Designing with CY7C480xV25 Synchronous FIFOs

Introduction

The ever-increasing bandwidth demand in data communication systems has driven the need for higher speed devices with greater bus-widths. The CY7C480xV25 is the first family of Cypress Semiconductor's unidirectional FIFOs to operate at 200+MHz with 80-bit parallel interface, delivering more than 32 Gb/s of bandwidth. This family also provides a density of up to 5 Mb. This application note highlights the features and design considerations through a few application examples where this family of FIFOs is an excellent choice.

Please note that this document complements the data sheet published on Cypress website (www.cypress.com/cypress/prodgate/fifo/7c480xv25.html); readers are assumed to have read the data sheet before reading this document.

This document highlights certain features of the FIFOs in the context of applications, followed by system design considerations.

The following is an outline of this application note:

- Application Example 1: Network Card Interface
  Features addressed:
  - Bus Matching: Big/Little Endian, Bus Size
  - Programmable Flags: Parallel, Serial
- Application Example 2: Serial Backplane Interface
  Features addressed:
  - Bus Matching
  - Master vs. Partial Resets
  - Retransmit
- Additional Features:
  - Modes of Operation: CY standard vs. FWFT
  - Width Expansion
  - Depth Expansion
    - —Retransmit with FIFOs Cascaded in Depth
    - —Programming the Flags with FIFOs Cascaded in Depth
- Important Design Considerations
  - Reads/Writes
  - Power-up Sequence
- Conclusion

Network Card Interface

FIFOs provide a seamless way for data interchange in multiprocessor, multiple I/O communication systems where different devices and/or different processors need to communicate at different speeds and bus interface requirements. In this system configuration, FIFOs can be used for setting up parallel-bus mailboxes between CPUs and for rate decoupling and data transfer between I/O and memory. *Figure 1*shows the block diagram of such system.

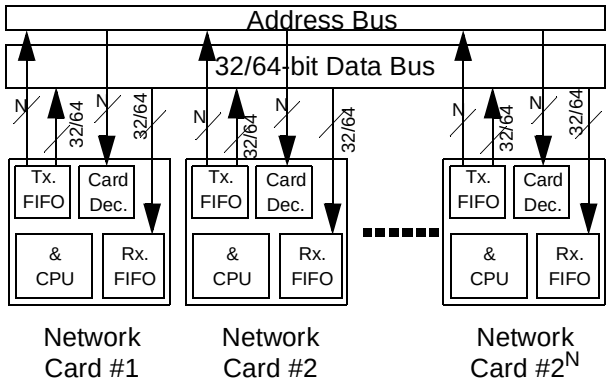

Figure 1. Parallel Backplane Example.

Each card can have two FIFOs, one to transmit, the other to receive. When a CPU needs to send data to another CPU or to an I/O line card, it sends its data (using DMA, for higher-speed operation) to FIFO at its full speed and raises a flag/interrupt to notify data availability. The I/O card, or the other CPU can then take data out of this FIFO over backplane bus at a different rate with a different bus interface type. Such arrangement overcomes several design issues affecting the speed of data transfer. Among them are the following.

The CPU on Each Network Card May Be Different

One can be using a different bus width and Endian arrangement compared to another. The bus matching feature on this FIFO will resolve this issue.

Bus Matching: Big/Little Endian, Bus Size

The FIFO can be configured to transfer data from Big to Little Endian and vice versa using the BE/FWFT pin with a master reset (MR), as well as to accommodate different bus-widths. For instance, card 1 may have a Big Endian CPU with 64-bit data bus, and is required to interface to a 32-bit back plane. In this scenario, the FIFO can be configured to interface with the on-board data bus using an 80-bit interface (SIZE1A and SIZE2A pins both tied to ground) and interface with the backplane with a 40-bit bus through the bus-matching function (SIZE1B tied to ground while SIZE2B tied to VDDQ). If the receiving card is a Little Endian machine, all that is required of the FIFO is during the master reset operation, the BE/FWFT signal is held LOW at the rising edge of MR. Data at the FIFO outputs will then be in Little Endian format. Unused outputs can be left unconnected. See *Figure 2*.

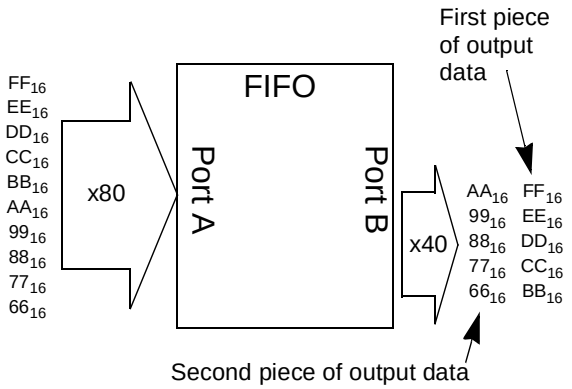

Figure 2. x80 to x40 Bus Match, Little Endian.

Rate Decoupling

Each of the network cards may be operating at a different speed. Without FIFOs, data transfer between two cards will need to be performed at the slower rate between the two. Using FIFOs will provide an excellent rate decoupling such that each card only deals with one of the FIFO ports, which can be run at maximum speed, and let the FIFO handle the transfer through the backplane at a different speed.

Programmable Flags

The programmable flags are also very useful to indicate whether the FIFO is almost full or almost empty, thus can be used as a trigger for devices on both sides to write to or read from the FIFO. This family of FIFOs also provides a serial programming mode in addition to the usual programming through the parallel port A. A dedicated serial input data pin with its associated enable signal allows the flags to be programmed serially during power up. This is particularly useful if Port A of the FIFO is shared with other devices such that parallel programming through Port A is inconvenient during power up.

Data Associated Addressing

Besides the obvious advantage of being able to operate at 200+ MHz, which delivers more than 32 Gb/s of data, having 80 bits of parallel interface provides an additional convenience. Instead of separate cycles for addressing and data transfers, systems can associate parallel bits with data words so devices can use the associated bits to perform address decoding and data transfers at the same time. This provides efficient data routing inside a complex system.

Assuming that the FIFO is interfacing with the network card with 64-bit data bus, the additional 16 bits can be used as address and/or command codes attached to that specific 64-bit data. For instance, if there are 16 network cards connected to the back plane, 4 of those 16 bits can be used as address bits ranging from 0000 to 1111, each address representing one of the 16 network cards.

Serial Backplane Interface

In typical data communication systems such as switches and routers, serial data is received, possibly processed, and retransmitted to its destination at a different rate. Such systems usually have blocks of memory and possibly multiple CPUs to process and switch incoming data to other serial links. One example would be a 100 Mb/s WAN serial data link could be switched to a much slower T1 line at 1.5 Mb/s through such a system. Without a FIFO, fast in-coming data will quickly fill up the memory and clog the data throughput of the system. Using a FIFO will allow data transfer at maximum speed from the memory to the FIFO, and have the SERDES (Serializer-Deserializer) clock out the data at T1 rate. Such frequency decoupling will free up the memory and the CPU(s) and reduce the chance of clogging up the memory.

Bus Matching

Having the bus matching feature on both ports provides a definite advantage, even if the system data bus is much narrower than 80 bits. For example, if the system data bus is 32-bits wide, the 80-bit FIFO can either take two 32-bit data at a time (with some additional logic), or, the width can be configured to a 40-bit bus using the bus matching feature. Both methods still achieve maximum throughput. The output port can be configured to interface with a SERDES of appropriate bus size. Any unused input pins should be pull to ground through a resistor, 1 kΩ should be sufficient.

Master vs. Partial Resets

Like other Cypress synchronous FIFOs, this family of FIFOs operates with 2 pointers during normal operation. The write pointer will be incremented when data is written into the FIFO, and the read pointer will be incremented when data is read from it. The FIFO can be treated like a circular buffer where the pointers loop around indefinitely. The difference between the two pointers will determine the state of the flags. The master reset operation basically clears the memory by bringing both pointers to location zero and setting the endian arrangement, bus size on both ports, programmable flag offsets, and timing mode (CY standard or First Word Fall Through, FWFT). The partial reset only brings the pointers back to location zero but keeps the rest of the settings. Master reset should be used at power-up to configure the FIFO, while partial reset can be used to clear the memory and keep the rest of the settings, such as when retransmit is required.

Retransmit

The retransmit operation is very useful in the event of an error or if a block of data needs to be repeatedly broadcast. Pulsing the retransmit pin essentially brings the read pointer back to location zero, and allows the old data to be read until the read pointer arrives at the location of the write pointer. Due to the fact that the pointers loop around in the memory, one cannot be sure if the data transmitted from location zero to the location of the write pointer is the desired block to be resent. Therefore, the FIFO should first be partial reset to bring both pointers back to location zero, then the block of data to be resent is written into the FIFO. With that, the data can be resent by reading from the FIFO. The same block of data can be repeatedly resent by pulsing the retransmit signal and read the FIFO to empty. Note that retransmit operation can only be performed in CY standard mode, and when the FIFO is in standalone or cascaded in width mode.

Additional Features

This section describes features offered by this family of FIFOs that are not restricted to any particular application.

CY vs. FWFT Mode

The CY7C480xV25 FIFOs have two modes of operation selected through the BE/$\overline{FWFT}$ pin: Cypress (CY) Standard Mode and First Word Fall Through Mode (FWFT). In the CY Standard Mode, the first word written into an empty FIFO is deposited into the memory array. A read operation is required to access that word (along with all other words residing in the memory). In the FWFT Mode, the first word written into an empty FIFO appears automatically on the outputs, and no read operation is required for that first word. Nevertheless, accessing subsequent words does necessitate a read operation. FWFT Mode is required for cascading two or more FIFOs, but can also be used in stand-alone mode. These timing modes are selected only after the rising edge of $\overline{MR}$, and the state of the signal should remain static throughout device operation. Note that $\overline{PR}$ does not affect the timing mode.

In the CY Standard Mode, each read operation performed on the FIFO retrieves the next piece of data to the FIFO output data bus. Thus the first read operation performed on the FIFO retrieves the first word onto the FIFO output data bus. The device reading from the FIFO (shares the same clock as CLKB of the FIFO) will latch the first piece of data on the second rising edge of the clock, which may also be the second read operation of the FIFO. The FWFT Mode however, allows the first piece of data to be at the output data bus before a read operation is performed, thus the first actual read operation actually retrieves the second piece of data from the FIFO. Such arrangement allows the device reading from the FIFO to latch the first piece of data at the first read operation. The switching waveform (*Figure 3*) and block diagram (*Figure 4*) illustrates these timing modes.

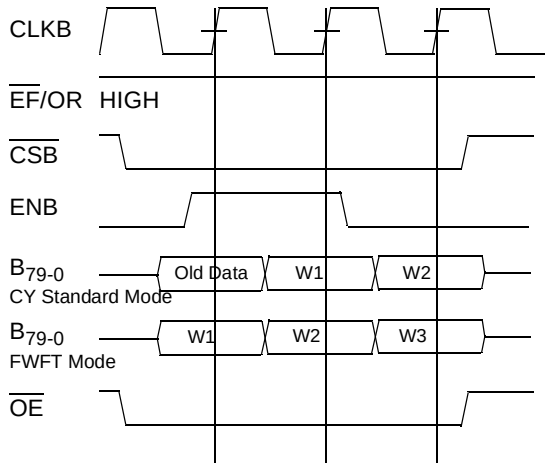

Figure 3. Comparing the Read Cycle Timing for CY Standard and FWFT Modes.

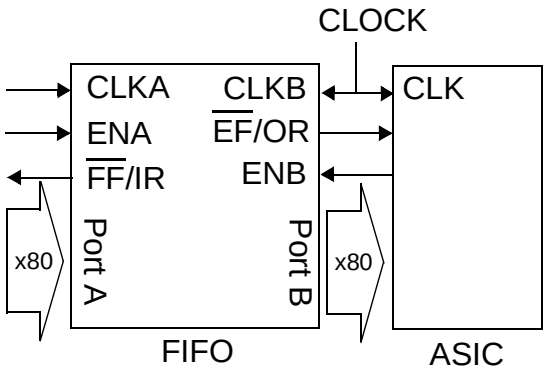

Figure 4. Example of an ASIC Reading from the FIFO.

Figure 4 assumes a typical FIFO interface with an ASIC on the read side. Referring to the switching waveforms in *Figure 4*, at the first enabled rising edge of CLKB (i.e. rising edge of CLKB with ENB HIGH) after $\overline{CSB}$ is LOW, B_{79-0} for CY Standard Mode still contains the previous piece of data read before $\overline{CSB}$ goes LOW. This first enabled CLKB rising edge basically retrieves the first word (W1) from the FIFO, which is latched into the ASIC on the second rising edge of CLKB. For FWFT Mode however, W1 is already on the output data bus at the first enabled rising edge of CLKB, thus the ASIC can latch it in on the same clock edge. The first enabled CLKB rising edge also retrieves W2 and updates $\overline{EF}$/OR flag if necessary. Thus before the arrival of the second enabled CLKB rising edge, W2 would have been available at the output data bus. Therefore, for CY Standard Mode, if the FIFO contains N words, ENB will need to be asserted for the duration of N CLKB rising edges, while N+1 CLKB rising edges will be required for the ASIC to read all N words. For the FWFT mode, only N CLKB rising edges are required to read all N words (ENB needs to be asserted for the same duration of N CLKB edges)

Note that retransmit operation only applies to CY standard mode, not in FWFT mode.

Width Expansion Configuration

In the event that there is a need for very wide bus data transfer, such as 128-bit parallel interface, Cypress x80 FIFOs can meet the demand by providing the ability to connect more than one of these FIFOs together. Status flags can be detected from any one FIFO with the exception of the $\overline{EF}$ and $\overline{FF}$ in the Cypress Standard mode and IR and OR in the FWFT mode. This is due to the uncontrolled skew between Read Clock and Write Clock which may assert/deassert the flag of one of the FIFOs one cycle earlier/later than the other FIFO. In CY Standard Mode, this problem can be avoided by ANDing $\overline{EF}$ of every FIFO, and separately ANDing $\overline{FF}$ of every FIFO. In FWFT mode, composite flags can be created by ORing IR of every FIFO, and separately ORing OR of every FIFO. *Figure 5* shows a width expansion using two CY7C480xV25 devices. A79–A0 from each device form a 128-bit wide input bus (SIZE1A and SIZE2A both tied to ground) and B79–B0 from each device form a 128-bit wide output bus (SIZE1B and SIZE2B both tied to ground). Additional FIFOs can be added to achieve larger bus width.

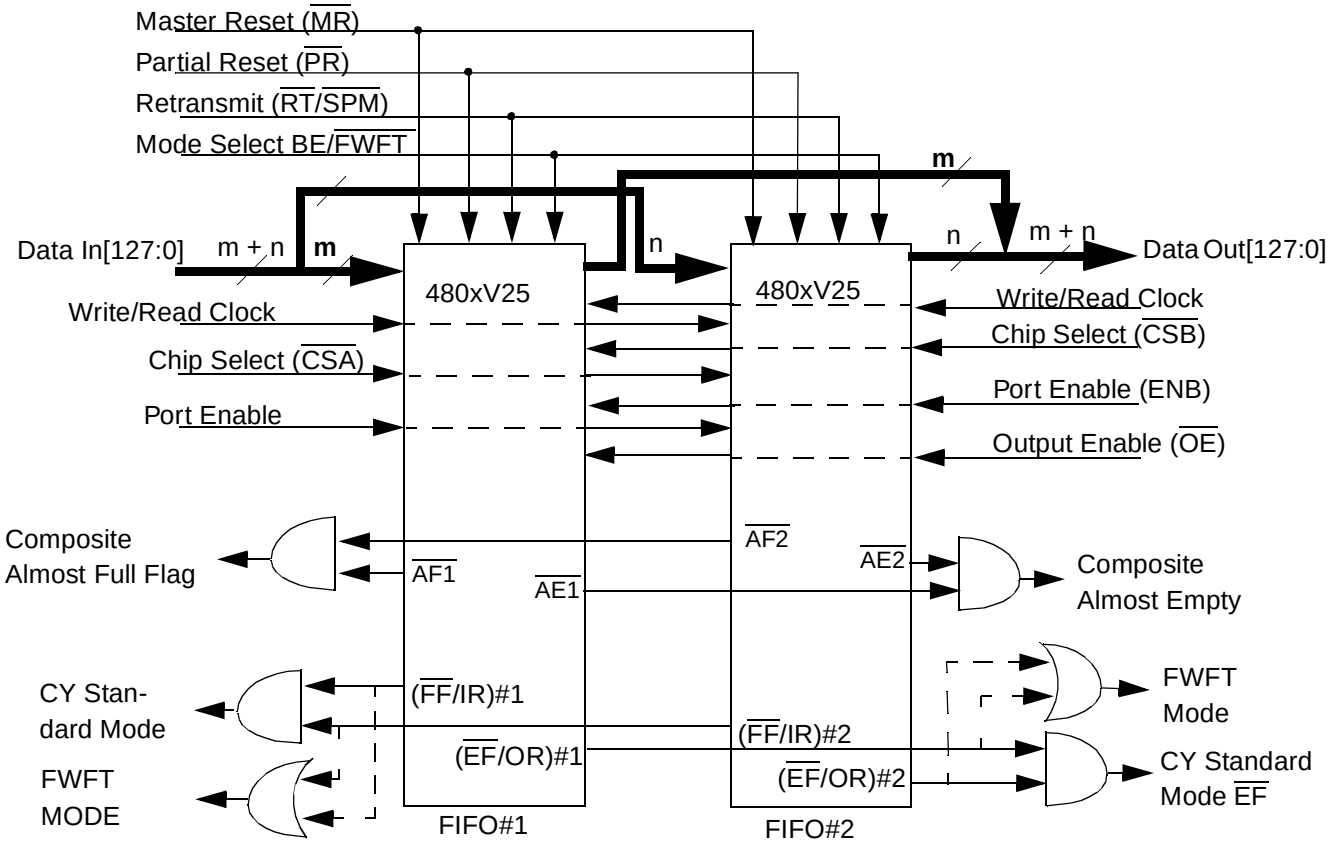

Figure 5. Block Diagram of Two CY7C480xV25 FIFOs in Width Expansion (128-bit Interface).

Depth Expansion Configuration (FWFT mode only)

The CY7C480xV25 can be easily configured to meet the demand for more depth in the memory array required by some applications. For example, two CY7C4808V25 (5-Mb density) FIFOs can be cascaded in depth to achieve a 10-Mb FIFO. In FWFT mode, the FIFOs can be connected in series (the data outputs of one FIFO connected to the data inputs of the next) with no external logic necessary. *Figure 6* shows a depth expansion using two CY7C480xV25 FIFOs. Bus width of 80 bits on both ports is assumed in this example. The state of SIZE can be changed to accommodate bus width of different sizes.

Care should be taken to select FWFT mode during Master Reset for all FIFOs except the last one in the depth expansion configuration. The first word written to an empty configuration will pass from one FIFO to the next until it finally appears at the outputs of the last FIFO in the chain (the reference clock of each FIFO must be free-running). Each time the data word appears at the outputs of one FIFO, that device's OR flag goes HIGH, enabling the write to the next FIFO in line. The first free location created by reading from a full depth expansion configuration will bubble up from the last FIFO to the previous one until it finally moves into the first FIFO of the chain. Each time a free location is created in one FIFO of the chain, that device's IR flag goes HIGH, enabling the preceding FIFO to write a word to fill it. The Reference Clock line should be tied to either CLKA, CLKB or another clock source, whichever is the fastest. This configuration will result in data moving, as quickly as possible, to the end of the chain and free locations to the beginning of the chain.

Retransmit with FIFOs Cascaded in Depth

The retransmit operation cannot be applied when the FIFOs are cascaded in depth, because the data in all the FIFOs (except FIFO1) will be overwritten by the contents of FIFO1.

Programming the Flags with FIFOs in Cascaded Mode

In cascaded mode, the easiest way to program the FIFOs' flags is through serial programming (besides using the default offset values of 8, 16 or 64). Parallel programming can still be done but requires independent control of MR, RT/SPM, FS0/SEN and FS1/SD signals for each FIFO. Basically, FIFO1 should be master reset and programmed first, then the offset values for all subsequent FIFOs are written into FIFO1. When the EF/OR signal of FIFO 1 goes HIGH, then FIFO2 should be master reset independently. After the rising edge of FF/IR (refer to data sheet), FIFO2 will be programmed through FIFO1. Similarly, when the EF/OR signal of FIFO2 goes HIGH then FIFO3 should be master reset independently, and FIFO3 will be programmed through FIFO2 after the rising edge of FF/IR. The same applies all the way to the N^{th} FIFO.

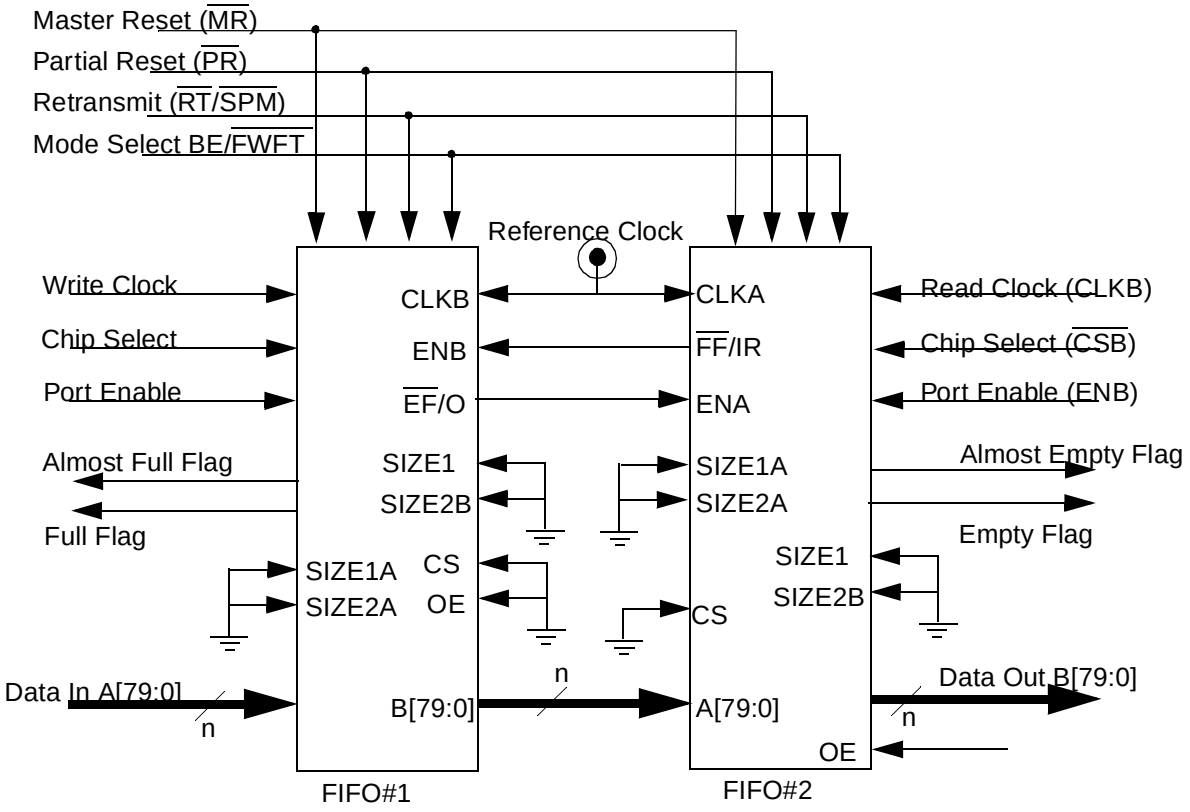

Figure 6. Block Diagram of Two CY7C480xV25 FIFOs in Depth Expansion.

Recommended Design Considerations

Reads/Writes

Performing write and read operations by toggling ENA or ENB rather than CSA and CSB is recommended, i.e. these chip selects should be asserted except when the data lines are meant to be taken off the bus. This is because port A or B is turned off when CSA or CSB is deasserted. This may result in the flags not responding to the clock and not providing the latest status of the FIFO.

Power-up Sequence

The CY7C480xV25 family of FIFOs has three power supply pins: V_{DD} at 2.5V, V_{DDQ} at 1.5V and V_{REF} at 0.75V. V_{DD} should be powered up before V_{DDQ} to ensure proper operation of the device. In addition, as required by the EIA/JEDEC Standard (EIA/JESD8-6), V_{REF} should track V_{DDQ}, which can be implemented through a voltage divider connection as shown in *Figure 7* below.

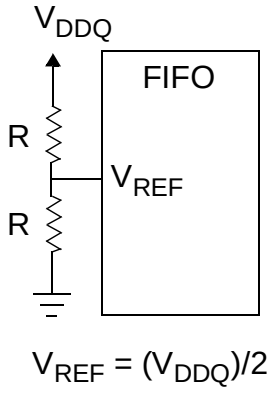

Figure 7. Supply for V_{REF}.

Conclusion

With a bandwidth of more than 32 Gb/s, the CY7C480xV25 family of FIFOs is an ideal solution in frequency decoupling for bandwidth-intensive applications in data communications. These FIFOs provide bus matching capability on both ports to accommodate different bus sizes without the need for any external glue logic. These FIFOs simplify customers' design and provide the system designer with an effective interface to 80-bit and wider devices.

For further information, please visit the Cypress web site at www.cypress.com. The web site also provides the latest data sheets, models, and any related documentation.

© Cypress Semiconductor Corporation, 2001. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress Semiconductor product. Nor does it convey or imply any license under patent or other rights. Cypress Semiconductor does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress Semiconductor products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress Semiconductor against all charges.