QSPI与I2S使用PDMA传输

NuMicro® 32 位系列微控制器范例代码介绍

文件信息

代码简述	QSPI 与 I2S 使用 PDMA 传输数据范例代码
BSP 版本	M480 BSP CMSIS V3.03.001
开发平台	NuMaker-PFM-M487 Ver3.0

The information described in this document is the exclusive intellectual property of Nuvoton Technology Corporation and shall not be reproduced without permission from Nuvoton.

Nuvoton is providing this document only for reference purposes of NuMicro microcontroller based system design. Nuvoton assumes no responsibility for errors or omissions.

All data and specifications are subject to change without notice.

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

www.nuvoton.com

1 功能介绍

1.1 简介

QSPI与I2S使用PDMA传输数据，QSPI0示范如何使用Basic模式传输数据，I2S0示范如何使用Scatter-Gather模式传输

1.2 原理

将QSPI0的TX与RX对接，I2S0的TX和RX对接进行测试。程序里设置了64笔32bit的数据数据透过PDMA Basic 模式将数据放到QSPI0的TX，QSPI0把数据从TX传输给RX，RX藉由PDMA Basic 模式的方式传到内存的MasterRxBuffer中，I2S0的TX再通过PDMA Scatter-Gather 模式从内存MasterRxBuffer中获取数据，传给I2S0的RX，然后经过PDMA Scatter-Gather 模式传给内存SlaveRxBuffer，最后将SlaveRxBuffer中的64笔数据透过UART0数据打印出来，并验证数据互传是否成功。(由于I2S在开始传输数据前会先打一笔无效数据，故SlaveRxBuffer数据数组需宣告为65笔。)

1.3 执行结果

执行结果，以及打印数据。
QSPI传输数据如下

UART #1

```
+------------------------------------------------------------+
|                  QSPI + I2S + PDMA Sample Code             |
+------------------------------------------------------------+


QSPI0 to I2S0 Loop test with PDMA

QSPI0 TX                QSPI0 RX
0x55000001              0x55000001
0x55000002              0x55000002
0x55000003              0x55000003
0x55000004              0x55000004
0x55000005              0x55000005
0x55000006              0x55000006
0x55000007              0x55000007
0x55000008              0x55000008
0x55000009              0x55000009
0x5500000A              0x5500000A
0x5500000B              0x5500000B
0x5500000C              0x5500000C
0x5500000D              0x5500000D
0x5500000E              0x5500000E
0x5500000F              0x5500000F
0x55000010              0x55000010
0x55000011              0x55000011
0x55000012              0x55000012
0x55000013              0x55000013
0x55000014              0x55000014
0x55000015              0x55000015
0x55000016              0x55000016
0x55000017              0x55000017
0x55000018              0x55000018
0x55000019              0x55000019
0x5500001A              0x5500001A
0x5500001B              0x5500001B
0x5500001C              0x5500001C
0x5500001D              0x5500001D
0x5500001E              0x5500001E
0x5500001F              0x5500001F
0x55000020              0x55000020
0x55000021              0x55000021
0x55000022              0x55000022
0x55000023              0x55000023
0x55000024              0x55000024
0x55000025              0x55000025
0x55000026              0x55000026
0x55000027              0x55000027
0x55000028              0x55000028
0x55000029              0x55000029
0x5500002A              0x5500002A
0x5500002B              0x5500002B
0x5500002C              0x5500002C
0x5500002D              0x5500002D
0x5500002E              0x5500002E
0x5500002F              0x5500002F
0x55000030              0x55000030
0x55000031              0x55000031
0x55000032              0x55000032
0x55000033              0x55000033
0x55000034              0x55000034
0x55000035              0x55000035
0x55000036              0x55000036
0x55000037              0x55000037
```

```
0x55000038              0x55000038
0x55000039              0x55000039
0x5500003A              0x5500003A
0x5500003B              0x5500003B
0x5500003C              0x5500003C
0x5500003D              0x5500003D
0x5500003E              0x5500003E
0x5500003F              0x5500003F
0x55000040              0x55000040


Exit QSPI .
```

I2S 传输数据如下

UART #1

I2S TX Buffer	I2S RX Buffer
0x55000001	0x55000001
0x55000002	0x55000002
0x55000003	0x55000003
0x55000004	0x55000004
0x55000005	0x55000005
0x55000006	0x55000006
0x55000007	0x55000007
0x55000008	0x55000008
0x55000009	0x55000009
0x5500000A	0x5500000A
0x5500000B	0x5500000B
0x5500000C	0x5500000C
0x5500000D	0x5500000D
0x5500000E	0x5500000E
0x5500000F	0x5500000F
0x55000010	0x55000010
0x55000011	0x55000011
0x55000012	0x55000012
0x55000013	0x55000013
0x55000014	0x55000014
0x55000015	0x55000015
0x55000016	0x55000016
0x55000017	0x55000017
0x55000018	0x55000018
0x55000019	0x55000019
0x5500001A	0x5500001A
0x5500001B	0x5500001B
0x5500001C	0x5500001C
0x5500001D	0x5500001D
0x5500001E	0x5500001E
0x5500001F	0x5500001F
0x55000020	0x55000020
0x55000021	0x55000021
0x55000022	0x55000022
0x55000023	0x55000023
0x55000024	0x55000024
0x55000025	0x55000025
0x55000026	0x55000026
0x55000027	0x55000027
0x55000028	0x55000028
0x55000029	0x55000029
0x5500002A	0x5500002A
0x5500002B	0x5500002B
0x5500002C	0x5500002C
0x5500002D	0x5500002D
0x5500002E	0x5500002E
0x5500002F	0x5500002F
0x55000030	0x55000030
0x55000031	0x55000031
0x55000032	0x55000032
0x55000033	0x55000033
0x55000034	0x55000034
0x55000035	0x55000035
0x55000036	0x55000036
0x55000037	0x55000037
0x55000038	0x55000038
0x55000039	0x55000039
0x5500003A	0x5500003A
0x5500003B	0x5500003B
0x5500003C	0x5500003C
0x5500003D	0x5500003D
0x5500003E	0x5500003E

```
0x5500003F              0x5500003F
0x55000040              0x55000040


Exit I2S .
```

2 代码介绍

QSPI0 使用 PDMA Basic 模式传输：

```
    /* Enable PDMA channels */
    PDMA_Open(PDMA, (1 << QSPI_MASTER_TX_DMA_CH) | (1 << QSPI_MASTER_RX_DMA_CH) );

    /* Set transfer width (32 bits) and transfer count */
    PDMA_SetTransferCnt(PDMA, QSPI_MASTER_TX_DMA_CH, PDMA_WIDTH_32, TEST_COUNT);

    /* Set source/destination address and attributes */
    PDMA_SetTransferAddr(PDMA, QSPI_MASTER_TX_DMA_CH,
    (uint32_t)g_au32MasterToSlaveTestPattern, PDMA_SAR_INC, (uint32_t)&QSPI0->TX,
    PDMA_DAR_FIX);

    /* Set request source; set basic mode. */
    PDMA_SetTransferMode(PDMA, QSPI_MASTER_TX_DMA_CH, PDMA_QSPI0_TX, FALSE, 0);

    /* Single request type. SPI only support PDMA single request type. */
    PDMA_SetBurstType(PDMA, QSPI_MASTER_TX_DMA_CH, PDMA_REQ_SINGLE, 0);

    /* Disable table interrupt */
    PDMA->DSCT[QSPI_MASTER_TX_DMA_CH].CTL |= PDMA_DSCT_CTL_TBINTDIS_Msk;

    /* Set transfer width (32 bits) and transfer count */
    PDMA_SetTransferCnt(PDMA, QSPI_MASTER_RX_DMA_CH, PDMA_WIDTH_32, TEST_COUNT);

    /* Set source/destination address and attributes */
    PDMA_SetTransferAddr(PDMA, QSPI_MASTER_RX_DMA_CH, (uint32_t)&QSPI0->RX, PDMA_SAR_FIX,
    (uint32_t)g_au32MasterRxBuffer, PDMA_DAR_INC);

    /* Set request source; set basic mode. */
    PDMA_SetTransferMode(PDMA, QSPI_MASTER_RX_DMA_CH, PDMA_QSPI0_RX, FALSE, 0);

    /* Single request type. SPI only support PDMA single request type. */
    PDMA_SetBurstType(PDMA, QSPI_MASTER_RX_DMA_CH, PDMA_REQ_SINGLE, 0);

    /* Disable table interrupt */
    PDMA->DSCT[QSPI_MASTER_RX_DMA_CH].CTL |= PDMA_DSCT_CTL_TBINTDIS_Msk;

    /* Enable SPI master DMA function */
    QSPI_TRIGGER_TX_PDMA(QSPI0);
    QSPI_TRIGGER_RX_PDMA(QSPI0);
```

I2S 使用 PDMA Scatter-Gather 模式传输：

```
    /* Enable PDMA channels */
    PDMA_Open(PDMA, (1 << I2S0_TX_DMA_CH) | (1 << I2S0_RX_DMA_CH));

    /* Tx description */
    g_asDescTable_TX[0].CTL = ((BUFF_LEN - 1) << PDMA_DSCT_CTL_TXCNT_Pos) | PDMA_WIDTH_32
    | PDMA_SAR_INC | PDMA_DAR_FIX | PDMA_REQ_SINGLE | PDMA_OP_SCATTER;
    g_asDescTable_TX[0].SA = (uint32_t)&g_au32MasterRxBuffer;
    g_asDescTable_TX[0].DA = (uint32_t)&I2S0->TXFIFO;
    g_asDescTable_TX[0].FIRST = (uint32_t)&g_asDescTable_TX[1] - (PDMA->SCATBA);

    g_asDescTable_TX[1].CTL = ((BUFF_LEN - 1) << PDMA_DSCT_CTL_TXCNT_Pos) | PDMA_WIDTH_32
    | PDMA_SAR_INC | PDMA_DAR_FIX | PDMA_REQ_SINGLE | PDMA_OP_BASIC;
    g_asDescTable_TX[1].SA = (uint32_t)&g_au32MasterRxBuffer[BUFF_LEN];
    g_asDescTable_TX[1].DA = (uint32_t)&I2S0->TXFIFO;
    g_asDescTable_TX[1].FIRST = (uint32_t)&g_asDescTable_TX[0] - (PDMA->SCATBA);
```

```
    /* Rx description */
    g_asDescTable_RX[0].CTL = ((BUFF_LEN - 1) << PDMA_DSCT_CTL_TXCNT_Pos) | PDMA_WIDTH_32
    | PDMA_SAR_FIX | PDMA_DAR_INC | PDMA_REQ_SINGLE | PDMA_OP_SCATTER;
    g_asDescTable_RX[0].SA = (uint32_t)&I2S0->RXFIFO;
    g_asDescTable_RX[0].DA = (uint32_t)&g_au32SlaveRxBuffer;
    g_asDescTable_RX[0].FIRST = (uint32_t)&g_asDescTable_RX[1] - (PDMA->SCATBA);

    g_asDescTable_RX[1].CTL = (((BUFF_LEN+1) - 1) << PDMA_DSCT_CTL_TXCNT_Pos) |
   PDMA_WIDTH_32 | PDMA_SAR_FIX | PDMA_DAR_INC | PDMA_REQ_SINGLE | PDMA_OP_BASIC;
    g_asDescTable_RX[1].SA = (uint32_t)&I2S0->RXFIFO;
    g_asDescTable_RX[1].DA = (uint32_t)&g_au32SlaveRxBuffer[BUFF_LEN];
    g_asDescTable_RX[1].FIRST = (uint32_t)&g_asDescTable_RX[0] - (PDMA->SCATBA);

    /* Set PDMA Transfer Mode */
    PDMA_SetTransferMode(PDMA, I2S0_TX_DMA_CH, PDMA_I2S0_TX, 1,
    (uint32_t)&g_asDescTable_TX[0]);
    PDMA_SetTransferMode(PDMA, I2S0_RX_DMA_CH, PDMA_I2S0_RX, 1,
    (uint32_t)&g_asDescTable_RX[0]);
```

3 软件与硬件环境

- 软件环境
  - BSP 版本
    - M480 Series BSP CMSIS v3.03.000
  - IDE 版本
    - Keil uVersion 5.26
- 硬件环境
  - 电路组件
    - NuMaker-PFM-M487 Ver3.0

4 目录信息

- EC_M480_QSPI_PDMA_I2S_V1.00
  - Library — Sample code header and source files
    - CMSIS — Cortex® Microcontroller Software Interface Standard (CMSIS) by Arm® Corp.
    - Device — CMSIS compliant device header file
    - StdDriver — All peripheral driver header and source files
  - SampleCode
    - ExampleCode — Source file of example code

5 如何执行范例程序

1. 根据目录信息章节进入 ExampleCode 路径中的 KEIL 文件夹，双击 M480_QSPI_PDMA_I2S.uvproj。
2. 进入编译模式接口
   a. 编译
   b. 下载代码至内存
   c. 进入 / 离开除错模式
3. 进入除错模式接口
   a. 执行代码

6 修订纪录

Date	Revision	Description
Sep. 10, 2019	1.00	1. 初始发布.

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.

Please note that all data and specifications are subject to change without notice.
All the trademarks of products and companies mentioned in this datasheet belong to their respective owners.