

# ***FIFO Mailbox-Bypass Registers: Using Bypass Registers to Initialize DMA Control***

***Kam Kittrell and Steve Strom  
Advanced System Logic – Semiconductor Group***

SCAA007A  
March 1996



## **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.

Copyright © 1996, Texas Instruments Incorporated

## Contents

	<i>Title</i>	<i>Page</i>
<b>Introduction</b>	.....	
<b>Mailbox-Bypass Register Operation</b>	.....	
<b>DMA Controller Description</b>	.....	
<b>Initializing DMA Controllers With Mailbox-Bypass Registers</b>	.....	
<b>Conclusion</b>	.....	

## List of Illustrations

<i>Figure</i>	<i>Title</i>	<i>Page</i>
1	SN74ACT3641 FIFO Functional Block Diagram .....	
2	Timing Diagram for Mail1 Register and $\overline{\text{MBF1}}$ Status Flag .....	
3	TMS320C31 Memory Map in Microprocessor Mode Showing Mailbox-Bypass Register and FIFO SRAM Addresses .....	
4	TMS320C31 DSP Block Diagram .....	
5	Using DMA Control to Transfer Large Data Blocks .....	
6	Using Mailbox-Bypass Registers to Initialize DMA Control .....	

## **Introduction**

This application report describes the operation of FIFO mailbox-bypass registers and shows their application in an example that implements the direct memory access (DMA) control of a digital signal processor (DSP). All Texas Instruments (TI) 32- and 36-bit FIFOs contain mailbox-bypass registers that transmit priority data from one FIFO port to the other, either from port A to port B or port B to port A, without storing the data in the FIFO SRAM buffer. The SN74ACT3641, a unidirectional, clocked-DSP, application-specific FIFO (1K x 36), is used as the example for our discussion. In the following scenario, this device connects to a DMA controller that may be viewed as an integral part of any generic processing element. This discussion focuses on the DMA controller that is resident on board the TMS320C31 DSP.

## **Mailbox-Bypass Register Operation**

The operation of the FIFO mailbox-bypass register is straightforward. Each FIFO that features the mailbox option has two 32- or 36-bit bypass registers (Mail1 and Mail2) to pass command and control information between both ports without queuing the information in the FIFO SRAM buffer. The functional block diagram of the SN74ACT3641 is shown in Figure 1. The associated timing of the Mail1 register in the FIFO functional block diagram is shown in Figure 2.

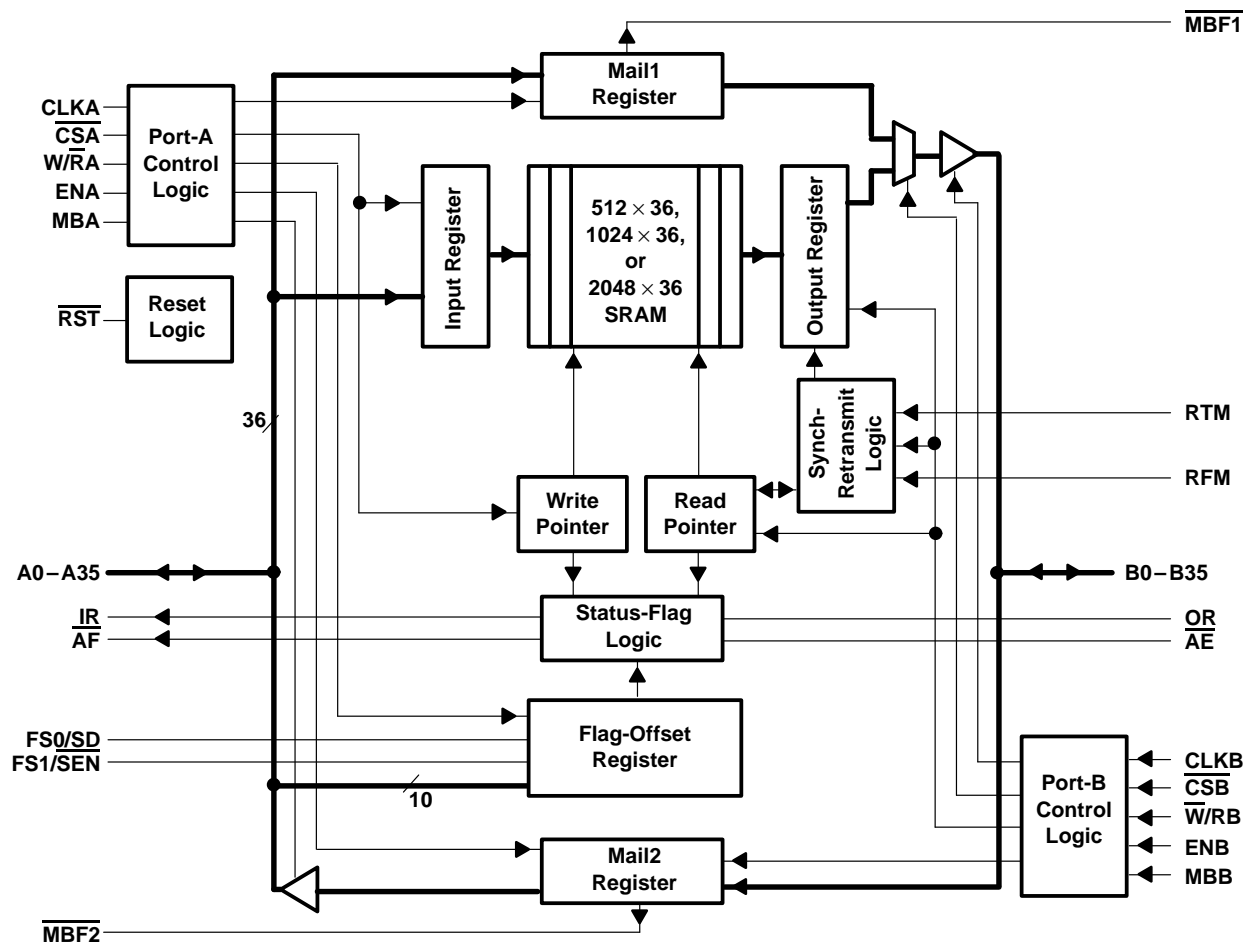
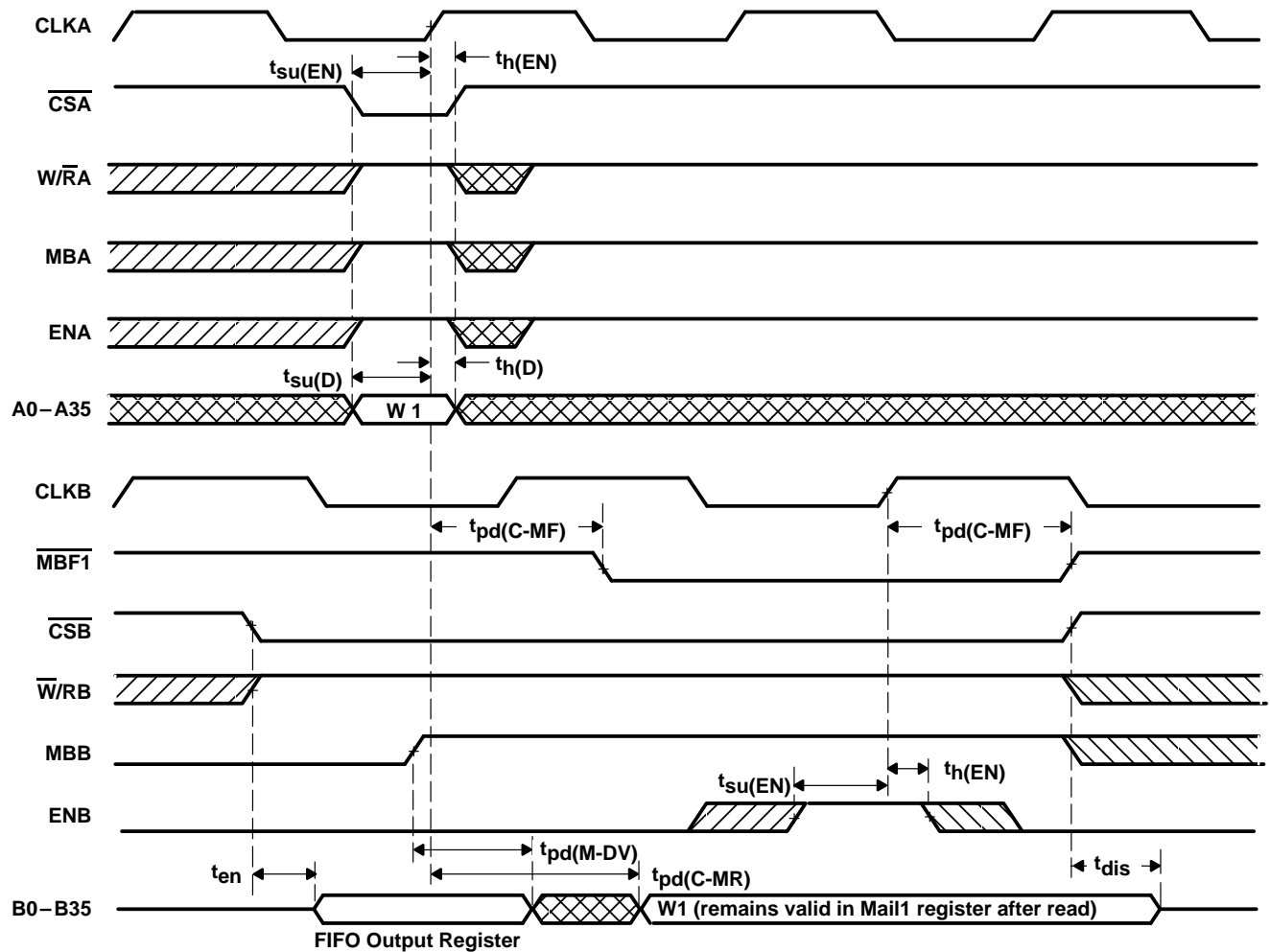


Figure 1. SN74ACT3641 FIFO Functional Block Diagram

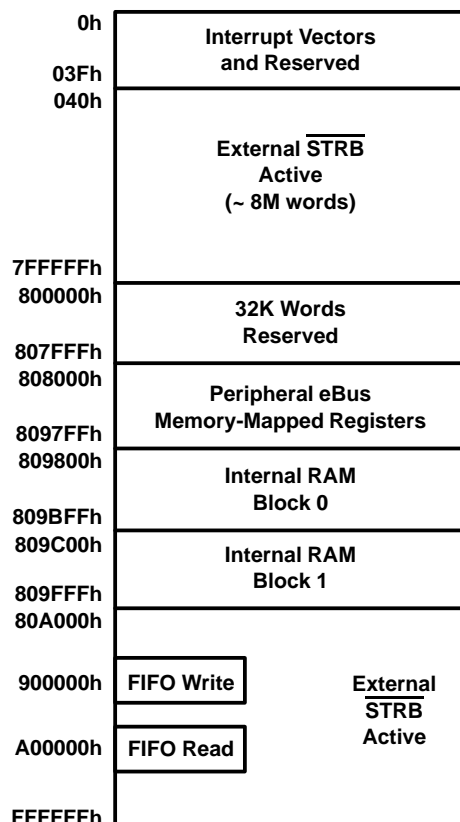


**Figure 2. Timing Diagram for Mail1 Register and  $\overline{\text{MBF1}}$  Status Flag**

Operation of the SN74ACT3641 mailbox-bypass registers is summarized below and shown in Figure 2. The mailbox-select inputs (MBA and MBB) are used to choose between a mail register and the FIFO SRAM for a port data transfer operation. A low-to-high transition on CLKA writes the data on A0–A35 to the Mail1 register when a port-A write is selected by  $\overline{\text{CSA}}$ ,  $\overline{\text{W/RA}}$ , and ENA while MBA is high. Likewise, a low-to-high transition on CLKB writes the data on B0–B35 to the Mail2 register when a port-B write is selected by  $\overline{\text{CSB}}$ ,  $\overline{\text{W/RB}}$ , and ENB while MBB is high. Writing data to a mail register sets its corresponding status flag ( $\overline{\text{MBF1}}$  or  $\overline{\text{MBF2}}$ ) low. Attempted writes to a mail register are ignored while the mail flag is low.

When the port-B data outputs (B0–B35) are active and the mailbox-select input (MBB) is low, data on the bus comes from the FIFO output register. When the port-B data outputs are active and the port-B mailbox-select input is high, data on the bus comes from the mailbox register (Mail1). The Mail2 register data is always present on the port-A data outputs (A0–A35) when they are active. The Mail1 register flag ( $\overline{\text{MBF1}}$ ) is set high by a low-to-high transition on CLKB when MBB is high and a port-B read is selected by  $\overline{\text{CSB}}$ ,  $\overline{\text{W/RB}}$ , and ENB. The Mail2 register flag ( $\overline{\text{MBF2}}$ ) is set high by a low-to-high transition on CLKA when MBA is high and a port-A read is selected by  $\overline{\text{CSA}}$ ,  $\overline{\text{W/RA}}$ , and ENA. Data in a mail register remains intact after it is read and changes only when new data is written to the register.

The mailbox-bypass registers can be easily mapped to a location within a DSP that is separate from the location assigned to the FIFO SRAM within the same memory map. This is made possible by virtue of the mailbox-register control terminals and timing similarities between the mailbox-register read/write operation and the FIFO SRAM read/write operation. An example of assigning the mailbox register and FIFO to the memory map of the TMS320C31 is shown in Figure 3.



**Figure 3. TMS320C31 Memory Map in Microprocessor Mode Showing Mailbox-Bypass Register and FIFO SRAM Addresses**

### DMA Controller Description

The TMS320C31 DSP contains an on-board direct-memory-access (DMA) controller that minimizes the requirement for the CPU to perform input/output operations. The TMS320C31 DSP block diagram is shown in Figure 4. Because of the DMA controller, the TMS320C31 can operate with slow external memories, peripherals, or analog-to-digital converters, for example, without slowing the computational throughput of the CPU.

Address and data buses are specifically dedicated to the DMA. As a result, there exists minimal conflict between the CPU and DMA controller. Using its address generators, source and destination registers, and transfer counter, the DMA controller can react to interrupts much like the CPU. Performing data transfers based upon interrupts received enables the DMA to execute input/output transfers that are typically the task of the CPU. While the DMA is receiving and transmitting data, the CPU is permitted to continue processing data.

A DMA transfer consists of two operations: reading from a memory location and writing to a memory location. These read and write operations can be as a block or a single word. The DMA controller can read from and write to any location in the TMS320C31 memory map, including all memory-mapped peripherals.

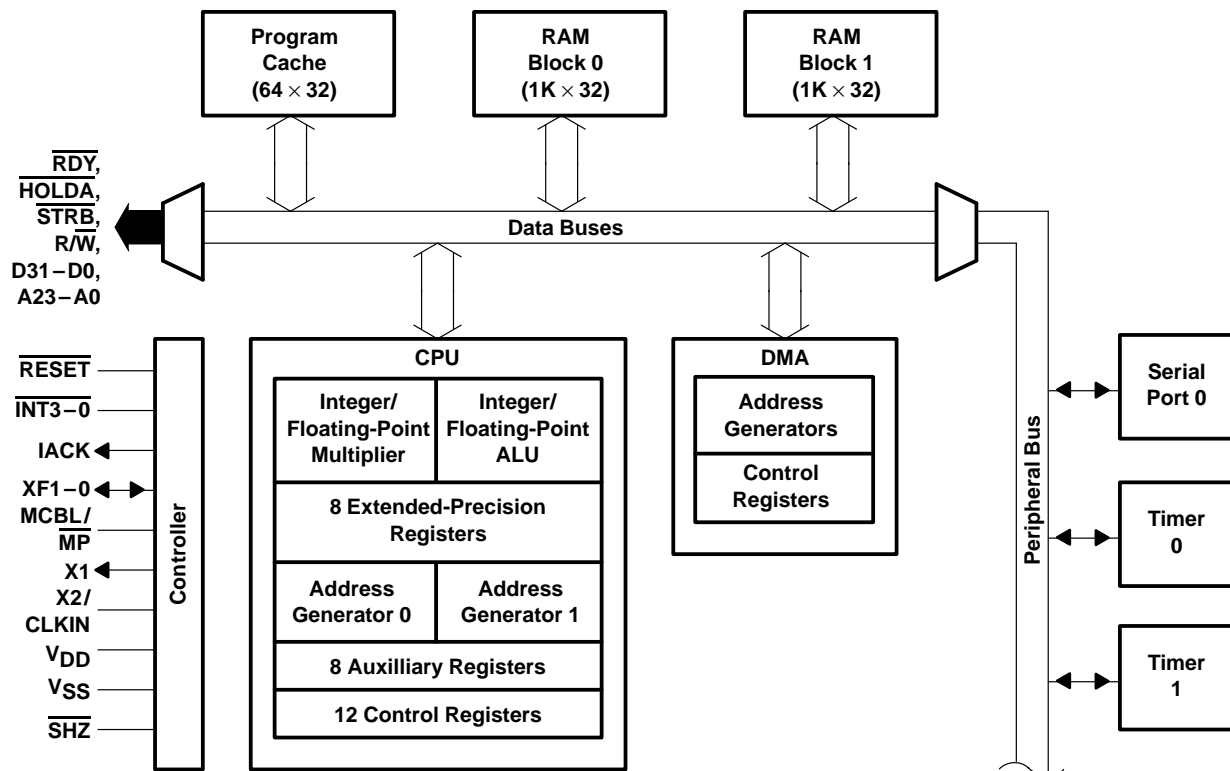
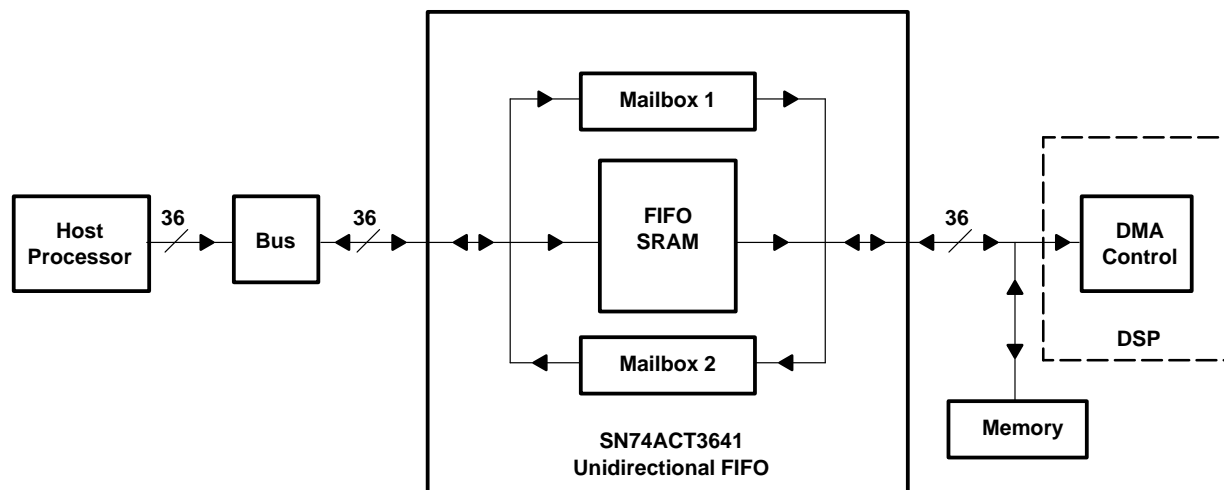


Figure 4. TMS320C31 DSP Block Diagram

### Initializing DMA Controllers With Mailbox-Bypass Registers

FIFO memories, such as the SN74ACT3641, typically channel data between a generic local or backplane bus and a TMS320C31 DSP. The FIFO collects incoming information from the bus and develops packets or vectors of data for transfer to the DSP via the DMA controller (see Figure 5). The packet size of the stored data is easily defined by using the FIFO programmable almost-empty or almost-full flags and, if necessary, its empty and full flags. Data transfers via the DMA controller are performed block by block from the FIFO to the DSP or to off-chip memory such as RAM. The DMA-controlled transfer is preferred for moving large blocks of data. Instead of using the DSP's CPU for each single-word transfer, investing in a small amount of setup overhead allows the DMA to initialize the transfer of several words. In this case, the DMA controller becomes the bus master and performs the block transfer while the CPU is not using the external bus. The CPU is free to accomplish its primary task of performing mathematical operations.

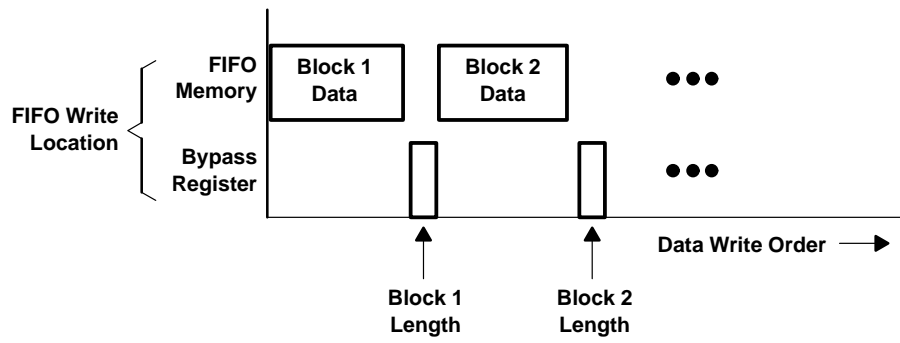




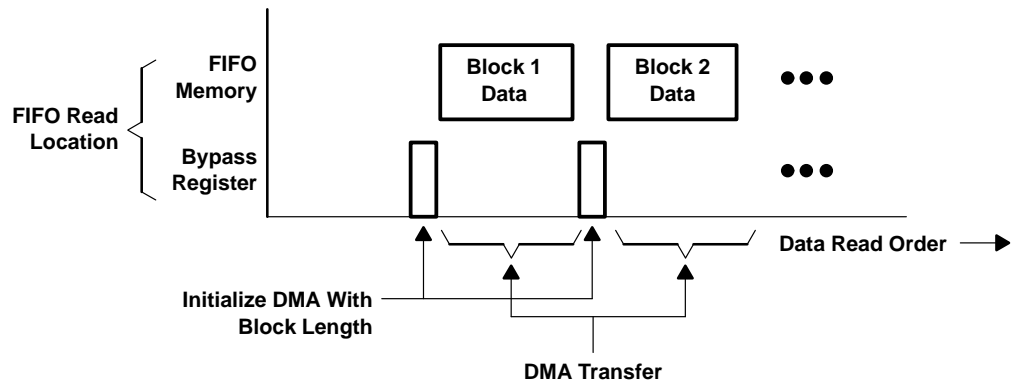
**Figure 5. Using DMA Control to Transfer Large Data Blocks**

Generally, the DMA controller requires the following information for data-block transfers: location of the data, destination of the data, and size of the data block to be transferred. The DMA controller is initialized by using the FIFO mailbox-bypass registers. This concept is shown in Figure 6. As previously, mailbox-bypass registers are useful in separating a control word from the data in a FIFO queue. In the example shown in Figure 6, the bypass registers of the SN74ACT3641 FIFO provide the DMA controller with the block-length initialization before performing the block transfer. At the same time, the mailbox-register status flags alert the DMA controller that the FIFO data is ready for transfer. In other instances, the mailbox register can also store a destination address in the DSP memory for incoming data.

In the earlier discussion, data flow through the mailbox register has been assumed to be in the bus-to-DSP direction. However, because of the bidirectional nature of the mailbox feature, the Mail2 register (see Figure 5) can also be used for transferring data in the DSP-to-bus direction. Many bus architectures support burst writes or have virtual addresses that can use the mailbox-bypass register for initialization.



(a) BUS-TO-FIFO DATA FLOW



(b) FIFO-TO-BUS DATA FLOW

Figure 6. Using Mailbox-Bypass Registers to Initialize DMA Control

## Conclusion

Mailbox-bypass registers are very useful in performing block-data transfers from a bus to a processing element or vice versa. Integrating the 32- or 36-bit mailbox registers on board the FIFO chip significantly reduces the device count per system. Likewise, implementing on-board mailbox registers with access timing similar to the FIFO SRAM also reduces the requirement for control logic.

TI's DSP application-specific FIFOs, in addition to the other application-specific FIFOs with mailbox-bypass registers, are summarized in Table 1.

**Table 1. FIFOs Featuring Mailbox-Bypass Registers**

DEVICE	ORGANIZATION	SPEED SORTS	MAX FREQ	MAX ACCESS	APPLICATION
SN74ACT3622	$256 \times 36 \times 2$	-15, -20, -30	67	11	DSP
SN74ACT3621	$512 \times 36$	-15, -20, -30	67	11	DSP
SN74ACT3632	$512 \times 36 \times 2$	-15, -20, -30	67	11	DSP
SN74ACT3638	$512 \times 32 \times 2$	-15, -20, -30	67	11	DSP
SN74ACT3641	$1K \times 36$	-15, -20, -30	67	11	DSP
SN74ACT3642	$1K \times 36 \times 2$	-15, -20, -30	67	11	DSP
SN74ACT3651	$2K \times 36$	-15, -20, -30	67	11	DSP
SN74ABT3611	$64 \times 36$	-15, -20, -30	67	10	High Bandwidth
SN74ABT3612	$64 \times 36 \times 2$	-15, -20, -30	67	10	High Bandwidth
SN74ABT3613	$64 \times 36$	-15, -20, -30	67	10	Internetworking
SN74ABT3614	$64 \times 36 \times 2$	-15, -20, -30	67	10	Internetworking