Multi-Channel Buffered Serial Port – SPI Mode Tutorial

For EEL 4930: Digital Signal Processing

November 5, 2014

J. Marsar

Introduction

This tutorial will serve as a guide to initializing the Multi-Channel Buffered Serial Port (McBSP) peripheral for SPI-mode on TMS320F2833x and TMS320F2823x DSP’s. Please reference the peripheral documentation via TI’s Control Suite or directly from the website at www.ti.com/lit/ug/sprufb7b/sprufb7b.pdf. The McBSP features full-duplex communication and can be interfaced with serial devices. This document will focus on setting up the McBSP in a standard Serial Peripheral Interface (SPI) mode. The device provides two McBSP ports, each with its own controls. The following instructions can be used for port A or B.

McBSP Interface:

The DSP will be initialized as the SPI master, which means it will supply the serial clock and slave enable signals. The table below shows how the McBSP pins are used for SPI communication.

<table>
<thead>
<tr>
<th>McBSP</th>
<th>SPI</th>
<th>DSP GPIO (Port A)</th>
<th>DSP GPIO (Port B)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLKX</td>
<td>SPICLK</td>
<td>22</td>
<td>14/26</td>
<td>Serial clock: All data transfers will run off of this clock. Connect this pin to the clock of the external device.</td>
</tr>
<tr>
<td>DX</td>
<td>SPIMOSI</td>
<td>20</td>
<td>12/24</td>
<td>Data Transmit/Master-Out Slave-In: The DSP will send serial data to the external device on this line.</td>
</tr>
<tr>
<td>DR</td>
<td>SPIMISO</td>
<td>21</td>
<td>13/25</td>
<td>Data Receive/Master-In Slave-Out: The external device will send serial data to the DSP on this line.</td>
</tr>
<tr>
<td>FSX</td>
<td>SPISTE</td>
<td>23</td>
<td>15/27</td>
<td>Frame Sync Slaves Enable: Enables the external device for serial communication. The frame signifies the start and end of each transfer.</td>
</tr>
</tbody>
</table>

• Note that in SPI-mode, the CLKR and FSR signals are not used. Instead, they are tied internally to CLKX and FSX.
• If the external device only supports one-way communication, DX and DR will not both be needed. In the case of a simple Analog to Digital Converter, only the receive pin is used because the DSP does not need to send information to the ADC.
• When initializing the McBSP, the appropriate GPIO pins need to be configured as McBSP pins. See the GPIO register documentation for more information on how to do this.
Initialization Procedure:

1. Before making any changes to the McBSP control registers, make sure that the transmitter, receiver, and sample rate generator are in reset. On power-up, all are already in the reset state. You can find the reset signals in the SPCR1/2 registers. It is also a good idea to clear all of the control registers, especially if you have not reset the DSP since the peripheral was last used. This will ensure that no undesired functionalities occur.

   Note: steps 2-5 can be completed in any order.

2. Configure the McBSP as SPI Master
   - In master mode, CLKX and FSX need to be generated internally. Use FSXM and CLKXM in the PCR register to accomplish this.
   - Set the polarity of FSX by writing to the FSXP bit in the PCR register. Since the external device most likely has a low true enable, the frame synchronization signal should also be active low.

3. For SPI protocol, clock stop needs to be enabled with the desired functionality.
   - In SPCR1, set the CLKSTP bits to 10b or 11b to enable clock stop. The LSB determines whether or not the clock will run with a half-cycle delay. For simplicity, disable this clock delay function by writing 10b.
   - In the PCR register, write CLKXP and CLKRP to the value corresponding to the desired functionality. This will determine whether TX/RX data is sampled on the rising edge or falling edge of the clock signal. To determine these values, refer to the documentation of the external device. You will want to transmit and receive data on opposite edges. Consider a configuration that does not follow this rule: if the DSP transmits on a rising edge and the slave device receives on the rising edge of the same clock, there is no guaranteeing that the read value is correct.
   - See page 62 of the McBSP manual for timing diagrams for the various clock stop modes.

4. Configure the serial clock to the desired rate.
   - Set the CLKSM bit in the SRGR2 register to 1. Since the SCLKME bit in the PCR register is 0, the McBSP will use the low-speed clock of the DSM to generate the serial clock.
   - Write to CLKGDV in the SRGR1 register to set the serial clock frequency. To determine what frequency to run at, refer to the timing specifications for the external device. Use the following formula to select the divide-down value. Remember that LSPCLK is one quarter the speed of the main DSP clock.

\[
 f_{CLKX} = \frac{LSPCLK}{CLKGDV + 1}
\]

   - FSGM in SRGR2 should be set to 0 so a frame-sync pulse is generated when data is transferred into the transmit shift register, XSR1. Writing data to the data transmit register, DXR1/2 will initialize a data transfer.
5. Configure Receive and Transmit functionality.
   • Both the receive and transmit need to have a 1-bit data delay to allow for proper setup time on the FSX signal. Find the XDATDLY and RDATDLY bits in the XCR2 and RCR2 registers, respectively.
   • Determine the word length of the transmit and receive packets. The McBSP peripheral supports word lengths of 8, 12, 16, 20, 24, and 32 bits. This number corresponds to the number of clock pulses that the SPI Master will generate during each frame.
     i. It is very important to look at the external device’s documentation to see how the serial communication needs to be interfaced. Pay attention to how many clock pulses the external device needs. For example, the device may need additional clock cycles for setup time, stop bits, start bits, parity bits, etc. However, the SPI Master will consider all bits as data, so the others must be ignored. Experimentation will likely be necessary to determine which received bits correspond to the data.
     ii. Once the word length has been determined, write the appropriate value to the RWDLEN1 and XWDLEN1 bits in the RCR1 and XCR2 registers, respectively. Since TX and RX are both run off of the same clock, the word length needs to be the same for each. RWDLEN2 and XWDLEN2 may be ignored because they are only used for dual-phrase frames, which are not supported in SPI-mode.
   • Specify the justification of the receive data by writing to RJUST in the SPCR1 register. Since the data will be written into registers that may be larger than the data-length, this setting determines whether the data bits should be left-justified or right-justified; sign-extended or not.

6. Release the sample rate generator from reset by writing a 1 to GRST in SPRCR2. Before continuing, wait at least two sample rate generator clock periods.

7. Bring the transmitter, receiver, and frame sync generator out of reset by enabling the appropriate bits in the SPRCR1/2 registers.

Using the McBSP SPI-Mode:

The McBSP is now ready to send and receive data! To transmit, data needs to be written into the DXR1/2 registers. The data is then automatically moved into the shift register and shifted out on the DX line during a frame-sync pulse. At the same time, received bits will be shifted from the DR line into the receive shift registers, RSR1/2. Once the transfer is over, the data will be moved into the data receive registers, DRR1/2. If the word length is greater than 16-bits, both DXR and DRR registers are used, with #2 containing the most significant bits. In this case, DXR2 must be written to before DXR1 and DRR2 must be read before DRR1.

**Important!** A transmit is required to initialize a receive. Even if the DSP is only reading data from the external device, it must write dummy-data to the data transmit register. After writing to the data transmit register, do not read from the data receive registers until the frame-sync is completed. Either add a short delay or poll the RRDY signal in the SPCR1 register.