XAPP228 (v1.0) September 24, 2002

Quad-Port Memories in Virtex Devices

Author: Nick Sawyer and Marc Defossez

Summary

This application note describes how the existing dual-port block memories in the Spartan™-II and Virtex™ families can be used as Quad-Port memories. This essentially involves a data access time (halved) versus functionality (doubled) trade-off. The overall bandwidth of the block memory in terms of bits per second will remain the same.

Introduction

The Virtex architecture features access to internal block memories. The Virtex, Spartan-II, Spartan-IIE, and Virtex-E families contain 4K bit blocks. The Virtex-II and Virtex-II Pro™ families have 18K bit blocks. These blocks are fully synchronous, true dual-port structures, i.e., the user can Read or Write to and from each port independently (with the exception of simultaneous Read and Write to the same address). In addition, each port has a separate clock, and the data widths for each port are independently programmable, ideal for data transfer between clock domains, i.e., FIFOs. A block diagram of the dual-port RAM blocks is shown in Figure 1.

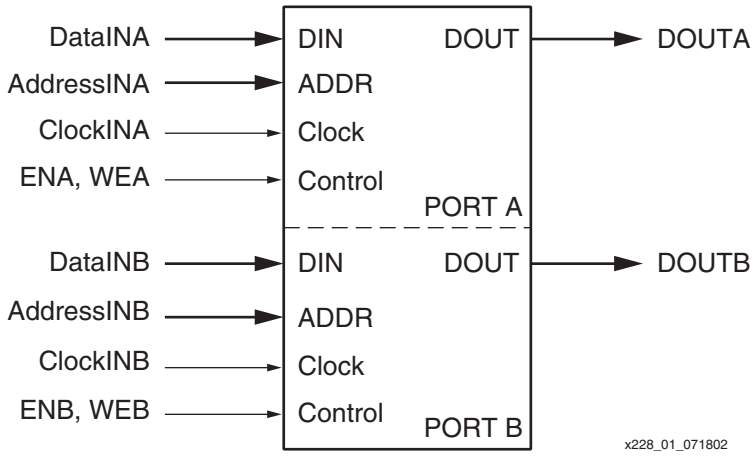

Figure 1: **Basic Block RAM Structure**

Some applications need (or are more efficient when using) quad-port memories. It is very easy to implement a quad-port memory function using the existing dual-port RAMs by adding a doubled clock and some extra logic.

Implementation

There are now four ports available to the designer, A, B, C, and D. Each can be used for independent Read or Write, with two provisos. There should never be a simultaneous Read and Write to the same address. The same clock (CLKA) should be shared for port A and port C and similarly, port B and port D should also share the same clock (CLKB).

Each of the clock inputs is doubled in frequency using a DLL in the Virtex series or Spartan families, or a DCM in the Virtex-II series. This doubled clock is phase aligned by the DCM to the original (system) clock, reducing the potential for asynchronous logic. The operation of the circuit is shown in Figure 2. A block diagram of the four-port functionary is shown in Figure 3.

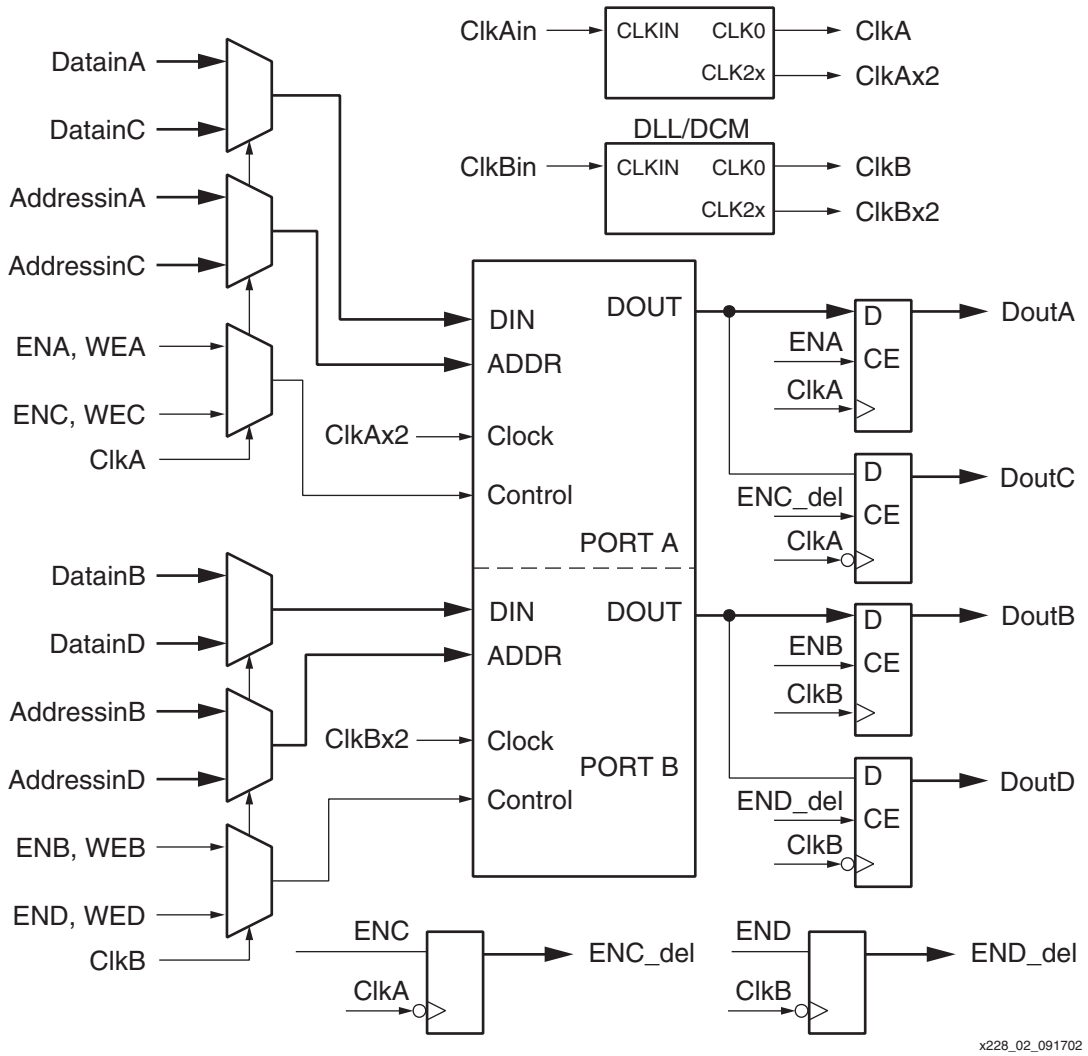

Figure 2: **Quad-Port Functionality**

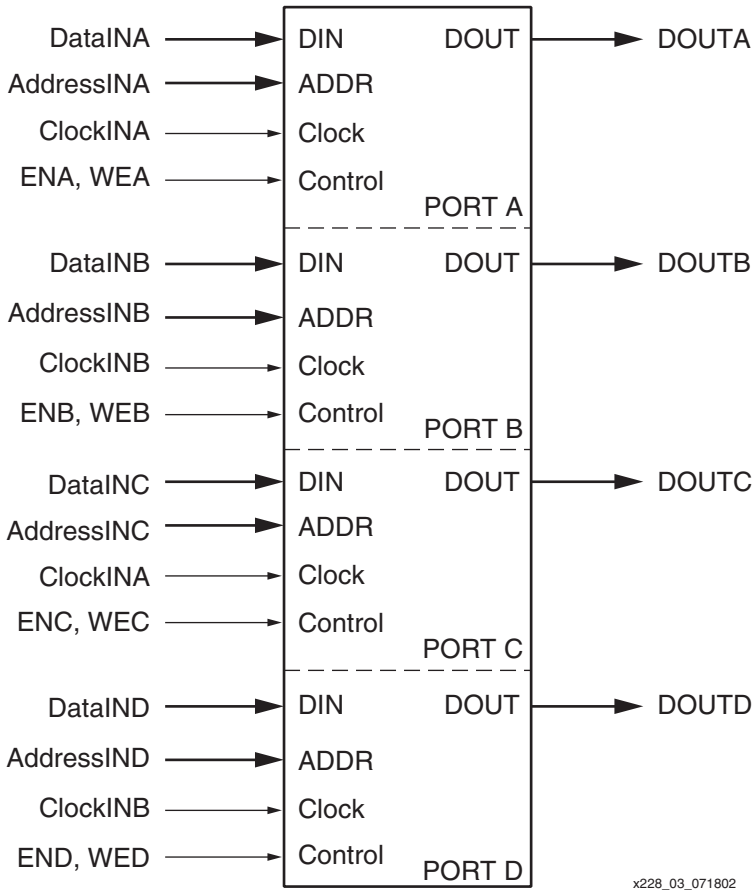

Figure 3: **Four-Port Functionality**

A timing diagram is shown in Figure 4. Considering the case for ports A and C (B and D are identical). At the rising edge of the system clock, the data, address and control for ports A and C will all change (point 1, in Figure 4). The information for port A is now applied to the RAM via a multiplexer. As the RAM is running at twice the rate of the system clock, this information will be registered into the memory at the falling edge of the system clock, i.e., the next rising edge of the 2x clock (point 2, in Figure 4). Output data for port A will be available after the clock-to-out time for the RAM, and as the system will expect this data to be valid for a whole cycle, it is then re-registered on the rising edge of the system clock (point 3, in Figure 4), so as to be valid when necessary. Meanwhile the MUX changes state at position 2, and the data address and control for port C pass through the multiplexer on the second half of the clock cycle. They are registered by the RAM on the rising edge of the system clock in the normal way (point 3, in Figure 4). The output data for port C will be valid following the clock-to-out time of the RAM, and is then reregistered by the falling edge of the system clock (point 4, in Figure 4) so as to be valid when necessary. The system will therefore see valid data for both ports at point 5 in Figure 4.

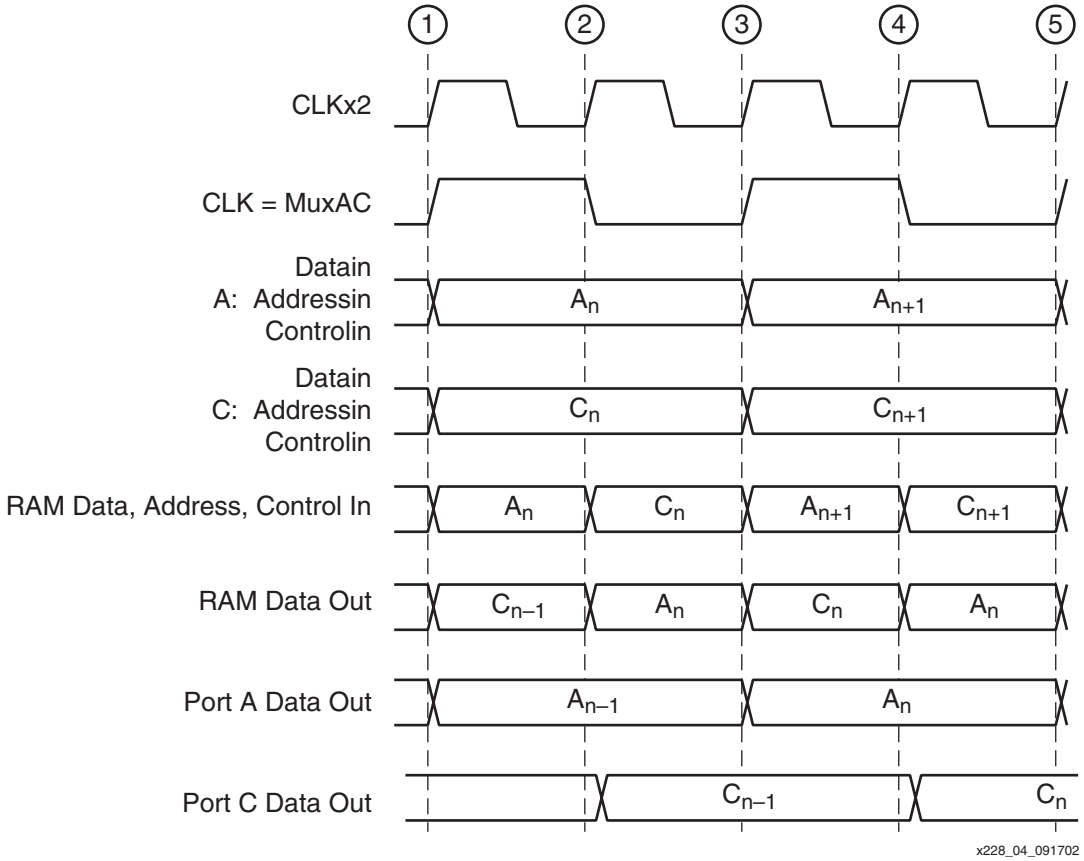

Figure 4: **Timing Diagram**

The multiplexer control signal is the clock. For easier timing analysis, minimize the number of logic connections (as opposed to clock) on the clock tree. The signal is the inverse of the clock signal reregistered by the x2 clock. There is only one logic load on the clock signal making it easier to control.

Because of over-clocking, the frequency of operation of the block RAM is effectively reduced. However, the quad-port RAM will work with clocks of up to 125 MHz in a Virtex-II -5 speed grade device, and up to 80MHz in a Virtex-E -7 speed grade device.

Reference Design

The design concept is suitable for use in either the Spartan-II, Virtex, or Virtex-II series of devices. The only difference is the size of the block RAM; 4K bits in the Spartan and Virtex series with a maximum data width of 16 bits, and 18K bits in the Virtex-II series with the maximum data width of 36 bits when using the parity bits as well. The fully synthesizable design files are available on the Xilinx FTP site in **xapp228.zip**. Written in both VHDL and Verilog, the reference design files include examples of a quad-port memory for the Virtex-E device where each port is 256 x 16, and for a 1K x 18 four-port memory for the Virtex-II series. Each of the four ports is fully Read and Write enabled.

Conclusion

Because the RAM is now being double-rate clocked, the maximum frequency of operation will be reduced by about 50%. The additional logic overhead includes extra logic and flip-flops. However, there are many situations where the availability of a true quad-port memory is still desirable. As an example, when the user has a requirement for two wide and shallow FIFOs not requiring all the RAM bits available in a block memory. By using a quad-port memory, and allocating half of the RAM to each FIFO, the design will use one block RAM and not two. Another example would be a coefficient generator, using one port to Read and /or Write updated coefficients, while the other three ports can be used in a Read only mode.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
09/24/02	1.0	Initial Xilinx release.