TMS320C6000 McBSP Interface to an ST-BUS Device

Jamon Bowen
Shaku Anjanaiah
Digital Signal Processing Solutions

ABSTRACT

This document describes how the multichannel buffered serial ports (McBSPs) in the Texas Instruments TMS320C6000™ digital signal processor (DSP) are used to communicate to a single-rate Serial Telecom (ST)-BUS-compliant device.

The McBSP receives the framing signal, clock, and data from the ST-BUS™ device and processes them to generate internal frame syncs and clocks for correct data reception. The highly programmable features of the McBSP make it easy to interface to ST-BUS signals. This application report focuses on the single rate ST-BUS, wherein the ST-BUS system clock and the data rate (number of bits per second) are equal. Hence, the name “single rate”, which applies only to a 2.048 MHz system clock. The usage of McBSP registers and sample code to perform the above function are described in this document. Project collateral discussed in this application report can be downloaded from the following URL: http://www.ti.com/lit/zip/SPRA511.

Contents

List of Figures

List of Tables

1 Design Problem

How do I use the multichannel buffered serial port in the TMS320C6201 to communicate to a single-rate ST-BUS-compliant device?

2 ST-BUS Requirements

The ST-BUS[1] is a synchronous serial bus with data transfer rates of 2.048, 4.096, or 8.192 Mbps. The interface to an ST-BUS device comprises the clock, frame, and data signals. These signals are available on the McBSP and are programmable, thereby making it a glueless interface.

The ST-BUS data stream comprises of frames with a period of 125 µs, or a frame rate of 8000 frames per sec. This 8 kHz sampling rate (twice the highest signal frequency in order to retain all the information in the stream) corresponds to the 3.5 to 4 kHz voice band frequency. The frame signal indicates the start of a frame and each frame carries blocks of 8-bit data.

The clocks for ST-BUS data can be 2.048, 4.096, 8.192, or 16.384 MHz. Note that these clocks are always twice the data rate, except for 2.048 MHz. Since the 2.048 MHz ST-BUS clock rate can also be the data rate, it is referred to as a single-rate ST-BUS. An example of a double-rate ST-BUS would be a 2.048 Mbps data stream, clocked by a 4.096MHz clock. The following sections describe the hardware and software interface of the C6000 McBSP to a single rate ST-BUS device.[2]

3 McBSP Operation for ST-BUS

The ST-BUS-compliant device that the McBSP is interfacing to, is the master of frames and clock. This means that the ST-BUS device should provide a 2.048 MHz clock, C2, which becomes the external clock source to the McBSP via the CLKS pin. Also, the framing signal, $\overline{FO}$, generated by the ST-BUS device, is used as the receive frame sync (FSR) input to the McBSP. The data transmitted on STo by the ST-BUS device, is received on the DR pin of the McBSP. These connections are shown in Figure 1.

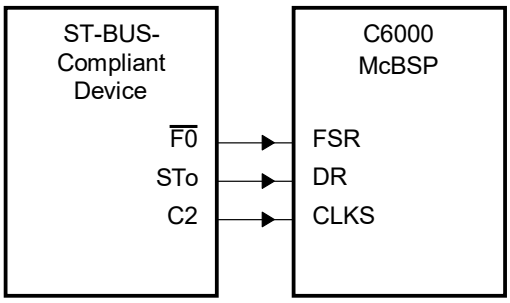
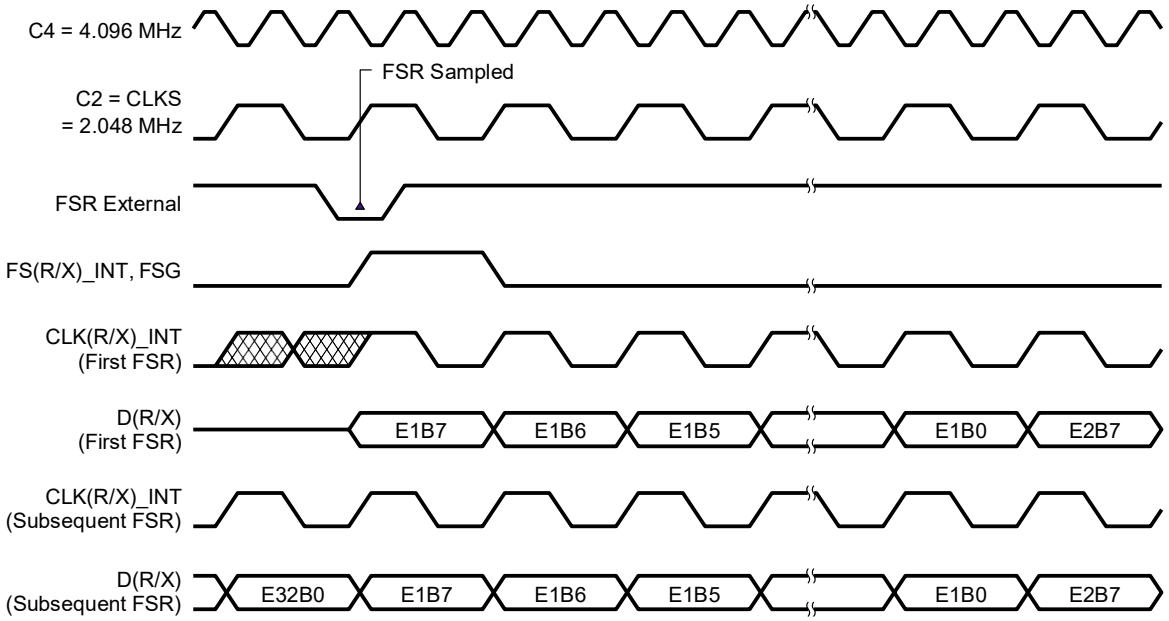


Figure 1. McBSP Connection for 2.048-MHz Single-Rate ST Bus

[1] See Mitel application note MSAN-126, ST-BUS Generic Device Specification (Rev. B).

[2] It is recommended that you be familiar with the features of the C6000 McBSP by reading the *TMS320C6000 Peripheral Reference Guide* (SPRU190), especially section 12.5.5.

In order for the McBSP to recognize an ST-BUS data stream, the GSYNC bit in the sample rate Generator register must be set. The ST-BUS-provided frame sync, $\overline{FO}$,is an active-low signal that is low for one-half of the CLKS period. The GSYNC bit causes the external frame sync that arrives on FSR to be sampled on the rising edge of CLKS and, in turn, generates an internal frame sync in the McBSP that is active-high for one CLKS clock period. This internal frame sync, FSR_int, is used as the reference for data reception. This is shown in Figure 2.



ExBy = Element x Bit y

Figure 2. Single-Rate ST-BUS Example

In the 2.048 MHz single-rate ST-BUS example shown in Figure 2, the data stream comprises 32 elements of 8 bits each, in each frame. Each new frame starts with a new frame sync signal, $\overline{FO}$. Since the McBSP receiver does not know when the *first* frame will arrive on FSR, and therefore does not know when to get out of reset and receive data, special interrupts should be used. This is made easy with the new frame sync interrupt that is available on the McBSP, and which works even when the receiver is in reset. The receive CPU interrupt (RINT) can be programmed to detect a new frame sync pulse, after which the CPU can safely take the receiver out of reset. Further initialization details are discussed in section 5 of this document.

For the case of a double-rate ST-BUS clock, 4.096, 8.192, and 16.38-MHz clocks are supported with data rates equal to half of the ST-BUS clock rate. The ST-BUS clock drives the CLKS pin of the McBSP. The frame sync provided by the ST-BUS ($\overline{FO}$) is active-low for one ST-BUS clock period, and drives the FSR pin of the McBSP. This external frame sync is sampled by the McBSP on the falling edge of CLKS to generate the required internal frame sync. The timing diagram for a double rate ST-BUS clock is shown in Figure 3.

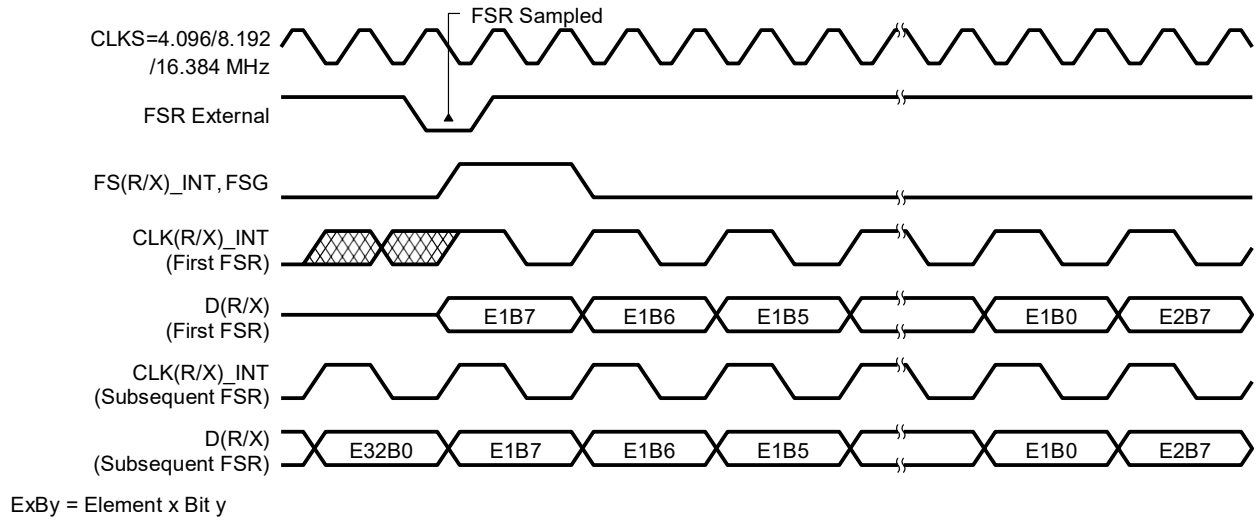


Figure 3. Double-Rate ST-BUS Timing Diagram

The McBSP register setup for a double-rate operation is the same as the single rate, with the following exceptions:

- CLKGDV = 1, so that the data rate is half the clock rate (CLKS).
- CLKSP = 1 ensures that the internal clocks CLKG, CLKR_int, CLKX_int are generated on the falling edge of CLKS; the same edge that generates the FSG, FSR_int, and FSX_int.

4 McBSP Register Configuration

As shown in Figure 1, FSR, CLKS, and DR are inputs.

- **Framing Signal:** The polarity of the incoming frame sync signal ($\overline{FO}$) must be inverted to provide the necessary active-high input signal to the McBSP. The external frame sync pulse dictates the arrival of a new frame; therefore, the frame period (FPER) and frame width (FWID) is not used/programmed.

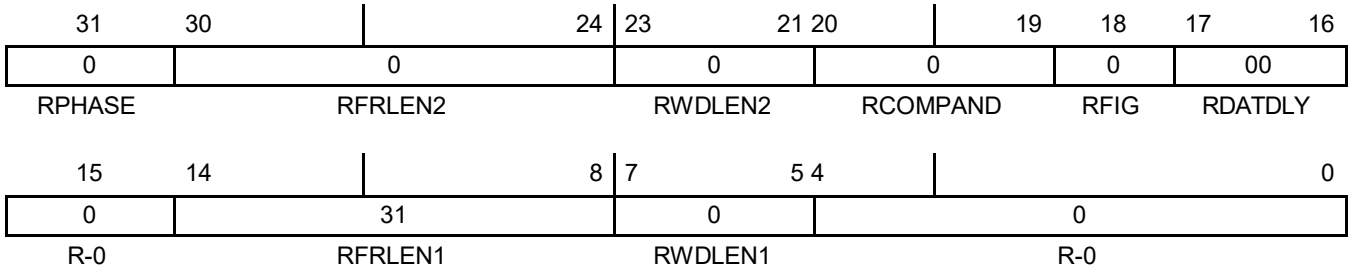
  Although $\overline{FSR}$ is treated as input, the FSRM bit in PCR and $\overline{FRST}$ bit in SPCR must be set to 1. The $\overline{FRST}$ bit must be set to enable the frame sync signal generation. FSRM = 1 indicates that the internally generated FSR_int will be used to detect the arrival of data, but will not be output on FSR pin because GSYNC = 1 disables the output buffer.

  Since it is a single-phase frame, with each frame comprising of 32 elements with 8 bits each, FRLEN1 = 31 and WDLEN1 = 0.

- **Data Delay:** Since there is no delay between the arrival of the first bit of data and the generation of internal frame sync FSR_int, the receiver should be set up for a data delay of zero.

- **Clocks:** Although CLKR pin of the McBSP is not used in this set up, it is necessary to configure it as an output. Another important parameter is the polarity of CLKS signal. The CLKS polarity determines the edge that samples the incoming frame sync signal and also the edge that generates internal clocks CLKG, CLKR_int, CLKX_int, and internal frame-sync signals FSG, FSR_int, and FSX_int. For the single-rate ST-BUS case, the rising edge of CLKS does the above function.

The various bit settings for the above requirements are shown in Figure 4 through Figure 7 and Table 1.



Legend: R = Read

Figure 4. Receive Control Register (RCR)

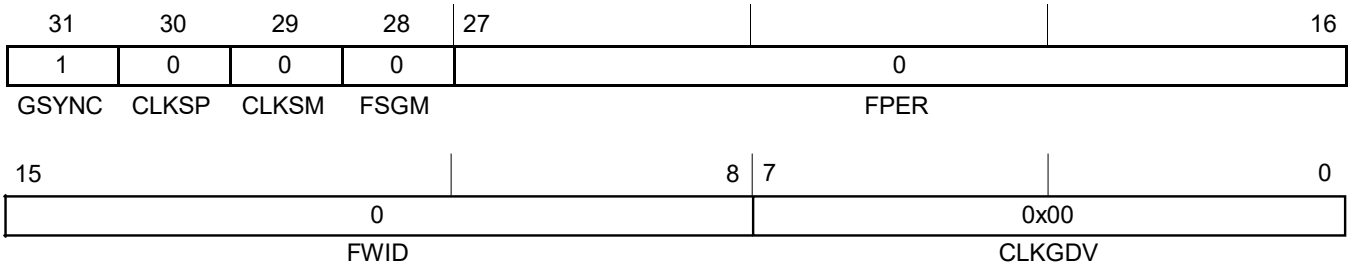
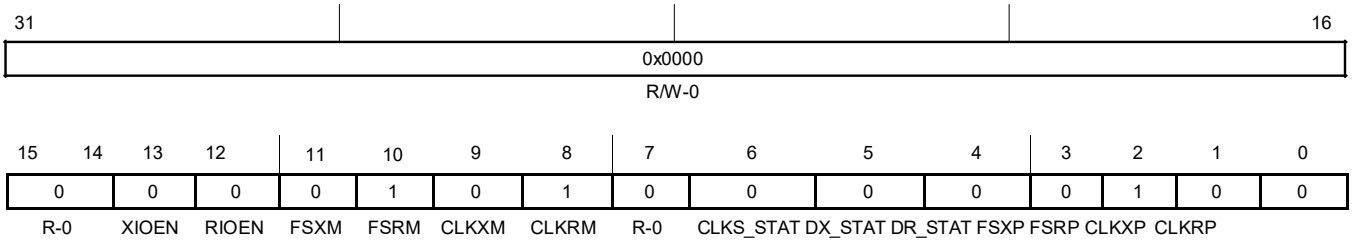
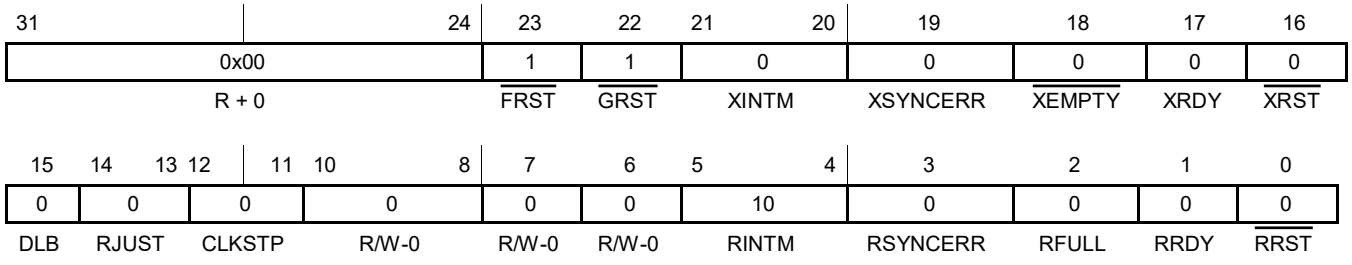


Figure 5. Sample Rate Generator Register (SRGR)



Legend: R = Read; R/W = Read/write

Figure 6. Pin Control Register (PCR)



Legend: R/W = Read/write

Figure 7. Serial Port Control Register (SPCR)

Table 1. Bit Field Values for McBSP Registers

Register (Bit Field No.)	Bit Field Name	Value (in Binary) Slave (Receiver)
RCR[17:16]	RDATDLY	0 (default)
SPCR[5:4]	RINTM	10b
SPCR[23]	$\overline{\text{FRST}}$	1
SPCR[22]	$\overline{\text{GRST}}$	1
SRGR[31]	GSYNC	1
SRGR[29]	CLKSM	0 (default)
SRGR[7:0]	CLKGDV	0 (default)
PCR[10]	FSRM	1
PCR[8]	CLKRM	1
PCR[2]	FSRP	1

5 McBSP Initialization

Typical applications use the (E)DMA to service the McBSP. Please refer to *TMS320C6000 McBSP Initialization* (SPRA488) to program the McBSP control registers and (E)DMA registers for proper serial port operation. In addition to this, the following step is required since the new frame sync interrupt is used to wake up the McBSP.

1. After the (E)DMA has been started, the first frame sync that arrives on FSR will wake up the receiver. This is done in the ISR corresponding to CPU_INT15. The interrupt service routine (ISR) should also disable this interrupt, so that subsequent frame syncs do not cause unnecessary enabling of the receiver that has already been taken out of reset.

When the next frame sync arrives, the receiver provides the read sync event to the (E)DMA, which causes transfer of data from DRR to the specified destination address. The receiver continues to receive data until the required number of frames has been received.

6 Sample Code Setup

The example code included with this document was tested on two McBSPs (McBSP0 and McBSP1) on a TMS320C6211 device. The block diagram of this test setup is shown in Figure 8.

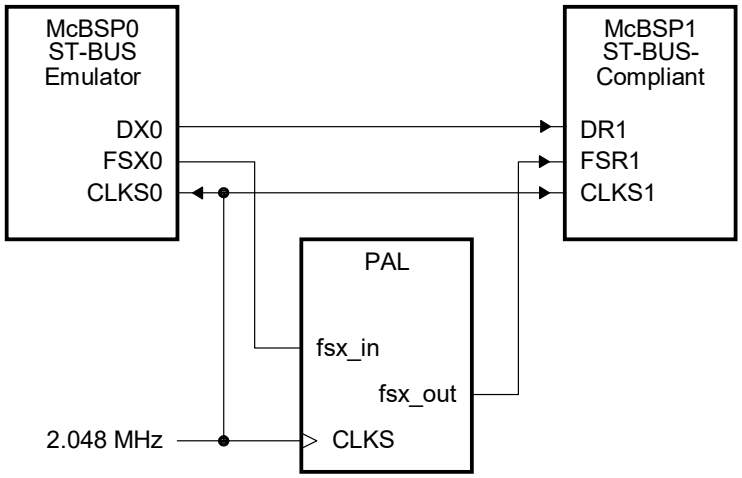


Figure 8. ST-BUS Emulator and McBSP

McBSP0 is configured as the ST-BUS transmitter (master), which provides the frame sync and data. The ST-BUS clock in this example is 2.048 MHz. The PAL uses FSX0 (fsx_in) to generate the active-low frame sync signal, fsx_out, at the appropriate rising edge of CLKS 0/1. Signal fsx_out is equivalent to FSR_ext signal, shown in Figure 2. The PAL equations in VHDL are included with this document. The resulting signals, due to the above set up, are shown in Figure 9.

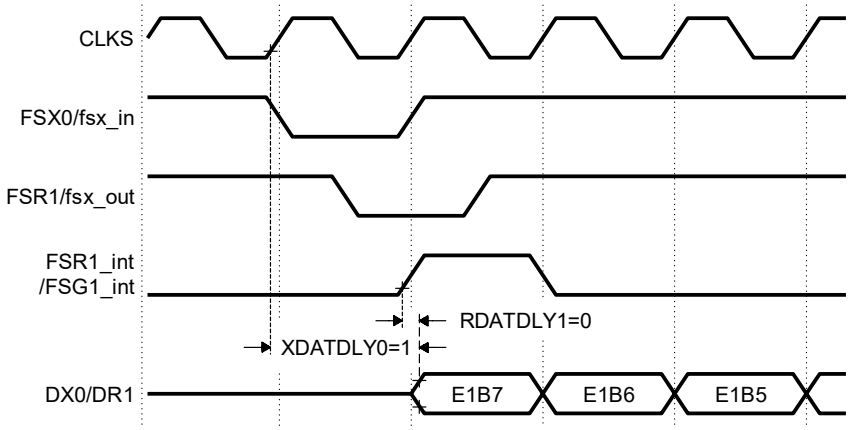


Figure 9. Timing Diagram for Example Single-Rate ST-BUS Setup

7 References

1. Mitel application note, MSAN-126, ST-BUS Generic Device Specification (Rev. B).
2. *TMS320C6000 Peripherals Reference Guide* (SPRU190).
3. *TMS320C6000 McBSP Initialization* (SPRA488).

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265