FIFOs Using Virtex-II Shift Registers

Author: Lakshmi Gopalakrishnan

XAPP256 (v1.3) January 5, 2005

Summary

The shift registers available in Virtex-II™ series devices, including Virtex-II Pro™ devices, are ideal when building synchronous FIFOs. By using the flexibility of the shift register LUT primitive (SRL16), FIFOs can be built with any width with a 1-bit resolution. With cascaded SRL16 shift registers (SRLC16), a flexible depth in multiples of 16 is available.

Introduction

This application note describes a synchronous FIFO built using the SRL16 shift registers. This includes synthesizable code for configuring FIFOs of any width and depths of 128 and 256 bits.

The SRL16 shift registers are ideal for building smaller synchronous FIFOs. These shift registers are actually configured using the Look Up Table (LUT) and do not use flip-flops available in the slice. Hence, they become useful in a design with little logic left to build the FIFO and make better utilization of Virtex-II resources.

Virtex-II SRLC16 Shift Register

See Figure 1 for a diagram of the Virtex-II SRLC16 Shift Register.

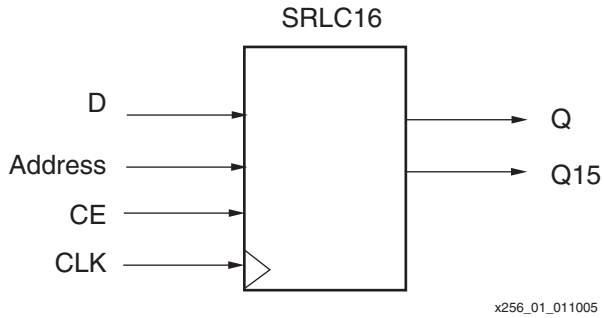

Figure 1: **SRLC16 Shift Register**

FIFOs Using the SRLC16 Shift Registers

Figure 2 is a block diagram of a synchronous FIFO. A binary up/down counter is used to generate the READ and WRITE addresses. Table 1 lists the port definitions for a synchronous FIFO design. The address is incremented for a write and is decremented for a read. The design is also used for simultaneous read and writes.

Table 1: **Port Definitions**

Signal Name	Port Direction	Port Width
CLK	Input	1
SINIT	Input	1
WR_EN	Input	1
RD_EN	Input	1

Table 1: **Port Definitions *(Continued)***

Signal Name	Port Direction	Port Width
DATA_IN[N:0]	Input	N+1
DATA_OUT	Output	N+1
FULL	Output	1
EMPTY	Output	1
DATA_COUNT[C:0]	Output	C+1

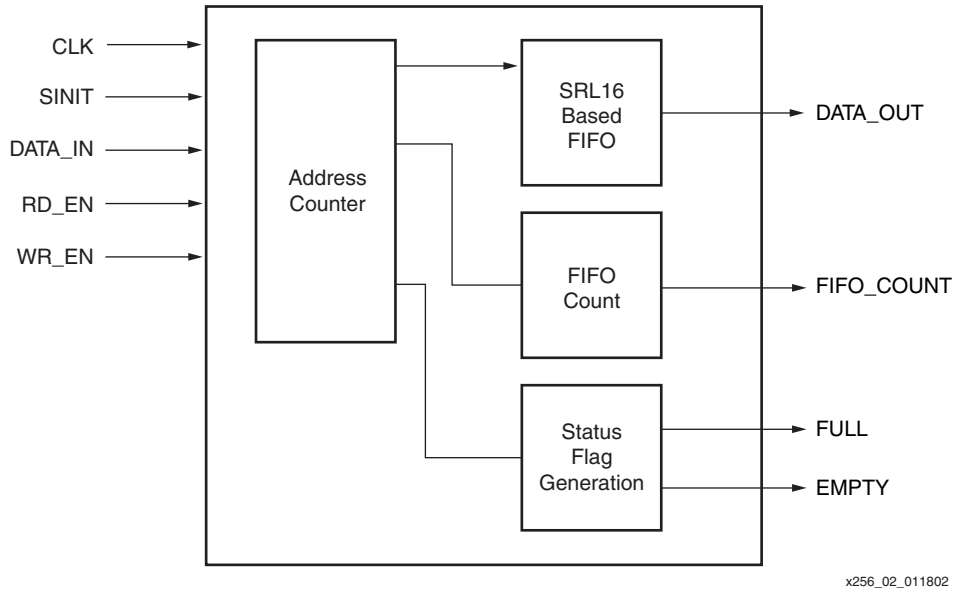

Figure 2: **Synchronous FIFO Using SRLC16 Shift Registers**

Synchronous FIFO Operation

To perform a write, the write enable signal is driven high prior to a rising clock. Before performing a write check the FIFO to make sure it is not FULL. The address for the write is provided by the address counter. After the data has been written, the address counter increments the address by one.

To perform a read, the clocking of data into the FIFO is disabled. During each READ, the address from the address counter is decremented by one. The registered data to be read is available on the output of the MUX in the SRL16 register. It is available on the data output port. To show how full the FIFO is, the address counter is also used as a data counter.

The FULL flag is generated when the FIFO is full and no more data can be written into the FIFO. For a 128-bit deep FIFO, the FULL flag is generated when data is written into the 127th address location and the Write Enable is High. The EMPTY flag signals when the FIFO is empty and no more data can be read from the FIFO. This is generated while the last address location is being read.

Design Resources

By using one CLB and cascading eight SRLC16 shift registers, a 128-bit deep FIFO is generated (Figure 3). Using the multiplexers F5, F6, and F7, multiplexes the output of the four shift registers. Similarly a 256-bit deep FIFO is generated using two CLBs and one more

multiplexer, (F8). Generate synchronous FIFOs of any width by instantiating as many cascaded shift registers as required.

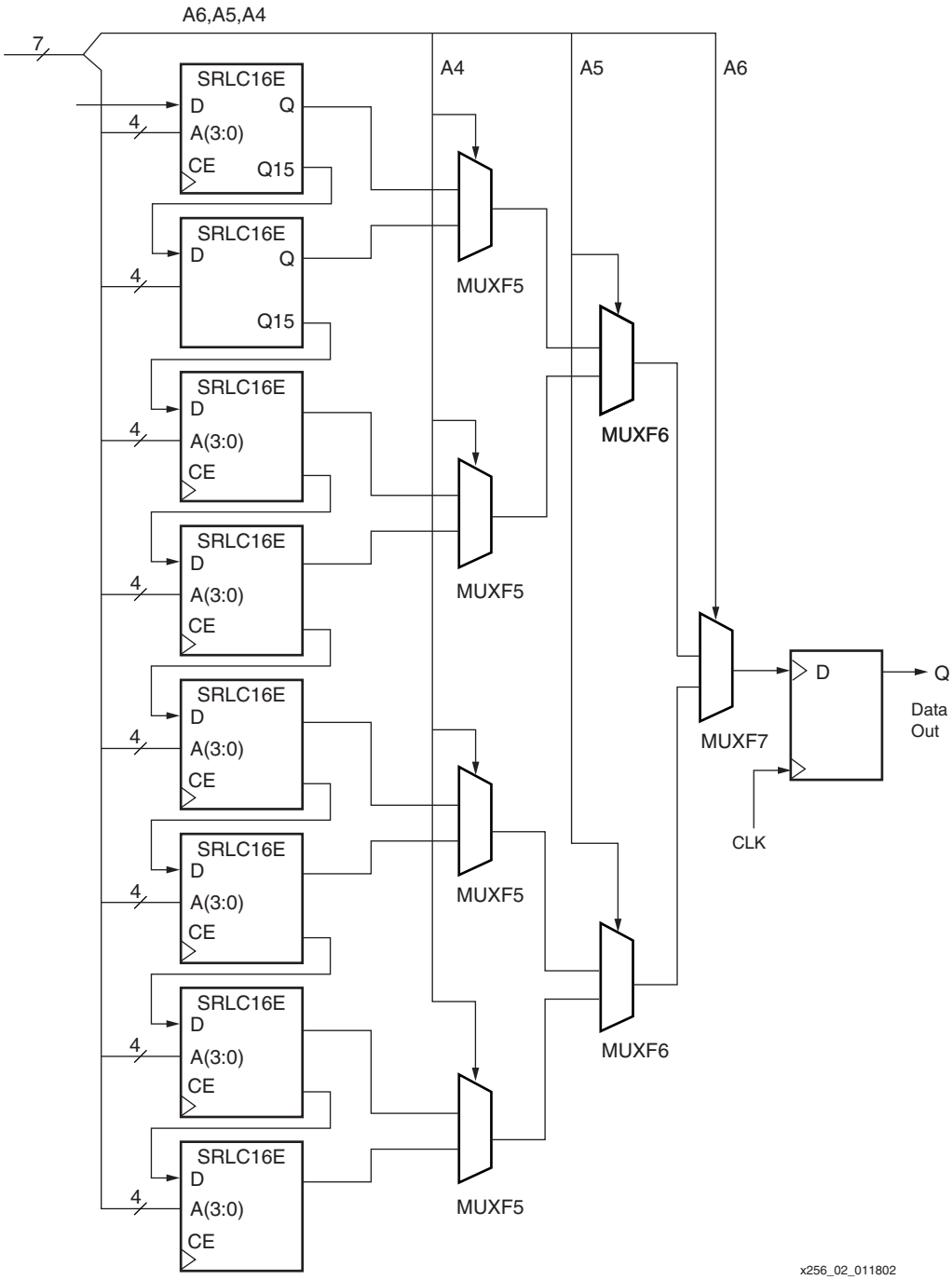

Figure 3: **Generating a 128-Bit Deep FIFO Using SRLC16E Modules**

Reference Designs

The application note provides designs with 128-bit and 256-bit deep FIFOs of any width. The reference design is available in VHDL.

Design	Description
FIFO128.vhd	128-bit deep synchronous FIFO
FIFO256.vhd	256-bit deep synchronous FIFO

Conclusion

Building synchronous FIFOs using the SRLC16 shift registers is an ideal solution for data storage with limited resources. The FIFOs built using the Virtex-II shift registers serve as efficient and fast synchronous FIFOs.

Revision History

The revision history for this document is as follows:

Date	Version	Revision
01/15/01	1.0	Initial Xilinx release.
01/18/01	1.1	Modified Figure 2.
01/21/02	1.2	Updated for Virtex-II Series, including Virtex-II Pro devices.
01/05/05	1.3	Removed reference to Verilog files.