TEXAS INSTRUMENTS

Designing a Seamless Processor Interface Using FIFO Bus Matching and Byte Swapping

Application Report

1994

Advanced System Logic

TEXAS INSTRUMENTS

Printed in U.S.A.
1194–CP

SCAA020

Template Release Date: 7–20–94

TEXAS INSTRUMENTS

Designing a Seamless Processor Interface Using FIFO Bus Matching and Byte Swapping

1994

Advanced System Logic

Texas Instruments

Printed in U.S.A.
1194–CP

SCAA020

Application Report

Designing a Seamless Processor Interface Using FIFO Bus Matching and Byte Swapping

First-In, First-Out Technology

Steven Strom
Advanced System Logic – Semiconductor Group

TEXAS INSTRUMENTS

SCAA020

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 © 1994, Texas Instruments Incorporated

Contents

List of Illustrations

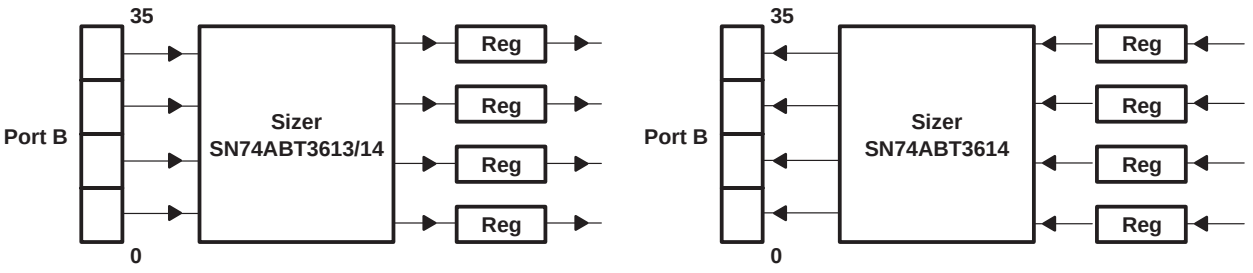
Introduction

Texas Instruments (TI) first-in, first-out memories (FIFOs) SN74ABT3613 (64 × 36) and SN74ABT3614 (64 × 36 × 2) are specifically targeted for internetworking applications. These highly integrated FIFOs are designed using TI's advanced clocked architecture and contain the necessary logic to perform the functions of bus matching, byte swapping, parity generation, and parity checking. Bus matching and byte swapping features are incorporated within a 36-bit single-chip solution; therefore, the need for multiple devices and discrete glue logic to control bus arbitration is eliminated. These high-speed, application-specific, internetworking FIFOs help to simplify the interconnection of complex networks, reduce system input/output (I/O) bottlenecks, resolve critical timing delays, and decrease board space requirements.

This report focuses on the data manipulation requirements of a digital signal processor (DSP)-based message system and a complex instruction set computer (CISC)-based broadband communications application. The first example shows a TMS320C31 DSP used with a microcontroller to store and retrieve compressed messages. The SN74ABT3614 bus-sizing feature is used to match the 32-bit DSP to the 8-bit microcontroller. In the second example, the SN74ABT3614 is used for bus matching and byte swapping of data that is supplied by an EtherNet controller to a CISC host processor. A brief functional description of the bus-matching and byte-swapping features is included in this report, followed by a detailed analysis of how these features are used to achieve the data transfer requirements of the example applications. Note that TI's FIFOs implement 9-bit bytes where one bit is reserved for parity.

Bus Matching

The bus-matching or bus-sizing feature on-board TI's Internetworking FIFOs offers a flow-through design that maintains port-to-port transparency and eliminates the need for discrete bus arbitration control logic. Bus matching is performed by the FIFO on the port-B side of the device (see Figure 1). Configurations supported by this feature consist of long-word format (36 bits), word format (18 bits), and byte format (9 bits), in either the big-endian or little-endian configuration for the byte and word size options. Big-endian format is associated with reduced instruction set computer (RISC)-based microprocessors such as the Motorola MC68000, whereas little-endian format is used in CISC-based processors like the Intel i486™. The difference in the formats is based on the placement of the most and least significant bytes as shown in Figure 2. The port-B bus size can be dynamically changed and operates synchronously to the port-B clock to communicate with peripherals of varying bus widths.



(a) Port-B Reads (SN74ABT3613/14) **(b) Port-B Writes (SN74ABT3614 only)**

Figure 1. Bus Sizer

i486 is a trademark of Intel Corporation.

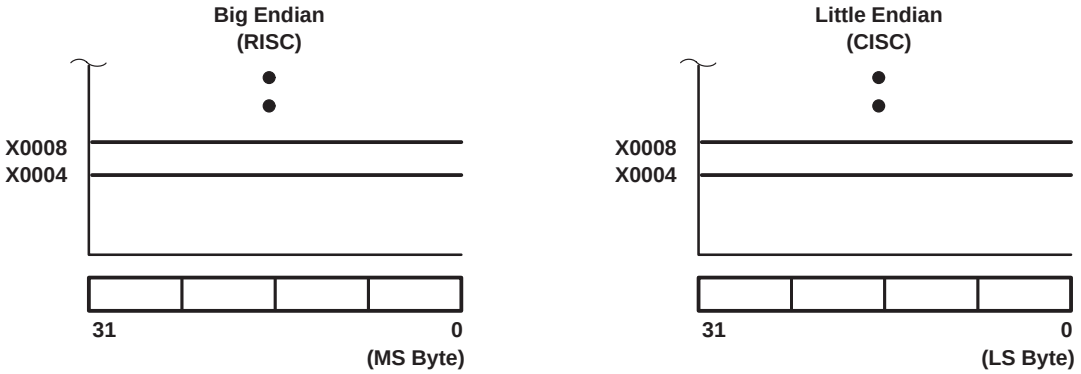


Figure 2. Big-Endian and Little-Endian Formats

Byte Swapping

The internetworking FIFOs are designed to provide maximum flexibility and ease of use. In addition to the bus-matching feature, a byte-swapping option also is included. The byte-swapping feature allows communication between systems with mixed bus protocols such as those used by RISC (big-endian format) and CISC (little-endian format) microprocessors (see Figure 2). Like bus matching, byte swapping is performed on the port-B side of the FIFO (see Figure 3). There are several variations of byte swapping that are permitted, depending on how the user has programmed the FIFO. These include no swap, byte swap, word swap, and byte-word swap.

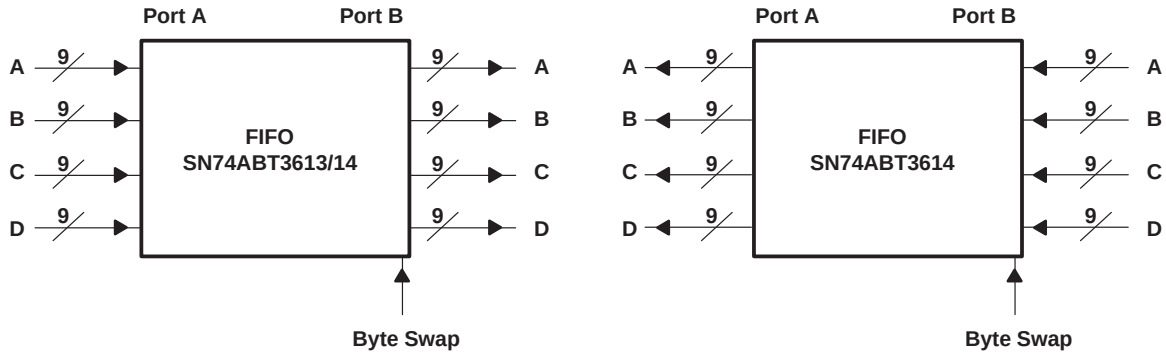


Figure 3. Byte Swapping

The bus-matching and byte-swapping operations are always handled in auxiliary registers, either after the data is read from the FIFO static random-access memory (SRAM) during a port-B read cycle (SN74ABT3613/14), or before the data is written to the FIFO SRAM during a port-B write cycle (SN74ABT3614 only). Depending on the system requirements, these features can be implemented together or separately. The following sequence is used for calculating byte order when simultaneously using both features on a data read: First, the data is read from memory; next, the byte-swapping function is performed; and finally, the bus sizing is executed. For writes to the FIFO: First, the bus sizing is applied; next the required swap is performed; and, finally, the result is written to memory.

A Seamless DSP Interface for a Message Storage/Retrieval System

Figure 4 shows a message storage and retrieval system using a TMS320C31 DSP and an 8-bit microcontroller. In this example, the DSP receives digitized data from an analog-to-digital (A/D) converter and compresses the data for storage. The 8-bit microcontroller directs the compressed data to a mass storage system, such as a tape or disk storage device. The high-speed FIFO, in this application, not only provides rate matching and buffering, but it also provides bus matching between the 32-bit DSP and the 8-bit microcontroller. In the retrieval mode, the bidirectional SN74ABT3614 FIFO allows four 8-bit bytes from the microcontroller to be written to port B and sized to a 32-bit long word to be read by the DSP. In the process, the format is configured as little endian, as required by this application. Recall from the description of bus matching that the format (big or little endian) can be selected when the various bus-sizing options are chosen.

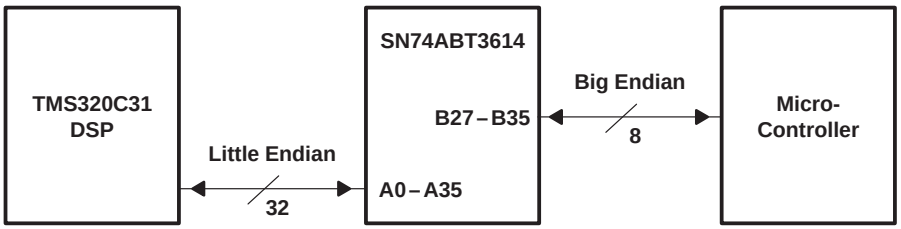


Figure 4. Message Storage/Retrieval System

In the storage mode, when data is flowing from the DSP to the microcontroller for mass storage, the FIFO also is capable of funnelling 32-bit long words down to four individual 8-bit bytes in big-endian format. The DSP writes the data to the FIFO as shown in Figure 5, while the microcontroller reads the data from the FIFO in big-endian, byte-size format as shown in Figure 6.

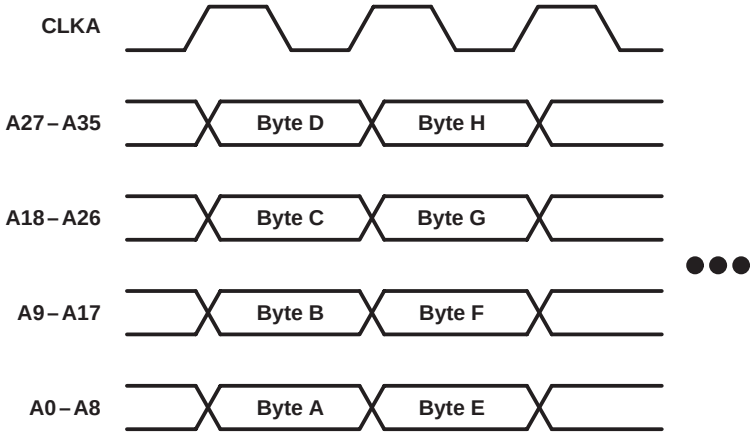


Figure 5. FIFO Port-A Write: 32-Bit Long-Word Format

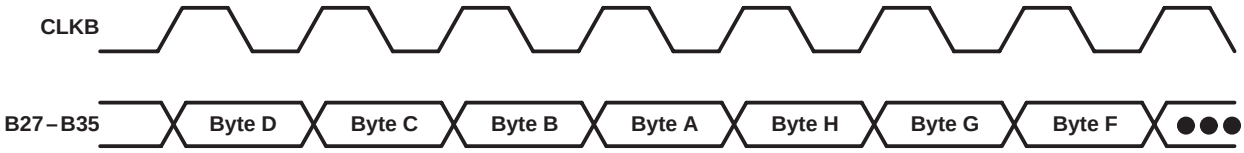


Figure 6. FIFO Port-B Read: 8-Bit Bytes, Big-Endian Format

Figure 7 shows another complex data transfer example for which the SN74ABT3614 internetworking FIFO provides a simple and cost-effective solution. In this high-bandwidth communications application, an EtherNet controller is required to interface with a CISC host processor. For instructive purposes, this example focuses on the flow of data from the 8-bit microcontroller to the 32-bit processor. The specific sequence of data mandated by this application results in the use of the bus-matching and byte-swapping functions of the SN74ABT3614. This one-chip solution provides an excellent fit for this application because of its data manipulation capabilities, high-speed performance, reliability, and small package size.

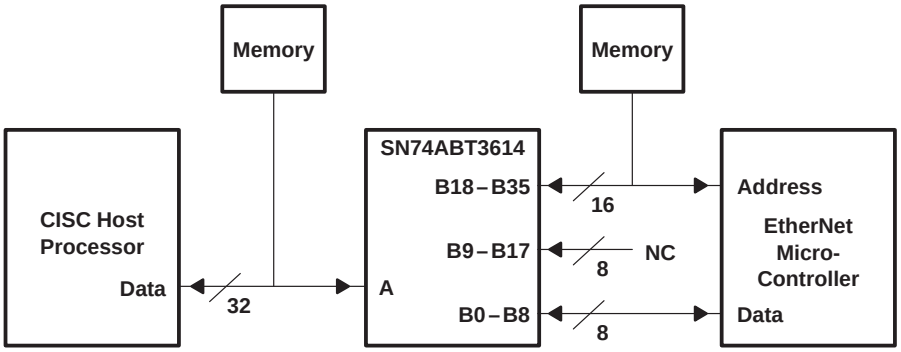


Figure 7. High-Bandwidth Communications Application

In Figure 7, the microcontroller provides information to the host processor using its address and data ports. The address is in the form of a 16-bit word, whereas the data is provided in 8-bit bytes. Figure 8 shows a timing diagram of the preferred format of address and data as required by the 32-bit processor. The address/data cycle consists of a 16-bit address followed by four 32-bit long words. As shown, the word-length address must reside in the least significant position of the first long word read by the processor, i.e., little-endian format. As shown in Figure 8, the information contained in the most significant half of this first long word is not used in this application. The four long words following the address, also organized in little-endian format, comprise 16 bytes of data as supplied by the microcontroller.

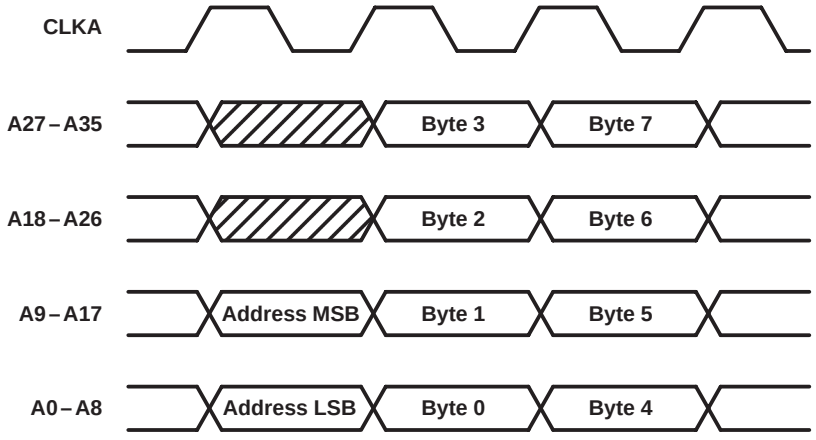


Figure 8. Host Processor Reading Data From Port A of the FIFO

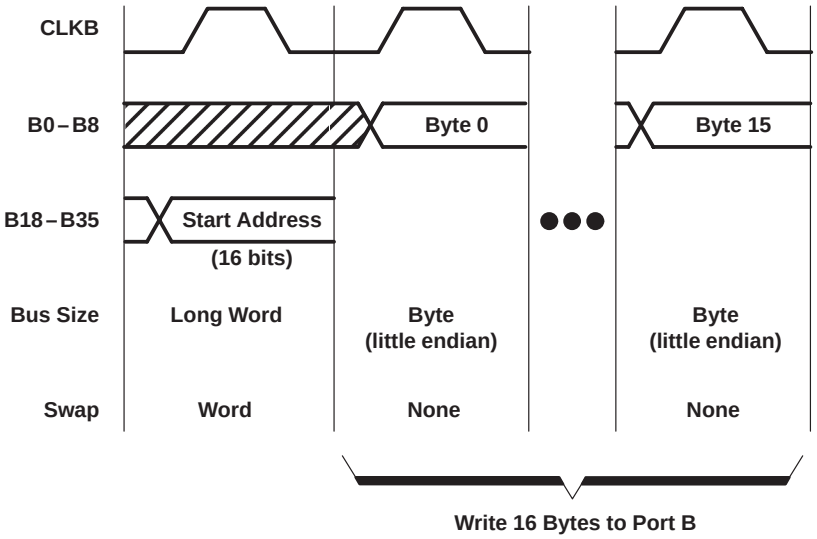


Figure 9. Microcontroller Writing Data to Port B of the FIFO

To configure the information for a port A read as shown in Figure 8, the bus-matching and byte-swapping functions of the SN74ABT3614 internetworking FIFO are used as shown in Figure 9. The microcontroller first writes the 16-bit address to the upper half or most significant part of port B of the FIFO (B18–B35). Performing a word swap on port B, this word-length address is transposed to the least significant part of this 36-bit port (B0–B17) and written to the FIFO's internal SRAM. On the other hand, the 8-bit data is written to the least significant byte of port B (B0–B8) by the EtherNet controller. Using bus matching, four 8-bit data bytes are combined to form a full 32-bit long word, which is consequently written to the FIFO's SRAM. Four long words are stored in the FIFO in this manner, completing the address/data cycle. The information stored in the FIFO can then be read by the host processor in the sequence described above and as shown in Figure 8.

Conclusion

The bus-matching and byte-swapping features of TI's internetworking FIFOs permit the establishment of a seamless interface between processors, such as DSPs, and other devices or buses with varying widths. Bus matching and byte swapping can be accomplished in a single-chip solution with the high-performance SN74ABT3613 and SN74ABT3614 FIFOs. There is no requirement for external devices or glue logic to implement these functions. In fact, as shown in the CISC-based example above, there exists the capability to dynamically change from one feature to the other, e.g., byte swapping to bus matching, merely by programming the FIFO accordingly. The internetworking FIFOs permit simple solutions with powerful feature sets, built-in reliability, high-speed performance, and small packaging options.