



XAPP214 (v1.0) July 24, 2000

Virtex Device Quad DataRate (QDR) SRAM Interface

Author: Tony Williams

Summary

The Virtex™ series of FPGAs provides access to a variety of on-chip and off-chip RAM resources. In addition to the on-chip distributed RAM and block SelectRAM+™ features, Virtex FPGAs are able to interface to a variety of external high-speed memory devices. The combination of high-speed SelectI/O™ resources and on-chip Delay-Locked Loop (DLL) circuits enables a high-bandwidth interface to Quad DataRate (QDR™) architecture SRAMs. This application note describes the implementation of an interface using the Cypress CY7C1302V25 QDR SRAM.

Introduction

With the continuous demand for higher performance data processing systems, memory devices are evolving to more closely match the needs of these applications. Beyond simply making conventional memory devices faster, specialized memory products that optimize memory bandwidth for a specific system architecture are successfully increasing overall performance in a variety of data processing systems. Examples of specialized memory products include dual-port memories, FIFOs, and CAMs.

The Quad Datarate (QDR) SRAM architecture is one such evolution. Departing from the conventional memory architecture, the QDR SRAM provides two separate memory-access ports: a read port, and a write port. Both ports operate independently of each other and provide the means to perform simultaneous reads and writes into memory. Established Double Datarate (DDR) technology is used to accelerate the throughput of each port, resulting in an overall quadrupling of memory bandwidth. Since both ports are separated, the problems associated with conventional bidirectional read/write ports (such as bus turnaround) are avoided.

This application note presents a general-purpose interface to a Cypress Semiconductor QDR SRAM device. Implemented in a Virtex XCV150 device to operate internally at 100 MHz (externally at 200 MHz), the interface achieves a peak bandwidth of 900 MBytes/s with the QDR SRAM 18-bit data ports. The design occupies approximately 110 slices (6% of the available resources of an XCV150) and is available in the form of a synthesizable VHDL and user constraints file.

QDR SRAM Architecture

The QDR SRAM device described is the Cypress Semiconductor CY7C1302V25, a 2.5V synchronous pipelined 9 Mb SRAM, organized as 256K words of 36 bits (Figure 1).

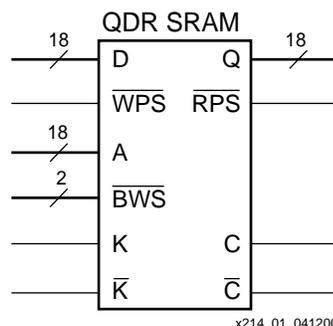


Figure 1: Cypress QDR SRAM Inputs and Outputs.

The device provides two independent memory access ports, one for read access, one for write access, and a common 18-bit address bus. Two control signals, Read Port Select (RPS) and Write Port Select (WPS), control activity on the two ports.

The device is organized internally as a 36-bit memory with 18-bit read and write data ports, where individual memory access always involves the exchange of two 18-bit data words. Writes may involve all 36 bits or any combination of 9-bit groups, depending on the two Byte Write Select (BWS) control inputs.

The QDR SRAM monitors the addresses to which simultaneous reads and writes occur. If a match is detected, the data being written to memory is "forwarded" to the read port. In this manner, the most up-to-date data is always available to the read port, even during simultaneous read/writes.

The QDR SRAM uses a pair of differential clocks, one for the write and address ports (K and K-bar), and one for the read port (C and C-bar). The purpose of the read-port clock is to enable board level deskewing of data launched from several QDR SRAM devices. In this application note, a single QDR SRAM is used, and the device is operated in a single clock mode (both C and C-bar tied to V_{DD}) where K / K-bar controls both read and write accesses.

QDR Read and Write Timing

Read and write access to the QDR SRAM begins on the rising edge of the QDR SRAM clock, K. A memory read is initiated by asserting RPS and providing the desired read address at the A inputs. With the QDR SRAM in single-clock mode (used throughout this application note), the first group of 18 bits from the selected memory location appears at the Q outputs on the next rising clock edge. The second group of 18 bits appears at the outputs on the following falling clock edge (Figure 2).

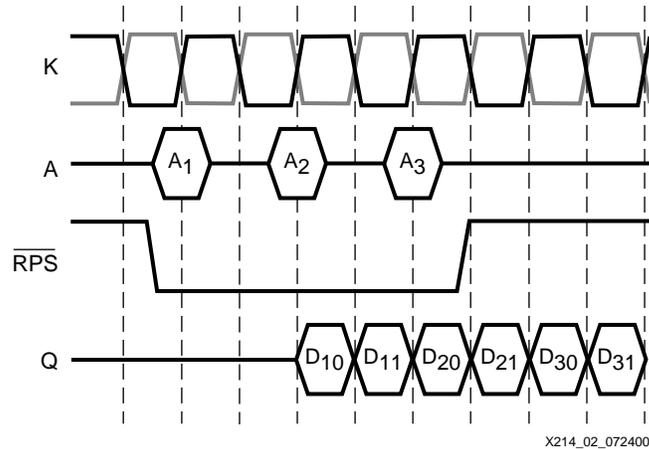


Figure 2: QDR SRAM Read Access Timing

Where all 36 data bits are being written, a memory-write is initiated by asserting \overline{WPS} and providing the first group of 18 data bits at the D input. The falling edge of K is then used to latch the second group of 18 data bits at the D input and the address at the A input. See (Figure 3).

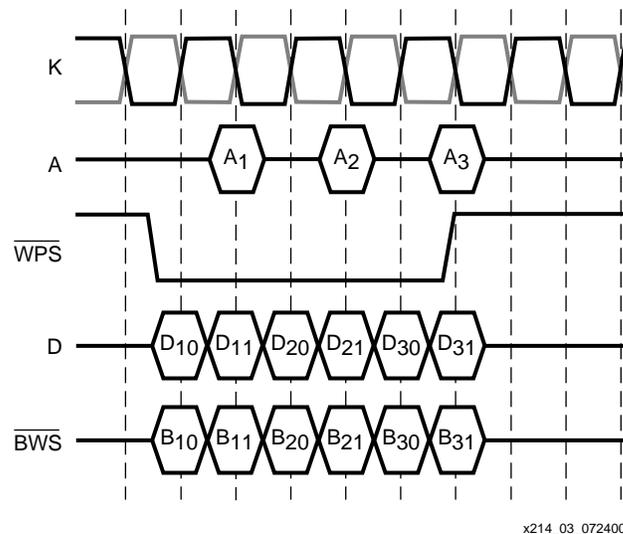


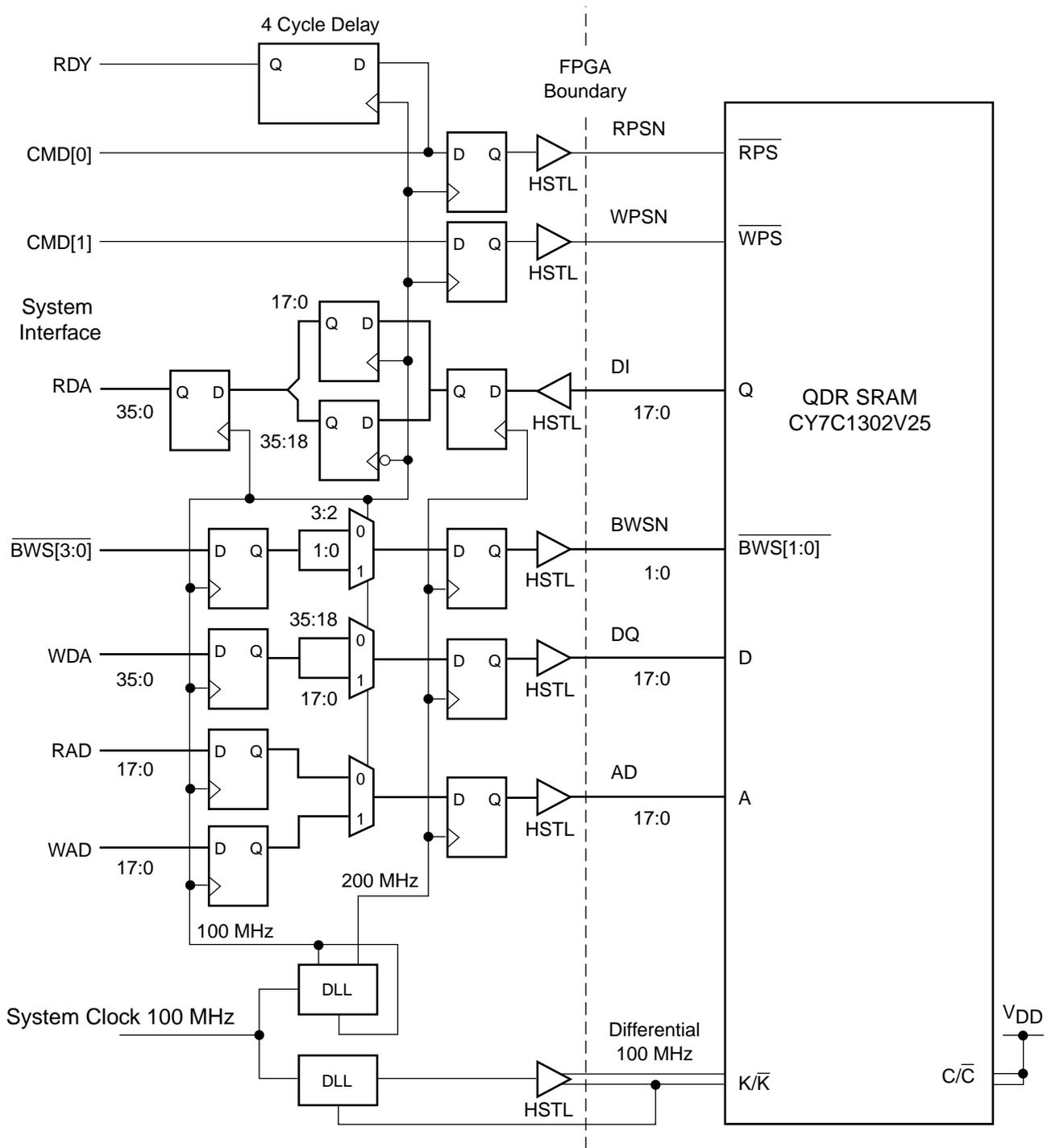
Figure 3: QDR SRAM Write Access

Read and write accesses may overlap, since the Q port is used only during a read and the D (and \overline{BWS}) ports are used only during writes. The shared address bus, A, is time-multiplexed, with read addresses being latched on rising clock (K) edges, and write addresses captured on falling edges.

During write access, 18-bit write data may be divided into two 9-bit groups under the control of the \overline{BWS} control inputs. Asserting $\overline{BWS}[0]$ permits the lower 9 bits [8:0] of data to be written to memory.

The Interface

A functional block diagram of the QDR SRAM interface is shown in Figure 4. The main function performed by the interface is the datarate doubling necessary in both the read and write datapaths.



x214_04_072400

Figure 4: Functional Block Diagram

Memory accesses can be initiated on every cycle of the 100 MHz system clock, and are triggered by asserting the appropriate code on the command (CMD) inputs. Command codes are described in [Table 1](#).

Table 1: Interface Command Codes

| CMD[1:0] | Function |
|----------|-----------------------------|
| 11 | No Operation |
| 10 | Read Access |
| 01 | Write Access |
| 00 | Simultaneous Read and Write |

Write Datapath

On the system side, the write-request input CMD[1] is asserted, and an 18-bit write address (WAD), 4-bit $\overline{\text{BWS}}$ word, and 36-bit data word (WDA) are presented to the interface. The data is separated into two 18-bit words before being communicated to the QDR SRAM at 200 MHz. The address is similarly communicated to the SRAM, but is first interleaved with any memory-read accesses taking place.

During memory writes, groups of 9 bits may be individually masked, preserving the corresponding bits in memory. The four active Low $\overline{\text{BWS}}$ inputs are linked to 9-bit groups as shown in [Table 2](#). Deasserting all four $\overline{\text{BWS}}$ inputs effectively cancels the write operation.

Table 2: Relationship between $\overline{\text{BWS}}$ and WDA bits

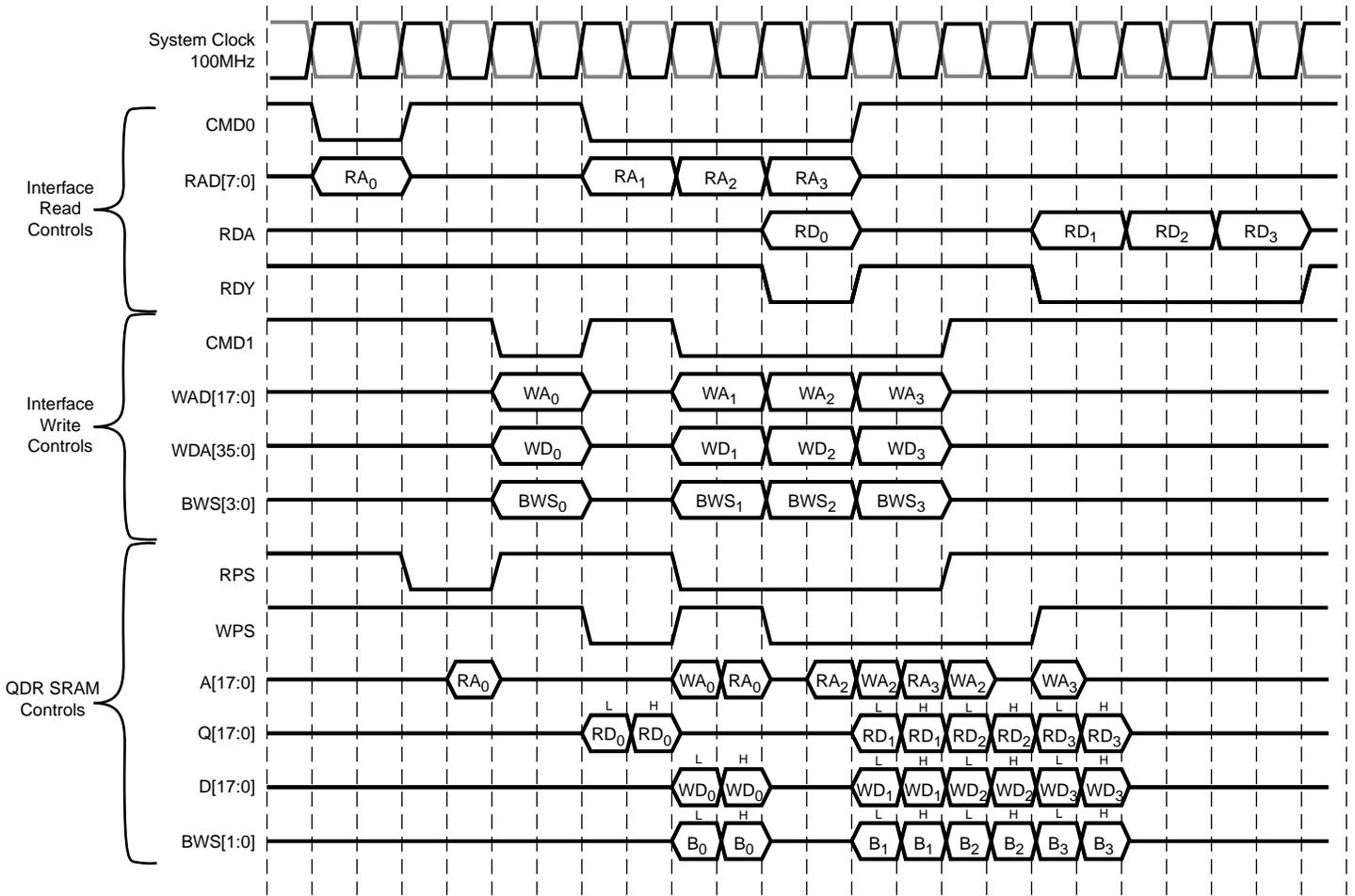
| | |
|----------------------------|------------|
| $\overline{\text{BWS}}[0]$ | WDA[8:0] |
| $\overline{\text{BWS}}[1]$ | WDA[17:9] |
| $\overline{\text{BWS}}[2]$ | WDA[26:18] |
| $\overline{\text{BWS}}[3]$ | WDA[35:27] |

Read Datapath

On the system side, an 18-bit read address is presented to the interface and the read-request input CMD[0] is asserted. The read access is interleaved with any current memory-write accesses. Before presentation to the system, the interface receives and reformats both 18-bit data words from the QDR SRAM at 200 MHz into the intended 36-bit word width.

Since the reformatting process and QDR SRAMs memory access cycle are both pipelined, four system clock cycles are consumed before the requested data is available. The interface indicates the availability of data by asserting its ready output $\overline{\text{RDY}}$.

Memory read, write, and simultaneous read/write cycles are illustrated in [Figure 5](#).



x214_05_072400

Figure 5: Example Timing Waveforms Showing Read, Write, and Simultaneous Read/Write Memory Accesses

Functional Description

In Figure 4 on page 4, data is exchanged between the FPGA and the QDR SRAM on both rising and falling edges of K, the memory clock. To achieve this, the 100 MHz system clock is doubled using the Virtex on-chip DLL. The resulting 200 MHz clock is distributed to rising-edge triggered I/O flip-flops. The use of DLLs ensures that the external 100 MHz memory clock and both internal 100 MHz and 200 MHz clocks are precisely phase locked with the source system clock.

All double-datarate communication between the FPGA and the QDR SRAM is implemented using I/O registers placed in the FPGA I/O blocks. This achieves precise control over the I/O timing.

During memory writes, the 36-bit write-data and accompanying 4-bit \overline{BWS} word are divided into two 18-bit words, and two 2-bit words respectively. The division is performed by a multiplexer where the select line is the 100 MHz internal clock. The low-order bits are selected during the High phase of the clock, the high-order bits during the Low phase. The output registers, clocked at 200 MHz, capture both 18-bit data words and both 2-bit \overline{BWS} words, launching them toward the QDR SRAM.

Similarly, the 18-bit read and write addresses are captured from the system side of the interface and multiplexed into the output register. Once again, the multiplexer select is the

Board Layout Guidelines

The following guidelines are for the Virtex -6 speed grade and QDR SRAM -100 speed grade devices operating at 100 MHz as described above.

- Place the Virtex device and the QDR SRAM physically close to each other.
- To minimize crosstalk, keep all address, data and control lines as short as possible.
- To ensure identical propagation delays, match the memory clock traces (K and \bar{K}) as closely a possible.
- Have the feedback connection between the memory clock (K) and the Virtex device commence at the mid-point of the trace connecting the Virtex device and the QDR SRAM.
- All signals between the Virtex device and the QDR SRAM are HSTL Class I. Refer to application note [XAPP133](#) (Using the Virtex Select/O Resource) for guidelines on termination techniques and simultaneous switching guidelines.

Conclusion

VHDL reference design code is available on the Xilinx web site at [xapp214.zip](#) or [xapp214.tar.gz](#).

Revision History

The following table shows the revision history for this document.

| Date | Version | Revision |
|---------|---------|-------------------------|
| 7/24/00 | 1.0 | Initial Xilinx release. |