



XAPP229 (v1.1) April 26, 2004

## Wider Block Memories

Author: Nick Sawyer, Marc Defossez

### Summary

This application note describes how memories wider than 36 bits can be efficiently implemented in the Virtex-II™ and Spartan™-3 architectures. The clock doubling method used is similar to the method described for quad-port memories in [XAPP228](#). The resulting memories are used in either dual-port or single-port mode.

### Introduction

Ever since the original Virtex architecture was introduced, access to block memories in Xilinx FPGAs has existed. These memories are 18-Kbit blocks in Spartan-3, Virtex-II, and Virtex-II Pro™ devices. The blocks are fully synchronous, true dual-port structures; that is, the user can read or write to and from each port independently (with the exception of simultaneous read and write operations to the same address). In addition, each port has a separate clock. The data widths for each port are independently programmable, which is ideal for data transfer (FIFOs) between clock domains. [Figure 1](#) shows a block diagram of the dual-port RAM blocks.

Some applications need (or are more efficient when using) wider and shallower block memories than the 512 deep x 36 wide dual-port mode or the 256 deep x 72 wide single-port mode (the widest memories available in the Virtex-II or Spartan-3 architectures). Two or more block memories can be used in parallel, but this can be inefficient when a requirement exists for fairly shallow memory. However, if the requirement is for very shallow (only 16 or 32 deep) memory, then the most efficient solution uses the distributed RAM mode of the Virtex-II or Spartan-3 CLB logic.

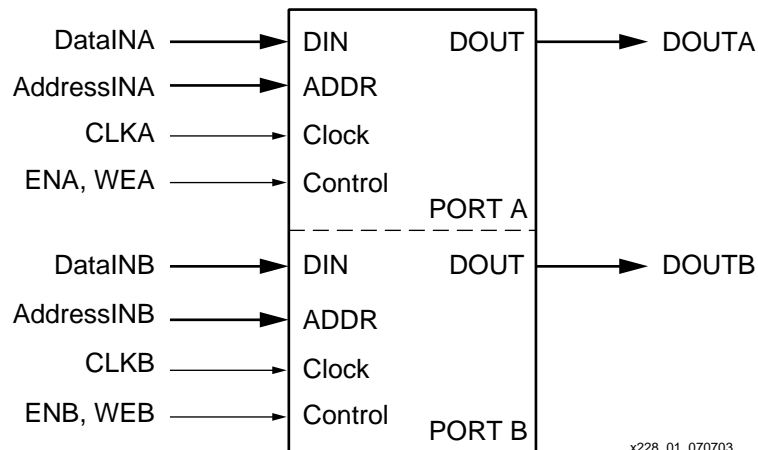


Figure 1: Basic Block RAM Structure

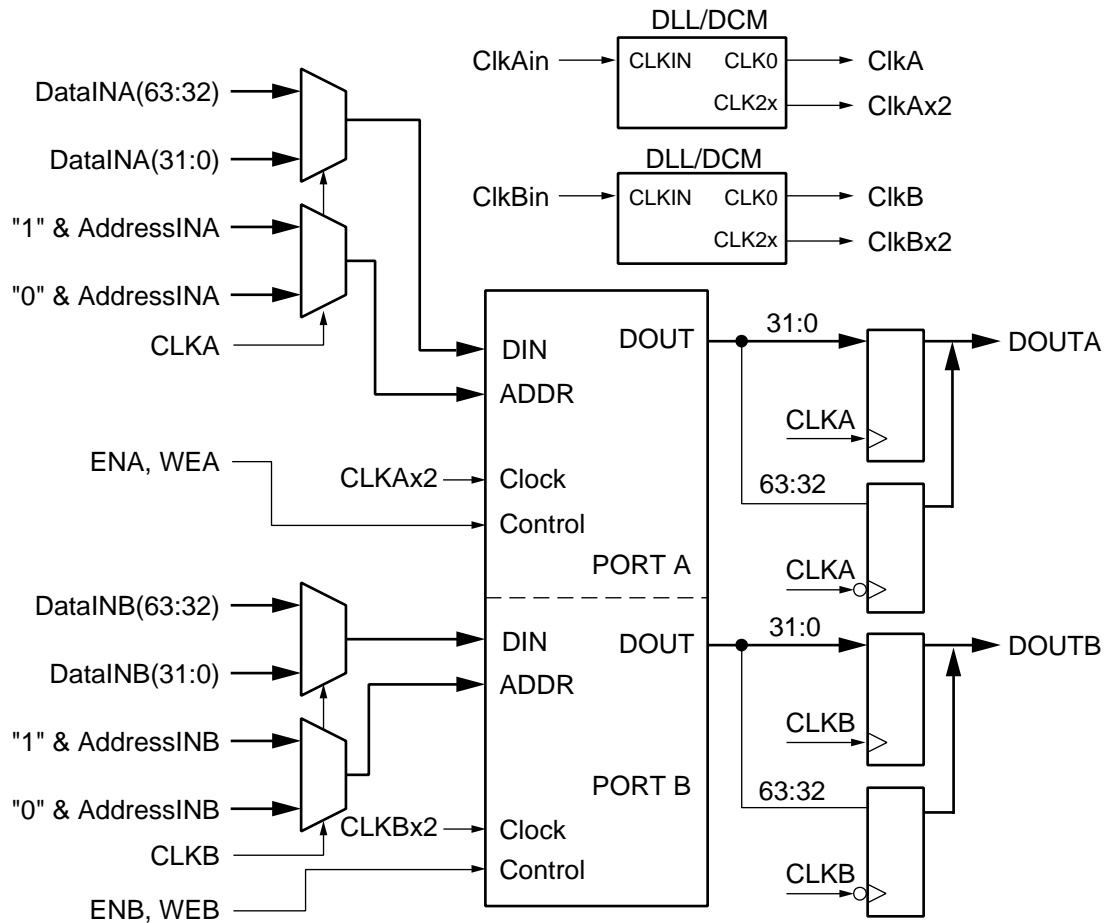
© 2003 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and further disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change 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.

## Dual-Port Implementation

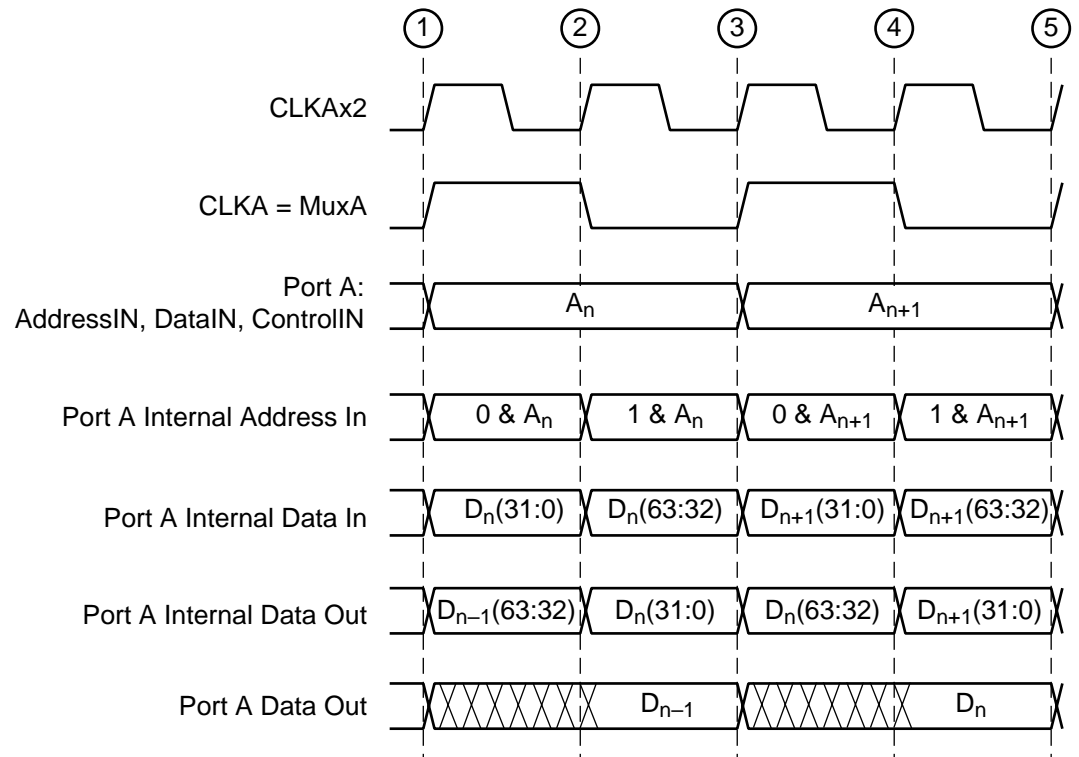
Operation of the dual-port circuit is shown in Figure 2, with a timing diagram shown in Figure 3. There are still two ports available to the user; however, each is now twice as wide, and half as deep as a basic block RAM. This provides the 256 deep x 72 wide dual-port memory functionality, which obviously works at a slower frequency than that of the basic block RAM, due to the extra logic involved. The multiplexing of the extra (parity) data bits is identical to that of the main data bits, and for the sake of clarity this is not shown.

Using a Digital Clock Manager (DCM), each of the two clock inputs is doubled in frequency. This doubled clock is phase aligned by the DCM to the original (system) clock, eliminating concerns about asynchronous logic.



x229\_02\_041604

Figure 2: Wide Dual-Port Functionality



x229\_03\_070703

Figure 3: Wide Dual-Port Timing Diagram

In Figure 3, note that only Port A is shown; the timing for Port B is identical.

Considering the case for ports A (B is identical), at the rising edge of the system clock, the data, address, and control for port A all change (position 1 in Figure 3). The low word of the data is now applied to the RAM via a multiplexer, together with the supplied address plus an extra address bit, which is set to zero.

Because the RAM is running at twice the rate of the system clock, this information is registered into the memory at the falling edge of the system clock; that is, the next rising edge of the 2x clock (position 2 in Figure 3). The low word of the output data for port A is available after the clock-to-out time for the RAM. Since the system expects this data to be valid for a whole cycle, it is re-registered on the rising edge of the system clock (position 3 in Figure 3), making it valid when necessary.

Meanwhile, the MUX changes state at position 2, and the low word of data and the supplied address plus an extra address bit set to '1' 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 (position 3). The high word of output data is valid following the clock-to-out time of the RAM, and is then re-registered by the falling edge of the system clock (position 4) making it valid when necessary. The system therefore sees valid data for the port at position 5.

The signal used to control the multiplexer is the clock itself, but it is a good idea to minimize the number of logic (as opposed to clock) connections on the clock tree, as this makes timing analysis easier. The signal actually used is therefore the clock signal re-registered by the x2 clock. Because there is only one logic load on the clock signal, the signal is easier to control.

## Single-Port Implementation

A wide single-port block RAM is a specific implementation of the dual-port design described previously, where the clock for both port A and port B is the same. The low half of the data word is applied to and read from port A, and the high half of the data word is applied to and read from port B. The only functional difference is in the addressing mechanism. The RAM is split into four quadrants, two for the double access that occurs in port A, and two for port B. Using this mechanism, the result is a single-port RAM that is 128 deep x 144 wide. The block diagram for this functionality is shown in Figure 4, and the timing is shown in Figure 5

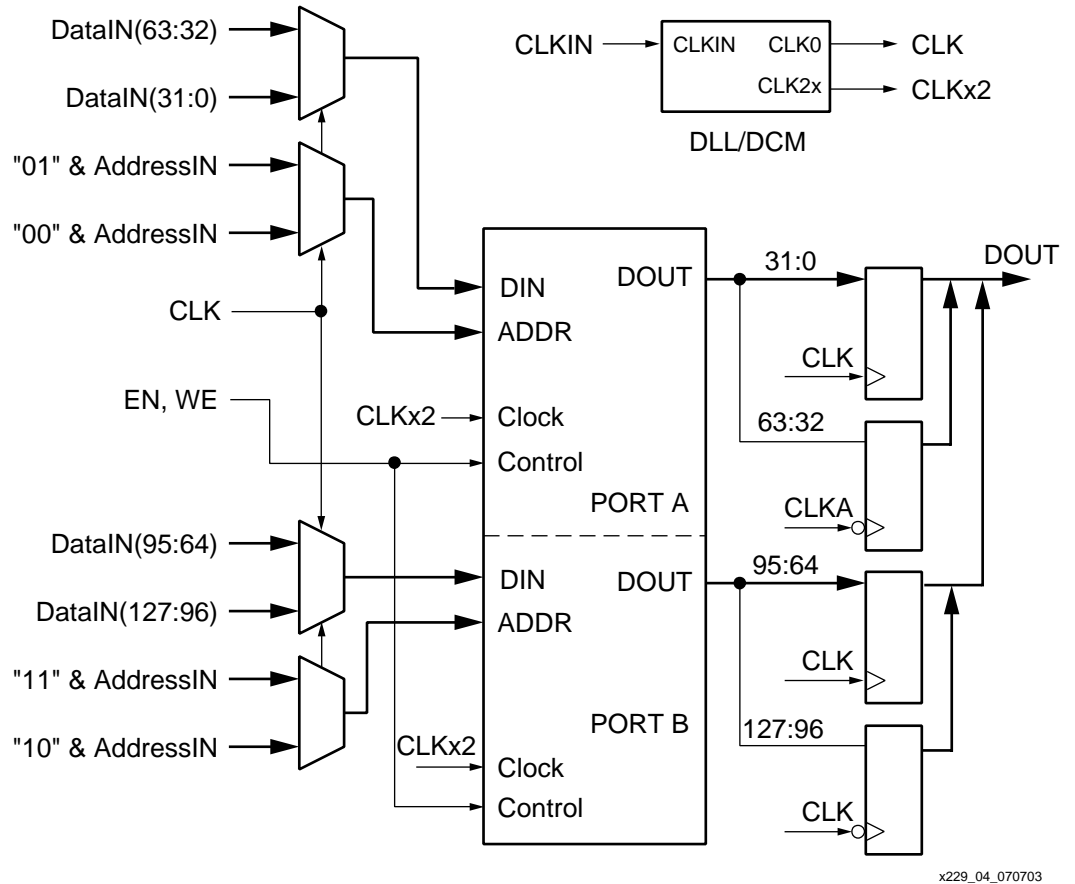


Figure 4: Wide Single-Port Functionality

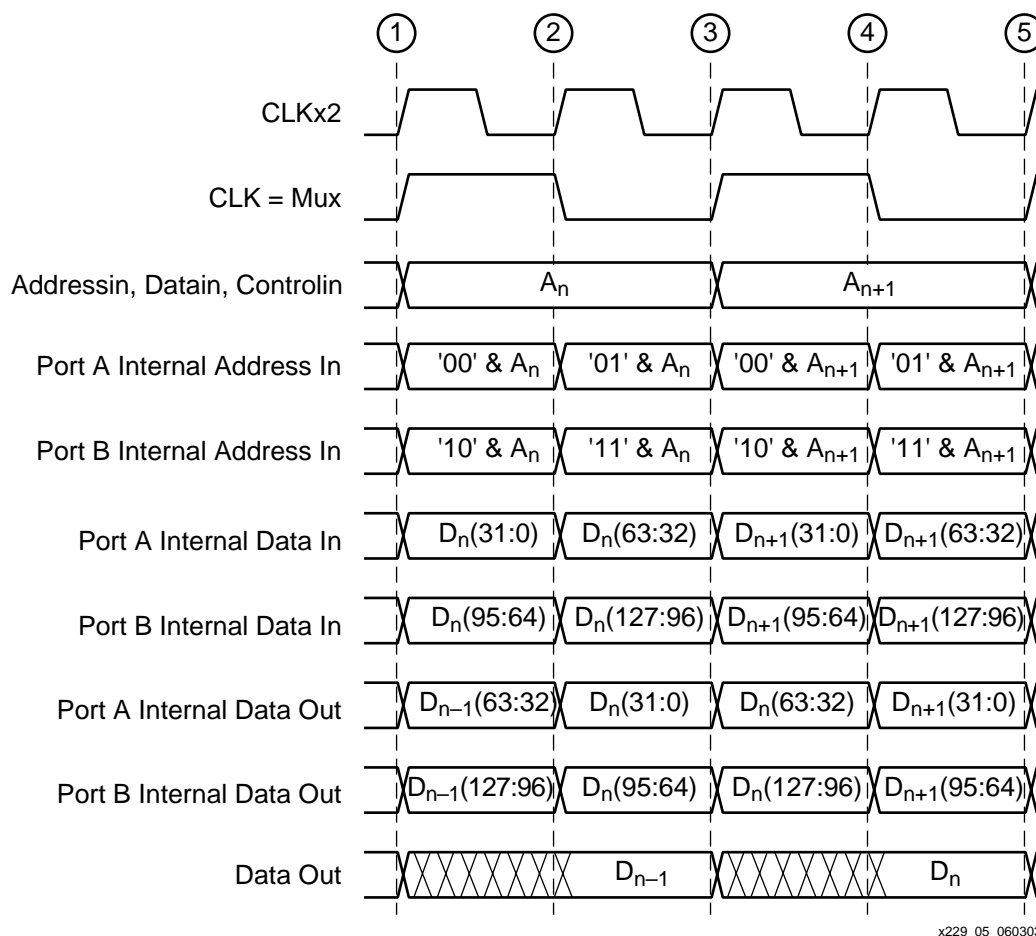


Figure 5: Wide Single-Port Timing Diagram

## Reference Design

Either design concept is suitable for use in either Spartan-3, Virtex-II, or Virtex-II Pro devices. The design files in [XAPP229.zip](#) are fully synthesizable and are provided in both VHDL and Verilog for the single-port and dual-port cases described in this application note.

## Conclusion

Because RAM is now being double-rate clocked, the maximum frequency of operation is reduced by about 50%. The overhead due to extra required logic and flip-flops is also a factor. However, many situations exist where the availability of efficient wider memory can more than make up for these drawbacks. One example is dual-port buffers requiring only 8 Kbits of data per buffer, but the data is 64 bits wide. By implementing this in two block RAM memories, it is only 22% efficient ( $8K/(18K + 18K)$ ). Using the technique described in this application note, only one block RAM is required, and the efficiency rises to 45%, thus meeting speed requirements. The single-port and dual-port reference designs both run at around 140 Mhz in a Virtex-II Pro grade -6 device, without floorplanning.

## Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/27/03	1.0	Initial Xilinx release.
04/26/04	1.1	Changed two signal labels in <a href="#">Figure 2</a> .