Data-Width Conversion FIFOs using the Virtex Block SelectRAM Memory

XAPP205, October 25, 1999 (Version 1.1)

Application Note: Nick Camilleri

Summary

The Virtex FPGA series provides dedicated on-chip blocks of 4096-bit dual-port synchronous RAM (Block SelectRAM+™). The Block SelectRAM feature is ideal for use in FIFO applications. This application note describes how to create a common-clock (synchronous) version and an independent-clock (asynchronous) version of a FIFO for data-width conversion with different width Read and Write data ports.

Xilinx Family

Virtex™ FPGAs (XCV series)

Introduction

This application note is an enhancement to XAPP131 (170MHz Synchronous and Asynchronous FIFOs using the VIrtex Block SelectRAM Memory). In general, only the changes from XAPP131 will be covered. The reader is strongly encouraged to read XAPP131 before proceeding in this application note.

This application note describes four different data-width conversion FIFOs. The first has a common clock with a 255x16 Write port and a 1020x4 Read port. The second has a common clock, a 1020x4 Write port, and a 255x16 Read port. The last two are similar FIFOs with independent Read and Write clocks. The signal names in parenthesis are a reference to the name in the Verilog code.

Table 1: Port Definitions - Common-Clock design

Signal Name	Port Direction	Port Width
clock_in	input	1
fifo_gsr_in	input	1
write_enable_in	input	1
write_data_in	input	16 or 4
read_enable_in	input	1
read_data_out	output	4 or 16
full_out	output	1
empty_out	output	1
fifocount_out	output	4

Synchronous Design using Common Clocks

For the synchronous version of the data-width conversion FIFO the control logic is very similar to the synchronous control logic in application note XAPP131. Binary counters are used instead of Linear Feedback Shift Registers (LFSRs) for both the read (read_addr) and write (write_addr) address counters. Switching to binary counters makes the simulation easier to read and does not impact the timing performance, since the address counters are not in the critical path. Because the address widths are different, the Read and Write addresses can not be mapped one for one. The difference in width between these two addresses is referred to as a minor address (read/write_addr_minor). The higher 8 bits of the addresses are the same on the write side and on the read side. For example, for a 255 x 16 Read port and a 1020 x 4 Write port FIFO, the Read address requires 8 bits, and the Write address requires 10 bits. Therefore, the upper 8 bits of each address are compared in the flag control logic, and the extra 2 bits are the minor address. A block diagram of the Common Clock Design is shown in Figure 1. The timing diagram is shown in Figure 2.

To perform a Read, the read enable signal (read_enable) is driven High prior to a rising clock edge. The read data signal (read_data) will be presented on the outputs during the next clock cycle. To do a Burst Read, simply leave read_enable High for as many clock cycles as desired. The last word has been read when Empty becomes active after a Read. If another Read is attempted, read_data will be invalid.

To perform a Write, the write data signal (write_data) must be present on the inputs, and the write enable signal (write_enable) must be driven High prior to a rising clock edge. As long as the Full flag is not set, data will be written into the FIFO. To do a Burst Write, the write_enable is left High, and new write_data must be available every cycle.

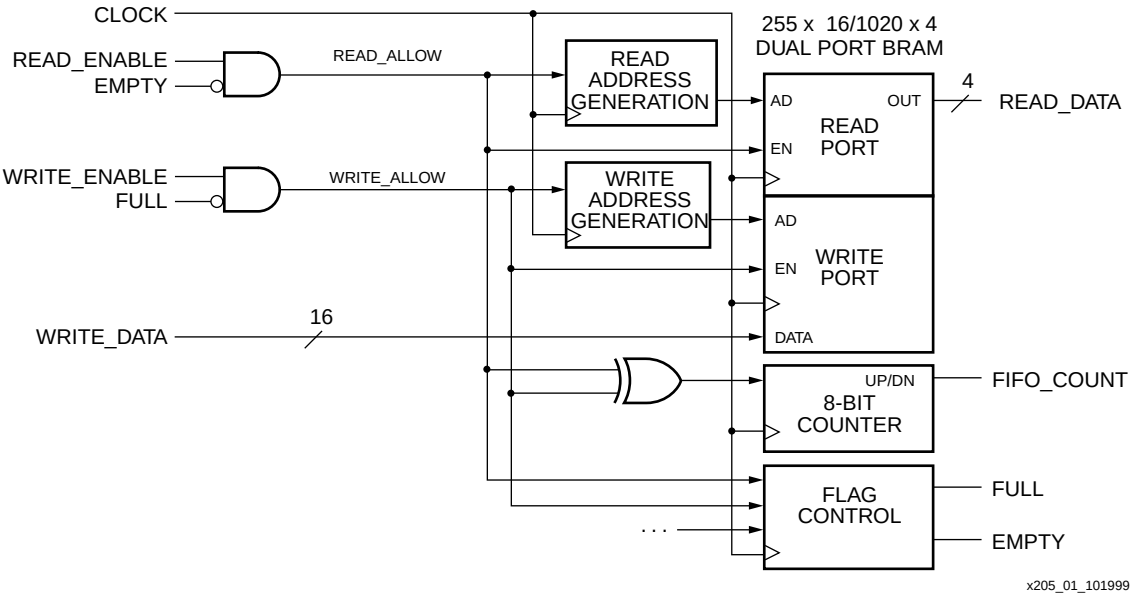


Figure 1: Block Diagram for a Common Clock design

To Read or Write 4-bit data, a secondary enable signal for the control logic is needed (read/write_allow_minor). The corresponding read/write_allow signal for that port will only go active when every 4th nibble is read/written. This signal controls the read/write address counter.

All the initialization values for the address registers have been updated for the change to 8-bit addressing.

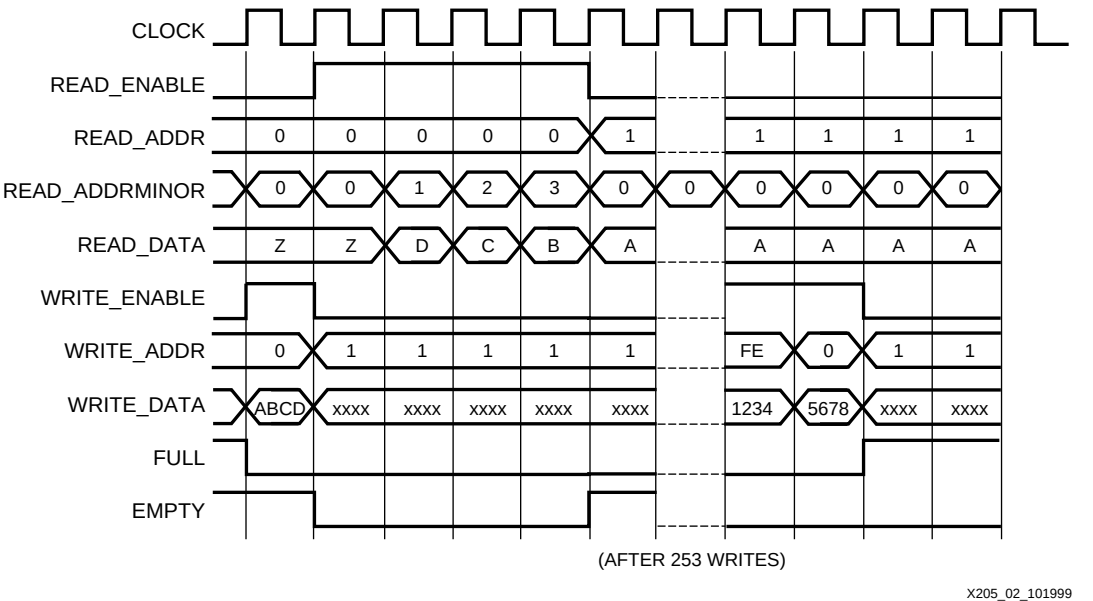


Figure 2: Timing Diagram for Common Clock Design

Table 2: Port Definitions - Independent-Clock design

Signal Name	Port Direction	Port Width
write_clock_in	input	1
read_clock_in	input	1
fifo_gsr_in	input	1
write_enable_in	input	1
write_data_in	input	16 or 4
read_enable_in	input	1
read_data_out	output	4 or 16
full_out	output	1
empty_out	output	1
fifostatus_out	output	4

FIFO Design using Independent Clocks

The control logic for an independent clock version relies heavily on the Gray code address counters and equality comparators to perform the deceptively tricky empty/full flag generation. The minor addresses are also included in the control logic, for the previously mentioned 4-bit operations. Figure 3 is a block diagram of the Independent Clock Design. The corresponding timing diagram is shown in Figure 4.

The control logic for the FIFO status outputs have been updated from XAPP131 to be a more accurate capacity indicator for the FIFO. The fifostatus_out signal now represent 1/2 full, 1/4 full, etc., as shown in Table 3.

The fifostatus_out outputs have a one-cycle latency for write operations, and a two-cycle latency for reads.

Table 3: FIFO Status Out Signal Descriptions

FIFO status:	bit 3	bit 2	bit 1	bit 0	...means FIFO is:
	1	1	1	1	15/16 Full
	1	1	1	0	7/8 Full
	1	1	0	1	13/16 Full
	1	1	0	0	3/4 Full
	1	0	1	1	11/16 Full
	1	0	1	0	5/8 Full
	1	0	0	1	9/16 Full
	1	0	0	0	1/2 Full
	0	1	1	1	7/16 Full
	0	1	0	0	3/8 Full
	0	1	0	1	5/16 Full
	0	1	0	0	1/4 Full
	0	0	1	1	3/16 Full
	0	0	1	0	1/8 Full
	0	0	0	1	1/16 Full
	0	0	0	0	< 1/16 Full

Reference Designs

Each of the four different data-width conversion FIFOs have been simulated using the ModelTech Simulator with individual test bench designs. The designs have not been tested in hardware.

The Verilog files described in Table 4 are located at:

http://www.xilinx.com/support/troubleshoot/htm_index/app_xapp.htm

Table 4: Reference Design Names and Descriptions.

Design	Description
fifoctlr_ccmw1.v	Common Clocks, 1020x4 Read, 255x16 Write
fifoctlr_ccmw2.v	Common Clocks, 255x16 Read, 1020x4 Write
fifoctlr_icmw1.v	Independent Clocks, 1020x4 Read, 255x16 Write
fifoctlr_icmw2.v	Independent Clocks, 255x16 Read, 1020x4 Write
testb_ccmw1.v	Testbench for fifoctlr_ccmw1.v
testb_ccmw2.v	Testbench for fifoctlr_ccmw2.v
testb_icmw1.v	Testbench for fifoctlr_icmw1.v, with Read clock faster than Write clock
testb_icmw1b.v	Testbench for fifoctlr_icmw1.v, with Write clock faster than Read clock
testb_icmw2.v	Testbench for fifoctlr_icmw2.v, with Read clock faster than Write clock
testb_icmw2b.v	Testbench for fifoctlr_icmw2.v, with Write clock faster than Read clock

Conclusion

The dedicated blocks of true, dual-port, synchronous RAM in the Virtex family is used in this application note to create effective data-width conversion FIFOs.

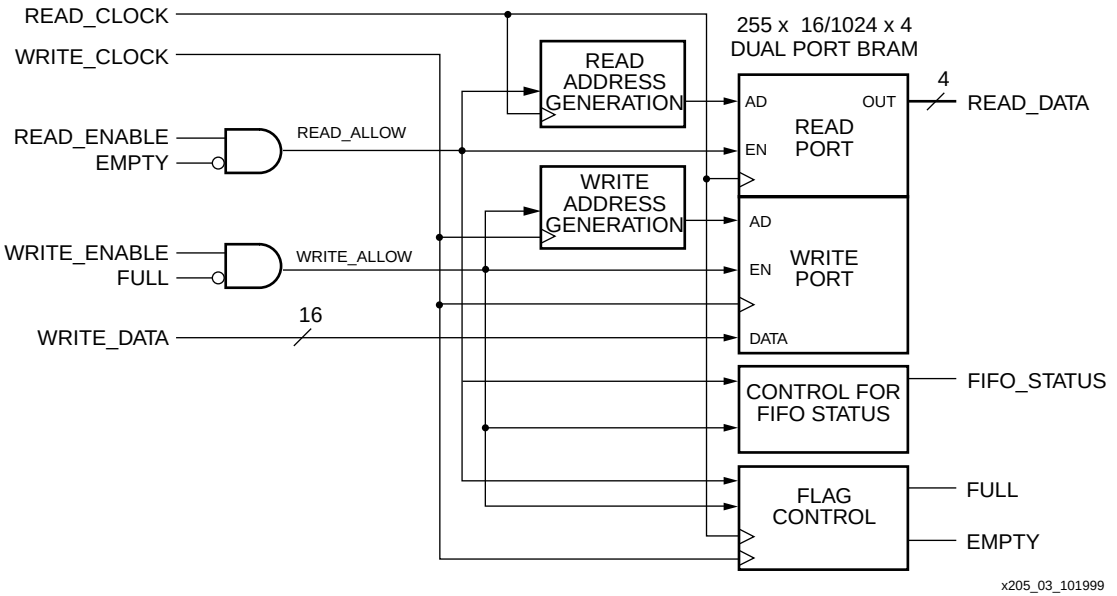
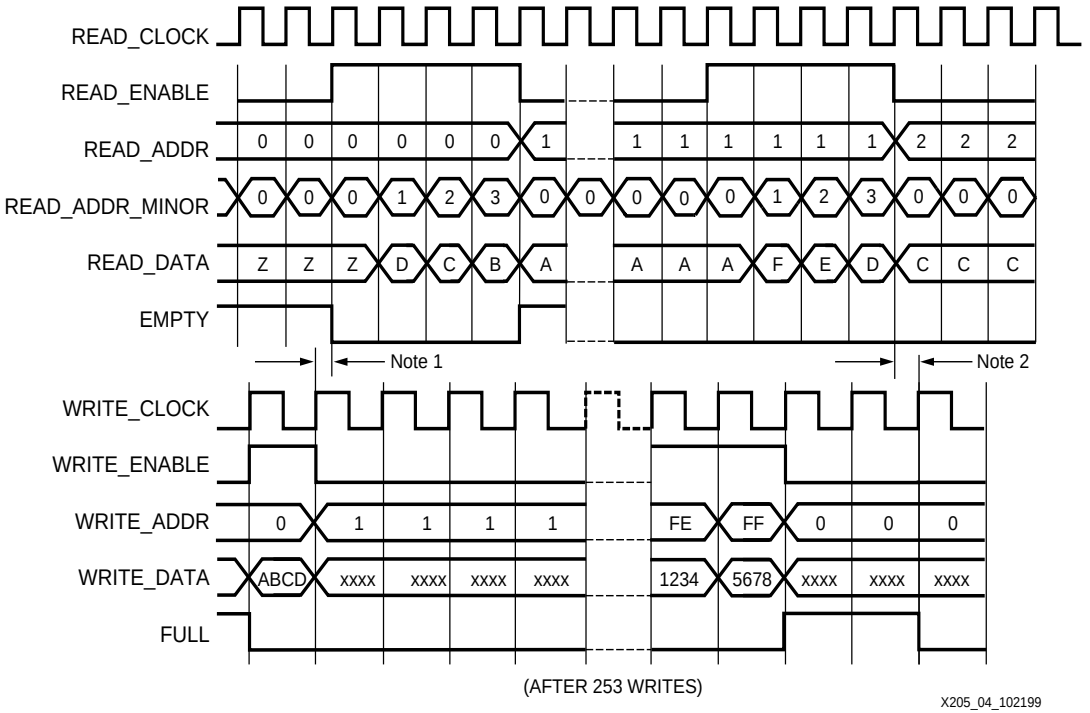


Figure 3: Block Diagram for an Independent Clock Design



Note 1: EMPTY will go low if the Write Gray Addresses meet the set-up time before the rising edge of READ_CLOCK. If not, then EMPTY will go low when this requirement is met.

Note 2: FULL will go low if the Read Gray Addresses meet the set-up time before the rising edge of the WRITE_CLOCK. If not, then FULL will go low when this requirement is met.

Figure 4: Timing Diagram for an Independent Clock

Revision History

Date	Revision	Activity
9/21/99	1.0	Initial Release
10/25/99	1.1	Revised timing diagrams of Figure 2, Figure 4