

# Interfacing a Blackfin DSP to High-Speed Converters for Wireless Applications

by Jeff Sondermeyer [jeff.sondermeyer@analog.com],  
Jeritt Kent [jeritt.kent@analog.com],  
Martin Kessler [martin.kessler@analog.com], and  
Rick Gentile [richard.gentile@analog.com]

## INTRODUCTION

In the 1970s and '80s high-speed mixed-signal designs were most frequently constrained by the limitations of digital circuitry, not analog. As an example, high-speed parallel converters (>10 MSPS) have been available from industry leaders like Analog Devices, Inc. (ADI), since the 1970s. Now, high-resolution data is being handled at higher sample rates (for example, 14 bits at >50 MSPS) by both analog-to-digital and digital-to-analog converters (ADCs and DACs). In addition, more and more applications are demanding intensive real-time algorithms. These factors mandate faster programmable general-purpose (GP) digital signal processors (DSPs) to handle the challenges presented by high-speed data rates.

Until recently, most designers had to interface high-speed parallel converters to application-specific ICs (ASICs) or fast *field-programmable gate arrays* (FPGAs). Devices like these are capable of resolving the many required simultaneous parallel digital operations; but they are often inflexible and can be prohibitively expensive. Now, with the recent introduction of Blackfin™ DSPs, such as the ADSP-21535, users have available a programmable general-purpose 16-bit fixed-point vector DSP—with a 300-MHz-capable core—that can handle the sustained input/output (I/O) and core throughputs required to process data from the many available high-speed converters. Depending on the core clock frequency, a maximum *system clock* (SCLK) of 133 MHz can be achieved. [This SCLK should not be confused with the serial clock for the *serial peripheral interface* (SPI).]

## Why Choose a General-Purpose DSP?

GP DSPs typically cost much less than their closest digital-processing counterparts—FPGAs and ASICs—and they are easily programmed. In addition, since GP DSP design cycles are much shorter, time to market can be faster. With FPGAs/ASICs, users must often hire or consult professionals with specialized design skills. They may even be required to send their intellectual property (IP) out-of-house, incurring risks to the confidentiality of hardware, firmware, and software. On the other hand, GP DSP code can be stored in read-only memory (ROM) or masked into a DSP (such as members of the ADSP-2153x family), which further protects IP. Finally, GP DSPs are fully programmable, in contrast to ASIC implementations, where every change requires a costly redesign.

(time and money). These factors easily motivate many engineers to consider GP DSP as the solution of choice, especially when core rates can approach those of “Pentium®-class” chips.

The ADSP-21535, the first member of ADI's Blackfin family, was designed to work optimally in a computer-bus environment, while newer designs available soon (within the year) will have a parallel peripheral interface (PPI), specifically designed to work with I/O data. In the interim, however, the ADSP-21535's power can be made available for use in urgently needed designs, such as wireless applications, by using it with a small amount of readily available external circuitry.

What are the issues? In general, to guarantee sufficient data-processing bandwidth, the DSP needs a minimum clock speed an order of magnitude ( $10\times$ ) faster than the converter's sample rate. In turn, the amount of processing bandwidth needed depends upon the DSP's interface capabilities, which are in turn influenced by several other factors. These considerations include: block processing versus sample processing, the existence of a *direct memory-access* (DMA) controller, multi-ported memory, and whether external FIFOs are used. Fortunately, the ADSP-21535 has a full DMA controller that operates independently of the core, with multi-ported level-1 (L1) and level-2 (L2) memories. The combination of core speed, an independent DMA controller, and a large multi-ported on-board memory (308 Kbytes), allows the ADSP-21535 to perform efficient block processing at high data rates. For example, if the Revision 2.2-compliant, 33-MHz, 32-bit (4 bytes) peripheral component interconnect (PCI) interface is used (not shown in this application), transfer bandwidths can be achieved that approach 132 MB/s.

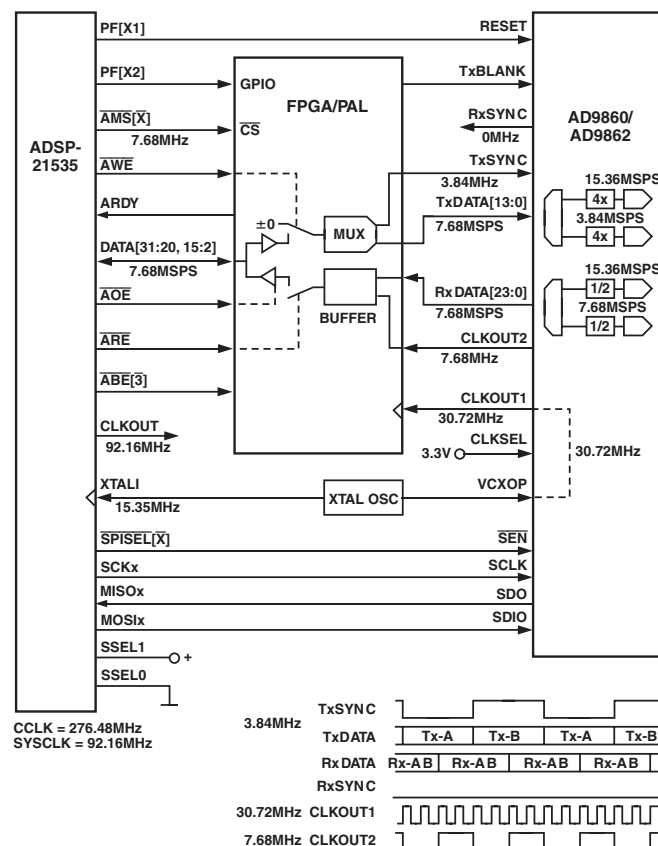


Figure 1. External logic connections between the ADSP-21535 and the AD9860/AD9862.

The ADSP-21535's *external bus interface unit* (EBIU) provides interfaces to asynchronous (ASYN) external memories. If the PCI bus must be used for other system communications, the EBIU is the only available parallel interface to connect the ADSP-21535 to a high-speed converter. To combine the DSP-mastered, asynchronous control of this port with the synchronous, continuous data stream of converters may pose somewhat of a challenge for a system designer.

This article describes one particular hardware implementation, utilizing low-pin-count, low-cost, commonly available "glue logic" devices, such as a *programmable array logic* chip (PAL), a *complex programmable logic device* (CPLD), or an FPGA. This logic performs the control functions between the AD9860/AD9862 Mixed Signal Front End (MxFE™) and the ASYN external memory bus of the ADSP-21535. The application depicted in Figure 1 is for an *orthogonal frequency-division multiplexed* (OFDM) wireless portable terminal. The ADC and DAC are time-shared (*time-division multiplexed*, or TDM) over the ASYN interface of the DSP. (The information given here applies equally to other parallel high-speed ADCs and DACs.)

An Engineering Note\* is available describing the details of the interconnection scheme. It assumes that the reader has information on hand about both the ADSP-21535 and the AD9860/AD9862, including the "ADSP-2153x/ADSP21535 Blackfin™ DSP Hardware Reference" and the data sheet for the AD9860/AD9862. These can be found at <http://www.analog.com>.

### Design Goals

One of the early design goals for this project was to minimize the amount of external control logic necessary to interface the DSP and the converter(s). Driven by cost, Engineering wanted to eliminate any FIFOs or memory within the external logic device. An additional constraint was to avoid routing the data buses through the logic, thereby reducing the number of pins, package size, and cost of the logic device. The initial design shown in Figure 1 combines all functions (including data latching) into a single logic device. However, production models of this design will utilize inexpensive tri-state-able latches driven by a logic device. These latches or buffers will multiplex (pack) the samples from the DSP memory interface to the 12-/14-bit DAC as well as buffer or demultiplex (unpack) the 10-/12-bit ADC samples to the DSP memory interface.

### Design Challenges

One of the key factors in any mixed-signal/DSP design is a solid understanding of the constraints and consequent trade-offs between the devices. The following discussion will illustrate the various trade-offs that must be considered when interfacing ADCs/DACs to the ADSP-21535.

Some of the major design constraints are:

- The OFDM modulation scheme for this design drove a required converter sample rate of 15.36 MSPS.
- The AD9860/AD9862 has a dual 10-/12-bit, 64-MSPS ADC and a dual 12-/14-bit, 128-MSPS DAC.
- Unlike SHARC® processors, which have a DMA request and DMA grant (i.e., DMA can be mastered from external devices),

\*"Interfacing the ADSP-21535 to High-Speed Converters (like those on the AD9860/AD9862) over the External Memory Bus." A PDF of this file and a ZIP file of software code for EE-162 can be found on the DSP Application Notes Page under EE-162.

<<http://www.analog.com/library/applicationNotes/dsp/applicationNotes.html>>.

the ADSP-21535 has only one set of internal memory DMA channels (memDMA), which must be mastered from the DSP.

In addition, when the ADSP-21535 ASYN interface is connected to devices that do not contain FIFOs or memory, all latencies must be thoroughly understood. For example, every time the memDMA relinquishes the bus after a burst of eight transfers, it requires ten SCLK cycles to begin the next transfer.

Future Blackfin family members will have programmable priority levels for the DMA controller, as well as a dedicated high-speed parallel interface—with DMA request and DMA grant signaling. With a dedicated PPI, these future Blackfin products will not require the ASYN memory interface to connect with parallel converters.

The approach used here assumes that the memory interface is dedicated to the converters. Multiplexing external SRAM/SDRAM memory with the converter(s) would be difficult and is not recommended, especially considering that there is only one memDMA, and it would need to be shared. The existence of a large on-board L2 memory (256 Kbytes) minimizes the need for any external memory. However, it is permissible to multiplex the parallel converter(s) with a Flash or EPROM for the initial booting process.

This design uses a TDM time-slice approach for sharing the external bus between the ADCs and the DACs, because simultaneous access is not possible here, since the single memory interface does either a read or a write—and there is only one set of memDMA channels (source and destination).

The ADSP-21535 will support a maximum SCLK of 133 MHz (peak DMA bandwidth). At this rate, and with no external FIFO, the memDMA could sustain a transfer (32-bit word) rate of 133 MSPS/10 (nine cycles are required for bus acquisition and one for next transfer), or 13.3 M words/s. However, the SCLK of the ADSP-21535 is derived from the core clock (CCLK). CCLK in turn is generated via the PLL divider, whose available ratios are 1 to 31—and there are only four available divide ratios: 2.0, 2.5, 3.0, and 4.0. So one possible combination of CCLK and divisor that will allow a 133-MHz SCLK is CCLK = 266 MHz and CCLK/SCLK = 2. But if the core must run at 300 MHz, as in this application, the highest SCLK that can be obtained is 120 MHz (divisor = 2.5) to stay under the maximum 133 MHz.

Now, since the ASYN memory interface is 32 bits wide, up to two 16-bit samples (in this case I and Q) can be packed into each word. This effectively halves the word rate that the DSP must process. (With a 15.36-MSPS converter sample rate, the DSP will "see" 7.68 MSPS.) The highest external converter sample rate that the memDMA will support under these conditions is  $2 \times 120/10 = 24$  MSPS. Furthermore, the SCLK must be an integer multiple of the converter sample rate to ensure proper phase alignment between converter timing and DSP timing and eliminate the need for any external FIFOs. Therefore, the highest converter sample rate that the ADSP-21535 will support at a 300-MHz core rate is  $2 \times 120/10 \text{ M} = 24$  MSPS—or twice the memDMA rate, as discussed in Commandment #10 (at the end of this article). Since the DSP will only process the packed data at half this rate, 12 MSPS is the maximum rate that the memDMA can sustain, i.e., 12 M words/second. Higher sample rates can be processed by the ADSP-21535 if small external FIFOs are included between the converter(s) and the EBIU.

**Table 1. Possible parameter scenarios for the ADSP-21535.**

Converter Sample Rate (MSPS)	CCLK (MHz)	CCLK/SCLK Divide Ratio	SCLK (MHz)	memDMA (Mwrites/s)	SCLK/Converter Sample Rate
15.360*	276.48	3.0	92.16	9.216	6
24.000	300	2.5	120	12.0	5
26.600	266	2.0	133*	13.3	5
20.000	300*	3.0	100	10.0	5
15.000	300*	4.0	75	7.5	5
10.000	300	2.5	120	12.0	12
8.8670	266	2.0	133*	13.3	15
0.8867	26.6	2.0	13.30	1.33	15*
CSR <= 26.6	CCLK < 300	2.0, 2.5, 3.0, or 4.0	SCLK < 133	memDMA < 13.3	5 <= integer <= 15

\*Denotes driving parameter

Recall now that OFDM requirements dictated a 15.36-MSPS converter sample rate. To obtain a SCLK that is an integer multiple of this converter sample rate, one must choose a *phase-locked-loop* (PLL) multiplier that is an integer multiple of one of the four available divisor ratios (2.0, 2.5, 3.0, or 4.0). With a PLL multiplier of 18, the maximum CCLK allowed is 276.48 MHz. This in turn limits the SCLK to an integer multiple of 3, because  $276.48/3 = 92.16$  MHz. (A divide ratio of 2 would give an SCLK over the 133-MHz maximum.) Under these constraints, the maximum sustained rate that the memDMA can support is  $92.16/10 = 9.21$  M words/s.

#### DMA Considerations

Careful consideration must be given to the combined, required, “sustained” DMA performance. Since the memDMA is a shared resource over the DMA bus (DAB), other DMA activity is arbitrated on this bus. This application requires a 10-Mbit/s serial channel on a *serial port* (SPORT) that also must arbitrate for the DAB. This will consume an additional 625 K words/s at 16 bits/word of DMA bandwidth. The ADSP-21535 can support a maximum of 133 M words/s (peak) DMA bandwidth, and the SPORT has higher arbitration priority over the memDMA (see Table 1). So the SPORT DMA should effectively utilize the above-mentioned ten-cycle delay and allow most, if not all, of the 9.21 M words/s to be used by the memDMA. There are  $9.21 \text{ M} - 15.36 \text{ M}/2 (= 1.53 \text{ M})$  words/second of additional bandwidth, which should provide enough margin to sustain a 7.68 MSPS rate.

**Table 2: Arbitration Priority**

DAB Master	Arbitration Priority
SPORT0 RCV DMA Controller	0—highest
SPORT1 RCV DMA Controller	1
SPORT0 XMT DMA Controller	2
SPORT1 XMT DMA Controller	3
USB DMA Controller	4
SPI0 DMA Controller	5
SPI1 DMA Controller	6
UART0 RCV Controller	7
UART1 RCV Controller	8
UART0 XMT Controller	9
UART1 XMT Controller	10
Memory DMA Controller	11—lowest

Analysis of the DMA engine within the ADSP-21535 reveals a few other considerations. While the DMA engine supports two types of DMA transfers—descriptor-based and autobuffer-based—the memDMA controller does not support autobuffer-based DMA. Therefore, descriptor-based transfers must be used. The descriptor fetch from L1/L2 memory involves two 5-word block moves, one for the source descriptor and another for the destination descriptor. In addition, the memDMA has a 16-entry 32-bit FIFO that is filled from the source and emptied from the destination. If both descriptors are loaded simultaneously, 39 SCLK cycles (worst case) are required from L2. The destination descriptor load has priority over the source load to avoid overrunning the FIFO. Thus, in this example, the amount of time required to load both descriptors simultaneously is  $(1/92.16 \text{ M}) \times 39 = 423 \text{ ns}$ . The DMA engine descriptor load performs best when the descriptors are loaded from L2 memory. If the descriptors are located in L1 memory, there are additional delays. The worst-case source-plus-destination descriptor load time from L1 is 65 SCLK cycles. To process data effectively at these sample rates, ping-pong buffers are normally used. (In this design, two 1024-word buffers are utilized.) This technique allows data to be filled into one buffer while the core processes the other buffer. As a reference, the complete VisualDSP++™ 2.0 project program is available from ADI.

There are two phases of operation that must be analyzed: Samples must be received by the DSP from the ADC (receiver TDM phase); and samples must be transmitted from the DSP to the DAC (transmitter TDM phase).

#### Receiver TDM Phase

During the receiver phase, data moves thus: ADC→EBIU→(source)→memDMA→FIFO→L1/L2 (destination). At the 15.36-MHz converter sample rate, a new 32-bit sample arrives at the DSP every  $1/7.68 \text{ M} = 130.2 \text{ ns}$ . As seen from the descriptor load-time latency, 423 ns, something must be done to avoid overrunning the DSP and losing samples. Fortunately, the converters are attached to an external bus, and the address bus is not being used. Thus, when moving samples into the DSP, one can set up the source descriptor with the maximum transfer count, 65536 words, and destination descriptor with intended ping-pong buffer transfer size, 1024 words. In this way, upon receiving an interrupt from the core every 1024 words, only the destination descriptor is reloaded, and the load time is reduced to  $20 \text{ SCLKs} \times 1/92.16 \text{ M} = 217 \text{ ns}$ . As noted, this design uses a TDM scheme in which the ADC and DAC occupy individual



time slices. The multiplex rate is a variable 5 ms to 8 ms. Since the ADC and DAC data are interleaved, at a worst case, the interface changes from receiver to transmitter and back every 8 ms. Therefore,  $65536 \text{ words} \times 130.2 \text{ ns}$ , or 8.5 ms, is sufficient time, and the source descriptor only needs to be set up once at the beginning of each receiver TDM phase. Finally, the 16-entry memDMA FIFO “hides” the destination descriptor load time, because the source is still filling the FIFO while the destination descriptor is being loaded from memory. In a worst-case scenario, the memDMA FIFO will accumulate only a few samples of data before the descriptor is reloaded. Then, these samples are burst into memory. So the need for an external FIFO on the receiver side is eliminated, and no samples are lost.

### Transmitter TDM Phase

During the transmit phase (data to the DAC), data movement is in the opposite direction: L2/L1 (source)→memDMA→FIFO→EBIU (destination)→DAC. Unlike the receiver mode, the source descriptor must be updated every 1024 words. This will require 20 SCLK cycles, or 217 ns. However, since the memDMA (9.21 M words/s) is running slightly faster than the sample rate (7.68 M words/s), this should maintain 16 samples in the memDMA FIFO, which will feed the DAC while the descriptor loads. The destination descriptor transfer count can be fixed at 65536 words. Again, no external FIFO is required, and no samples are lost.

### LOGIC OVERVIEW AND TIMING

In avoiding the need for FIFOs in the external logic, it is still important to synchronize the converter clocks to the DSP system clock, SCLK. This limits the available ADSP-21535 clocking

options in accordance with the sample rate. Minimally, SCLK must be evenly divisible by the converter sample rate, and CCLK may need to be evenly divisible by the converter sample rate as well. (There is only one non-integer divisor, 2.5—it may not be usable in some cases). External latches or buffers must be used to align the data from the converters with the timing of the DSP (see Figures 2 and 3 for sample skew and delay). The four-wire DSP SPI port is directly connected to the AD9860/AD9862 SPI port. To ensure proper power sequencing and initialization, the DSP should reset the converter(s). To further reduce the pin count of the external logic, another option available on the AD9860/AD9862 (not shown here) allows two 10-/12-bit ADC values to be time-multiplexed onto a single 10-/12-bit RXDATA bus. While this would eliminate one of the two 10-/12-bit buses, it requires the external logic to demultiplex the data before it is transmitted to the DSP.

All data movement is controlled or mastered by the memDMA within the DSP. When the ADC data is read (see Figure 2), the external logic must drive the data and the ARDY signal. The external logic must sample the /AOE pin to check when data can be driven to the ADSP-21535. The /AOE signal indicates to the external logic that the DMA controller is ready to take data. The receiver three-state machine is shown at the bottom of the figure.

When data is being sent out to the DAC (see Figure 3), the external logic has to sample the /AWE signal and then drive ARDY. /AWE indicates to the external logic when the DMA controller is ready with new data. The transmitter four-state machine is shown at the bottom of the figure.

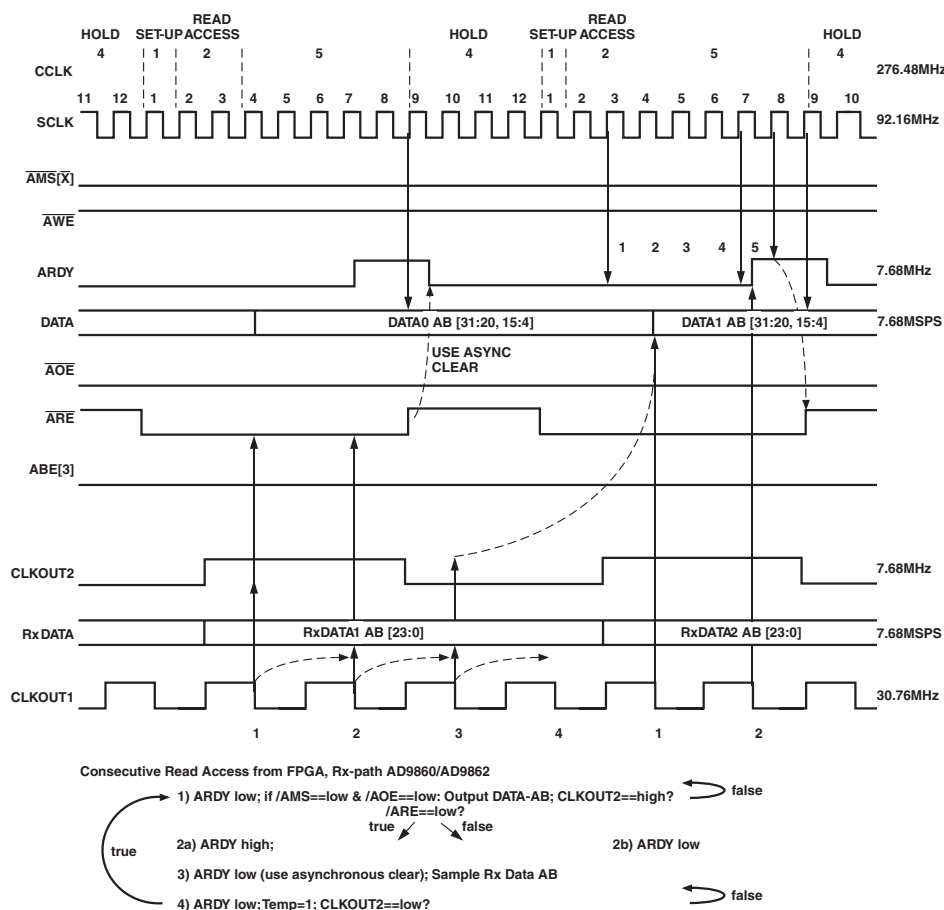


Figure 2. Receive timing and state machine.

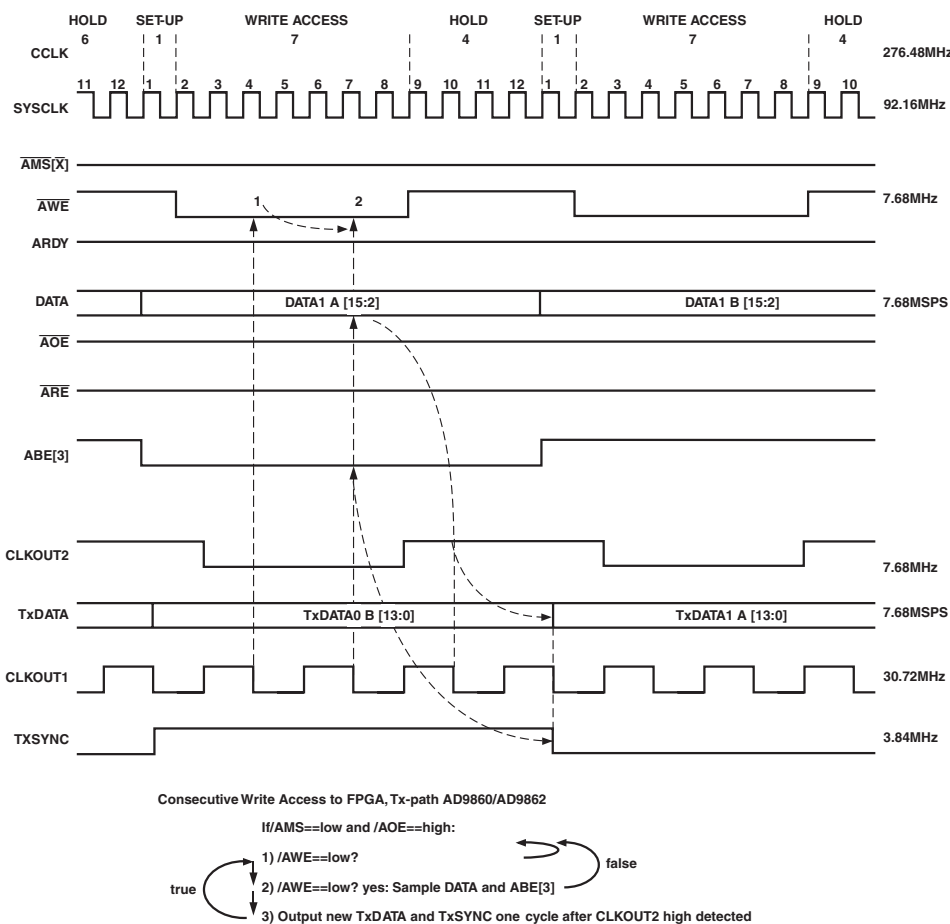


Figure 3. Transmit timing and state machine.

## CONCLUSIONS

Even though the ADSP-21535 was not specifically designed to interface to high-speed parallel converters, it is available *now*, with all its many other advantages, for designs requiring rapid time-to-market. New-generation devices, such as the ADSP-21532, which has a dedicated parallel peripheral interface (PPI), will arrive soon to provide more-definitive solutions in such applications at lower cost. In order to fill an urgent need, we're suggesting a low-cost "FIFO-free" solution that can be used until the next generation parts are available in production quantities. This will allow today's 300 MHz ADSP-21535 to interface to ADCs and DACs with sample rates of up to 24 MSPS. If the ADSP-21535 core can be clocked specifically at 266 MHz, the highest converter sample rate is limited only by the maximum SCLK that the ADSP-21535 can support (133 MHz) and the inter-burst 10-cycle memDMA latency:  $2 \times 133 \text{ M}/10 = 26.6 \text{ MHz}$ .

The following set of interfacing rules, or "ten commandments," will help in optimizing performance of the ADSP-21535 when used with high-speed converters:

### The ADSP-21535's Ten Commandments

1. The ADSP-21535's maximum allowed core frequency is 300 MHz.
2. The ADSP-21535's maximum allowed system clock (SCLK) is 133 MHz.
3. The ADSP-21535's memDMA has a worst-case 10-cycle reacquisition latency every time the DMA bus is relinquished.

4. Derive the ADSP-21535's SCLK from CCLK; there are four available divide ratios between CCLK and SCLK (2.0, 2.5, 3.0, and 4.0).
5. If the ADSP-21535's core runs at the maximum (300 MHz), the maximum SCLK is 120 MHz. See Commandments 1, 2, and 4.
6. The maximum ADSP-21535 memDMA rate is  $133 \text{ M}/10 = 13.3 \text{ M words/s}$ . See Commandments 2 and 3.
7. In order to obtain the fastest ADSP-21535 SCLK (133 MHz) and memDMA transfer rate (13.3 M words/s), the core must not run at maximum, but at 266 MHz, with a CCLK/SCLK divide ratio of 2. This utilizes all but 34 MIPS.
8. When not using external FIFOs, the ADSP-21535's core should operate at a minimum rate an order of magnitude ( $10\times$ ) greater than the greatest external interfacing converter sample rate to provide sufficient processing bandwidth.
9. To eliminate the need for external FIFOs, the ADSP-21535's SCLK should be an integer multiple of the external interfacing converter sample rates to ensure proper phase alignment between converter timing and DSP timing.
10. To halve the sample rate that the DSP has to process, the external logic should pack up to two 16-bit samples into each 32-bit word. In fact, the maximum converter sample rate is twice the memDMA rate if two 16-bit words are packed into one 32-bit word. 