PAN1070 SPI Sample Application Note

PAN-CLT-VER-B0, Rev 0.1

PanchipMicroelectronics

www.panchip.com

修订历史

版本	修订日期	描述
V0.1	2023-10-12	初始版本创建

目录

第1章 例程演示内容

1.1 测试内容

1. 寄存器默认值 （Register default value）
2. 数据位选择 （Data size select）
   a) 4 bit 数据
   b) 5 bit 数据
   c) 6 bit 数据
   d) 7 bit 数据
   e) 8 bit 数据
   f) 9 bit 数据
   g) 10 bit 数据
   h) 11 bit 数据
   i) 12 bit 数据
   j) 13 bit 数据
   k) 14 bit 数据
   l) 15 bit 数据
   m) 16 bit 数据
3. 波特率修改 （Baudrate modify）
   a) 波特率 100K
   b) 波特率 500K
   c) 波特率 1M
   d) 波特率 2M
   e) 波特率 5M
   f) 波特率 10M
   g) 自定义波特率
4. 帧格式选择 （Frame format select）
   a) Motorola SPI format with SPO=0, SPH=0
   b) Motorola SPI format with SPO=0, SPH=1
   c) Motorola SPI format with SPO=1, SPH=0
   d) Motorola SPI format with SPO=1, SPH=1
   e) TI Synchronous Serial Frame Format
5. 使用中断方式收发数据 （Interrupt Transfer）
6. 使用 DMA 方式收发数据 （DMA transfer）
   a) Burst Transaction 收发数据
   b) Single Transaction 收发数据
   c) Burst & Single Transaction 收发数据
7. 简单 SPI 传输演示（2 块板子互相收发数据）
8. 低比特先发送功能

1.2 环境准备

1.2.1 软件环境

1.2.1.1 待测代码

测试工程文件：

<PAN1070-DK>\03_MCU\mcu_samples\SPI\keil\SPI.uvprojx

测试源文件目录：

<PAN1070-DK>\03_MCU\mcu_samples \SPI\src

1.2.1.2 软件工具

1、SecureCRT（用于显示 PC 与 COB 的交互过程，打印 log 等）

2、KingstVIS（逻辑分析仪 LA1010 配套软件）

1.2.2 硬件环境

1、PAN1070 COB Test Board

a) UART0（测试交互接口，TX：P16，RX：P17，波特率：921600）

b) SPI0（待测 SPI，CLK: P04，CS: P03，MOSI: P11，MISO: P12）

c) SPI1（辅助测试 SPI，CLK: P22，CS: P23，MOSI: P21，MISO: P24）

d) SWD（用来调试和烧录程序，SWDCLK: P00，SWDIO: P01）

2、逻辑分析仪（波形抓取工具）

3、JLink（SWD 调试与烧录工具）

第2章 例程演示流程

2.1 环境配置

2.1.1 测试程序编译烧录

打开测试工程，确保可以编译通过。编译之前，可以在 spi_common.h 中配置待测 SPI 为 SPI0 或 SPI1。本测试工程默认将 SPI0 作为待测 SPI（Target SPI），将 SPI1 作为辅助测试 SPI（Auxiliary SPI）。

2.1.2 硬件接线

接线方面，对于前 7 个 case，需要待测 COB 板：

1. 将 COB 板的 RX0 和 TX0 进行跳线，把 USB->UART 与 PC 相连；

2. 将 SPI0（待测模块）与 SPI1（辅助测试模块）对应引脚相连。

	SPI0	SPI1
SPI_CLK	P04	P22
SPI_CS	P03	P23
SPI_MISO	P12	P24
SPI_MOSI	P11	P21

对于最后一个 case（Simple Transmission Demo），需要准备 2 个 COB 板，并将两个板子对应的 Target SPI 引脚相连。

2.2 SPI 工作流程

参考 User Manual 文档。

2.3 测试程序初始化

硬件连线完成并烧录测试程序后，执行 Reset，观察串口是否正常打印测试主菜单。

```
+-------------------------------------------------------------------------+
|                        PN107 SPI Sample Code.                           |
+-------------------------------------------------------------------------+
|   Press key to start specific testcase:                                 |
|                                                                         |
|   Input '0'    Testcase 0: Register Default Value Check.                |
|   Input '1'    Testcase 1: Data Length Select and Stop Bit Test.        |
|   Input '2'    Testcase 2: Baudrate Test.                               |
|   Input '3'    Testcase 3: Frame Format Test.                           |
|   Input '4'    Testcase 4: Interrupt Test.                              |
|   Input '5'    Testcase 5: Transfer With DMA Test.                      |
|   Input '6'    Testcase 6: Simple Transmission Demo.                    |
|   Input '7'    Testcase 7: Data lsb send receive Test.                  |
|   Input '8'    Testcase 8: Three wire spi Test.                         |
|   Input '9'    Testcase 8: Three wire spi simple transmission Test.     |
+-------------------------------------------------------------------------+
```

2.4 基本功能验证

2.4.1 SPI 所有寄存器默认状态 SPI_RegisterDefaultValueCheckCase0()

在主菜单下，输入‘0’命令 打印所有寄存器默认值：

测试目的：

检查所有 SPI 相关寄存器复位 Default 值状态。

测试预期：

寄存器默认值应和 PAN1070 Datasheet 上 SPI 模块默认值一致。

测试现象：

```
[19:47:41.194]发→◇0□
[19:47:41.201]收←◆
Target SPI0 Register Default Values:
-----------------------------
CR0         = 0x00000000
CR1         = 0x00000080
DR          = 0x00000000
SR          = 0x00000003
CPSR        = 0x00000000
IMSC        = 0x00000000
RIS         = 0x00000008
MIS         = 0x00000000
ICR         = 0x00000000
DMACR       = 0x00000000

Auxiliary SPI1 Register Default Values:
-----------------------------
CR0         = 0x00000000
CR1         = 0x00000080
DR          = 0x00000000
SR          = 0x00000003
CPSR        = 0x00000000
IMSC        = 0x00000000
RIS         = 0x00000008
MIS         = 0x00000000
ICR         = 0x00000000
DMACR       = 0x00000000
SPI Test OK, Success case: 0
```

测试分析：

参考芯片手册对比寄存器信息，两个 SPI Port 均与手册一致，测试 OK。

2.4.2 数据位选择 SPI_DataSizeSelectTestCase1()

在主菜单下，输入‘1’命令 进入 Subcase 菜单：

```
+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Transfer  4-bit data.                           |
|    Input 'B'    Transfer  5-bit data.                           |
|    Input 'C'    Transfer  6-bit data.                           |
|    Input 'D'    Transfer  7-bit data.                           |
|    Input 'E'    Transfer  8-bit data.                           |
|    Input 'F'    Transfer  9-bit data.                           |
|    Input 'G'    Transfer 10-bit data.                           |
|    Input 'H'    Transfer 11-bit data.                           |
|    Input 'I'    Transfer 12-bit data.                           |
|    Input 'J'    Transfer 13-bit data.                           |
|    Input 'K'    Transfer 14-bit data.                           |
|    Input 'L'    Transfer 15-bit data.                           |
|    Input 'M'    Transfer 16-bit data.                           |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

2.4.2.1 4 bit 数据

测试目的：

验证 4-bit 数据设定下收发数据是否正常。

测试预期：

能够准确收发 4-bit 数据，无法准确收发 4-bit 以上的数据。

测试现象：

输入‘A’命令，首先 Target SPI（Master）发送 8 个数据（范围是 0x0~0xF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致；接着 Target SPI（Master）发送 8 个数据（范围是 0x10~0xFFFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的不一致。

```
a
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0x000c 0x000d 0x000e 0x000f
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x000c 0x000d 0x000e 0x000f

Send data by Target SPI (should be invalid):
 0x0010 0x0011 0x0012 0x0013 0x00cc 0x00dd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x000c 0x000d 0x000e 0x000f

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Transfer  4-bit data.                           |
|    Input 'B'    Transfer  5-bit data.                           |
|    Input 'C'    Transfer  6-bit data.                           |
|    Input 'D'    Transfer  7-bit data.                           |
|    Input 'E'    Transfer  8-bit data.                           |
|    Input 'F'    Transfer  9-bit data.                           |
|    Input 'G'    Transfer 10-bit data.                           |
|    Input 'H'    Transfer 11-bit data.                           |
|    Input 'I'    Transfer 12-bit data.                           |
|    Input 'J'    Transfer 13-bit data.                           |
|    Input 'K'    Transfer 14-bit data.                           |
|    Input 'L'    Transfer 15-bit data.                           |
|    Input 'M'    Transfer 16-bit data.                           |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

前 8 个数据 Master 发送的均是小于 16 的数据，4-bit 能够 cover 到，因此 Slave 能够准确接收到，符合预期。

后 8 个数据 Master 发送的均是大于等于 16 的数据，4-bit 已经无法 cover 到，因此 Slave 无法准确接收，符合预期。

2.4.2.2 5 bit 数据

测试目的：

验证 5-bit 数据设定下收发数据是否正常。

测试预期：

能够准确收发 5-bit 数据，无法准确收发 5-bit 以上的数据。

测试现象：

输入‘B’命令，首先 Target SPI（Master）发送 8 个数据（范围是 0x0~0x1F），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致；接着 Target SPI（Master）发送 8 个数据（范围是 0x20~0xFFFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的不一致。

```
b
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0x001c 0x001d 0x001e 0x001f
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x001c 0x001d 0x001e 0x001f

Send data by Target SPI (should be invalid):
 0x0020 0x0021 0x0022 0x0023 0x00cc 0x00dd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x000c 0x001d 0x000e 0x001f

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Transfer  4-bit data.                           |
|    Input 'B'    Transfer  5-bit data.                           |
|    Input 'C'    Transfer  6-bit data.                           |
|    Input 'D'    Transfer  7-bit data.                           |
|    Input 'E'    Transfer  8-bit data.                           |
|    Input 'F'    Transfer  9-bit data.                           |
|    Input 'G'    Transfer 10-bit data.                           |
|    Input 'H'    Transfer 11-bit data.                           |
|    Input 'I'    Transfer 12-bit data.                           |
|    Input 'J'    Transfer 13-bit data.                           |
|    Input 'K'    Transfer 14-bit data.                           |
|    Input 'L'    Transfer 15-bit data.                           |
|    Input 'M'    Transfer 16-bit data.                           |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

前 8 个数据 Master 发送的均是小于 32 的数据，5-bit 能够 cover 到，因此 Slave 能够准确接收到，符合预期。

后 8 个数据 Master 发送的均是大于等于 32 的数据，5-bit 已经无法 cover 到，因此 Slave 无法准确接收，符合预期。

2.4.2.3 6 bit 数据

测试目的：

验证 6-bit 数据设定下收发数据是否正常。

测试预期：

能够准确收发 6-bit 数据，无法准确收发 6-bit 以上的数据。

测试现象：

输入‘C’命令，首先 Target SPI（Master）发送 8 个数据（范围是 0x0~0x3F），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致；接着 Target SPI（Master）发送 8 个数据（范围是 0x40~0xFFFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的不一致。

```
C
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0x003c 0x003d 0x003e 0x003f
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x003c 0x003d 0x003e 0x003f

Send data by Target SPI (should be invalid):
 0x0040 0x0041 0x0042 0x0043 0x00cc 0x00dd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x000c 0x001d 0x002e 0x003f

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Transfer  4-bit data.                           |
|    Input 'B'    Transfer  5-bit data.                           |
|    Input 'C'    Transfer  6-bit data.                           |
|    Input 'D'    Transfer  7-bit data.                           |
|    Input 'E'    Transfer  8-bit data.                           |
|    Input 'F'    Transfer  9-bit data.                           |
|    Input 'G'    Transfer 10-bit data.                           |
|    Input 'H'    Transfer 11-bit data.                           |
|    Input 'I'    Transfer 12-bit data.                           |
|    Input 'J'    Transfer 13-bit data.                           |
|    Input 'K'    Transfer 14-bit data.                           |
|    Input 'L'    Transfer 15-bit data.                           |
|    Input 'M'    Transfer 16-bit data.                           |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

前 8 个数据 Master 发送的均是小于 64 的数据，6-bit 能够 cover 到，因此 Slave 能够准确接收到，符合预期。

后 8 个数据 Master 发送的均是大于等于 64 的数据，6-bit 已经无法 cover 到，因此 Slave 无法准确接收，符合预期。

2.4.2.4 7 bit 数据

测试目的：

验证 7-bit 数据设定下收发数据是否正常。

测试预期：

能够准确收发 7-bit 数据，无法准确收发 7-bit 以上的数据。

测试现象：

输入‘D’命令，首先 Target SPI（Master）发送 8 个数据（范围是 0x0~0x7F），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致；接着 Target SPI（Master）发送 8 个数据（范围是 0x80~0xFFFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的不一致。

```
d
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0040 0x0044 0x0068 0x006c 0x007e 0x007f
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0040 0x0044 0x0068 0x006c 0x007e 0x007f

Send data by Target SPI (should be invalid):
 0x0080 0x0081 0x0082 0x0083 0x00cc 0x00dd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x004c 0x005d 0x006e 0x007f

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Transfer  4-bit data.                           |
|    Input 'B'    Transfer  5-bit data.                           |
|    Input 'C'    Transfer  6-bit data.                           |
|    Input 'D'    Transfer  7-bit data.                           |
|    Input 'E'    Transfer  8-bit data.                           |
|    Input 'F'    Transfer  9-bit data.                           |
|    Input 'G'    Transfer 10-bit data.                           |
|    Input 'H'    Transfer 11-bit data.                           |
|    Input 'I'    Transfer 12-bit data.                           |
|    Input 'J'    Transfer 13-bit data.                           |
|    Input 'K'    Transfer 14-bit data.                           |
|    Input 'L'    Transfer 15-bit data.                           |
|    Input 'M'    Transfer 16-bit data.                           |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

前 8 个数据 Master 发送的均是小于 128 的数据，7-bit 能够 cover 到，因此 Slave 能够准确接收到，符合预期。

后 8 个数据 Master 发送的均是大于等于 128 的数据，7-bit 已经无法 cover 到，因此 Slave 无法准确接收，符合预期。

2.4.2.5 8 bit 数据

测试目的：

验证 8-bit 数据设定下收发数据是否正常。

测试预期：

能够准确收发 8-bit 数据，无法准确收发 8-bit 以上的数据。

测试现象：

输入‘E’命令，首先 Target SPI（Master）发送 8 个数据（范围是 0x0~0xFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致；接着 Target SPI（Master）发送 8 个数据（范围是 0x100~0xFFFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的不一致。

```
e
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0x00cc 0x00dd 0x00ee 0x00ff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x00cc 0x00dd 0x00ee 0x00ff

Send data by Target SPI (should be invalid):
 0x0100 0x0101 0x0102 0x0103 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x00cc 0x00dd 0x00ee 0x00ff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Transfer  4-bit data.                           |
|    Input 'B'    Transfer  5-bit data.                           |
|    Input 'C'    Transfer  6-bit data.                           |
|    Input 'D'    Transfer  7-bit data.                           |
|    Input 'E'    Transfer  8-bit data.                           |
|    Input 'F'    Transfer  9-bit data.                           |
|    Input 'G'    Transfer 10-bit data.                           |
|    Input 'H'    Transfer 11-bit data.                           |
|    Input 'I'    Transfer 12-bit data.                           |
|    Input 'J'    Transfer 13-bit data.                           |
|    Input 'K'    Transfer 14-bit data.                           |
|    Input 'L'    Transfer 15-bit data.                           |
|    Input 'M'    Transfer 16-bit data.                           |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

前 8 个数据 Master 发送的均是小于 256 的数据，8-bit 能够 cover 到，因此 Slave 能够准确接收到，符合预期。

后 8 个数据 Master 发送的均是大于等于 256 的数据，8-bit 已经无法 cover 到，因此 Slave 无法准确接收，符合预期。

2.4.2.6 9 bit 数据

测试目的：

验证 9-bit 数据设定下收发数据是否正常。

测试预期：

能够准确收发 9-bit 数据，无法准确收发 9-bit 以上的数据。

测试现象：

输入‘F’命令，首先 Target SPI（Master）发送 8 个数据（范围是 0x0~0x1FF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致；接着 Target SPI（Master）发送 8 个数据（范围是 0x200~0xFFFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的不一致。

```
f
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0x01cc 0x01dd 0x01ee 0x01ff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x01cc 0x01dd 0x01ee 0x01ff

Send data by Target SPI (should be invalid):
 0x0200 0x0201 0x0202 0x0203 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x00cc 0x01dd 0x00ee 0x01ff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Transfer  4-bit data.                           |
|    Input 'B'    Transfer  5-bit data.                           |
|    Input 'C'    Transfer  6-bit data.                           |
|    Input 'D'    Transfer  7-bit data.                           |
|    Input 'E'    Transfer  8-bit data.                           |
|    Input 'F'    Transfer  9-bit data.                           |
|    Input 'G'    Transfer 10-bit data.                           |
|    Input 'H'    Transfer 11-bit data.                           |
|    Input 'I'    Transfer 12-bit data.                           |
|    Input 'J'    Transfer 13-bit data.                           |
|    Input 'K'    Transfer 14-bit data.                           |
|    Input 'L'    Transfer 15-bit data.                           |
|    Input 'M'    Transfer 16-bit data.                           |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

前 8 个数据 Master 发送的均是小于 512 的数据，9-bit 能够 cover 到，因此 Slave 能够准确接收到，符合预期。

后 8 个数据 Master 发送的均是大于等于 512 的数据，9-bit 已经无法 cover 到，因此 Slave 无法准确接收，符合预期。

2.4.2.7 10 bit 数据

测试目的：

验证 10-bit 数据设定下收发数据是否正常。

测试预期：

能够准确收发 10-bit 数据，无法准确收发 10-bit 以上的数据。

测试现象：

输入‘G’命令，首先 Target SPI（Master）发送 8 个数据（范围是 0x0~0x3FF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致；接着 Target SPI（Master）发送 8 个数据（范围是 0x400~0xFFFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的不一致。

```
g
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0x03cc 0x03dd 0x03ee 0x03ff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x03cc 0x03dd 0x03ee 0x03ff

Send data by Target SPI (should be invalid):
 0x0400 0x0401 0x0402 0x0403 0xcccc 0xeddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x00cc 0x01dd 0x02ee 0x03ff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Transfer  4-bit data.                           |
|    Input 'B'    Transfer  5-bit data.                           |
|    Input 'C'    Transfer  6-bit data.                           |
|    Input 'D'    Transfer  7-bit data.                           |
|    Input 'E'    Transfer  8-bit data.                           |
|    Input 'F'    Transfer  9-bit data.                           |
|    Input 'G'    Transfer 10-bit data.                           |
|    Input 'H'    Transfer 11-bit data.                           |
|    Input 'I'    Transfer 12-bit data.                           |
|    Input 'J'    Transfer 13-bit data.                           |
|    Input 'K'    Transfer 14-bit data.                           |
|    Input 'L'    Transfer 15-bit data.                           |
|    Input 'M'    Transfer 16-bit data.                           |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

前 8 个数据 Master 发送的均是小于 1024 的数据，10-bit 能够 cover 到，因此 Slave 能够准确接收到，符合预期。

后 8 个数据 Master 发送的均是大于等于 1024 的数据，10-bit 已经无法 cover 到，因此 Slave 无法准确接收，符合预期。

2.4.2.8 11 bit 数据

测试目的：

验证 11-bit 数据设定下收发数据是否正常。

测试预期：

能够准确收发 11-bit 数据，无法准确收发 11-bit 以上的数据。

测试现象：

输入‘H’命令，首先 Target SPI（Master）发送 8 个数据（范围是 0x0~0x7FF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致；接着 Target SPI（Master）发送 8 个数据（范围是 0x800~0xFFFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的不一致。

```
h
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0x07cc 0x07dd 0x07ee 0x07ff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x07cc 0x07dd 0x07ee 0x07ff

Send data by Target SPI (should be invalid):
 0x0800 0x0801 0x0802 0x0803 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x04cc 0x05dd 0x06ee 0x07ff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Transfer  4-bit data.                           |
|    Input 'B'    Transfer  5-bit data.                           |
|    Input 'C'    Transfer  6-bit data.                           |
|    Input 'D'    Transfer  7-bit data.                           |
|    Input 'E'    Transfer  8-bit data.                           |
|    Input 'F'    Transfer  9-bit data.                           |
|    Input 'G'    Transfer 10-bit data.                           |
|    Input 'H'    Transfer 11-bit data.                           |
|    Input 'I'    Transfer 12-bit data.                           |
|    Input 'J'    Transfer 13-bit data.                           |
|    Input 'K'    Transfer 14-bit data.                           |
|    Input 'L'    Transfer 15-bit data.                           |
|    Input 'M'    Transfer 16-bit data.                           |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

前 8 个数据 Master 发送的均是小于 2048 的数据，11-bit 能够 cover 到，因此 Slave 能够准确接收到，符合预期。

后 8 个数据 Master 发送的均是大于等于 2048 的数据，11-bit 已经无法 cover 到，因此 Slave 无法准确接收，符合预期。

2.4.2.9 12 bit 数据

测试目的：

验证 12-bit 数据设定下收发数据是否正常。

测试预期：

能够准确收发 12-bit 数据，无法准确收发 12-bit 以上的数据。

测试现象：

输入‘I’命令，首先 Target SPI（Master）发送 8 个数据（范围是 0x0~0xFFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致；接着 Target SPI（Master）发送 8 个数据（范围是 0x1000~0xFFFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的不一致。

```
i
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0x0fcc 0x0fdd 0x0fee 0x0fff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x0fcc 0x0fdd 0x0fee 0x0fff

Send data by Target SPI (should be invalid):
 0x1000 0x1001 0x1002 0x1003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x0ccc 0x0ddd 0x0eee 0x0fff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Transfer  4-bit data.                           |
|    Input 'B'    Transfer  5-bit data.                           |
|    Input 'C'    Transfer  6-bit data.                           |
|    Input 'D'    Transfer  7-bit data.                           |
|    Input 'E'    Transfer  8-bit data.                           |
|    Input 'F'    Transfer  9-bit data.                           |
|    Input 'G'    Transfer 10-bit data.                           |
|    Input 'H'    Transfer 11-bit data.                           |
|    Input 'I'    Transfer 12-bit data.                           |
|    Input 'J'    Transfer 13-bit data.                           |
|    Input 'K'    Transfer 14-bit data.                           |
|    Input 'L'    Transfer 15-bit data.                           |
|    Input 'M'    Transfer 16-bit data.                           |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

前 8 个数据 Master 发送的均是小于 4096 的数据，12-bit 能够 cover 到，因此 Slave 能够准确接收到，符合预期。

后 8 个数据 Master 发送的均是大于等于 4096 的数据，12-bit 已经无法 cover 到，因此 Slave 无法准确接收，符合预期。

2.4.2.10 13 bit 数据

测试目的：

验证 13-bit 数据设定下收发数据是否正常。

测试预期：

能够准确收发 13-bit 数据，无法准确收发 13-bit 以上的数据。

测试现象：

输入‘J’命令，首先 Target SPI（Master）发送 8 个数据（范围是 0x0~0x1FFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致；接着 Target SPI（Master）发送 8 个数据（范围是 0x2000~0xFFFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的不一致。

```
j
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0x1fcc 0x1fdd 0x1fee 0x1fff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x1fcc 0x1fdd 0x1fee 0x1fff

Send data by Target SPI (should be invalid):
 0x2000 0x2001 0x2002 0x2003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x0ccc 0x1ddd 0x0eee 0x1fff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Transfer  4-bit data.                           |
|    Input 'B'    Transfer  5-bit data.                           |
|    Input 'C'    Transfer  6-bit data.                           |
|    Input 'D'    Transfer  7-bit data.                           |
|    Input 'E'    Transfer  8-bit data.                           |
|    Input 'F'    Transfer  9-bit data.                           |
|    Input 'G'    Transfer 10-bit data.                           |
|    Input 'H'    Transfer 11-bit data.                           |
|    Input 'I'    Transfer 12-bit data.                           |
|    Input 'J'    Transfer 13-bit data.                           |
|    Input 'K'    Transfer 14-bit data.                           |
|    Input 'L'    Transfer 15-bit data.                           |
|    Input 'M'    Transfer 16-bit data.                           |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

前 8 个数据 Master 发送的均是小于 8192 的数据，13-bit 能够 cover 到，因此 Slave 能够准确接收到，符合预期。

后 8 个数据 Master 发送的均是大于等于 8192 的数据，13-bit 已经无法 cover 到，因此 Slave 无法准确接收，符合预期。

2.4.2.11 14 bit 数据

测试目的：

验证 14-bit 数据设定下收发数据是否正常。

测试预期：

能够准确收发 14-bit 数据，无法准确收发 14-bit 以上的数据。

测试现象：

输入‘K’命令，首先 Target SPI（Master）发送 8 个数据（范围是 0x0~0x3FFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致；接着 Target SPI（Master）发送 8 个数据（范围是 0x4000~0xFFFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的不一致。

```
k
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0x3fcc 0x3fdd 0x3fee 0x3fff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x3fcc 0x3fdd 0x3fee 0x3fff

Send data by Target SPI (should be invalid):
 0x4000 0x4001 0x4002 0x4003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x0ccc 0x1ddd 0x2eee 0x3fff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Transfer  4-bit data.                           |
|    Input 'B'    Transfer  5-bit data.                           |
|    Input 'C'    Transfer  6-bit data.                           |
|    Input 'D'    Transfer  7-bit data.                           |
|    Input 'E'    Transfer  8-bit data.                           |
|    Input 'F'    Transfer  9-bit data.                           |
|    Input 'G'    Transfer 10-bit data.                           |
|    Input 'H'    Transfer 11-bit data.                           |
|    Input 'I'    Transfer 12-bit data.                           |
|    Input 'J'    Transfer 13-bit data.                           |
|    Input 'K'    Transfer 14-bit data.                           |
|    Input 'L'    Transfer 15-bit data.                           |
|    Input 'M'    Transfer 16-bit data.                           |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

前 8 个数据 Master 发送的均是小于 16384 的数据，14-bit 能够 cover 到，因此 Slave 能够准确接收到，符合预期。

后 8 个数据 Master 发送的均是大于等于 16384 的数据，14-bit 已经无法 cover 到，因此 Slave 无法准确接收，符合预期。

2.4.2.12 15 bit 数据

测试目的：

验证 15-bit 数据设定下收发数据是否正常。

测试预期：

能够准确收发 15-bit 数据，无法准确收发 15-bit 以上的数据。

测试现象：

输入‘L’命令，首先 Target SPI（Master）发送 8 个数据（范围是 0x0~0x7FFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致；接着 Target SPI（Master）发送 8 个数据（范围是 0x8000~0xFFFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的不一致。

```
]
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0x7fcc 0x7fdd 0x7fee 0x7fff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x7fcc 0x7fdd 0x7fee 0x7fff

Send data by Target SPI (should be invalid):
 0x8000 0x8001 0x8002 0x8003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x4ccc 0x5ddd 0x6eee 0x7fff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Transfer  4-bit data.                           |
|    Input 'B'    Transfer  5-bit data.                           |
|    Input 'C'    Transfer  6-bit data.                           |
|    Input 'D'    Transfer  7-bit data.                           |
|    Input 'E'    Transfer  8-bit data.                           |
|    Input 'F'    Transfer  9-bit data.                           |
|    Input 'G'    Transfer 10-bit data.                           |
|    Input 'H'    Transfer 11-bit data.                           |
|    Input 'I'    Transfer 12-bit data.                           |
|    Input 'J'    Transfer 13-bit data.                           |
|    Input 'K'    Transfer 14-bit data.                           |
|    Input 'L'    Transfer 15-bit data.                           |
|    Input 'M'    Transfer 16-bit data.                           |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

前 8 个数据 Master 发送的均是小于 32768 的数据，15-bit 能够 cover 到，因此 Slave 能够准确接收到，符合预期。

后 8 个数据 Master 发送的均是大于等于 32768 的数据，15-bit 已经无法 cover 到，因此 Slave 无法准确接收，符合预期。

2.4.2.13 16 bit 数据

测试目的：

验证 16-bit 数据设定下收发数据是否正常。

测试预期：

能够准确收发 16-bit 数据。

测试现象：

输入‘M’命令，Target SPI（Master）发送 8 个数据（范围是 0x0~0xFFFF），Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致。

```
m
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Transfer  4-bit data.                           |
|    Input 'B'    Transfer  5-bit data.                           |
|    Input 'C'    Transfer  6-bit data.                           |
|    Input 'D'    Transfer  7-bit data.                           |
|    Input 'E'    Transfer  8-bit data.                           |
|    Input 'F'    Transfer  9-bit data.                           |
|    Input 'G'    Transfer 10-bit data.                           |
|    Input 'H'    Transfer 11-bit data.                           |
|    Input 'I'    Transfer 12-bit data.                           |
|    Input 'J'    Transfer 13-bit data.                           |
|    Input 'K'    Transfer 14-bit data.                           |
|    Input 'L'    Transfer 15-bit data.                           |
|    Input 'M'    Transfer 16-bit data.                           |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

16-bit 能够 cover 到 SPI Module 硬件支持的所有情况，因此 Slave 均能够准确接收到，符合预期。

2.4.3 波特率修改 SPI_BaudrateTestCase2()

在主菜单下，输入 ‘2’ 命令 进入 Subcase 菜单：

```
+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    16-bit data under 100K baudrate.                |
|    Input 'B'    16-bit data under 500K baudrate.                |
|    Input 'C'    16-bit data under 1M   baudrate.                |
|    Input 'D'    16-bit data under 2M   baudrate.                |
|    Input 'E'    16-bit data under 5M   baudrate.                |
|    Input 'F'    16-bit data under 10M  baudrate.                |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

2.4.3.1 波特率 100K

测试目的：

验证 100K 波特率下是否工作正常。

测试预期：

100K 波特率下能够准确收发数据。

测试现象：

输入 ‘A’ 命令，出现波特率过低的提示。

```
aTest failed, SPI baudrate is too high or to low!

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    16-bit data under 100K baudrate.                |
|    Input 'B'    16-bit data under 500K baudrate.                |
|    Input 'C'    16-bit data under 1M   baudrate.                |
|    Input 'D'    16-bit data under 2M   baudrate.                |
|    Input 'E'    16-bit data under 5M   baudrate.                |
|    Input 'F'    16-bit data under 10M  baudrate.                |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

由于测试 APB 时钟直接从 48MHz AHB 过来，并未分频，从而导致 SPI Driver 未能找到合适的分频系数来产生 100KHz 的 SPI 时钟，于是报错，是预期现象。

2.4.3.2 波特率 500K

测试目的：

验证 500K 波特率下是否工作正常。

测试预期：

500K 波特率下能够准确收发数据。

测试现象：

输入‘B’命令，Target SPI（Master）发送 8 个数据，Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致。

```
b
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    16-bit data under 100K baudrate.                |
|    Input 'B'    16-bit data under 500K baudrate.                |
|    Input 'C'    16-bit data under 1M   baudrate.                |
|    Input 'D'    16-bit data under 2M   baudrate.                |
|    Input 'E'    16-bit data under 5M   baudrate.                |
|    Input 'F'    16-bit data under 10M  baudrate.                |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

Slave 端能准确收到 Master 发来的数据，说明 500K 波特率下 SPI 收发数据均正常，符合预期。

2.4.3.3 波特率 1M

测试目的：

验证 1M 波特率下是否工作正常。

测试预期：

1M 波特率下能够准确收发数据。

测试现象：

输入‘C’命令，Target SPI（Master）发送 8 个数据，Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致。

```
C
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    16-bit data under 100K baudrate.                |
|    Input 'B'    16-bit data under 500K baudrate.                |
|    Input 'C'    16-bit data under 1M   baudrate.                |
|    Input 'D'    16-bit data under 2M   baudrate.                |
|    Input 'E'    16-bit data under 5M   baudrate.                |
|    Input 'F'    16-bit data under 10M  baudrate.                |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

Slave 端能准确收到 Master 发来的数据，说明 1M 波特率下 SPI 收发数据均正常，符合预期。

2.4.3.4 波特率 2M

测试目的：

验证 2M 波特率下是否工作正常。

测试预期：

2M 波特率下能够准确收发数据。

测试现象：

输入‘D’命令，Target SPI（Master）发送 8 个数据，Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致。

```
d
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    16-bit data under 100K baudrate.                |
|    Input 'B'    16-bit data under 500K baudrate.                |
|    Input 'C'    16-bit data under 1M   baudrate.                |
|    Input 'D'    16-bit data under 2M   baudrate.                |
|    Input 'E'    16-bit data under 5M   baudrate.                |
|    Input 'F'    16-bit data under 10M  baudrate.                |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

Slave 端能准确收到 Master 发来的数据，说明 2M 波特率下 SPI 收发数据均正常，符合预期。

2.4.3.5 波特率 5M

测试目的：

验证 5M 波特率下是否工作正常。

测试预期：

5M 波特率下能够准确收发数据。

测试现象：

输入‘E’命令，Target SPI（Master）发送 8 个数据，Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的数据一致。

```
e
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    16-bit data under 100K baudrate.                |
|    Input 'B'    16-bit data under 500K baudrate.                |
|    Input 'C'    16-bit data under 1M   baudrate.                |
|    Input 'D'    16-bit data under 2M   baudrate.                |
|    Input 'E'    16-bit data under 5M   baudrate.                |
|    Input 'F'    16-bit data under 10M  baudrate.                |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

Slave 端能准确收到 Master 发来的数据，说明 5M 波特率下 SPI 收发数据均正常，符合预期。

2.4.3.6 波特率 10M

测试目的：

验证 10M 波特率下是否工作正常。

测试预期：

10M 波特率下能够准确收发数据。

测试现象：

输入‘F’命令，Target SPI（Master）发送 8 个数据，Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的数据一致。

```
f
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    16-bit data under 100K baudrate.                |
|    Input 'B'    16-bit data under 500K baudrate.                |
|    Input 'C'    16-bit data under 1M   baudrate.                |
|    Input 'D'    16-bit data under 2M   baudrate.                |
|    Input 'E'    16-bit data under 5M   baudrate.                |
|    Input 'F'    16-bit data under 10M  baudrate.                |
|    Input 'G'    16-bit data under baudrate by user.             |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

Slave 端能准确收到 Master 发来的数据，说明 10M 波特率下 SPI 收发数据均正常，符合预期。

SPI 波特率过高可能导致传输出现问题，建议 SPI Clock 不应超过 APB Clock 的 1/12 比较好，由于测试程序的 AHB 和 APB 时钟均为 48M，因此 SPI Clock 最好不超过 48M / 12 = 4M。

2.4.3.7 自定义波特率

测试目的：

验证自定义波特率下是否工作正常。

测试预期：

自定义波特率下能够准确收发数据。

测试现象：

输入‘G’命令，Target SPI（Master）发送 8 个数据，Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的数据一致。

```
[10:12:33.022]发→◇G□
[10:12:33.026]收←◆
input the baudrate value u want

[10:12:34.335]发→◇16000000
□
[10:12:34.339]收←◆baudrate select:16000000

Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
```

Fail case：

```
[10:12:25.256]发→◇G□
[10:12:25.259]收←◆
input the baudrate value u want

[10:12:26.372]发→◇16100000
□
[10:12:26.377]收←◆baudrate select:16100000

Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0002 0x0004 0x0007 0x9999 0xbbbb 0xdddd 0xfffe
```

测试分析：

48M APB clk 情况下， SPI 波特率最高为 16M。

2.4.4 帧格式选择 SPI_FrameFormatTestCase3()

在主菜单下，输入‘3’命令 进入 Subcase 菜单：

```
+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Motorola SPI Format with SPO=0, SPH=0.          |
|    Input 'B'    Motorola SPI Format with SPO=0, SPH=1.          |
|    Input 'C'    Motorola SPI Format with SPO=1, SPH=0.          |
|    Input 'D'    Motorola SPI Format with SPO=1, SPH=1.          |
|    Input 'E'    TI Synchronous Serial Frame Format.             |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

2.4.4.1 帧格式设定为 Motorola SPI Format，且 SPO=0，SPH=0

测试目的：

验证帧格式设置为 Motorola SPI 格式，且 SPO=0，SPH=0 的情况下，收发数据是否正常。

测试预期：

收发数据正常。

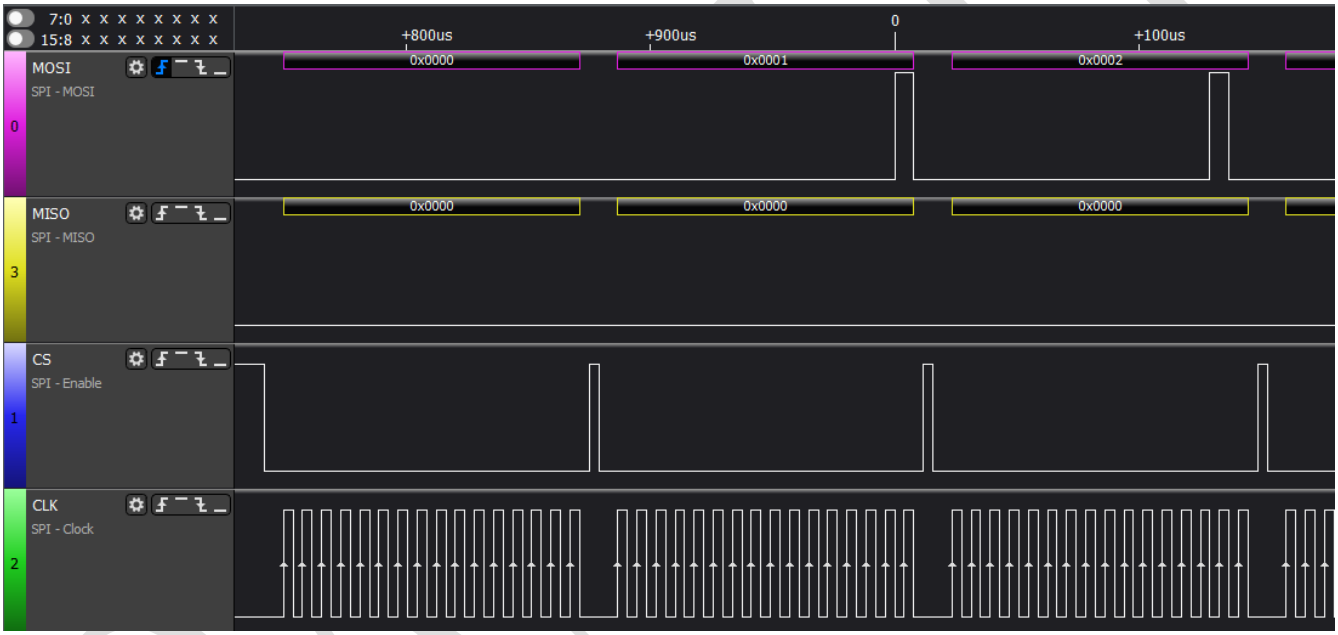
测试现象：

输入‘A’命令，Target SPI（Master）发送 8 个数据，Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致。

```
a
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff

+---------------------------------------------------------------------+
|    Press key to test specific function:                             |
|                                                                     |
|    Input 'A'    Motorola SPI Format with SPO=0, SPH=0.              |
|    Input 'B'    Motorola SPI Format with SPO=0, SPH=1.              |
|    Input 'C'    Motorola SPI Format with SPO=1, SPH=0.              |
|    Input 'D'    Motorola SPI Format with SPO=1, SPH=1.              |
|    Input 'E'    TI Synchronous Serial Frame Format.                 |
|    Press ESC key to back to the top level case list.                |
+---------------------------------------------------------------------+
```

使用逻辑分析仪抓取波形如下图所示：



测试分析：

从 Log 可知 Slave 能准确收到 Master 发来的数据，SPI 收发数据均正常。

从波形图可以看出，CLK 信号在空闲状态下是低电平，说明 SPO=0 配置生效；数据在 CLK 第一个边沿（上升沿）被采样，CS 信号在传输两个 Data 之间的空隙短暂变为高电平，说明 SPH=0 配置生效。

综上，数据与波形均符合预期。

2.4.4.2 帧格式设定为 Motorola SPI Format，且 SPO=0，SPH=1

测试目的：

验证帧格式设置为 Motorola SPI 格式，且 SPO=0，SPH=1 的情况下，收发数据是否正常。

测试预期：

收发数据正常。

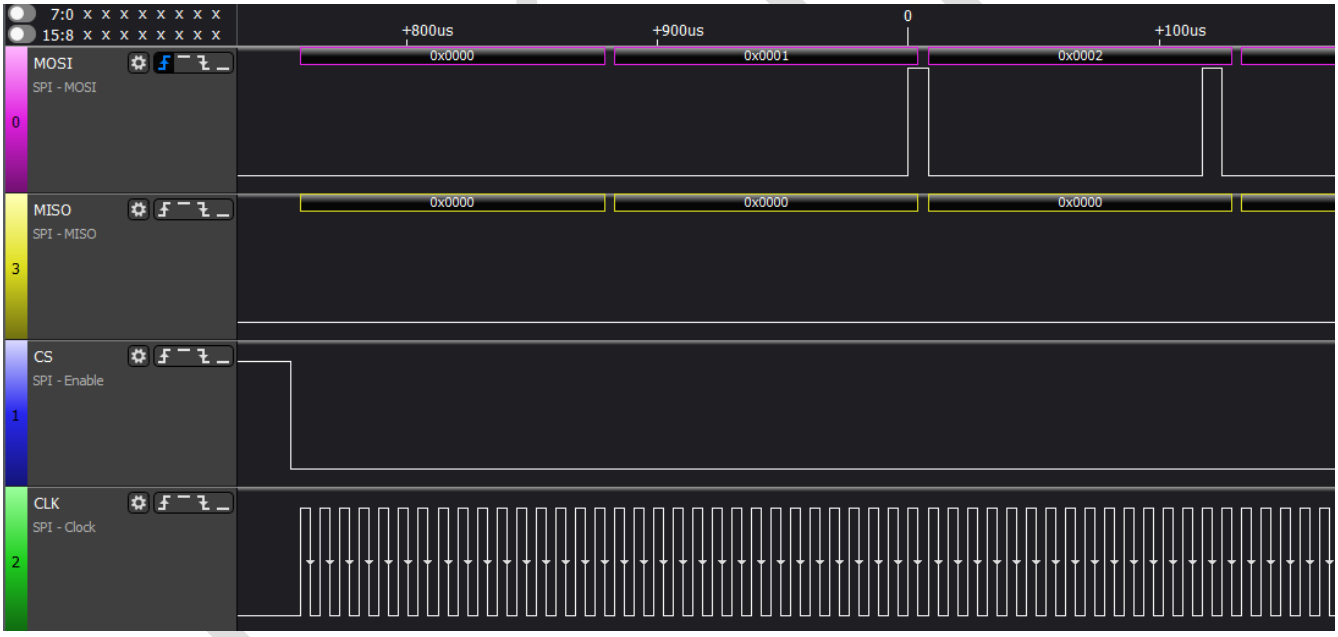
测试现象：

输入‘B’命令，Target SPI（Master）发送 8 个数据，Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致。

```
b
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Motorola SPI Format with SPO=0, SPH=0.          |
|    Input 'B'    Motorola SPI Format with SPO=0, SPH=1.          |
|    Input 'C'    Motorola SPI Format with SPO=1, SPH=0.          |
|    Input 'D'    Motorola SPI Format with SPO=1, SPH=1.          |
|    Input 'E'    TI Synchronous Serial Frame Format.             |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

使用逻辑分析仪抓取波形如下图所示：



测试分析：

从 Log 可知 Slave 能准确收到 Master 发来的数据，SPI 收发数据均正常。

从波形图可以看出，CLK 信号在空闲状态下是低电平，说明 SPO=0 配置生效；数据在 CLK 第二个边沿（下降沿）被采样，CS 信号在连续传输多个数据的情况下一直保持低电平，说明 SPH=1 配置生效。

综上，数据与波形均符合预期。

2.4.4.3 帧格式设定为 Motorola SPI Format，且 SPO=1，SPH=0

测试目的：

验证帧格式设置为 Motorola SPI 格式，且 SPO=1，SPH=0 的情况下，收发数据是否正常。

测试预期：

收发数据正常。

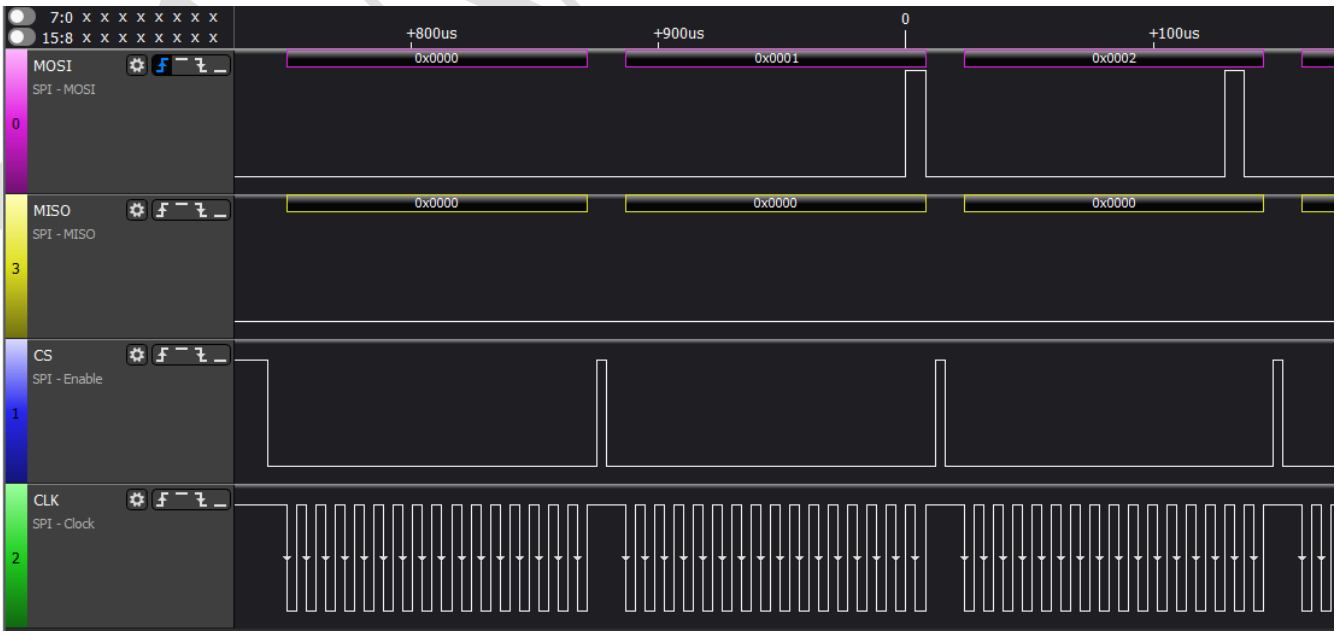
测试现象：

输入‘C’命令，Target SPI（Master）发送 8 个数据，Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致。

```
C
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Motorola SPI Format with SPO=0, SPH=0.          |
|    Input 'B'    Motorola SPI Format with SPO=0, SPH=1.          |
|    Input 'C'    Motorola SPI Format with SPO=1, SPH=0.          |
|    Input 'D'    Motorola SPI Format with SPO=1, SPH=1.          |
|    Input 'E'    TI Synchronous Serial Frame Format.             |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

使用逻辑分析仪抓取波形如下图所示：



测试分析：

从 Log 可知 Slave 能准确收到 Master 发来的数据，SPI 收发数据均正常。

从波形图可以看出，CLK 信号在空闲状态下是高电平，说明 SPO=1 配置生效；数据在 CLK 第一个边沿（下降沿）被采样，CS 信号在连续传输多个数据的情况下，在每两个数据传输间隙会短暂拉高，说明 SPH=0 配置生效。

综上，数据与波形均符合预期。

2.4.4.4 帧格式设定为 Motorola SPI Format，且 SPO=1，SPH=1

测试目的：

验证帧格式设置为 Motorola SPI 格式，且 SPO=1，SPH=1 的情况下，收发数据是否正常。

测试预期：

收发数据正常。

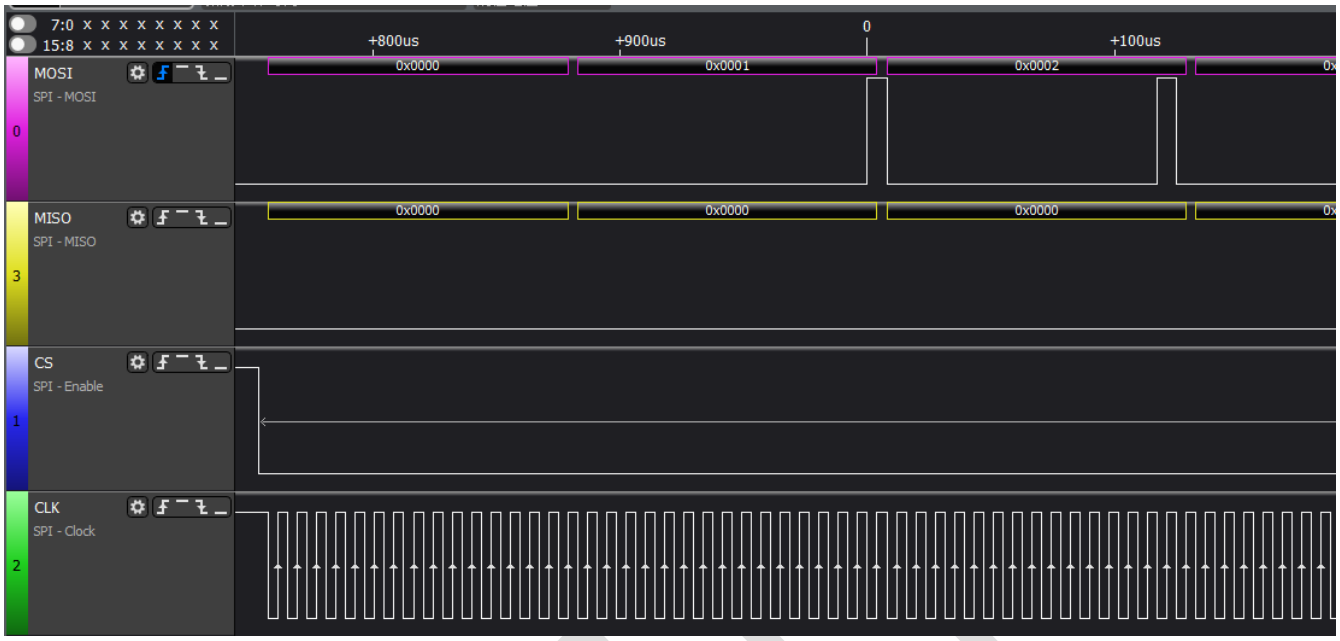
测试现象：

输入‘D’命令，Target SPI（Master）发送 8 个数据，Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致。

```
d
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Motorola SPI Format with SPO=0, SPH=0.          |
|    Input 'B'    Motorola SPI Format with SPO=0, SPH=1.          |
|    Input 'C'    Motorola SPI Format with SPO=1, SPH=0.          |
|    Input 'D'    Motorola SPI Format with SPO=1, SPH=1.          |
|    Input 'E'    TI Synchronous Serial Frame Format.             |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

使用逻辑分析仪抓取波形如下图所示：



测试分析：

从 Log 可知 Slave 能准确收到 Master 发来的数据，SPI 收发数据均正常。

从波形图可以看出，CLK 信号在空闲状态下是高电平，说明 SPO=1 配置生效；数据在 CLK 第二个边沿（上升沿）被采样，CS 信号在连续传输多个数据的情况下一直保持低电平，说明 SPH=1 配置生效。

综上，数据与波形均符合预期。

2.4.4.5 帧格式设定为 TI Synchronous Serial Frame Format

测试目的：

验证帧格式设置为 TI Synchronous Serial Frame 格式，收发数据是否正常。

测试预期：

收发数据正常。

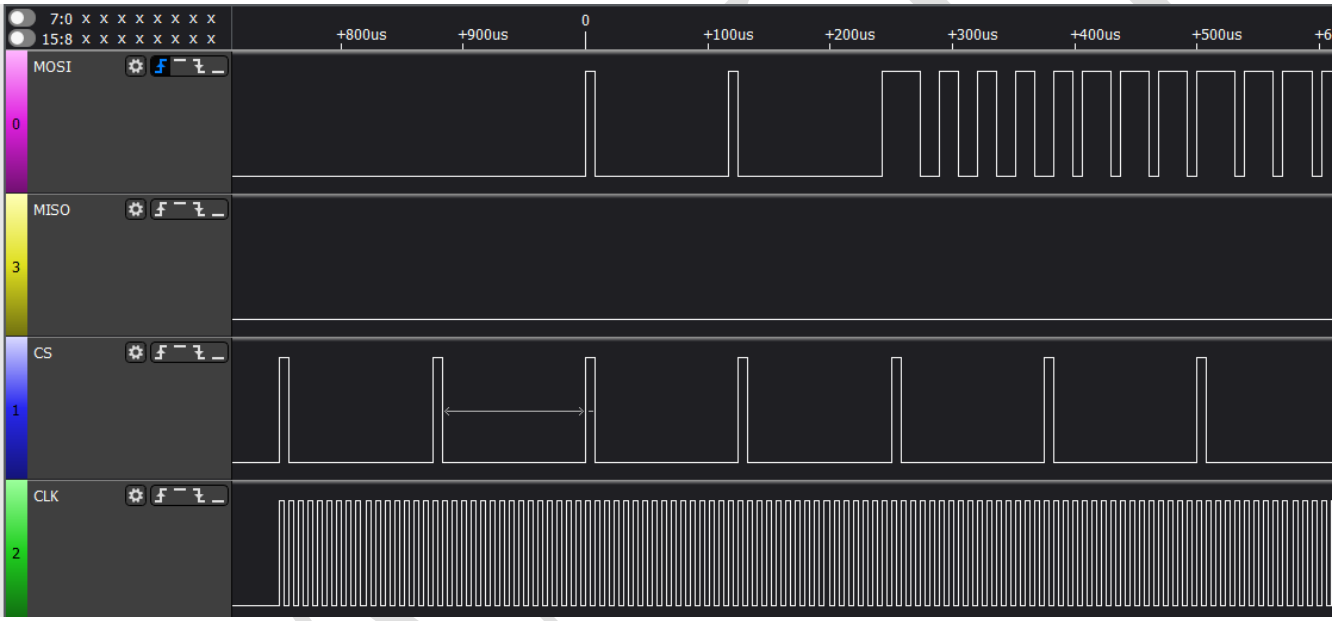
测试现象：

输入‘E’命令，Target SPI（Master）发送 8 个数据，Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致。

```
e
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Motorola SPI Format with SPO=0, SPH=0.          |
|    Input 'B'    Motorola SPI Format with SPO=0, SPH=1.          |
|    Input 'C'    Motorola SPI Format with SPO=1, SPH=0.          |
|    Input 'D'    Motorola SPI Format with SPO=1, SPH=1.          |
|    Input 'E'    TI Synchronous Serial Frame Format.             |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

使用逻辑分析仪抓取波形如下图所示：



测试分析：

从 Log 可知 Slave 能准确收到 Master 发来的数据，SPI 收发数据均正常。

从波形图可以看出，CLK 信号在空闲状态下是低电平，CS 信号在连续传输多个数据的情况下，相邻两个数据之间电平会短暂拉高，且此时 CLK 信号仍然连续，符合 TI Synchronous Seral Frame 格式特点。

综上，数据与波形均符合预期。

2.4.5 使用中断方式收发数据 SPI_InterruptTestCase4()

在主菜单下，输入‘4’命令 进入 Subcase 菜单：

```
+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Send/Recv data with master interrupt enabled.   |
|    Input 'B'    Rx FIFO overrun test.                           |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

2.4.5.1 使用中断方式正常收发数据

测试目的：

验证中断方式下收发数据是否正常。

测试预期：

收发数据正常。

测试现象：

输入‘A’命令，Master 向 Slave 发送 160 Bytes 的数据，同时也从 Slave 端接收数据。其中 Master 端的发送接收均使用中断方式，Slave 端的发送接收均使用查询方式。

Master 发，Slave 收的数据：

```
a
Send data by Target SPI with Tx INT enabled:
 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07
 0x08 0x09 0x0a 0x0b 0x0c 0x0d 0x0e 0x0f
 0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17
 0x18 0x19 0x1a 0x1b 0x1c 0x1d 0x1e 0x1f
 0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27
 0x28 0x29 0x2a 0x2b 0x2c 0x2d 0x2e 0x2f
 0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37
 0x38 0x39 0x3a 0x3b 0x3c 0x3d 0x3e 0x3f
 0x40 0x41 0x42 0x43 0x44 0x45 0x46 0x47
 0x48 0x49 0x4a 0x4b 0x4c 0x4d 0x4e 0x4f
 0x50 0x51 0x52 0x53 0x54 0x55 0x56 0x57
 0x58 0x59 0x5a 0x5b 0x5c 0x5d 0x5e 0x5f
 0x60 0x61 0x62 0x63 0x64 0x65 0x66 0x67
 0x68 0x69 0x6a 0x6b 0x6c 0x6d 0x6e 0x6f
 0x70 0x71 0x72 0x73 0x74 0x75 0x76 0x77
 0x78 0x79 0x7a 0x7b 0x7c 0x7d 0x7e 0x7f
 0x80 0x81 0x82 0x83 0x84 0x85 0x86 0x87
 0x88 0x89 0x8a 0x8b 0x8c 0x8d 0x8e 0x8f
 0x90 0x91 0x92 0x93 0x94 0x95 0x96 0x97
 0x98 0x99 0x9a 0x9b 0x9c 0x9d 0x9e 0x9f

Data received by Auxiliary SPI:
 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07
 0x08 0x09 0x0a 0x0b 0x0c 0x0d 0x0e 0x0f
 0x10 0x11 0x12 0x13 0x14 0x15 0x16 0x17
 0x18 0x19 0x1a 0x1b 0x1c 0x1d 0x1e 0x1f
 0x20 0x21 0x22 0x23 0x24 0x25 0x26 0x27
 0x28 0x29 0x2a 0x2b 0x2c 0x2d 0x2e 0x2f
 0x30 0x31 0x32 0x33 0x34 0x35 0x36 0x37
 0x38 0x39 0x3a 0x3b 0x3c 0x3d 0x3e 0x3f
 0x40 0x41 0x42 0x43 0x44 0x45 0x46 0x47
 0x48 0x49 0x4a 0x4b 0x4c 0x4d 0x4e 0x4f
 0x50 0x51 0x52 0x53 0x54 0x55 0x56 0x57
 0x58 0x59 0x5a 0x5b 0x5c 0x5d 0x5e 0x5f
 0x60 0x61 0x62 0x63 0x64 0x65 0x66 0x67
 0x68 0x69 0x6a 0x6b 0x6c 0x6d 0x6e 0x6f
 0x70 0x71 0x72 0x73 0x74 0x75 0x76 0x77
 0x78 0x79 0x7a 0x7b 0x7c 0x7d 0x7e 0x7f
 0x80 0x81 0x82 0x83 0x84 0x85 0x86 0x87
 0x88 0x89 0x8a 0x8b 0x8c 0x8d 0x8e 0x8f
 0x90 0x91 0x92 0x93 0x94 0x95 0x96 0x97
 0x98 0x99 0x9a 0x9b 0x9c 0x9d 0x9e 0x9f
```

Master 收，Slave 发的数据：

```
Send data by Auxiliary SPI:
 0x11 0x11 0x11 0x11 0x11 0x11 0x11 0x11
 0x11 0x11 0x11 0x11 0x11 0x11 0x11 0x11
 0x22 0x22 0x22 0x22 0x22 0x22 0x22 0x22
 0x22 0x22 0x22 0x22 0x22 0x22 0x22 0x22
 0x33 0x33 0x33 0x33 0x33 0x33 0x33 0x33
 0x33 0x33 0x33 0x33 0x33 0x33 0x33 0x33
 0x44 0x44 0x44 0x44 0x44 0x44 0x44 0x44
 0x44 0x44 0x44 0x44 0x44 0x44 0x44 0x44
 0x55 0x55 0x55 0x55 0x55 0x55 0x55 0x55
 0x55 0x55 0x55 0x55 0x55 0x55 0x55 0x55
 0x66 0x66 0x66 0x66 0x66 0x66 0x66 0x66
 0x66 0x66 0x66 0x66 0x66 0x66 0x66 0x66
 0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x77
 0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x77
 0x88 0x88 0x88 0x88 0x88 0x88 0x88 0x88
 0x88 0x88 0x88 0x88 0x88 0x88 0x88 0x88
 0x99 0x99 0x99 0x99 0x99 0x99 0x99 0x99
 0x99 0x99 0x99 0x99 0x99 0x99 0x99 0x99
 0xaa 0xaa 0xaa 0xaa 0xaa 0xaa 0xaa 0xaa
 0xaa 0xaa 0xaa 0xaa 0xaa 0xaa 0xaa 0xaa

Data received by Target SPI with Rx INT enabled:
 0x11 0x11 0x11 0x11 0x11 0x11 0x11 0x11
 0x11 0x11 0x11 0x11 0x11 0x11 0x11 0x11
 0x22 0x22 0x22 0x22 0x22 0x22 0x22 0x22
 0x22 0x22 0x22 0x22 0x22 0x22 0x22 0x22
 0x33 0x33 0x33 0x33 0x33 0x33 0x33 0x33
 0x33 0x33 0x33 0x33 0x33 0x33 0x33 0x33
 0x44 0x44 0x44 0x44 0x44 0x44 0x44 0x44
 0x44 0x44 0x44 0x44 0x44 0x44 0x44 0x44
 0x55 0x55 0x55 0x55 0x55 0x55 0x55 0x55
 0x55 0x55 0x55 0x55 0x55 0x55 0x55 0x55
 0x66 0x66 0x66 0x66 0x66 0x66 0x66 0x66
 0x66 0x66 0x66 0x66 0x66 0x66 0x66 0x66
 0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x77
 0x77 0x77 0x77 0x77 0x77 0x77 0x77 0x77
 0x88 0x88 0x88 0x88 0x88 0x88 0x88 0x88
 0x88 0x88 0x88 0x88 0x88 0x88 0x88 0x88
 0x99 0x99 0x99 0x99 0x99 0x99 0x99 0x99
 0x99 0x99 0x99 0x99 0x99 0x99 0x99 0x99
 0xaa 0xaa 0xaa 0xaa 0xaa 0xaa 0xaa 0xaa
 0xaa 0xaa 0xaa 0xaa 0xaa 0xaa 0xaa 0xaa
```

测试分析：

由 Log 可以看到，Master 端使用中断方式，发送和接收数据均正常，符合预期。

2.4.5.2 触发 Receive FIFO Overrun 中断

测试目的：

验证 Receive FIFO Overrun 中断是否正常。

测试预期：

Rx FIFO Overrun 中断开启后，可以成功被触发。

测试现象：

输入‘B’命令，Master 开启 Rx Overrun 中断，并持续产生 CLK 信号，从 Slave 端接收数据。发现 Log 打印多条“SPI Rx Overrun!”字样。

```
bSPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!
SPI Rx Overrun!

+------------------------------------------------------------------+
|    Press key to test specific function:                          |
|                                                                  |
|    Input 'A'    Send/Recv data with master interrupt enabled.    |
|    Input 'B'    Rx FIFO overrun test.                            |
|    Press ESC key to back to the top level case list.             |
+------------------------------------------------------------------+
```

测试分析：

当 Rx FIFO Overrun 中断开启后，Rx FIFO 被填满后，继续有数据被接收，将会触发 Overrun 中断，从 Log 看到的现象是符合预期的。

2.4.6 使用 DMA 方式收发数据 SPI_TransferWithDmaTestCase5()

在主菜单下，输入‘5’ 命令 进入 Subcase 菜单：

```
+------------------------------------------------------------------+
|    Press key to test specific function:                          |
|                                                                  |
|    Input 'A'    Burst transaction.                               |
|    Input 'B'    Single transaction.                              |
|    Input 'C'    Both single and burst transactions.              |
|                                                                  |
|    Press ESC key to back to the top level case list.             |
+------------------------------------------------------------------+
```

2.4.6.1 Burst Transaction 收发数据

测试目的：

验证 DMA Burst Transaction 下 SPI 收发数据是否正常。

测试预期：

DMA Burst Transaction 下 SPI 可以正常收发数据。

测试现象：

输入‘A’命令，Master 向 Slave 发送 128 Bytes 的数据，同时也从 Slave 端接收数据。其中 Master 端的发送接收均使用 DMA 方式，Slave 端的发送接收均使用查询方式。

从 Log 可以看到，Master 使用 DMA 方式发送数据，Slave 准确收到；Slave 发送数据，Master 使用 DMA 方式也准确接收到。

```
aStart DMA Rx/Tx...
Send data by DMA done (channel=1, data_len=128 bytes).
DMA Interrupt Trigger Count: tfr=2 dsttfr=14, dsttfr_tx=14, err_tx=0

Data sent by DMA:
 0x0000 0x0001 0x0002 0x0003 0x0004 0x0005 0x0006 0x0007
 0x0008 0x0009 0x000a 0x000b 0x000c 0x000d 0x000e 0x000f
 0x0010 0x0011 0x0012 0x0013 0x0014 0x0015 0x0016 0x0017
 0x0018 0x0019 0x001a 0x001b 0x001c 0x001d 0x001e 0x001f
 0x0020 0x0021 0x0022 0x0023 0x0024 0x0025 0x0026 0x0027
 0x0028 0x0029 0x002a 0x002b 0x002c 0x002d 0x002e 0x002f
 0x0030 0x0031 0x0032 0x0033 0x0034 0x0035 0x0036 0x0037
 0x0038 0x0039 0x003a 0x003b 0x003c 0x003d 0x003e 0x003f
Data received by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x0004 0x0005 0x0006 0x0007
 0x0008 0x0009 0x000a 0x000b 0x000c 0x000d 0x000e 0x000f
 0x0010 0x0011 0x0012 0x0013 0x0014 0x0015 0x0016 0x0017
 0x0018 0x0019 0x001a 0x001b 0x001c 0x001d 0x001e 0x001f
 0x0020 0x0021 0x0022 0x0023 0x0024 0x0025 0x0026 0x0027
 0x0028 0x0029 0x002a 0x002b 0x002c 0x002d 0x002e 0x002f
 0x0030 0x0031 0x0032 0x0033 0x0034 0x0035 0x0036 0x0037
 0x0038 0x0039 0x003a 0x003b 0x003c 0x003d 0x003e 0x003f

Data received by DMA done (channel=0, data_len=128 bytes):
DMA Interrupt Trigger Count: tfr=0, srctfr=16, srctfr_rx=16, err_rx=0

Data sent by Auxiliary SPI:
 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011
 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011
 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022
 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022
 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033
 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033
 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044
 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044

Data received by DMA:
 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011
 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011
 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022
 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022
 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033
 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033
 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044
 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044
```

测试分析：

使用 DMA 收发 128 字节数据，测试程序中 SPI Burst Length 设置为 4 字节，Data Width 设置为 2 字节，所以一次 Burst 传输 8 字节，可以整除 128，因此 DMA 只会有 Burst Transaction，从现象看收发数据均正常，符合预期。

2.4.6.2 Single Transaction 收发数据

测试目的：

验证 DMA Single Transaction 下 SPI 收发数据是否正常。

测试预期：

DMA Single Transaction 下 SPI 可以正常收发数据。

测试现象：

输入‘B’命令，Master 向 Slave 发送 4 Bytes 的数据，同时也从 Slave 端接收数据。其中 Master 端的发送接收均使用 DMA 方式，Slave 端的发送接收均使用查询方式。

从 Log 可以看到，Master 使用 DMA 方式发送数据，Slave 准确收到；Slave 发送数据，Master 使用 DMA 方式也准确接收到。

```
bStart DMA Rx/Tx...
Send data by DMA done (channel=1, data_len=4 bytes).
DMA Interrupt Trigger Count: tfr=2 dsttfr=1, dsttfr_tx=1, err_tx=0

Data sent by DMA:
 0x0000 0x0001
Data received by Auxiliary SPI:
 0x0000 0x0001

Data received by DMA done (channel=0, data_len=4 bytes):
DMA Interrupt Trigger Count: tfr=0, srctfr=1, srctfr_rx=1, err_rx=0

Data sent by Auxiliary SPI:
 0x0011 0x0011

Data received by DMA:
 0x0011 0x0011
```

测试分析：

使用 DMA 收发 4 字节数据，测试程序中 SPI 一次 Burst 为 2*4=8 字节，大于需要收发的数据大小，因此 DMA 只会有 Single Transaction，从现象看收发数据均正常，符合预期。

2.4.6.3 Burst & Single Transaction 收发数据

测试目的：

验证 DMA Burst 和 Single Transaction 均有的情况下 SPI 收发数据是否正常。

测试预期：

DMA Burst 和 Single Transaction 均有的情况下 SPI 可以正常收发数据。

测试现象：

输入‘C’命令，Master 向 Slave 发送 132 Bytes 的数据，同时也从 Slave 端接收数据。其中 Master 端的发送接收均使用 DMA 方式，Slave 端的发送接收均使用查询方式。

从 Log 可以看到，Master 使用 DMA 方式发送数据，Slave 准确收到；Slave 发送数据，Master 使用 DMA 方式也准确接收到。

```
cStart DMA Rx/Tx...
Send data by DMA done (channel=1, data_len=132 bytes).
DMA Interrupt Trigger Count: tfr=2 dsttfr=16, dsttfr_tx=16, err_tx=0

Data sent by DMA:
 0x0000 0x0001 0x0002 0x0003 0x0004 0x0005 0x0006 0x0007
 0x0008 0x0009 0x000a 0x000b 0x000c 0x000d 0x000e 0x000f
 0x0010 0x0011 0x0012 0x0013 0x0014 0x0015 0x0016 0x0017
 0x0018 0x0019 0x001a 0x001b 0x001c 0x001d 0x001e 0x001f
 0x0020 0x0021 0x0022 0x0023 0x0024 0x0025 0x0026 0x0027
 0x0028 0x0029 0x002a 0x002b 0x002c 0x002d 0x002e 0x002f
 0x0030 0x0031 0x0032 0x0033 0x0034 0x0035 0x0036 0x0037
 0x0038 0x0039 0x003a 0x003b 0x003c 0x003d 0x003e 0x003f
 0x0040 0x0041
Data received by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0x0004 0x0005 0x0006 0x0007
 0x0008 0x0009 0x000a 0x000b 0x000c 0x000d 0x000e 0x000f
 0x0010 0x0011 0x0012 0x0013 0x0014 0x0015 0x0016 0x0017
 0x0018 0x0019 0x001a 0x001b 0x001c 0x001d 0x001e 0x001f
 0x0020 0x0021 0x0022 0x0023 0x0024 0x0025 0x0026 0x0027
 0x0028 0x0029 0x002a 0x002b 0x002c 0x002d 0x002e 0x002f
 0x0030 0x0031 0x0032 0x0033 0x0034 0x0035 0x0036 0x0037
 0x0038 0x0039 0x003a 0x003b 0x003c 0x003d 0x003e 0x003f
 0x0040 0x0041

Data received by DMA done (channel=0, data_len=132 bytes):
DMA Interrupt Trigger Count: tfr=0, srctfr=18, srctfr_rx=18, err_rx=0

Data sent by Auxiliary SPI:
 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011
 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011
 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022
 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022
 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033
 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033
 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044
 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044
 0x0055 0x0055

Data received by DMA:
 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011
 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011 0x0011
 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022
 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022 0x0022
 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033
 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033 0x0033
 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044
 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044 0x0044
 0x0055 0x0055
```

测试分析：

使用 DMA 收发 132 字节数据，测试程序中 SPI Burst 传输 2*4=8 字节，无法整除 132，因此 DMA 会先进入 Burst Transaction 传输前 128 字节，再进入 Single Transaction 传输剩下的 4 字节，从现象看收发数据均正常，符合预期。

2.4.7 简单 SPI 传输演示 SPI_SimpleTransmissionDemoCase6()

在主菜单下，输入‘6’命令 进入 Subcase 菜单：

```
+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|    (Just use Target SPI to demonstrate simple transmission)     |
|                                                                 |
|    Input 'A'    Send data, act as Master.                       |
|    Input 'B'    Recv data, act as Slave.                        |
|    Input 'C'    Send data, act as Slave.                        |
|    Input 'D'    Recv data, act as Master.                       |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

2.4.7.1 COB A 作为 Master 发、COB B 作为 Slave 收

测试目的：

演示 2 个板子之间 Master 发数据、Slave 收数据的流程

测试预期：

待测 SPI 可以正常在 2 个板子之间收发数据。

测试现象：

首先将 COB A 与 COB B 的待测 SPI 引脚连接在一起，然后：

COB B 输入‘B’命令，作为 Slave 等待接收数据；

COB A 输入‘A’命令，作为 Master 向 Slave 发送数据。

从 Log 可以看到，Master 端发送数据后，Slave 端收到了字符串："Master Send, Slave Recv."。

```
aSend data, act as Master.

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|    (Just use Target SPI to demonstrate simple transmission)     |
|                                                                 |
|    Input 'A'    Send data, act as Master.                       |
|    Input 'B'    Recv data, act as Slave.                        |
|    Input 'C'    Send data, act as Slave.                        |
|    Input 'D'    Recv data, act as Master.                       |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+


bRecv data, act as Slave...
Data rcvd: "Master Send, Slave Recv."

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|    (Should test with 2 boards: a master and a slave)            |
|                                                                 |
|    Input 'A'    Send data, act as Master.                       |
|    Input 'B'    Recv data, act as Slave.                        |
|    Input 'C'    Send data, act as Slave.                        |
|    Input 'D'    Recv data, act as Master.                       |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

Slave 端程序先进入同步接收流程 SPI_RecvMultiByteSync()：循环检查 Rx FIFO 中是否有数据，一旦检测到即将其读出，当接收到期望长度的数据后，结束接收流程；

Master 端随后开启同步发送流程 SPI_SendMultiByteSync()：循环检查 Tx FIFO 是否已满，若不满则向其填入数据，当填入到期望长度的数据后，等待 FIFO 变为空（作为 Master 的时候，SPI 硬件会自动产生 CLK 信号，并将 FIFO 中的数据以设定好的格式向外发出），随后结束发送流程。

2.4.7.2 COB A 作为 Master 收、COB B 作为 Slave 发

测试目的：

演示 2 个板子之间 Master 收数据、Slave 发数据的流程

测试预期：

待测 SPI 可以正常在 2 个板子之间收发数据。

测试现象：

首先将 COB A 与 COB B 的待测 SPI 引脚连接在一起，然后：

COB B 输入‘C’命令，作为 Slave 等待发送数据；

COB A 输入‘D’命令，作为 Master 接收 Slave 发过来的数据。

从 Log 可以看到，Slave 端发送数据后，Master 端收到了字符串：" Slave Send, Master Recv."。

```
cSend data, act as Slave....

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|    (Should test with 2 boards: a master and a slave)            |
|                                                                 |
|    Input 'A'    Send data, act as Master.                       |
|    Input 'B'    Recv data, act as Slave.                        |
|    Input 'C'    Send data, act as Slave.                        |
|    Input 'D'    Recv data, act as Master.                       |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+

dRecv data, act as Master.
Data rcvd: "Slave Send, Master Recv."

+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|    (Just use Target SPI to demonstrate simple transmission)     |
|                                                                 |
|    Input 'A'    Send data, act as Master.                       |
|    Input 'B'    Recv data, act as Slave.                        |
|    Input 'C'    Send data, act as Slave.                        |
|    Input 'D'    Recv data, act as Master.                       |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
```

测试分析：

Slave 端先进入同步发送流程 SPI_SendMultiByteSync()：循环检查 Tx FIFO 是否已满，若不满则向其填入数据，当填入到期望长度的数据后，等待 FIFO 变为空（作为 Slave，SPI 硬件只有在接收到 Master 端发过来的 CLK 信号的时候，才会将 FIFO 中的数据向外发出），数据发送完成后，结束发送流程。

Master 端随后进入同步接收流程 SPI_MasterRecvMultiByteSync()：向 Tx FIFO 中填入数据，以使 SPI 硬件产生 CLK 信号，然后等待 Tx FIFO 变为空，以保证完整的 CLK 信号已发送至 Slave 端，最后读取 Rx FIFO 中的内容；如此循环直到接收到预期长度的数据，再结束接收流程。

2.4.8 低比特先发送演示 SPI_DataLsbSendRcvTestCase7()

在主菜单下，输入‘7’命令 进入低比特先发送功能测试菜单

测试目的：

测试低比特先发送功能是否正常

测试预期：

低比特功能使能情况下能够准确收发数据

测试现象：

输入‘1’命令，使能发送端 SPI LSB 功能

输入‘A’命令，Target SPI（Master）发送 8 个数据，Auxiliary SPI（Slave）同时接收，发现接收到的数据是移位后的数据。

```
-------+
7

+-----------------------------------------------------------------+
|    Press key to enable lsb send or not:                         |
|    Input '1'    target lsb enable,aux disable.                  |
|    Input '2'    target lsb disable,aux enable.                  |
|    Input '3'    target and aux lsb send enable.                 |
1
+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Motorola SPI Format with SPO=0, SPH=0.          |
|    Input 'B'    Motorola SPI Format with SPO=0, SPH=1.          |
|    Input 'C'    Motorola SPI Format with SPO=1, SPH=0.          |
|    Input 'D'    Motorola SPI Format with SPO=1, SPH=1.          |
|    Input 'E'    TI Synchronous Serial Frame Format.             |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
a
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x8000 0x4000 0xc000 0x3333 0xbbbb 0x7777 0xffff
```

输入‘2’命令，使能接收端 SPI LSB 功能

输入‘A’命令，Target SPI（Master）发送 8 个数据，Auxiliary SPI（Slave）同时接收，发现接收到的数据是移位后的数据。

```
-------+
7

+-----------------------------------------------------------------+
|    Press key to enable lsb send or not:                         |
|    Input '1'    target lsb enable,aux disable.                  |
|    Input '2'    target lsb disable,aux enable.                  |
|    Input '3'    target and aux lsb send enable.                 |
2
+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Motorola SPI Format with SPO=0, SPH=0.          |
|    Input 'B'    Motorola SPI Format with SPO=0, SPH=1.          |
|    Input 'C'    Motorola SPI Format with SPO=1, SPH=0.          |
|    Input 'D'    Motorola SPI Format with SPO=1, SPH=1.          |
|    Input 'E'    TI Synchronous Serial Frame Format.             |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
a
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x8000 0x4000 0xc000 0x3333 0xbbbb 0x7777 0xffff
```

输入‘3’命令，使能发送接收端 SPI LSB 功能

输入‘A’命令，Target SPI（Master）发送 8 个数据，Auxiliary SPI（Slave）同时接收，发现接收到的数据与发送的一致

```
-------+
7

+-----------------------------------------------------------------+
|    Press key to enable lsb send or not:                         |
|    Input '1'    target lsb enable,aux disable.                  |
|    Input '2'    target lsb disable,aux enable.                  |
|    Input '3'    target and aux lsb send enable.                 |
3
+-----------------------------------------------------------------+
|    Press key to test specific function:                         |
|                                                                 |
|    Input 'A'    Motorola SPI Format with SPO=0, SPH=0.          |
|    Input 'B'    Motorola SPI Format with SPO=0, SPH=1.          |
|    Input 'C'    Motorola SPI Format with SPO=1, SPH=0.          |
|    Input 'D'    Motorola SPI Format with SPO=1, SPH=1.          |
|    Input 'E'    TI Synchronous Serial Frame Format.             |
|    Press ESC key to back to the top level case list.            |
+-----------------------------------------------------------------+
a
Send data by Target SPI (should be valid):
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
Receive data by Auxiliary SPI:
 0x0000 0x0001 0x0002 0x0003 0xcccc 0xdddd 0xeeee 0xffff
```

测试分析：

打印与预期均符合，符合预期

第3章 注意事项

1、SPI 的 clock 总是由 master 主动产生，slave 被动接受；Clock 产生条件是 Tx FIFO 中有数据，当数据被发完，clock 信号也随即停止；

2、GPIO 默认是 INPUT_MODE，而使用 MFP 将某根 pin 复用成 SPI 的时候，PINMUX 硬件会自动将 GPIO 的/OE(Output Enable)拉低（生效），于是这根 pin 将既可作输入也可做输出。需要注意的是，如果确实需要 INPUT_MODE 生效（如 master 的 MISO 引脚），还应该 Enable Digital Path（将 DINOFF 对应 bit 清零）；

3、注意 Interrupt test，master send with interrupt，要先 enable SPI，后 enable 中断才 ok，如果反过来，发送的数据是正确的，但接收端收到的是错误的数据；

4、若 dma source/destination 是 memory，则对应的 srctfr/dsttfr 中断应当 disable（enable 无意义，因为 memory 不存在 transaction level 的概念）；

5、若 Master 需要接收数据，则 slave 应在 master clock 过来之前，提前准备好待发送的数据至 FIFO，否则可能导致 Master 接收数据丢失；