M261 Power Management

Application Note for 32-bit NuMicro® Family

Document Information

Abstract	Introduce power modes including configuration, wake-up method and sample code.
Apply to	NuMicro® M261 series

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

www.nuvoton.com

Table of Contents

1 Overview

With the development of Internet of Things (IoT), devices originally work independently can be provided with sensors to communicate with each other through the Internet now. In the IoT application, microcontrollers (MCUs) need to process sensor data and network communication with high efficiency, and require low power consumption to increase the device lifetime. The NuMicro® M261 series microcontroller provides some power modes with different power consumption level and wake-up time. In the following chapters, Chapter 2 introduces different power modes. Chapter 3 demonstrates sample code of power mode configuration and wake-up method. The last Chapter 4 lists the power modes comparisons for user reference to design their low power and efficient application.

2 Power Mode Introduction

The NuMicro® M261 power modes include Normal mode, Idle mode and Power-down mode. The system starts up in Normal mode. In Idle mode, only CPU clock is disabled while other peripherals work normally. If system is waiting for an interrupt only, you can set system in Idle mode to save power and wake-up quickly. If system does not need to work for a long time, you can set system in Power-down mode. The M261 provides some different Power-down modes with different power consumption. But the lower power consumption mode needs more wake-up time and causes more data lost. This section introduces these power modes, configuration and wake-up method.

The M261 series includes the following power modes:

- Normal mode
  System startup power mode supplies power to all system, peripherals and memories.
- Idle mode
  Only CPU clock is disabled. Other peripherals keep working normally. The fastest power mode returns to Normal mode.
- Power-down mode
  Besides CPU clock is disabled, more clock sources and peripherals are disabled to save power. Wake-up time is longer because after system wake-up, it needs to wait clock stable and initialization. Some power modes even disable memories power that causes data lost. There are six Power-down modes:
  1. Normal Power-down mode (PD)
  2. Low Leakage Power-down mode (LLPD)
  3. Fast Wake-up Power-down mode (FWPD)
  4. Ultra Low Leakage Power-down mode (ULLPD)
  5. Standby Power-down mode (SPD)
  6. Deep Power-down mode (DPD)

Figure 2-1 shows the NuMicro® M261 power distribution:

- Digital power V_{DD} supplies power to HXT, LIRC, I/O, DPD control and internal voltage regulator. The internal voltage regulator supplies power to other digital logics which are divided into two groups $V_{DD(standby)}$ and $V_{DD(core)}$ domain.
- $V_{DD(standby)}$ supplies power to Standby Power-down mode (SPD) control logic which controls wake-up sources of SPD and supplies power to system SRAM bank0 to retain data.

- $V_{DD(core)}$ supplies power to HIRC, CPU, peripherals, Flash and system SRAM bank1.
- Analog power AV_{DD} supplies power to analog peripherals.
- RTC power V_{BAT} supplies power to LXT and RTC.

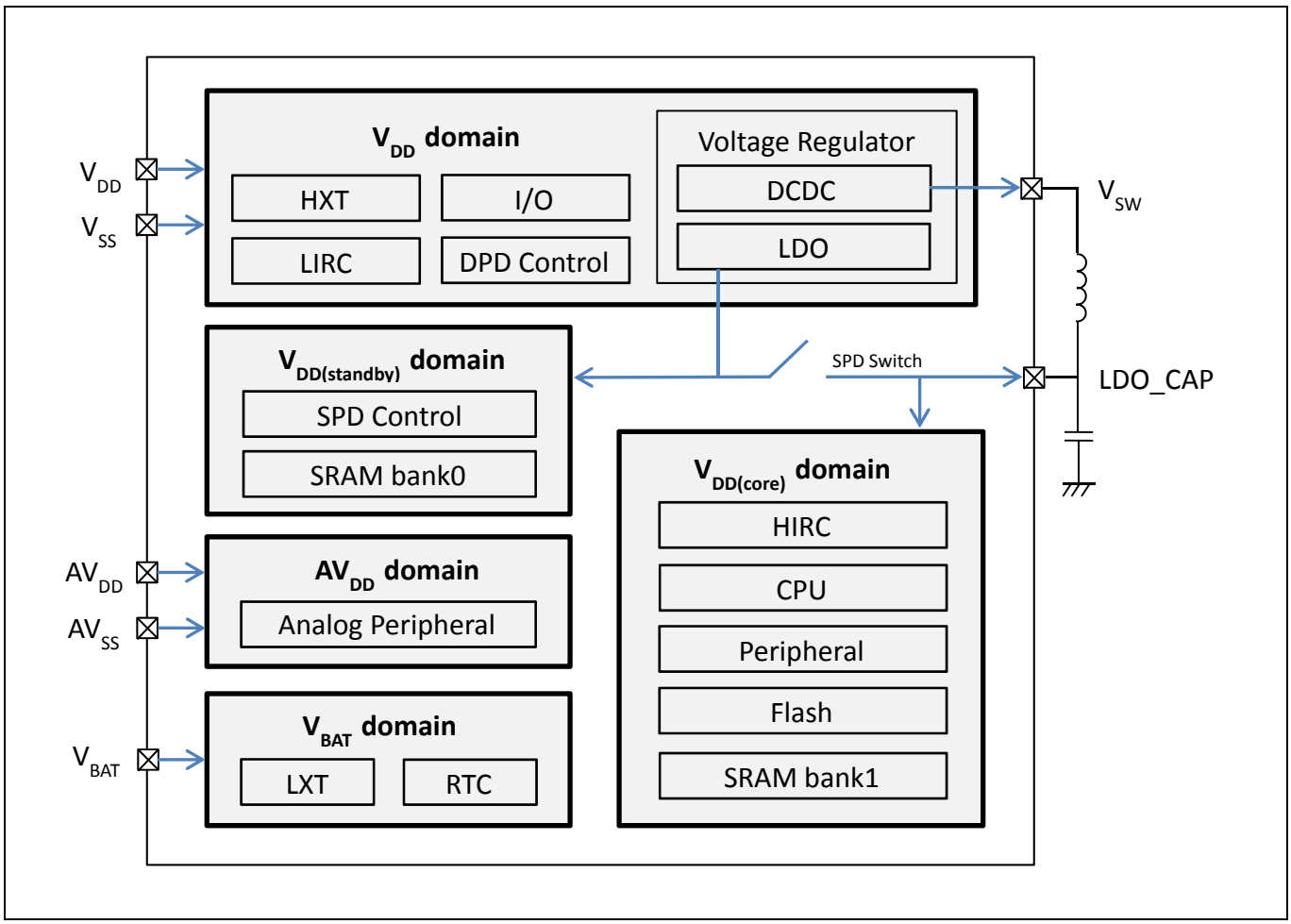

Figure 2-1 M261 Power Distribution

In Standby Power-down mode (SPD), system disables $V_{DD(core)}$ power by SPD switch off and left $V_{DD(standby)}$ domain. In $V_{DD(standby)}$ domain, SPD control logic control wake-up source and system SRAM bank0 can retain data. In Deep Power-down Mode (DPD), the internal voltage regulator is disabled remain DPD control logic to control wake-up source. All system SRAM (including bank0 and bank1) are powered off. All data will be lost after wake-up from Deep Power-down mode (DPD).

Figure 2-2 shows M261 power mode transition. The system is powered on in Normal mode. In Normal mode, you can select different voltage regulator output voltage level (Power Level 0 or Power Level 1) and voltage regulator type (LDO or DCDC). System can enter Idle mode, wait interrupt to wake up and return to Normal mode at any voltage regulator level and voltage regulator type. If system enters Power-down mode, its voltage regulator type must be

in LDO mode at any voltage regulator level (Power Level 0 or Power Level 1). System entering Power-down mode in DCDC mode is not supported. If system enters Standby Power-down mode (SPD) or Deep Power-down mode (DPD), it will reset and return to default voltage regulator level Power Level 0 and voltage regulator LDO mode.

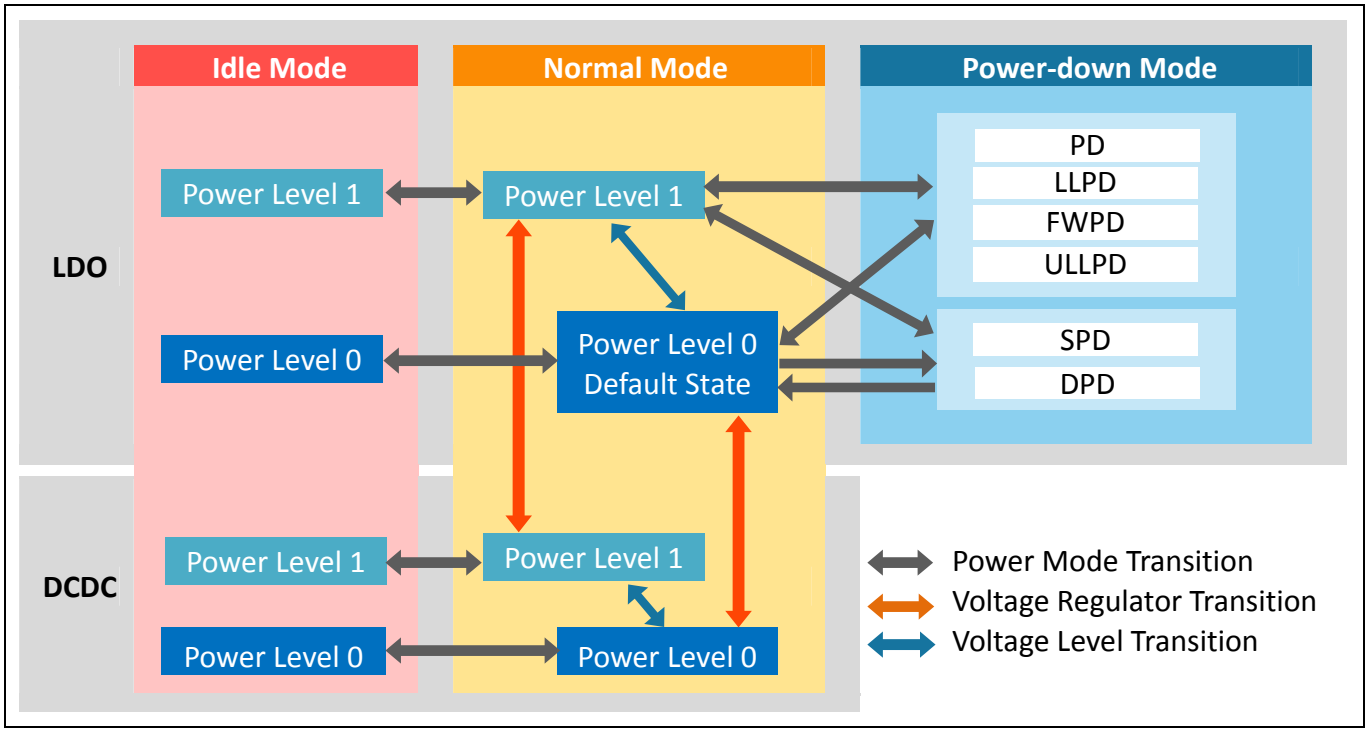

Figure 2-2 M261 Power Mode Transition

2.1 Normal Mode

The system starts up in Normal mode. All clock sources and peripherals can be enabled or disabled by user in register CLK_PWRCRL (System Power-down Control Register), CLK_AHBCLK (AHB Devices Clock Enable Control Register), CLK_APBCLK0 (APB Devices Clock Enable Control Register 0) and CLK_APBCLK1 (APB Devices Clock Enable Control Register 1). You can disable unused clock or peripheral to save power.

Voltage Level

The voltage level can be selected as Power Level 0 and Power Level 1 in Normal mode. The default system voltage level is Power Level 0. In M261, Power Level 0 indicates voltage level is 1.26V and CPU maximum operation frequency is 64MHz. Power level 1 indicates voltage level is 1.2V and CPU maximum operation frequency is 48 MHz.

To set voltage level as Power Level 0, set Power Level Selection, PLSEL(SYS_PLCTL[1:0]) = 0.

```
/* Set power level to Power Level 0 */
```

```
SYS->PLCTL = (SYS->PLCTL&(~SYS_PLCTL_PLSEL_Msk))|SYS_PLCTL_PLSEL_PL0;
```

Or you can call function SYS_SetPowerLevel() to set voltage level as Power Level 0.

```
/* Set power level to Power Level 0 */
SYS_SetPowerLevel(SYS_PLCTL_PLSEL_PL0);
```

To set voltage level as Power Level 1, set Power Level Selection, PLSEL(SYS_PLCTL[1:0]) = 1.

```
/* Set power level to Power Level 1 */
SYS->PLCTL = (SYS->PLCTL&(~SYS_PLCTL_PLSEL_Msk))|SYS_PLCTL_PLSEL_PL1;
```

Or you can call function SYS_SetPowerLevel() to set voltage level as Power Level 1.

```
/* Set power level to Power Level 1 */
SYS_SetPowerLevel(SYS_PLCTL_PLSEL_PL1);
```

Voltage Regulator

The voltage regulator can be selected as LDO or DCDC mode. The default system voltage regulator is LDO. You can select DCDC mode to save power. An inductor component has to be connected in circuit when using DCDC mode. After setting DCDC mode, system will detect inductor connection in 20us and set Main Voltage Regular Type Change Busy Flag, MVRCBUSY (SYS_PLSTS[1]) = 1, to indicate the voltage regulator is switching. If the inductor has been detected, system will switch voltage regulator type to DCDC and clear switch busy flag, MVRCBUSY (SYS_PLSTS[1]) = 0. If the inductor cannot be detected, system will not change current voltage regulator type and set Inductor for DCDC Connect Status, LCONS (SYS_PLSTS[3]) = 1, and Main Voltage Regular Type Change Error, MVRCERR (SYS_PLSTS[2]) = 1. Current voltage regulator type keeps in LDO mode. You can check it by Current Main Voltage Regular Type, CURMVR (SYS_PLSTS[12]).

DCDC mode is only supported in Normal mode and Idle mode. Entering Power-down Mode when voltage regulator type is DCDC mode is not allowed. System will ignore this Power-down request and set Power-down Mode Invalid Transition Flag, PDINVTRF (SYS_PLSTS[4]). Before setting Power-down mode, you have to check if the voltage regulator type is LDO mode and switch back to DCDC mode after wake-up if needed.

To set voltage regulator as LDO, set Main Voltage Regulator Type Selection, MVRS(SYS_PLCTL[4]) = 0.

```
/* Set main voltage regulator type to LDO mode */
SYS->PLCTL = (SYS_PLCTL & (~SYS_PLCTL_MVRS_Msk)) | SYS_PLCTL_MVRS_LDO;
```

Or call function SYS_SetPowerRegulator() to set voltage regulator as LDO.

```
/* Set main voltage regulator type to LDO mode */
SYS_SetPowerRegulator(SYS_PLCTL_MVRS_LDO);
```

To set voltage regulator as DCDC, set Main Voltage Regulator Type Selection MVRS(SYS_PLCTL[4]) = 1, wait Main Voltage Regular Type Change Busy Flag, MVRCBUSY (SYS_PLSTS[1]) is cleared, check voltage regulator type transition is correct or not by Main Voltage Regular Type Change Error, MVRCERR(SYS_PLSTS[2]). If the voltage regulator

type transition is not correct, the voltage regulator type will keep in LDO mode.

```
/* Set main voltage regulator type to DCDC mode */
SYS->PLCTL = (SYS_PLCTL & (~SYS_PLCTL_MVRS_Msk)) | SYS_CVCTL_MVRS_DCDC;

/* Wait main voltage regulator type change busy flag is cleared */
while( SYS->PLSTS & SYS_PLSTS_MVRCBUSY_Msk );

/* Check main voltage regulator type change status */
if( SYS->PLSTS & SYS_PLSTS_MVRCERR_Msk )
{
     printf("Set main voltage regulator type to DCDC error!");

     /* Clear main voltage regulator type change error flag */
     SYS->PLSTS = SYS_PLSTS_MVRCERR_Msk;
}
```

Or call function SYS_SetPowerRegulator() to set voltage regulator as DCDC mode.

```
/* Set main voltage regulator type to DCDC mode */
SYS_SetPowerRegulator(SYS_PLCTL_MVRS_DCDC);
```

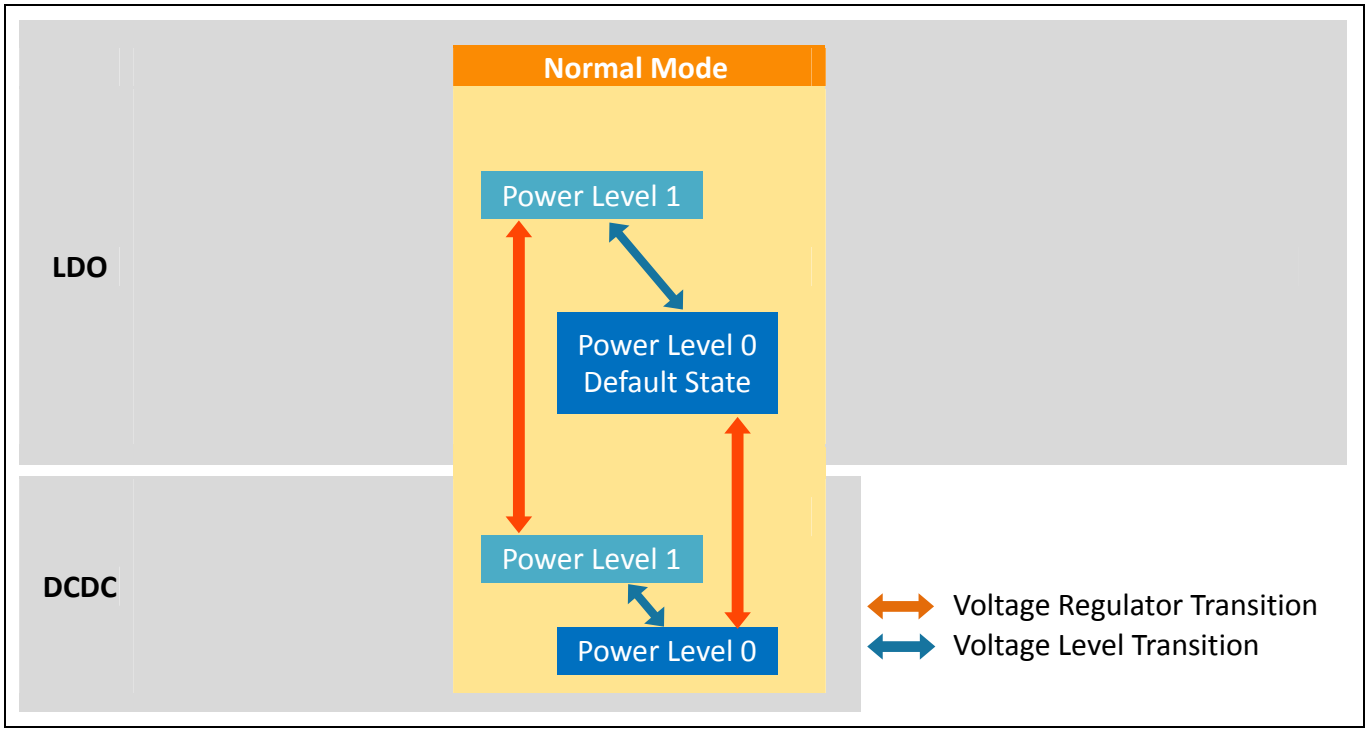

Figure 2-3 Voltage Regulator and Voltage Level Transition in Normal Mode

Figure 2-3 shows any voltage level (Power Level 0 or Power Level 1) and voltage regulator (LDO or DCDC) can switch between each other in Normal mode.

2.2 Idle Mode

If system is waiting for an interrupt only, you can set system in Idle mode. In Idle mode, only CPU clock is disabled, other peripherals work normally. System waits interrupt to wake up, return to Normal mode and keeps working. All system and peripheral interrupts can wake-up system from Idle mode.

錯誤! 找不到參照來源。 shows the transition between Normal mode and Idle mode can operate in any voltage level (Power Level 0 or Power Level 1) and voltage regulator (LDO or DCDC).

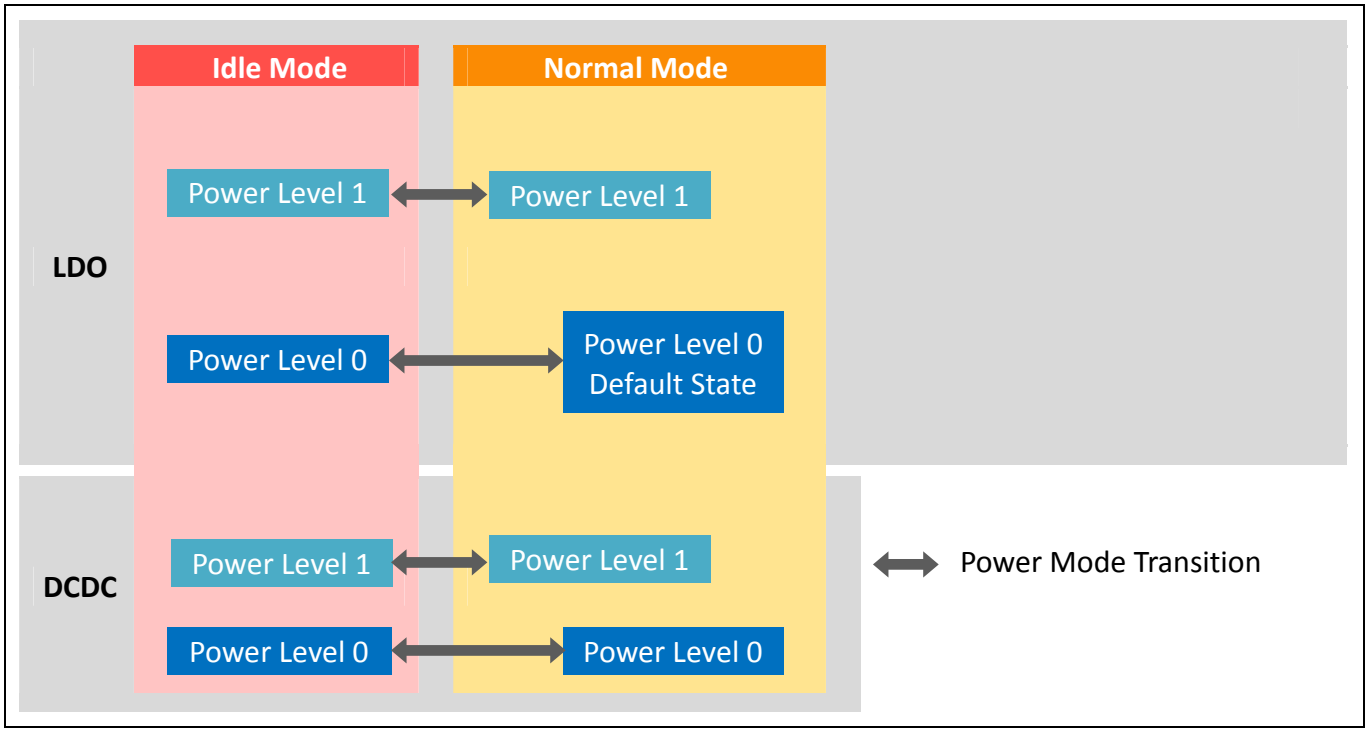

Figure 2-4 Normal Mode and Idle Mode Transition

- **Idle mode configuration:**

Set Processor Low Power Mode Selection to select the system's low power mode as Sleep Mode, SLEEPDEEP (SCR[2]) = 0. Set the System Power-down Enable Bit to set the system to enter Idle mode after executing the WFI instruction, PDEN (CLK_PWRCTL [7]) = 0. Finally, when the WFI instruction is executed, the program will stop and system enters Idle mode.

Set Processor Low Power Mode Selection to select system low power mode is sleep mode, SLEEPDEEP (SCR[2]) = 0. Set System Power-down Enable Bit to set system enters Idle mode after executing WFI instruction. After executing WFI instruction, CPU will stop and system enters Idle mode.

```
/* Set idle mode */
void CLK_Idle(void)
{
    /* Set the processor uses sleep as its low power mode */
    SCB->SCR &= ~SCB_SCR_SLEEPDEEP_Msk;

    /* Set chip in Idle mode because of WFI command */
    CLK->PWRCTL &= ~CLK_PWRCTL_PDEN_Msk;

    /* Chip enter Idle mode after CPU run WFI instruction */
    __WFI();
```

```
}
```

You can also call function CLK_Idle() and system will enter Idle mode.

```
/* Set Idle mode */
CLK_Idle();
```

- **Idle mode wake-up source:**

All system and peripheral interrupts can wake-up system from Idle mode.

2.3 Power-down Mode

You can set system into Power-down mode when system does not need to work for a long time. NuMicro® M261 provides some power moddes with different power consumption level. The less power consumption mode needs more wake-up time and causes more data lost. Figure 2-5 shows power mode transition between Normal mode and Power-down mode. If system enters Power-down mode, its voltage regulator type must be in LDO mode at any voltage regulator level (Power Level 0 or Power Level 1). If system enters Standby Power-down mode (SPD) or Deep Power-down mode (DPD), it will reset and return to default voltage regulator level Power Level 0 and voltage regulator LDO mode.

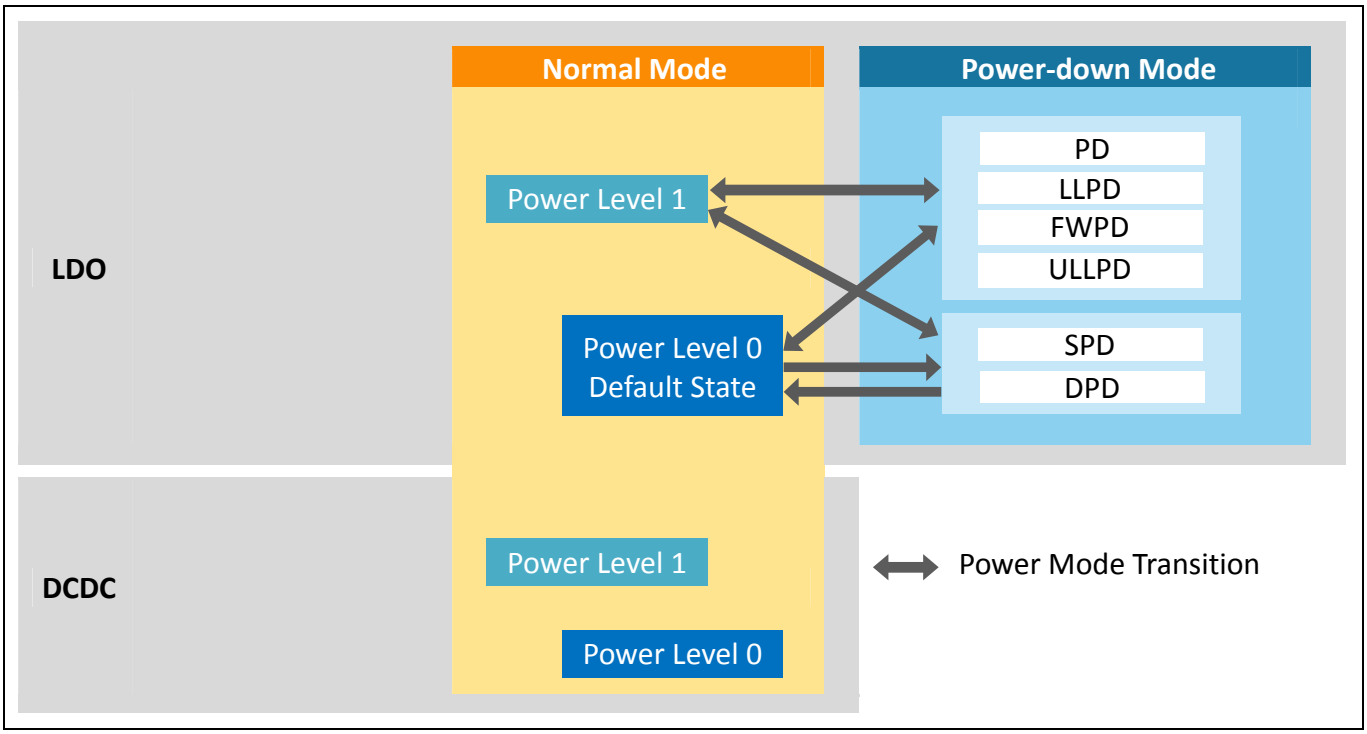

Figure 2-5 Normal Mode and Power-down Mode Transition

Power-down Mode Configuration

Set Power-down Mode Selection, PDMSEL (CLK_PMUCTL[2:0]) to select which Power-down mode. Set Processor Low Power Mode Selection to select system low power mode is Deep sleep mode, SLEEPDEEP (SCR[2]) = 1. Set System Enter Power-down Enable Bit to set system enters Power-down mode after executing WFI instruction. After executing WFI instruction CPU will stop and system enters Power-down mode. Table 2-1 lists different Power-down mode configuration.

Power-down Mode	PDMSEL (CLK_PMUCTL[2:0])	SLEEPDEEP (SCR[2])	PDEN (CLK_PWRCTL [7])	Run WFI Instruction
PD	0	1	1	Yes
LLPD	1			
FWPD	2			
ULLPD	3			
SPD	4			
DPD	6			

Table 2-1 Different Power-down Mode Configuration

In the sample code, set Power-down Mode Selection, PDMSEL (CLK_PMUCTL[2:0]), to select a Power-down mode.

```
/* Select Power-down mode */
void CLK_SetPowerDownMode(uint32_t u32PDMode)
{
    CLK->PMUCTL = (CLK->PMUCTL & (~CLK_PMUCTL_PDMSEL_Msk)) | (u32PDMode);
}
```

Set SLEEPDEEP (SCR[2]) = 1 to set system low power mode is deep sleep mode. Set PDEN (CLK_PWRCTL [7]) = 1, System Enter Power-down Enable Bit. After executing WFI instruction, CPU will stop and enter Power-down mode.

```
/* Set Power-down mode */
void CLK_PowerDown(void)
{
    /* Set the processor uses deep sleep as its low power mode */
    SCB->SCR |= SCB_SCR_SLEEPDEEP_Msk;

    /* Set system Power-down enabled */
```

```
    CLK->PWRCTL |= CLK_PWRCTL_PDEN_Msk;

    /* Chip enter Power-down mode after CPU run WFI instruction */
    __WFI();
}
```

SRAM Power Control in Power-down Mode

SRAM power control in Power-down mode can be configured by register SYS_SRAMPCTL (System SRAM Power Mode Control Register) and SYS_SRAMPPCT (Peripheral SRAM Power Mode Control Register). In system SRAM bank0, it can be configured every 8 Kbytes. In system SRAM bank1, it can be configured every 16 Kbytes. Other peripheral such as FMC, PDMA, USBD and CAN SRAM can be configured. Figure 2-6 and Figure 2-7 list which SRAM power can be controlled and their corresponding control registers.

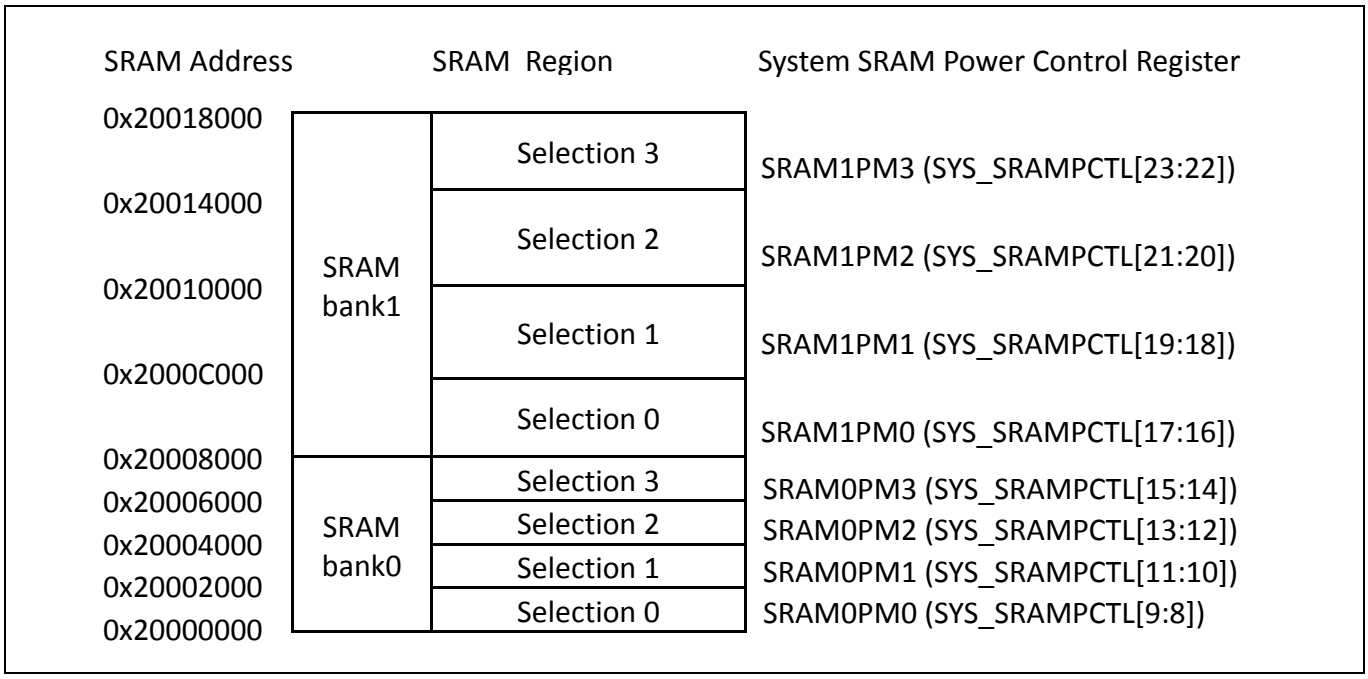

Figure 2-6 System SRAM Power Control in Power-down Mode

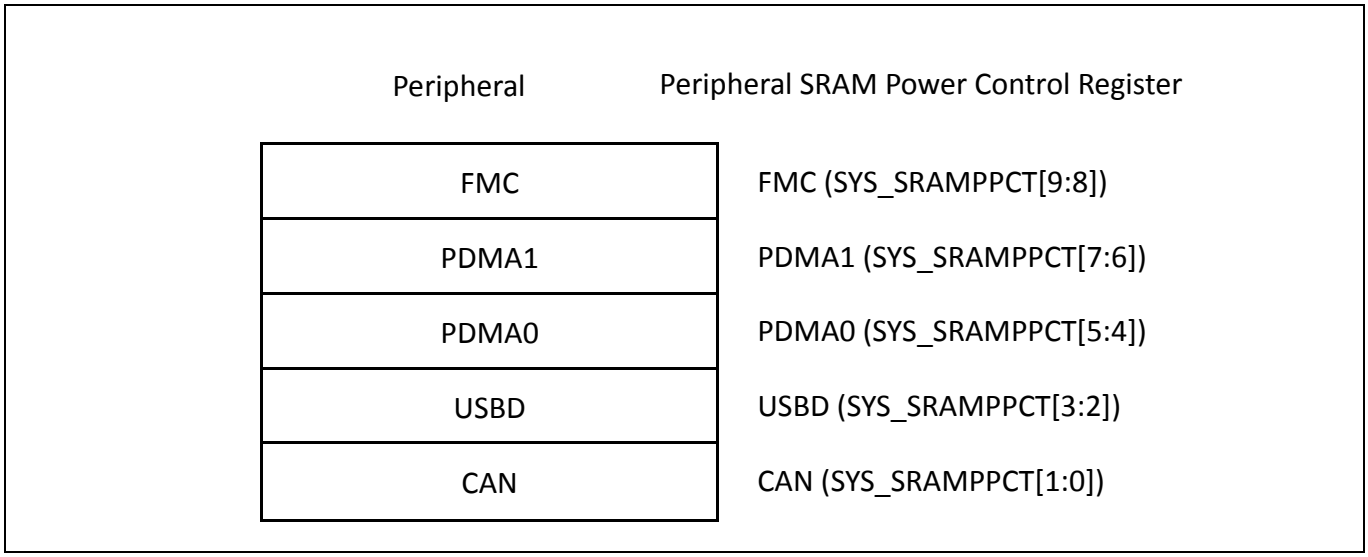

Figure 2-7 Peripheral SRAM Power Control in Power-down Mode

SRAM power mode in Power-down mode can be selected as Normal mode, Retention mode and Power Shut-down mode. This configuration has to be configured before entering Power-down mode and it will be effective when system in Power-down mode. The power consumption from low to high is Power Shut-down mode < Retention mode < Normal mode.

In Retention mode, only SRAM control logic is disabled. In Power Shut-down mode, SRAM control logic and memory power are both disabled. If SRAM power mode is selected as Power Shut-down mode, data will be lost after wake-up. In Standby Power mode (SPD), the system SRAM bank1 and peripheral SRAM are all in Power Shut-down mode. The data will be lost after wake-up form SPD Power-down mode. In Deep Power-down mode (DPD), all SRAM (including system SRAM bank0, system SRAM bank1 and peripheral SRAM) are all in Power Shut-down mode. All SRAM data will be lost after wake-up from Deep Power-down mode (DPD). Table 2-2 lists SRAM power control in different Power-down mode.

SRAM power control in different Power-down mode	**PD LLPD FWPD ULLPD**	**SPD**	**DPD**
System SRAM bank0	Register Control	Register Control	Power Shut-down Mode
System SRAM bank1	Register Control	Power Shut-down Mode	Power Shut-down Mode
Peripheral SRAM	Register Control	Power Shut-down Mode	Power Shut-down Mode

Table 2-2 SRAM Power Control in Different Power-down Mode

For example, to set system SRM bank1 power mode in Power-down mode as Power Shut-down mode, set Bank1 SRAM Power Mode Select 0~3, SRAM1PM3 (SYS_SRAMPCTL [23:22]) = 2, SRAM1PM2 (SYS_SRAMPCTL [21:20]) = 2, SRAM1PM1 (SYS_SRAMPCTL [19:18]) = 2 and SRAM1PM0 (SYS_SRAMPCTL [17:16]) = 2.

```
/* Set SRAM bank 1 is Power Shut-down mode when Power-down */
SYS_SetSSRAMPowerMode(SYS_SRAMPCTL_SRAM1PM0_Msk, SYS_SRAMPCTL_SRAM_POWER_SHUT_DOWN);
SYS_SetSSRAMPowerMode(SYS_SRAMPCTL_SRAM1PM1_Msk, SYS_SRAMPCTL_SRAM_POWER_SHUT_DOWN);
SYS_SetSSRAMPowerMode(SYS_SRAMPCTL_SRAM1PM2_Msk, SYS_SRAMPCTL_SRAM_POWER_SHUT_DOWN);
SYS_SetSSRAMPowerMode(SYS_SRAMPCTL_SRAM1PM3_Msk, SYS_SRAMPCTL_SRAM_POWER_SHUT_DOWN);
```

2.3.1 Normal Power-down Mode (PD)

In Normal Power-down mode (PD), LDO enters low power mode. CPU clock is stopped. All clock source will be disabled expect LXT and LIRC. LXT and LIRC can be controlled by a register. If they are enabled and peripheral clock source are selected as LXT or LIRC, the peripheral can keep working in Power-down mode. System waits wake-up source occurred and returns to Normal mode. After wake-up, system waits for LDO recovery, and clock source is enabled again and stable. The Normal Power-down mode (PD) wake-up sources include EINT, GPIO, UART, USBD, USBH, OTG, CAN, BOD, WDT, SDH, Timer, I²C, USCI, RTC and ACMP. They need to be configured in the peripheral registers.

- **Normal Power-down mode (PD) configuration:**

Call function CLK_SetPowerDownMode() to set Normal Power-down mode (PD) and call function CLK_PowerDown() then system will enter Normal Power-down mode (PD).

```
/* Set Power-down mode (PD) */
CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_PD);

/* Enter Power-down mode */
CLK_PowerDown();
```

- **Normal Power-down mode (PD) wake-up source:**

Normal Power-down mode (PD) wake-up sources are EINT, GPIO, UART, USBD, USBH, OTG, CAN, BOD, WDT, SDH, Timer, I²C, USCI, RTC and ACMP.

- **Normal Power-down mode (PD) data retention:**

Peripheral configuration will be retained. Peripheral SRAM data retention is controlled by the SYS_SRAMPPCT (Peripheral SRAM Power Mode Control Register) register. System SRAM data retention is controlled by SYS_SRAMPCTL (System SRAM Power Mode Control Register) register.

2.3.2 Low Leakage Power-down Mode (LLPD)

In Low Leakage Power-down mode (LLPD), LDO voltage drops down from current working voltage to 0.9V to save power. After system wake-up from Low Leakage Power-down mode (LLPD), besides waiting for LDO recovery and clock source stable, system also needs to wait for LDO voltage rising to the original working voltage (Power Level 0 or Power Level 1). The Low Leakage Power-down mode (LLPD) wake-up sources are the same as Normal Power-down mode (PD).

- **Low Leakage Power-down mode (LLPD) configuration:**

Call function CLK_SetPowerDownMode() to set Low Leakage Power-down mode (LLPD) and call function CLK_PowerDown() then system will enter Low Leakage Power-down mode (LLPD).

```
/* Set Low Leakage Power-down mode (LLPD) */
CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_LLPD);

/* Enter Power-down mode */
CLK_PowerDown();
```

- **Low Leakage Power-down mode (LLPD) wake-up source:**

Low Leakage Power-down mode (LLPD) wake-up sources are EINT, GPIO, UART, USBD, USBH, OTG, CAN, BOD, WDT, SDH, Timer, I²C, USCI, RTC and ACMP.

- **Low Leakage Power-down Mode (LLPD) data retention:**

Peripheral configuration will be retained. Peripheral SRAM data retention is controlled by SYS_SRAMPPCT (Peripheral SRAM Power Mode Control Register) register. System SRAM data retention is controlled by SYS_SRAMPCTL (System SRAM Power Mode Control Register) register.

2.3.3 Fast Wake-up Power-down Mode (FWPD)

In Fast Wake-up Power-down mode (FWPD), LDO is not in lower mode so system does not need to wait LDO recovery after wake-up. The wake-up time is faster than Normal Power-down mode (PD) but it consumes more power, too. Fast Wake-up Power-down mode (FWPD) wake-up sources are the same as Normal Power-down mode (PD).

- **Fast Wake-up Power-down Mode (FWPD) Configuration:**

Call function CLK_SetPowerDownMode() to set Fast Wake-up Power-down mode (FWPD) and call function CLK_PowerDown() then system will enter Fast Wake-up Power-down mode

(FWPD).

```
/* Set Fast Wake-up Power-down mode (FWPD) */
CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_FWPD);

/* Enter Power-down mode */
CLK_PowerDown();
```

- **Fast Wake-up Power-down Mode (FWPD) wake-up source:**

Fast Wake-up Power-down mode (FWPD) wake-up sources are EINT, GPIO, UART, USBD, USBH, OTG, CAN, BOD, WDT, SDH, Timer, I²C, USCI, RTC and ACMP.

- **Fast Wake-up Power-down Mode (FWPD) data retention:**

Peripheral configuration will be retained. Peripheral SRAM data retention is controlled by SYS_SRAMPPCT (Peripheral SRAM Power Mode Control Register) register. System SRAM data retention is controlled by SYS_SRAMPCTL (System SRAM Power Mode Control Register) register.

2.3.4 Ultra Low Leakage Power-down Mode (ULLPD)

In Ultra Low Leakage Power-down mode (ULLPD), LDO voltage will drop down from current working voltage to lower 0.8V to save power. After system wake-up from Ultra Low Leakage Power-down mode (ULLPD), it needs to wait more LDO voltage rising time. Ultra Low Leakage Power-down mode (ULLPD) wake-up sources are the same as Normal Power-down mode (PD).

- **Ultra Low Leakage Power-down Mode (ULLPD) configuration:**

Call function CLK_SetPowerDownMode() to set Ultra Low Leakage Power-down mode (ULLPD) and call function CLK_PowerDown() then system will enter Ultra Low Leakage Power-down mode (ULLPD).

```
/* Set Ultra Low Leakage Power-down mode (ULLPD) */
CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_ULLPD);

/* Enter Power-down mode */
CLK_PowerDown();
```

- **Ultra Low Leakage Power-down Mode (ULLPD) wake-up sources:**

The Ultra Low Leakage Power-down mode (ULLPD) wake-up sources are EINT, GPIO, UART, USBD, USBH, OTG, CAN, BOD, WDT, SDH, Timer, I²C, USCI, RTC and ACMP.

- **Ultra Low Leakage Power-down Mode (ULLPD) data retention:**

Peripheral configuration will be retained. Peripheral SRAM data retention is controlled by SYS_SRAMPPCT (Peripheral SRAM Power Mode Control Register) register. System SRAM data retention is controlled by SYS_SRAMPCTL (System SRAM Power Mode Control Register) register.

2.3.5 Standby Power-down Mode (SPD)

In Standby Power-down mode (SPD), all power supply is disabled expect SPD control logic to control SPD Power-down mode wake-up and system SRAM bank0 to retain data. After wake-up from Standby Power-down mode (SPD), system resets and executes code from the beginning again. All peripheral configurations return to default value. All SRAM data will be lost expect system SRAM bank0 data can be retained.

After wake-up from Standby Power-down mode (SPD), GPIO will keep their states before entering Standby Power-down mode (SPD). They cannot be controlled by GPIO register or peripherals after wake-up. To control GPIO, you have to write 1 to register CLK_IOPDCTL (GPIO Standby Power-down Control Register) to release this GPIO hold state function. If the GPIO hold state function is not released, GPIO cannot be controlled by any GPIO or peripheral registers, including: UART cannot print message and ICE cannot download code. Remember to release GPIO hold state function after system wake-up from SPD Standby Power-down mode (SPD) by writing 1 to CLK_IOPDCTL register. Then you can control GPIO normally through GPIO or peripherals. Standby Power-down mode (SPD) wake-up sources includes GPIO (PA, PB, PC and PD), BOD, LVR, ACMP, Wake-up Timer and RTC.

Set register CLK_IOPDCTL (GPIO Standby Power-down Control Register) to release GPIO hold state function after wake-up from Standby Power-down mode (SPD).

```
/* Release I/O hold status after wake-up from Standby Power-down mode (SPD) */
CLK->IOPDCTL = CLK_IOPDCTL_IOHR_Msk;
```

- **Standby Power-down Mode (SPD) configuration:**

Call function CLK_SetPowerDownMode() to set Standby Power-down mode (SPD) and call function CLK_PowerDown() then system will enter Standby Power-down mode (SPD).

```
/* Set Standby Power-down mode (SPD) */
CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_SPD);

/* Enter Power-down mode */
CLK_PowerDown();
```

- **Standby Power-down Mode (SPD) wake-up sources:**

The Standby Power-down mode (SPD) wake-up sources includes GPIO (PA, PB, PC and PD), BOD, LVR, ACMP, Wake-up Timer and RTC.

- **Standby Power-down Mode (SPD) data retention:**

Peripheral configuration and peripheral SRAM data will be lost. System SRAM bank0 data retention is controlled by user in SYS_SRAMPCTL (System SRAM Power Mode Control Register) register. System SRAM bank1 data will be lost.

2.3.6 Deep Power-down Mode (DPD)

In Deep Power-down mode (DPD), CPU will be stopped. All power supply will be disabled expect DPD control logic to control DPD Power-down mode wake-up source. After wake-up from Deep Power-down mode (DPD), system resets and executes code from the beginning again, all peripheral configuration and SRAM data will be lost. The Deep Power-down mode (DPD) wake-up sources include Wake-up Timer, RTC and Wake-up Pin. Deep Power-down mode (DPD) disables more logic to save power and needs more wake-up time to re-start system.

- **Deep Power-down Mode (DPD) configuration:**

Call function CLK_SetPowerDownMode() to set Deep Power-down Mode (DPD) and call function CLK_PowerDown() then system will enter Deep Power-down Mode (DPD).

```
/* Set Deep Power-down mode (DPD) */
CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_DPD);

/* Enter Power-down mode */
CLK_PowerDown();
```

- **Deep Power-down Mode (DPD) wake-up source:**

Deep Power-down mode (DPD) wake-up sources includes Wake-up Timer, RTC and Wake-up Pin.

- **Deep Power-down Mode (DPD) data retention:**

Peripheral configuration and peripheral SRAM data will be lost. System SRAM (both bank0 and bank1) data will be lost.

3 Sample Code

This section shows configuration and wake-up method of different power modes.

- Normal mode (section 3.1)
  Set different voltage regulator and voltage level.
- Idle mode (section 3.2)
  System enters Idle mode and wakes up by WDT interrupt.
- Power-down mode (section 3.3)
  System enters different Power-down mode and wakes up by RTC 1 second tick interrupt.
  - Normal Power-down mode (PD) (section 3.3.1)
  - Low Leakage Power-down mode (LLPD) (section 3.3.2)
  - Fast Wake-up Power-down mode (FWPD) (section 3.3.3)
  - Ultra Low Leakage Power-down mode (ULLPD) (section 3.3.4)
  - Standby Power-down mode (SPD) (section 3.3.5)
  - Deep Power-down mode (DPD) (section 3.3.6)
- SPD Power-down mode wake-up and return (section 3.4)
  Using system SRAM bank0 data retention function to execute the code continually after wake-up from Standby Power-down mode.

3.1 Normal Mode

The system is powered on in Normal mode. The default voltage regulator type is LDO mode. To save power, you can select lower power consumption DCDC mode. An inductor component should be connected in circuit to use DCDC mode. If there is no inductor connected, system cannot switch to DCDC mode and set Main Voltage Regular Type Change Error, MVRCERR (SYS_PLSTS[2]). DCDC mode is not supported in Power-down mode. Before entering Power-down mode, you should make sure that system is in LDO mode and switch back to DCDC mode after wake-up if needed.

```
void main(void)
{
    /* Init System, peripheral clock and multi-function I/O */
    SYS_Init();

    /* Init UART0 for printf */
    UART0_Init();

    /* Set main voltage regulator type to DCDC mode */
    printf("Set main voltage regulator type to DCDC mode ");
```

```
    if( SYS_SetPowerRegulator(SYS_PLCTL_MVRS_DCDC) == 0 )
        printf("[no inductor connect]\n");

    /* Set main voltage regulator type to LDO mode */
    printf("Set main voltage regulator type to LDO mode ");
    SYS_SetPowerRegulator(SYS_PLCTL_MVRS_LDO);

    /* Unlock protected registers before entering Power-down mode */
    SYS_UnlockReg();

    /* Enter Power-down mode */
    CLK_PowerDown();

    /* End of sample code */
    while(1);
}
```

The default system voltage level is Power Level 0 (1.26V) and the maximum CPU operation frequency is 64 MHz. To save power, you can set Power Level 1 (1.2V) and CPU maximum operation frequency is lower 48MHz. To set voltage level from higher voltage to lower voltage, for example, switch from 1.26V to 1,2V, you need to make sure that the CPU operation frequency is lower or equal to 48MHz, and then switch to 1.2V.

```
void main(void)
{
    /* Set power level to 1.26V */
    SYS_SetPowerLevel(SYS_PLCTL_PLSEL_PL0);

    /* Set core clock as 64MHz from PLL */
    CLK_SetCoreClock(64000000);

    /* Set core clock as 48MHz from PLL */
    CLK_SetCoreClock(48000000);

    /* Set power level to 1.2V */
    SYS_SetPowerLevel(SYS_PLCTL_PLSEL_PL1);

    while(1);
}
```

3.2 Idle Mode

If CPU is waiting for an interrupt only, you can set system in Idle mode. It is the fastest power mode can return to Normal mode to keep working. The following sample code shows how to wake up system from Idle mode by WDT time-out interrupt. The WDT time-out interrupt configuration is: enable WDT clock, select WDT clock source as LIRC, enable WDT interrupt and set time-out interval is 16384 LIRC clocks.

```
/* WDT time-out wake-up source setting */
void WDT_Init(void)
{
    /* Enable WDT module clock */
    CLK_EnableModuleClock(WDT_MODULE);

    /* Select WDT module clock source as LIRC */
    CLK_SetModuleClock(WDT_MODULE, CLK_CLKSEL1_WDTSEL_LIRC, 0);

    /* Enable WDT NVIC */
    NVIC_EnableIRQ(WDT_IRQn);

    /* Configure WDT settings and start WDT counting */
    WDT_Open(WDT_TIMEOUT_2POW14, NULL, FALSE, TRUE);

    /* Enable WDT interrupt function */
    WDT_EnableInt();
}
```

The WDT time-out interrupt occurs every 16384 LIRC clocks. Clear WDT time-out interrupt flag in WDT interrupt handler. The WDT time-out interrupt will wake up system if system is in Idle mode.

```
volatile uint8_t g_u8IsINTEvent = 0;

/* WDT IRQ Handler */
void WDT_IRQHandler(void)
{
    /* Check if WDT time-out interrupt occurred */
    if(WDT_GET_TIMEOUT_INT_FLAG())
    {
        /* Clear WDT time-out interrupt flag */
        WDT_CLEAR_TIMEOUT_INT_FLAG();
    }

    g_u8IsINTEvent = 1;
```

```
}
```

After WDT configuration call CLK_Idle() to enter Idle mode and wait system wake-up by WDT time-out interrupt.

```
void main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* WDT time-out wake-up source setting */
    WDT_Init();

    /* Enter Idle mode */
    CLK_Idle();

    /* Check if WDT time-out interrupt and wake-up occurred or not */
    while(g_u8IsINTEvent == 0);

    /* End of sample code */
    while(1);
}
```

3.3 Power-down Mode

The following shows wake-up from different Power-down modes by RTC tick interrupt. To configure RTC tick before entering Power-down mode: enable RTC clock, set RTC clock source as LXT and set RTC tick interval as 1 second.

```
/* RTC tick wake-up source setting */
void RTC_Init(void)
{
    /* Set PF multi-function pins for X32_OUT(PF.4) and X32_IN(PF.5) */
    SYS->GPF_MFPL = (SYS->GPF_MFPL & (~SYS_GPF_MFPL_PF4MFP_Msk)) | X32_OUT_PF4;
    SYS->GPF_MFPL = (SYS->GPF_MFPL & (~SYS_GPF_MFPL_PF5MFP_Msk)) | X32_IN_PF5;

    /* Enable LXT clock and wait for LXT clock ready*/
    CLK_EnableXtalRC(CLK_PWRCTL_LXTEN_Msk);
    CLK_WaitClockReady(CLK_STATUS_LXTSTB_Msk);

    /* Enable RTC peripheral clock */
    CLK_EnableModuleClock(RTC_MODULE);
```

```
    /* Wait RTC access enable */
    RTC_WaitAccessEnable();

    /* Clear RTC tick interrupt flag */
    RTC_CLEAR_TICK_INT_FLAG(RTC);

    /* Enable RTC NVIC */
    NVIC_EnableIRQ(RTC_IRQn);

    /* Enable RTC tick interrupt and wake-up function will be enabled also */
    RTC_EnableInt(RTC_INTEN_TICKIEN_Msk);
    RTC_SetTickPeriod(RTC_TICK_1_SEC);
}
```

After RTC tick interrupt configuration, system enters RTC tick interrupt every 1 second. Clear RTC tick interrupt in RTC interrupt handler. RTC tick interrupt will wake system up if system is in Power-down mode.

```
volatile uint32_t g_u32RTCTickINT = 0;

/* RTC IRQ Handler */
void RTC_IRQHandler(void)
{
    /* Check if RTC tick interrupt occurred */
    if(RTC_GET_TICK_INT_FLAG(RTC) == 1)
    {
        /* Clear RTC tick interrupt flag */
        RTC_CLEAR_TICK_INT_FLAG(RTC);
    }

    g_u32RTCTickINT++;
}
```

3.3.1 Normal Power-down Mode (PD)

After RTC tick configuration, call function CLK_SetPowerDownMode() to set Normal Power-down mode (PD) and call CLK_PowerDown() then system will enter Normal Power-down mode (PD). System wakes up from Normal Power-down mode (PD) by RTC tick interrupt and returns to Normal mode.

```
void main(void)
{
    /* Unlock protected registers */
```

```
    SYS_UnlockReg();

    /* RTC tick wake-up source setting */
    RTC_Init();

    /* Set Normal Power-down mode (PD) */
    CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_PD);

    /* Enter Power-down mode */
    CLK_PowerDown();

    /* Wait RTC tick interrupt and wake-up */
    while( g_u32RTCTickINT == 0);

    /* End of sample code */
    while(1);
}
```

3.3.2 Low Leakage Power-down Mode (LLPD)

After RTC tick configuration, call function CLK_SetPowerDownMode() to set Low Leakage Power-down mode (LLPD) and call CLK_PowerDown() then system will enter Low Leakage Power-down mode (LLPD). System wakes up from Low Leakage Power-down mode (LLPD) by RTC tick interrupt and returns to Normal mode.

```
void main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* RTC tick wake-up source setting */
    RTC_Init();

    /* Set Low Leakage Power-down mode (LLPD) */
    CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_LLPD);

    /* Enter Power-down mode */
    CLK_PowerDown();

    /* Wait RTC tick interrupt and wake-up */
    while( g_u32RTCTickINT == 0);
```

```
    /* End of sample code */
    while(1);
}
```

3.3.3 Fast Wake-up Power-down Mode (FWPD)

After RTC tick configuration, call function CLK_SetPowerDownMode() to set Fast Wake-up Power-down mode (FWPD) and call CLK_PowerDown() then system will enter Fast Wake-up Power-down mode (FWPD). System wakes up from Fast Wake-up Power-down mode (FWPD) by RTC tick interrupt and returns to Normal mode.

```
void main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* RTC tick wake-up source setting */
    RTC_Init();

    /* Set Fast Wake-up Power-down mode (FWPD) */
    CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_FWPD);

    /* Enter Power-down mode */
    CLK_PowerDown();

    /* Wait RTC tick interrupt and wake-up */
    while( g_u32RTCTickINT == 0);

    /* End of sample code */
    while(1);
}
```

3.3.4 Ultra Low Leakage Power-down Mode (ULLPD)

After RTC tick configuration, call function CLK_SetPowerDownMode() to set Ultra Low Leakage Power-down mode (ULLPD) and call CLK_PowerDown() then system will enter Ultra Low Leakage Power-down mode (ULLPD). System wakes up from Ultra Low Leakage Power-down mode (ULLPD) by RTC tick interrupt and returns to Normal mode.

```
void main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();
```

```
    /* RTC tick wake-up source setting */
    RTC_Init();

    /* Set Ultra Low Leakage Power-down mode (ULLPD) */
    CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_ULLPD);

    /* Enter Power-down mode */
    CLK_PowerDown();

    /* Wait RTC tick interrupt and wake-up */
    while( g_u32RTCTickINT == 0);

    /* End of sample code */
    while(1);
}
```

3.3.5 Standby Wake-up Power-down Mode (SPD)

After RTC tick configuration, call function CLK_SetPowerDownMode() to set Standby Power-down mode (SPD) and call CLK_PowerDown() then system will enter Standby Power-down mode (SPD). System wakes up from Standby Power-down mode (SPD) by RTC tick interrupt and, resets and returns to Normal mode. After system wakes up, you have to write 1 to register CLK_IOPDCTL (GPIO Standby Power-down Control Register) to release GPIO hold state function and can control GPIO normally.

```
void main(void)
{
    /* if wake-up from Standby Power-down mode (SPD) or Deep Power-down mode (DPD) */
    If(CLK_GetPMUWKSrc())
    {
        /* Release I/O hold status after wake-up from Standby Power-down mode (SPD) */
        CLK->IOPDCTL = 1;

        /* Clear Power Manager Status register */
        CLK->PMUSTS = CLK_PMUSTS_CLRWK_Msk;
    }

    /* Unlock protected registers */
    SYS_UnlockReg();

    /* RTC tick wake-up source setting */
```

```
    RTC_Init();

    /* Set Standby Power-down mode (SPD) */
    CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_SPD);

    /* Enter Power-down mode */
    CLK_PowerDown();

    /* Wait RTC tick interrupt and wake-up */
    while( g_u32RTCTickINT == 0);

    /* End of sample code */
    while(1);
}
```

3.3.6 Deep Wake-up Power-down Mode (DPD)

After RTC tick configuration, call function CLK_SetPowerDownMode() to set Deep Power-down mode (DPD) and call CLK_PowerDown() then system will enter Deep Power-down mode (DPD). System wakes up from Deep Power-down mode (DPD) by RTC tick interrupt and, resets and returns to Normal mode.

```
void main(void)
{
    /* Unlock protected registers */
    SYS_UnlockReg();

    /* RTC tick wake-up source setting */
    RTC_Init();

    /* Set Deep Power-down mode (DPD) */
    CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_DPD);

    /* Enter Power-down mode */
    CLK_PowerDown();

    /* Wait RTC tick interrupt and wake-up */
    while( g_u32RTCTickINT == 0);

    /* End of sample code */
    while(1);
}
```

3.4 SPD Power-down Mode Wake-up and Return

This section shows how to continue executing code after wake-up from Standby Power-down mode (SPD) without re-executing code from the beginning by using system SRAM bank0 data retention function. Figure 2-1 shows the flow chart of sample code.

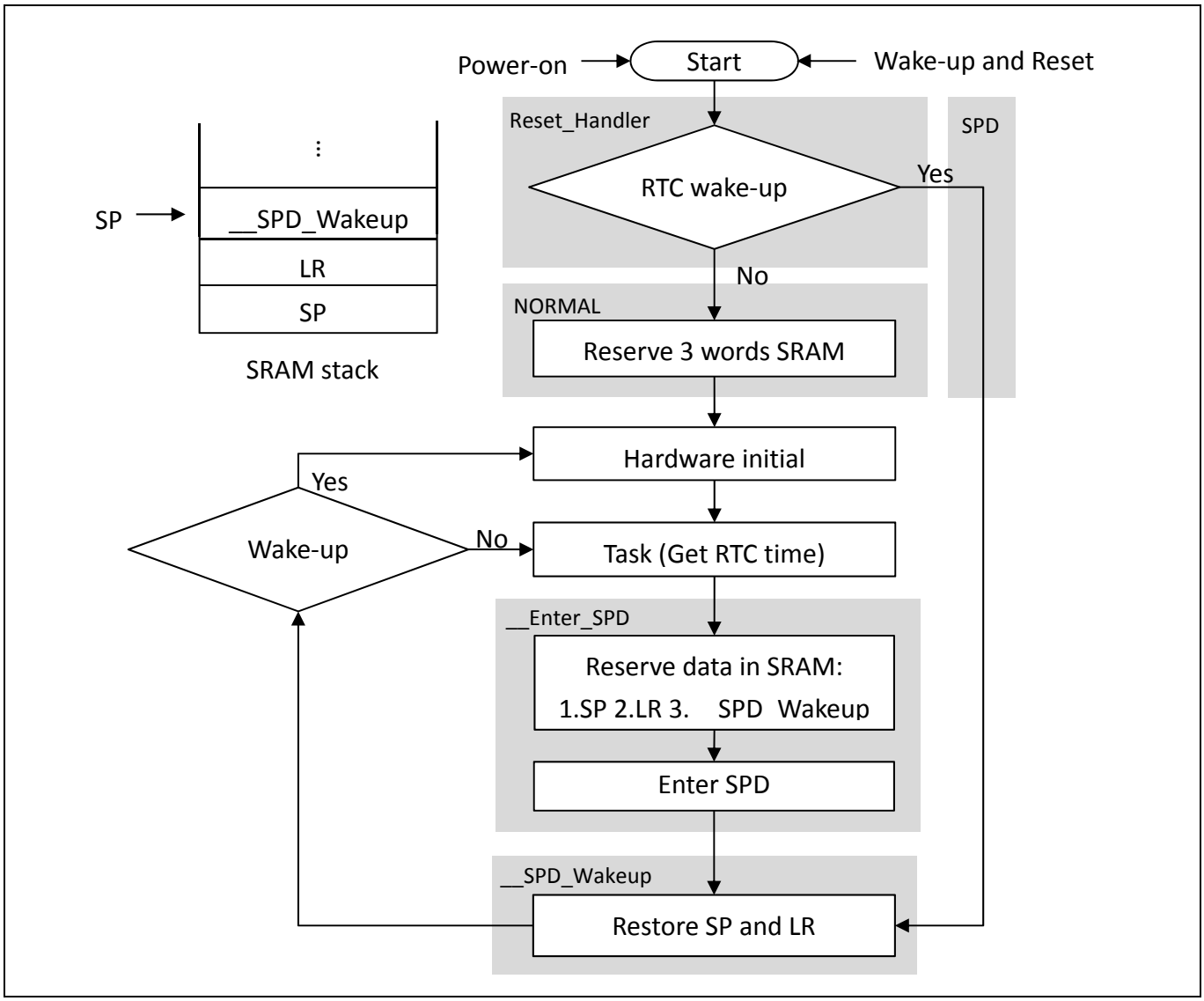

Figure 3-1 SPD Power-down Mode Wake-up and Return Flow Chart

At the beginning of the program(Reset_Handler), read RTC wake-up from SPD flag RTCWK(CLK_PMUSTS[2]) to check if system has waken-up from Standby Power-down mode (SPD). If RTCWK is 0, system does not wake up from Standby Power-down mode (SPD). It is a normal power-on process (NORMAL). Reserve 3 words SRAM stack space, configure system initialization and then continue the program.

mystartup_M261.s

```
Reset_Handler   PROC
                EXPORT  Reset_Handler             [WEAK]
                IMPORT  SystemInit
                IMPORT  __main

                LDR     r0, =0x40000294 ; Check RTC wake-up from SPD flag
                LDR     r0, [r0, #0]
                MOVS    r1, #4
                ANDS    r0, r0, r1
                BEQ     NORMAL
SPD                                     ; Wake-up from SPD
                SUB     sp, sp, #12
                POP     {PC}            ; Execute __SPD_Wakeup
NORMAL                                  ; Normal Power-on process
                MOV     r0, #0          ; Reserved 3 words stack space to retain data
                PUSH    {r0}
                PUSH    {r0}
                PUSH    {r0}

                LDR     R0, =SystemInit
                BLX     R0
                LDR     R0, =__main
                BX      R0
                ENDP
```

Before entering Standby Power-down mode (SPD), you can save current LR and R0 to R7 register values in SRAM stack and then call __Enter_SPD function. The __Enter_SPD function saves SP, LR register value and __SPD_Wakeup function address in SRAM stack, which is reserved at the beginning of program. __SPD_Wakeup is the function that needs to be executed after wake-up. After saving the data, you can run WFI instruction and system will enter Standby Power-down mode (SPD).

main.c

```
void PowerDownFunction(void)
{

    /* Select SPD Power-down mode */
    CLK_SetPowerDownMode(CLK_PMUCTL_PDMSEL_SPD);
```

```
    /* Set the processor uses deep sleep as its low power mode */
    SCB->SCR |= SCB_SCR_SLEEPDEEP_Msk;

    /* Set system Power-down enabled */
    CLK->PWRCTL |= CLK_PWRCTL_PDEN_Msk;

    /* Reserve R0-R7, LR and enter Power-down mode */
    __set_PRIMASK(1);
    __ASM volatile("push {r0-r7} \n");
    __ASM volatile("push {lr} \n");
    __Enter_SPD();

    /* Restore R0-R7 and LR */
    __ASM volatile("pop {r0} \n");
    __ASM volatile("mov lr, r0 \n");
    __ASM volatile("pop {r0-r7} \n");
    __set_PRIMASK(0);

    /* Initialization after wake-up from SPD */
    if(CLK->PMUSTS&CLK_PMUSTS_RTCWK_Msk)
    {
        SYS_UnlockReg();       /* Unlock protected registers */
        SYS_Init();            /* Init System, peripheral clock and multi-function I/O */
        UART0_Init();          /* Init UART0 for printf */
        PA10 = PA10;           /* LED toggle in RTC interrupt */
        GPIO_SetMode(PA, BIT10, GPIO_MODE_OUTPUT);
        CLK->IOPDCTL = 1;      /* Release I/O hold status */
        RTC_Init();            /* Init RTC */
    }

}
```

mystartup_M261.s

```
__Enter_SPD     PROC                        ; Enter Power-down
                EXPORT      __Enter_SPD     ; Save SP, LR and __SPD_Wakeup
                LDR     r0, =__SPD_Wakeup
                MOV     r1, lr
                MOV     r2, sp
                MOV     r3, #0
```

```
                LDR     r3, [r3]
                MOV     sp, r3
                PUSH    {r0-r2}
                WFI
                POP     {PC}            ; Execute __SPD_Wakeup
__SPD_Wakeup                            ; Restore SP and LR
                POP     {r1,r2}
                MOV     sp, r2
                BX      r1

                ENDP
```

System will reset after wake-up from Standby Power-down mode (SPD). After program is re-executed, read RTC wake-up from SPD flag RTCWK (CLK_PMUSTS[2]) again. This flag is set to 1. It indicates that system is waken-up form Standby Power-down mode (SPD). After wake-up, it needs to execute __SPD_Wakeup function. The __SPD_Wakeup function will get the SP and LR register value from SRAM stack, restore SRAM stack address by SP, set PC register value to LR address. The program will execute code to LR address, get actual LR address and previous R0 to R7 register values from current stack. Now the program returns to the state after entering Power-down mode. The SRAM data are retained but peripheral register configuration are reset to default value. You just need to configure peripheral register setting again and then the program can be executed continuously without re-executing code.

4 Conclusion

The NuMicro® M261 series has different power modes for users to design their application both high efficiency and low power. The following lists the cmparsion of these power modes including entry configuration, wake-up sources, power concumption and wake-up time.

- Normal mode
- Idle mode
- Power-down mode
  - Normal Power-down mode (PD)
  - Low Leakage Power-down mode (LLPD)
  - Fast Wake-up Power-down mode (FWPD)
  - Ultra Low Leakage Power-down mode (ULLPD)
  - Standby Power-down mode (SPD)
  - Deep Power-down mode (DPD)

The power consumption from the lowest to the highest is: **DPD < SPD < ULLPD < LLPD < PD < FWPD < Idle mode < Normal mode**.

The wake-up time from the shortest to the longest is: **Idle mode < FWPD < PD < LLPD < ULLPD < DPD < SPD**. (Normal mode does not have wake-up time).

Note: For details of power consumption and wake-up time, please refer to the M261 Datasheet.

Table 4-1 shows the power mode configuration comparison:

Power Mode	Power Mode Configuration
Normal mode	System default state
Idle mode	1. SLEEPDEEP (SCR[2]) = 0. 2. Run WFI instruction.
Power-down mode (PD)	1. PDMSEL (CLK_PMUCTL[2:0]) = 0. 2. SLEEPDEEP (SCR[2]) = 1. 3. PDEN (CLK_PWRCTL [7]) = 1. 4. Run WFI instruction.
Low Leakage Power-down mode (LLPD)	1. PDMSEL (CLK_PMUCTL[2:0]) = 1. 2. SLEEPDEEP (SCR[2]) = 1.

	3. PDEN (CLK_PWRCTL [7]) = 1. 4. Run WFI instruction.
Fast Wake-up Power-down mode (FWPD)	1. PDMSEL (CLK_PMUCTL[2:0]) = 2. 2. SLEEPDEEP (SCR[2]) = 1. 3. PDEN (CLK_PWRCTL [7]) = 1. 4. Run WFI instruction.
Ultra Low Leakage Power-down mode (ULLPD)	1. PDMSEL (CLK_PMUCTL[2:0]) = 3. 2. SLEEPDEEP (SCR[2]) = 1. 3. PDEN (CLK_PWRCTL [7]) = 1. 4. Run WFI instruction.
Standby Power-down mode (SPD)	1. PDMSEL (CLK_PMUCTL[2:0]) = 4. 2. SLEEPDEEP (SCR[2]) = 1. 3. PDEN (CLK_PWRCTL [7]) = 1. 4. Run WFI instruction.
Deep Power-down mode (DPD)	1. PDMSEL (CLK_PMUCTL[2:0]) = 6. 2. SLEEPDEEP (SCR[2]) = 1. 3. PDEN (CLK_PWRCTL [7]) = 1. 4. Run WFI instruction.

Table 4-1 Different Power Mode Configuration Comparison

Table 4-2 shows the power mode wake-up source comparison:

Power Mode	**Wake-up Source**
Normal mode	System default state no wake-up source.
Idle mode	Any system and peripheral interrupt.
Power-down mode (PD)	Wake-up source: EINT, GPIO, UART, USBD, USBH, OTG, CAN, BOD, WDT, SDH, Timer I²C, USCI, RTC and ACMP.
Low Leakage Power-down mode (LLPD)	
Fast Wake-up Power-down mode (FWPD)	
Ultra Low Leakage Power-down mode (ULLPD)	

Standby Power-down mode (SPD)	SPD wake-up source: GPIO (PA, PB, PC and PD), BOD, LVR, ACMP, Wake-up Timer and RTC.
Deep Power-down mode (DPD)	DPD wake-up source: Wake-up Timer, RTC and Wake-up Pin.

Table 4-2 Different Power Mode Wake-up Source Comparison

Table 4-3 shows the power mode data retention comparison:

Power mode	**Data Retention**
Normal mode	1.Peripheral configuration is retained. 2.Peripheral SRAM data is retained. 3.System SRAM data is retained.
Idle mode	
Power-down mode (PD)	1.Peripheral configuration is retained. 2.Peripheral SRAM data retention control by register SYS_SRAMPPCT. 3.System SRAM data retention control by register SYS_SRAMPCTL.
Low Leakage Power-down mode (LLPD)	
Fast Wake-up Power-down mode (FWPD)	
Ultra Low Leakage Power-down mode (ULLPD)	
Standby Power-down mode (SPD)	1.Peripheral configuration is lost. 2.Peripheral SRAM data is lost. 3.System SRAM bank0 data retention control by register SYS_SRAMPCTL. 4.System SRAM bank1 data is lost.
Deep Power-down mode (DPD)	1.Peripheral configuration is lost. 2.Peripheral SRAM data is lost. 3.System SRAM data is lost.

Table 4-3 Power Mode Data Retention Comparison

Revision History

Date	Revision	Description
2019.04.11	1.00	1. Initially issued.

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.