Interfacing TI Clocked FIFOs With TI Floating-Point Digital Signal Processors

First-In, First-Out Technology

SCAA005A
March 1996

TEXAS INSTRUMENTS

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.

Contents

Title *Page*

Introduction **5**

DSP Applications Using FIFOs **5**

Communication Between a TI Bidirectional Clocked FIFO and a TI Floating-Point DSP **7**
DMA Considerations 10
Example Control of the SN74ACT3632 Using the DMA 11

Programming the FIFO Almost-Full-Flag and Almost-Empty-Flag Offsets **16**
Calculating FIFO Flag-Offset Values 16

Hardware Interface **17**
Read and Write Cycles 18
Interrupt Generation 19

Conclusion **20**

List of Illustrations

Figure *Title* *Page*

1 Clocked FIFOs Used for High-Speed Data Acquisition 6
2 Clocked FIFOs Used in Pipelined Image-Processing Systems 7
3 Bidirectional Clocked FIFO Used for Bus-Speed Matching 7
4 TMS320C31 DSP 8
5 SN74ACT3632 512 × 36 × 2 Bidirectional FIFO 9
6 TMS320C31 Memory Map in Microprocessor Mode 10
7 Routine for Almost-Empty Flag CPU Interrupt to Schedule FIFO Reads 12
8 Scheduling FIFO Writes 13
9 Routine for DMA Interrupt to CPU When Transfer Counter Reaches Zero 15
10 Almost-Full-Flag and Almost-Empty-Flag Offset Selection 17
11 TMS320C31-40 Interface to an SN74ACT3632-30 FIFO 17
12 TMS320C31-40 Read-Read-Write Timing Diagram With an SN74ACT3632 512 × 36 × 2 FIFO 18
13 TMS320C31 Interrupt Generation by FIFO2 Almost-Empty Flag 19
14 TMS320C31 Interrupt Generation by FIFO1 Almost-Full Flag 19

Introduction

Digital signal processors (DSPs) are used in a variety of applications to analyze real-time data or speed computationally intensive tasks. A DSP is a microprocessor tuned to the task of number crunching by an instruction set that conveniently ties together special hardware components needed for fast floating-point and fixed-point math and by powerful input/output (I/O) functions that keep data flowing quickly. Design of the I/O for a digital-signal-processing system is one of the major factors that dictates the machine's performance. First-in, first-out (FIFO) memories often are used as data rate buffers to optimize the throughput of digital-signal-processing systems and increase overall performance.

A FIFO is a dual-port memory with built-in write and read addressing to pass out data in the same order it is written. Data reads and writes can be done asynchronous to one another. Flag circuitry indicates when the queue is empty or full, preventing simultaneous read/write access to the same memory location. Advanced FIFO memories from Texas Instruments (TI) produced in CMOS or BiCMOS technology also have user-programmable almost-empty and almost-full flags to measure the number of words in memory. FIFOs provide a seamless bridge between two buses operating at different clock speeds and acting as temporary data bins to exchange information between two systems without handshaking delay.

TI's TMS320C3x and TMS320C4x processors are popular DSPs that include a 40-/32-bit floating-/fixed-point math unit, one or two 32-bit external buses, and an on-board direct-memory-access (DMA) controller. Unidirectional and bidirectional clocked FIFO devices from TI frequently are used to support systems built around these processors. Attractive features offered by TI clocked FIFOs are synchronous (clocked) interface on each port, asynchronous I/O capability, programmable flags, maximum write/read frequencies up to 80 MHz, maximum read access times as low as 9 ns, and fine-pitch surface-mount packaging.

DSP Applications Using FIFOs

DSP systems doing real-time data analysis or control functions use analog-to-digital (A/D) converters to translate continuous-time, real-valued signals into discrete-time, integer-valued sequences. The rate used to sample the analog signal is chosen based on the frequency bandwidth of the signal. This sample rate is independent of the microprocessor-bus rate, and asynchronous buffering is required to pass the information to the DSP. Serial ports on the TMS320C3x/C4x processors provide an asynchronous interface with A/D converters and are adequate when the incoming data traffic has a relatively low bit rate. For higher bit rates, unidirectional clocked FIFOs provide a parallel buffer between the converters and the DSP bus.

Figure 1 shows several digitized signals, each using a FIFO for rate buffering to the processor bus. An example of this application is multiplexing several analog telephone lines for compression or symbol detection. An input signal packet is gathered in the FIFO and burst into memory by the DMA unit on the TMS320C3x/C4x. This method also is useful when the analog data is sampled at a high rate for short duration, as in some medical-imaging equipment. Each FIFO holds its A/D samples in queue until the processor retrieves the information that must be completed before the next sampling period. The block labeled FIFO Enable can have a single-memory-space address and control the FIFOs in round-robin fashion as the DMA fills the random-access memory (RAM) with digitized signals.

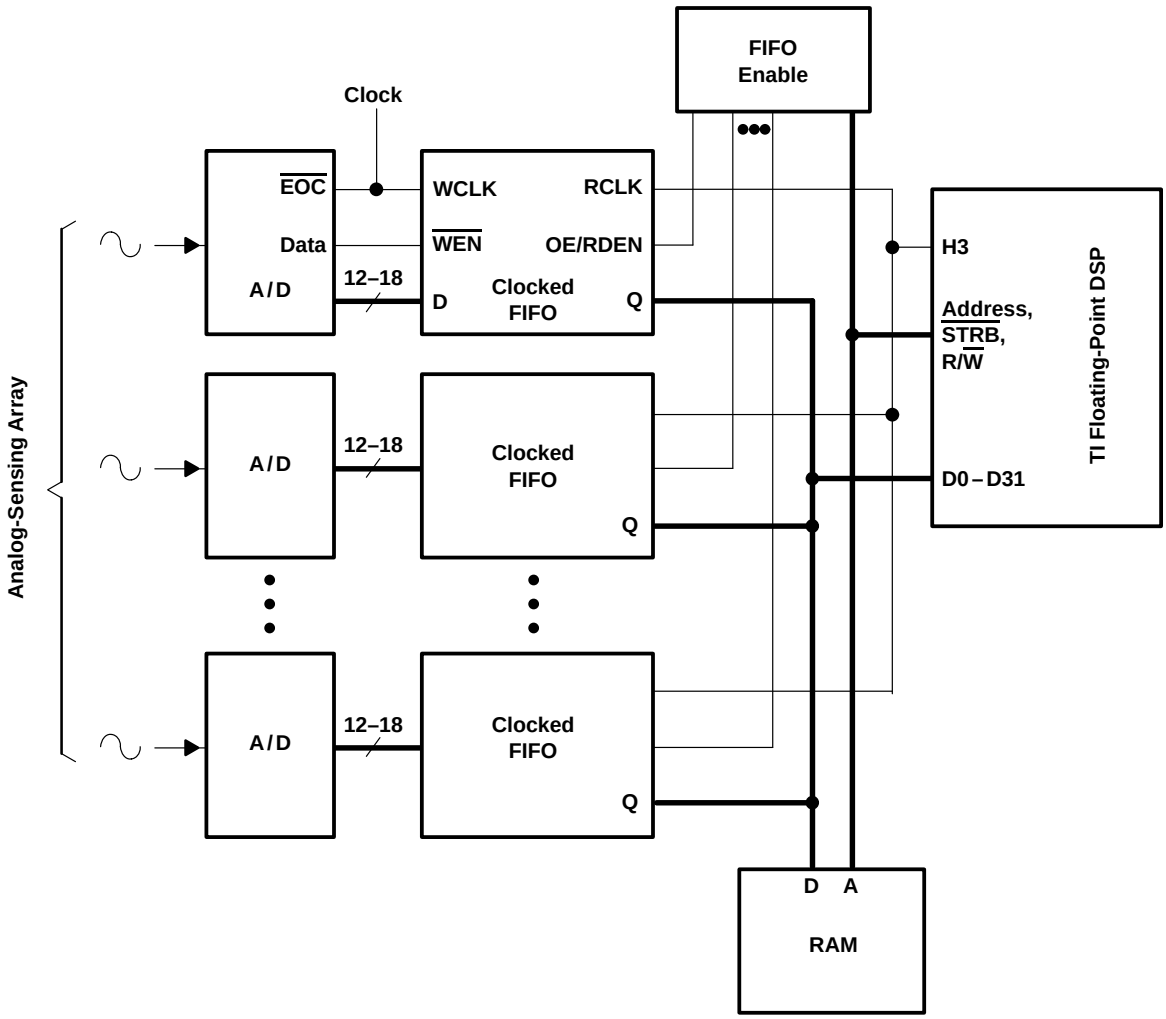


Figure 1. Clocked FIFOs Used for High-Speed Data Acquisition

DSP algorithms are drawn as functional boxes with interconnecting lines representing data streams. This concept can translate directly to a hardware organization as shown in the pipelined image-processing system (see Figure 2), wherein the unidirectional TI clocked FIFOs provide the data connection between the TI floating-point DSPs. The FIFO that connects the bus to the first processing element (PE) performs the task of rate matching, as the bus generally operates at a slower rate than the DSP bus. FIFOs that connect adjacent PEs are used as packet builders; that is, a packet of data is stored and then detected with the use of the almost-full/almost-empty or half-full flags and read by the next processor. Transferring a known packet size simplifies DMA control. The FIFO interconnect between PEs eliminates the need for processor interlock protocols and reduces clock-distribution requirements by allowing each PE to utilize its own independent clock.

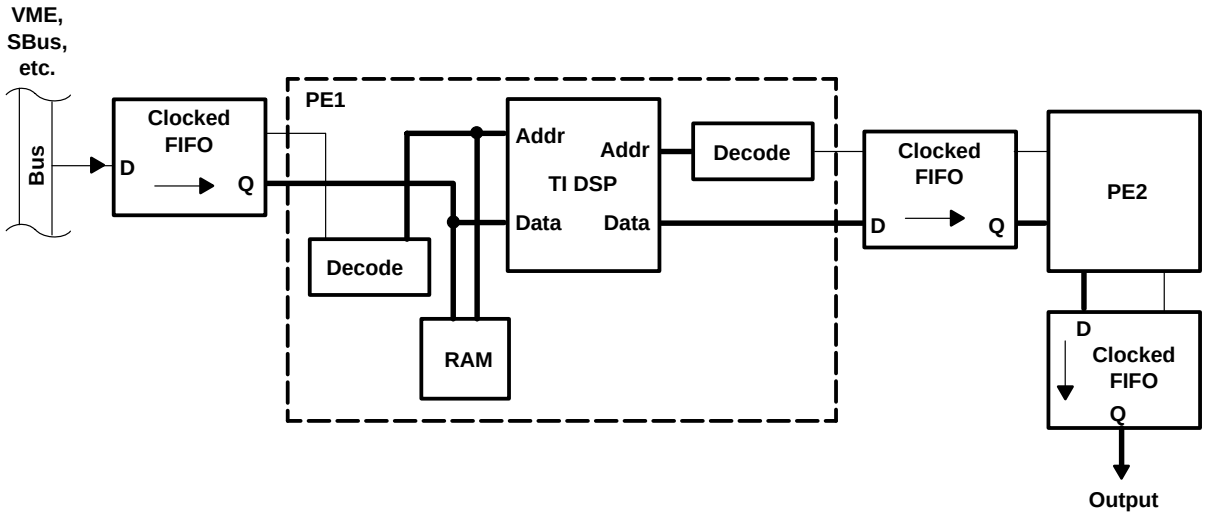


Figure 2. Clocked FIFOs Used in Pipelined Image-Processing Systems

Software applications often are written for a general-purpose workstation platform, incorporating signal-processing functions that are not efficiently performed by the workstation. A solution to this problem is to use a special-purpose DSP system as a coprocessor for the application and communicate with the host computer via a local or backplane bus (see Figure 3). The bidirectional clocked FIFO is most useful when data traffic is heavy both to and from the host computer, such as when the host provides the input data and receives the processed results. Bidirectional FIFOs also can be used as instruction queues between a host processor and the DSP. The FIFO in the datapath provides clock partitioning so each bus can operate at its maximum rate; it also eliminates transfer delay required for a bus request to be granted to either the host or the DSP.

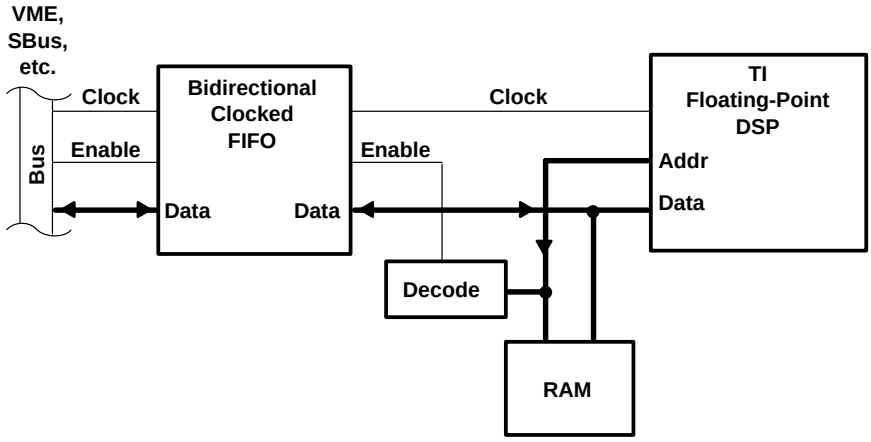


Figure 3. Bidirectional Clocked FIFO Used for Bus-Speed Matching

Communication Between a TI Bidirectional Clocked FIFO and a TI Floating-Point DSP

An interface between a TI floating-point DSP and a TI clocked FIFO was created as an example of FIFO memory mapping, DMA considerations, flag-offset programming, and bus-cycle control. The processor chosen for the example was the TMS320C31-40 (see Figure 4) since its functions and terminals are a subset of the TMS320C30 and comparable to the TMS320C40. The FIFO chosen was the SN74ACT3632, which is a bidirectional device that contains two 512-word by 36-bit FIFOs to buffer data in opposite directions (see Figure 5). A single SN74ACT3632 device in either a 132-pin quad flat package or 120-pin thin quad flat package provides a 32-bit bidirectional datapath. The clocked architecture of the FIFO simplifies the interface by directly using many of the DSP bus-control signals. The bidirectional function provides both data read and write examples.

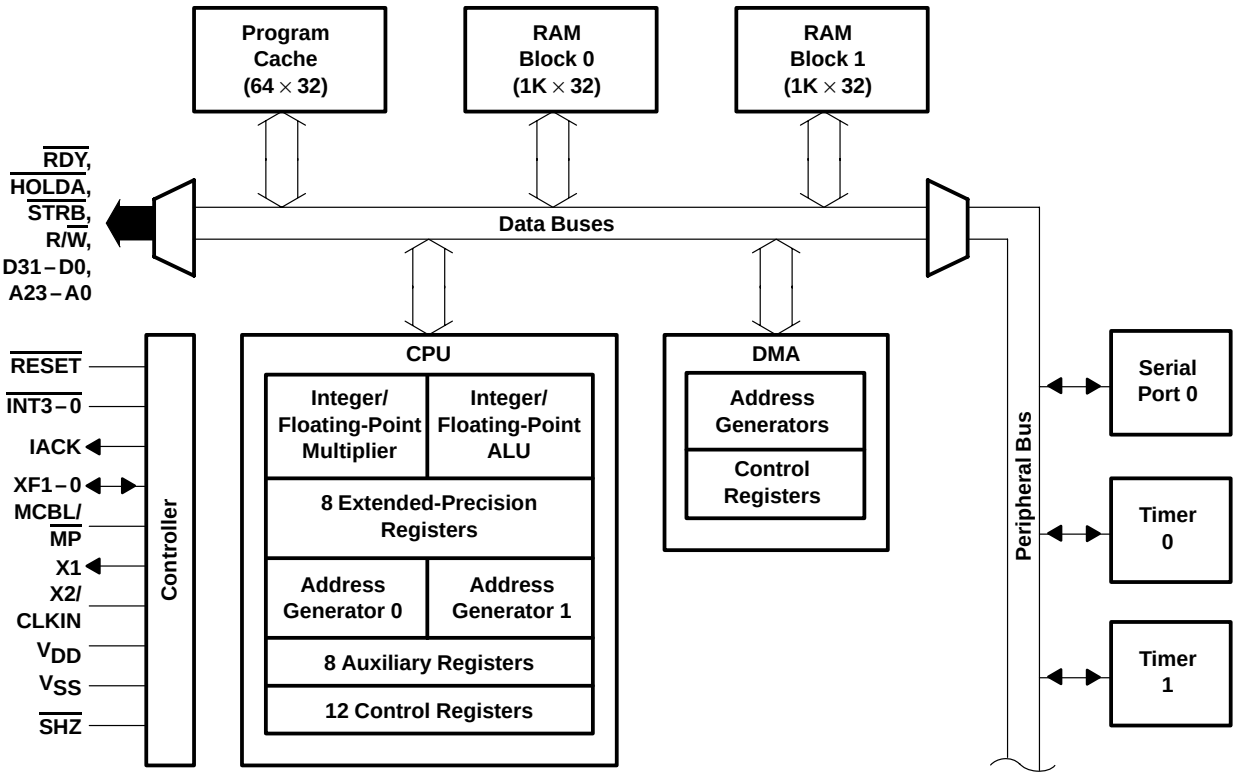


Figure 4. TMS320C31 DSP

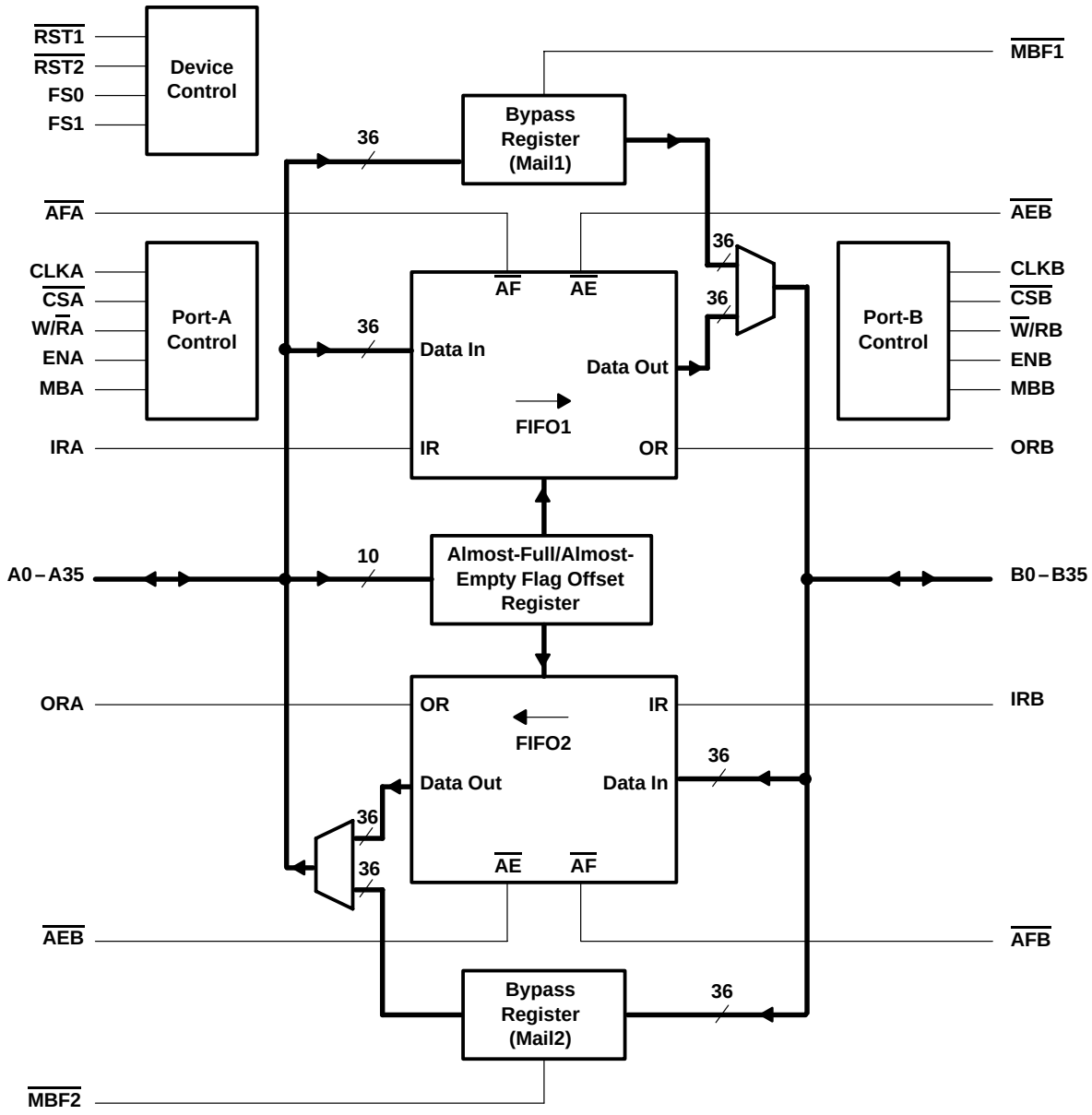


Figure 5. SN74ACT3632 512 × 36 × 2 Bidirectional FIFO

Figure 6 shows a memory map of a TMS320C31 in microprocessor mode. The address allocation for SN74ACT3632 functions is shown in the last block. These addresses are assigned assuming the 8M-word space between 040h and 7FFFFFh are adequate for the application's external memory needs. Different addresses are provided for FIFO read and FIFO write. FIFO addresses are separated in the map to minimize the number of address lines used to decode an SN74ACT3632 operation.

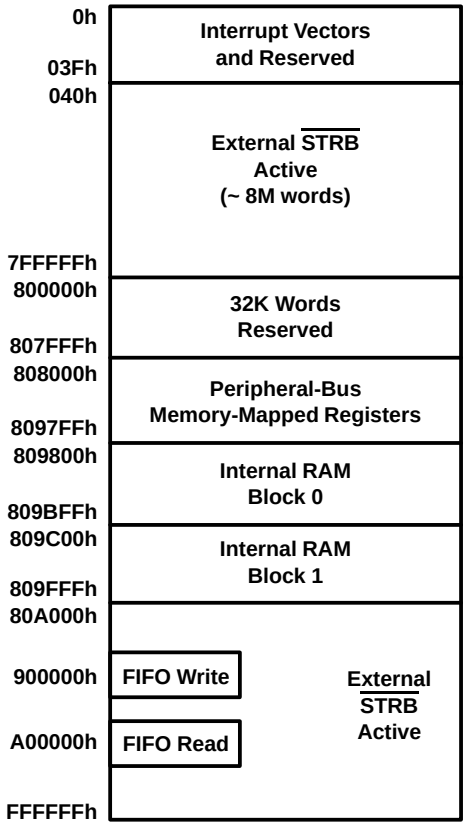


Figure 6. TMS320C31 Memory Map in Microprocessor Mode

Although not shown in this example, another address can be allocated for the 36-bit bypass registers present on the SN74ACT3632. Bypass registers are useful elements for separating a control word from the data in a FIFO queue. An external mail flag is set low on the SN74ACT3632 when new data is written to its corresponding register. This signals the receiving bus of an available control word, and the external mail flag is set high when the bypass register is read. One use of the bypass registers is to pass packet-size information between the SN74ACT3632 ports for DMA initialization.

The FIFO addresses in the memory map are accessible by the processor's DMA or through single-word load and store instructions. DMA transfer is the preferred method for moving large blocks of data. Instead of using the CPU for each single-word transfer, a small DMA setup overhead is needed to initialize the transfer of several words. The DMA becomes a bus master and performs the block transfer while the CPU is not using the external bus. This frees the CPU to accomplish its primary task of floating-point and fixed-point mathematical operations.

DMA Considerations

The DMA processor gives the user flexibility in designing the FIFO data flow control. The transfer counter that initializes the DMA for the number of transfers is decremented after each transfer is complete. Once the transfer counter reaches zero, transfers can be stopped and an interrupt can be sent to the CPU. Each DMA transfer can be synchronized to the source, synchronized to the destination, or synchronized to both the source and destination for a transfer by using the DSP interrupts. The user also can choose to increment or decrement the source and destination addresses after each transfer. These flexible features create several decisions to be made by the designer.

The method used to initiate the DMA for a FIFO transfer also introduces several considerations. One method is to schedule a FIFO write or read in software, where the DMA is initiated at a particular point in the program. This is the simplest method, but requires the most knowledge about the data-transfer characteristics (e.g., knowing at what point in the program data is ready to be transferred to or from the FIFO). This often is difficult to assess for FIFO reads when the data is queued asynchronous to the DSP program and is more useful for initiating FIFO writes. DMA synchronization to the SN74ACT3632 using the output-ready (OR) flag of the reading FIFO and the input-ready (IR) flag of the writing FIFO prevents reading from an empty FIFO or writing to a full FIFO. DMA synchronization is needed in this instance, since a read or write to a FIFO that is not ready results in multiple wait states on the DSP bus until the FIFO is ready for the specific transfer. Furthermore, DMA synchronization eliminates the need for generating a TMS320C31 ready ($\overline{\text{RDY}}$) signal based on the FIFO flags. This is a great benefit due to the small address valid to $\overline{\text{RDY}}$ maximum delay specification that must be met for zero-wait-state operation.

Another method to schedule a DMA operation is through hardware by using the programmable almost-full and almost-empty flags available on the SN74ACT3632. These flags can be used to interrupt the DSP when a data packet is available for transfer (or when space is available to receive a packet transfer). This can be done with fixed packet sizes or packets of variable length. For a fixed packet length, the programmable FIFO flags are set to show when the FIFO is ready to transmit or receive an entire packet of data. The DMA is then set for a transfer length equal to the packet size. This eliminates the need for source/destination synchronization, while ensuring reads are not attempted from an empty FIFO nor writes attempted to a full FIFO.

Hardware scheduling of a DMA operation using variable packet lengths generally needs to use DMA synchronization for controlling the FIFO. The SN74ACT3632 almost-full and almost-empty flags can be programmed to indicate when a portion of the packet in the receiving FIFO has been stored or when a portion of the packet space is available in the transmitting FIFO. DMA synchronization to the SN74ACT3632 with the IR and OR flags prevents FIFO overflow and underflow. The additional hardware required to support this method includes either four TMS320C31 interrupt lines ($\overline{\text{IRQn}}$) or an external device that combines the function of two or more interrupt lines into a single interrupt line.

Example Control of the SN74ACT3632 Using the DMA

The following example illustrates the use of the SN74ACT3632 FIFO to channel data between a TMS320C31 DSP and a generic local or backplane bus. The emphasis of the example is on the DSP-to-FIFO interface hardware and software. Data transfers to and from the DSP are in fixed packet sizes. This requirement results in minimal hardware complexity and a slight increase in software complexity. The higher software complexity required to divide a large data transfer in the fixed-length packets and track the number of packets delivered or received.

Data to be received by the DSP is put in the FIFO asynchronous to the DSP program execution; therefore, a hardware-scheduling mechanism is used to initiate the DMA to read data from the SN74ACT3632. DMA writes to the FIFO are scheduled in software.

Figure 7 shows the flow chart for controlling FIFO reads. FIFO reads are scheduled with hardware using packet-detecting interrupts generated by the almost-empty flag of a receiving FIFO. A combination of enabling the interrupt and polling its status initiates FIFO reads. The enabled interrupt initiates the first packet transfer after a FIFO empty condition, and interrupt polling handles transfers when multiple packets are stored in the FIFO. The read transfers are fixed in packet size and require no DMA synchronization. CPU interrupts from the FIFO almost-empty flag are disabled at the first of the routine and are not enabled at the end of the routine. Before a DMA operation is initiated, the interrupt routine checks if the DMA transfer counter (TCOUNT) is zero to avoid interfering with any concurrent DMA operation. If the DMA is being used, the destination address is written to a read schedule table and the routine terminates. If the TCOUNT value is zero, a DMA sequence is initiated to move the data packet from the FIFO to processor memory. An additional algorithm, such as concatenating a number of blocks together, manages the memory placement of the incoming data block.

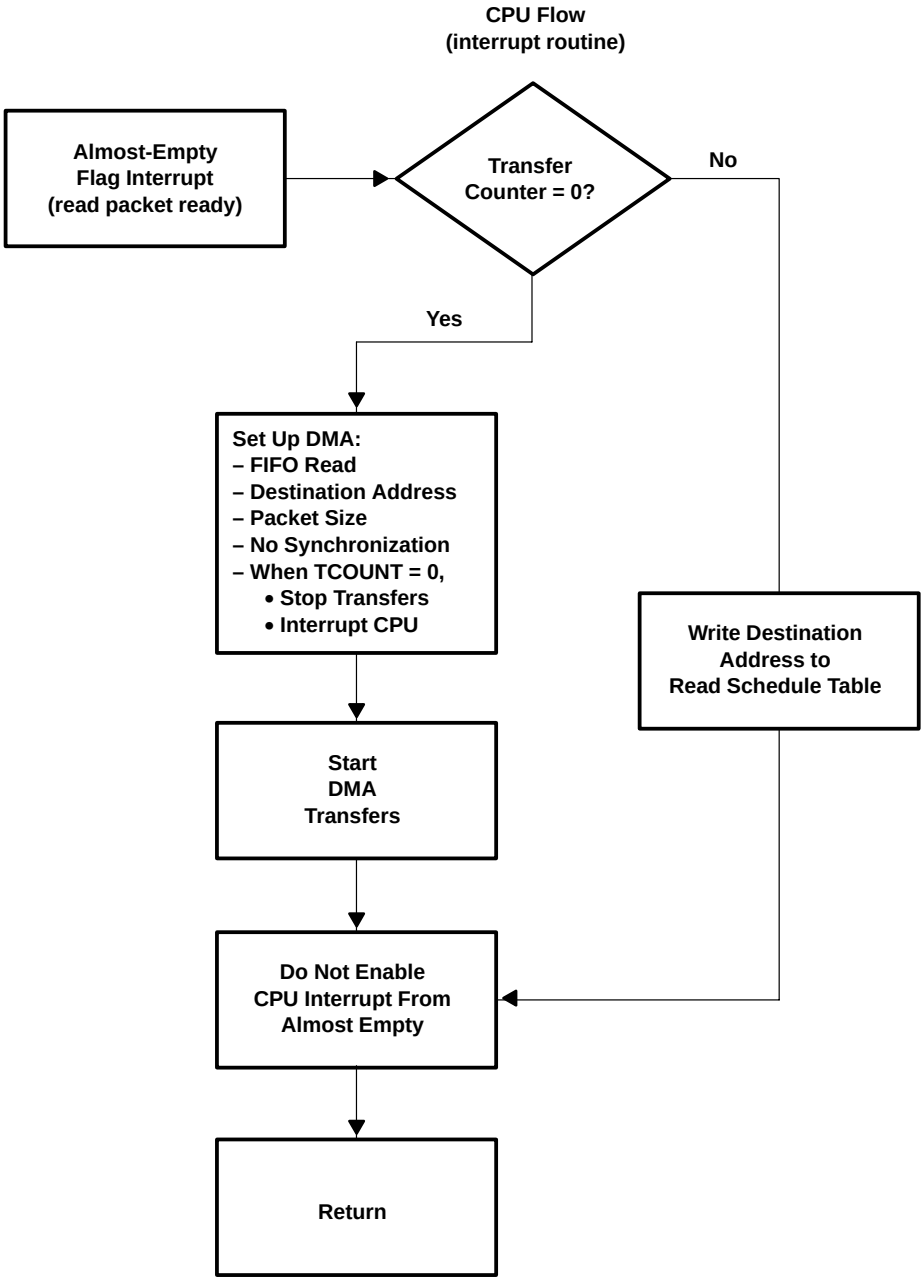


Figure 7. Routine for Almost-Empty Flag CPU Interrupt to Schedule FIFO Reads

The read schedule table comprises two memory locations, organized as a stack. The bottom of the stack is an arbitrary null pointer that indicates an empty table on top of the stack. Only one schedule location is needed since the interrupt from the receiving FIFO indicates the presence of a singular packet. When the interrupt routine finds the DMA in use, the beginning DMA destination address is pushed on top of the stack. This buffered information can be used at the end of the current DMA operation to initiate a new sequence.

Figure 8 shows an example flow chart for controlling FIFO writes. FIFO writes are scheduled in software by polling the almost-full flag of the transmit buffer on the SN74ACT3632. Packet sizes are assumed fixed in length to mirror the FIFO read operation and eliminate the need for DMA destination synchronization. The source address is written to a write schedule table if either the almost-full flag indicates adequate space is not available for a block write or if the DMA is in use. This write schedule

table is a circular buffer with two memory pointers indicating the head and the tail of the buffer. To schedule a write to the FIFO, the beginning source address is written to the buffer and the write pointer is incremented.

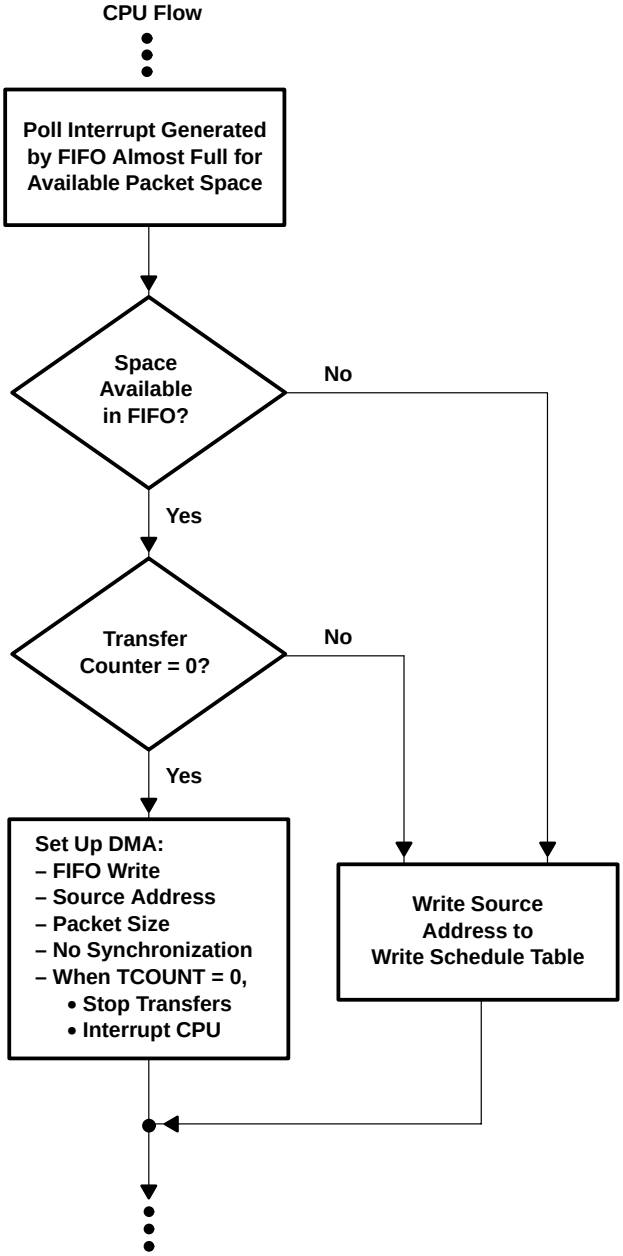


Figure 8. Scheduling FIFO Writes

For both FIFO writes and FIFO reads, no destination or source synchronization is used since the block sizes are known and packet-detecting mechanisms are used. The DMA transfer counter is loaded with the block size and the DMA global control is programmed to stop transfers and interrupt the CPU when the counter is zero. When writing to the FIFO from memory, the source address is incremented or decremented after each transfer and the destination address is not changed. The converse is true for reading FIFO data and placing the block in memory.

Figure 9 shows a flow chart of the interrupt routine initiated when the transfer counter reaches zero. Scheduled FIFO reads have priority over FIFO writes in the program, but scheduled FIFO writes can take precedence. The internal TMS320C31

interrupt-flag (IF) bits controlled by the FIFO flags are cleared at the beginning of the routine. This ensures a disabled interrupt is reflected by its IF status if the preceding DMA transfer sequence resulted in disabling one of the external interrupt lines.

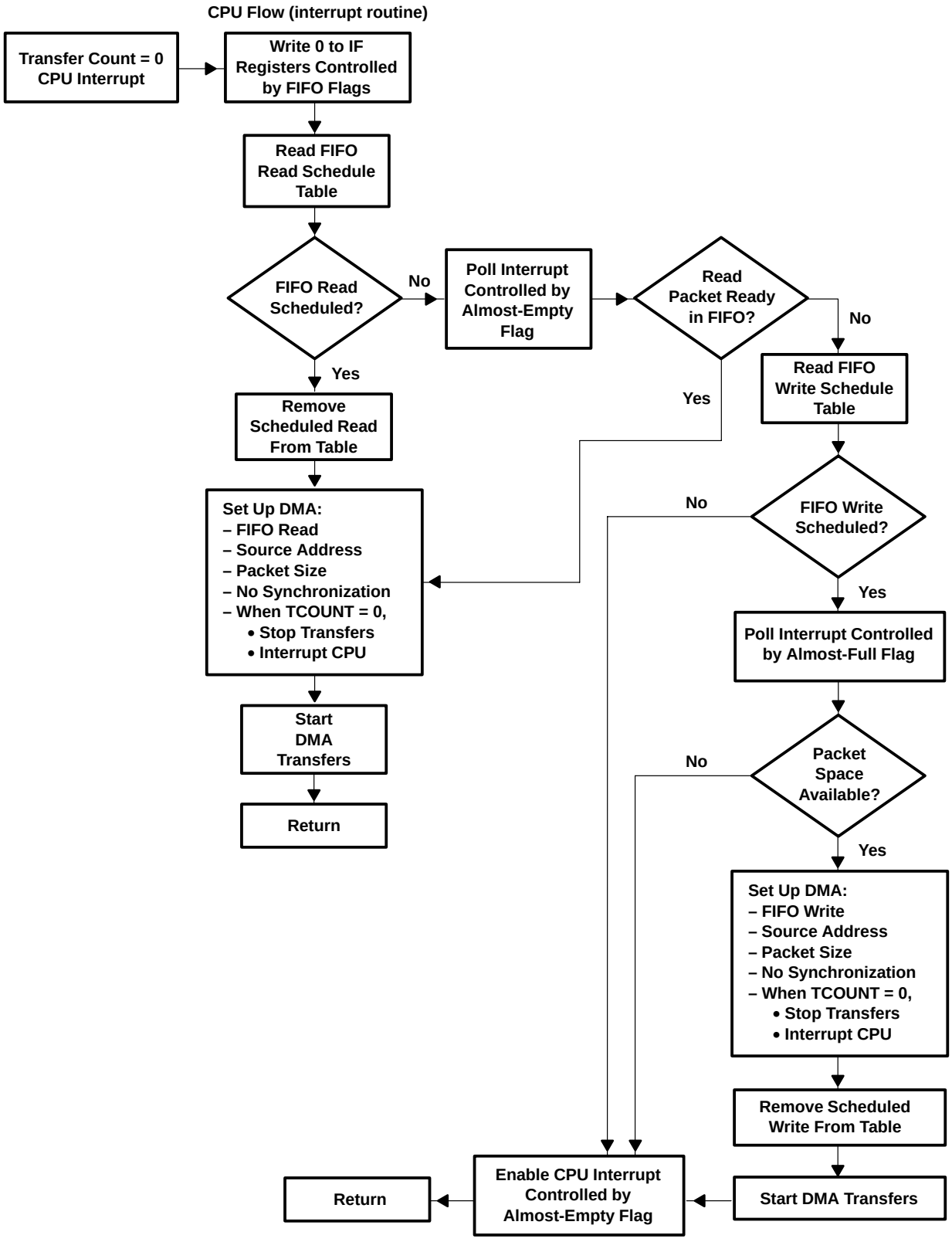


Figure 9. Routine for DMA Interrupt to CPU When Transfer Counter Reaches Zero

When FIFO writes or reads are scheduled in a table, a DMA sequence is started using the source or destination address found in the table. The almost-empty flag of the receiving FIFO is polled if there are no scheduled reads, and a read is started if a packet is present. The almost-full flag of the transmitting FIFO is polled before starting a DMA sequence writing the FIFO to ensure sufficient space is allocated for the transfer. If there are no scheduled FIFO reads and a data packet is not ready for transmission, the CPU interrupt generated by the almost-empty flag is enabled to use hardware-driven scheduling of FIFO reads.

Programming the FIFO Almost-Full-Flag and Almost-Empty-Flag Offsets

The SN74ACT3632 flexible flag-programming scheme aids the designer in creating a custom packet size. A choice of three hardware-coded values can be selected during reset or the offsets can be programmed by the user. Two SN74ACT3632 flag-select pins (FS0, FS1) are tied low to put the device in the user-programmable mode. After a reset with FS0 and FS1 low, the first four data writes to FIFO1 from port A of the device load offset values for the four almost-full and almost-empty flags of the device.

Port A of the SN74ACT3632 is connected to the TMS320C31 bus in this example so that the DSP can choose the FIFO offsets during system initialization. Both FS0 and FS1 on the SN74ACT3632 are tied to ground so that the first four FIFO writes on port A result in programming the almost-full/almost-empty flag offsets. This offset information is not stored in FIFO memory, and the device automatically begins normal operation when the programming is complete. Bypass registers on the SN74ACT3632 enable the DSP to gather flag offset values from the system controlling the opposite port of the FIFO. Flag offset values can be programmed from 1 to 508 by the binary value on SN74ACT3632 bits A0–A8. Input levels on FIFO bits A9– A35 are ignored for flag-offset programming.

A FIFO almost-empty ($\overline{AEA}$, $\overline{AEB}$) flag is low when the number of words stored in its buffer is less than or equal to the flag's offset value, and high when the number of stored words exceeds the offset value. A FIFO almost-full ($\overline{AFA}$, $\overline{AFB}$) flag is low when the number of empty locations in a FIFO is less than or equal to the flag's offset value and is high when the number of empty locations is greater than the offset value. Flag-offset values are easily selected based on packet size.

Calculating FIFO Flag-Offset Values

In the following example, the port-A almost-empty ($\overline{AEA}$) flag is used to alert the TMS320C31 of an available packet in FIFO2 and the port-A almost-full ($\overline{AFA}$) flag signals the processor of an available packet space in FIFO1. Figure 10 shows a method to choose flag-offset values assuming a packet size of 128. Almost-full flag selection is straightforward since the $\overline{AFA}$ flag is high when offset + 1 or more empty locations are available in FIFO1. The first-word fallthrough characteristic of the SN74ACT3632 must be considered for almost-empty flag offset selection.

First-word fallthrough refers to the method used to send new data to a FIFO output register. New data is read from a FIFO to its output register on a rising edge of the FIFO reading clock when one of these two conditions is true:

A new word is available in memory and the FIFO's output-ready (ORA, ORB) flag is low.

A new word is available in memory, the FIFO's output-ready (ORA, ORB) flag is high, and a FIFO read is selected by the port enables.

A new word is available in a FIFO when at least two low-to-high transitions of the FIFO reading clock have occurred since the word was written to the FIFO.

A word stored in an empty FIFO is automatically shifted to the FIFO output register. This unsolicited read frees one location in FIFO memory; therefore, the almost-empty offset selection must be made with this characteristic in mind. Although the first-word fallthrough seems to complicate the FIFO control, this characteristic is beneficial from the hardware-design standpoint.

FLAG	INDICATION	CHOSEN PACKET SIZE	STORED PACKET SIZE AFTER FIRST-WORD FALLTHROUGH	FLAG OFFSET VALUE
$\overline{\text{AEA}}$	High level indicates available packet in FIFO2	128 (P)	127 (P – 1)	126 (P – 2)

a) CHOOSING AN $\overline{\text{AEA}}$ OFFSET FOR THE SN74ACT3632

FLAG	INDICATION	CHOSEN PACKET SIZE	FLAG OFFSET VALUE
$\overline{\text{AFA}}$	High level indicates available packet space in FIFO1	128 (P)	127 (P – 1)

b) CHOOSING AN $\overline{\text{AFA}}$ OFFSET FOR THE SN74ACT3632

Figure 10. Almost-Full-Flag and Almost-Empty-Flag Offset Selection

Hardware Interface

Figure 11 shows the connections necessary to interface an SN74ACT3632-30 FIFO to a TMS320C31-40 DSP. The decode programmable logic device (PLD) is a simple circuit to translate the TMS320C31 address selection into an SN74ACT3632 write or read enable. The FIFO mailbox-select (MBA, MBB) pins are tied to ground in this example, but also can have a unique TMS320C31 address when the bypass registers are needed. Flag-select (FS0, FS1) inputs also are tied to ground to put the SN74ACT3632 in processor-programming mode upon reset.

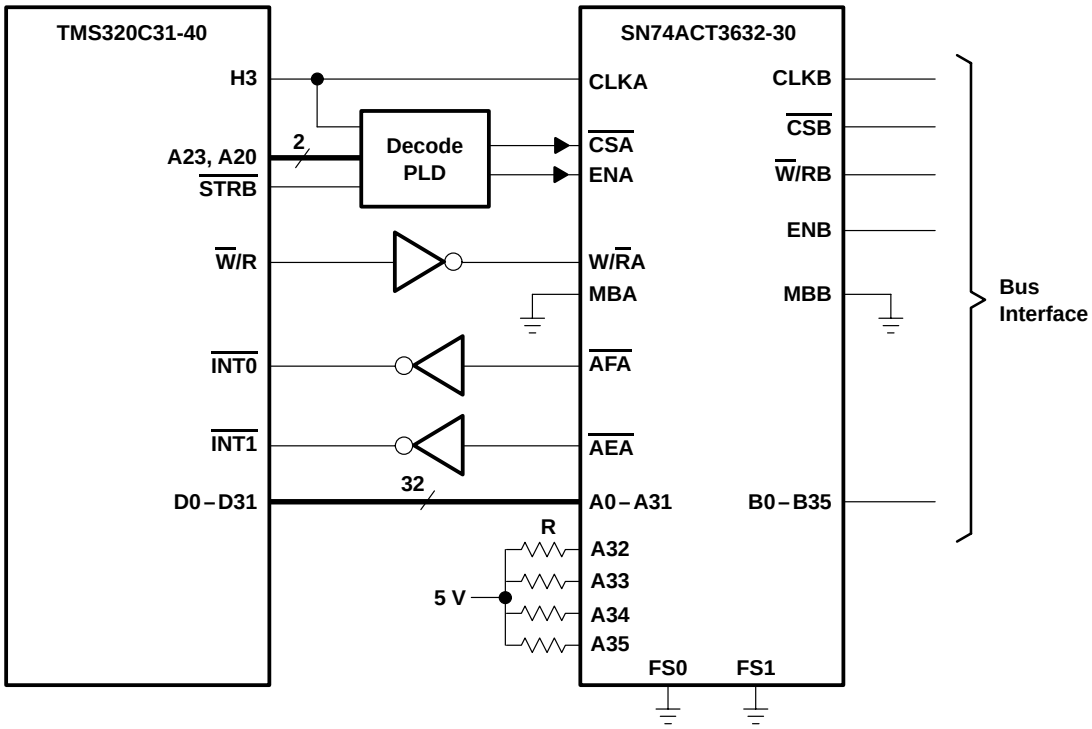
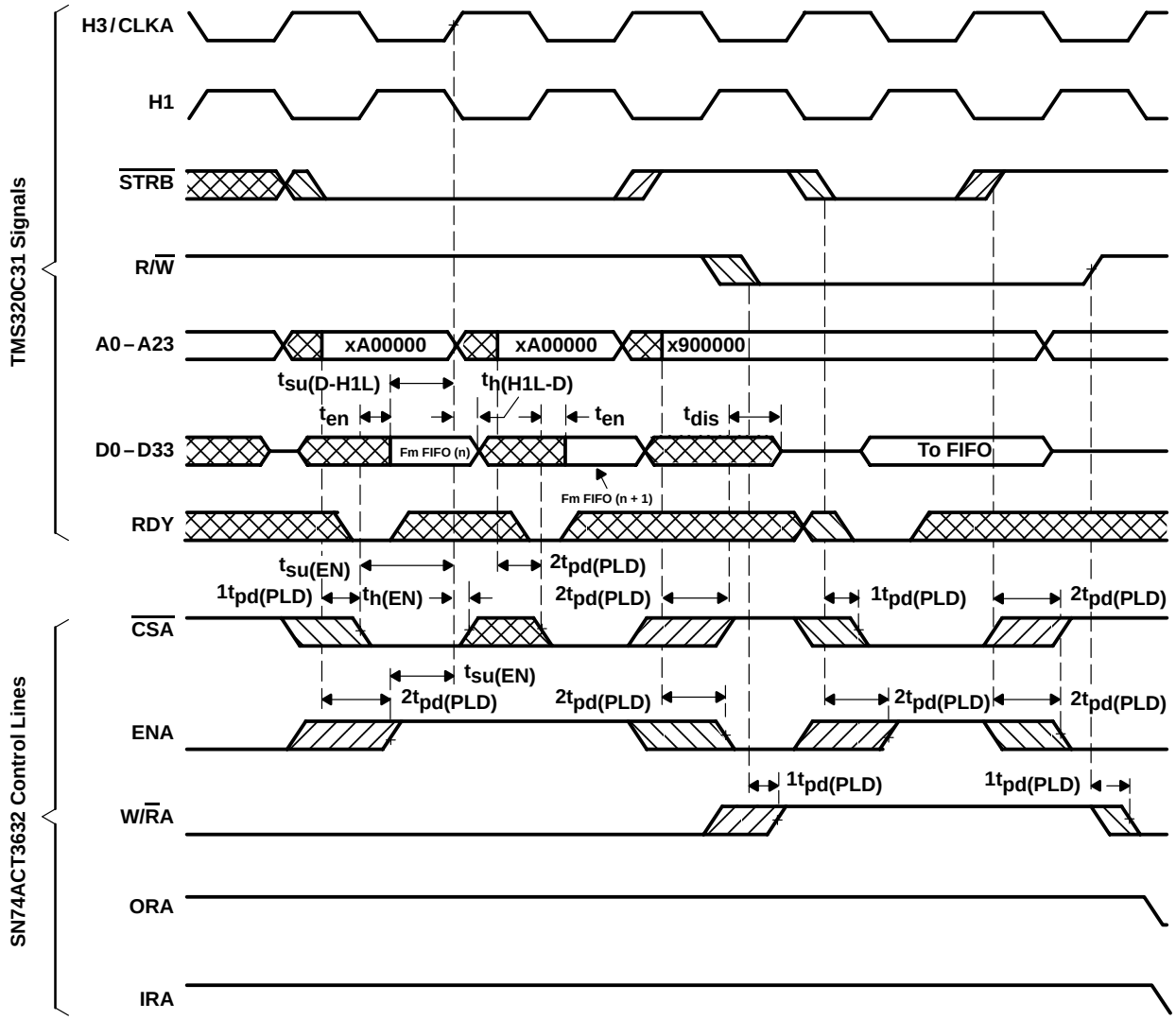


Figure 11. TMS320C31-40 Interface to an SN74ACT3632-30 FIFO

Read and Write Cycles

Figure 12 shows TMS320C31-40 read-read-write timing diagram in which the SN74ACT3632 port-A control lines accept the processor read and write cycles. Because the W/$\overline{R}$A input separates read and write cycles for the SN74ACT3632, the port-A chip select ($\overline{CSA}$) is logically sufficient for FIFO control. Due to timing considerations, the port-A enable (ENA) also is used to control FIFO read and write operations.

The H3 signal from the processor is used as an SN74ACT3632 rising-edge clock (CLKA). The rising edge of H3 occurs between 0 ns and 4 ns after the falling edge of H1, which is the processor bus-synchronizing event. According to the TMS320C31-40 timing, the processor address and $\overline{STRB}$ signals can change at the time of the falling edge of H1. If the minimum delay through the decode circuitry to a valid $\overline{CSA}$ signal is less than 5 ns, the SN74ACT3632 $\overline{CSA}$ to rising edge of CLKA hold time is violated.



Time values:

$1t_{pd(PLD)}$ = one PLD delay (3 ns min, 10 ns max)
$2t_{pd(PLD)}$ = two PLD delays (6 ns min, 10 ns max)
$t_{su(EN)}$ = 6 ns min
$t_{h(EN)}$ = 1 ns (hold time) + 4 ns (H1 low to H3 high) = 5 ns min

Figure 12. TMS320C31-40 Read-Read-Write Timing Diagram With an SN74ACT3632 512 × 36 × 2 FIFO

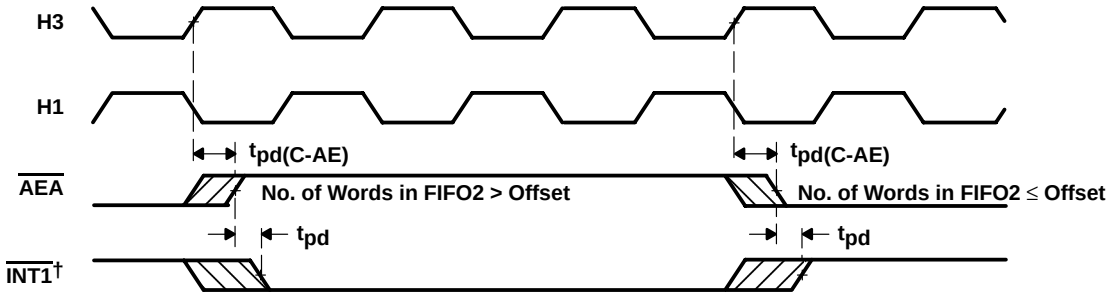
The minimum and maximum propagation delay times of a 10-ns PLD are 3 ns and 10 ns, respectively. A signal that feeds through the PLD twice has min/max switching of 6/20 ns, which eliminates the $\overline{CSA}$ hold-time problem but delays when the FIFO output bus is enabled. The solution to this problem is to design the $\overline{CSA}$ signal to have a single delay (3 ns, 10 ns) when switching from high to low for quick FIFO bus-enable times and a double delay (6 ns, 20 ns) when switching from low to high for proper $\overline{CSA}$ hold times. To prevent a $\overline{CSA}$ high-to-low transition from enabling a FIFO port-A transfer too early, the ENA signal is the inverse of the $\overline{CSA}$ signal with double delays (6 ns, 20 ns) for both high-to-low and low-to-high transitions.

A low on the TMS320C31 $\overline{RDY}$ signal during the low-to-high transition of H1 informs the processor that the present data-transfer cycle terminates on the next H1 falling edge. The first word written to an empty FIFO is automatically read to the FIFO output register. This data is available when the processor attempts to access it, since a FIFO2 read is attempted only when a data packet is available in FIFO2. Therefore, the first-word fallthrough characteristic of the SN74ACT3632 ensures FIFO reads can be done with zero wait cycles. The only constraint affecting zero-wait-state read access is the time from address valid to port-A enable on the SN74ACT3632, which is easily met.

In the following example, FIFO access is attempted only when a FIFO is ready for a packet transfer. Since the SN74ACT3632-30 easily supports zero-wait operation, $\overline{RDY}$ can be asserted low each time the SN74ACT3632 is addressed by the processor. An example of generating zero-wait $\overline{RDY}$ signals for an address space is given in section 12.2.2 of the *TMS320C3x User's Guide*. It is difficult to generate a $\overline{RDY}$ signal based on the status of the FIFO1 input-ready (IRA) and FIFO2 output-ready(OR) flags due to the 7-ns address valid to $\overline{RDY}$ maximum delay for the TMS320C31-40.

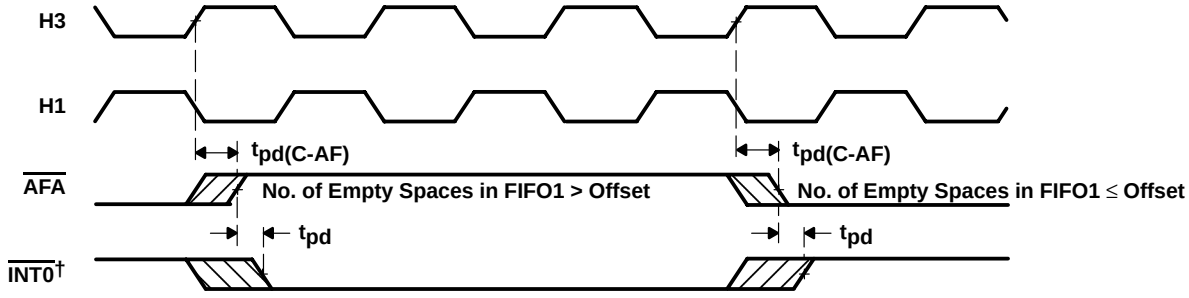
Interrupt Generation

Interrupt generation using the FIFO2 almost-empty ($\overline{AEA}$) flag and the FIFO1 almost-full ($\overline{AFA}$) flag is accomplished by inverting the signal. Figure 13 shows a low level on $\overline{INT1}$, indicating an available packet in FIFO2. Figure 14 shows a low level on $\overline{INT0}$, indicating an available packet space in FIFO1.



† Low level on $\overline{INT1}$ indicates an available FIFO2 packet space.

Figure 13. TMS320C31 Interrupt Generation by FIFO2 Almost-Empty Flag



† Low level on $\overline{INT0}$ indicates an available FIFO1 packet space.

Figure 14. TMS320C31 Interrupt Generation by FIFO1 Almost-Full Flag

Conclusion

FIFO memories are used in DSP systems for matching two datapaths with asynchronous clock or data rates. The SN74ACT3632 $512 \times 36 \times 2$ clocked FIFO provides a single-chip bidirectional buffering solution that interfaces nicely with TI floating-point DSPs. Programmable FIFO flags enable a variety of DMA control techniques to be used in handling data flow, and the FIFO control signals are easily derived from TMS320C3x outputs. Available in a variety of speed options, the SN74ACT3632 can interface a DSP to buses operating up to 67 MHz.

IMPORTANT NOTICE

Texas Instruments (TI) reserves the right to make changes to its products or to discontinue any semiconductor product or service without notice, and advises its customers to obtain the latest version of relevant information to verify, before placing orders, that the information being relied on is current.

TI warrants performance of its semiconductor products and related software to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are utilized to the extent TI deems necessary to support this warranty. Specific testing of all parameters of each device is not necessarily performed, except those mandated by government requirements.

Certain applications using semiconductor products may involve potential risks of death, personal injury, or severe property or environmental damage ("Critical Applications").

TI SEMICONDUCTOR PRODUCTS ARE NOT DESIGNED, INTENDED, AUTHORIZED, OR WARRANTED TO BE SUITABLE FOR USE IN LIFE-SUPPORT APPLICATIONS, DEVICES OR SYSTEMS OR OTHER CRITICAL APPLICATIONS.

Inclusion of TI products in such applications is understood to be fully at the risk of the customer. Use of TI products in such applications requires the written approval of an appropriate TI officer. Questions concerning potential risk applications should be directed to TI through a local SC sales office.

In order to minimize risks associated with the customer's applications, adequate design and operating safeguards should be provided by the customer to minimize inherent or procedural hazards.

TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein. Nor does TI warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of TI covering or relating to any combination, machine, or process in which such semiconductor products or services might be or are used.