QDR SRAM Controller Reference Design for Stratix & Stratix GX Devices

May 2004, ver. 1.0 Application Note 349

Introduction

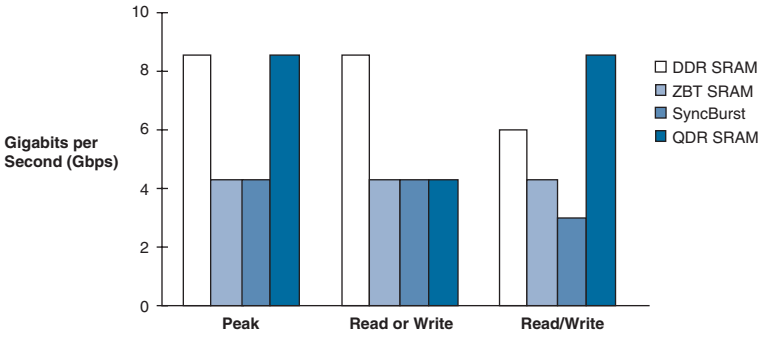
The explosive growth of the Internet has increased the demand for high-speed data communications systems that require fast processors and high-speed interfaces to peripheral components. While the processors in these systems have improved in performance, SRAM performance has not kept pace. New SRAM architectures are evolving to support the high-throughput requirements of current systems. One such architecture is quad data rate (QDR) SRAM, which provides more than four times the bandwidth of other SRAM architectures.

Most existing SRAM solutions are designed for PCs and have interfaces that move data efficiently for long bursts of reads or writes. In contrast, most communications applications require data transfer between the SRAM and the memory controller that alternates between read and write cycles. Devices with bidirectional interfaces, such as standard synchronous pipelined SRAM devices, do not perform well in these applications.

The QDR Consortium, comprised of Cypress Semiconductor, Hitachi, Integrated Device Technology, Inc., Micron Technology, NEC, and Samsung, designed the QDR SRAM architecture for high-performance communications systems such as routers and asynchronous transfer mode (ATM) switches. QDR SRAM devices, which can transfer four words per clock cycle, fulfill the requirements facing next-generation communications system designers. QDR SRAM devices provide concurrent reads and writes, zero latency, and increased data throughput, allowing simultaneous access to the same address location. Their innovative architecture allows them to outperform other SRAM devices by up to four times in networking applications, where reads and writes are balanced, as shown in Figure 1.

This application note provides an overview of QDR SRAM and outlines the advantages of using Stratix® and Stratix GX devices to interface to QDR SRAM devices. Additionally, this document describes the functionality of the QDR SRAM controller reference design for Stratix devices, and explains how to synthesize, place-and-route, and simulate the reference design. The reference design allows communication system designers to implement a QDR SRAM interface in Stratix and Stratix GX devices at clock speeds of up to 167 MHz.

Figure 1. SRAM Memory Architecture Performance Comparison *Note (1)*



Note to Figure 1:

(1) Figure 1 compares the performance of QDR SRAM devices versus other SRAM architectures such as double data rate (DDR), zero-bus turnaround (ZBT), and SyncBurst SRAM. This comparison assumes a 125-MHz clock speed for each memory architecture.

For more information on QDR SRAM devices, go to www.qdrsram.com.

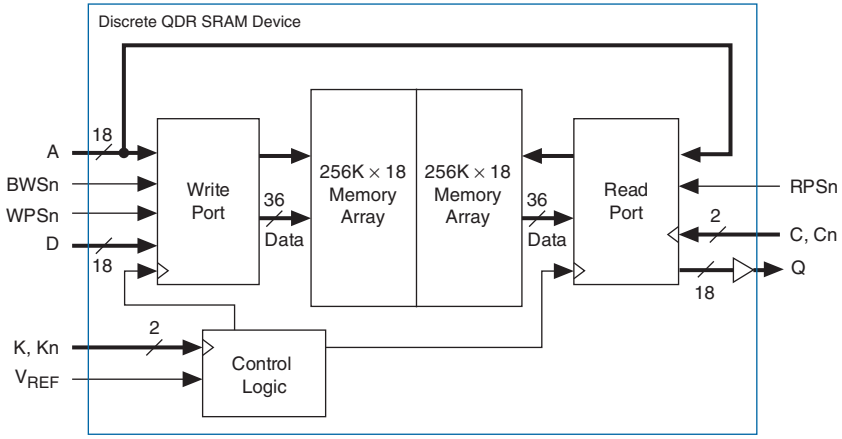
QDR SRAM Functional Description

The following features distinguish QDR SRAM devices from other SRAM devices:

- Separate write data (D) and read data (Q) ports that support simultaneous reads and writes and allow back-to-back transactions without the contention issues that can occur when using a single bidirectional data bus
- A shared address bus that alternately carries the read and write addresses
- A memory core made up of multiple SRAM arrays, permitting double data rate (DDR) access and a transfer rate of up to four words on every clock cycle

Figure 2 shows a block diagram of the QDR SRAM burst-of-2 architecture.

Figure 2. QDR SRAM Burst-of-2 Architecture



QDR SRAM Interface Signals

This section provides a description of the clock, control, address, and data signals on a QDR SRAM device, which is necessary to understand the functionality of the device.

Clock Signals

QDR SRAM devices have two pairs of clocks: K and Kn, and C and Cn. The positive input clock, K, is the logical complement of the negative input clock, Kn. Similarly, the positive output clock, C, and the negative output clock, Cn, are complements. The QDR SRAM device uses the K and Kn clocks for write accesses and the C and Cn clocks for read accesses. QDR SRAM devices also have a single-clock mode, where the K and Kn clocks are used for both reads and writes. In this mode, the C and Cn clocks are tied to the supply voltage V_{DD}.

Control Signals

QDR SRAM devices use two control signals, write port select (WPSn) and read port select (RPSn), to control write and read operations, respectively. A third control signal, byte write select (BWSn), writes only one byte of data at a time, if necessary.

Address Signals

QDR SRAM devices use one address bus (A) for both read and write addresses.

Data Signals

QDR SRAM devices use two unidirectional data buses, one for writes (D) and one for reads (Q).

QDR SRAM Functionality

QDR SRAM devices have a two-word or four-word burst capability. The burst metric signifies the number of data words that are read or written on a single access; however, both types of QDR SRAM devices provide the same overall bandwidth at a given clock speed.

Burst-of-2 QDR SRAM

Burst-of-2 QDR SRAM devices support two-word data transfers on all write and read transactions, requiring a relatively simple controller implementation. The sections below outline the basic burst-of-2 functionality for write-only, read-only, and combined read/write operations.

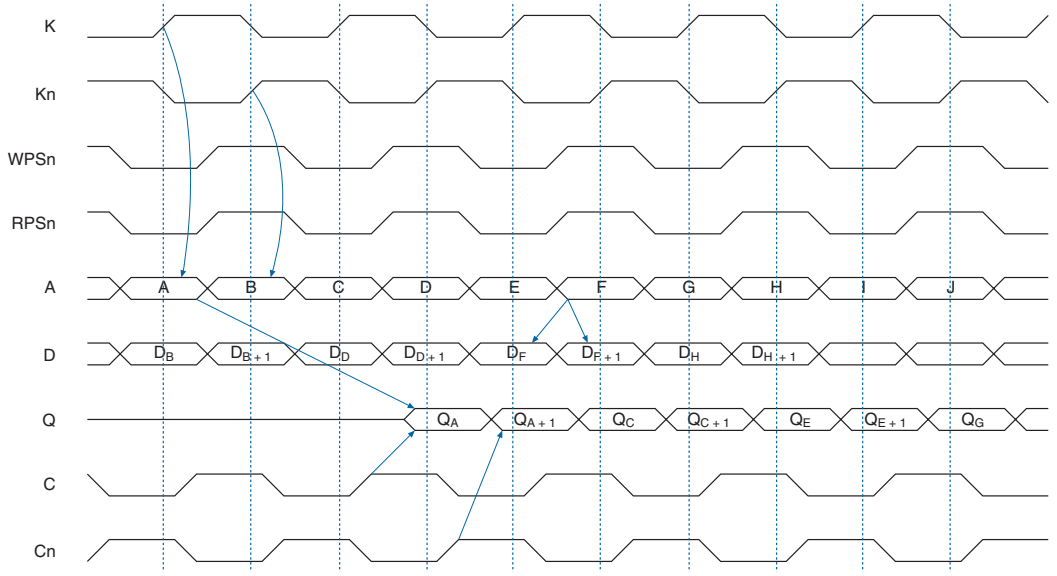
Write Cycle
On the rising edge of the K clock, the QDR SRAM device latches the control signals WPSn and BWSn and the lower data word on D. On the rising edge of the Kn clock, the QDR SRAM device latches the write address on A and the upper data word on D, thus completing a write cycle.

Read Cycle
On the rising edge of the K clock, the QDR SRAM device latches the control signal RPSn and the read address A. After a one-clock-cycle latency, the rising edge of C clocks out the lower data word at address A onto the Q bus. The next rising edge of Cn clocks out the upper data word, thus completing a read cycle. Single-clock mode uses K and Kn clocks for both reads and writes. See "QDR SRAM Device Clock Modes" on page 17 for more information.

Read/Write Cycle
Independent read and write data paths, along with the cycle-shared address bus, allow read and write operations to occur in the same clock cycle. Performing concurrent reads and writes does not change the functionality of either transaction. If a read request occurs simultaneously with a write request at the same address, the data on D is forwarded to Q; therefore, latency is not required to access valid data.

Figure 3 shows the burst-of-2 timing diagram for reads and writes.

Figure 3. Burst-of-2 Timing Diagram



Burst-of-4 QDR SRAM Devices

Burst-of-4 QDR SRAM devices support four-word data transfers on all writes and reads, which reduces address bus activity; however, the control circuitry needed to interface to burst-of-4 QDR SRAM devices is more complicated than control circuitry for burst-of-2 QDR SRAM devices. The following sections outline the basic burst-of-4 functionality for writes, reads, and read/write operations.

Write Cycle

On the rising edge of the K clock, the QDR SRAM device latches the control signals WPSn and BWSn and the write address A. On the following K clock rise, the QDR SRAM device latches the first data word on D. On the next Kn clock rise, the second data word is latched. The third and fourth words are latched in on the subsequent K and Kn clock rises, respectively, thus completing a write cycle.

Read Cycle

On the rising edge of the K clock, the QDR SRAM device latches the control signal RPSn and the read address A. After a one-clock-cycle latency, the rising edge of C clocks out the first data word at address A onto the Q bus. The next rising edge of Cn clocks out the second data word. The subsequent C and Cn clock rises clock out the third and fourth

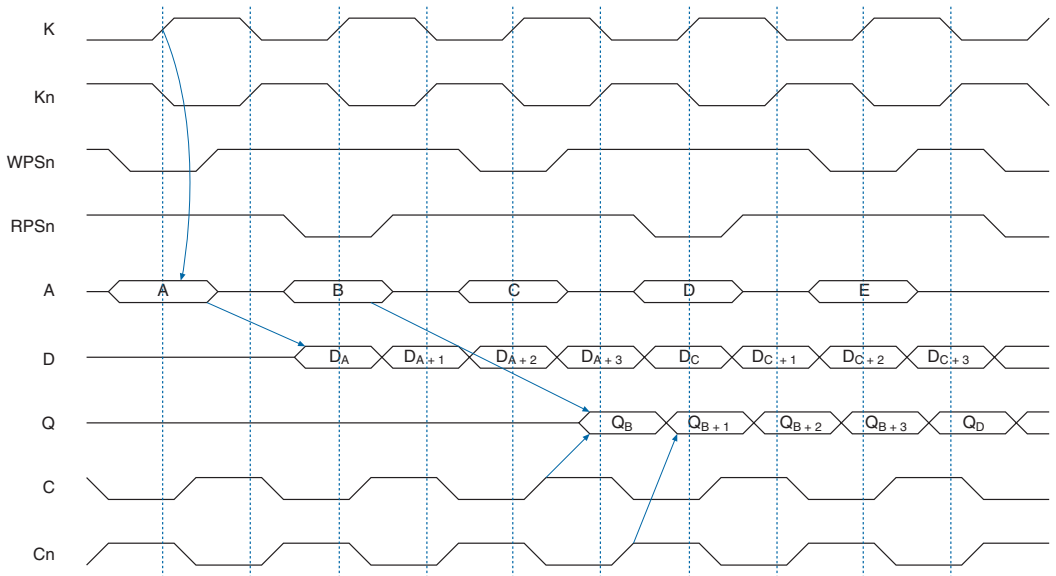
words, respectively, thus completing a read cycle. Single-clock mode uses the K and Kn clocks for both reads and writes. See "QDR SRAM Device Clock Modes" on page 17 for more information.

Read/Write Cycle

The independent read and write data paths and the cycle-shared address bus allow read and write operations to occur on subsequent clock cycles. Performing concurrent read and writes does not change the functionality of either transaction. If a read request occurs simultaneously with a write request at the same address, the data on D is forwarded to Q; therefore, latency is not required to access valid data.

Figure 4 shows the burst-of-4 timing diagram for reads and writes.

Figure 4. Burst-of-4 Timing Diagram



QDRII SRAM

The QDR Consortium has announced the QDRII architecture specification, designed for operation at clock speeds of up to 333 MHz. QDRII SRAM devices incorporate second-generation improvements in power consumption, packaging, and timing characteristics. These additional features increase the ease of system design and enable QDRII to become the memory architecture of choice for 10-gigabit and 40-gigabit networking systems.

For more information on QDRII, visit **www.qdrsram.com**.

QDRII SRAM device timing requirements are not as strict as QDR SRAM device timing requirements for a given clock speed. Therefore, because Stratix and Stratix GX devices support 167-MHz QDR SRAM devices, they can easily meet the timing requirements for interfacing with 167-MHz QDRII SRAM devices. Stratix and Stratix GX devices can also interface with QDRII SRAM devices at higher clock speeds.

Controller Implementation: Stratix Advantages

When using QDR SRAM devices in a system, a memory controller generates all of the signals needed for the QDR SRAM device and serves as the interface between the QDR SRAM device and the rest of the system. Altera® Stratix and Stratix GX devices have been designed to interface at high speeds with memories such as QDR SRAM. These advantages allow Stratix and Stratix GX devices to interface with QDR SRAM devices at clock speeds of up to 166.67 MHz (at a total bandwidth of 12 gigabits per second (Gbps)), making them an ideal solution for memory-intensive applications. Stratix and Stratix GX devices offer the following features to interface with QDR SRAM devices:

- DDR I/O registers provide built-in functionality that simplifies the task of interfacing to QDR SRAM devices.
- Stratix and Stratix GX I/O buffers, which are compliant with the HSTL I/O standard, enable fast data transfers to QDR SRAM devices.
- High-density Stratix and Stratix GX devices provide up to 79,040 logic elements (LEs) that can be utilized for custom logic connecting to the memory controller.

HSTL I/O Standard

QDR SRAM device pins use the HSTL I/O standard, which is fully supported in Stratix and Stratix GX devices at 333 megabits per second (Mbps) switching rates and beyond.

For more details, see the *Selectable I/O Standards in Stratix & Stratix GX Devices* chapter in the *Stratix Device Handbook*.

The HSTL I/O standard is used for applications designed to operate in the 0.0- to 1.5-V HSTL logic switching range. This standard defines single ended input and output specifications for all HSTL compliant digital integrated circuits. The single-ended input standard specifies an input voltage range of $-0.3\ V \leq V_I \leq V_{CCIO} + 0.3\ V$. The 1.5-V HSTL I/O standard in Stratix and Stratix GX devices is compatible with the 1.8-V HSTL I/O standard in APEX™ 20KE and APEX 20KC devices because the

input and output voltage thresholds are compatible. Using the 1.5-V V_{CCIO}, HSTL requires a 0.75-V input reference voltage (V_{REF}) and a 0.75-V termination voltage (V_{TT}). Stratix and Stratix GX devices support both input and output levels with V_{REF} and V_{TT}.

DDR I/O Registers

The main advantage of QDR SRAM devices is their ability to be written to and read from simultaneously on both the rising and falling clock edges. This ability quadruples the throughput of the QDR SRAM device. To take advantage of this high throughput, the Altera QDR SRAM controller uses the advanced I/O elements (IOEs) in Stratix and Stratix GX devices. These structures allow the Stratix or Stratix GX device to receive and transmit data on both the positive and negative clock edges and to meet the strict timing requirements of QDR SRAM devices.

Refer to *Stratix Device Family Data Sheet* and the *External Memory Interfaces* chapter in the *Stratix Device Handbook*; and the *Stratix GX Programmable Logic Device Family Data Sheet* for more information on the DDR I/O feature.

Reference Design Description

The Altera QDR SRAM controller reference design implements a QDR SRAM controller targeting an EP1S25F780C6 device. You can use this design as:

- A memory interface module that you can incorporate into a larger system-on-a-programmable-chip (SOPC) design
- An example that you can reference when targeting a different device in the Stratix or Stratix GX families

The reference design implements several optional pipeline registers in the core of the Stratix device. These registers allow the designer to place the controller's user interface signals on device pins for more straightforward simulation, while still maintaining 166.67-MHz operation for the standalone controller. In a typical system, however, custom logic is implemented in the logic array of the Stratix or Stratix GX device, and hence, the user interface signals feed the controller from within the device. In this case, Altera recommends that you remove the optional pipeline stages to reduce latency through the controller (see "Compile in the Quartus II Software" on page 24).

The reference design provides an interface to the Cypress CY7C1302V25-167 device, a 9-Mbyte, pipelined, burst-of-2 QDR SRAM device. You can implement controllers for larger QDR SRAM devices, burst-of-4 QDR SRAM devices, QDR SRAM devices from other vendors, QDRII SRAM devices, or a memory bank consisting of multiple QDR

SRAM devices, with little or no changes to the reference design. You should fully verify any modifications to the reference design using your design tool flow.

Controller Structure & Operation

Figure 5 shows a block diagram of the controller reference design.

Figure 5. QDR SRAM Controller Reference Design Block Diagram

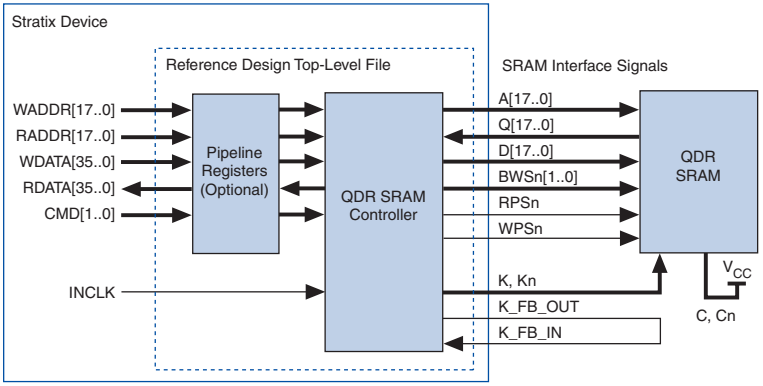


Table 1 describes the function of each controller pin on the QDR SRAM device interface.

Table 1. QDR SRAM Device Interface Signals (Part 1 of 2)

Type	Direction	Name	Description
Clock	Output	K, Kn	K and Kn are output by the Stratix or Stratix GX device and are clock inputs to the QDR SRAM device. All transactions are initiated synchronously on the rising edge of K or Kn. These clocks are generated from the rising and falling edges of WRITE_CLK_90.
		K_FB_OUT	K_FB_OUT is fed back to the controller as K_FB_IN to imitate the data flight times to and from the QDR SRAM device. (See Figure 9 on page 16 for more details on the clocking scheme).
	Input	K_FB_IN	The controller uses the K_FB_IN clock to generate READ_CLK for clocking in data from the QDR SRAM device.

Table 1. QDR SRAM Device Interface Signals (Part 2 of 2)

Type	Direction	Name	Description
Control	Output	RPSn	This active-low read port select signal is clocked out on the rising edge of WRITE_CLK and sampled by the QDR SRAM device on the rising edge of K.
		WPSn	This active-low write port select signal is clocked out on the rising edge of WRITE_CLK and sampled by the QDR SRAM device on the rising edge of K.
		BWSn[1..0]	This active-low byte write select signal is clocked out on the rising edge of WRITE_CLK and sampled by the QDR SRAM device on the rising edge of K. You can use this signal to individually select which bytes are written or read. For the reference design, both bytes are active on writes. You can add logic for individual byte writes if desired.
Address	Output	A[17..0]	The QDR SRAM device uses the same address signals for the read and write ports. The address inputs to the QDR SRAM device are clocked out of the controller using WRITE_CLK and sampled on the rising edge of K for reads and on the rising edge of Kn for writes.
Data	Input	Q[17..0]	Read data output from QDR SRAM. The QDR SRAM device can transfer two words during each clock cycle because the device outputs data on the rising edges of both K and Kn. On the subsequent falling edge of READ_CLK, the controller captures the QDR SRAM device's output data on the rising edge of K. The controller captures data output on the rising edge of Kn on the subsequent rising edge of READ_CLK.
	Output	D[17..0]	Write data input to the QDR SRAM device. The controller clocks data out on the rising and falling edges of WRITE_CLK and the QDR SRAM captures it on the next rising edges of K and Kn. Therefore, two words can be transferred to the QDR SRAM device during each clock cycle.
Reserved	Output	reserved1_VCC	Reserved pin driving high.
		reserved2_GND	Reserved pin driving low.
		reserved3_GND	Reserved pin driving low.
		reserved4_VCC	Reserved pin driving high.
		reserved5_VCC	Reserved pin driving high.
		reserved6_GND	Reserved pin driving low.

Table 2 shows the user interface signals for the controller. In a system implementation of the controller, these controller ports would typically be connected to custom logic within the Stratix or Stratix GX device.

Table 2. User Interface Signals

Type	Direction	Name	Description
Clock	Input	INCLK	User input clock, which is used to generate WRITE_CLK and WRITE_CLK_90. This clock is 166.67 MHz in the reference design.
Control	Input	CMD[1..0]	User command input, which is sampled on the rising edge of WRITE_CLK.
Address	Input	RADDR[17..0] WADDR[17..0]	User read and write address inputs, which are sampled on the rising edge of WRITE_CLK.
Data	Input	WDATA[35..0]	Data input for write operations, which is sampled on the rising edge of WRITE_CLK.
	Output	RDATA[35..0]	Data output from read operations; it is output from the controller on the rising edge of READ_CLK.

Table 3 shows the commands accepted by the controller.

Table 3. Controller Commands

Command	Code (CMD[1..0])
Idle	00
Read	01
Write	10
Read/Write	11

The Altera QDR SRAM controller is a synchronous interface. Because the read and write data paths for the controller are independent, the controller can perform reads and writes together or separately.

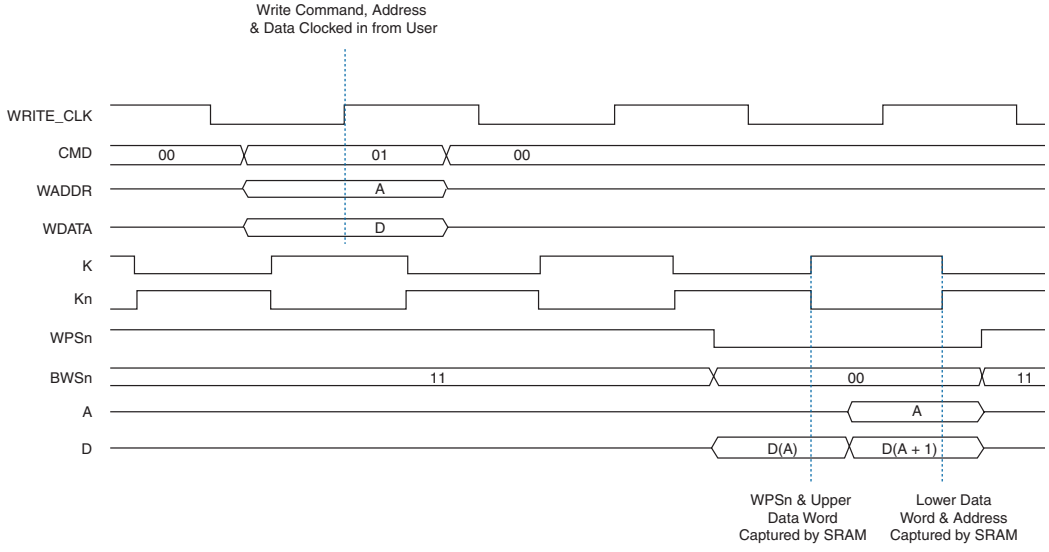
To use the controller, you must first provide an input clock (INCLK), which is fed into a fast phase-locked loop (PLL). Once the PLL has locked onto the input clock signal, it generates two clocks for the controller: WRITE_CLK and WRITE_CLK_90, a 90-degree-phase-shifted version of WRITE_CLK. A second fast PLL generates READ_CLK. For more information, see "Clock Generation" on page 15.

At the rising edge of WRITE_CLK, the controller should receive an idle, read, write, or read/write two-bit command, as shown in Table 3 on page 11. The controller should receive the appropriate read address (RADDR) simultaneously with a read or read/write command. Similarly, the controller should receive the write address (WADDR) and write data (WDATA) simultaneously with a write or read/write command.

At the QDR SRAM side of the controller, the DDR I/O registers output data, address, and control signals as well as the K and Kn clocks, which are generated using WRITE_CLK_90. WRITE_CLK_90 is also used to output another clock, K_FB_OUT, which is fed back to the controller as K_FB_IN. K_FB_IN is sent to a PLL, which generates READ_CLK.

For write operations, the upper data word (D) is output and the write port select (WPSn) and byte write select (BWSn) signals are asserted on the rising edge of WRITE_CLK. The QDR SRAM device captures these signals on the rising edge of K. On the falling edge of WRITE_CLK, the address (A) and the lower data word are sent to the QDR SRAM device. These signals are captured on the rising edge of Kn. Figure 6 shows the functionality of the controller for write operations.

Figure 6. Write Cycle Waveform (Burst-of-2)

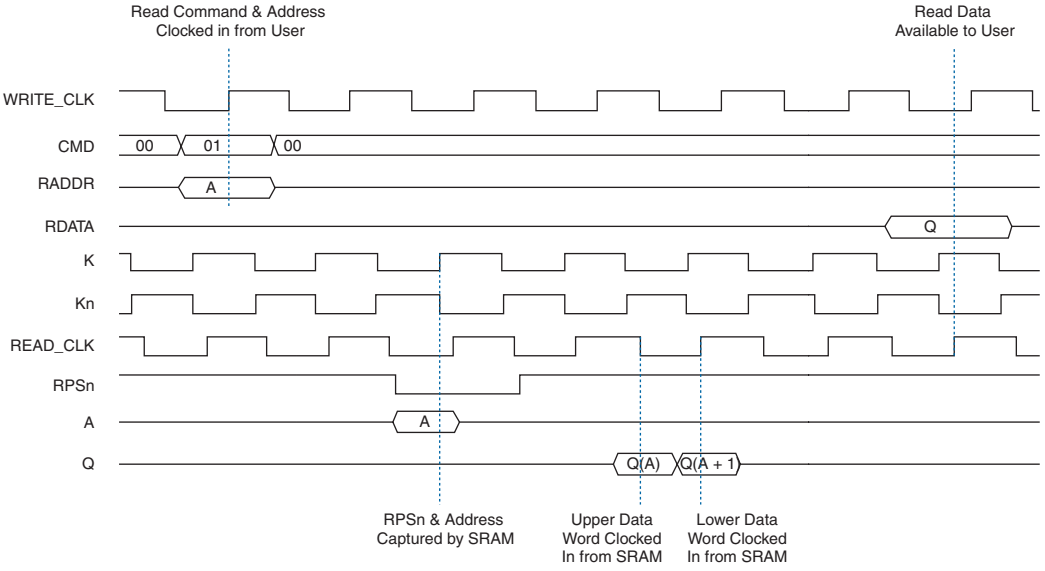


For read operations, the address (A) and read port select (RPSn) signals are output on the rising edge of WRITE_CLK. After the QDR SRAM device captures this information on the rising edge of K, the upper and lower

data words at that address are driven to Q on the next rising edges of K and Kn, respectively. The controller captures the words on the READ_CLK clock's falling and rising edges, respectively. The data is then sent back to the read data (RDATA) ports where it is output on the rising edge of READ_CLK. Figure 7 shows the functionality of the controller for read operations.

This reference design assumes that the QDR SRAM device is operating in single-clock mode. Therefore, the K and Kn clocks are used for reads as well as writes. See "Clock Generation" on page 15 for more details.

Figure 7. Read Cycle Waveform (Burst-of-2)



Constraints

The reference design requires constraints to meet the strict timing requirements of QDR SRAM devices. The reference design shows examples of the appropriate pin placement and I/O standard assignments. You can view the assignments required for the reference design by viewing the current assignments floorplan in the Altera Quartus® II software.

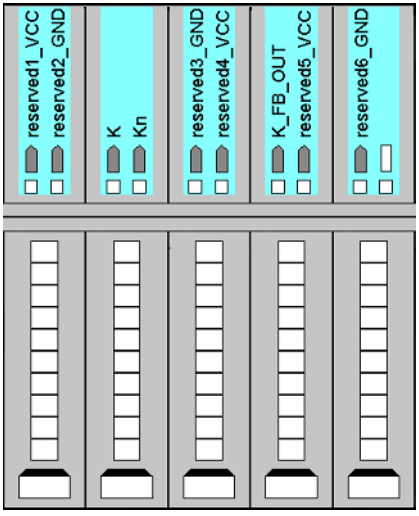
This section provides the required assignments in interface successfully between a QDR SRAM device and a EP1S25F780C6 Stratix device. Other Stratix or Stratix GX devices may require additional location assignments to ensure high-speed operation of the controller.

Pin Placement

To ensure proper pin placement and I/O buffer configuration for the controller, you must use a Quartus II Constraint File (**.csf**) with the following assignments:

- Place the following outputs from the controller on Stratix or Stratix GX device column pins: `A[17..0]`, `D[17..0]`, `RPSn`, `WPSn`, `BWSn[1..0]`, `K`, `Kn`, and `K_FB_OUT`. This placement ensures faster clock-to-out times (t_{CO}) than could be achieved if the controller outputs were placed on row pins.
- `K`, `Kn`, and `K_FB_OUT` must be assigned to consecutive column pins. Separate these pins by alternating reserved pins tied to V_{CC} and ground to preserve signal integrity, as shown in Figure 8.
- You can place the `Q[17..0]` inputs to the controller on either column or row pins.
- Add the assignment **Decrease Input Delay to Input Register = On** to the `Q[17..0]` inputs to minimize the setup time (t_{SU}) on those pins.

Figure 8. Reserved Pin Placement around K, Kn & K_FB_OUT Clock Pins



HSTL I/O Standard Assignments

The QDR SRAM device interface requires the use of the HSTL I/O standard. Stratix and Stratix GX devices are designed to drive out and receive HSTL I/O signals at switching rates greater than 333 Mbps (166.67 MHz, double data rate).

Refer to *Selectable I/O Standards in Stratix & Stratix GX Devices* chapter in the *Stratix Device Handbook* for more details on HSTL.

To implement the HSTL I/O interface, perform the following steps:

1. Use the **Assignment Organizer** in the Quartus II software to assign the HSTL Class I I/O standard to all pins that interface with the QDR SRAM device (`A[17..0]`, `D[17..0]`, `Q[17..0]`, `RPSn`, `WPSn`, `BWSn[1..0]`, `K`, `Kn`, `K_FB_IN`, and `K_FB_OUT`).

2. The Quartus II software automatically places reference voltage (`VREF`) pins appropriately, as required by the HSTL Class I I/O standard.

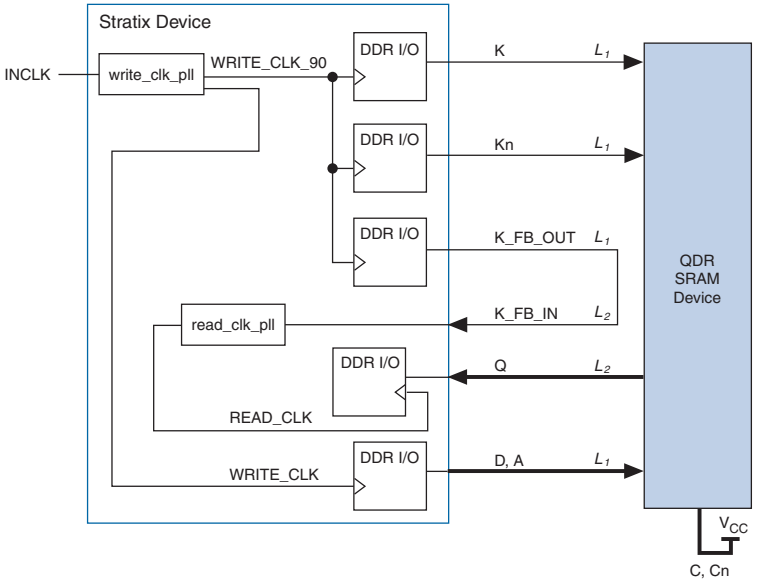
Clock Generation

The controller clocking scheme maintains consistent and robust high-frequency operation of the QDR SRAM device. The Altera QDR SRAM controller reference design requires two fast PLLs and two global clock resources in the Stratix or Stratix GX device to perform clock generation. The design uses the following clocks:

- `INCLK`—Input clock from the user
- `WRITE_CLK` and `WRITE_CLK_90`—True and 90-degree-phase-shifted controller clocks
- `K` and `Kn`—QDR SRAM clocks
- `K_FB_IN` and `K_FB_OUT`—Controller feedback clock
- `READ_CLK`—Read data capture clock

Figure 9 shows how these clocks are generated and used for the QDR SRAM device interface.

Figure 9. Clock Generation *Note (1)*



Note to Figure 9:

(1) All L1 traces should be of equal length. All L2 traces should be of equal length. L1 traces do not need to be the same length as L2 traces.

Internal Clocks

You must supply an input clock (INCLK), nominally 166.67 MHz, to the design. This clock feeds an on-chip fast PLL that generates the true (non-phase-shifted) clock for data and address (WRITE_CLK) and the 90-degree-shifted clock (WRITE_CLK_90) for the K and Kn outputs.

If necessary, you can supply a lower-frequency input clock and multiply the clock to 166.67 MHz using use the ClockBoost® feature in the PLL.

The controller uses WRITE_CLK_90 and the DDR I/O registers with the inputs tied to V_{CC} and ground to generate a differential clock signal for the QDR SRAM device. The result is a clock signal (K) and a 180° phase-shifted clock signal (Kn), each with the same frequency as WRITE_CLK.

READ_CLK, whose purpose is to clock in the read data from the QDR SRAM device, is generated from a feedback clock using a second PLL, as described in the following sections.

QDR SRAM Device Clocks

The Stratix or Stratix GX device outputs the K and Kn clocks and the data, address, and control lines to the QDR SRAM device. This action negates the effect of signal skew on the write and read request operations because the propagation delays for K and Kn from the Stratix or Stratix GX device to the QDR SRAM device are equal to the delays on the data signals. For the controller to operate properly, the trace lengths (and therefore the propagation times) of the data in (D), address (A), and control signals should be made equal to the trace lengths of the K and Kn clocks.

Because data is transported both to and from the QDR SRAM device, you should use a similar strategy to eliminate signal skew on read operations. One method is to feed the K clock back into the controller and then input this feedback clock to a PLL to generate the READ_CLK, which is used capture the read data. In this case, the length of the feedback trace between the QDR SRAM device and the controller should be equal to the read data (Q) trace length.

The disadvantage of this method is that the extra tap on the K trace (without an additional tap on Kn) can cause skew between the K and Kn clock signals. Because of this issue, the reference design outputs an additional clock, K_FB_OUT, to emulate the effect of feedback from K. For this method to work properly, the feedback trace length must match the sum of the D and Q trace lengths. K_FB_OUT is returned to the controller as K_FB_IN, and used to drive the READ_CLK clock.

QDR SRAM Device Clock Modes

QDR SRAM devices can use two pairs of clocks: K and Kn for writes, and C and Cn for reads, all provided by the controller. This arrangement is especially useful when a bank of multiple QDR SRAM devices is driven by a single controller. In this case, the K and Kn traces can be tapped off at various points to drive C and Cn, respectively, to compensate for differences in flight times between each QDR SRAM device and the controller.

However, the number of loads that must be driven by K and Kn can have an effect on the switching times of these outputs. Furthermore, when a controller drives a single QDR SRAM device, C and Cn are unnecessary because propagation delays from the controller to the QDR SRAM device and back are already equal. For these reasons, this reference design assumes that the QDR SRAM device is operating in single-clock mode. In single-clock mode, the C and Cn inputs are connected to V_{CC}, and the K and Kn inputs are used for both reads and writes.

Component Placement

Finally, Altera recommends that you place the Stratix or Stratix GX device adjacent to the QDR SRAM device on the circuit board. This positioning keeps trace lengths to a minimum and further minimizes any skew caused by board delay.

Timing

Because data is transferred between the controller and the QDR SRAM device at high speeds, you should take special care to avoid setup or hold violations for the QDR SRAM, Stratix, or Stratix GX device. This section discusses the timing issues that may arise when designing a high-speed QDR SRAM device interface.

Write Cycle

When designing for correct write-cycle timing, meeting the setup and hold requirements of the QDR SRAM device is the primary concern. Setup and hold specifications for the CY7C1302V25-167 device are 0.7 ns each.

The controller drives both the QDR clock and data signals; therefore, the clock-to-output delay from the Stratix or Stratix GX device is the same for both sets of pins. Preliminary timing characteristics show that the clock-to-output delay from the Stratix or Stratix GX column pins under worst-case temperature and voltage conditions can range from 3.2 to 3.5 ns, depending on the pin placement. The board delays for the clock and data are assumed to be roughly equal, because the signal trace lengths should be matched (see "Clock Generation" on page 15).

At a clock speed of 166.67 Mhz, the bit period—the length of time between each data bit—is 3 ns. Because `K` and `Kn` are generated from `WRITE_CLK_90`, while data and address are generated from `WRITE_CLK`, there is a timing cushion of one-half of the bit period each way to meet setup and hold times at the QDR SRAM device.

The following calculations apply for 166.67-MHz controller-to-QDR-SRAM data transfers. The calculations allow for up to 0.1 ns of board-induced skew.

$$[t_{CO}\ (\text{Stratix Clock}) - t_{CO}\ (\text{Stratix Data and Address})] + \text{Board Skew (Clock} - \text{Data)} + t_{SU}\ (\text{QDR SRAM}) < (\text{Bit Period})/2$$

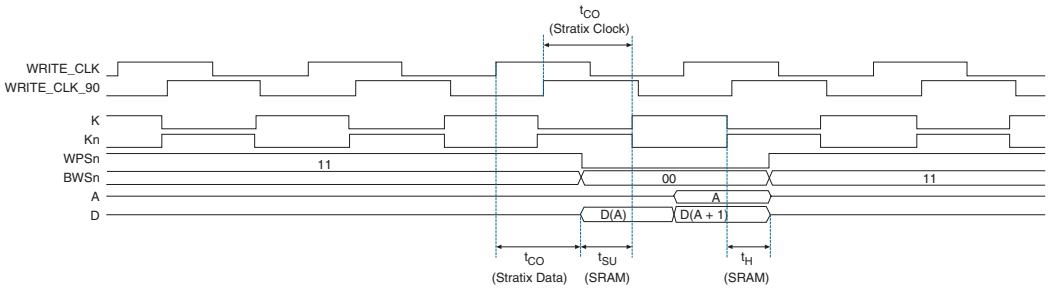
$$[3.5\ \text{ns} - 3.2\ \text{ns}] + 0.1\ \text{ns} + 0.7\ \text{ns} = 1.1\ \text{ns} < 1.5\ \text{ns}$$

[t_{CO} (Stratix Data and Address) – t_{CO} (Stratix Clock)] + Board Skew (Clock – Data) + t_H (QDR SRAM) < (Bit Period)/2

[3.5 ns – 3.2 ns] + 0.1 ns + 0.7 ns = 1.1 ns < 1.5 ns

Figure 10 shows the write cycle timing waveform for the QDR SRAM interface pins at 166.67 MHz.

Figure 10. Write Cycle Timing Waveform



In addition to setup and hold times, an additional concern is the clock-to-clock skew between K and Kn (t_{KHKH}). The 167-MHz QDR SRAM specification allows for up to 0.6-ns difference between the rising edges of K and Kn. Because Stratix and Stratix GX clock-to-out times can vary with pin position, place K and Kn on adjacent pins and check their t_{CO} times carefully in the Quartus II **Timing Analyzer**. Preliminary timing shows that the controller meets the t_{KHKH} specification when the K and Kn pins are adjacent.

Read Cycle

The read request and address signals are sent to the QDR SRAM device along with the K and Kn clocks in a similar manner to the write data. Therefore, the write timing parameters apply to these signals as well.

Additionally, when the QDR SRAM device sends read data to the controller, the design must meet the Stratix or Stratix GX device setup and hold times. Preliminary timing characteristics show that the worst-case setup time for the Q pins in the reference design is 0.0 ns and worst-case hold time is 0.3 ns.

The clock-to-output specification for the QDR SRAM device determines the arrival time of the Q signal. For the CY7C1302V25-167 device, the maximum t_{CO} value is 2.5 ns and the minimum t_{CO} value (i.e., the data output hold time t_{DOH}) is 1.2 ns.

You can ignore board delay because flight times for the K_FB signal and the Q bus are roughly equal. Regardless, the timing calculation allows for up to 0.1 ns of board-induced skew between the clock and data lines. Maximum allowed clock skew for K and Kn is determined by the QDR SRAM device t_{KHKH} specification as ±0.3 ns. This clock skew allowance is included in the read calculations as well.

The QDR SRAM device sends data out on the rising edge of K, and the controller captures it on the falling edge of READ_CLK. For a clock speed of 166.67 MHz, there is a window of 3 ns between the rising and falling edges. Subtracting the QDR SRAM device clock-to-output delay of 2.5 ns leaves 0.5 ns of margin for the Stratix setup times and board and clock skew. This margin is sufficient to meet the Stratix device's setup time.

The QDR SRAM device data output hold time is 1.2 ns. This margin is large enough to allow for the Stratix or Stratix GX device hold time.

For the reference design, the following calculations apply for data transfers from the QDR SRAM device to the controller:

t_{CO} (QDR SRAM) + Board Skew (K_FB – Data) + Clock Skew (K – Kn) + t_{SU} (Stratix) < Bit Period

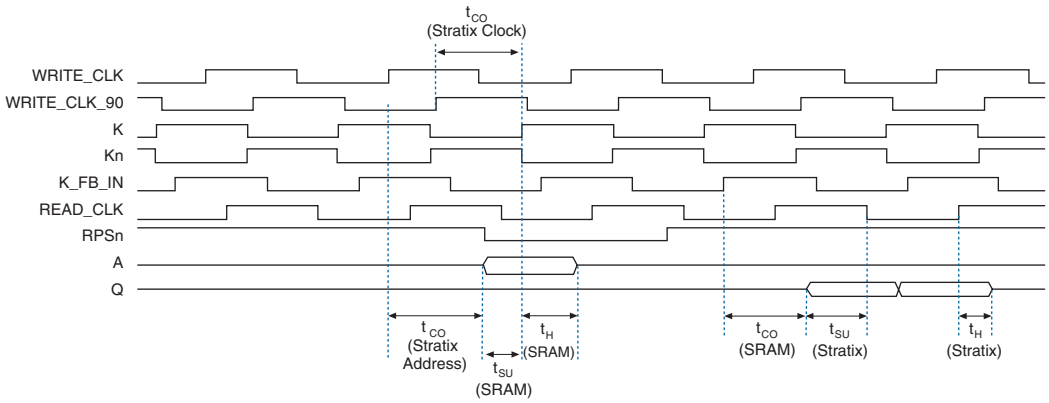
2.5 ns + 0.1 ns + 0.3 ns + 0.0 ns = 2.9 ns < 3 ns

t_{DOH} (QDR SRAM) – Board Skew (K_FB –Data) – Clock Skew (K – Kn) – t_{H} (Stratix) > 0 ns

1.2 ns – 0.1 ns – 0.3 ns – 0.3 ns = 0.5 ns > 0 ns

Figure 11 shows the read cycle timing waveform for the QDR SRAM device interface pins at 166.67 MHz.

Figure 11. Read Cycle Timing Waveform



Read/Write Cycle

The QDR SRAM controller has independent read and write paths. Therefore, timing does not change for a standalone read or write versus a combined read/write operation.

Getting Started

This section describes how to install the QDR SRAM controller reference design and walks you through the design flow.

This description is for the Verilog HDL version of the reference design; the description for the VHDL version is similar. However, the filenames have different extensions.

Hardware & Software Requirements

To use the QDR SRAM controller reference design, you must have the following software installed on your system:

- The Quartus II software version 2.1 or later
- The ModelSim-Altera software version 5.6a or later

This walkthrough uses the Quartus II software version 2.1 and the ModelSim-Altera software version 5.6a on a PC running Windows NT version 4.0.

Design Installation

Altera provides the QDR SRAM controller reference design as two executable files, one for VHDL and one for Verilog HDL. To install the files, perform the following steps:

You can download both the Verilog HDL and VHDL version of the reference design from the Altera web site at www.altera.com.

1. Save the executable file, **qdr_rd_stratix_vlog-<*version number*>.exe** (Verilog HDL) or **qdr_rd_stratix_vhdl-<*version number*>.exe** (VHDL), onto your hard disk. You can delete this file after you finish installing.

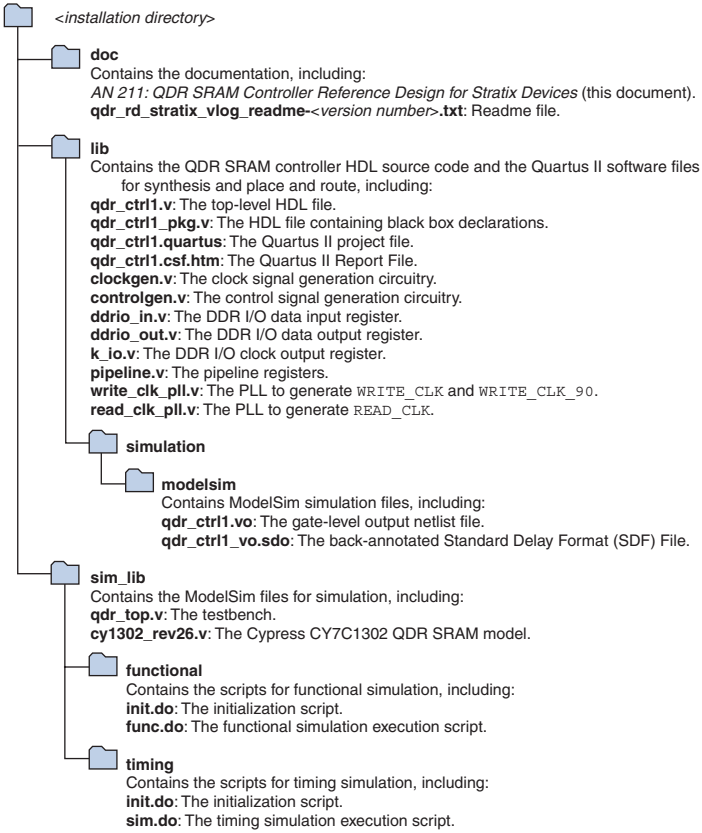
2. Double-click the **qdr_rd_stratix_vlog-<*version number*>.exe** (Verilog HDL) or **qdr_rd_stratix_vhdl-<*version number*>.exe** (VHDL) file in the Windows Explorer to launch the installer.

3. Follow the on-line instructions to complete the installation. The default installation directory is

   c:\Altera\qdr_rd_stratix_vlog-<*version number*>.

Figure 12 shows the directory structure created by the reference design installer. It also describes selected files (Verilog HDL design files only; the VHDL files have similar functionality).

Figure 12. QDR SRAM Controller Directory Structure



Design Walkthrough

Altera provides the source files of the reference design, which you can use to synthesize, place-and-route, and simulate the design. This section walks you through the design flow for the reference design. The steps include:

- Compile in the Quartus II software
- Simulate in the ModelSim-Altera software

The reference design includes the results for each step; therefore, you do not need to perform each step unless you have altered the design files. For example, you can run a timing simulation without first compiling the design because the Quartus II software place-and-route results are shipped with the reference design.

Compile in the Quartus II Software

The <*installation directory*>**\lib** directory contains the Quartus II software version 2.1 project files, including source files for synthesis and place-and-route within the Quartus II software and necessary constraint files for the design to meet the required clock frequencies and I/O timing in the EP1S25F780C6 device.

The following source files are included in the <*installation directory*>**\lib** directory:

- **qdr_ctrl1.v—**Top-level of the QDR SRAM controller.
- **qdr_ctrl1_pkg.v—**Declarations for the black box modules.
- **write_clk_pll.v** and **read_clk_pll.v—**PLL instantiation files created by the Quartus II MegaWizard® Plug-In Manager. These files instantiate the parameterized `altclocklock` function, which generates a PLL in the Stratix device. In the reference design, **write_clk_pll.v** generates `WRITE_CLK` and `WRITE_CLK_90`, and **read_clk_pll.v** generates `READ_CLK`.
- **k_io.v**, **ddrio_out.v**, and **ddrio_in.v—**DDR I/O instantiation files created by the Quartus II MegaWizard Plug-In Manager. **k_io.v** generates the clocks, **ddrio_out.v** generates the output data, and **ddrio_in.v** generates the input data.
- **clockgen.v** and **controlgen.v—**Contain the clock signals (`K`, `Kn`, `K_FB_IN`) and control signals (`BWSn`, `RPSn`, and `WPSn`) generation circuitry, respectively.
- **pipeline.v—**Adds three pipeline stages to both the read and write paths.

The QDR SRAM controller instantiates these pipeline registers so that the design meets the $\mathbf{f_{MAX}}$ performance requirements. The source code (**qdr_ctrl1.v** for Verilog HDL or **qdr_ctrl1_pkg.vhd** for VHDL) includes the parameter `INCLUDE_PIPELINE_REGS`, which controls whether the extra pipeline stages are added to the write and read paths. You can set this parameter to `FALSE` to eliminate the pipeline stages.

To compile the Altera-provided project files, follow the steps below:

1. Run the Quartus II software.
2. Choose **Open Project** (File menu).
3. Browse to the <*installation directory*>**\lib** directory.
4. Select the project file **qdr_ctrl1.quartus** and click **Open**.
5. Choose **Compile Mode** (Processing menu).

6. Choose **Start Compilation** (Processing menu).

Simulate in the ModelSim-Altera Software

The **sim_lib** directory contains an HDL testbench file (**qdr_top.v**) that instantiates the QDR SRAM controller and the QDR SRAM model (**cy1302_rev26.v**). The testbench implements four pipelined writes, four pipelined reads, a standalone write operation, a read/write operation, and a standalone read operation, to demonstrate the functionality of the controller. The testbench adds delay to the board traces to model the propagation delay between the Stratix or Stratix GX device and the QDR SRAM device. You can model different board delay scenarios by changing these values.

Altera provides the following scripts to perform functional and timing simulation in the ModelSim-Altera software.

- **init.do**—This script creates the work library and pre-compiles the correct simulation libraries for functional or timing simulation.
- **func.do**—This script, located in the **functional** subdirectory, compiles the controller source files, the model, and the testbench, and displays the appropriate waveforms.
- **sim.do**—This script, located in the **timing** subdirectory, compiles the gate-level HDL output netlist file generated by the Quartus II software (**\lib\simulation\modelsim\qdr_ ctrl1.vo**), the model, and the testbench, and performs a timing simulation with a back-annotated Standard Delay Format File (**.sdf**) (**\lib\simulation\modelsim\qdr_ ctrl1_v.sdo**).

Before using the **init.do** script, you should update it so that the paths in the script point to the location in which you installed the Quartus II software.

To perform functional simulation, perform the following steps:

1. Run the ModelSim-Altera software.

2. Change your working directory to the *<installation directory>***\sim_lib\functional** directory.

3. Type the following commands in the **Command Window**:

```
do init.do ↵
do func.do ↵
```

To perform timing simulation, perform the following steps:

1. Copy the files in the <*installation directory*>**\lib\simulation\modelsim** directory to the <*installation directory*>**\sim_lib\timing** directory.
2. Run the ModelSim-Altera software.
3. Change your working directory to the <*installation directory*>**\sim_lib\timing** directory.
4. Type the following commands in the **Command Window**:

```
do init.do ↵
do sim.do ↵
```

For Verilog HDL simulation, the ModelSim-Altera software may show setup violations at the start of simulation as the initial values settle through the controller. However, there should not be any setup violations during actual operation of the controller.

Instantiation within an SOPC Design

You can instantiate the QDR SRAM controller design files in an HDL file and integrate it with an SOPC design. This section describes the steps you should follow to integrate the controller into an SOPC design. Generally, you should use the information found in "Design Walkthrough" on page 23 as a reference when integrating the controller with your system.

Synthesis

If you are using a third-party synthesis tool for your SOPC design, Altera recommends that you black box the QDR SRAM controller through that synthesis tool, as synthesis of the QDR SRAM controller source files outside of the Quartus II software has not been tested.

Place & Route

Before compiling your SOPC design in the Quartus II software, you must add the **lib** directory to your Quartus II project as a user library. Search for "User Libraries" in Quartus II Help for more information.

To ensure that your design meets the QDR timing requirements, you should generate an appropriate constraint file for your SOPC design. See "Constraints" on page 13. Refer to the Quartus II project provided in the **lib** directory for example project assignments.

Simulation

The simulation testbench and scripts shipped with the reference design are intended to demonstrate the correct operation of the stand-alone QDR SRAM device interface. Altera recommends that you generate your own simulation environment that is better-suited to your SOPC design and EDA tools.

Resource Usage

For Stratix or Stratix GX devices, the QDR SRAM controller reference design requires the device resources shown in Table 4.

Table 4. Resource Usage

Logic Cells	PLLs	Global Clocks	I/O Pins
220	2	2	68 *(1)*

Note to Table 4:

(1) 68 I/O pins, plus additional VREF pins, are needed to interface to the QDR SRAM device. An additional 111 pins are necessary if you want to interface with the controller from outside the Stratix or Stratix GX device, as implemented in the reference design.

Support

For information or support for the QDR SRAM controller reference design, go to mysupport.altera.com or contact Altera Applications.

Conclusion

QDR SRAM devices were designed for high-bandwidth communications applications and outperform other memory devices by up to four times in networking applications. The advanced features of Altera's Stratix oand Stratix GX devices help communications system designers take advantage of QDR SRAM technology and achieve the greatest possible performance when interfacing with QDR SRAM devices. Designers can use the QDR SRAM controller reference design to quickly implement a QDR SRAM controller in a Stratix or Stratix GX device to provide the highest possible memory performance for their systems.

Reference

CY7C1302V25 9-Mb Pipelined SRAM with QDR Architecture Advance Information, Cypress Semiconductor Corporation

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
lit_req@altera.com