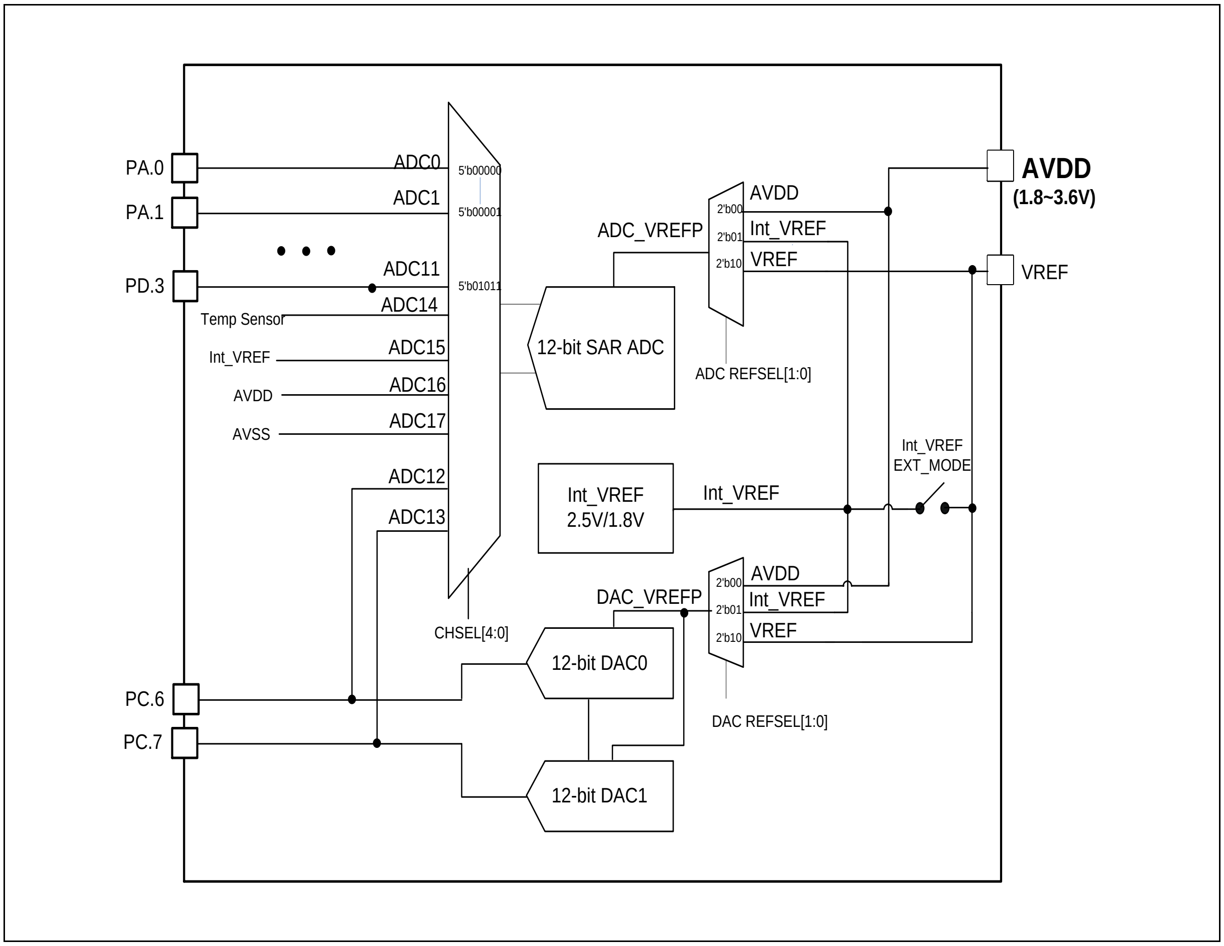


Figure 5.22-1 ADC and DAC Block Diagram

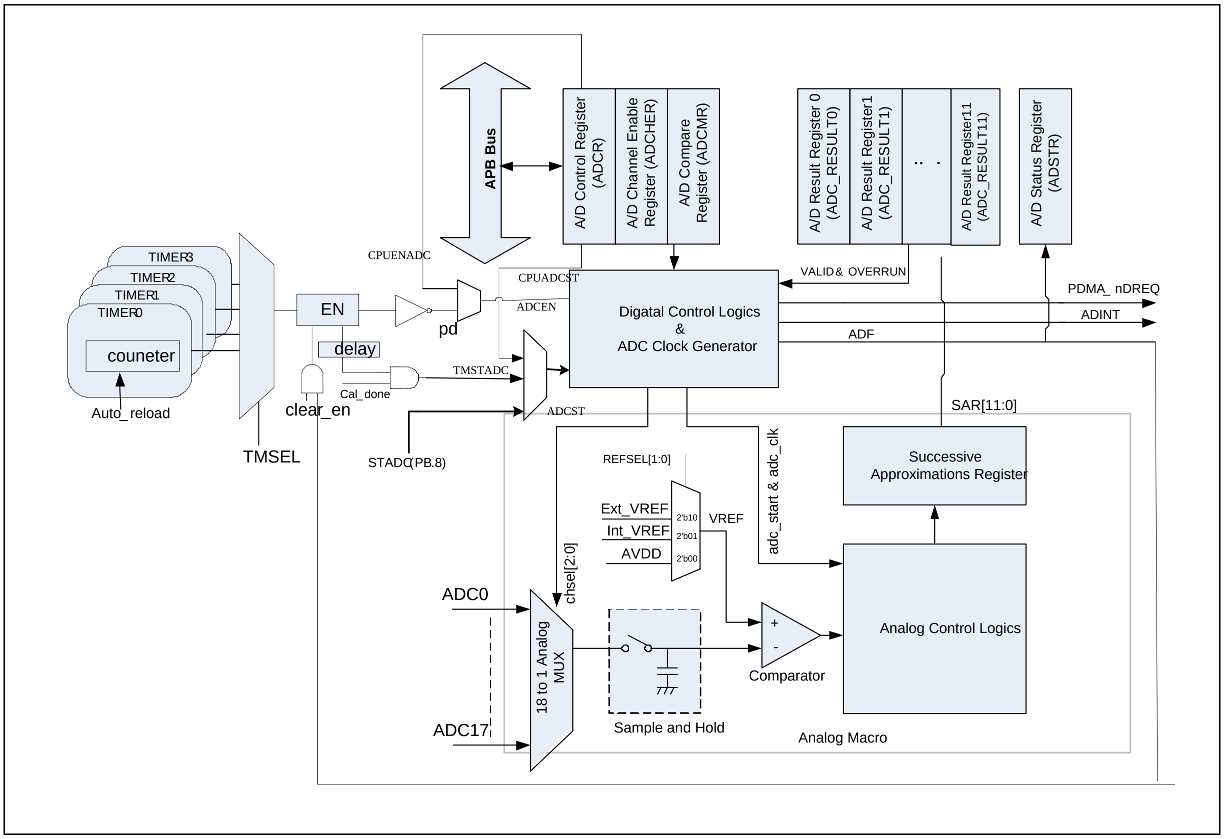


Figure 5.22-2 ADC Controller Block Diagram

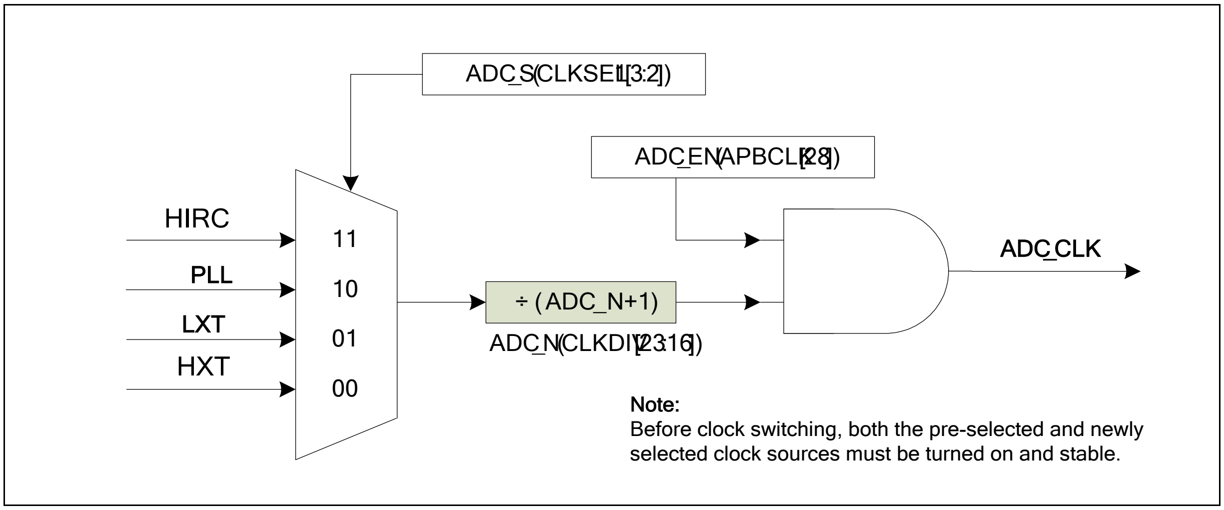


Figure 5.22-3 ADC Clock Control

5.22.4 Functional Description

The A/D converter is operated by successive approximation with 12-bit resolution. The ADC has three operation modes: Single mode, Single-cycle Scan mode and Continuous Scan mode. When changing the operating mode or analog input channel enabled, in order to avoid incorrect operation, software must clear the ADST bit to 0 in ADCR register. After the operation, the A/D converter discards current conversion and enters idle state while ADST bit is cleared.

In some applications for saving power, ADC can be enabled by a time-out (TMRx Chy) signal and start A/D conversion after a delay time interval and enter power-down state after converting fixed amount of conversion data transferred to memory through PDMA, There are four time-out source(Timer0~3) to enable ADC by setting TMSEL[1:0] in ADCON register.

5.22.4.1 Single Mode

In single mode, A/D conversion is performed only once on the specified single channel. The operations are as follows.

1. A/D conversion is started when the ADST bit in ADCR is set to 1 either by software or by external trigger input or by timer event selected by TMSEL[1:0] in ADCR register .
2. When A/D conversion is finished, the result is stored in the A/D data register corresponding to the channel.
3. On completion of conversion, the ADF bit in ADSR is set to 1 and ADC interrupt (ADINT) is requested if the ADIE bit is set to 1.
4. The ADST bit remains 1 during A/D conversion. When A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters in idle state. If the ADST bit is cleared to 0 by software during A/D conversion, A/D conversion will stop and enter in idle state.

After the previous conversion is complete, repeat method 1-4 to the next conversion PS: if the ADC clock is much lower than PCLK, wait for at least one ADC clock to start the next conversion

Note: If software enables more than one channel in single mode, the least channel is converted and other enabled channels will be ignored.

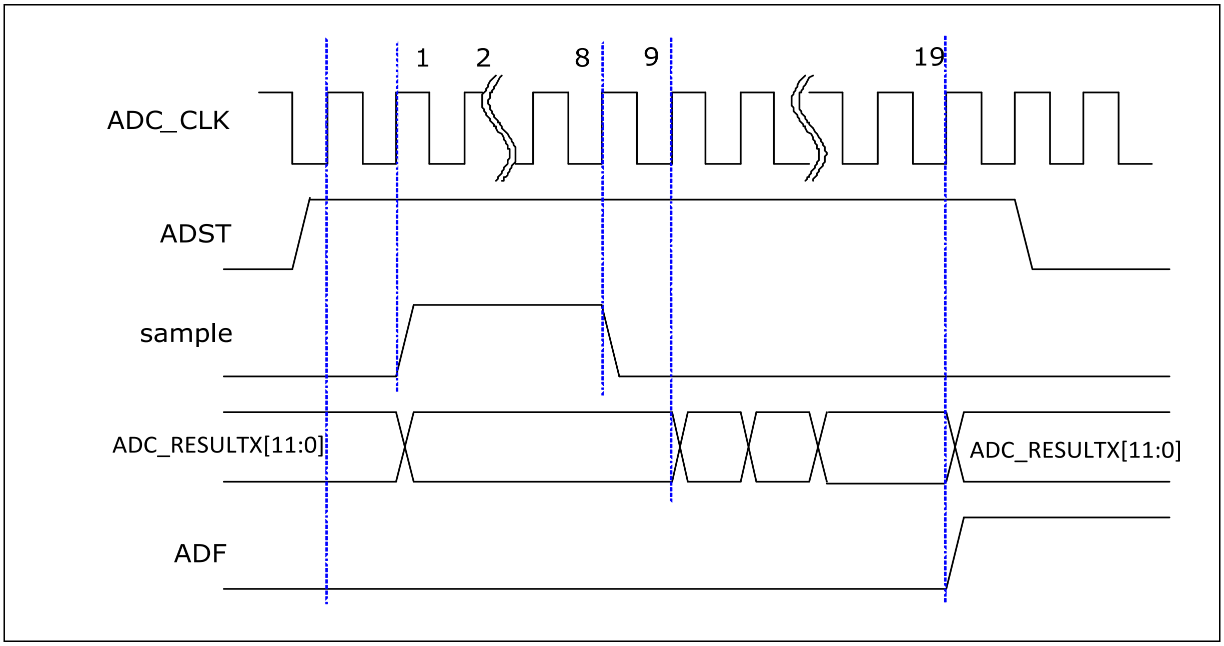


Figure 5.22-4 ADC Single Mode Conversion Timing Diagram

5.22.4.2 Single-Cycle Scan Mode

In single-cycle scan mode, A/D conversion will sample and convert the specified channels once in the sequence from the least numbered channel to the highest numbered channel. Operations are described as follows.

1. When the ADST bit in ADCR is set to 1 by software or by an external trigger input or by timer event selected by TMSEL[1:0] in ADCR register, A/D conversion starts on the lowest numbered channel.
2. When A/D conversion for each enabled channel is completed, the result is sequentially transferred to the A/D data register corresponding to each channel.
3. When conversions of all the enabled channels are completed, the ADF bit in ADSR is set to 1. If the ADIE bit is set to 1 at this time, an ADINT interrupt is set after A/D conversion ends.
4. After A/D conversion ends, the ADST bit is automatically cleared to 0 and the A/D converter enters in idle state. If the ADST bit is cleared to 0 by software during A/D conversion, A/D conversion will stop after current conversion complete and enter in idle state.

An example timing diagram for single-cycle scan is shown below:

(In this example, channel 0,2,3 and 7 are enabled.)

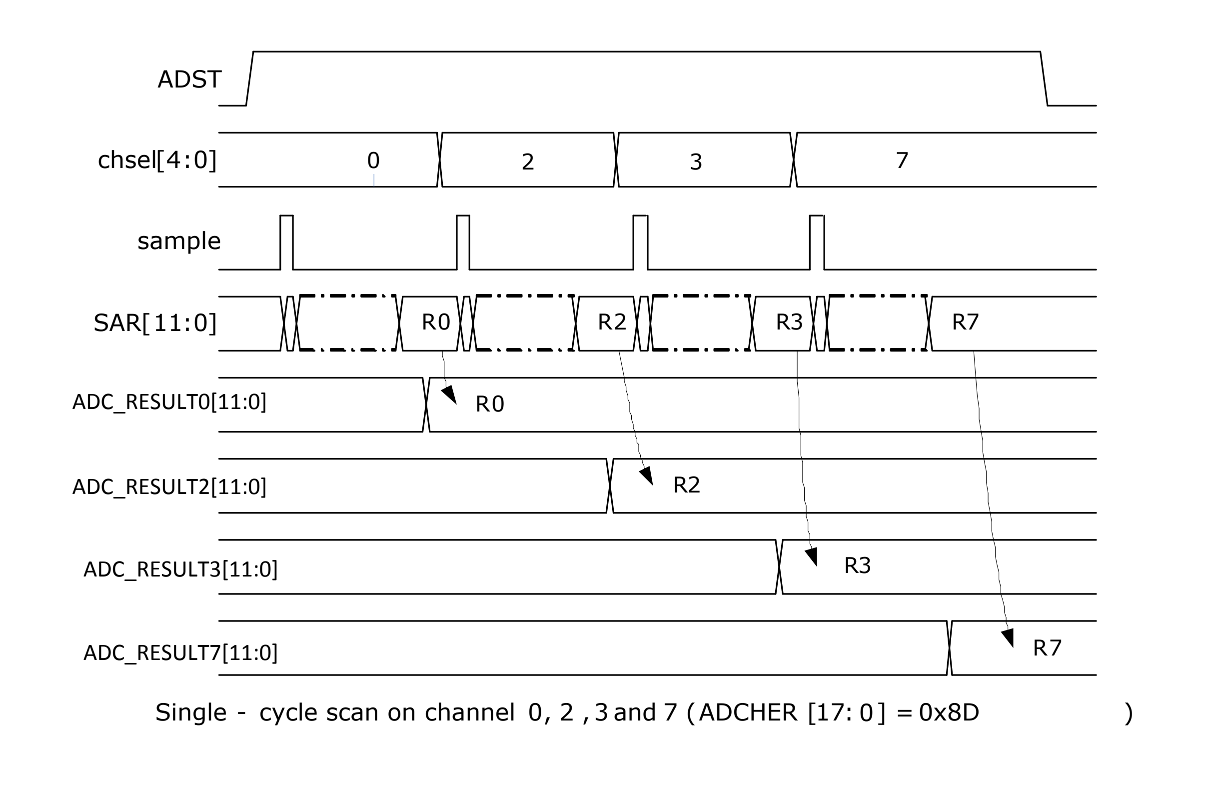


Figure 5.22-5 ADC Single-cycle Scan on Enabled Channels Timing Diagram

5.22.4.3 Continuous Scan Mode

In continuous scan mode, A/D conversion is performed sequentially on the specified channels that enabled by CHEN bits in ADCHER register (maximum 8 external channels and one internal channel for ADC and internal DAC0, DAC1). The operations are as follows.

1. When the ADST bit in ADCR is set to 1 by software or external trigger input or by timer event selected by TMSEL[1:0] in ADCR register, A/D conversion starts on the channel with the lowest number.

2. When A/D conversion for each enabled channel is completed, the result of each enabled channel is stored in the A/D data register corresponding to each enabled channel.
3. Once when all of the enabled channel sequentially completes A/D converting, the ADF bit (ADSR[0]) will be set to 1. If the ADIE bit is set to 1 at this time, an ADINT interrupt is set after A/D conversion ends.
4. Following the step 3, conversion of the first enabled channel starts again.
5. Steps 2 to 4 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops and the A/D converter enters the idle state.

An example timing diagram for continuous scan on enabled channels (0, 2, 3 and 7) is shown below:

(In this example, channel 0, 2, 3 and 7 are enabled.)

(This example is only appropriate for ADC.)

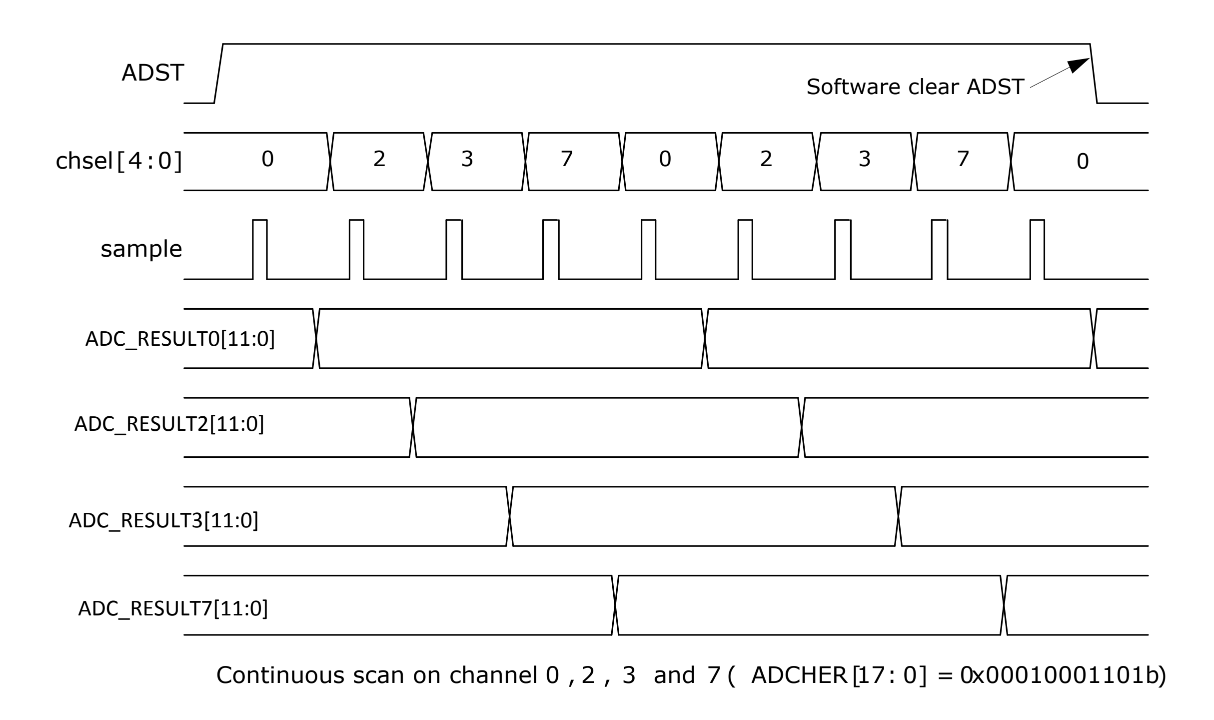


Figure 5.22-6 ADC Continuous Scan on Enabled Channels Timing Diagram

5.22.4.4 ADC Started by External Rriggering

A/D conversion can be triggered by external pin request. When the ADCR.TRGEN is set to high to enable ADC external trigger function, setting the TRGS[1:0] bits to 00b is to select external trigger input from the STADC pin. Software can set TRGCOND[1:0] to select trigger condition is falling/rising edge or low/high level. An 8-bit sampling counter is used to deglitch. If level trigger condition is selected, the STADC pin must be kept at defined state at least 8 PCLKs. The ADST bit will be set to 1 [a]t the 9th PCLK and start to conversion. In level trigger mode conversion is continuous as long as the external trigger input is in asserted state if external trigger input is pull at low (or high state). It is stopped only when external condition trigger condition disappears. If edge trigger condition is selected, the high and low state must be kept at least 4 PLCKs. When a trigger signal with pulse width smaller than the specified width (4 PCLKs), conversion is not triggered.

5.22.4.5 Conversion Result Monitor by Compare Mode

The ADC controller provides two sets of compare register ADCMPR0 and ADCMPR1 to monitor at most two specified channel conversion results from A/D conversion module, refer to Figure 5.22-7. Software can select which channel to be monitored by set CMPCH(ADCMPRx[5:]0]) and CMPCOND bit is used to check conversion result is either less than or greater than (equal to) the specified value in CMPD[11:0]. When the conversion of the channel specified by CMPCH is completed, the comparing action will be triggered one time automatically. When the compare result meets the setting, compare match counter will increase by 1, Once the counter value reaches the setting of (CMPMATCNT+1),CMPF bit will be set to 1, If CMPIE bit is set, an ADINT interrupt request is generated. Software can use this function to monitor whether an external analog input voltage traverse the specified threshold in scan mode without imposing a load on software. Detailed logics diagram is shown below.

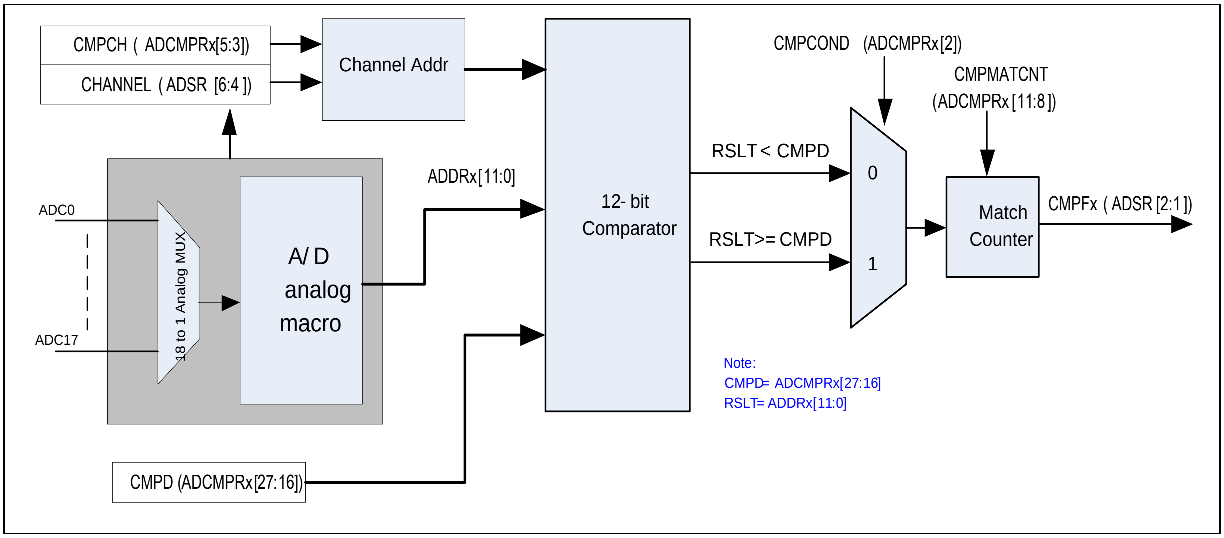


Figure 5.22-7 ADC Conversion Result Monitor Logic Diagram

5.22.4.6 Interrupt Sources

The A/D converter generates a conversion end flag, ADF in ADSR register at the ending moment of A/D conversion. If ADIE bit in ADCR is set, the conversion end interrupt is asserted via ADINT occurs. If CMPIE bit is enabled and A/D conversion result meets the setting in ADCMPR register, monitor interrupt occurs, and ADINT will be set also. CPU can clear CMPF and ADF to stop interrupt request.

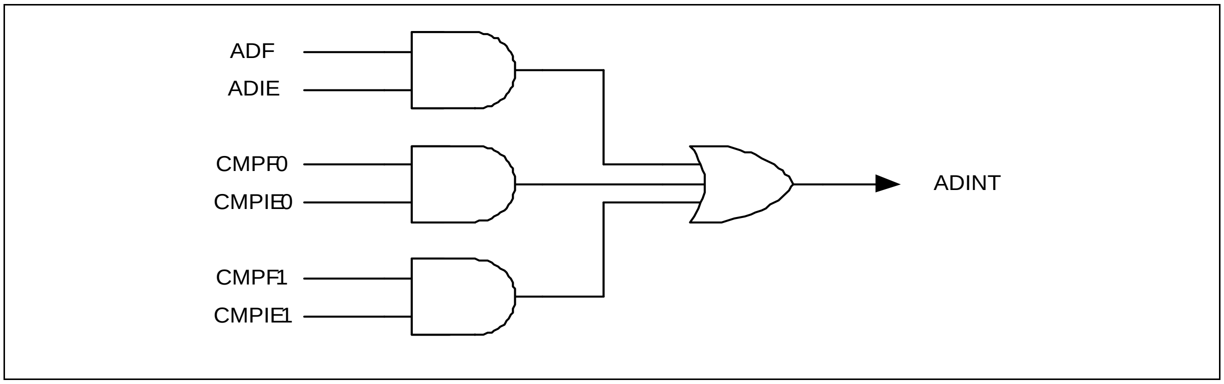


Figure 5.22-8 ADC Controller Interrupt

5.22.4.7 Peripheral DMA Request

When A/D conversion is finished, the converted result is loaded into AD_RESULTx(x=0~10) register and VALID bit is set to 1. If PTEN bit in ACDR is set, ADC controller will generate PDMA request to ask a data transfer. Having the converted result read by PDMA in response to PDMA request enables continuous conversion to be achieved without CPU intervention.

5.22.4.8 ADC Enabled by Timer Event

Users can configure ADC to use timer trigger function by programming TMSEL,TMTRGMOD and TMPDMACNT in ADCR register. If AD is power down, timer event can enable ADC. TMSEL in ADCR register selects timer event source.

After ADC wakes up, it starts to transfer and pass the ADC_RESUT to memory by PDMA. User should configure PTEN in ADCR register to enable PDMA transfer and configure ADMD to run ADC in continuous, single or single cycle mode; and configure TMPDMACNT in ADCR register to specify the amount of ADC_RESULT that PDMA will deliver to memory each time the timer event occurs. After PDMA have delivered the amount of ADC_RESULT specified in TMPDMACNT, ADC will go to power down until the next timer event coming.

After the total amount of ADC_RESULT configured in PDMA byte count register have been delivered to memory, ADC will go to power down, this time the ADC will not be waken up by the following timer event any more.

All the configurations should be done before the system entering power down because CPU can't read and write register while the whole system is power down. In single-cycle and single mode, PDMA transfer count should be exact the same with enabled channel count.

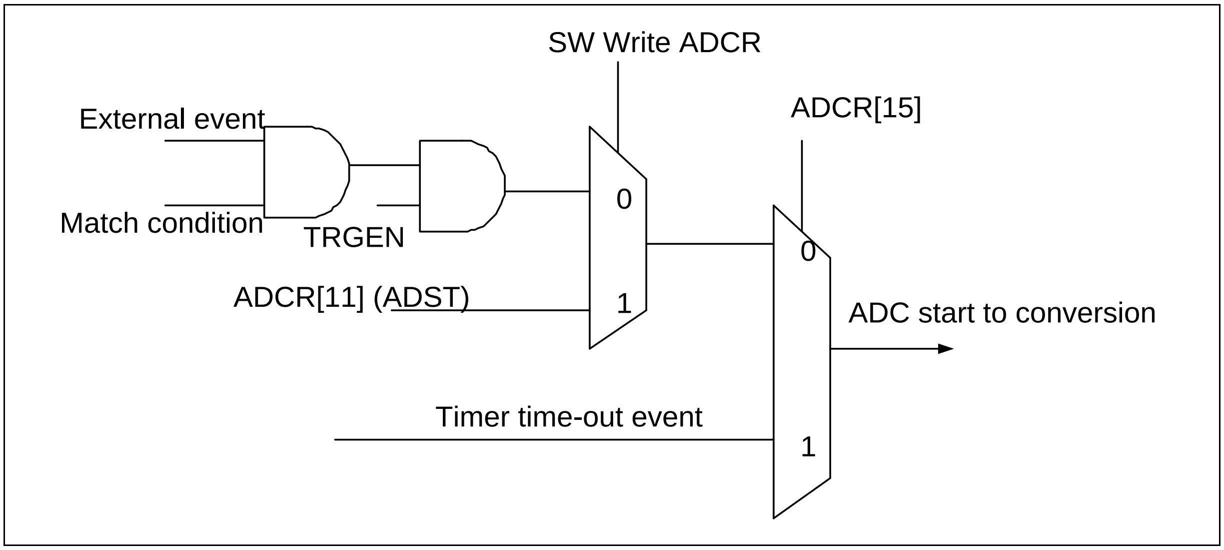


Figure 5.22-9 ADC Start Conversion Conditions

5.22.4.9 ADC Sampling Time

The figure below shows the (simplified) equivalent circuit of the S/H (sample and hold) input network, where C_S is the storage capacitor, R_S is the resistance of the sampling switch and R_I is the output impedance of the signal source V_I. The Figure 5-16 shows the situation where the conversion cycle j+1 starts immediately after conversion cycle n ends. In this case the duration of the sampling phase is, approximately, 1.5 x ADC_CLK. C_S must be charged in that phase, and it must be ensured that the voltage at its terminals becomes sufficiently near V_I. To guarantee this, R_I may not take arbitrarily large values.

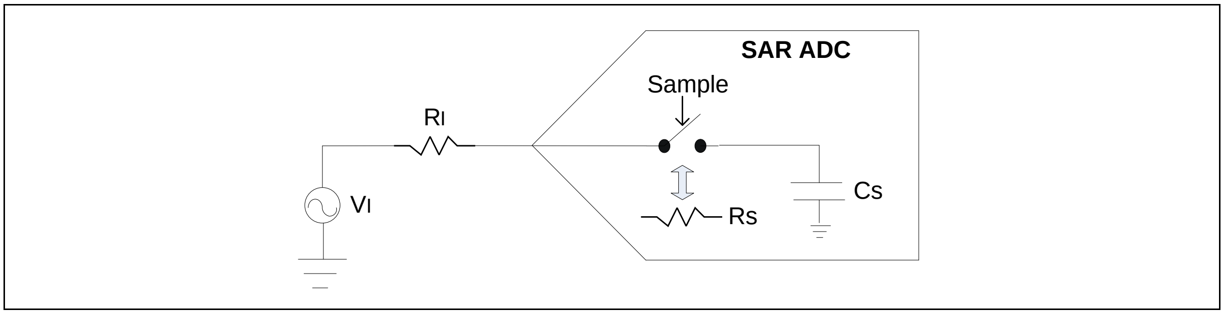


Figure 5.22-10 Model of the sampling network

Figure 5.22-11 shows how the sampling time can be increased, to allow the operation with signal sources having a low driving capability: the adc_start signal is delayed during the number of clock cycles necessary to guarantee the accurate input signal sampling. During this period the chsel must remain unchanged. Note that this operation reduces the effective sampling rate.

The ADC has two types of inputs: channel 0~5 are fast inputs and channel 6~17 are slow inputs. All inputs have the same functionality, but during the sampling period, channel 0~5 inputs are faster than channel 6~17.

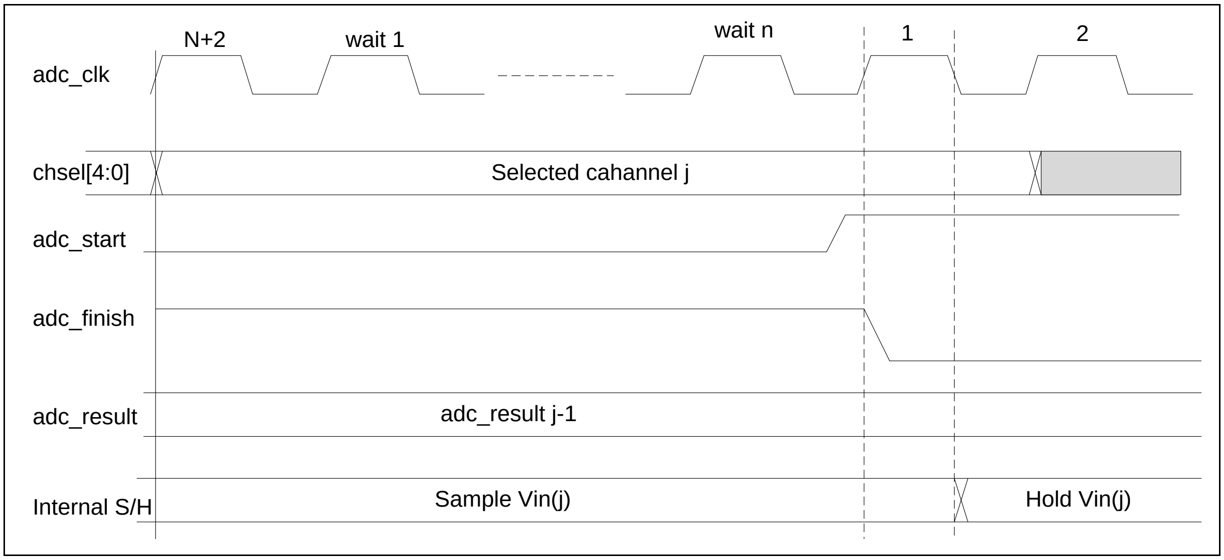


Figure 5.22-11 Increased Sampling Time Waveform

For both types of inputs, it is possible to achieve the maximum sampling frequency, but under certain conditions (depending either from the resolution mode (*RESSEL*) or from the output impedance of the signal source) the sampling period should be delayed during the necessary clock cycles to guarantee the sampling precision (above figure).

The following graphics indicates the number of additional sample and hold cycles (n), necessary for a wide range of Ri (signal source output resistance) values, for all ADC input channels depending on the resolution mode (*RESSEL*). Use sampling counter registers (ADCCHSAMP0 and ADCCHSAMP1) to add additional sample and hold cycle (n).

Please note that these graphics refer to the additional sampling and hold clock cycles (i.e. in the situations where n=0, the sampling period is 1.5 × *ADC_CLK*).

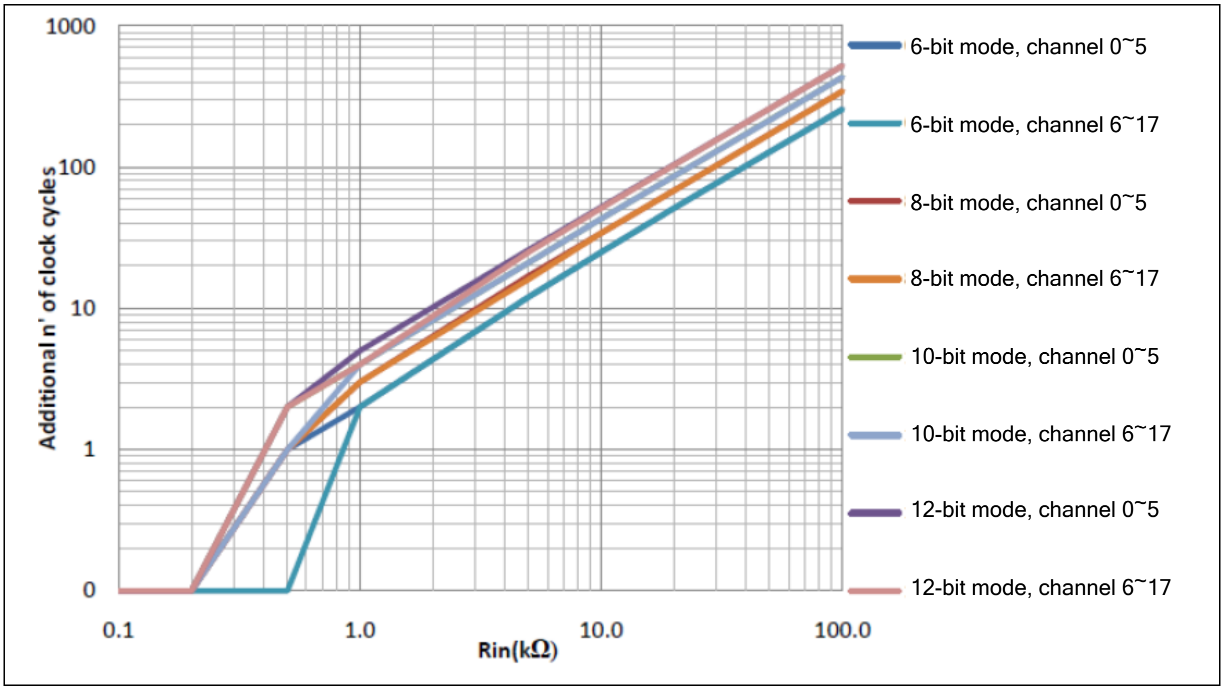


Figure 5.22-12 Additional Sample and Hold Clock Cycles (n) as a Function of the Signal Source Output Resistance Rin (kΩ)

The results presented in the graphic above were measured under normal conditions (typical process corner, V_{DD} = AV_{DD} = 3.3V, LDO output = 1.8V, Tjunction= 50°C, ADC_CLK=42 MHz, V_{REF}= AV_{DD}).

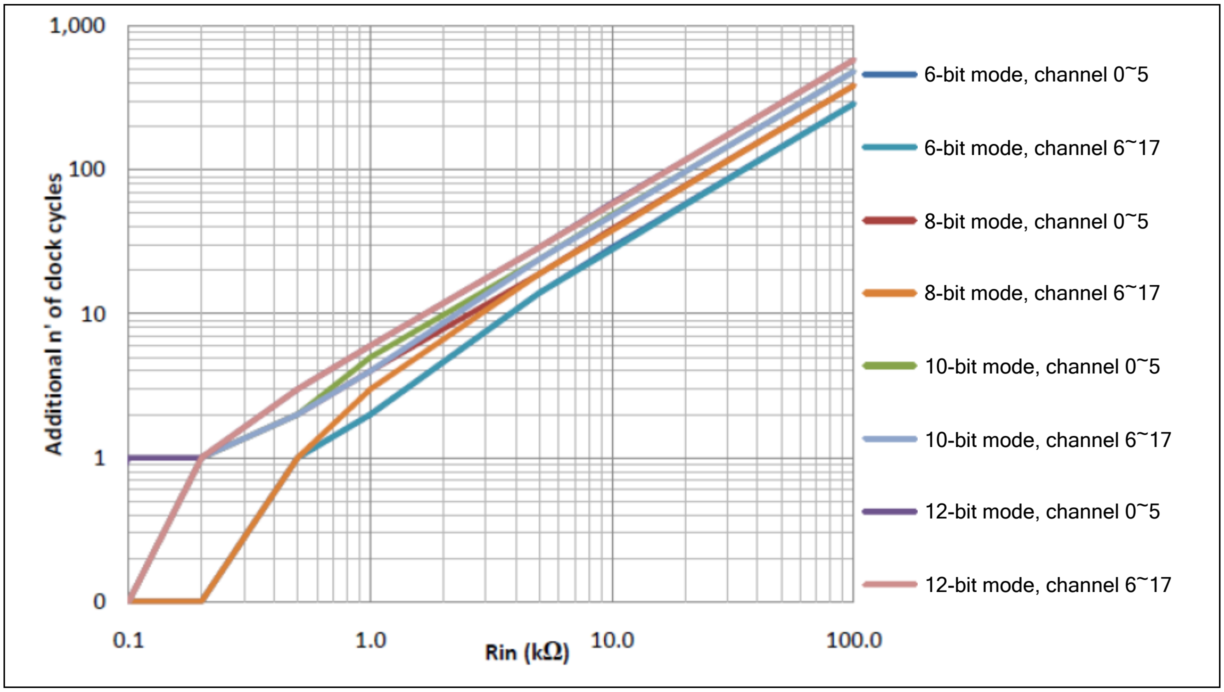


Figure 5.22-13 Additional Sample and Hold Clock Cycles (n) as a Function of the Signal Source Output Resistance Rin (kΩ)

The results presented in the graphic above were measured under the worst case conditions (slow process corner, V_{DD} = AV_{DD} = 1.8V, LDO output = 1.62V, Tjunction= -40°C, ADC_CLK=42 MHz, V_{REF}= AV_{DD}).

5.22.4.10 ADC Power-down mode

There are two Power-down modes user can select, including Power-down mode, and Standby mode. User can configure PWDMOD in ADCPWD register to determine what Power-down mode that user want to be before disabling ADEN in ADCR register.

In different Power-down mode (power down,standby), the power up sequence are quite different, user should know currently Power-down mode and configure PWDMOD in ADCPWD register to determine what power up sequence that user want to be before enabling ADCEN in ADCR register, if the sequence was wrong, ADC would be mal-function .

The difference between those Power-down modes are power consumption and the stable time after resuming from each Power-down mode, the least power consumption is Power-down mode and then p standby mode,and stable time are in reversed order Before ADC entering power down, make sure that ADC is stop (by disabling ADST) and all conversion are completed (by polling ADF)

5.22.4.11 ADC Offset Calibration

To decrease the effect of electrical random noise, the ADC performs calibration to get average offset measurement. Afterwards, in normal operation, the digital block applies the calibrated word to the internal ADC capacitor array, so that the offset voltage is removed.

User can set CALEN to high and select CALSEL to 1(to do calibration) and write 1 to CALSTART, then waiting for CALDONE bit to high; when CALDONE is high, the calibration is complete and the calibration word is in ADC_CALWORD register.

User can also load the specified calibrated word to ADC_CALWORD register to save time to complete the calibration method. The configuration are the same except setting CALSEL to 0, and waiting for CALDONE bit to high; when CALDONE is high, the load calibration word is complete and the loaded calibration word is now applied to ADC.

5.22.4.12 Slectable Resolution

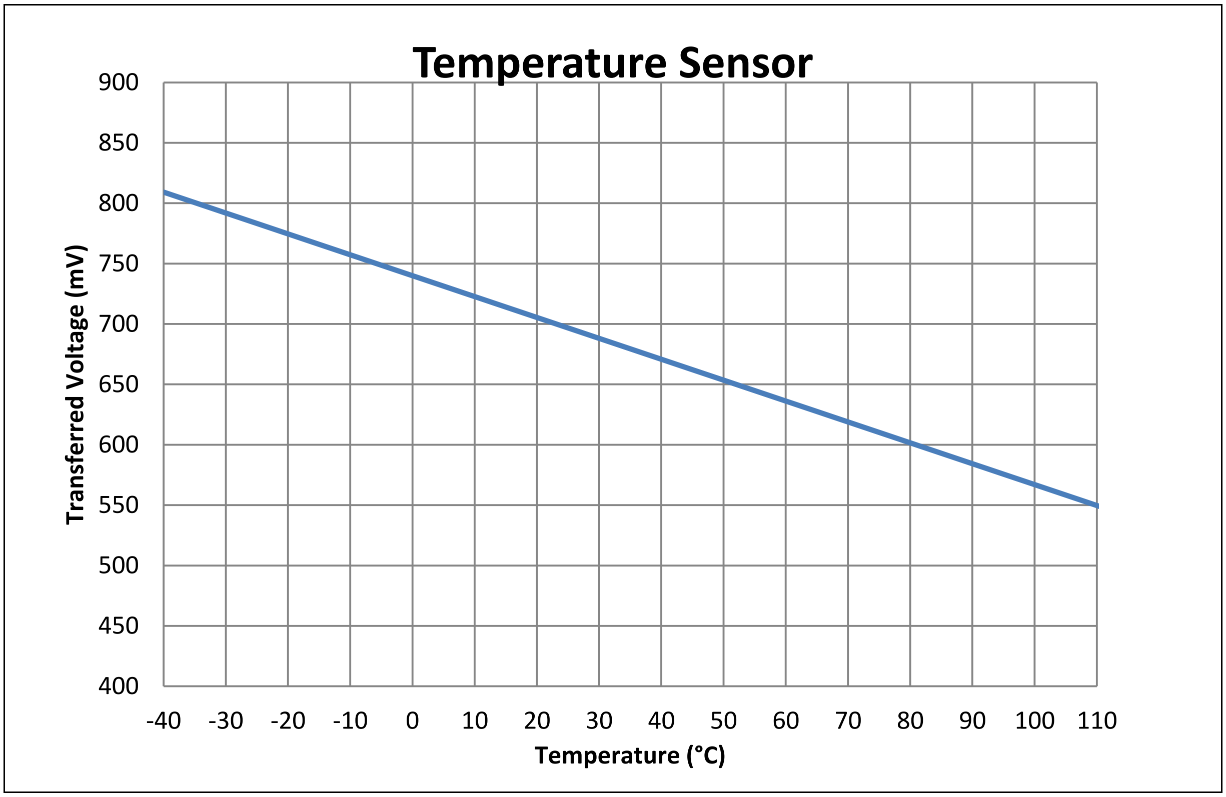
The ADC has the selectable resolution between 12, 10, 8 and 6 bit. User can choose the resolution by setting RESSEL in ADCR register.

The resolution selection can only be updated after the end of the conversion (ADF becomes high), different resolutions will result in the different conversion cycle. Take 12 bit resolution for example, it will take 19 ADC clock cycle to complete one channel conversion; and 17 ADC clock cycle,15 ADC clock cycle, 14 ADC clock cycle for the resolution 10 bits, 8 bits, and 6 bits.

5.22.4.13 Temperature Sensor

The figure below shows the typical temperature sensor transfer function. The formula for the output voltage (V_{TEMP}) is as below equation.

V_{TEMP} (mV) = -1.73 (mV/℃) x Temperature (℃) + 740 (mV).



5.22.4.14 Internal Reference Voltage

The internal reference voltage (Int_VREF) is an internal fixed reference voltage regardless of power supply variations. The Int_VREF output is internally connected to ADC input channel15 (ADC15) and Analog Comparator's negative input side. For battery power detection application, user can connect V_{REF} pin to AV_{DD} for ADC reference voltage. The Int_VREF can be used as ADC input channel such that user can convert the ADC value to estimate AV_{DD} voltage with following formula and the block diagram is shown as Figure 5.22-14.

AV_{DD} = ((2 ^ N) / R) * Int_VREF

N: ADC resolution

R: ADC conversion result

Int_VREF: Internal reference voltage

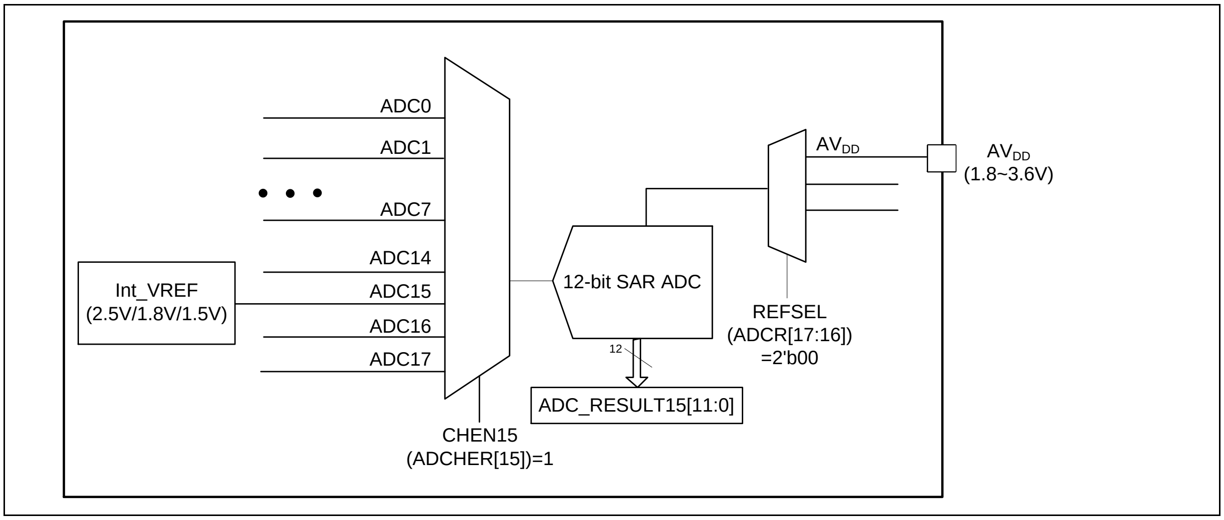


Figure 5.22-14 Int_VREF for Measuring AV_{DD} Application Block Diagram

For example, The setting value for Int_VREF is 1.8 V, the ADC is 12-bit resolution, and Int_VREF is set as ADC input channel to trigger ADC conversion. Assuming ADC conversion result is 2048.

AV_{DD} = 3.6 V

N = 12

R = 2048

Int_VREF = 1.8 V

AV_{DD} = ((2 ^ 12) / 2048) * 1.8 = (4096 / 2048) * 1.8 = 3.6 V

If the ADC conversion result is 2457

AV_{DD} = ((2 ^ 12) / 2458) * 1.8 = (4096 / 2458) * 1.8 = 3 V

If the ADC conversion result is 2949

AV_{DD} = ((2 ^ 12) / 2949) * 1.8 = (4096 / 2949) * 1.8 = 2.5 V

5.22.5 Register and Memory Map

R: read only, **W**: write only, **R/W:** both read and write

Register	Offset	R/W	Description	Reset Value
ADC Base Address: **ADC_BA = 0x400E_0000**				
ADC_RESULT0	ADC_BA+0x00	R	A/D Data Register 0	0x0000_0000
ADC_RESULT1	ADC_BA+0x04	R	A/D Data Register 1	0x0000_0000
ADC_RESULT2	ADC_BA+0x08	R	A/D Data Register 2	0x0000_0000
ADC_RESULT3	ADC_BA+0x0C	R	A/D Data Register 3	0x0000_0000
ADC_RESULT4	ADC_BA+0x10	R	A/D Data Register 4	0x0000_0000
ADC_RESULT5	ADC_BA+0x14	R	A/D Data Register 5	0x0000_0000
ADC_RESULT6	ADC_BA+0x18	R	A/D Data Register 6	0x0000_0000
ADC_RESULT7	ADC_BA+0x1C	R	A/D Data Register 7	0x0000_0000
ADC_RESULT8	ADC_BA+0x20	R	A/D Data Register 8	0x0000_0000
ADC_RESULT9	ADC_BA+0x24	R	A/D Data Register 9	0x0000_0000
ADC_RESULT10	ADC_BA+0x28	R	A/D Data Register 10	0x0000_0000
ADC_RESULT11	ADC_BA+0x2C	R	A/D Data Register 11	0x0000_0000
ADC_RESULT12	ADC_BA+0x30	R	A/D Data Register 12	0x0000_0000
ADC_RESULT13	ADC_BA+0x34	R	A/D Data Register 13	0x0000_0000
ADC_RESULT14	ADC_BA+0x38	R	A/D Data Register 14	0x0000_0000
ADC_RESULT15	ADC_BA+0x3C	R	A/D Data Register 15	0x0000_0000
ADC_RESULT16	ADC_BA+0x40	R	A/D Data Register 16	0x0000_0000
ADC_RESULT17	ADC_BA+0x44	R	A/D Data Register 17	0x0000_0000
ADCR	ADC_BA+0x48	R/W	A/D Control Register	0x0001_0000
ADCHER	ADC_BA+0x4C	R/W	A/D Channel Enable Register	0x0000_0000
ADCMPR0	ADC_BA+0x50	R/W	A/D Compare Register 0	0x0000_0000
ADCMPR1	ADC_BA+0x54	R/W	A/D Compare Register 1	0x0000_0000
ADSR	ADC_BA+0x58	R/W	A/D Status Register	0x0000_0000
ADPDMA	ADC_BA+0x60	R	A/D PDMA current transfer data Register	0x0000_0000
ADCPWD	ADC_BA+0x64	R/W	ADC Power Management Register	0x0001_E002
ADCCALCTL	ADC_BA+0x68	R/W	ADC Calibration Control Register	0x0000_0009
ADCCALWORD	ADC_BA+0x6C	R/W	A/D calibration load word register	0xXXXX_XXXX

ADCCHSAMP0	ADC_BA+0x70	R/W	ADC Channel Sampling Time Counter Register Group 0	0x0000_0000
ADCCHSAMP1	ADC_BA+0x74	R/W	ADC Channel Sampling Time Counter Register Group 1	0x0000_0000

5.22.6 Register Description

A/D Data Registers (ADC_RESULT0~ ADC_RESULT10)

Register	Offset	R/W	Description	Reset Value
ADC_RESULT0	ADC_BA+0x00	R	A/D Data Register 0	0x0000_0000
ADC_RESULT1	ADC_BA+0x04	R	A/D Data Register 1	0x0000_0000
ADC_RESULT2	ADC_BA+0x08	R	A/D Data Register 2	0x0000_0000
ADC_RESULT3	ADC_BA+0x0C	R	A/D Data Register 3	0x0000_0000
ADC_RESULT4	ADC_BA+0x10	R	A/D Data Register 4	0x0000_0000
ADC_RESULT5	ADC_BA+0x14	R	A/D Data Register 5	0x0000_0000
ADC_RESULT6	ADC_BA+0x18	R	A/D Data Register 6	0x0000_0000
ADC_RESULT7	ADC_BA+0x1C	R	A/D Data Register 7	0x0000_0000
ADC_RESULT8	ADC_BA+0x20	R	A/D Data Register 8	0x0000_0000
ADC_RESULT9	ADC_BA+0x24	R	A/D Data Register 9	0x0000_0000
ADC_RESULT10	ADC_BA+0x28	R	A/D Data Register 10	0x0000_0000
ADC_RESULT11	ADC_BA+0x2C	R	A/D Data Register 11	0x0000_0000
ADC_RESULT12	ADC_BA+0x30	R	A/D Data Register 12	0x0000_0000
ADC_RESULT13	ADC_BA+0x34	R	A/D Data Register 13	0x0000_0000
ADC_RESULT14	ADC_BA+0x38	R	A/D Data Register 14	0x0000_0000
ADC_RESULT15	ADC_BA+0x3C	R	A/D Data Register 15	0x0000_0000
ADC_RESULT16	ADC_BA+0x40	R	A/D Data Register 16	0x0000_0000
ADC_RESULT17	ADC_BA+0x44	R	A/D Data Register 17	0x0000_0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="8">Reserved</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="6">Reserved</td><td>OVERRUN</td><td>VALID</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td colspan="4">Reserved</td><td colspan="4">RSLT</td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">RSLT</td></tr>
</table>

Bits	Description	
[31:18]	**Reserve**	**Reserved**
[17]	**OVERRUN**	**Over Run Flag** **I**t is a mirror to OVERRUN bit in ADC_RESULTx
[16]	**VALID**	**Data Valid Flag** It is a mirror of VALID bit in ADC_RESULTx
[15:12]	**Reserved**	**Reserved**
[11:0]	**RSLT**	**A/D Conversion Result** This field contains 12 bits conversion results.

A/D Control Register (ADCR)

Register	Offset	R/W	Description	Reset Value
ADCR	ADC_BA+0x48	R/W	A/D Control Register	0x0001_0000

31	30	29	28	27	26	25	24
TMPDMACNT							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
-				RESSEL		REFSEL	
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
TMTRGMOD	-	TMSEL[1:0]		ADST	DIFF	PTEN	TRGEN
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
TRGCOND		TRGS		ADMD		ADIE	ADEN

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:24]</td><td>TMPDMACNT</td><td>PDMA Count
When each timer event occur PDMA will transfer TMPDMACNT +1 ADC result in the amount of this register setting
Note: The total amount of PDMA transferring data should be set in PDMA byte count register. When PDMA finish is set, ADC will not be enabled and start transfer even though the timer event occurred</td></tr>
<tr><td>[23:20]</td><td>-</td><td>Reserved</td></tr>
<tr><td>[19:18]</td><td>RESSEL</td><td>Resolution Selection
00: 6 bits
01: 8 bits
10: 10 bits
11: 12 bits</td></tr>
<tr><td>[17:16]</td><td>REFSEL</td><td>Reference Voltage Source Selection
00: Select power as reference voltage
01: Select Int_VREF as reference voltage
10: Select V_{REF} as reference voltage</td></tr>
<tr><td>[15]</td><td>TMTRGMOD</td><td>Timer Event Trigger ADC Conversion
1 = ADC Enabled by TIMER OUT event
setting TMSEL to select timer event from timer0~3
0 = This function Disabled.</td></tr>
<tr><td>[14]</td><td>-</td><td>Reserved</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[13:12]</td><td>TMSEL</td><td>Select A/D Enable Time-out Source
<table><tr><td>00</td><td>TMR0</td></tr><tr><td>01</td><td>TMR1</td></tr><tr><td>10</td><td>TMR2</td></tr><tr><td>11</td><td>TMR3</td></tr></table></td></tr>
<tr><td>[11]</td><td>ADST</td><td>A/D Conversion Start
1 = Conversion starts.
0 = Conversion stopped and A/D converter enter idle state.
ADST bit can be set to 1 from two sources: software write and external pin STADC. ADST is cleared to 0 by hardware automatically at the end of single mode and single-cycle scan mode on specified channels. In continuous scan mode, A/D conversion is continuously performed sequentially unless software writes 0 to this bit or chip reset.
Note: After ADC conversion done, SW needs to wait at least one ADC clock before to set this bit high again.</td></tr>
<tr><td>[10]</td><td>DIFF</td><td>Differential Mode Selection
1 = ADC is operated in differential mode
0 = ADC is operated in single-ended mode
The A/D analog input ADC_CH0/ADC_CH1 consists of a differential pair. So as ADC_CH2/ADC_CH3, ADC_CH4/ADC_CH5, ADC_CH6/ADC_CH7, ADC_CH8/ADC_CH9 and ADC_CH10/ADC_CH11. The even channel defines as plus analog input voltage (V_{plus}) and the odd channel defines as minus analog input voltage (V_{minus}).
Differential input voltage (V_{diff}) = V_{plus} - V_{minus}, where V_{plus} is the analog input; V_{minus} is the inverted analog input.
In differential input mode, only the even number of the two corresponding channels needs to be enabled in CHEN (ADCHER[11:0]). The conversion result will be placed to the corresponding data register of the enabled channel.
Note: Calibration should calibrated each time when switching between single-ended and differential mode</td></tr>
<tr><td>[9]</td><td>PTEN</td><td>PDMA Transfer Enable
1 = PDMA data transfer in ADC_RESULT 0~17 Enabled
0 = PDMA data transfer Disabled.
When A/D conversion is completed, the converted data is loaded into ADC_RESULT 0~10, software can enable this bit to generate a PDMA data transfer request.
When PTEN=1, software must set ADIE=0 to disable interrupt. PDMA can access ADC_RESULT 0-17 registers by block or single transfer mode.</td></tr>
<tr><td>[8]</td><td>TRGE</td><td>External Trigger Enable
Enable or disable triggering of A/D conversion by external STADC pin.
1 = Enabled,
0 = Disabled,</td></tr>
</table>

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[7:6]</td><td>TRGCOND</td><td>External Trigger Condition
These two bits decide external pin STADC trigger event is level or edge. The signal must be kept at stable state at least 8 PCLKs for level trigger and 4 PCLKs at high and low state.
<table><tr><td>00</td><td>Low level</td></tr><tr><td>01</td><td>High level</td></tr><tr><td>10</td><td>Falling edge</td></tr><tr><td>11</td><td>Rising edge</td></tr></table></td></tr>
<tr><td>[5:4]</td><td>TRGS</td><td>Hardware Trigger Source
<table><tr><td>00</td><td>A/D conversion is started by external STADC pin.</td></tr><tr><td>Others</td><td>Reserved</td></tr></table>Software should disable TRGE and ADST before change TRGS.
In hardware trigger mode, the ADST bit is set by the external trigger from STADC, However software has the highest priority to set or cleared ADST bit at any time.</td></tr>
<tr><td>[3:2]</td><td>ADMD</td><td>A/D Converter Operation Mode
<table><tr><td>00</td><td>Single conversion</td></tr><tr><td>01</td><td>Reserved</td></tr><tr><td>10</td><td>Single-cycle scan</td></tr><tr><td>11</td><td>Continuous scan</td></tr></table></td></tr>
<tr><td>[1]</td><td>ADIE</td><td>A/D Interrupt Enable
1 = A/D interrupt function Enabled.
0 = A/D interrupt function Disabled.
A/D conversion end interrupt request is generated if ADIE bit is set to 1.</td></tr>
<tr><td>[0]</td><td>ADEN</td><td>A/D Converter Enable
1 = Enabled.
0 = Disabled.
Before starting A/D conversion, this bit should be set to 1. Clear it to 0 to disable A/D converter analog circuit power consumption.</td></tr>
</table>

A/D Channel Enable Register (ADCHER)

Register	Offset	R/W	Description	Reset Value
ADCHER	ADC_BA+0x4C	R/W	A/D Channel Enable Register	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
-						CHEN17	CHEN16
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CHEN15	CHEN14	CHEN13	CHEN12	CHEN11	CHEN10	CHEN9	CHEN8
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHEN7	CHEN6	CHEN5	CHEN4	CHEN3	CHEN2	CHEN1	CHEN0

Bits	Description	
[31:18]	**Reserved**	-
[17]	**CHEN17**	**Analog Input Channel 17 Enable (Convert AV_{SS})** 1 = Enabled. 0 = Disabled.
[16]	**CHEN16**	**Analog Input Channel 16 Enable (Convert AV_{DD})** 1 = Enabled. 0 = Disabled.
[15]	**CHEN15**	**Analog Input Channel 15 Enable (Convert Int_VREF)** 1 = Enabled. 0 = Disabled.
[14]	**CHEN14**	**Analog Input Channel 14 Enable (Convert V_{TEMP})** 1 = Enabled. 0 = Disabled.
[13]	**CHEN13**	**Analog Input Channel 13 Enable (Convert DAC1 Output Voltage)** 1 = Enabled. 0 = Disabled.
[12]	**CHEN12**	**Analog Input Channel 12 Enable (Convert DAC0 Output Voltage)** 1 = Enabled. 0 = Disabled.
[11]	**CHEN11**	**Analog Input Channel 11 Enable(Convert input voltage from PD.3)** 1 = Enabled. 0 = Disabled.

Bits	Description	
[10]	**CHEN10**	**Analog Input Channel 10 Enable (Convert Input Voltage from PD.2)** 1 = Enabled. 0 = Disabled.
[9]	**CHEN9**	**Analog Input Channel 9 Enable for DAC1 (Convert Input Voltage from PD.1)** 1 = Enabled. 0 = Disabled.
[8]	**CHEN8**	**Analog Input Channel 8 Enable for DAC0 (Convert Input Voltage from PD.0)** 1 = Enabled. 0 = Disabled.
[7]	**CHEN7**	**Analog Input Channel 7 Enable (Convert Input Voltage from PA.7)** 1 = Enabled. 0 = Disabled.
[6]	**CHEN6**	**Anaslog Input Channel 6 Enable (Convert Input Voltage from PA.6)** 1 = Enabled. 0 = Disabled.
[5]	**CHEN5**	**Analog Input Channel 5 Enable (Convert Input Voltage from PA.5)** 1 = Enabled. 0 = Disabled.
[4]	**CHEN4**	**Analog Input Channel 4 Enable (Convert Input Voltage from PA.4)** 1 = Enabled 0 = Disabled
[3]	**CHEN3**	**Analog Input Channel 3 Enable(Convert input voltage from PA.3)** 1 = Enabled. 0 = Disabled.
[2]	**CHEN2**	**Analog Input Channel 2 Enable (Convert Input Voltage from PA.2)** 1 = Enabled. 0 = Disabled.
[1]	**CHEN1**	**Analog Input Channel 1 Enable(Convert input voltage from PA.1)** 1 = Enabled. 0 = Disabled.
[0]	**CHEN0**	**Analog Input Channel 0 Enable (Convert Input Voltage from PA.0)** 1 = Enabled. 0 = Disabled. If more than one channel in single mode is enabled by software, the least channel is converted and other enabled channels will be ignored.

A/D Compare Register 0/1 (ADCMPR0/1)

Register	Offset	R/W	Description	Reset Value
ADCMPR0	ADC_BA+0x50	R/W	A/D Compare Register 0	0x0000_0000
ADCMPR1	ADC_BA+0x54	R/W	A/D Compare Register 1	0x0000_0000

31	30	29	28	27	26	25	24
-				CMPD			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CMPD							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
-				CMPMATCNT			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-	CMPCH				CMPCOND	CMPIE	CMPEN

Bits	Description	
[31:28]	**Reserved**	-
[27:16]	**CMPD**	**Comparison Data** The 12 bits data is used to compare with conversion result of specified channel. Software can use it to monitor the external analog input pin voltage variation in scan mode without imposing a load on software.
[15:12]	**Reserved**	-
[11:8]	**CMPMATCNT**	**Compare Match Count** When the specified A/D channel analog conversion result matches the compare condition defined by CMPCOND[2], the internal match counter will increase 1. When the internal counter reaches the value to (CMPMATCNT +1), the CMPF bit will be set.

<table>
<tr><th>Bits</th><th colspan="3">Description</th></tr>
<tr><td rowspan="19">[7:3]</td><td rowspan="19">CMPCH</td><td colspan="2">Compare Channel Selection</td></tr>
<tr><td>00000</td><td>Channel 0 conversion result is selected to be compared.</td></tr>
<tr><td>00001</td><td>Channel 1 conversion result is selected to be compared.</td></tr>
<tr><td>00010</td><td>Channel 2 conversion result is selected to be compared.</td></tr>
<tr><td>00011</td><td>Channel 3 conversion result is selected to be compared.</td></tr>
<tr><td>00100</td><td>Channel 4 conversion result is selected to be compared.</td></tr>
<tr><td>00101</td><td>Channel 5 conversion result is selected to be compared.</td></tr>
<tr><td>00110</td><td>Channel 6 conversion result is selected to be compared.</td></tr>
<tr><td>00111</td><td>Channel 7 conversion result is selected to be compared.</td></tr>
<tr><td>01000</td><td>Channel 8 conversion result is selected to be compared.</td></tr>
<tr><td>01001</td><td>Channel 9 conversion result is selected to be compared.</td></tr>
<tr><td>01010</td><td>Channel 10 conversion result is selected to be compared.</td></tr>
<tr><td>01011</td><td>Channel 11 conversion result is selected to be compared</td></tr>
<tr><td>01100</td><td>Channel 12 conversion result is selected to be compared</td></tr>
<tr><td>01101</td><td>Channel 13 conversion result is selected to be compared</td></tr>
<tr><td>01110</td><td>Channel 14 conversion result is selected to be compared</td></tr>
<tr><td>01111</td><td>Channel 15 conversion result is selected to be compared</td></tr>
<tr><td>10000</td><td>Channel 16 conversion result is selected to be compared</td></tr>
<tr><td>10001</td><td>Channel 17 conversion result is selected to be compared</td></tr>
<tr><td>[2]</td><td>CMPCOND</td><td colspan="2">Compare Condition
1 = Set the compare condition as that when a 12-bit A/D conversion result is greater or equal to the 12-bit CMPD (ADCMPRx[27:16]), the internal match counter will increase by one.
0 = Set the compare condition as that when a 12-bit A/D conversion result is less than the 12-bit CMPD (ADCMPRx[27:16]), the internal match counter will increase one.
Note: When the internal counter reaches the value to (CMPMATCNT +1), the CMPF bit will be set.</td></tr>
<tr><td>[1]</td><td>CMPIE</td><td colspan="2">Compare Interrupt Enable
1 = Compare function interrupt Enabled.
0 = Compare function interrupt Disabled.
If the compare function is enabled and the compare condition matches the setting of CMPCOND and CMPMATCNT, CMPF bit will be asserted, in the meanwhile, if CMPIE is set to 1, a compare interrupt request is generated.</td></tr>
</table>

Bits	Description	
[0]	**CMPEN**	**Compare Enable** 1 = Compare Enabled. 0 = Compare Disabled. Set this bit to 1 to enable compare CMPD[11:0] with specified channel conversion result when converted data is loaded into ADC_RESULTx register. When this bit is set to 1, and CMPMATCNT is 0, the CMPF will be set once the match is hit

A/D Status Register (ADSR)

Register	Offset	R/W	Description	Reset Value
ADSR	ADC_BA+0x58	R/W	A/D Status Register	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
							INITRDY
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
							CHANNEL
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CHANNEL				BUSY	CMPF1	CMPF0	ADF

Bits	Description	
[31:17]	**-**	**Reserved**
[16]	**INITRDY**	**ADC Power-up Sequence Completed** 1 = ADC has been powered up since the last system reset 0 = ADC not powered up after system reset **Note:** This bit will be set after system reset occurred and automatically cleared by power-up event.
[15:9]	**reserved**	
[8:4]	**CHANNEL**	**Current Conversion Channel** This filed reflects current conversion channel when BUSY=1. When BUSY=0, it shows the next channel to be converted. It is read only.
[3]	**BUSY**	**BUSY/IDLE** 1 = A/D converter is busy at conversion. 0 = A/D converter is in idle state. This bit is a mirror of ADST bit in ADCR. That is to say if ADST = 1,then BUSY is 1 and vice versa It is read only.
[2]	**CMPF1**	**Compare Flag** When the selected channel A/D conversion result meets setting condition in ADCMPR1 then this bit is set to 1. And it is cleared by writing 1 to self. 1 = Conversion result in ADC_RESULTx meets ADCMPR1 setting 0 = Conversion result in ADC_RESULTx does not meet ADCMPR1 setting This flag can be cleared by writing 1 to it. **Note:** when this flag is set, the matching counter will be reset to 0,and continue to count when user write 1 to clear CMPF1

Bits	Description	
[1]	**CMPF0**	**Compare Flag** When the selected channel A/D conversion result meets setting condition in ADCMPR0 then this bit is set to 1. And it is cleared by writing 1 to self. 1 = Conversion result in ADC_RESULTx meets ADCMPR0setting 0 = Conversion result in ADC_RESULTx does not meet ADCMPR0setting This flag can be cleared by writing 1 to it. **Note:** When this flag is set, the matching counter will be reset to 0,and continue to count when user write 1 to clear CMPF0
[0]	**ADF**	**A/D Conversion End Flag** **A** status flag that indicates the end of A/D conversion. ADF is set to 1 at these two conditions: When A/D conversion ends in single mode When A/D conversion ends on all specified channels in scan mode. This flag can be cleared by writing 1 to it.

A/D PDMA Current Transfer Data Register (ADPDMA)

Register	Offset	R/W	Description	Reset Value
ADPDMA	ADC_BA+0x60	R	A/D PDMA current transfer data Register	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
-							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
-				AD_PDMA			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
AD_PDMA							

Bits	Description	
[31:12]	**Reserved**	-
[11:0]	**AD_PDMA**	**ADC PDMA Current Transfer Data Register** When PDMA transferring, read this register can monitor current PDMA transfer data. This is a read only register.

A/D Power Management Register (ADCPWD)

Register	Offset	R/W	Description	Reset Value
ADCPWD	ADC_BA+0x64	R/W	ADC Power Management Register	0x0001_E002

31	30	29	28	27	26	25	24
-							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
-							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-				PWDMOD		PWDCALEN	PWUPRDY

Bits	Description	
[31:4]	**Reserved**	-
[3:2]	**PWDMOD**	**Power-down Mode** Set this bit fields to select ADC power down mode when system power-down. 00 = Reserved. 01 = ADC Power down mode. 10 = ADC Standby mode. 11 = Reserved. **Note1:** Different PWDMOD has different power down/up sequence, in order to avoid ADC powering up with wrong sequence; user must keep PWMOD consistent each time in power down and power up. **Note2:** While the ADC is power up from power down mode without calibration, the PWDCALEN(ADCPWD[1]) is set to 0. (The calibration value will be reset)
[1]	**PWDCALEN**	**Power up Calibration Function Enable Control** 0 = Power up without calibration. 1 = Power up with calibration. **Note:** This bit work together with CALSEL (ADCCALCTL[3]), see the following {PWDCALEN,CALFBSEL} Description: PWDCALEN is 0 and CALFBSEL is 0: No need to calibrate. PWDCALEN is 0 and CALFBSEL is 1: No need to calibrate. PWDCALEN is 1 and CALFBSEL is 0: Load calibration word when power up. PWDCALEN is 1 and CALFBSEL is 1: Calibrate when power up.
[0]	**PWUPRDY**	**ADC Power-up Sequence Completed and Ready for Conversion** 1 = ADC is ready for conversion 0 = ADC is not ready for conversion;may be in power down state or in the progress of power up

A/D Calibration Control Register (ADCCALCTL)

Register	Offset	R/W	Description	Reset Value
ADCCALCTL	ADC_BA+0x68	R/W	ADC Calibration Control Register	0x0000_0009

31	30	29	28	27	26	25	24
-							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
-							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
-							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-				CALSEL	CALDONE	CALSTART	CALEN

Bits	Description	
[31:4]	-	**Reserved**
[3]	**CALSEL**	**Select Calibration Functional Block** 1 = Calibration functional block. 0 = Load calibration functional block.
[2]	**CALDONE**	**Calibrate Functional Block Complete** 1 = Selected functional block complete. 0 = Not yet.
[1]	**CALSTART**	**Calibration Functional Blcok Start** 1 = Starts calibration functional block. 0 = Stops calibration functional block. **Note:** This bit is set by SW and clear by HW;don't write 1 to this bit while CALEN = 0
[0]	**CALEN**	**Calibraion Function Enable** Enable this bit to turn on the calibration function block. 1 = Enabled. 0 = (BYPASSCAL).

A/D Calibration Word (ADCCALWORD)

Register	Offset	R/W	Description	Reset Value
ADCCALWORD	ADC_BA+0x6C	R/W	A/D calibration load word register	0xXXXX_XXXX

31	30	29	28	27	26	25	24
-							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
-							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
-							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-	CALWORD						

Bits	Description	
[31:7]	-	**Reserved**
[6:0]	**CALWORD**	**Calibration Word Register** Write to this register with the previous calibration word before load calibration action Read this register after calibration done **Note:** The calibration block contains two parts “CALIBRATION“ and “LOAD CALIBRATION”; if thecalibration block is config as “CALIBRATION“; then this register represent the result of calibration when calibration is completed; if config as “LOAD CALIBRATION” ; config this register before loading calibration action, after loading calibration complete, the laoded calibration word will apply to the ADC;while in loading calibration function the loaded value will not be equal to the orginal CALWORD until calibration is done.

A/D Channel Samping0 Register (ADCCHSAMP0)

Register	Offset	R/W	Description	Reset Value
ADCCHSAMP0	ADC_BA+0x70	R/W	ADC Channel Sampling Time Counter Register Group 0	0x0000_0000

31	30	29	28	27	26	25	24
CH7SAMPCNT				CH6SAMPCNT			
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
CH5SAMPCNT				CH4SAMPCNT			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CH3SAMPCNT				CH2SAMPCNT			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CH1SAMPCNT				CH0SAMPCNT			

Bits	Description	
[31:28]	**CH7SAMPCNT**	**Channel 7 Sampling Counter** The same as Channel 0 sampling counter table.
[27:24]	**CH6SAMPCNT**	**Channel 6 Sampling Counter** The same as Channel 0 sampling counter table.
[23:20]	**CH5SAMPCNT**	**Channel 5 Sampling Counter** The same as Channel 0 sampling counter table.
[19:16]	**CH4SAMPCNT**	**Channel 4 Sampling Counter** The same as Channel 0 sampling counter table.
[15:12]	**CH3SAMPCNT**	**Channel 3 Sampling Counter** The same as Channel 0 sampling counter table.
[11:8]	**CH2SAMPCNT**	**Channel 2 Sampling Counter** The same as Channel 0 sampling counter table.
[7:4]	**CH1SAMPCNT**	**Channel 1 Sampling Counter** The same as Channel 0 sampling counter table.

Bits	Description	
[3:0]	**CH0SAMPCNT**	**Channel 0 Sampling Counter**

CH0SAMPCNT	ADC Clock
0	0
1	1
2	2
3	4
4	8
5	16
6	32
7	64
8	128
9	256
10	512
11	1024
12	1024
13	1024
14	1024
15	1024

A/D Channel Samping1 Register (ADCCHSAMP1)

Register	Offset	R/W	Description	Reset Value
ADCCHSAMP1	ADC_BA+0x74	R/W	ADC Channel Sampling Time Counter Register Group 1	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
-				INTCHSAMPCNT			
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
CH11SAMPCNT				CH10SAMPCNT			
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
CH9SAMPCNT				CH8SAMPCNT			

Bits	Description	
[31:20]	-	**Reserved**
[19:16]	**INTCHSAMPCNT**	**Internal Channel (V_{TEMP}, AV_{DD}, AV_{SS}, Int_VREF, DAC0, DAC1) Sampling Counter** The same as Channel 0 sampling counter table.
[15:12]	**CH11SAMPCNT**	**Channel 11 Sampling Counter** The same as Channel 0 sampling counter table.
[11:8]	**CH10SAMPCNT**	**Channel 10 Sampling Counter** The same as Channel 0 sampling counter table.
[7:4]	**CH9SAMPCNT**	**Channel 9 Sampling Counter** The same as Channel 0 sampling counter table.
[3:0]	**CH8SAMPCNT**	**Channel 8 Sampling Counter** The same as Channel 0 sampling counter table.

5.23 Digital to Analog Converter (DAC)

5.23.1 Overview

DAC is a 12-bit voltage-output digital-to-analog converter. Two DACs are implemented in this chip.

5.23.2 Features

DAC is a 12-bit voltage-output DAC. DAC can use in conjunction with the PDMA controller. When two DACs are present, they may be grouped together for synchronous update operation.

- Int_VREF or V_{REF} or AV_{DD} reference voltage selection
- Synchronized update capability for two DACs
- DAC maximum conversion rate is 500 kSPS

5.23.3 Block Diagram

The block diagram of the two DACs is shown in the following figure.

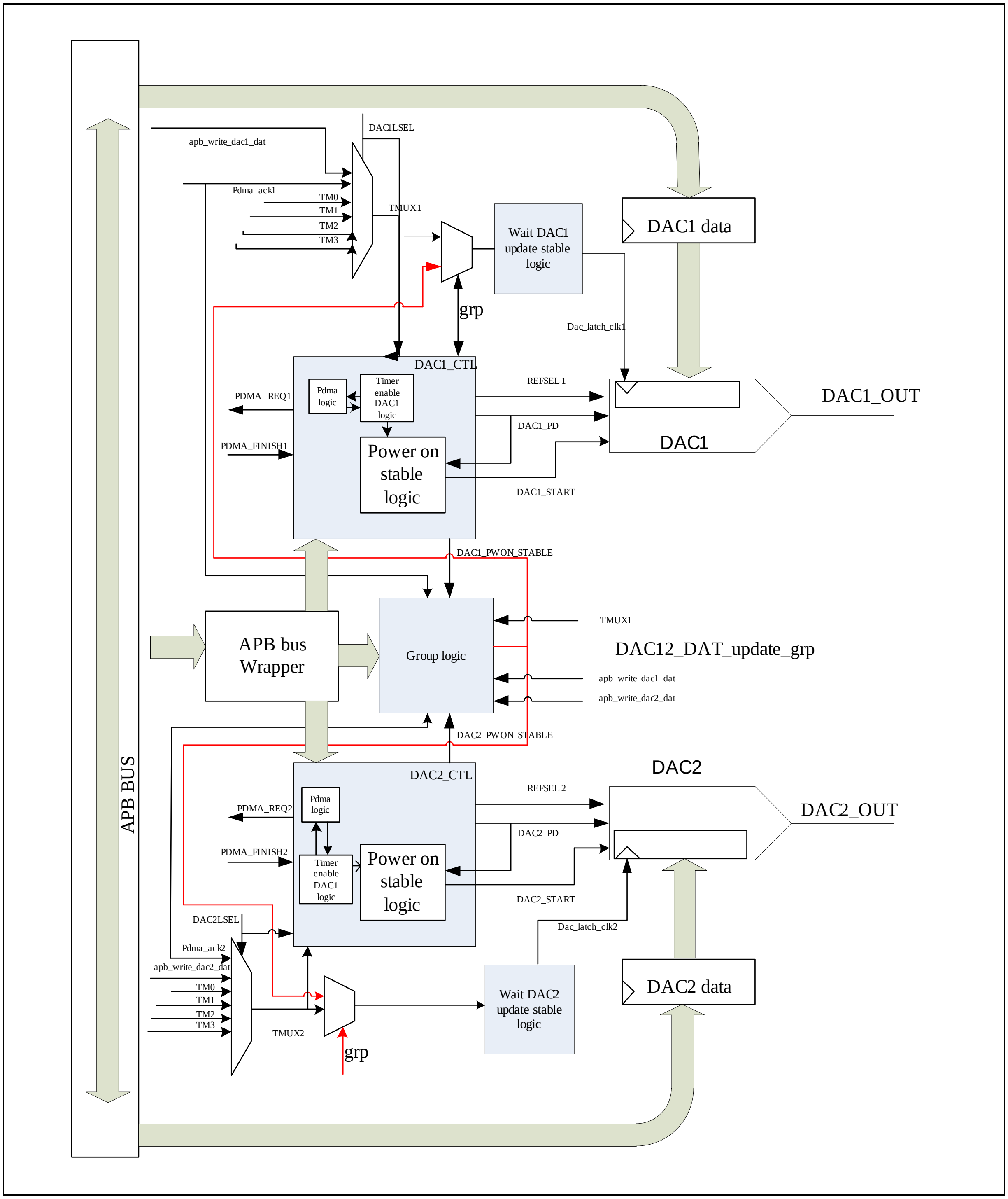


Figure 5.23-1 DAC0 and DAC1 Block Diagram

5.23.4 Functional Description

The DAC is configured by user software. The setup and operation of the DAC is discussed in the following sections.

5.23.4.1 DAC Core

The conversion time between each DACx_DAT is 2 us and the power on stable time is 6 us, both are at the worse case.The maximum DAC0 and DAC1 output voltage (full scale) is limited to the selected reference voltage.

5.23.4.2 DAC Reference Voltage

The reference for the DAC is configured to use either an external reference voltage or the internal voltage reference generator or AV_{DD} with three selectable voltage levels (Int_VREF \ V_{REF} \ AV_{DD}).

5.23.4.3 DAC Operation

DAC will update the output voltage level whenever user updates the DACx_DAT. There are several ways that user can update the DACx_DAT register by configuring the DACLSEL in DACx_CTL register.

When DACLSEL = 000, DAC will update the output voltage level when user write data to DACx_DAT register; When DACLSEL = 001, DAC will update the output voltage level when PDMA controller send acknowledge to DAC controller. Obviously, this configuration is for DAC's PDMA usage, user should also configure other PDMA setting together with this function; When DACLSEL = 010, 011, 100 or 101, DAC will update the output voltage level every time when the corresponding timer event coming and user should write DACx_DAT before each timer event coming. Owing to the limit of DAC's conversion time, the highest conversion rate is 500 kHz (1/2us). When user updates DACx_DAT register faster than 500 kHz, some data will lose and DAC will output the latest data in DACx_DAT. Because the operating frequency (PCLK) of DAC controller will be different case by case, user can adjust the value in WAITDACCONV of DAC01_COMCTL register to meet the limit of DAC conversion rate. Power on stable time is also adjustable by setting DACx_CTL[DACPWONSTBCNT] to meet the DAC 6us stable time.

Note: DAC has two timing requirements:

- Stable time: The time DAC ready to conversion after DAC power on from DAC power down state, user should wait 6 us to meet stable time requirement by setting DACx_CTL[DACPWONSTBCNT]
- Settle time: The time DAC converts digital data to analog data. User can set DAC01_COMCTL[WAITDACCONV] to meet 2us settle time requirement

5.23.4.4 Grouping DAC0 and DAC1

Two DACx can be grouped together with the GRP bit to synchronize the update of each DAC output. Hardware ensures that those two DACs in a group update simultaneously independent of any interrupt.

The DAC0 and DAC1 are grouped by setting the GRP bit of DAC01_COMCTL.When DAC0 and DAC1 are grouped:

The DAC0's DACLSEL bits select the update trigger for both DACs

When DAC0's DACLSEL = 000, DACs will output each updated DAC_DAT simultaneously after user writing to DAC0_DAT and DAC1_DAT register, the order of writing both registers are not matter.

When DAC0's DACLSEL = 001, DACs will output each updated DAC_DAT simultaneously after PDMA sending both acknowledge to DAC controller

When DAC0's DACLSEL = 010, DACs will output each updated DAC_DAT simultaneously after Timer0's timer event coming

When DAC0's DACLSEL = 011, DACs will output each updated DAC_DAT simultaneously after Timer1's timer event coming

When DAC0's DACLSEL = 100, DACs will output each updated DAC_DAT simultaneously after Timer2's timer event coming

When DAC0's DACLSEL = 101, DACs will output each updated DAC_DAT simultaneously after Timer3's timer event coming

Note:

When DAC0 and DAC1 are grouped and without PDMA operation (DACLSEL ≠ 001b), both DACx_DAT registers must be written to before the output– update – even if data for one or both of the DACs is not changed (PDMA operation will prepare DAC_DAT automatically for both DAC)

Figures below show a latch-update timing example for grouped and ungrouped DAC0 and DAC1.

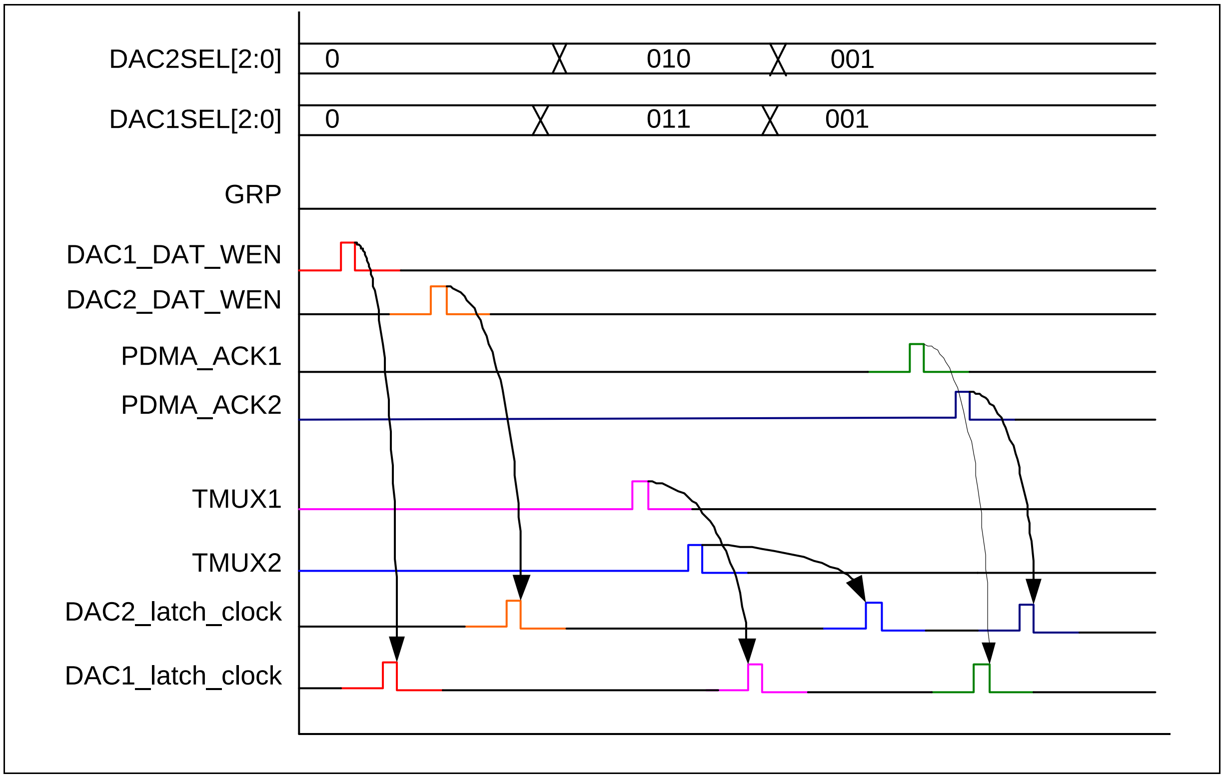


Figure 5.23-2 DAC0 and DAC1 Ungroup Update Example

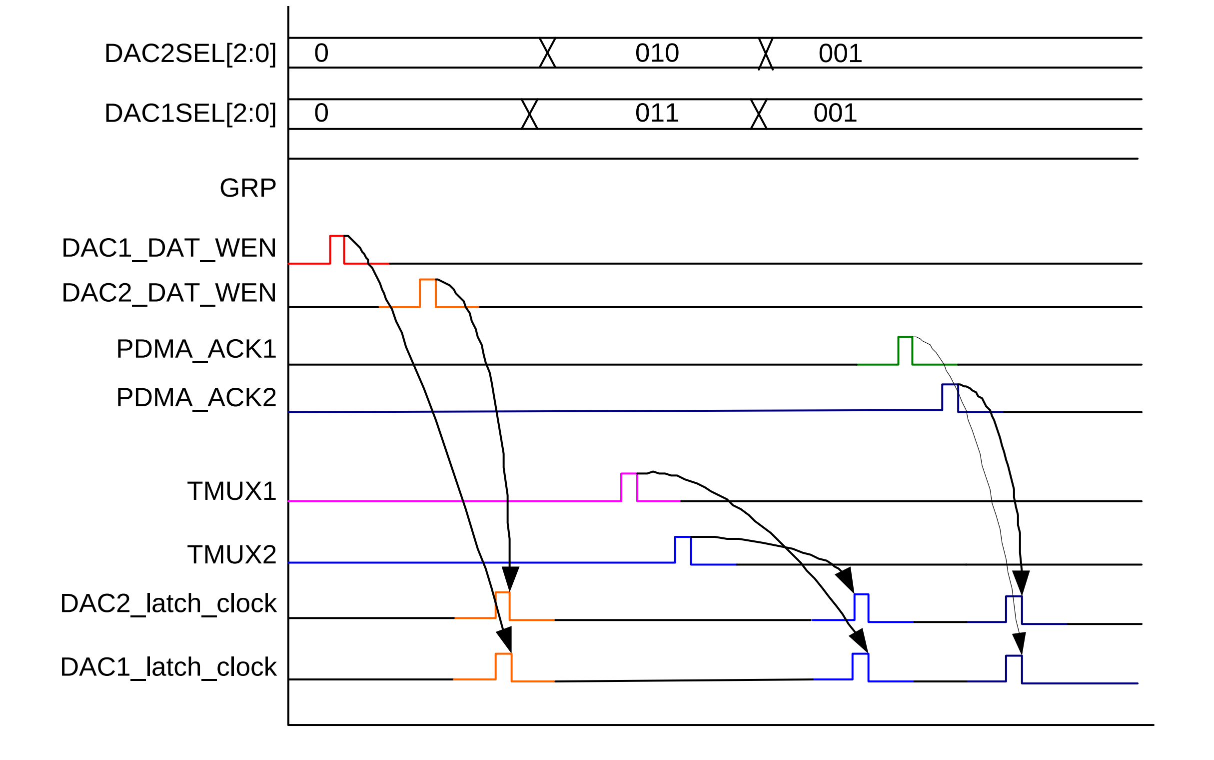


Figure 5.23-3 DAC0 and DAC1 Group Update Example

Note: DAC0,1 Settling Time:

The DMA controller is capable of transferring data to the DAC0,1 faster than the DAC0,1 output can settle. The user must assure the settling time is not violated when using the DMA controller. See the device-specific data sheet for parameters.

5.23.4.5 DAC0,1 Interrupts

DACIFG bit is set when DACx_DATA is latched internally. DACIFG bit indicates that the DACx is ready for new data. If DACIE bit is set, the DACIFG generates an interrupt request. Users must write 1 to clear DACIFG.

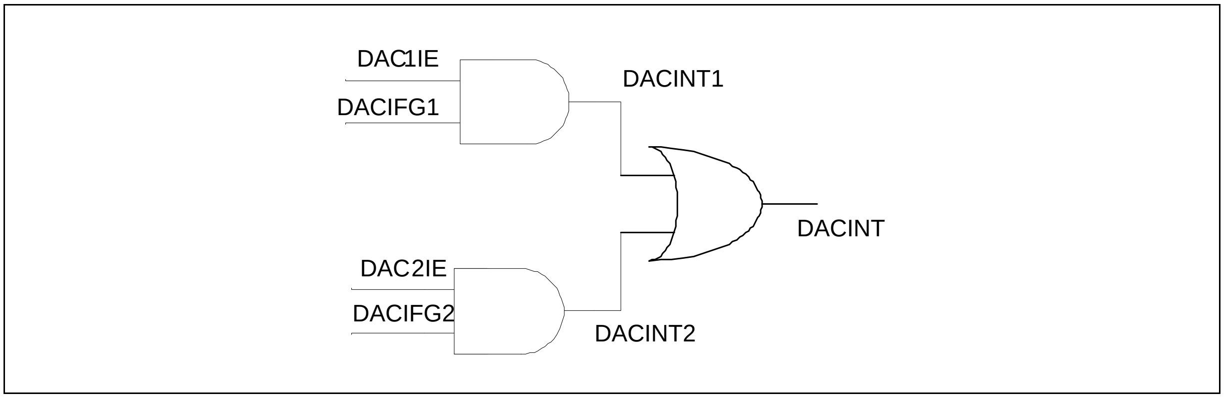


Figure 5.23-4 DAC Interrupt

5.23.5 Registers and Memory Map

R: read only, W: write only, R/W: both read and write

Register	Offset	R/W	Description	Reset Value
DAC Base Address: **DAC_BA = 0x400A_0000**				
DAC0_CTL	DAC_BA+0x00	R/W	DAC0 Control Register	0x0A00_0000
DAC0_DATA	DAC_BA+0x04	R/W	DAC0 Data Register	0x0000_0000
DAC0_STS	DAC_BA+0x08	R/W	DAC0 Status Register	0x0000_0000
DAC1_CTL	DAC_BA+0x10	R/W	DAC1 Control Register	0x0A00_0000
DAC1_DATA	DAC_BA+0x14	R/W	DAC1 Data Register	0x0000_0000
DAC1_STS	DAC_BA+0x18	R/W	DAC1 Status Register	0x0000_0000
DAC01_COMCTL	DAC_BA+0x20	R/W	DAC01 Common Control Register	0x0000_0000

5.23.6 Register Description

DAC Control Register (DAC Control, x = 0,1)

Register	Offset	R/W	Description	Reset Value
DAC0_CTL	DAC_BA+0x00	R/W	DAC0 Control Register	0x0A00_0000
DAC1_CTL	DAC_BA+0x10	R/W	DAC1 Control Register	0x0A00_0000

31	30	29	28	27	26	25	24
-							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
		DACPWONSTBCNT					
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
DACPWONSTBCNT							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-	DACLSEL			-		DACIE	DACEN

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:7]</td><td>-</td><td>Reserved</td></tr>
<tr><td>[21:8]</td><td>DACPWONSTBCNT</td><td>DACPWONSTBCNT
DAC need 6 us to be stable after DAC is power on from power down state.
This fied controls a internal counter (in PCLK unit) to guarantee DAC stable time requirement.</td></tr>
<tr><td>[6:4]</td><td>DACLSEL</td><td>DAC Load Selection
Select the load trigger for the DAC latch.
<table>
<tr><td>000</td><td>DAC latch loads when DACx_DAT written</td></tr>
<tr><td>001</td><td>PDMA ACK</td></tr>
<tr><td>010</td><td>Rising edge of TMR0</td></tr>
<tr><td>011</td><td>Rising edge of TMR1</td></tr>
<tr><td>100</td><td>Rising edge of TMR2</td></tr>
<tr><td>101</td><td>Rising edge of TMR3</td></tr>
</table></td></tr>
<tr><td>[3:2]</td><td>-</td><td>Reserved</td></tr>
</table>

Bits	Description	
[1]	**DACIE**	**DAC Interrupt Enable** 1 = Enabled 0 = Disabled
[0]	**DACEN**	**DAC Enable** 1 = Power on DAC 0 = Power down DAC **Note:** When DAC is powered on, DAC will automatically start conversion after waiting for DACPWONSTBCNT+1 PCLK cycle.

DACx Data Register (DACx_DATA, x=0,1)

Register	Offset	R/W	Description	Reset Value
DAC0_DATA	DAC_BA+0x04	R/W	DAC0 Data Register	0x0000_0000
DAC1_DATA	DAC_BA+0x14	R/W	DAC1 Data Register	0x0000_0000

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th></tr>
<tr><td colspan="8" align="center">-</td></tr>
<tr><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="8" align="center">-</td></tr>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th></tr>
<tr><td colspan="4" align="center">-</td><td colspan="4" align="center">DAC_DATA[11:8]</td></tr>
<tr><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8" align="center">DAC_DATA[7:0]</td></tr>
</table>

Bits	Description	
[31:12]	-	**Reserved**
[11:0]	**DAC Data**	**DAC data**

DAC Status Register (DACx STS,x=0,1)

Register	Offset	R/W	Description	Reset Value
DAC0_STS	DAC_BA+0x08	R/W	DAC0 Status Register	0x0000_0000
DAC1_STS	DAC_BA+0x18	R/W	DAC1 Status Register	0x0000_0000

31	30	29	28	27	26	25	24
-							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
-							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
-							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
-					**BUSY**	**DACSTFG**	**DACIFG**

Bits	Description	
[31:1]	-	**Reserved**
[2]	**BUSY**	**BUSY bit** 1 = DAC is busy. 0 = DAC is not busy.
[1]	**DACSTFG**	**DAC start flag** 1 = DAC has been started. 0 = DAC is not start yet. **Note:** this bit is read only.
[0]	**DACIFG**	**DAC Interrupt flag** 1 = Interrupt pending. 0 = No interrupt pending. **Note:** This bit is read only.

DAC Common Control Register (DAC01_COMCTL)

Register	Offset	R/W	Description	Reset Value
DAC01_COMCTL	DAC_BA+0x20	R/W	DAC01 Common Control Register	0x0000_0000

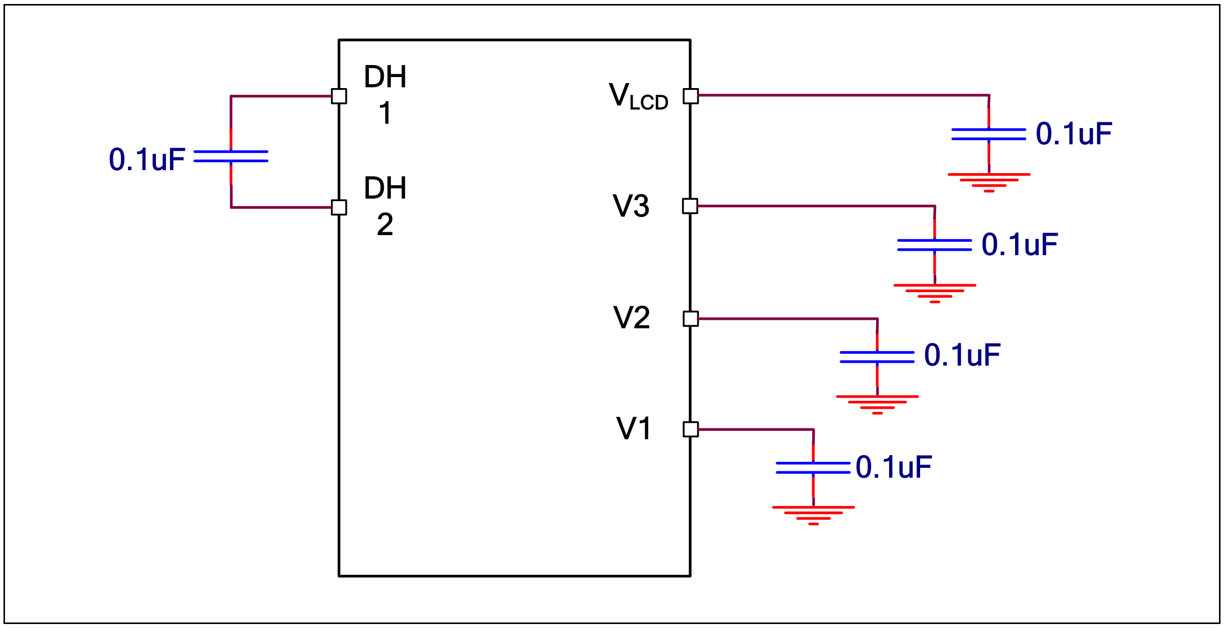
31	30	29	28	27	26	25	24
Reserved							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
Reserved							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
Reserved					**REFSEL**		**DAC12GRP**
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
WAITDACCONV[7:0]							

<table>
<tr><th>Bits</th><th colspan="2">Description</th></tr>
<tr><td>[31:11]</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>[10:9]</td><td>REFSEL</td><td>Reference Voltage Selection
<table><tr><td>00</td><td>AV_{DD}</td></tr><tr><td>01</td><td>Internal reference voltage</td></tr><tr><td>10</td><td>External reference voltage</td></tr></table>Note: Refer to Figure 5.22-1.</td></tr>
<tr><td>[8]</td><td>DAC01GRP</td><td>Group DAC0 and DAC1.
1 = Grouped
0 = Not grouped</td></tr>
<tr><td>[7:0]</td><td>WAITDACCONV</td><td>Wait DAC Conversion Complete
The DAC needs at least 2 us to settle down every time when each data deliver to DAC, which means user cannot update each DACx_data register faster than 2 us; otherwise data will lost. Setting this register can adjust the time interval in PCLK unit between each DACx_data into DAC in order to meet the 2 us requirement.</td></tr>
</table>

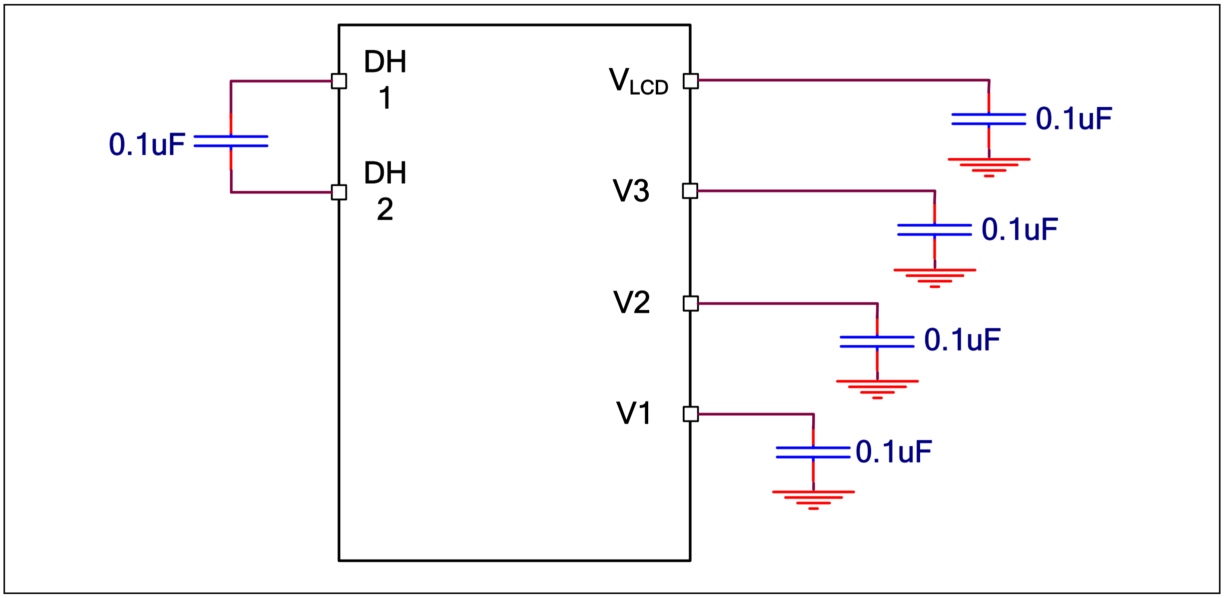
6 APPLICATION CIRCUIT

6.1 LCD Charge Pump

6.1.1 C-type 1/3 Bias

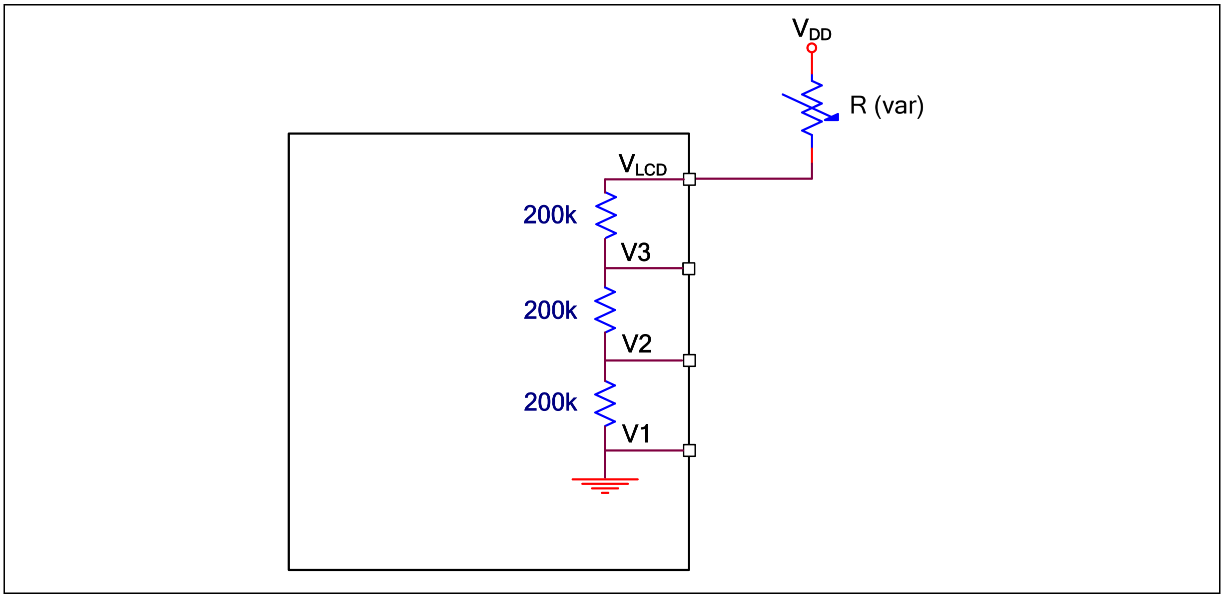


6.1.2 C-type 1/2 Bias



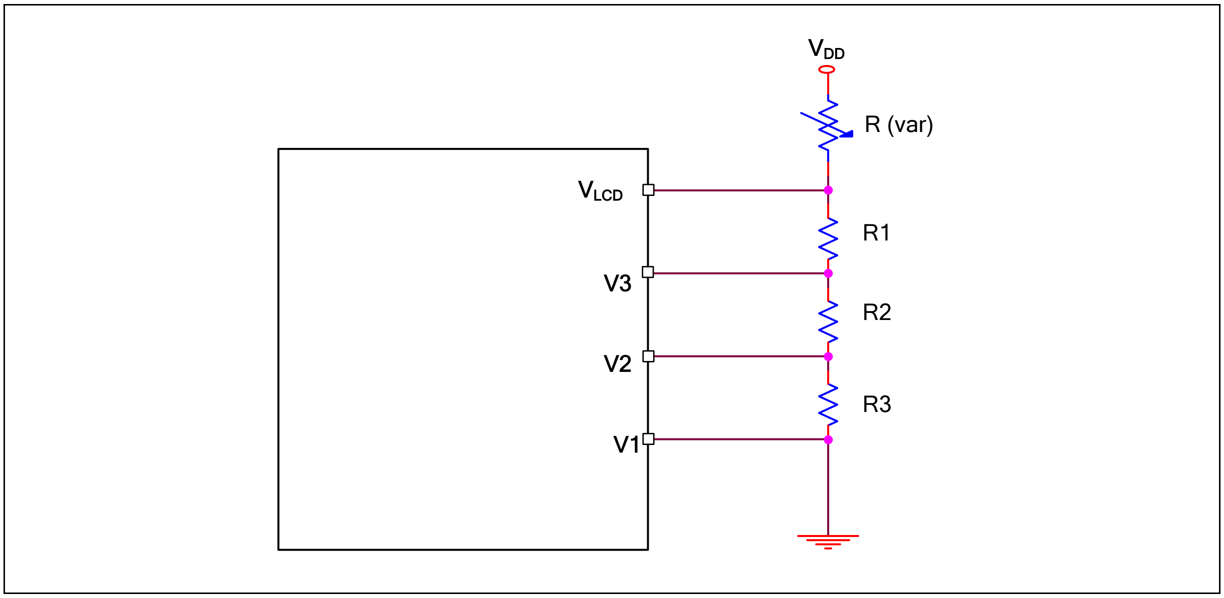
6.1.3 Internal R-type

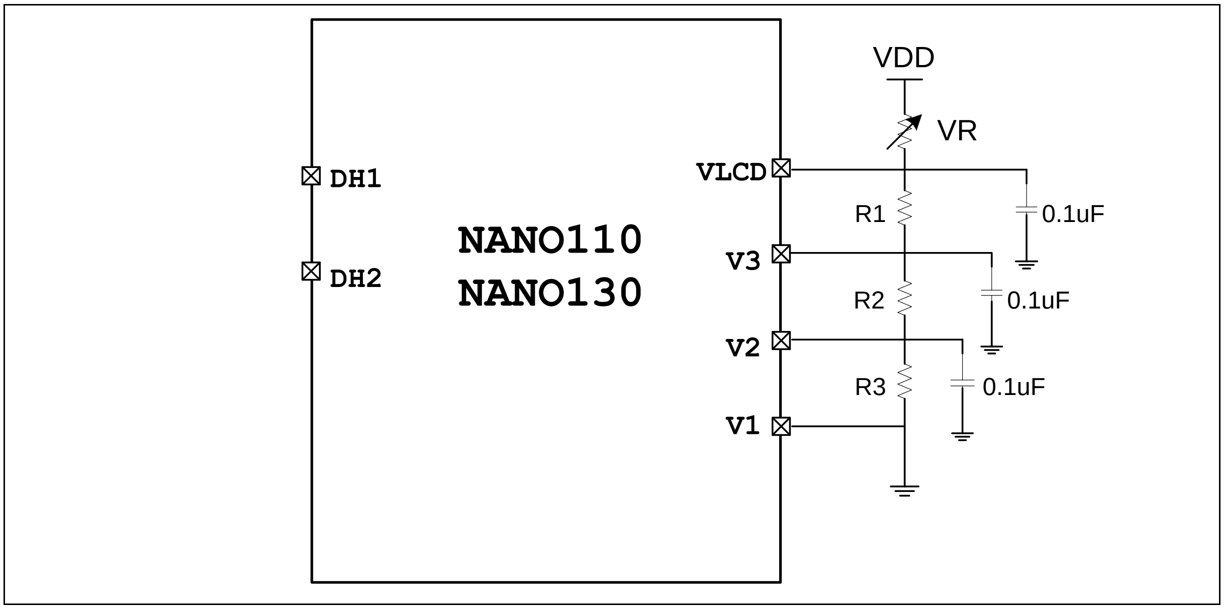
Nano110/130 series MCUs also support external R-type mode (bypass internal R) to reduce current consumption. For external R-type application, V_{LCD} is normally connected to system V_{DD}, or it can be connected to V_{DD} through an external variable resistor (VR) which is used for adjusting LCD contrast.



6.1.4 External R-type

To reduce the current, the resistor ladder value can be increased. At some point, when the resistor ladder value is increased, the contrast will become affected and the waveform shape will be altered. Therefore, capacitors around 0.1uF should be chosen and place closed to resistor ladder based on the contrast and size of the pixels on the glass.

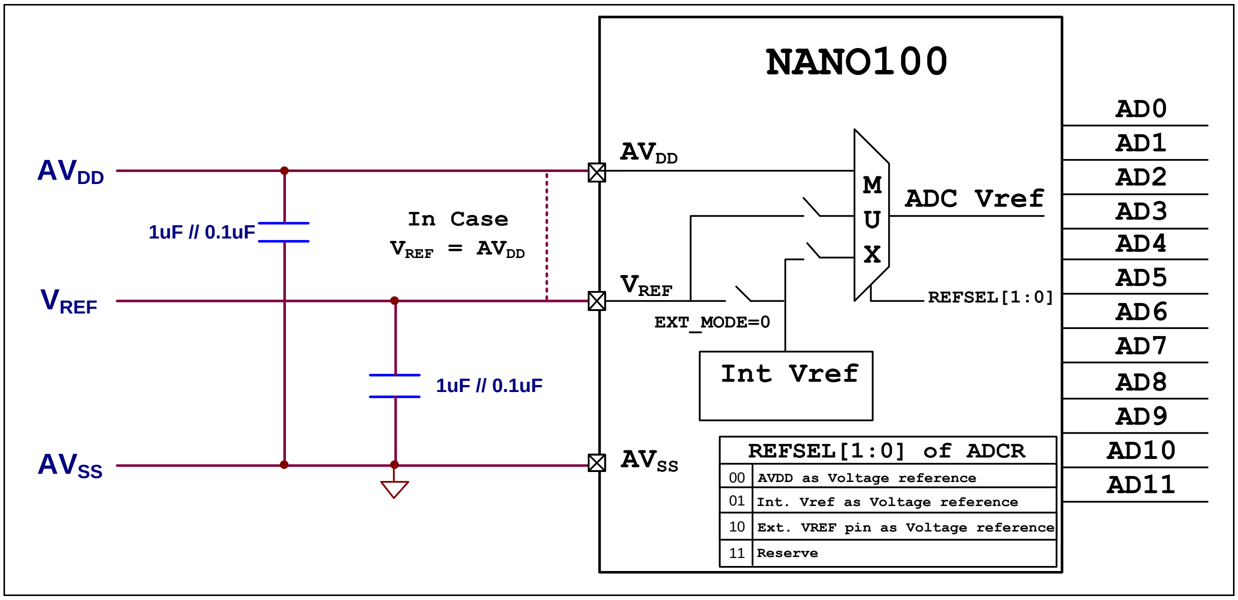



VDD
VR
VLCD
DH1
DH2
NANO110
NANO130
R1
0.1uF
V3
R2
0.1uF
V2
R3
0.1uF
V1

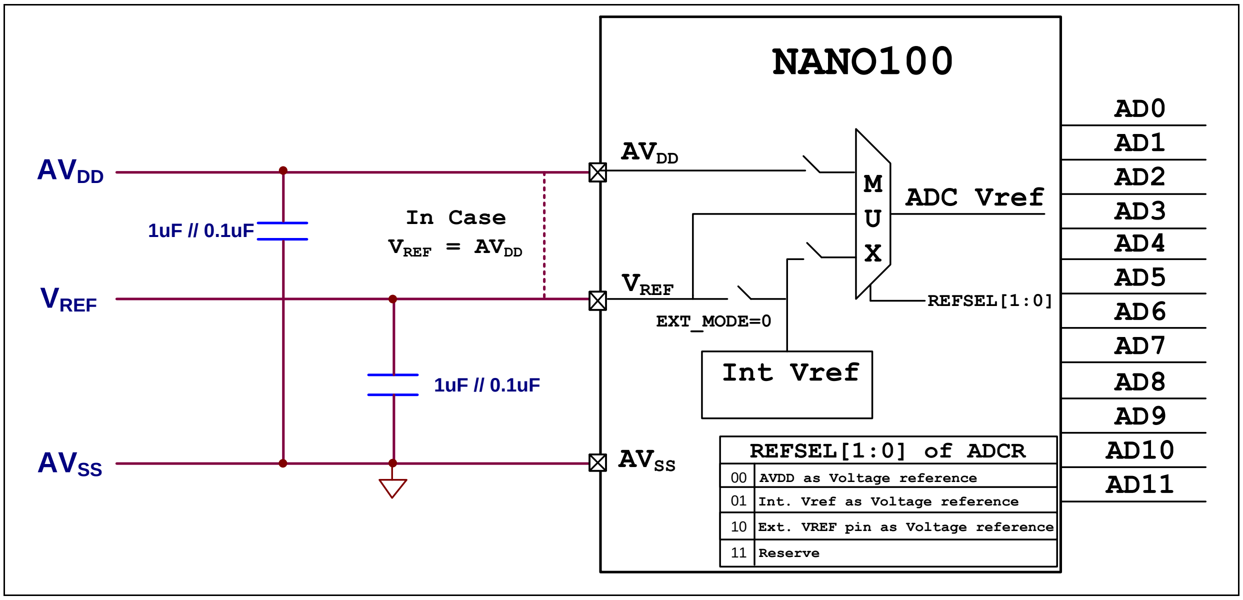
6.2 ADC Application Circuit

6.2.1 Voltage Reference Source

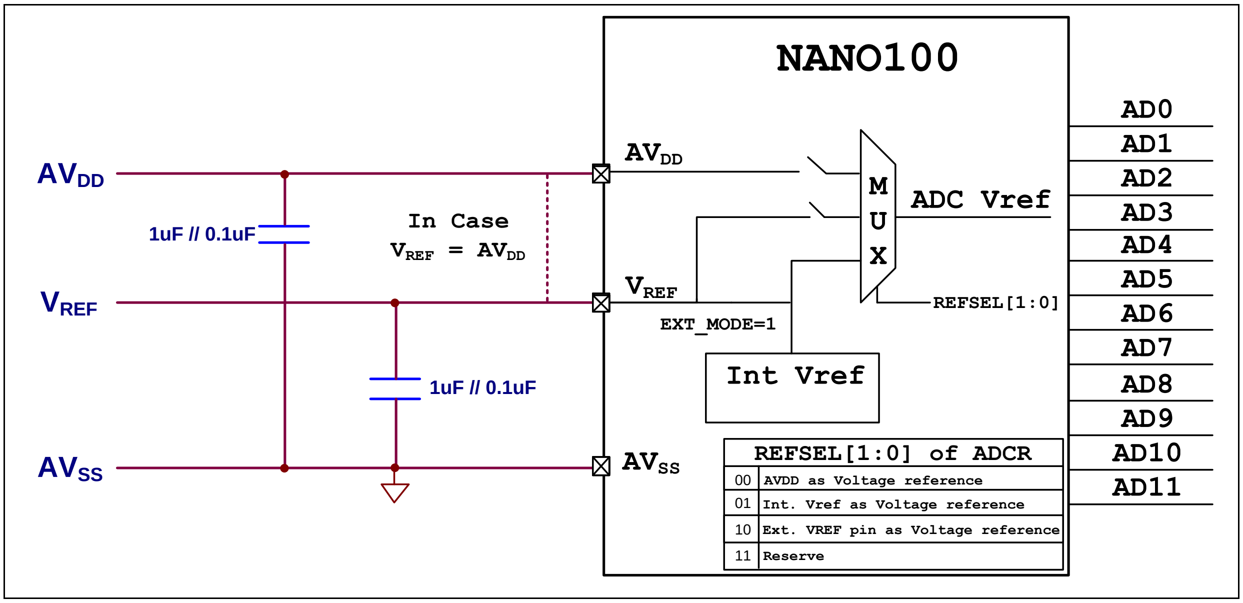
6.2.1.1 AV_{DD}



6.2.1.2 V_{REF} Pin



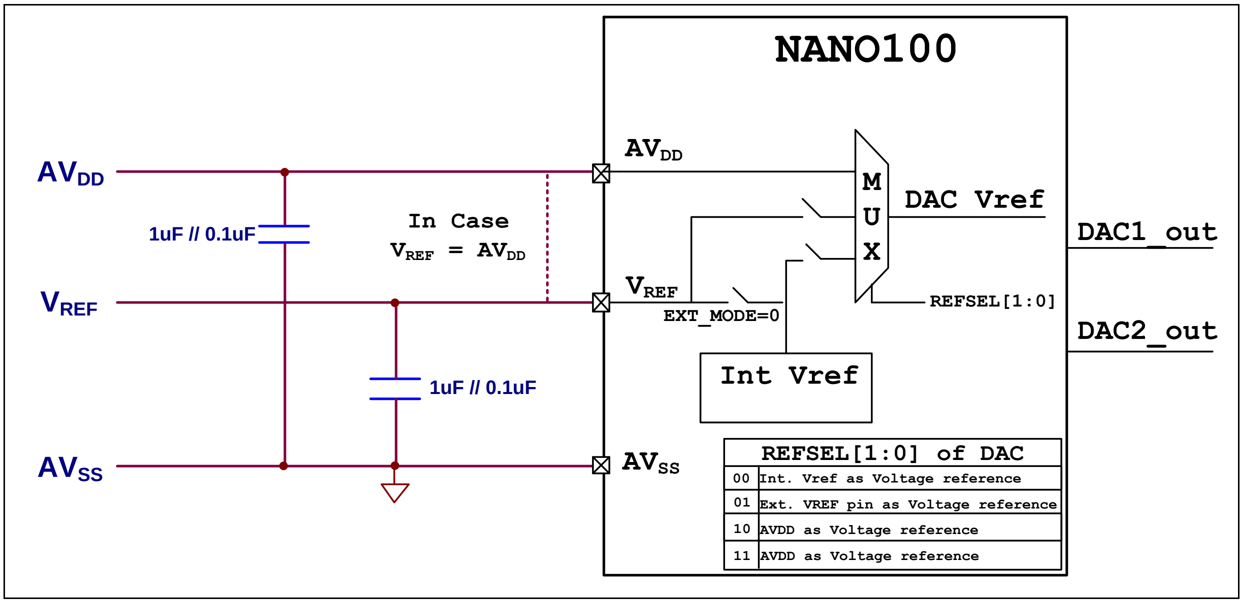
6.2.1.3 Int_VREF



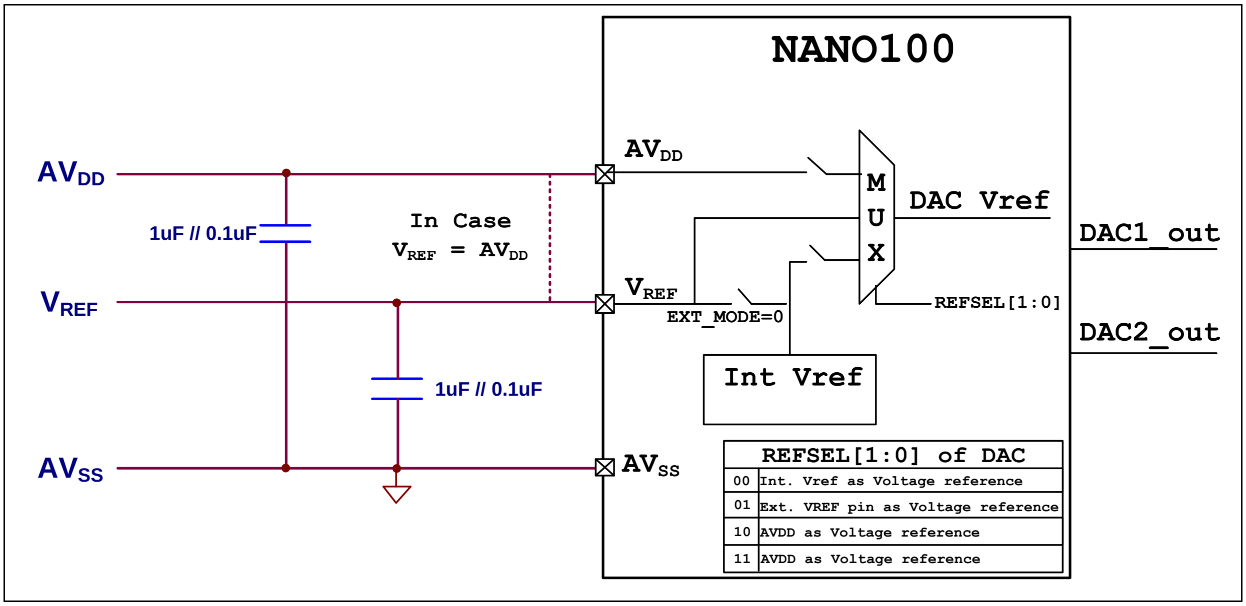
6.3 DAC Application Circuit

6.3.1 Voltage Reference Source

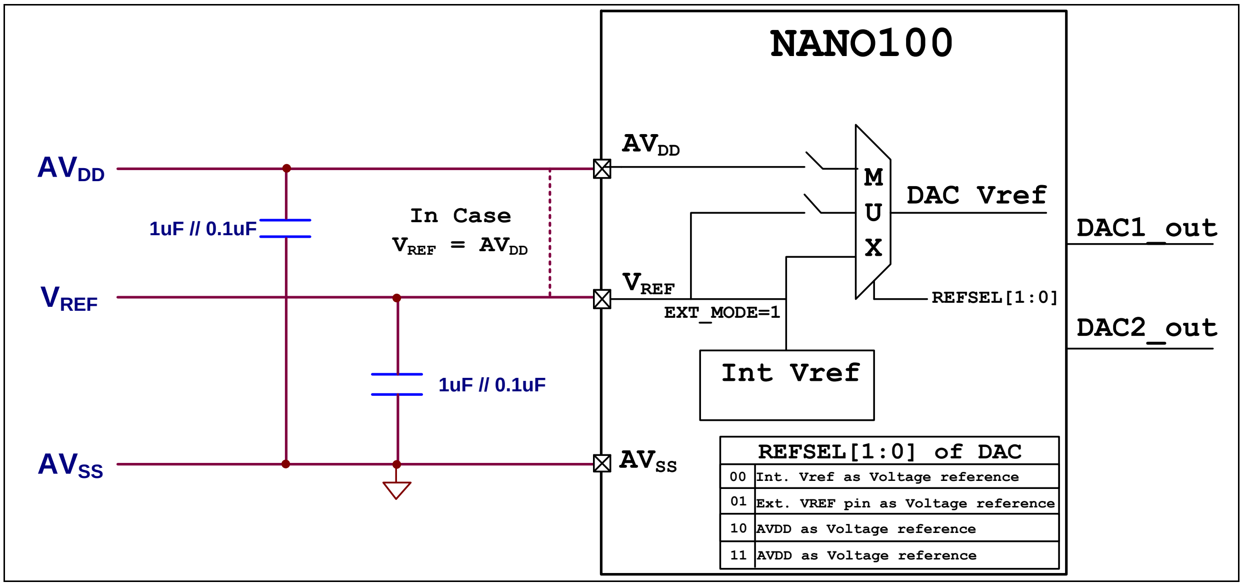
6.3.1.1 AV_{DD}



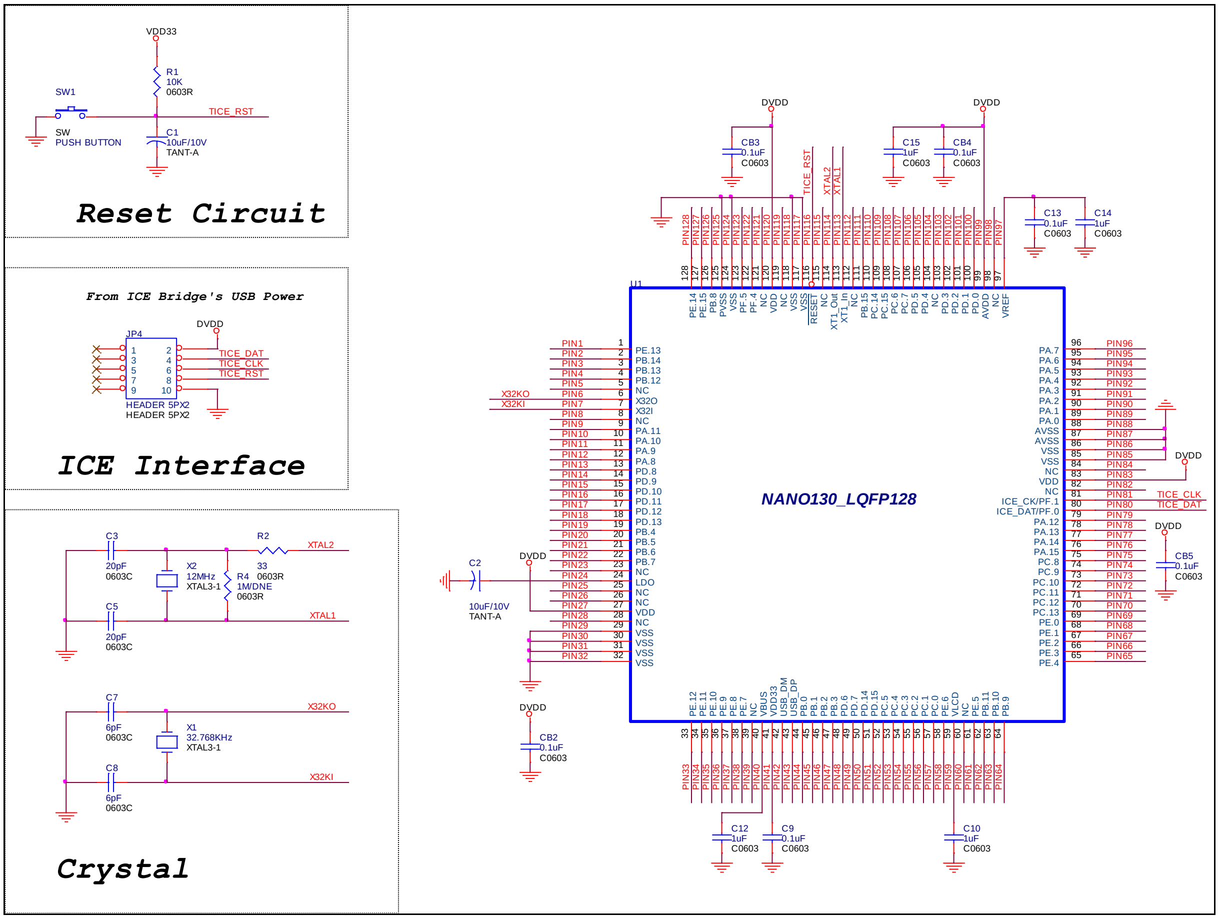
6.3.1.2 V_{REF} Pin



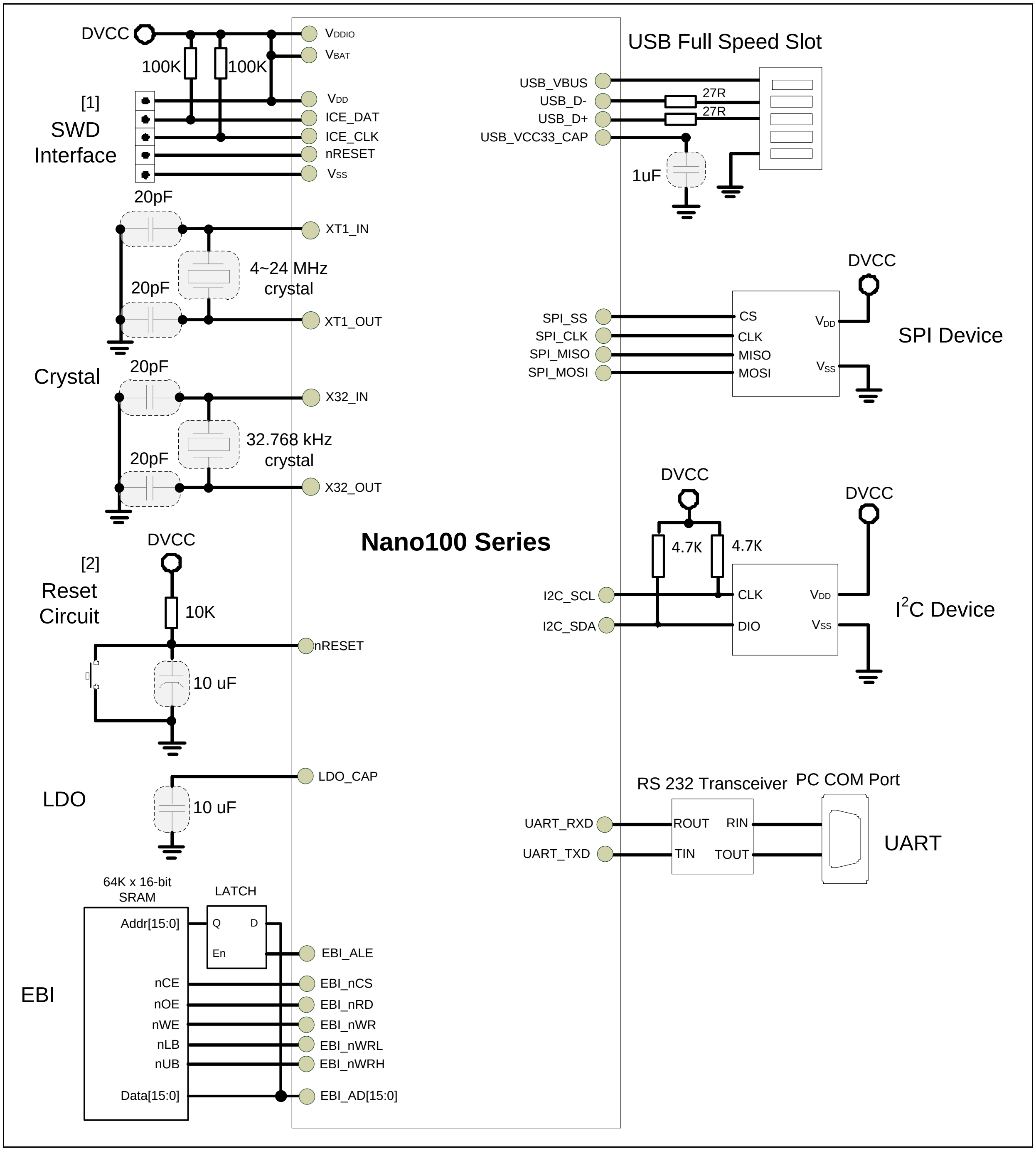
6.3.1.3 Int_VREF



6.4 Whole Chip Application Circuit



6.5 Peripheral Application Scheme



Note 1: It is recommended to use 100 kΩ pull-up resistor on both ICE_DAT and ICE_CLK pin.

Note 2: It is recommended to use 10 kΩ pull-up resistor and 10 μF capacitor on nRESET pin.

7 POWER COMSUMPTION

<table>
<tr><th>Part No</th><th colspan="2">Test Condition</th><th>VDD</th><th>CPU Clock</th><th>Current</th></tr>
<tr><td rowspan="13">Nano100 (B) series
128 KB Flash
16 KB RAM</td><td colspan="2" rowspan="2">Operating Mode:
CPU run while(1) in FLASH ROM
Clock = 12 MHz Crystal Oscillator
Disable all peripherial</td><td>3.3V</td><td>12 MHz</td><td>2.41 mA
200 µA/MHz</td></tr>
<tr><td>1.8V</td><td>12 MHz</td><td>N/A</td></tr>
<tr><td colspan="2" rowspan="2">Idle Mode:
CPU stop
Clock = 12 MHz Crystal Oscillator
Disable all peripherial</td><td>3.3V</td><td>12 MHz</td><td>900 µA
75 µA/MHz</td></tr>
<tr><td>1.8V</td><td>12 MHz</td><td>N/A</td></tr>
<tr><td rowspan="4">RTC + LCD Mode:
(RAM retention)
(Power down with 32K and LCD enabled)
CPU stop
Clock = 32.768 kHz Crystal Oscillator
Disable all peripherial except RTC and LCD circuit
Without panel loading</td><td>C-type</td><td rowspan="3">3.3V</td><td rowspan="3">-</td><td>10 µA</td></tr>
<tr><td>Internal R-type
(With 200kΩ Resistor ladder)</td><td>8.5 µA</td></tr>
<tr><td>External R-type
(With 1MΩ Resistor ladder)</td><td>4.5 µA</td></tr>
<tr><td>C-type/R-type</td><td>1.8V</td><td>-</td><td>N/A</td></tr>
<tr><td colspan="2" rowspan="2">RTC Mode: (RAM retention)
(Power down with 32K enabled)
CPU stop
Clock = 32.768 kHz Crystal Oscillator
Disable all peripherial except RTC circuit</td><td>3.3V</td><td>-</td><td>2.5 µA</td></tr>
<tr><td>1.8V</td><td>-</td><td>2.0 µA</td></tr>
<tr><td colspan="2" rowspan="2">Power-down Mode: (RAM retention)
CPU and all clocks stop</td><td>3.3V</td><td>-</td><td>1 µA</td></tr>
<tr><td>1.8V</td><td>-</td><td>0.8 µA</td></tr>
<tr><td colspan="2">Wake-Up from Power-down Mode</td><td>3.3V</td><td>7 µs</td><td>N/A</td></tr>
</table>

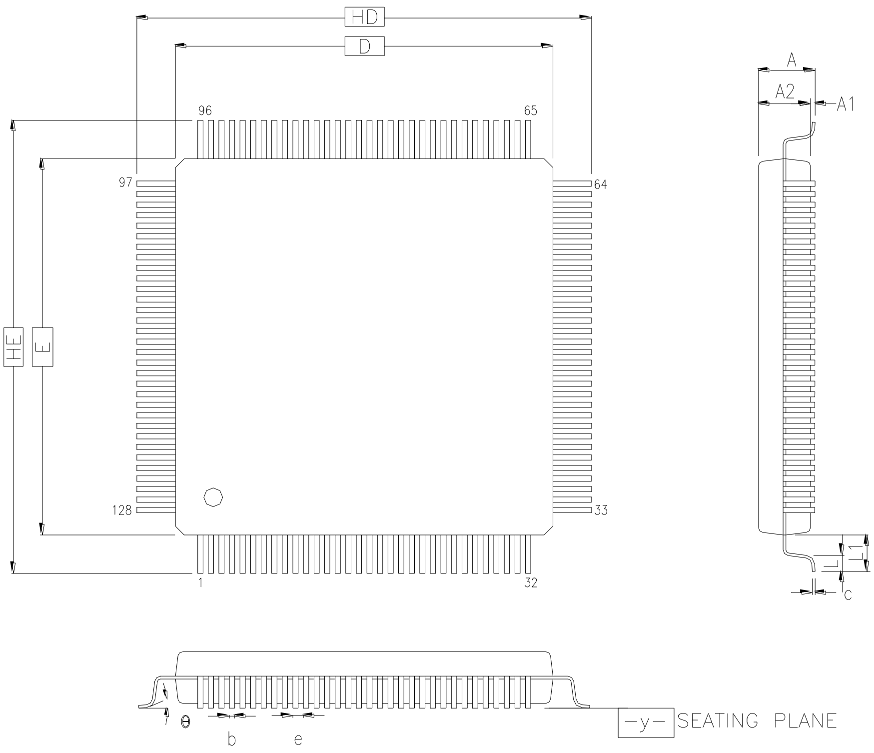
Note: Wake-up time: 7 µs from wake-up event to first CPU core valid clock; 10 µs from interrupt event to interrupt service routine first instruction.

8 ELECTRICAL CHARACTERISTIC

For information on the Nano100 Series electrical characteristics, please refer to NuMicro® Nano100 Series Datasheet.

9 PACKAGE DIMENSIONS

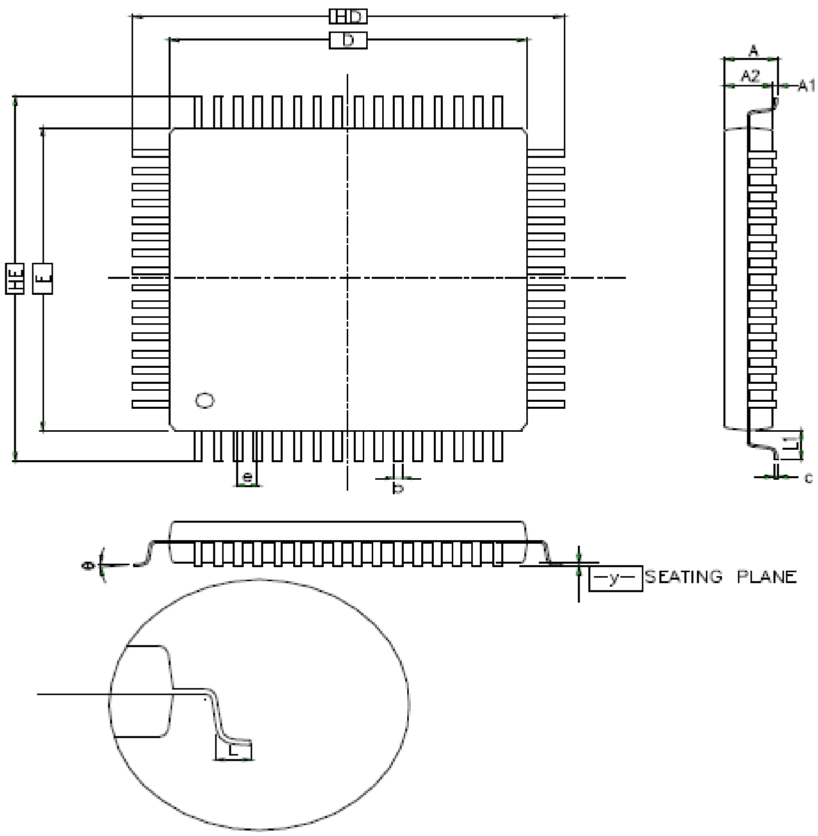
9.1 LQFP128 (14x14x1.4 mm footprint 2.0 mm)



COTROL DIMENSIONS ARE IN MILLIMETERS.

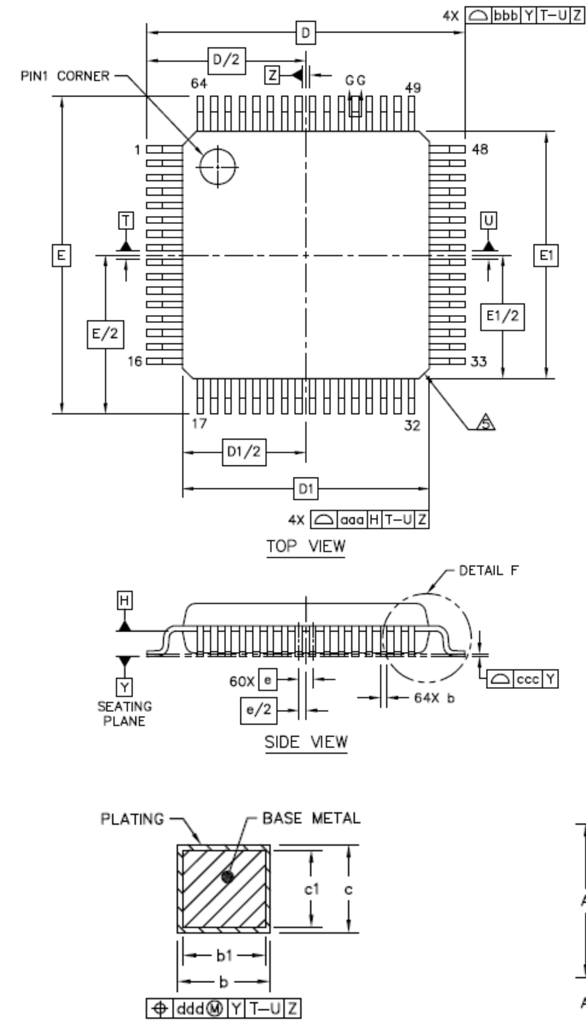
SYMBOL	MILLIMETER			INCH		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	—	—	1.60	—	—	0.063
A_1	0.05	—	0.15	0.002	—	0.006
A_2	1.35	1.40	1.45	0.053	0.055	0.057
HD	16.00 BSC.			0.630 BSC.		
D	14.00 BSC.			0.551 BSC.		
HE	16.00 BSC.			0.630 BSC.		
E	14.00 BSC.			0.551 BSC.		
b	0.13	0.16	0.23	0.005	0.006	0.009
e	0.40 BSC.			0.016 BSC.		
θ	0°	3.5°	7°	0°	3.5°	7°
c	0.09	—	0.20	0.004	—	0.008
L	0.45	0.60	0.75	0.018	0.024	0.030
L_1	1.00 REF			0.039 REF		
y	–	–	0.1	–	–	0.004

9.2 LQFP64 (10x10x1.4 mm footprint 2.0 mm)

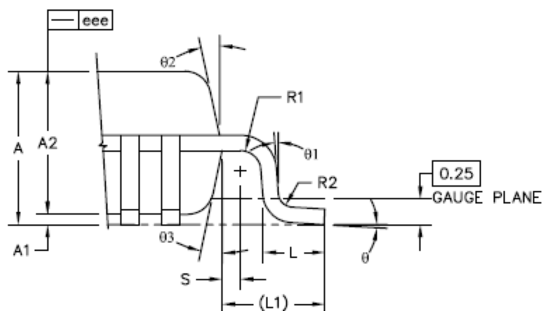


Symbol	Dimension in inch			Dimension in mm		
	Min	Nom	Max	Min	Nom	Max
A	—	—	0.063	—	—	1.60
A_1	0.002	—	0.006	0.05	—	0.15
A_2	0.053	0.055	0.057	1.35	1.40	1.45
b	0.007	0.008	0.011	0.17	0.20	0.27
c	0.004	—	0.008	0.09	—	0.20
D	—	0.393	—	—	10.00	—
E	—	0.393	—	—	10.00	—
$\boxed{e}$	—	0.020	—	—	0.50	—
H_D	—	0.472	—	—	12.00	—
H_E	—	0.472	—	—	12.00	—
L	0.018	0.024	0.030	0.45	0.60	0.75
L_1	—	0.039	—	—	1.00	—
y	—	0.004	—	—	0.10	—
θ	0°	3.5°	7°	0°	3.5°	7°

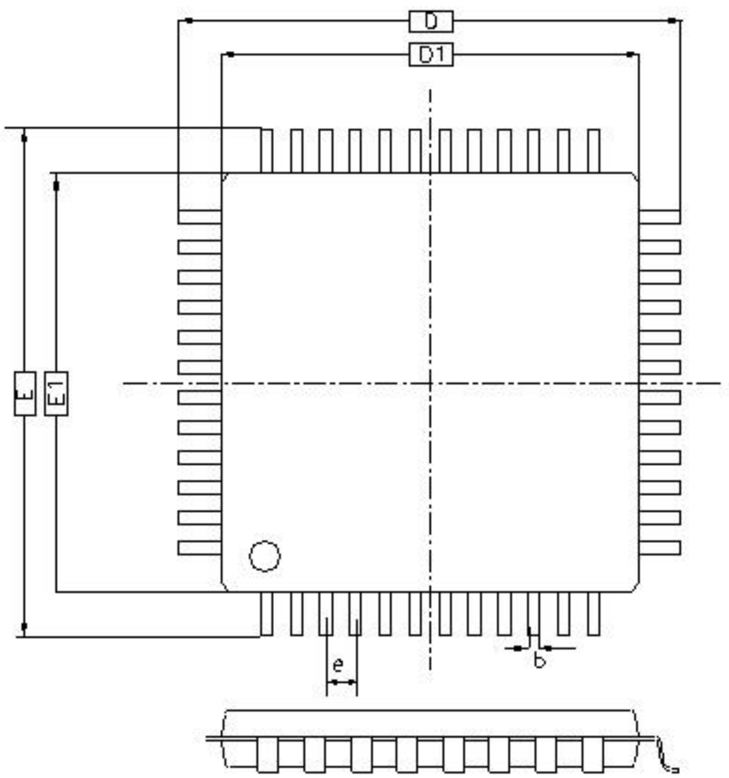
9.3 LQFP64 (7x7x1.4 mm footprint 2.0 mm)

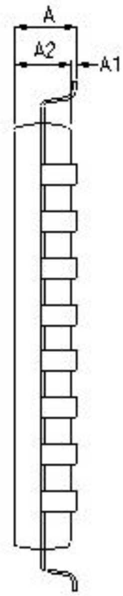


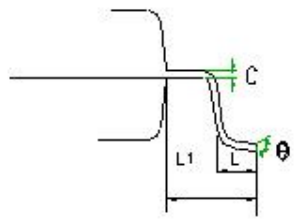
		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	---	---	1.6
STAND OFF		A1	0.05	---	0.15
MOLD THICKNESS		A2	1.35	1.4	1.45
LEAD WIDTH(PLATING)		b	0.13	0.18	0.23
LEAD WIDTH		b1	0.13	0.16	0.19
L/F THICKNESS(PLATING)		c	0.09	---	0.2
L/F THICKNESS		c1	0.09	---	0.16
BODY SIZE	X	D		9 BSC	
	Y	E		9 BSC	
	X	D1		7 BSC	
	Y	E1		7 BSC	
LEAD PITCH		e		0.4 BSC	
		L	0.45	0.6	0.75
FOOTPRINT		L1		1 REF	
		θ	0°	3.5°	7°
		θ1	0°	---	---
		θ2	11°	12°	13°
		θ3	11°	12°	13°
		R1	0.08	---	---
		R2	0.08	---	0.2
		S	0.2	---	---
PACKAGE EDGE TOLERANCE		aaa		0.2	
LEAD EDGE TOLERANCE		bbb		0.2	
COPLANARITY		ccc		0.08	
LEAD OFFSET		ddd		0.07	
MOLD FLATNESS		eee		0.05	



9.4 LQFP48 (7x7x1.4 mm footprint 2.0 mm)



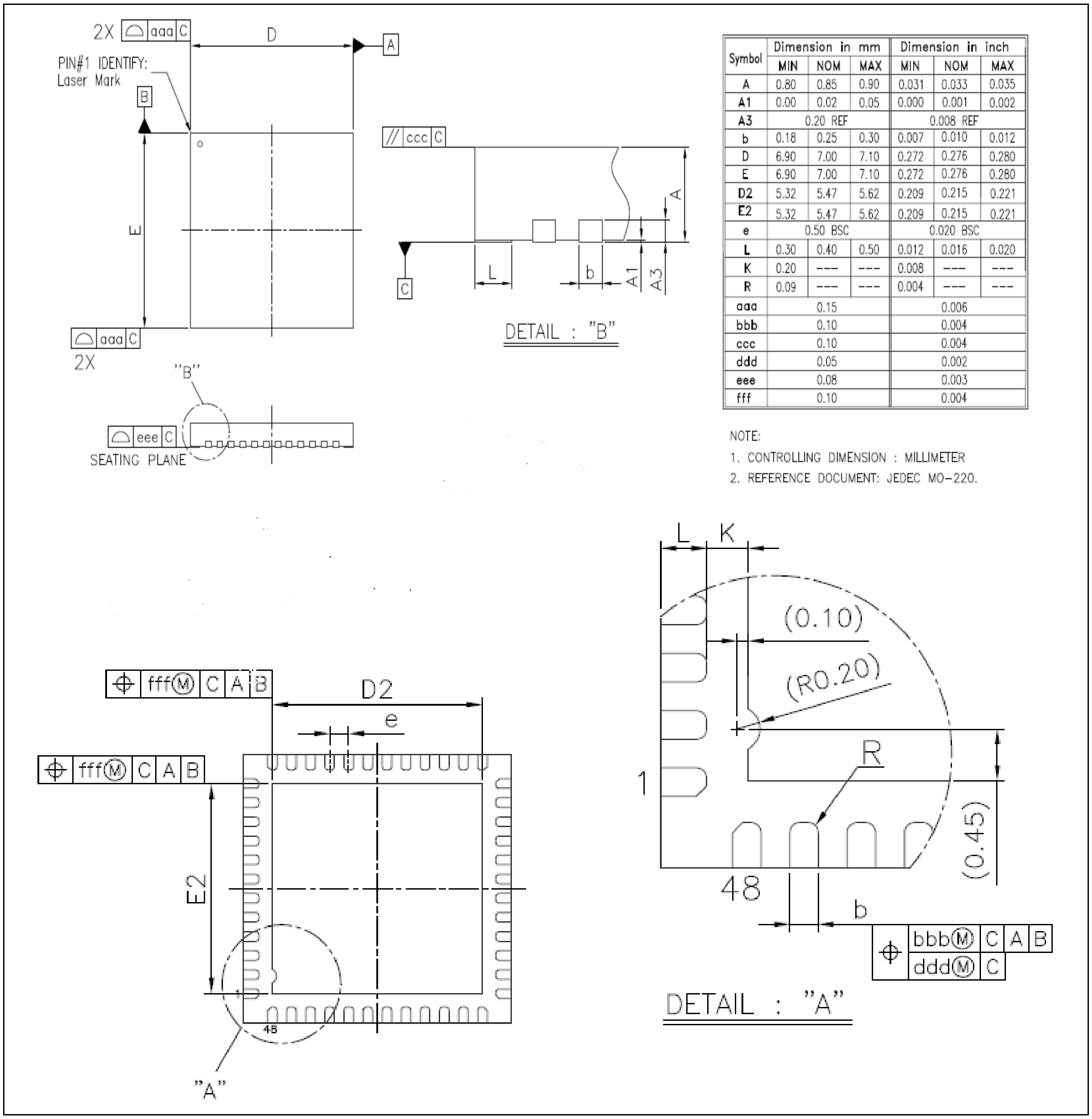




COTROL DIMENSIONS ARE IN MILLIMETERS.

SYMBOL	MILLIMETER			INCH		
	MIN	NOM.	MAX.	MIN	NOM.	MAX.
A	—	—	1.60	—	—	0.063
A_1	0.05	0.10	0.15	0.002	0.004	0.006
A_2	1 35	1.40	1.45	0.053	0 055	0.057
D1	6.90	7.00	7.10	0.272	0.276	0.260
E1	6 90	7.00	7 10	0.272	0.276	0.260
e	0.35	0.50	0.65	0.014	0.020	0.260
D	8.9	9.00	9.10	0.350	0.354	0.358
E	8.9	9.00	9.10	0.350	0.354	0.358
L	0 45	0.60	0 75	0.018	0 024	0 030
L1	—	1 00	—	—	0.039	—
C	0.09	—	0.20	0.0035	—	0.0079
θ	0°	—	7°	0°	—	7°
b	0.17	0.22	0.27	0.007	0.0087	0.011

9.5 QFN48 (7x7x0.85 mm)



Symbol	Dimension in mm MIN	NOM	MAX	Dimension in inch MIN	NOM	MAX
A	0.80	0.85	0.90	0.031	0.033	0.035
A1	0.00	0.02	0.05	0.000	0.001	0.002
A3		0.20 REF			0.008 REF	
b	0.18	0.25	0.30	0.007	0.010	0.012
D	6.90	7.00	7.10	0.272	0.276	0.280
E	6.90	7.00	7.10	0.272	0.276	0.280
D2	5.32	5.47	5.62	0.209	0.215	0.221
E2	5.32	5.47	5.62	0.209	0.215	0.221
e		0.50 BSC			0.020 BSC	
L	0.30	0.40	0.50	0.012	0.016	0.020
K	0.20	---	---	0.008	---	---
R	0.09	---	---	0.004	---	---
aaa		0.15			0.006	
bbb		0.10			0.004	
ccc		0.10			0.004	
ddd		0.05			0.002	
eee		0.08			0.003	
fff		0.10			0.004	

NOTE:
1. CONTROLLING DIMENSION : MILLIMETER
2. REFERENCE DOCUMENT: JEDEC MO-220.

10 REVISION HISTORY

Date	Revision	Description
2012.10.11	1.00	Initial release
2012.12.11	1.01	1. Added SmartCard UART mode description in Pin Description. 2. Unified the abbreviation (TMR) in the Timer Controller section. 3. Modified the specifications of external input clock. 4. Added LCD COM4 and COM5 description for each pin description and diagram. 5. Updated the ADC enabled by timer event description in the ADC section. 6. Changed Timer0/1 Ch0/1 to Timer x (x=0, 1, 2, 3) in the Timer Controller section. 7. Added register description of Countinuous Counting mode in Timer Controller section.
2012.12.17	1.02	1. Added description of reading UCID in ISP mode.
2012.12.28	1.03	1. Added a table about entering Power-down mode again in section 5.4.3. 2. Added R-type related description in LCD section. 3. Updated the operating current data of Run mode and Idle mode at each frequency and added related data at 42 MHz in chapter 8.
2013.01.02	1.04	1. Updated the table in Power Consumption section.
2013.03.05	1.05	1. Updated the descriptions in sections 5.20.4.5 and 5.22.4.8. 2. Updated the display modes from four to six in section 5.21.2. 3. Corrected the pin descriptions in section 3.4. 4. Updated bit 31 (CWDT_EN) description for Config0 in section 5.6.4.4. 5. Updated measuring condition in Figure 5.19-5 and Figure 5.19-6. 6. Added temperature sensor related description in section 5.22.4.13. 7. Updated clock control block diagram in section 5.4.3. 8. Updated temperature sensor of analog characteristic in chapter 8. 9. Corrected the setting of DMA_TX_EN and DMA_RX_EN registers in section 5.8.6. 10. Corrected the description and the example of RTC frequency compensation in section 5.13.4.4. 11. Corrected the formula of RTC_FCR [FRACTION] in section 5.13.6. 12. Corrected the channel of fast and slow input in section 5.22.4.9. 13. Corrected the setting of PDMA_TCR and TO_EN registers in section 5.8.6. 14. Corrected the description of PDSTS register in section 5.21.6.

		15. Corrected UART Clock Control Diagram in section 5.14.4. 16. Corrected IrDA TX/RX Timing Diagram in section 5.14.4.4. 17. Corrected Smart Card's feature to be half duplex in UART mode in section 5.15.2. 18. Added the description, "The Watchdog counter will be automatically reset when the chip is entering Power-down mode." in section 5.11.4.
2013.05.28	1.06	1. Added the MUX (LCD_CTL[3:1]) description for setting LCD_SEG0/1 or LCD_COM4/5 in section 5.21.6. 2. Updated the Nano110 LQFP128-pin diagram in section 3.3.2. 3. Added the Module Clock Output (MCLKO) register description and related settings in section 5.5.5. 4. Updated "12 MHz OSC has 2 % deviation within all temperarure range" in sections 2.1 to 2.4. 5. Updated the connection related description of CH1_SEL (DMA_DSSR0[12:8]) and CH4_SEL(DMA_DSSR1[12:8]) registers in section 5.8.6. 6. Updated DAC analog characteristics in chapter 8. 7. Modified the HXT_HF_ST (PWRCTL[12:11]) description in section 5.5.6. 8. Added Nano110RC2BN to the Nano110 LCD Line Selection Guide. 9. Corrected the typo of PA2_MFP and PA3_MFP register description in section 5.4.5.
2013.12.04	1.07	1. Updated Nano100 series selection code in section 3.1. 2. Added the Nano100 QFN48 package in section 3.2 and QFN48 package dimensions in chapter 9. 3. Modified the reset value of ADCPWD register and the bit description of PWDMOD in section 5.22.6. 4. Updated the FOUT related description "FOUT frequency must be greater than 48 MHz and less than 120 MHz" in section 5.5.6. 5. Fixed the typo of LCD characteristic in chapter 8. 6. Added a note that "When EVENT_EN is enabled, EXT_TMx(GPB) cannot be selected as clock source. However, the speed of selected clock must be 3 times greater than the speed of EXT_TMx(GPB)." in section 5.9.6. 7. Added a note that "Output voltage for ADC/LCD shared pins cannot be higher than V_{DD} because these pins are without 5V tolerance." for pin description in section 3.4, LCD overview in section 5.21.1 and Absolute Maximum Ratings in chapter 8. 8. Modified the schematic for ADC and DAC application circuit in section 6.2 and 6.3.
2016.03.31	1.08	1. Modified the pin description in section 3.4. 2. Added a note in all clock sourse block diagram of all peripherals sections that "Before clock switching, both the pre-selected and

		newly selected clock sources must be turned on and stable." 3. Updated bit fied description and note for PWDMOD(ADCPWD[3:2]) and PWDCALEN(ADCPWD[1]) in section 5.22.6. 4. Updated LCD application circuirt in section 6.1. 5. Fixed the typo of LCD Feature in section 5.21.2. 6. Added Internal Reference Voltage description in section 5.22.4.14
2019.07.19	1.09	1. Fixed the LCD clock source LIRC typo in section 5.5.4.2. 2. Fixed the typo of PB9 LCD_V3 and PB11 LCD_V1 in section 5.4.5. 3. Fixed the typo of PLL frequency in section 5.5.6. 4. Modified the LCD charge pump capacitor's conditions in section 5.21.7. 5. Fixed the Timer and UART clock status in Table 5.4-2. 6. Added the content of DMA register in section 5.8.5. 7. Removed the content in chapter 8. Please refer to datasheet. 8. Updated ADC and DAC application circuit in section 6.2 and 6.3.
2020.04.23	1.10	1. Added peripheral application scheme in section 6.5. 2. Added notes about the hardware reference design for ICE_DAT, ICE_CLK and nRESET pins in section 3.4 and 6.5.

Important Notice

Nuvoton Products are neither intended nor warranted for usage in systems or equipment, any malfunction or failure of which may cause loss of human life, bodily injury or severe property damage. Such applications are deemed, "Insecure Usage".

Insecure usage includes, but is not limited to: equipment for surgical implementation, atomic energy control instruments, airplane or spaceship instruments, the control or operation of dynamic, brake or safety systems designed for vehicular use, traffic signal instruments, all types of safety devices, and other applications intended to support or sustain life.

All Insecure Usage shall be made at customer's risk, and in the event that third parties lay claims to Nuvoton as a result of customer's Insecure Usage, customer shall indemnify the damages and liabilities thus incurred by Nuvoton.