



XAPP750 (v1.0) May 24, 2004

QDR II SRAM Local Clocking Interface for Virtex-II Pro Devices

Author: Olivier Despaux

Summary

This application note describes a 200 MHz four-word burst QDR II SRAM interface implemented in a Virtex-II Pro™ XC2VP20 FF1152 –6 device. This implementation uses local clocking for data capture and an automatic calibration circuit to adjust the look-up table (LUT) delay on the local clocks. This circuit uses one digital clock manager (DCM) that can be shared between several memory interfaces running at the same frequency. The design has been implemented using Samsung K7R323684M-FC20 memory devices. The aggregate throughput for this interface is 28.8 Gb/s, or 400 Mb/s per pin for a 36-bit write bus and a 36-bit read bus operating in DDR mode at 200 MHz.

This application note provides the following key information for the QDR II SRAM interface:

- Description of the controller and interface in the FPGA for the QDR II SRAM interface
- Details of the interface timing
- Overview of the implementation

Interface Description

QDR II SRAM Devices Overview

Specification details of QDR II SRAM devices are available through the QDR SRAM consortium, constituted by Cypress, Renesas, IDT, NEC, and Samsung. The QDR Consortium website is <http://www.qdrsr.com>. Refer to the “Appendix B: References” section for documentation on QDR SRAM.

QDR SRAM Interface Overview

This section presents the characteristics of the QDR II SRAM interface. It presents a block diagram of the interface, the state machine for the write and read four-word burst operation control, and the design specifications.

Interface Block Diagram

The signals from the user interface are labeled *USER_** and are summarized below:

- For memory writes, 36-bit data is presented on *USER_DWL* and *USER_DWH* in Single Data Mode (SDR). The corresponding write address is presented on *USER_A_WRITE*. A write is indicated when the Write Command signal, *W_n*, is asserted Low. The data and address are only synchronously registered before the transfer to QDR II SRAM to reduce the latency to a minimum value. A FIFO can be implemented on the transmit side as needed.
- For memory reads, the read address is presented on *USER_A_READ*. A read is indicated when the Read Command signal, *R_n*, is asserted Low. Resulting 72-bit data is output on *USER_Q* in SDR Mode at half the interface frequency (100 MHz for a 200 MHz interface). The data is the output of an asynchronous FIFO, so data can be read with any clock or frequency, using the *USER_DATA_VALID* indicator (asserted High to indicate valid data).
- *USER_CLK* is the clock input for the interface design.

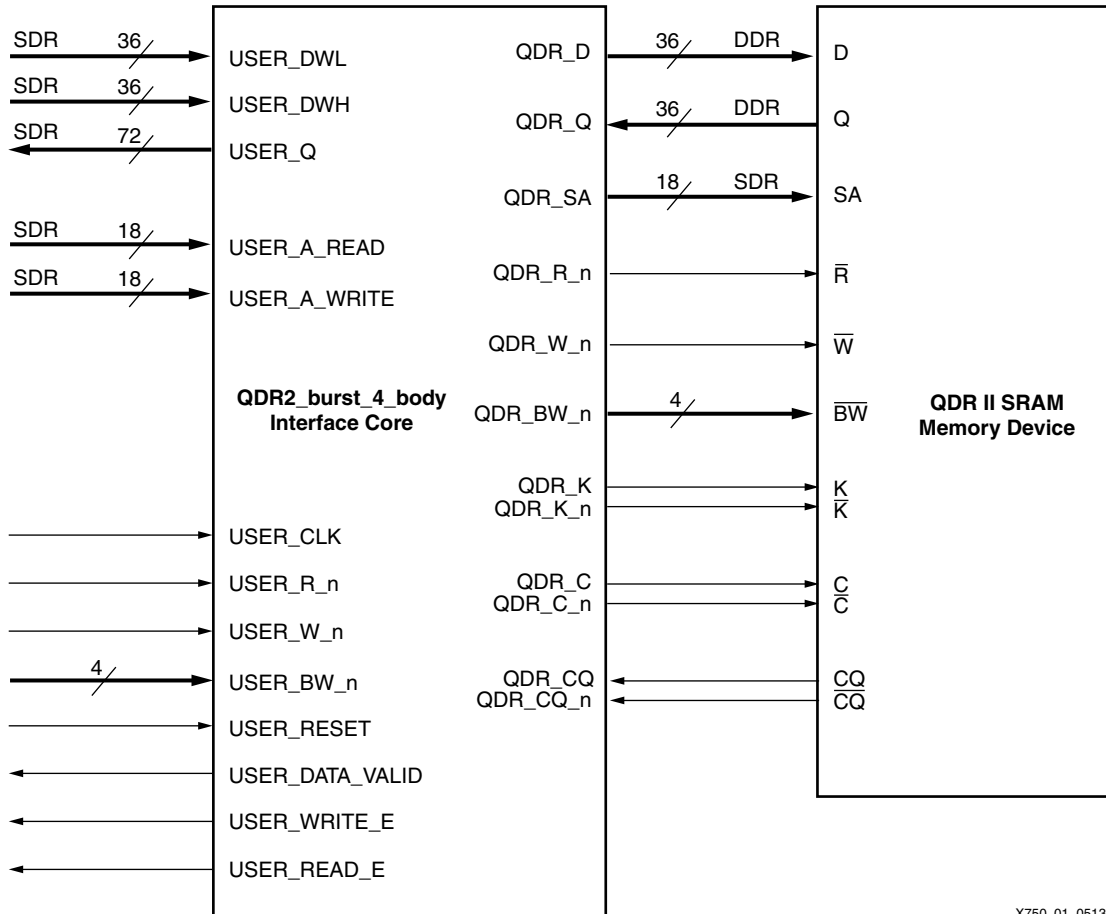
© 2004 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change 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.

- *USER_BW_n* is the word enable input for the memory.
- *USER_RESET* is the reset input for the interface design. It has a synchronous mechanism in this design and is an input to the DCM reset.
- *USER_WRITE_E* and *USER_READ_E* are output signals from the controller state machine that indicate the status of the write and read operations.

The signals directed to memory are self-explanatory and in compliance with the QDR II SRAM specification. Refer to the “[Appendix B: References](#)” section for references to the memory data sheets.

Figure 1 shows a block diagram of the QDR II SRAM reference design.

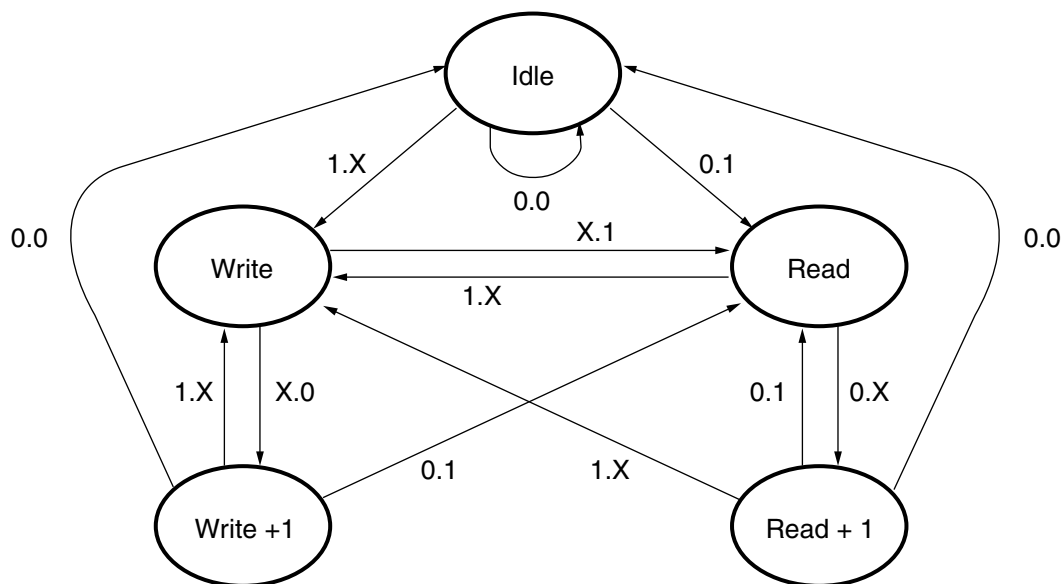


X750_01_051304

Figure 1: Top-Level Architecture Block Diagram of Reference Design

Controller State Machine

Figure 2 shows the state machine of the controller for read and write accesses to the QDR II SRAM. This process is required only for four-word burst interfaces because there are specific access rules for these devices. With four-word burst devices, each write and read access can occur every other clock period. However, write and read accesses can be interfaced at each clock period. It is possible to simplify the state machine to three cycles (Idle, Write, Read), but in this case the two extra stages do not provide specific outputs.



X.X = Write_command.Read_command, where '1' indicates an active command

X750_02_051304

Figure 2: Controller State Machine (Read and Write Commands)

Design Specifications

Table 1 summarizes the specifications for the QDR II SRAM reference design.

Table 1: Reference Design Specifications

Parameter	Specification
Maximum Operating Frequency	200 MHz
QDR II SRAM Operation	Four-word burst
Bus Width	36 bits
Signal Voltage	1.8V
Memory Device	Samsung K7R323684M
Local Clocking	Compliant with placement rules for local clocking
HDL Language	Verilog 2001
Write User Interface	Write / read command availability (user OK read / OK write), no user FIFO on transmit, asynchronous FIFO on receive
User Interface Timing	Compliant with QDR II four-word burst specification

Implementation Description

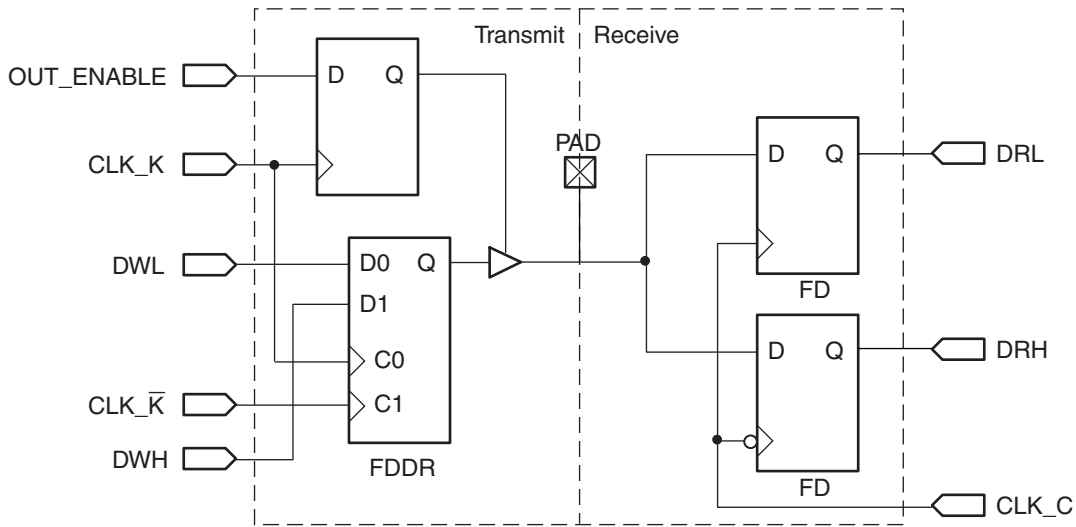
Physical Layer

The physical layer transmits and receives all the signals to and from the memories. The physical layer performs the following major functions:

- Writes data into the memory
- Reads data from the memory
- Provides all the necessary control signals
- Transfers the read data clock domain from the memory domain to the FPGA domain

Virtex-II Series DDR I/O Blocks

Virtex-II SelectIO™ inputs and outputs support DDR operations on both transmit and receive sides. Figure 3 shows the I/O block's physical implementation. The FDDR is used in this design. The FDs in the I/Os of the Receive block are not used, and the FDs in the CLBs are used when data is captured on local clocks.

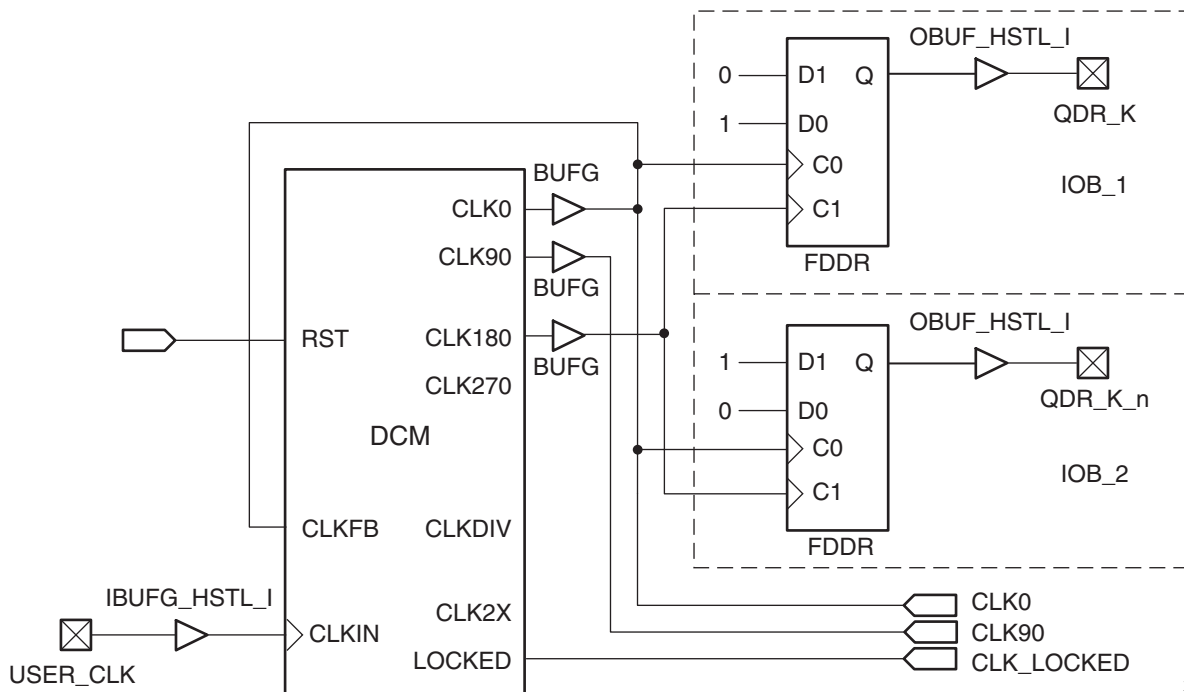


X255_03_051304

Figure 3: I/O Implementation in Virtex-II Series

Clocking Scheme

The QDR II SRAM standard requires sending data, address, and control signals center-aligned with respect to the forwarded synchronous clock. This requirement is achieved easily by using the clock-forwarding scheme to provide the clock to the FPGA. This technique removes the clock-to-out parameter of the FPGA because the control, address, data, and clock paths are identical. Figure 4 presents the current implementation of the clock-forwarding scheme.



X750_04_051304

Figure 4: Clock Forwarding Scheme

The reference design uses the phase-shifted outputs of the DCM to clock the interface on the transmit side. This configuration gives the best jitter and skew characteristics. However, there can be a worst case of ± 140 ps of skew between the CLK0 output and any other main output of any DCM across the chip. Using two outputs generates a worst-case skew of 140 ps, whereas using all four outputs can generate up to 280 ps of skew.

There are other possibilities, such as multiplying the clock frequency using one DCM and using another DCM to generate a phase-shifted version. This latter possibility generates more jitter than the solution presented here.

When designing-in Virtex-II Pro devices, local clock inverters are not recommended when using the DCM. Use instead the CLK0 and CLK180 outputs of the DCM to generate the signals sent to the QDR II SRAM. In cases where local inversion is used on clocks, refer to XAPP685.

Using Multiple QDR II SRAMs on the Same Bus

There are three reasons for using multiple QDR II SRAMs on the same bus:

1. To increase the density of the memory resource.
2. To increase the depth of the memory resource.
3. To divide the speed of the interface by using multiple devices to achieve a given bandwidth.

A design can require using QDR II SRAMs for one or more of the above reasons. The increase in density increases the bandwidth of the memory resource when considering a fixed clock frequency.

This reference design interfaces to one external QDR II SRAM device, but it is possible to adapt it to interface to more devices with only slight modifications to the control signal processing and the data bus width. QDR SRAM memory vendors have published technical papers on clocking strategies that cover both cases and give guidelines on clocking strategies.

Data Path

Transmit Side: Write Operations

At the memory device I/Os, data must be center-aligned with the clock. [Figure 5](#) shows the basic waveforms used to write data to the memory.

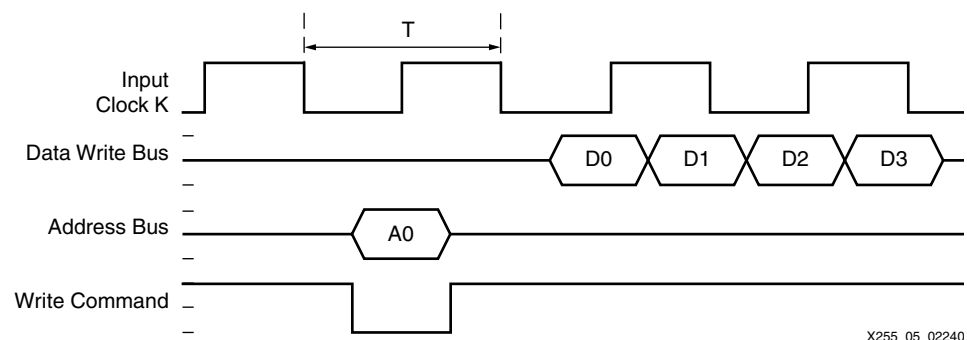
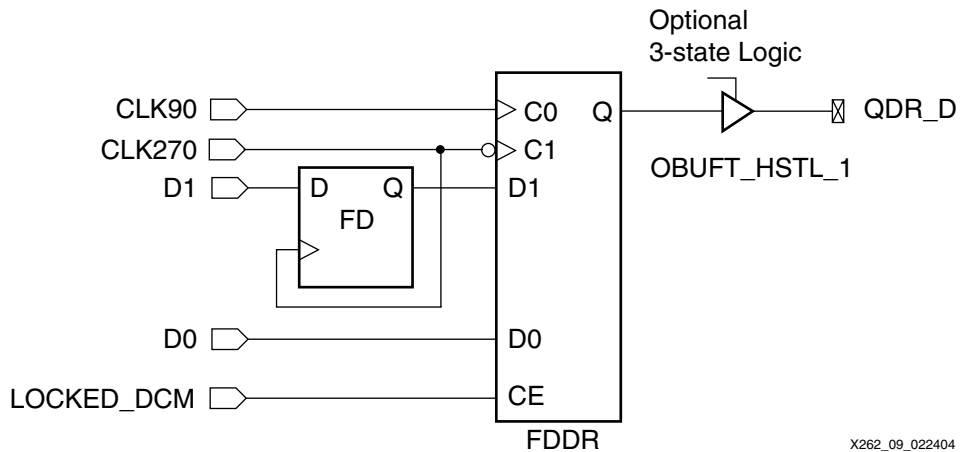


Figure 5: QDR II SRAM Write Waveforms (4 Word Burst)

QDR II SRAMs have setup and hold times that must be met. The center alignment of the data with respect to the clock is done using the CLK90 DCM output, considering that the forwarded clock is generated with the CLK0 output. This is implemented in the module or architecture called 'write_burst_4'. [Figure 6](#) is a schematic that illustrates the basics of this path.

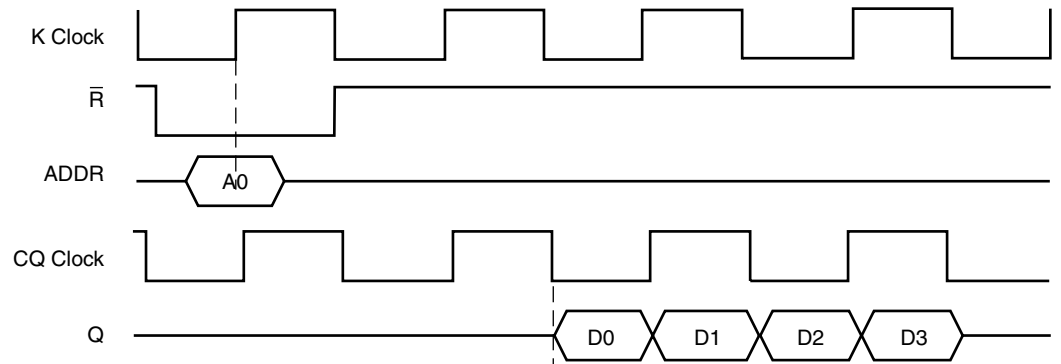


X262_09_022404

Figure 6: Transmit Path Schematics

Receive Side: Read Operations

Because this reference design is implemented in a source-synchronous fashion, the read data is captured by the FPGA using the clock provided by the memory device. QDR II SRAMs are true source-synchronous devices when using the CQ echo clock at the memory controller to execute data capture. Figure 7 shows the read waveforms at the memory device pins.

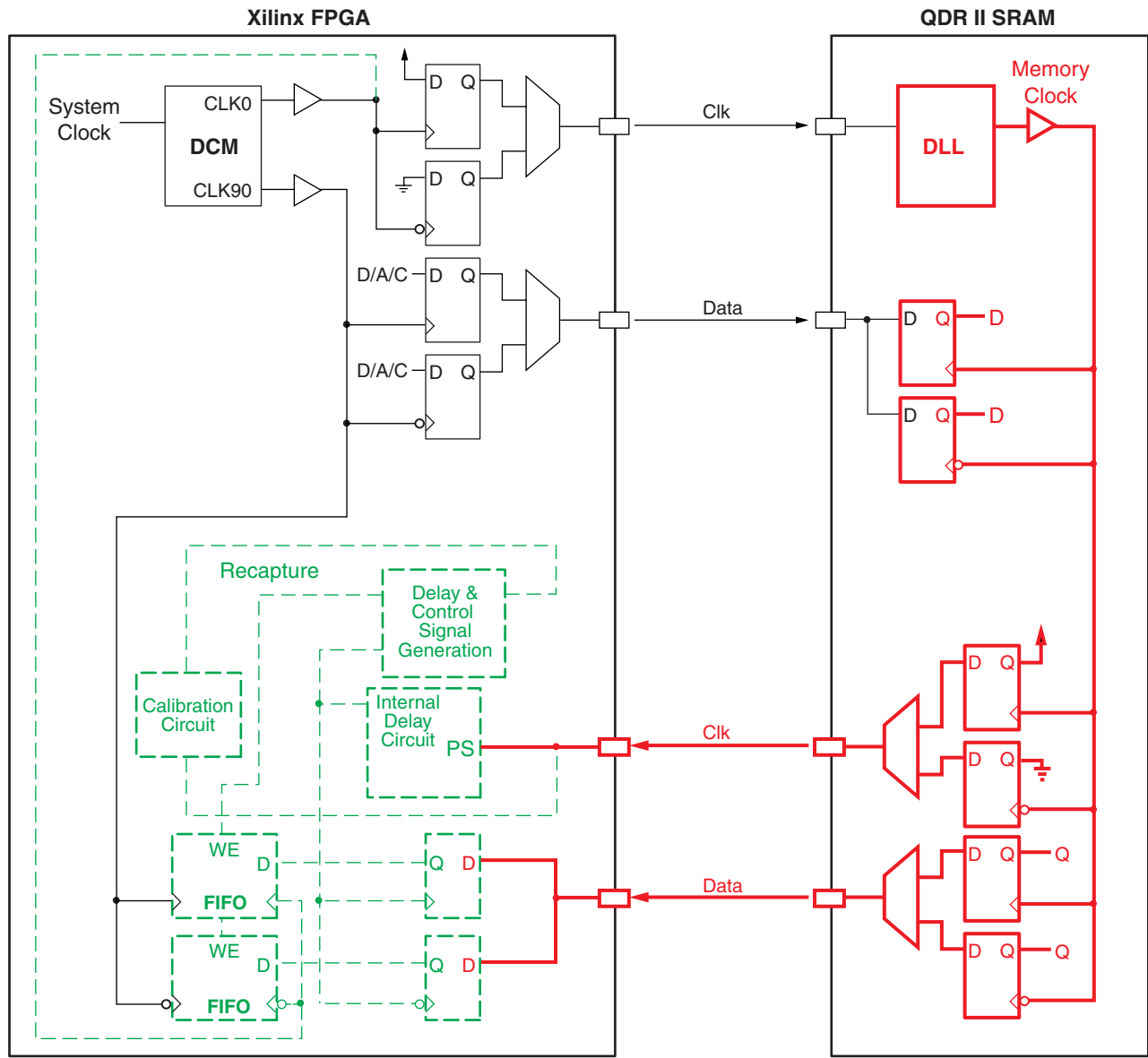


X750_07_022404

Figure 7: QDR II SRAM Read Waveforms

Clock Connections and Relations

Figure 8 shows the clock organization for this reference design. The FPGA clock domain is represented with thin, black lines, the QDR II SRAM with thick, red lines, and the FPGA data capture clock with dotted, green lines.



X750_03_051304

Figure 8: High-Level Diagram of the Clock Connections

Timing

Clocks, Address, and Control Signals

The FPGA generates all the clock and control signals for reads and writes to memory. The memory clocks typically are generated using a Double Data Rate (DDR) register.

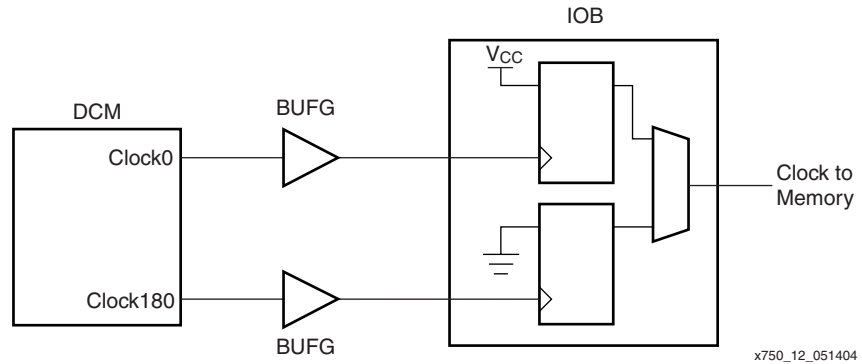


Figure 9: Clock Generation for Memory Device

As shown in Figure 9, a Digital Clock Manager (DCM) generates the clock and its inverted version. Generating the clock this way has several advantages:

- The data, control, and clock signals all go through similar delay elements while exiting the FPGA.
- Clock duty cycle distortion is minimal when global clock nets are used for the clock and the 180° phase-shifted clock.

All address and control signals are registered and output at the IOB. The address and control signals are registered using a clock that is 180° shifted from the clock signal to the memory. Such an approach enables the address and control signals to have additional time margin before they are registered. The address and control signals easily meet the required timing.

Table 2 shows an example timing analysis for a QDR II SRAM interface implemented using an XC2VP20FF1152 FPGA, -6 speed grade.

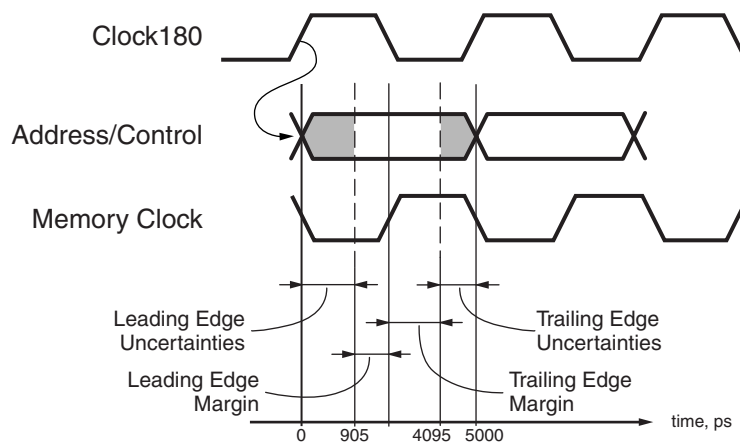
Table 2: Address and Control Signal Margins

Parameter	Value (ps)	Leading-Edge Uncertainties	Trailing-Edge Uncertainties	Meaning
T _{CLOCK}	5000	-	-	Clock period (200 MHz clock frequency)
T _{CLOCK_SKEW_FPGA}	±50	50	50	Minimal skew since right/left sides are being used and the bits are close together. The number from TRACE for the specific clock area used to generate the address and control signals.
T _{PACKAGE_SKEW} ⁽¹⁾	±65	65	65	Using the same bank reduces package skew
T _{SETUP}	600	600	0	Setup time from memory data sheet
T _{HOLD}	600	0	600	Hold time from memory data sheet
T _{PCB_LAYOUT_SKEW}	±50	50	50	Skew between layout lines on the board. Because board skew varies per design, values listed here are examples
T _{PHASE_OFFSET_ERROR_DCM}	±140	140	140	Offset between different phases of the DCM outputs
T _{DUTY_CYCLE_DISTORTION_DCM}	0	0	0	Duty-cycle distortion does not apply because clock and data are sent synchronously
T _{JITTER}	0	0	0	Since the clock and address are generated using the same clock, the same jitter exists in both. Therefore, no jitter needs to be included
Total Uncertainties	-	905	905	
Command Window	3190	905	4095	Worst-case window of 3190 ps

Notes:

- Values from the package skew spreadsheet for XOENP20.

Figure 10 illustrates the address and control signal margins.



X688_04_052404

Figure 10: Address and Control Signal Margins

Data Writes

Table 3 shows the extra margin calculations for write data capture. Figure 11 illustrates the calculated time margin window.

Table 3: Write Data Timing Margin Calculation

Parameter	Value (ps)	Leading-Edge Uncertainties	Trailing-Edge Uncertainties	Meaning
T_{CLOCK}	5000	-	-	Clock period (200 MHz clock frequency)
$T_{\text{CLOCK_PHASE}}$	2500	-	-	Clock phase (50% of clock period)
T_{DCD}	250	-	-	Overall duty cycle distortion of clock to memory (5% of clock period)
$T_{\text{DATA_PERIOD}}$	2250	-	-	Total data period, $T_{\text{CLOCK_PHASE}} - T_{\text{DCD}}$
$T_{\text{CLOCK_SKEW}}$	± 50	50	50	Minimal skew since right/left sides are being used and the bits are close together. The number from TRACE for the specific clock area used to generate the address and control signals.
$T_{\text{PACKAGE_SKEW}}$	± 65	65	65	Skew due to package pins and board layout. This can be reduced further with tighter layout
T_{SETUP}	400	400	0	Setup time from memory data sheet
T_{HOLD}	400	0	400	Hold time from memory data sheet
$T_{\text{PCB_LAYOUT_SKEW}}$	± 50	50	50	Skew between layout lines on the board. Because board skew varies per design, values listed here are examples
$T_{\text{PHASE_OFFSET_ERROR}}$	± 140	140	140	Offset error between different clocks from the same DCM
T_{JITTER}	0	0	0	The same DCM generates the clock and data. Hence they jitter together.
Total Uncertainties	-	705	705	Worst case for leading and trailing can never happen simultaneously
Window	840	705	1545	Total worst-case window is 740 ps

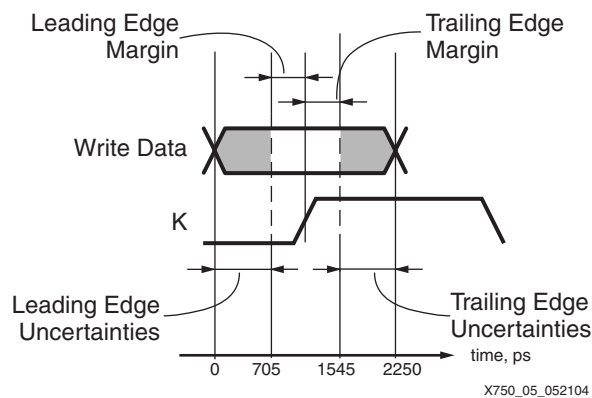


Figure 11: Write Data Margin

Data Captures

Data is captured using the delayed strobe. Data is captured directly in the FPGA fabric slices. [Table 4](#) shows the timing calculation for data capture. [Figure 12](#) illustrates the calculated data valid window.

Table 4: Data Valid Window Calculation

Parameter	Value (ps)	Leading-Edge Uncertainties	Trailing-Edge Uncertainties	Meaning
T_{CLOCK}	5000	-	-	Clock period
T_{PHASE}	2500	-	-	Clock phase (50% clock period)
$T_{\text{MEM_DCD}}$	250	-	-	Overall duty cycle distortion on receive clocks
$T_{\text{DATA_PERIOD}}$	2250	-	-	Total data period, $T_{\text{PHASE}} - T_{\text{MEM_DCD}}$
$T_{\text{CQ_TO_Q_SKEW}}$	± 350	350	350	CQ-to-data distortion from memory data sheet
$T_{\text{PACKAGE_SKEW}}$	± 65	65	65	This parameter depends on the exact package. Because the eight data bits are close together, skew is less than values listed
T_{SETUP}	240	240	0	Setup time from Virtex-II Pro data sheet for -6 part (T_{DICK})
T_{HOLD}	-50	0	-50	Hold time from Virtex-II Pro data sheet for -6 part (T_{CKDI})
T_{JITTER}	0	0	0	Data and strobe jitter together, because they are generated off the same clock
$T_{\text{LOCAL_CLOCK_LINE}}$	± 25	25	25	Observed skew is lower than this value, because loading is light and all bits are close together
$T_{\text{PCB_LAYOUT_SKEW}}$	± 50	50	50	Skew between data lines on the board
Uncertainties	-	730	440	
Window	1080	730	1810	Worst-case window of 1080 ps

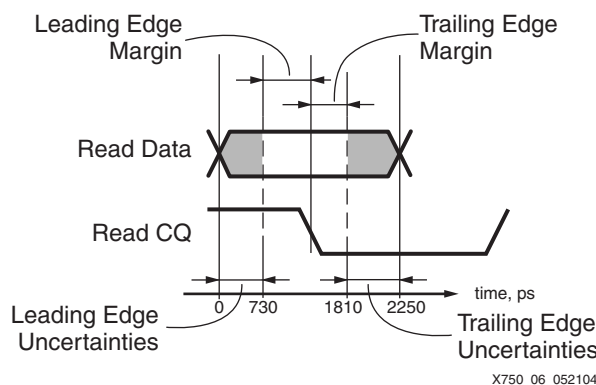


Figure 12: Data Valid Window

Conclusion

This application note describes a 200 MHz four-word burst QDR II SRAM interface implemented in a Virtex-II Pro FPGA. The total throughput for this interface is 28.8 Gb/s, or 400 Mb/s per pin for the 36-bit write bus and 36-bit read bus at 200 MHz.

The details of this reference design including fully synthesizable Verilog code are available in XAPP770C, subject to the terms of the license agreement, at the following website:

<http://www.xilinx.com/memory>.

Appendix A: Definition of Terms

Table 5 defines the mnemonics used in this application note.

Table 5: Definition of Terms

Abbreviation	Description
CLB	Configurable Logic Block
DAR	Double Address Rate
DCM	Digital Clock Manager
DDR	Double Data Rate
DLL	Delay-Locked Loop
FDDR	Double Data Rate Flip-Flop
FIFO	First In, First Out
GND	Ground
HSTL	High-Speed Transceiver Logic
IOB	Input/Output Block
JEDEC	Joint Electron Device Engineering Council
JTAG	Joint Test Action Group
LUT	Look-Up Table
PCB	Printed Circuit Board
QDR SRAM	Quad Data Rate Synchronous Static Random Access Memory
SAR	Single Address Rate
SDR	Single Data Rate
SSO	Simultaneous Switching Outputs
TAP	Test Access Port

Appendix B: References

When building a QDR II SRAM interface and controller, the following documentation is recommended highly. This high-quality source of information provides useful technical resources for achieving the desired interface:

- Xilinx Inc., UG002: *Virtex-II Platform FPGA User Guide*
<http://www.xilinx.com/bvdocs/userguides/ug002.pdf>
- Xilinx Inc., DS031: *Virtex-II Platform FPGA Data Sheet*
<http://www.xilinx.com/bvdocs/publications/ds031.pdf>
- Xilinx Inc., LogiCORE Data Sheet, Asynchronous FIFO
http://www.xilinx.com/ipcenter/catalog/logicore/docs/async_fifo.pdf
- Xilinx Inc., XAPP133: *Using the Virtex SelectIO Feature*
<http://www.xilinx.com/bvdocs/appnotes/xapp133.pdf>
- Xilinx Inc., XAPP685: *High-Speed Clock Architecture for DDR Designs Using Local Inversion*
<http://www.xilinx.com/bvdocs/appnotes/xapp685.pdf>
- Xilinx Inc., XAPP770C: *Using Local Clocking in the Physical Layer to Interface with QDR II SRAM*
<http://www.xilinx.com/memory>

- QDR consortium website
<http://www.qdrsram.com/>
- QDR II SRAM memory device vendors
 - ◆ Cypress Semiconductor at <http://www.cypress.com/>
 - ◆ Renesas Technology at <http://www.renesas.com/>
 - ◆ IDT, Inc. at <http://www.idt.com/>
 - ◆ Samsung Semiconductor, at <http://www.samsung.com/>
 - ◆ NEC Corporation at http://www.ic.nec.co.jp/memory/index_e.html/

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
05/24/04	1.0	Initial Xilinx release.