QDR SRAM Controller Function

December 2000, ver. 1.0 Application Note 133

QDR Overview

The explosive growth of the Internet has boosted the demand for high-speed data communication systems that require fast processors and high-speed interfaces to peripheral components. While the processors in these systems have improved in performance, static memories have 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 can provide bandwidth improvements more than four times greater than other SRAM architectures.

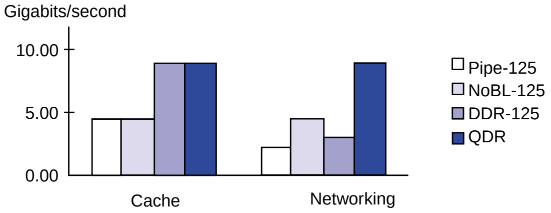
Most existing SRAM solutions were designed for PCs and have interfaces that transfer data efficiently for PC-type single input/output (I/O) applications. In contrast, most networking applications require continuous data transfer between the SRAM and the memory controller (e.g., continous transitions between read and write cycles through the memory). Single I/O devices like standard synchronous pipelined SRAMs do not perform well in these applications.

The QDR Consortium, comprised of Cypress Semiconductor, Integrated Device Technology, Inc., and Micron Technology, designed the QDR SRAM architecture for high-performance networking systems such as routers and ATM switches. QDR SRAMs are designed to handle the transfer of four data words through the SRAM in a single clock cycle. The memory provides simultaneous reads and writes, as well as zero latency and increased data throughput, guaranteeing simultaneous access to the same address location.

This application note describes implementation of a QDR SRAM controller in an Altera® APEX™ 20KE programmable logic device. The implementation allows designer to interface with the Cypress CY7C1302 QDR SRAM device at a 100-MHz clock speed, giving an overall system performance of 7.2 gigabits per second (Gbps).

Figure 1 compares the performance of QDR SRAMs versus other SRAM architectures. The comparison assumes that the interfaces operate at 166 MHz. The figure shows that the QDR SRAM outperforms all other types of SRAM and outperforms other architectures by four times in a networking application.

Figure 1. Performance Comparison



Source: QDR Consortium

APEX Interface

When using QDR SRAM in a system, a memory controller generates all the signals needed for the SRAM and serves as the interface to the system. Altera APEX 20KE devices—with their speed and configurability—are ideal for such a function. A QDR SRAM controller be implemented in an APEX 20KE device to provide a simplified interface to an industry-standard QDR SRAM device. Figure 2 provides a block diagram of the system.

Figure 2. Block Diagram

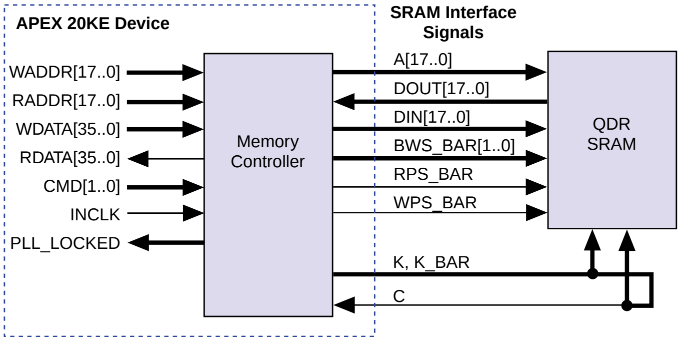


Table 1 describes the function of each pin on the SRAM-APEX interface.

Table 1. QDR SRAM Interface Signals

Signal Type	Signal Name	Description
Clock Outputs	K, K_bar	Clock inputs to the QDR SRAM. All transactions are initiated synchronously on the rising edge of K or K_bar.
Clock Input	C	K and K_bar are fed back as C and C_bar, respectively, at the QDR SRAM for use as output clocks. The controller uses the C clock as a feedback input for clocking in data from the QDR SRAM.
Control Outputs	RPS_bar	Active-low read port select signal, sampled on the rising edge of K.
	WPS_bar	Active-low write port select signal, sampled on the rising edge of K.
	BWS_bar[1..0]	Active-low byte write select signal, sampled on the rising edge of K. This signal is used to individually select which bytes are written or read.
Address Outputs	A[17..0]	The QDR SRAM uses the same address signals for the read and write ports. For the CY7C1302 device, the address inputs to the QDR SRAM are sampled on the rising edge of K for reads and on the rising edge of K_bar for writes.
Data Inputs	Dout[17..0]	Read data output from QDR SRAM. Two words can be transferred from the QDR SRAM during each clock cycle because the QDR outputs data on the rising edges of C and C_bar. Data output by the SRAM on the rising edge of C is captured by the controller on the subsequent falling edge of C. Data output on the rising edge of C_bar is captured on the subsequent rising edge of C.
Data Outputs	Din[17..0]	Write data input to the QDR SRAM. Data is captured by the QDR on the rising edges of K and K_bar. Therefore, two words can be transferred to the QDR SRAM during each clock cycle.

System-Level Issues

This section describes some of the issues to be considered when implementing a QDR SRAM controller in an APEX 20KE device. Because the QDR SRAM interface is high speed, special care must be taken when designing the controller interface. Design-specific issues include the Expanded HSTL I/O interface, clock generation, and delay minimization.

HSTL I/O Pins

The HSTL I/O standard is required for the QDR SRAM interface. APEX devices and CY7C1302 SRAM devices support Expanded HSTL, which uses a 1.8-V I/O supply voltage (V_{DDQ}). Characterization data has shown that APEX 20KE devices can drive out and receive Expanded HSTL signals at 100 MHz.

HSTL is a voltage-referenced I/O standard, very similar to the SSTL standard, which is supported by the Quartus software. To implement the Expanded HSTL interface, perform the following procedure.

1. In the Quartus software, use the **Assignment Organizer** to make an I/O standard assignment of SSTL-2 Class I on all APEX pins that interface with the SRAM (`A[17..0]`, `din[17..0]`, `dout [17..0]`, `rps_bar`, `wps_bar`, `bws_bar[1..0]`, `K`, `K_bar`, `C`).

2. To implement the reference voltage pins required by the Expanded HSTL standard, use the Quartus software to place `VREF` pin assignments on all I/O banks that hold the interface pins. The same `VREF` placement rules apply for Expanded HSTL as for SSTL. See the *Using I/O Standards in the Quartus Software White Paper* for further details on placing SSTL `VREF` pins.

3. Use the Quartus software to generate an SRAM Object File (**.sof**) for the APEX device. The same SOF generated with SSTL I/O pins can be used for Expanded HSTL I/O pins.

4. On the board, connect the V_{CCIO} of each Expanded HSTL I/O bank to 1.8 V. Connect all `VREF` pins to 0.75 V and terminate the Expanded HSTL I/O signals as specified by the SRAM vendor.

Clock Generation

The clocking scheme used for the controller is important to maintain high frequency operation. This clocking scheme requires the use of a PLL, which is available on-chip in APEX 20KE devices, and three global clock resources in the APEX device. (When ordering devices, be sure to specify the –X suffix for PLL-enabled devices.) The following clocks are required for a QDR SRAM controller:

- Input clock from the system
- 1x and 2x controller clocks
- SRAM clocks `K` and `K_bar`
- Controller feedback clock `C`

The system must supply an input clock, nominally 100 MHz. This clock can then be fed to the on-chip PLL, which generates a 1x clock and a 2x clock for the controller. The 1x clock inputs data from the system and pipes it through the controller. The data in, address, and control lines to the SRAM can be clocked at a double data rate using the 2x clock.

The controller must also generate a differential clock signal (K and K_bar) for the QDR SRAM device. This action can be performed by using a 2x clock to clock a T flipflop and register the true and complement of the TFF output. The result is a 1x clock signal (K) and a 180-degree-phase-shifted 1x clock signal (K_bar).

K and K_bar are output from the APEX device and sent to the SRAM along with the data, address, and control lines. This clocking scheme negates the effect of signal skew on write and read request operations, because the propagation delays for K and K_bar from the APEX device to the SRAM are equal to the delays on the data signals. For proper operation, the board designer should take care to equalize the trace length (and therefore the flight times) of the data in, address, and control signals with the K and K_bar clocks.

When the K clock reaches the SRAM, it is fed back to the controller as clock C and used to clock in any data arriving from the SRAM to the controller. Because the SRAM outputs are also registered with clock C, the data sent from the SRAM arrives at the controller at the same time as the clock, reducing skew on read operations. The board designer should equalize trace lengths of the data out bus and the C clock signal.

Additionally, Altera recommends that the board designer place the APEX device adjacent to the SRAM on the circuit board. This positioning keeps trace length to a minimum and further minimizes any skew caused by board delay.

Timing

Because data is exchanged between the controller and the SRAM at high speeds, special care must be taken to avoid setup or hold violations for the SRAM or APEX device. This section discusses the timing issues that may arise when designing a high-speed interface.

Write Cycle

When designing for proper write-cycle timing, meeting the setup and hold requirements of the SRAM is the primary concern. Setup and hold specifications for the CY7C1302 (100-MHz speed grade) are 1 ns each.

Both the QDR clock and data signals are driven from the controller, so the clock-to-output delay from the APEX device pins is the same for both sets of pins. As determined by characterization, the clock-to-output delay from the APEX pins when using Expanded HSTL can range from 2.2 ns to 4.9 ns (depending on temperature) but is consistent for both sets of pins. The same principle applies to board delay, because flight times for clock and data signals are equalized.

The K and K_bar outputs are clocked out from two registers, clkout and clkout_bar. If these registers are placed in logic element (LE) registers in the adjacent logic array block (LAB), their clock-to-output time is only slightly greater than the clock-to-output time of the data and address signals. Because K and K_bar are clocked out on the positive edge of the 2x clock while data and address are clocked out on the negative edge, there is a cushion of 2.5 ns each way for timing purposes.

The following calculations apply for controller-to-SRAM data transfers. The calculation allows for up to a 1.2-ns difference between clock $\mathbf{t_{CO}}$ and data/address $\mathbf{t_{CO}}$, and up to 0.2 ns of board skew.

[$\mathbf{t_{CO}}$(APEX Clock) – $\mathbf{t_{CO}}$(APEX Data and Address)] +
Board Skew (Clock – Data) + $\mathbf{t_{SU}}$ (SRAM) < 2.5 ns

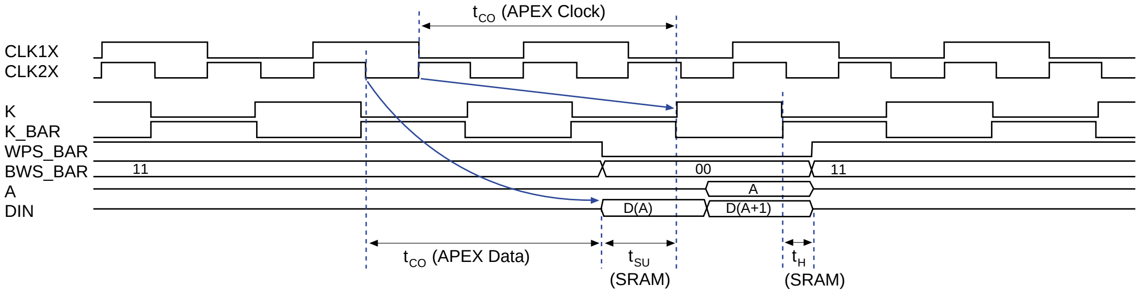
[4.9 ns – 3.7 ns] + 0.2 ns + 1.0 ns = 2.4 ns < 2.5 ns

[$\mathbf{t_{CO}}$(APEX Data and Address) – $\mathbf{t_{CO}}$(APEX Clock)] +
Board Skew (Clock – Data) + $\mathbf{t_{H}}$(SRAM) < 2.5 ns

[3.4 ns – 2.2 ns] + 0.2 ns + 1.0 ns = 2.4 ns < 2.5 ns

Figure 3 shows the write cycle timing waveform for the SRAM interface pins at 100 MHz.

Figure 3. Write Cycle Timing Waveform



Read Cycle

When read data is sent from the SRAM to the controller, setup times on the APEX device must be met. Low setup times can be achieved by using the programmable delay features available in the APEX device. Apply the following setting in the Quartus software to use the programmable delay feature, which decreases the delay from the `dout` input pins to the input registers, thereby decreasing the setup requirement:

```
DECREASE_INPUT_DELAY_TO_INTERNAL_CELLS = ON;
```

With this setting turned on, the setup requirement for the `dout` pins in an APEX 20KE device is 0.9 ns. Hold time requirements remain at 0.0 ns.

Arrival time of the `dout` signal is determined by the clock-to-output specification for the SRAM. For the CY7C1302, the maximum $\mathbf{t_{CO}}$ value is 3.0 ns, while the minimum $\mathbf{t_{CO}}$ value (i.e., data output hold time $\mathbf{t_{DOH}}$) is 1.2 ns. Board delay can be ignored, because flight times for the `C` signal and the `dout` bus are roughly equal. Regardless, the timing calculation allows for some skew between the clock and data lines.

Data is sent out of the SRAM on the rising edge of `C`, and is captured by the controller on the falling edge of `C`. For a clock speed of 100 MHz, there is a 5-ns window between the rising and falling edges. Subtracting a clock-to-output delay of 3 ns leaves 2 ns of setup time on the APEX pins. This timing meets the setup requirement of 0.9 ns, with 0.2 ns margin for any signal skew.

The following calculations apply for data transfer from the SRAM to the controller:

$\mathbf{t_{CO}}$ (SRAM) + Board Skew (Clock – Data) + $\mathbf{t_{SU}}$ (APEX) < 5 ns

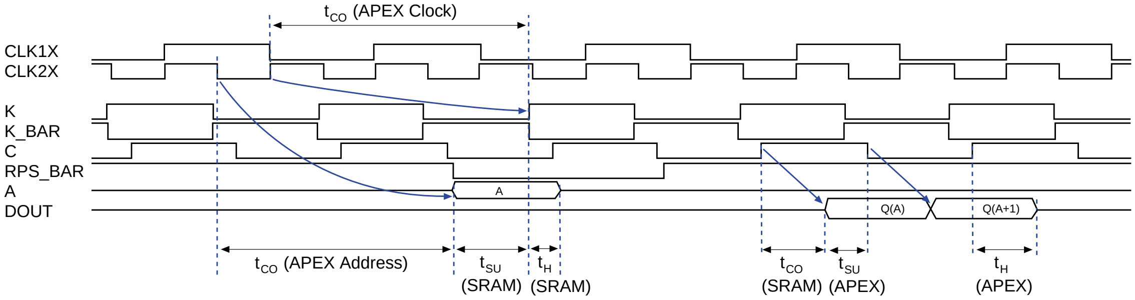
3.0 ns + 0.2 ns + 0.9 ns = 4.1 ns < 5 ns

$\mathbf{t_{DOH}}$ (SRAM) + Board Skew (Data – Clock) – $\mathbf{t_{H}}$ (APEX) > 0 ns

1.2 ns – 0.2 ns – 0.0 ns = 1.0 ns > 0 ns

Figure 4 shows the read cycle timing waveform for the SRAM interface pins at 100 MHz.

Figure 4. Read Cycle Timing Waveform



Conclusion

The QDR SRAM architecture was designed to greatly increase memory bandwidth for communications systems. Due to their speed and programmability, APEX 20KE devices are ideal for implementation of a QDR SRAM controller.

References

For more information , refer to the following documents:

- Cypress CY7C1302V25 QDR SRAM data sheet
- Altera *Using I/O Standards in the Quartus Software White Paper*

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com
Applications Hotline:
(800) 800-EPLD
Customer Marketing:
(408) 544-7104
Literature Services:
lit_req@altera.com