



XAPP195 (v1.1) August 17, 2004

Implementing Barrel Shifters Using Multipliers

Author: Paul Gigliotti

Summary

The Virtex™-II family of platform FPGAs is the first FPGA family to have multipliers embedded into the FPGA fabric. These multipliers, besides offering very fast and flexible multipliers, supporting several different multiplication modes of operation, can also function as barrel shifters. Specifically, each multiplier can be used as an 8-bit barrel shifter. This application note and accompanying Barrel 32 reference design are intended for design engineers creating general applications.

Introduction

Basic Barrel Shifter

A barrel shifter is simply a bit-rotating shift register. The bits shifted out the MSB end of the register are shifted back into the LSB end of the register. In a barrel shifter, the bits are shifted the desired number of bit positions in a single clock cycle. For example, an eight-bit barrel shifter could shift the data by three positions in a single clock cycle. If the original data was 11110000, one clock cycle later the result will be 10000111.

Functionally, since any bit can end up in any bit position, multiplexers are used to place the bits correctly for proper storage. Thus, a barrel shifter is implemented by feeding an N-bit data word into N, N-bit-wide multiplexers. An eight-bit barrel shifter is built out of eight flip-flops and eight 8-to-1 multiplexers; a 32-bit barrel shifter requires 32 registers and thirty-two, 32-to-1 multiplexers, and so on. A schematic representation of an 8-bit barrel shifter is shown in [Figure 1](#).

Eight-bit Barrel Shifter

To implement the eight 8-to-1 multiplexors in an eight-bit barrel shifter, it will require two slices per multiplexer, for a total of 16 slices. In the Virtex-II architecture, this uses four CLBs. It will also require an additional CLB for the registering of the outputs. These can be absorbed into the multiplexer CLBs. Virtex-II devices have embedded multipliers, and the functionality of an eight-bit barrel shifter can be implemented in a single MULT18X18 ([Figure 2](#)). Note, the control bus "SHIFT[7:0]", is a one-hot encoding of the shift desired. For example, 0000 0001 causes a multiplication by one, or a shift of zero; 0000 0010 causes a multiplication by two, or a shift of "1", 0000 0100 causes a multiplication by four, or a shift of "2", and so on.

© 2004 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

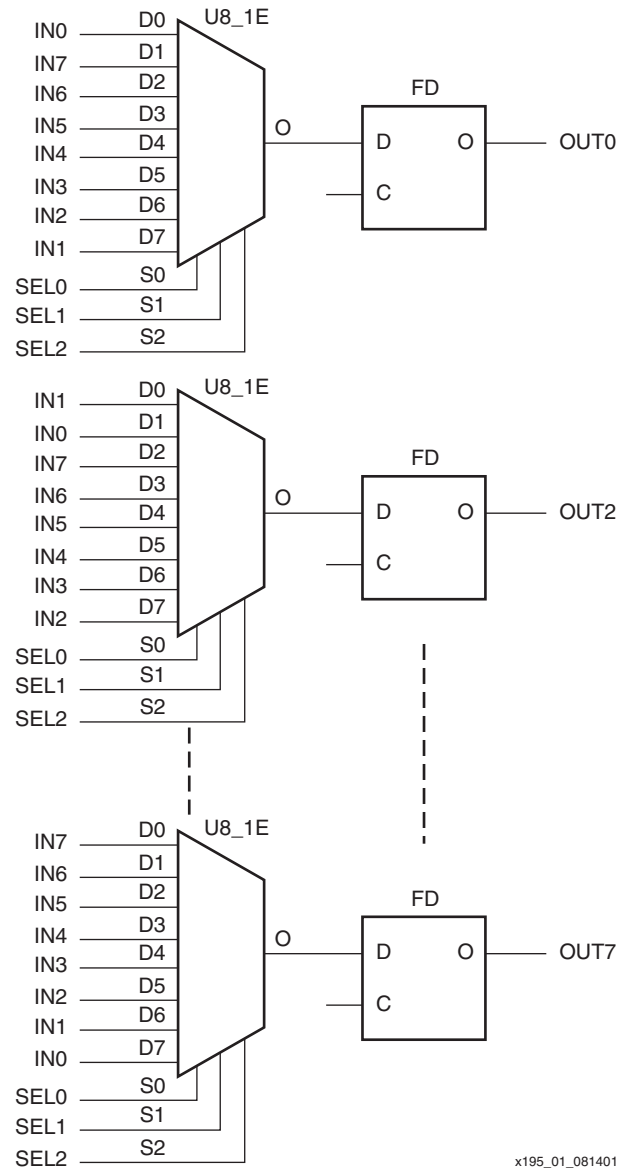


Figure 1: Eight-Bit Barrel Shifter

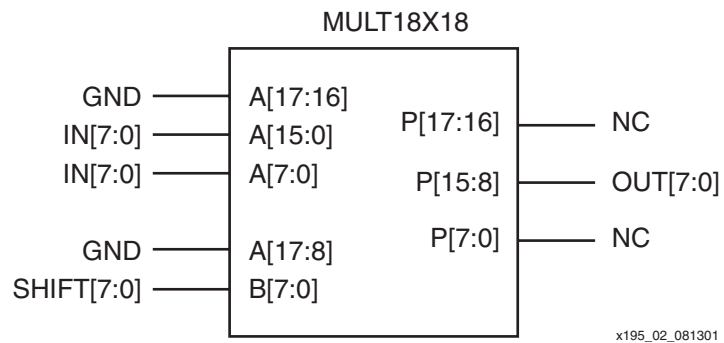


Figure 2: MULT18X18

Single-Cycle, 32-Bit Barrel Shifter

As previously mentioned, a 32-bit barrel shifter requires thirty-two, 32-to-1 multiplexers. A 32-to-1 multiplexer can be implemented in a Virtex-II device using two CLBs. Only sixty-four CLBs are required to accomplish all the required multiplexing. By using a Virtex-II multiplier-based barrel shifter, a 32-bit barrel shifter is built using four 8-bit barrel shifters and thirty-two 4-to-1 multiplexers.

The diagram on the left side of Figure 3 is a single-cycle, 32-bit barrel shifter. The input bus is broken down into four 8-bit words. The data is “processed” in two stages. The first stage is built out of the 8-bit barrel shifters. This stage provides the “fine” shifting, moving the bits from adjoining bytes. After the first stage the appropriate bits are stored in a byte, but the bytes need to be reordered. The reordering of the bytes, or “bulk” shifting, is provided in the second stage, shown on the right in Figure 3. As previously mentioned, the 8-bit barrel shifter requires the shift amount to be one-hot encoded. Also, the three LSBs are used to control the fine shifting, and the two MSBs are used to control the bulk shifting.

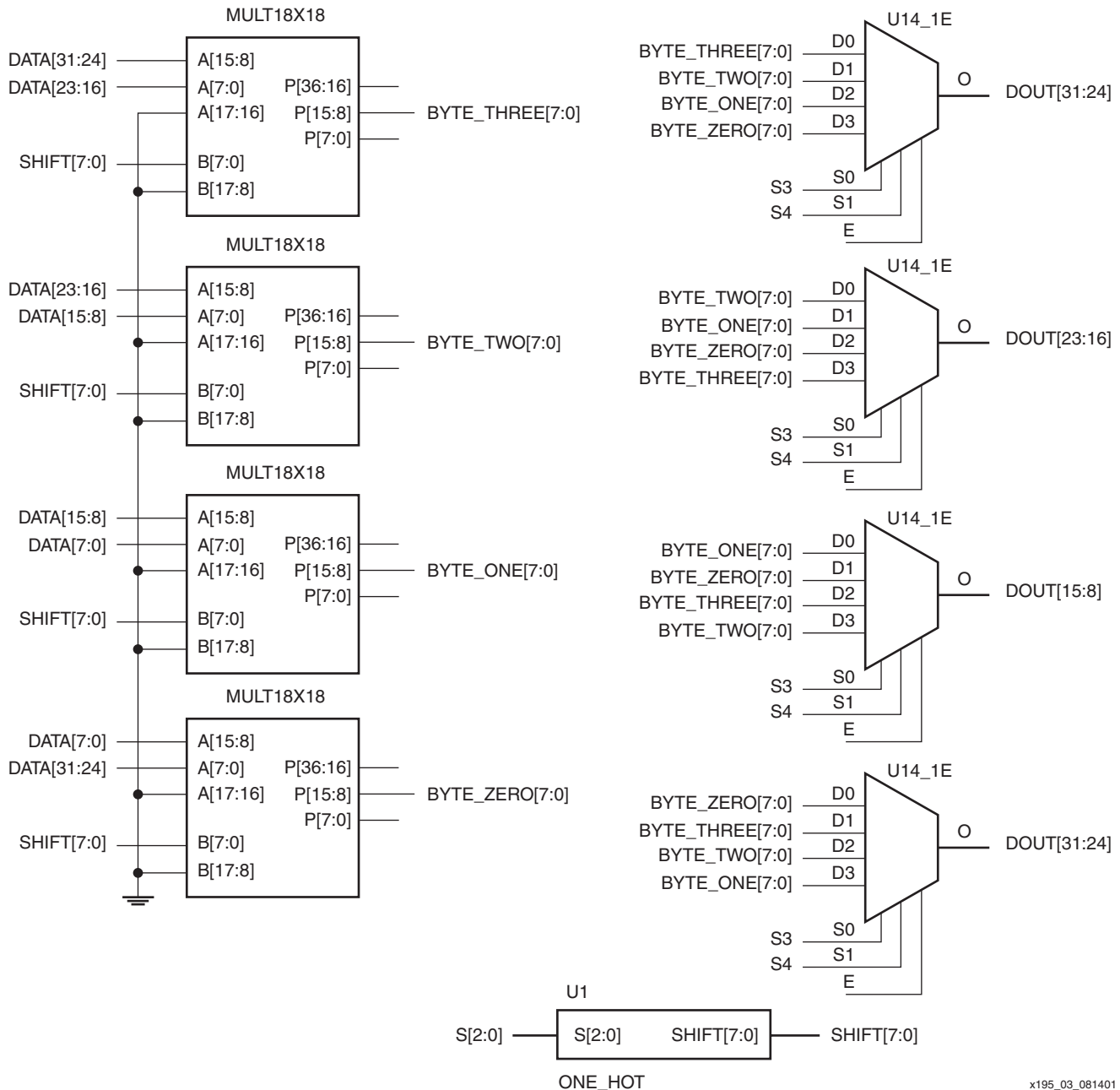


Figure 3: Single-Cycle, 32-bit Barrel Shifter

Four-Cycle, 32-bit Barrel Shifter

At the cost of latency, a more hardware efficient approach is available. The concept shown in Figure 4 is an 8-bit barrel shifter, implemented using one MULT18X18 to move the data into and out of the barrel shifter. The 8-bit barrel shifter is preceded by two 8-bit 4 x 1 MUXs to move the appropriate byte into the 8-bit barrel shifter. The output data from the barrel shifter is then latched into the appropriate byte of the output registers, via clock enables. A small state machine is used to generate the input-multiplexer select signals as well as the output-clock enables.

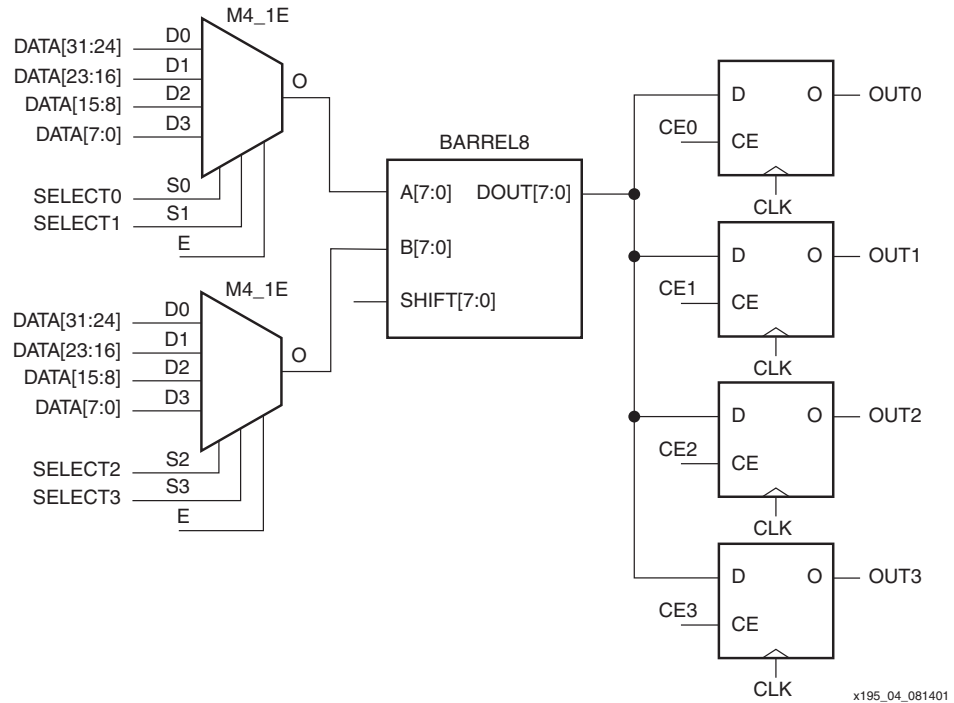


Figure 4: Control

Reference Design

The reference design files for this application note includes VHDL and Verilog code, Benchmark and Simulations, are located at [xapp195.zip](#).

Conclusion

Certain designs show the traditional approach to be more appropriate. Again, the traditional approach requires thirty-two, 32-by-1 multiplexers. Using the Virtex-II fabric, two CLBs configured as a 32-by-1 multiplexer produce a total design requiring 64 CLBs. The multiplier method requires eight LUTs to develop the one-hot shift value, four multipliers and thirty-two, 4-by-1 multiplexers. The eight LUTs used for a one-hot encoder are implemented in a single CLB. Each multiplexer uses a slice, or a total of eight CLBs for thirty-two, 4-by-1 multiplexers. The design is reduced down from 64 CLBs to nine CLBs (and four multipliers). This saves design real estate, but some placement flexibility is lost due to the locking of the barrel shifters to specific multiplier locations.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
07/20/04	1.0	Initial Xilinx release.
08/17/04	1.1	Minor edit to “Reference Design” section.