SPI

### Introduction

The Serial Peripheral Interface (SPI) is a high-speed synchronous serial input and output (I/O) port that allows a serial bit stream of programmed length (one to 16 bits) to be shifted into and out of the device at a programmed bit-transfer rate. The SPI is normally used for communications between the DSP controller and external peripherals or another controller.

For our case, we use SPI to build communication between DSP and ADC/DAC. The port supports a 16-level, receive and transmit FIFO for reducing CPU servicing overhead.

### *High-Speed SPI Capable GPIOs*

|  |  |  |  |
| --- | --- | --- | --- |
| **SPI pin** | **SPIA** | **SPIB** | **SPIC** |
| SPISIMO | GPIO58 | GPIO63 | GPIO69 |
| SPISOMI | GPIO59 | GPIO64 | GPIO70 |
| SPICLK | GPIO60 | GPIO65 | GPIO71 |
| SPISTE | GPIO61 | GPIO66 | GPIO72 |

### Usage

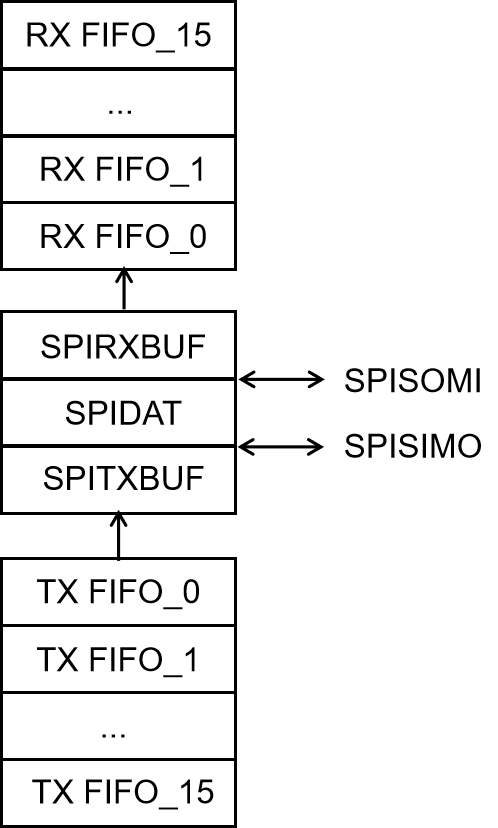
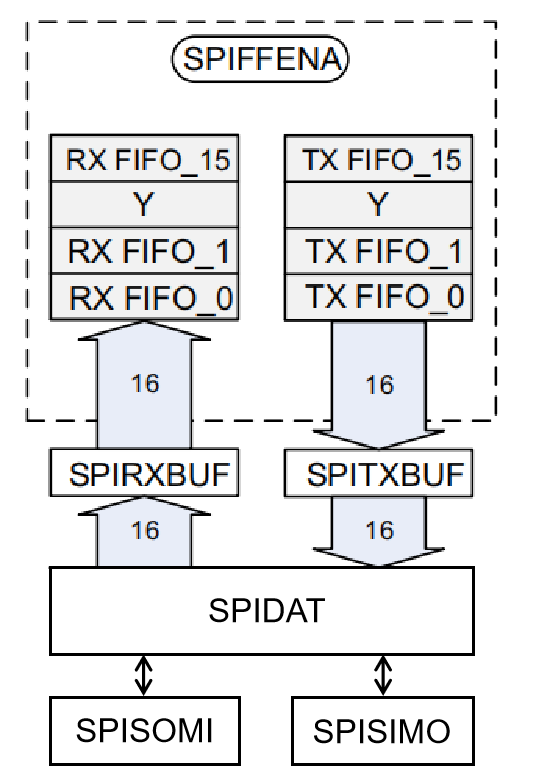
**Master mode**

The SPI can operate in master or slave mode (MASTER\_SLAVE = 1). The MASTER\_SLAVE bit selects the operating mode and the source of the SPICLK signal. The master data transfer by sending the SPICLK signal. For both the slave and the master, data is shifted out of the shift registers on one edge of the SPICLK and latched into the shift register on the opposite SPICLK clock edge. If the CLK\_PHASE bit is high, data is transmitted and received a half-cycle before the SPICLK transition.

We set our device as master. In this case, the SPI provides the serial clock on the SPICLK pin for the entire serial communications network. Data is output on the SPISIMO pin and latched from the SPISOMI pin.

Data written to SPIDAT or SPITXBUF initiates data transmission on the SPISIMO pin, MSB (most significant bit) first. Simultaneously, received data is shifted through the SPISOMI pin into the LSB (least significant bit) of SPIDAT. When the selected number of bits has been transmitted, the received data is transferred to the SPIRXBUF (buffered receiver) for the CPU to read.

### *SPI Module Master Configuration*



**Data format**

The four-bit SPICHAR register field specifies the number of bits in the data character (1 to 16). This information directs the state control logic to count the number of bits received or transmitted to determine when a complete character has been processed.

The following statements apply to characters with fewer than 16 bits:

• Data must be left-justified when written to SPIDAT and SPITXBUF.

• Data read back from SPIRXBUF is right-justified.

• SPIRXBUF contains the most recently received character, right-justified, plus any bits that remain from previous transmission(s) that have been shifted to the left

**High speed and FIFO mode**

In order to achieve the maximum rated speeds, we enabled the high speed mode. And we use SPI FIFO mode.

**Baud rate**

The SPI module supports 125 different baud rates and four different clock schemes. The SPIBRR register determines both the transmit and receive bit transfer rate for the network. SPIBRR can select 125 different data transfer rates. Depending on whether the SPI clock is in slave or master mode, the SPICLK pin can receive an external SPI clock signal or provide the SPI clock signal, respectively. In the master mode, the SPI clock is generated by the SPI and is output on the SPICLK pin, and can be no greater than the LSPCLK frequency divided by 4. The baud rate should be configured to not exceed the maximum rated GPIO toggle frequency.

### *Selection for baud rate*

For SPIBRR = 3 to 127:

For SPIBRR = 0, 1, or 2:

Where:

LSPCLK = Low-speed peripheral clock frequency of the device

SPIBRR = Contents of the SPIBRR in the master SPI device

To determine what value to load into SPIBRR, you must know the device system clock (LSPCLK) frequency (which is device-specific) and the baud rate at which you will be operating.

In our project, LSPCLK = 500kHz, SPI\_BRRA = 5, corresponding to 8.333MHz clock frequency; SPI\_BRRB = 13, corresponding to 3.57MHz clock frequency.

**Clock**

The clock polarity select bit (CLKPOLARITY) and the clock phase select bit (CLK\_PHASE) control four different clocking schemes on the SPICLK pin. CLKPOLARITY selects the active edge, either rising or falling, of the clock. CLK\_PHASE selects a half-cycle delay of the clock. The four different clocking schemes are as follows:

• Falling Edge Without Delay. The SPI transmits data on the falling edge of the SPICLK and receives data on the rising edge of the SPICLK.

• Falling Edge With Delay. The SPI transmits data one half-cycle ahead of the falling edge of the

SPICLK signal and receives data on the falling edge of the SPICLK signal.

• Rising Edge Without Delay. The SPI transmits data on the rising edge of the SPICLK signal and

receives data on the falling edge of the SPICLK signal.

• Rising Edge With Delay. The SPI transmits data one half-cycle ahead of the rising edge of the SPICLK signal and receives data on the rising edge of the SPICLK signal.

|  |  |  |
| --- | --- | --- |
| **SPICLK Scheme** | **CLKPOLARITY** | **CLK\_PHASE** |
| Rising edge without delay | 0 | 0 |
| Rising edge with delay | 0 | 1 |
| Falling edge without delay | 1 | 0 |
| Falling edge with delay | 1 | 1 |

**Registers**

### *SPI Base Address Table*

|  |  |  |  |
| --- | --- | --- | --- |
| **Device Registers** | **Register Name** | **Start Address** | **End Address** |
| SpiaRegs | SPI\_REGS | 0x0000\_6100 | 0x0000\_610F |
| SpibRegs | SPI\_REGS | 0x0000\_6110 | 0x0000\_611F |
| SpicRegs | SPI\_REGS | 0x0000\_6120 | 0x0000\_612F |

### *SPI\_REGS Registers*

|  |  |  |
| --- | --- | --- |
| **Offset** | **Acronym** | **Register Name** |
| 0h | SPICCR | SPI Configuration Control Register |
| 1h | SPICTL | SPI Operation Control Register |
| 2h | SPISTS | SPI Status Register |
| 4h | SPIBRR | SPI Baud Rate Register |
| 6h | SPIRXEMU | SPI Emulation Buffer Register |
| 7h | SPIRXBUF | SPI Serial Input Buffer Register |
| 8h | SPITXBUF | SPI Serial Output Buffer Register |
| 9h | SPIDAT | SPI Serial Data Register |
| Ah | SPIFFTX | SPI FIFO Transmit Register |
| Bh | SPIFFRX | SPI FIFO Receive Register |
| Ch | SPIFFCT | SPI FIFO Control Register |
| Fh | SPIPRI | SPI Priority Control Register |