



Technical notes on using Analog Devices DSPs, processors and development tools
 Contact our technical support at dsp.support@analog.com and at dsptools.support@analog.com
 Or visit our on-line resources <http://www.analog.com/ee-notes> and <http://www.analog.com/processors>

Using ADSP-BF561 Blackfin® Processor Flag Pins to Emulate the PCM Interface

Contributed by Shailendra Miglani

Rev 1 – March 23, 2005

Introduction

The PCM interface is a serial interface used to transfer speech data. This application note discusses how to emulate the PCM interface using four programmable flag pins (GPIOs) of the ADSP-BF561 Blackfin® processor. This application note applies to designs in which all of the serial interfaces present on the devices are in use and hence the need for emulation of the PCM interface arises.

Basics of the PCM Protocol

In the PCM mode of operation, there are four signals (PCM_CLK, PCM_SYNC, PCM_OUT, and PCM_IN).

There are three modes of operation for PCM:

- (1) Short frame sync mode
- (2) Long frame sync mode
- (3) Multi-slot mode

The timing and description of all three modes are discussed below.

In short frame sync mode (Figure 1), the falling edge of PCM_SYNC indicates the start of the PCM word. PCM_SYNC is always one clock cycle long. In the next rising edge following the frame sync, data is driven out.

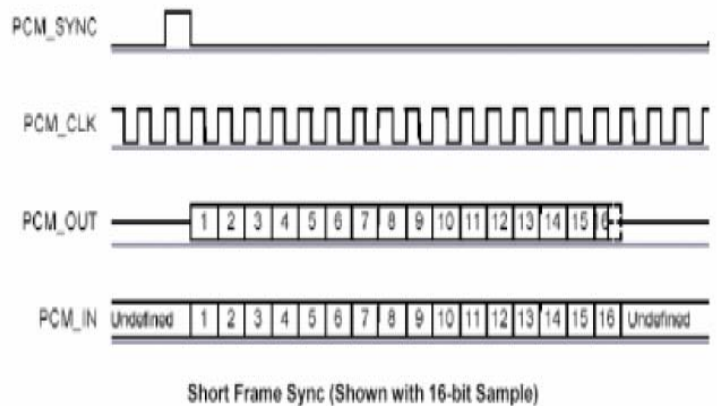


Figure 1. Short Frame Sync Mode

The timing for the long frame sync mode is shown in Figure 2:

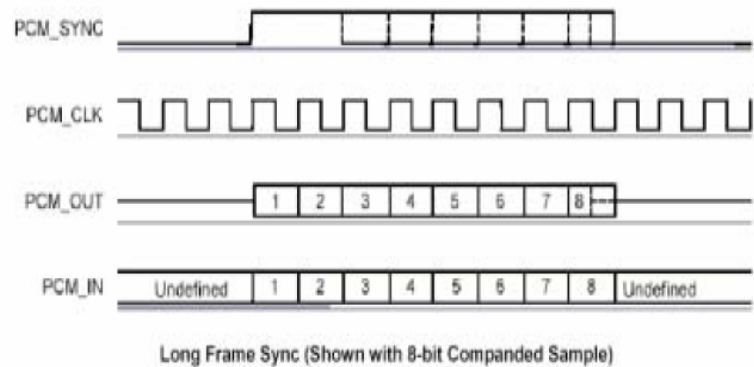


Figure 2. Long Frame Sync Mode

In long frame sync mode, the rising edge of PCM_SYNC indicates the start of the PCM word.

The timing for the multi-slot operation is shown in Figure 3.

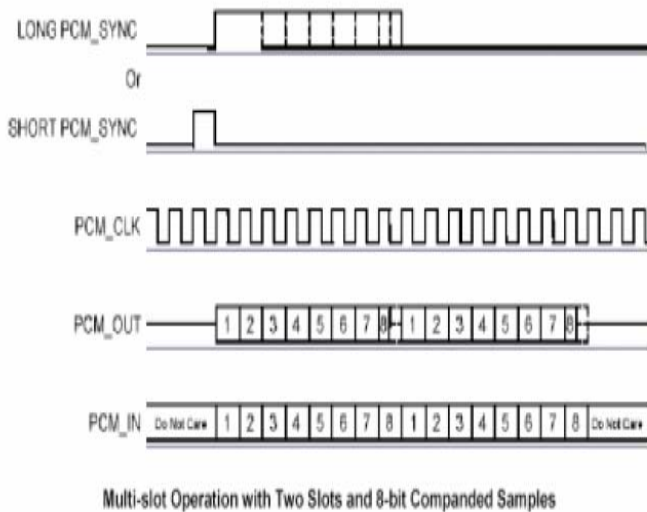


Figure 3. Multi-Slot Mode

In multi-slot mode, after receiving a short frame sync or a long frame sync, the transmitter sends out three words of data.

The programmable flag-based PCM interface is implemented for transmit as well as receive. The Serial Port of ADSP-BF561 is used to verify the logic for both transmit and receive.

When the flag pins transmit, the SPORT receives the data with the external clock and frame sync. When the flag pins receive the data, the SPORT transmits with the internal clock and frame sync.

Short and Long Frame Sync Implementation

The PCM interface can be emulated using four programmable flags. This implementation uses PF9, PF10, PF11, and PF12. PF9 is used as the clock. PF10 is used as the frame sync. PF11 is used as the data receive, and PF12 is used as the data transmit. For data receive, the data is sent through the SPORT and the data is received through the flags. In transmit mode, the data is transmitted through the SPORT and is received through the programmable flags.

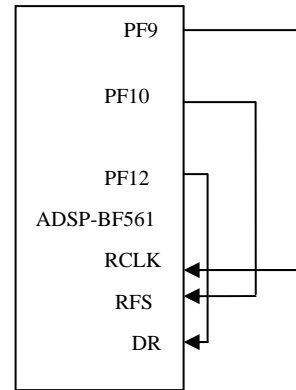


Figure 4. ADSP-BF561 Interface to SPORT (Flag in Transmit Mode) for Long and Short Frame Sync

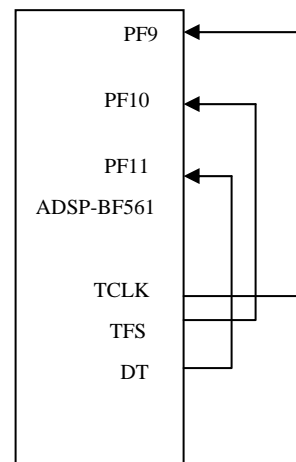


Figure 5. ADSP-BF561 Interface to SPORT (Flag in Receive Mode) for Long and Short Frame Sync

Transmit in Short and Long Frame Sync

In this mode, the core timer is used to generate the interrupt at a rate twice that of the desired clock rate, since the flag pins have to toggle twice per clock cycle. In this mode, the PF9, PF10, and PF11 flag pins are configured as outputs.

In short frame sync mode, when a timer interrupt is generated, the code branches to the ISR and PF9 is toggled inside the ISR to generate the clock. Eight-bit data transfers are assumed in this mode. The frame sync is asserted at the rising edge of the clock as per the specification of the PCM interface. The frame sync will be active for one clock cycle. After the frame sync is deasserted and on the immediate next rising

edge, the data is driven out bit by bit for one byte. Figure 6 shows timing for short frame sync mode.

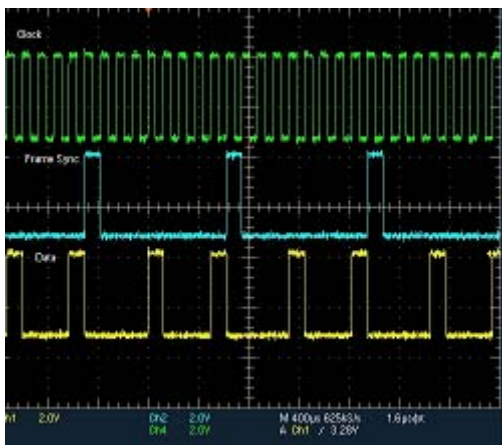


Figure 6. Waveforms of Clock (Green), Frame Sync (Green) and Data (Yellow) for Short Frame Sync

In long frame sync mode, when the timer interrupt is asserted the code branches to the ISR and starts toggling PF9 to generate the clock. Eight-bit data transfers are assumed in this scenario. The frame sync is asserted at the rising edge of the clock as per the specification of the PCM interface. After frame sync is asserted, in the same rising edge, the data is driven out bit by bit for one byte. Figure 7 shows timing for a long frame sync. The frame sync is active throughout the data transfer.

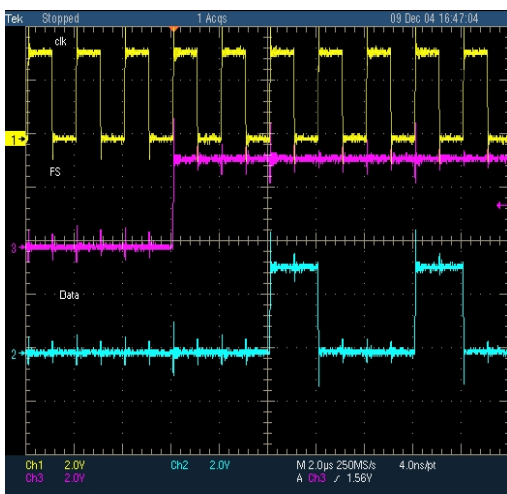


Figure 7. Waveforms of Clock (Yellow), Frame Sync (Violet) and Data (Green) for Long Frame Sync

To validate whether the logic has been implemented correctly, the data is received through the SPORT1 in early frame sync mode for short frame sync (and late frame sync mode for long frame sync).

Receive in Short and Long Frame Sync

In the receive mode, flag pins are connected to the Serial Ports and the data is transmitted through the SPORT interface and received through the PCM interface realized through the flag pins. In receive mode; PF9, PF10, and PF12 are configured as inputs. In the implementation, PF9 is configured as edge sensitive for rising edge, and the interrupt is enabled for the same. As soon as the clock is applied, the code branches into the interrupt service routine (ISR) and frame sync assertion is checked. Inside the ISR, if the frame sync is not shown asserted, the code continuously branches to the ISR, but no data is driven in.

In short frame sync mode, as soon as the frame sync is asserted and then deasserted (as mentioned earlier, this is checked inside the ISR of PF9) from the Serial Port, the data is driven in bit by bit by reading PF11. The width of the data is also eight bits.

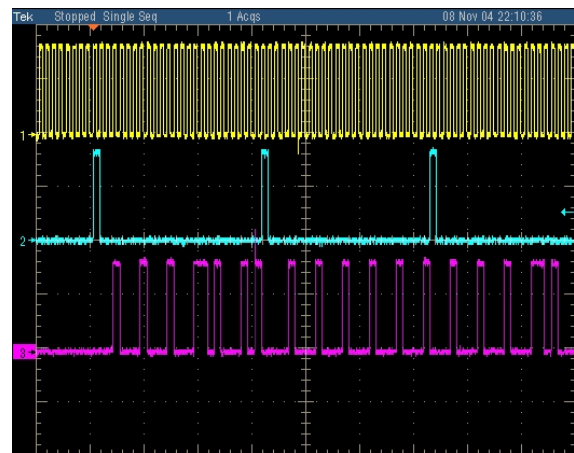


Figure 8. Waveforms of Clock (Yellow), Frame Sync (Green) and Data (Violet) for Multi-Slot Mode

In long frame sync mode, as soon as the frame sync is asserted from the Serial Port, the data is driven in bit by bit by reading PF11. The width of the data is eight bits.

Transmit and Receive in Multi-Slot Mode

For multi-slot mode, the SPORT interface supports a minimum of eight channels. In the present implementation, early frame sync is considered. In this mode, after the frame sync, three channels (words) are transmitted by the GPIOs, and three channels (words) are received by the SPORTs.

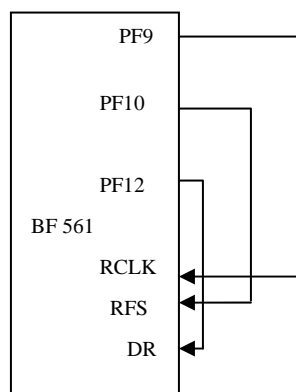


Figure 9. ADSP-BF561 Interface to SPORT (Flag in Transmit Mode) for Multi-Slot Mode

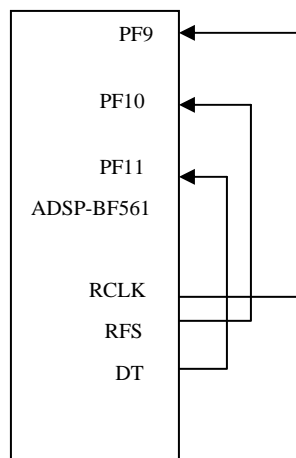


Figure 10. ADSP-BF561 Interface to SPORT (Flag in Receive Mode) for Multi-Slot Mode

In receive mode, when the SPORT is transmitting and the programmable flags are

receiving, the SPORT is transmitting three of the eight channels, and the programmable flags are receiving three channels. In this scenario, for there is no problem receiving data, since the logic of the flag pins are implemented in such a way that after the frame sync, it will receive three words and then wait for the next frame sync and then drive three data words. Hence, there is no problem for the logic as far as five inactive slots out of eight are concerned, since after receiving three data words, it will wait for the next frame sync.

The logic has been verified by transmitting a set of data by SPORT and receiving the same set by the flag pins.

Hence, for transmit, after getting the frame sync followed by the falling edge of frame sync, three words are transmitted. For the ADSP-BF561, a minimum of eight words are expected by the SPORT interface.

Hence, to verify that the logic is working correctly, three words are sent and then three words are received and then stopping with SPORT receiving in multichannel mode. The transmitted data is viewed on scope plots to verify the data transmission.

Clock Frequency

When the SPORT is transmitting the data, the clock rate is specified by the SPORT1_TSCCLKDIV register. For transmit, the clock is generated using flags. After generating the timer interrupt, inside the ISR the flag is toggled twice to generate the clock.

For example, to generate an 8 kHz clock rate, the interrupt for the timer should be generated at 16 kHz. If the ADSP-BF561 EZ-KIT Lite® board is running at a core clock frequency of 300 MHz and generating an interrupt at 16 kHz, the timer should be programmed to generate an interrupt for 18750 cycles. In a similar fashion, the timer parameters should be configured for the desired clock rate.

Test Case Implementation

Three test cases are attached to this application note:

- (1) Short frame sync mode
- (2) Long frame sync mode
- (3) Multi-slot mode

Each test case includes one transmit code example and one receive code example.

The transmit code example generates internal frame sync and clock using flags, and the receive code example takes the external frame and clock.

Each test case is verified by using SPORTs.

Conclusion

PCM interface has been successfully realized using the flag pins of ADSP-BF561 processor.

References

- [1] *ADSP-BF561 Blackfin Processor Hardware Reference*. Rev 0.2, Analog Devices, Inc.
- [2] *ADSP-BF561 Blackfin Embedded Symmetric Multiprocessor Data Sheet*. Rev PrE, Analog Devices, Inc.
- [3] *ADSP-BF561 EZ-KIT Lite Manual*. Rev 1.1, Analog Devices, Inc.

Document History

Revision	Description
<i>Rev 1 – March 23, 2005 by Shailendra Miglani</i>	Initial Release