Buffer speeds response time of first-in, first-out memory

by Jim Edrington
Applied Research Laboratories, University of Texas, Austin, Texas

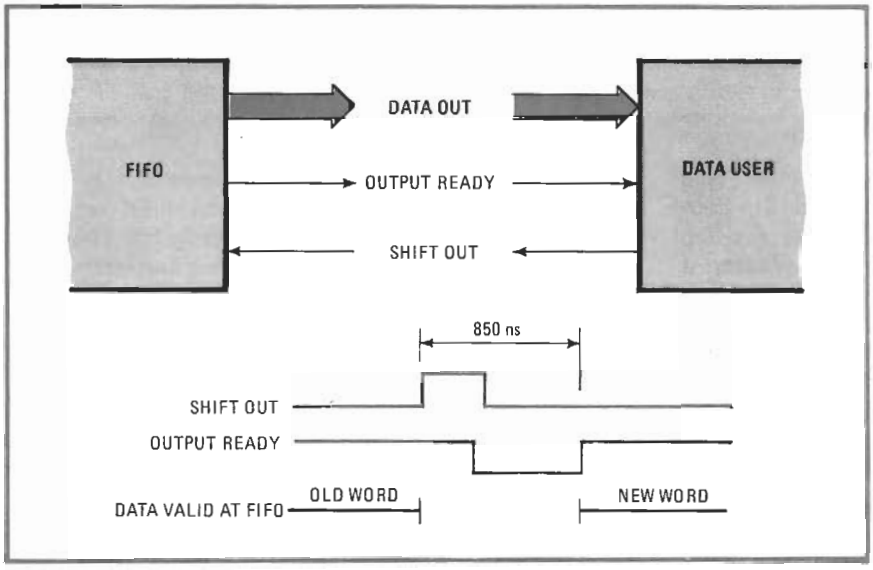
A pair of integrated circuits can interface a first-in, first-out serial memory to a computer or other digital system with which the FIFO is otherwise incompatible. The interface, consisting of a parallel-access shift-register buffer and a dual flip-flop, is connected exactly like the original FIFO interface but has a response time of less than 50 nanoseconds instead of 850 ns.

One of the most popular FIFOs is the 64-word-by-4-bit type 3341, an MOS device made primarily by Fairchild. Its input or output data rate is 1 megahertz, but it takes a long time to respond to a request for data transfer.

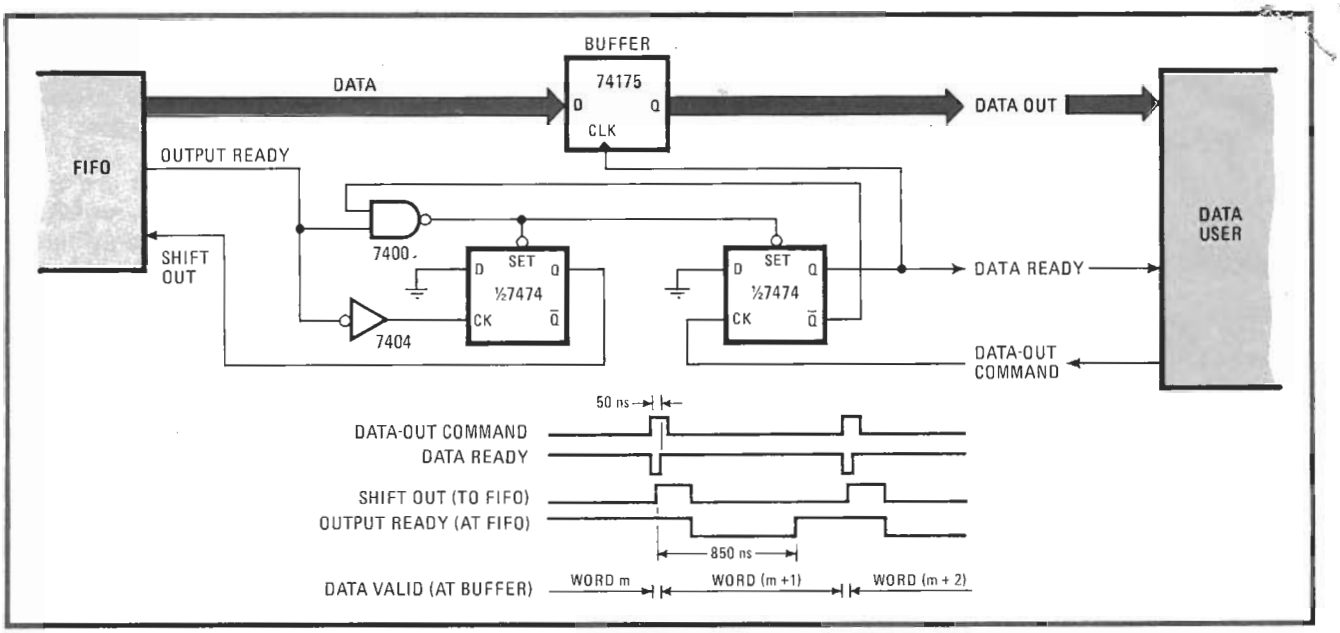
This delay is illustrated in Fig. 1. After reading the word on the FIFO output, the device using the data sends a shift-out signal to the FIFO, asking for a new word. The output-ready line of the FIFO then goes low for 850 ns, during which no valid output word can become available for transfer. When a new word is ready, the output-ready line goes high again.

Such a long delay between a request and an acknowledgment cannot be tolerated by the many digital systems that include fast transistor-transistor logic. An example is the direct-memory-access port of most computers. The DMA operates with a periodicity of 1,000 ns, which is comfortably longer than 850 ns, but unless it receives a data-ready signal within about 200 ns of the previous transfer, it will skip a cycle. In other words, the FIFO transfers words fast enough to work with the typical DMA, but neglects to alert the DMA till too late.

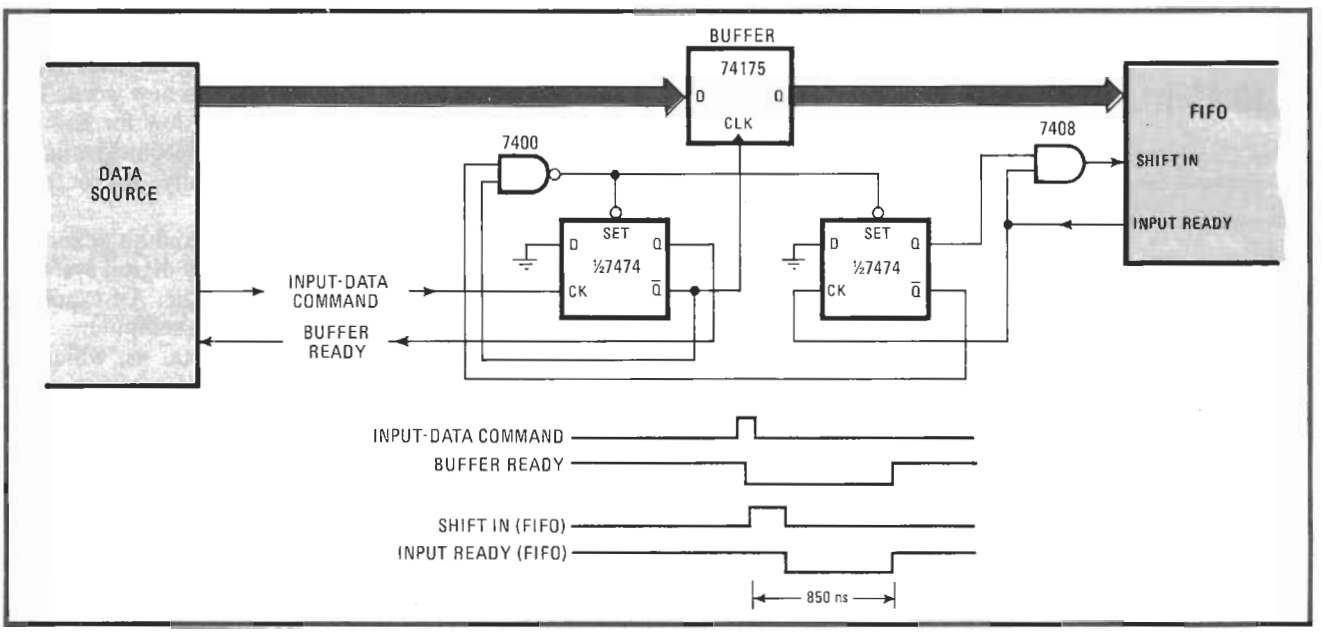
To provide a faster acknowledgment, a TTL parallel-



1. Slow acknowledgment. User system reads data word at FIFO output and sends a shift-out signal telling the FIFO to make a new word available. Then 850 nanoseconds elapse before the FIFO acknowledges that the new word is ready. This delay is too long for fast systems, even though their cycle period is 1,000 ns.



2. Fast acknowledgment. Here, user system reads data word at buffer register, and buffer then takes new word from FIFO and is ready for next data-out command in 50 ns. FIFO has 850 ns to prepare its new output word.



3. Feeding FIFO. Buffer gives prompt acknowledgment to data source that data has been read; when buffer-ready line goes low, source knows that data has been taken. Slow acknowledgment from buffer to FIFO does not hold up a system that requires fast acknowledgment.

access register such as the 74175 is used as a buffer stage between the FIFO and the computer. Two flip-flops provide control for the buffer. The shift-out flip-flop controls the interface between the FIFO and the buffer; it requests a new output word as soon as the buffer register is empty and a new word is available from the FIFO. The data-ready flip-flop takes the place of the FIFO's output-ready signal; it is set true whenever the TTL buffer is loaded with a new word, and it is cleared by a request for data from the DMA. In essence, the TTL buffer is reloaded immediately after each DMA request in time for the following DMA request, as shown in Fig. 2.

A similar circuit may be applied to the input of the FIFO to speed acknowledgment that data has been taken from the data source. In this case, shown in Fig. 3, the buffer register serves to hold the input word until the FIFO can accept it. When the buffer-ready line goes low, the DMA, I/O port, or other data source knows that the input word has been read into the buffer. The source can then immediately change state to validate a new word.

In neither case is the overall data flow rate increased—the upper limit is still the FIFO's internal rate. However, the buffer permits the FIFO to be used with many devices for which its response would otherwise be too slow. □

Designer's casebook is a regular feature in Electronics. We invite readers to submit original and unpublished circuit ideas and solutions to design problems. Explain briefly but thoroughly the circuit's operating principle and purpose. We'll pay $50 for each item published.