

Summary

This Application Note describes the use of an LCA device as an address controller that permits a standard DRAM to be used as deep FIFO.

Xilinx Family

XC3000/XC3100

Demonstrates

Non-linear Counters
Pseudo-random RAM Addressing

Introduction

A bit-serial FIFO buffer is a general-purpose tool to relieve system bottlenecks, e.g., in LANs, in communications, and in the interface between computers and peripherals. Small FIFOs are usually designed as asynchronous shift registers, but a larger FIFO with more than 256 locations is better implemented as a controller plus a two-port RAM, or as a controller plus a single-port RAM, either SRAM or DRAM.

SRAMs are fast and easy to use, but at least four times more expensive than DRAMs of equivalent size. Dynamic RAMs offer lower-cost data storage, but require complex timing and address multiplexing, which makes them unattractive in small designs. For FIFOs with more than 256K capacity, a DRAM offers the lowest cost solution, if the controller can be implemented in a compact and cost-effective way. An XC3020 Logic Cell Array can easily perform all the control and addressing functions with many gates left over for additional features. The XC3020 can be programmed to control one or more DRAMs for a FIFO of up to 16 megabytes, with data rates up to 16 Mbits per second serially or 16 Mbytes per second byte-parallel.

Logic Description

This FIFO DRAM controller comprises the following.

- Input/output buffer with synchronizing logic
- 20-bit Write pointer (counter)
- 20-bit Read pointer (counter)
- 20-bit full/empty comparator
- 10-bit address multiplexer
- Control and arbitration logic

Figure 1 is a block diagram of the FIFO Controller. The Write pointer defines the memory location where the incoming data is to be written, while the Read pointer defines the memory location where the next data can be read. The identity comparator between the address pointers signals when the FIFO is full or empty.

When the Write and Read pointers become identical as a result of a Write operation, the FIFO is full, and further Write operations must be prevented until data has been read out to create space in the memory. If the two pointers become identical as a result of a Read operation, the FIFO is empty and further Read operations must be prevented until new data has been written in. With a single-port RAM, Read and Write operations must be inherently sequential, and there is no danger of confusing the full and empty state, a problem that has plagued some two-port designs.

A straightforward design would use synchronous binary counters for the two pointers, but it is far more efficient to use linear feedback shift-register (LFSR) counters. Such counters require significantly less logic and are faster since they avoid the carry propagation delay inherent in binary counters. LFSR counters have two peculiarities: they count in a pseudo-random sequence, and they usually skip one state, i.e., a 20-bit LFSR counter repeats after $2^{20}-1$ clock pulses. In a FIFO Controller, both these issues are irrelevant; the address sequence is arbitrary, provided both counters sequence identically.

The RAS/CAS multiplexing of the 20-bit address is performed without an explicit multiplexer. Every other bit of the shift-register counter is used to provide the 10-bit address. Before the incrementing shift, these bits are used as the Row address. After incrementing, they are used as the Column address. The Column address of any position is thus identical with the Row address of the following position, but since the binary sequence of a shift register counter is pseudo-random anyhow, this is not a problem.

The address generation logic is shown in Figure 2. With this design, two shift-register counter bits fit into one XC3000-series CLB, with the identity comparator using the combinatorial portion of the same CLB, Figure 3.

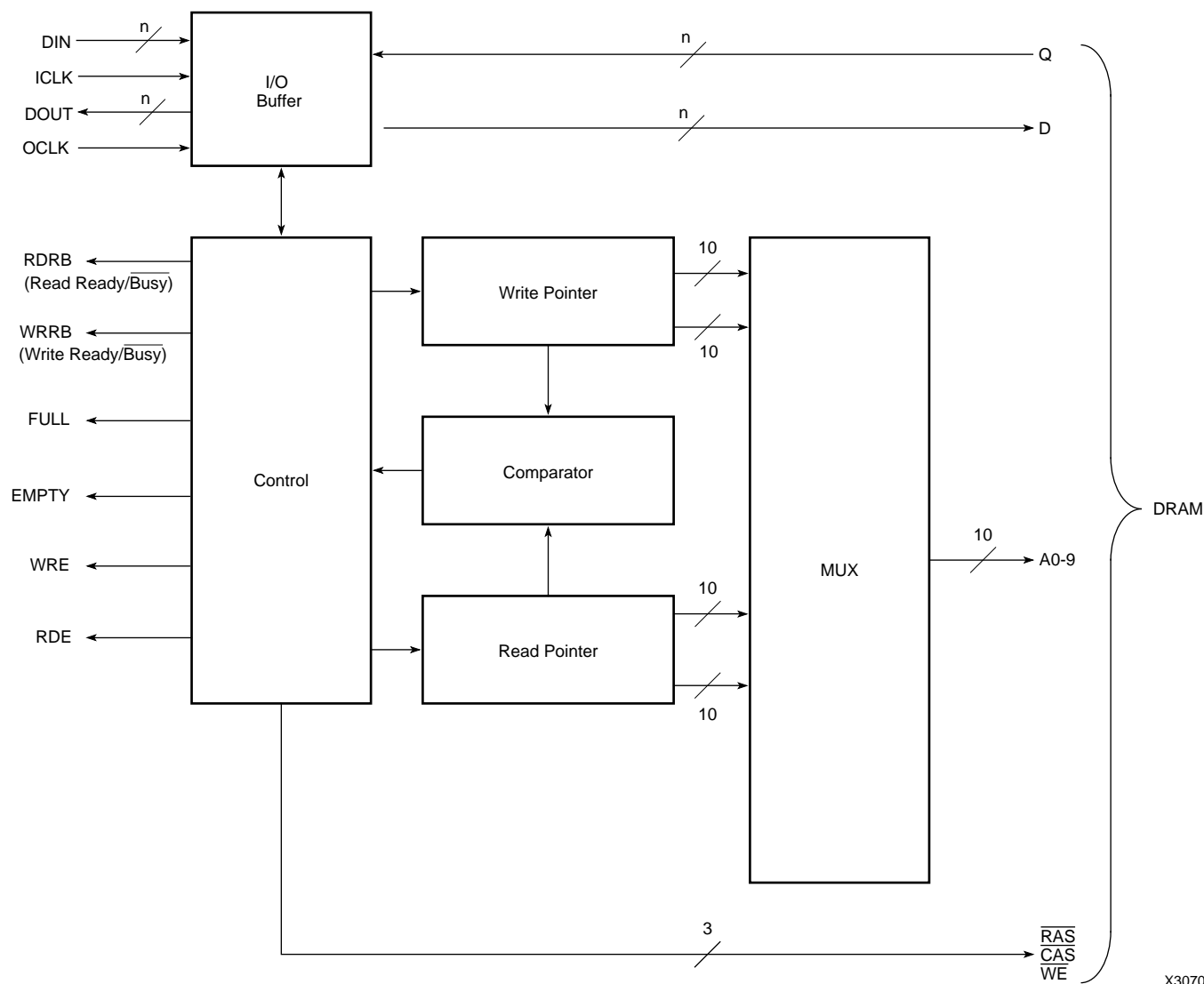


Figure 1. Megabit FIFO Controller in an XC3020

The FIFO controller permits the user to perform totally asynchronous Read and Write operations, while it synchronizes communication with the DRAM. The design takes advantage of the DRAM internal refresh counter by using CAS-before-RAS refresh/address strobes.

Both 20-bit pointers, plus their 20-bit identity comparator, plus the Row/Column multiplexer thus fit into only 20 CLBs; refresh timer and address multiplexer use another 10 CLBs and the data buffer plus control and arbitration logic take another 23 CLBs, for a total of 53, an easy fit in an XC3020.

This design can easily be modified for larger or smaller DRAMs. Other variations that might be considered are:

multiple parallel bits, e.g., byte-parallel operation, interrupt-driven control, multiplexed data for multiple parallel-bit storage, and byte parallel storage with bit-serial I/O. This latter case requires special attention when the FIFO is emptied after a non-integer number of bytes has been entered, and requires direct communication between the input Serial-to-Parallel converter and the output Parallel-to-Serial converter.

This design is available from Xilinx. Call the Applications Hot Line 408-559-7778 or 1-800-255-7778.

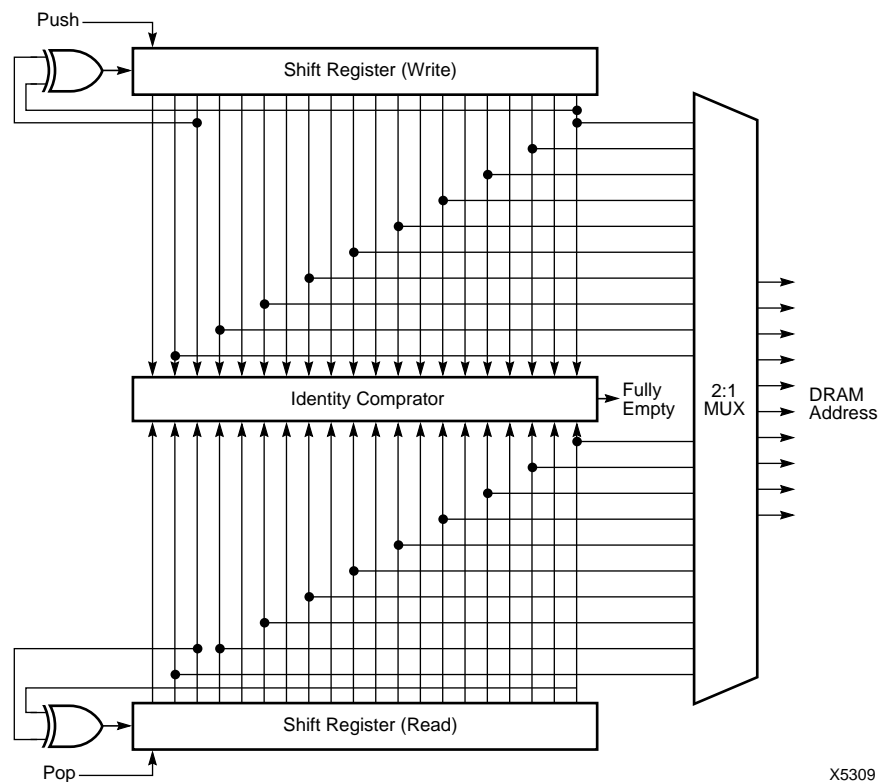


Figure 2. DRAM Address Generation

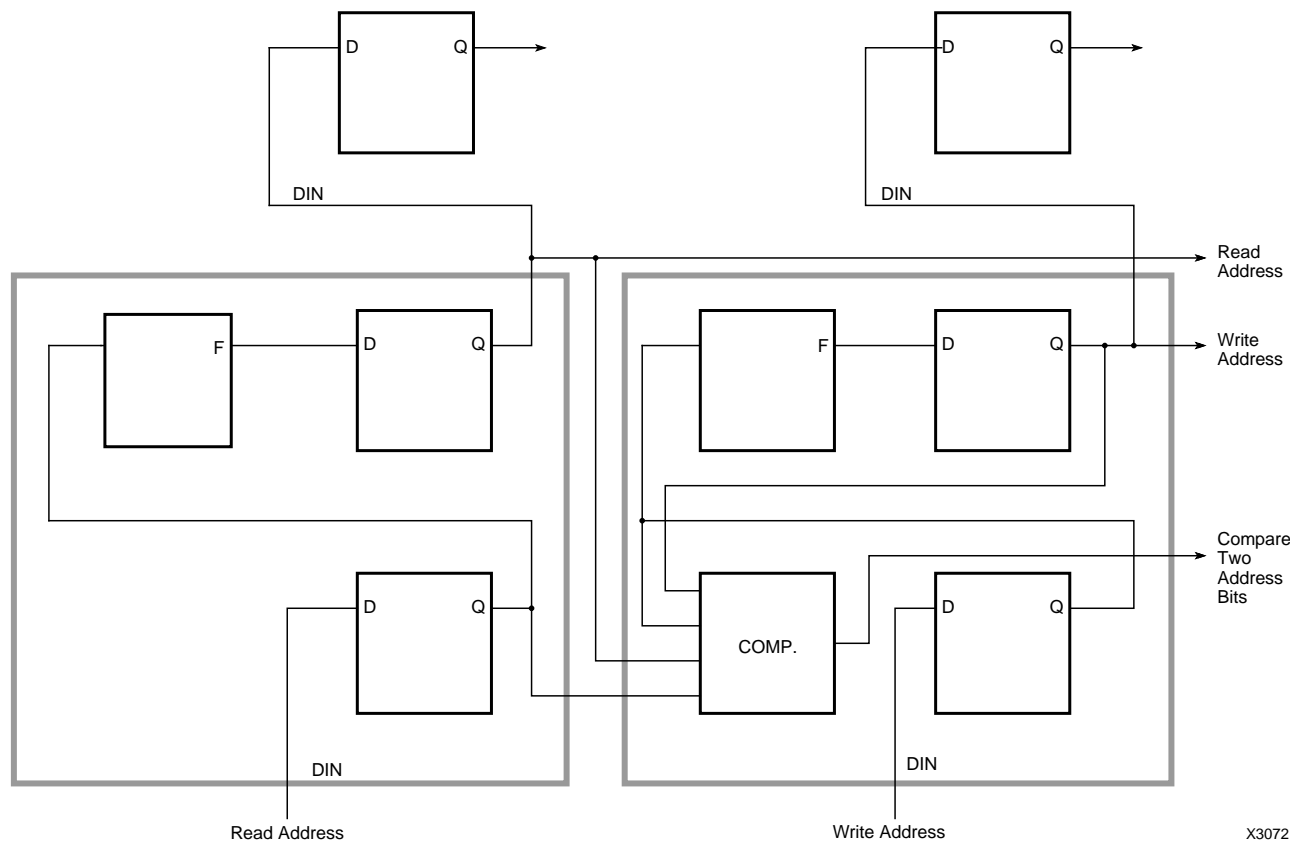


Figure 3. 2-Bit Slice of Two Counters and Comparator in Two CLBs