7. On-Chip Memory Implementations Using Cyclone Memory Blocks

C51007-1.4

Introduction

Cyclone® devices feature embedded memory blocks that can be easily configured to support a wide range of system requirements. These M4K memory blocks present a very flexible and fast memory solution that you can use to provide excellent memory bandwidth and density for a host of cost-sensitive applications.

You can use M4K memory blocks in various memory modes, including single-port, simple dual-port, true dual-port (also known as bidirectional dual-port), shift-register, ROM, and first-in first-out (FIFO) mode. M4K memory blocks also include advanced features such as support for byte-enable operation, parity-bit-based error correction, and mixed-port widths. This chapter describes these modes and other characteristics of the M4K memory blocks.

M4K Memory Features

Table 7–1 summarizes the features supported by the M4K memory block.

Table 7–1. Summary of M4K Memory Features (Part 1 of 2)

Performance	250 MHz
Total RAM bits (including parity bits)	4,608
Configurations	4K × 1 2K × 2 1K × 4 512 × 8 512 × 9 256 × 16 256 × 18 128 × 32 128 × 36 *(1)*
Parity bits	✓
Byte enable	✓
Single-port memory	✓
Simple dual-port memory	✓
True dual-port memory	✓
Embedded shift register	✓
ROM	✓

Table 7–1. Summary of M4K Memory Features (Part 2 of 2)	
FIFO buffer	✓
Simple dual-port mixed width support	✓
True dual-port mixed width support	✓
Memory initialization (.mif)	✓
Mixed-clock mode	✓
Power-up condition	Outputs cleared
Register clears	Input and output registers *(2)*
Same-port read-during-write	New data available at positive clock edge
Mixed-port read-during-write	Outputs set to unknown or old data

Notes to Table 7–1:

(1) The Altera® Quartus® II software will automatically cascade or concatenate multiple M4K memory blocks to provide deeper or wider memory functions.

(2) Asserting the clear port of the `rden` and byte-enable registers drives the output of these registers high.

Table 7–2 shows the memory capacity for M4K memory blocks in each Cyclone device.

Table 7–2. M4K Memory Distribution in Cyclone Devices			
Device	**Columns**	**Blocks**	**Total RAM Bits**
EP1C3	1	13	59,904
EP1C4	1	17	78,336
EP1C6	1	20	92,160
EP1C12	2	52	239,616
EP1C20	2	64	294,912

Parity Bit Support

M4K memory blocks support an optional parity bit for each data byte. Of the 4,608 bits of storage space available in an M4K block, 512 are available for use as parity-bit storage. The parity bit, along with logic implemented in logic elements (LEs), can facilitate parity-checking methods of error detection to ensure data integrity. You can also use parity-size data words to store user-specified control bits or as extra data bits to provide support for 9-bit, 18-bit, or 36-bit wide memories.

Byte-Enable Support

Byte-enable signals can be used to mask the input data so that only specific bytes in memory are overwritten. The unwritten bytes retain the data value that was last written to them. The write-enable signal (wren) is used in conjunction with byte-enable signals (byteena) to control the M4K block's write operations. The default value for the byteena signal is high (enabled), in which case no bytes are masked and writing is controlled only by the wren signals.

Asserting the clear port of the byte-enable register drives the byte-enable signal to its default high level.

M4K blocks support byte write operations when the write port has a data width of 16, 18, 32, or 36 bits. Table 7–3 summarizes how byteena controls which bits are masked.

Table 7–3. Byte Enable for M4K Blocks *Notes (1), (2)*

byteena	datain × 18	datain × 36
[0] = 1	[8..0]	[8..0]
[1] = 1	[17..9]	[17..9]
[2] = 1	—	[26..18]
[3] = 1	—	[35..27]

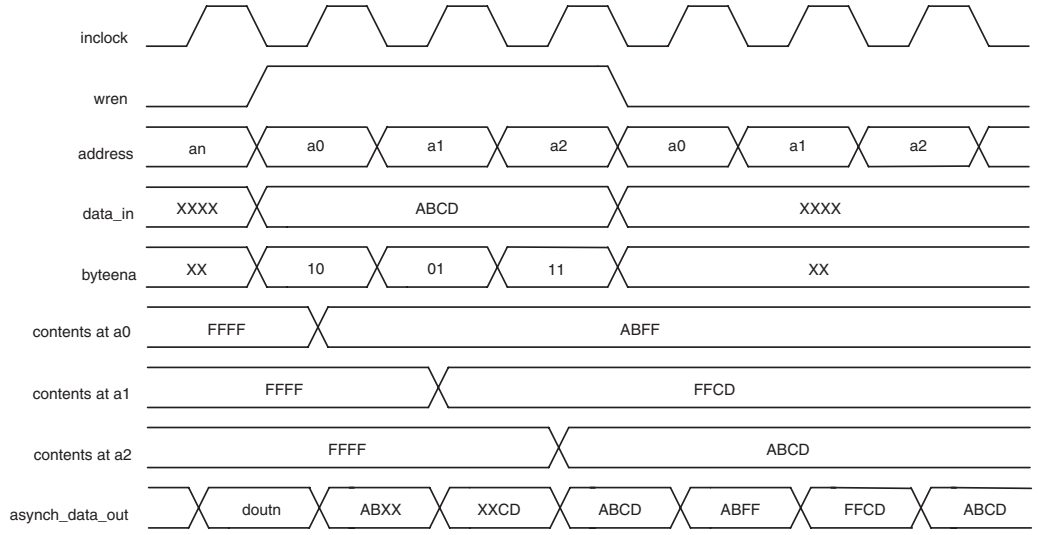
Notes to Table 7–3:

(1) Any combination of byte-enable signals is possible.

(2) Byte enables can be used in the same manner with 8-bit words, i.e., in × 16 and × 32 modes.

Figure 7–1 shows how both the wren and the byteena signals control the write operations of the RAM.

Figure 7–1. Byte-Enable Operation Functional Waveform



Power-up Conditions and Memory Initialization

Upon power-up, M4K memory is in an idle state. The outputs always power-up to zero, regardless of whether the output registers are used or bypassed. Even if a memory initialization file is used to pre-load the contents of the RAM block, the outputs will still power-up cleared. For example, if address 0 is pre-initialized to FF, the M4K blocks power-up with the output at 00.

Using M4K Memory

M4K memory blocks include input registers that synchronize write operations and output registers to pipeline designs and improve system performance. All M4K memory blocks are fully synchronous, meaning that all inputs are registered, but outputs can be either registered or combinatorial. M4K memory can emulate asynchronous memory.

Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

For more information, refer to *AN 210: Converting Memory from Asynchronous to Synchronous for Stratix and Stratix GX Designs*.

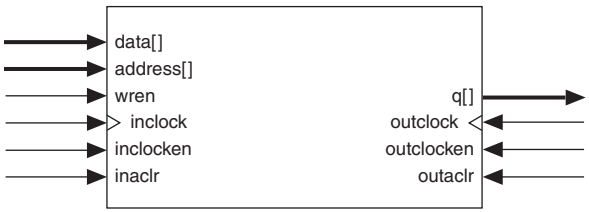
M4K memory blocks can operate in various modes, including:

- Single-port
- Simple dual-port
- True dual-port (bidirectional dual-port)
- Shift-register
- ROM
- FIFO

Implementing Single-Port Mode

Single-port mode supports non-simultaneous read and write operations. Figure 7–2 shows the single-port memory configuration for M4K blocks.

Figure 7–2. Single-Port Memory *Note (1)*



Note to Figure 7–2:
(1) Two single-port memory blocks can be implemented in a single M4K block.

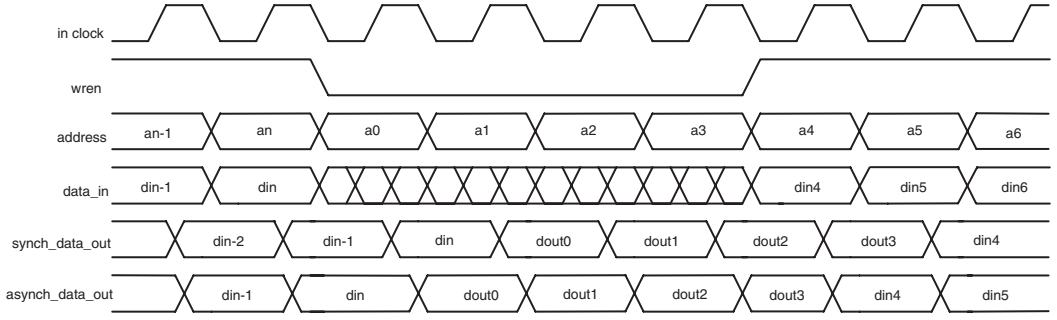
M4K memory blocks can also be divided in half and used for two independent single-port RAM blocks. The Quartus II software automatically uses this method of single-port memory packing when running low on memory resources. When deliberately assigning two single-port memories to one M4K block, first ensure that each of the two independent RAM blocks is equal to or less than half the size of the M4K block.

In the single-port RAM configuration, the outputs can only be in read-during-write mode, which means that during the write operation, data written to the RAM flows through to the RAM outputs. When the output registers are bypassed, the new data is available on the rising edge of the same clock cycle on which it was written.

For more information about read-during-write mode, see "Read-during-Write Operation at the Same Address" on page 7–20.

Figure 7–3 shows timing waveforms for read and write operations in single-port mode.

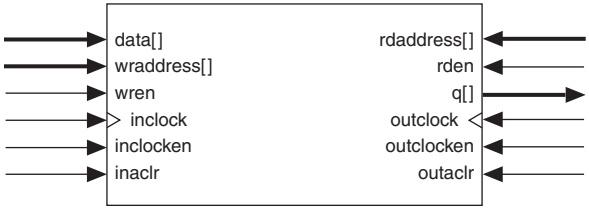
Figure 7–3. Single-Port Timing Waveforms



Implementing Simple Dual-Port Mode

Simple dual-port memory supports simultaneous read and write operations. Figure 7–4 shows the simple dual-port memory configuration for M4K blocks.

Figure 7–4. Simple Dual-Port Memory *Note (1)*



Note to Figure 7–4:
(1) Simple dual-port RAM supports read/write clock mode in addition to the input/output clock mode shown.

M4K memory supports mixed-width configurations, allowing different read and write port widths. This capability is useful for many applications, including implementing serializer-deserializers (SERDES) as well as interfacing with buses of differing widths. Table 7–4 shows the mixed-width configurations supported by the M4K blocks in Cyclone devices.

Table 7–4. M4K Block Mixed-Width Configurations (Simple Dual-Port Mode)

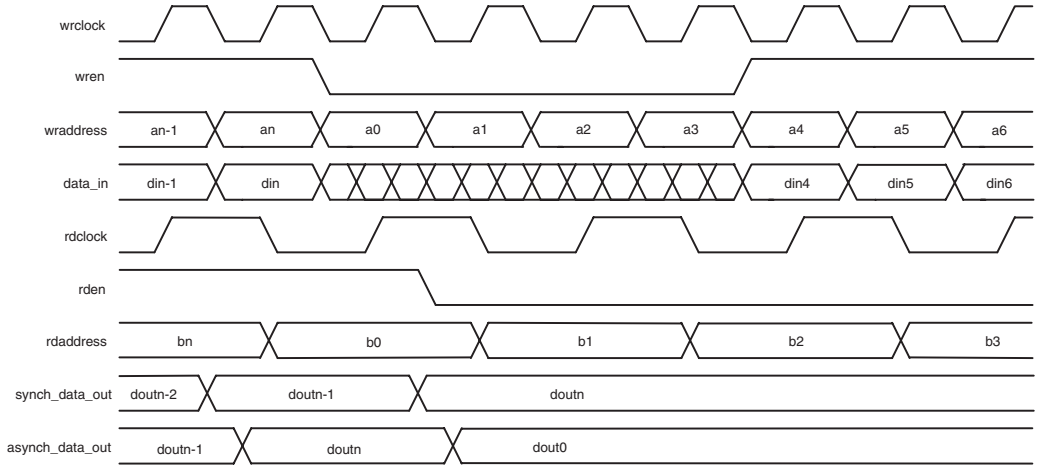
Read Port	Write Port								
	4K × 1	2K × 2	1K × 4	512 × 8	256 × 16	128 × 32	512 × 9	256 × 18	128 × 36
4K × 1	✓	✓	✓	✓	✓	✓	—	—	—
2K × 2	✓	✓	✓	✓	✓	✓	—	—	—
1K × 4	✓	✓	✓	✓	✓	✓	—	—	—
512 × 8	✓	✓	✓	✓	✓	✓	—	—	—
256 × 16	✓	✓	✓	✓	✓	✓	—	—	—
128 × 32	✓	✓	✓	✓	✓	✓	—	—	—
512 × 9	—	—	—	—	—	—	✓	✓	✓
256 × 18	—	—	—	—	—	—	✓	✓	✓
128 × 36	—	—	—	—	—	—	✓	✓	✓

In simple dual-port mode, M4K blocks have one write-enable and one read-enable signal. On the M4K block, asserting the clear port of the `rden` register drives `rden` high, which allows the read operation to occur. When the read-enable signal is deactivated, the current data is retained at the output ports. If the read-enable signal is activated during a write operation with the same address location selected, the simple dual-port RAM output is either unknown or can be set to output the old data stored at the memory address.

For more information, see "Read-during-Write Operation at the Same Address" on page 7–20.

Figure 7–5 shows timing waveforms for read and write operations in simple dual-port mode.

Figure 7–5. Simple Dual-Port Timing Waveforms

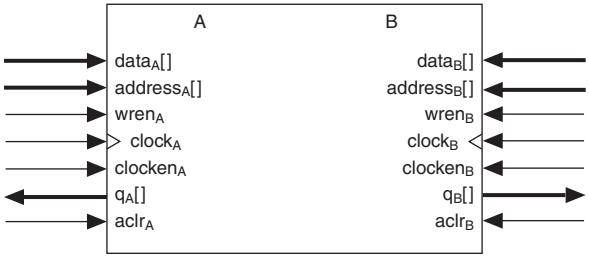


Implementing True Dual-Port Mode

M4K blocks offer a true dual-port mode to support any combination of two-port operations: two read operations, two write operations, or one read operation and one write operation at two different clock frequencies. True dual-port memory can be used to increase memory bandwidth in numerous applications. An example system that benefits from the use of true dual-port memory is a system containing an Altera Nios® embedded processor and a direct memory access (DMA) controller. Such a system will experience bottlenecks if the processor and the DMA controller need simultaneous access to single-port memory. The ability of both the processor and the DMA controller to access the M4K memory simultaneously, avoiding the need for arbitration, can dramatically improve bandwidth in this type of system.

Figure 7–6 shows the true dual-port memory configuration for M4K blocks.

Figure 7–6. True Dual-Port Memory *Note (1)*



Note to Figure 7–6:

(1) True dual-port memory supports input/output clock mode in addition to the independent clock mode shown.

The widest bit configuration of a single M4K block in true dual-port mode is 256 × 16-bit (or 256 × 18-bit with parity). The 128 × 32-bit (128 × 36-bit with parity) configuration of the M4K block is unavailable because the number of output drivers is equivalent to the maximum bit width of the M4K block. Because true dual-port RAM has outputs on two ports, the maximum width of the true dual-port RAM equals half of the total number of output drivers. However, multiple M4K blocks can be concatenated to support wider memory configurations. Table 7–5 lists the possible M4K RAM block configurations.

Table 7–5. M4K Block Mixed-Port Width Configurations (True Dual-Port Mode)

Port A	Port B						
	4K × 1	2K × 2	1K × 4	512 × 8	256 × 16	512 × 9	256 × 18
4K × 1	✓	✓	✓	✓	✓	—	—
2K × 2	✓	✓	✓	✓	✓	—	—
1K × 4	✓	✓	✓	✓	✓	—	—
512 × 8	✓	✓	✓	✓	✓	—	—
256 × 16	✓	✓	✓	✓	✓	—	—
512 × 9	—	—	—	—	—	✓	✓
256 × 18	—	—	—	—	—	✓	✓

In true dual-port mode, the RAM outputs can only be configured for read-during-write mode. This means that during write operation, data being written to the A or B port of the RAM flows through to the A or B outputs, respectively. When the output registers are bypassed, the new data is available on the rising edge of the same clock cycle it was written on.

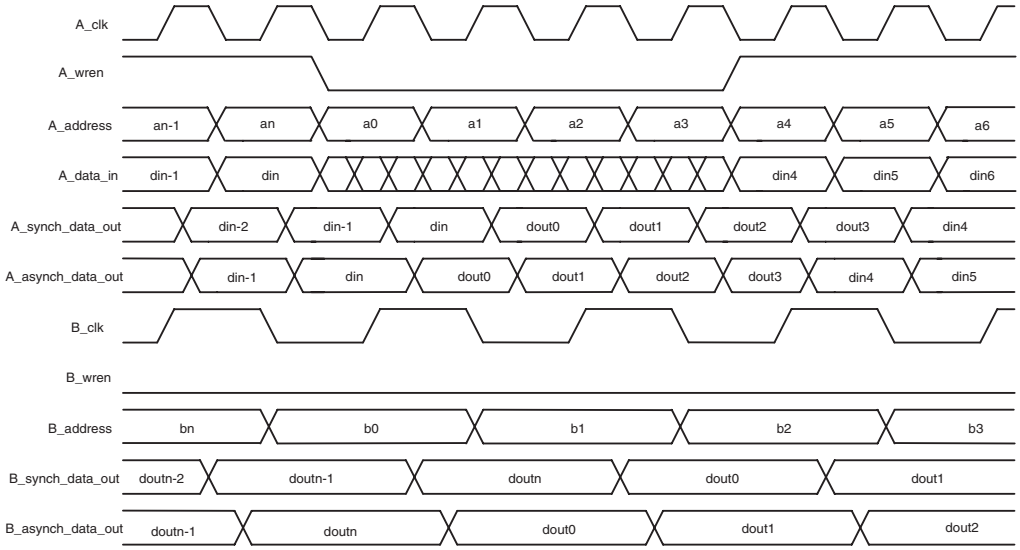
For sample waveforms and other information on mixed-port read-during-write mode, see "Read-during-Write Operation at the Same Address" on page 7–20.

Potential write conflicts must be resolved external to the RAM because simultaneously writing to the same address location at both ports results in unknown data storage at that location. For a valid write operation to the same address of the RAM block, the rising edge of the write clock for port A must occur following the minimum write cycle time interval after the rising edge of the write clock for port B. Since data is written into the M4K blocks at the falling edge of the write clock, the rising edge of the write clock for port A should occur following half of the minimum write cycle time interval after the falling edge of the write clock for port B. If this timing is not met, the data stored in that particular address will be invalid.

For more information about the minimum synchronous write cycle time, refer to the *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*.

Figure 7–7 shows true dual-port timing waveforms for a write operation at port A and a read operation at port B.

Figure 7–7. True Dual-Port Timing Waveforms



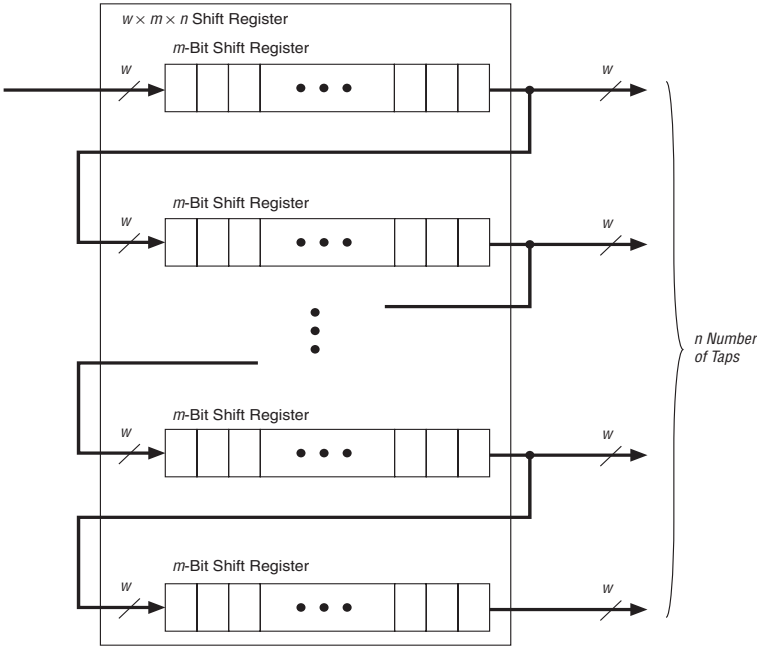
Implementing Shift-Register Mode

Embedded memory configurations can implement shift-register blocks for digital signal processing (DSP) applications, such as finite impulse response (FIR) filters, pseudo-random number generators, multi-channel filtering, and auto-correlation and cross-correlation functions. These and other DSP applications require local data storage, traditionally implemented with standard flip-flops that can quickly consume many logic cells for large shift registers. A more efficient alternative is to use embedded memory as a shift-register block, which saves logic cell and routing resources and provides a more efficient implementation.

The size of a (w × m × n) shift register is determined by the input data width (w), the length of the taps (m), and the number of taps (n). The size of a (w × m × n) shift register must be less than or equal to the 4,608 bits. In addition, the size of (w × n) must be less than or equal to 36 bits. If a larger shift register is required, memory blocks can be cascaded together.

Data is written into each address location at the falling edge of the clock and read from the address at the rising edge of the clock. The shift-register mode logic automatically controls the positive and negative edge clocking to shift the data in one clock cycle. Figure 7–8 shows the M4K memory block in shift-register mode.

Figure 7–8. M4K Shift-Register Memory Configuration



Implementing ROM Mode

M4K blocks can also be configured as ROM. ROM can be initialized in an M4K block by using a memory initialization file (**.mif**). Because all M4K memory configurations must have synchronous inputs, the address lines of the ROM are registered. ROM outputs can be registered or combinatorial. The read operation of the ROM is identical to the read operation of the single-port RAM configuration.

Implementing FIFO Buffers

FIFO buffer outputs are always combinatorial. Simultaneous read and write operations from an empty FIFO buffer are not supported.

Clock Modes

Depending on the M4K memory mode, independent, input/output, read/write, and/or single-port clock modes are available. Table 7–6 shows the clock modes supported by the M4K memory modes.

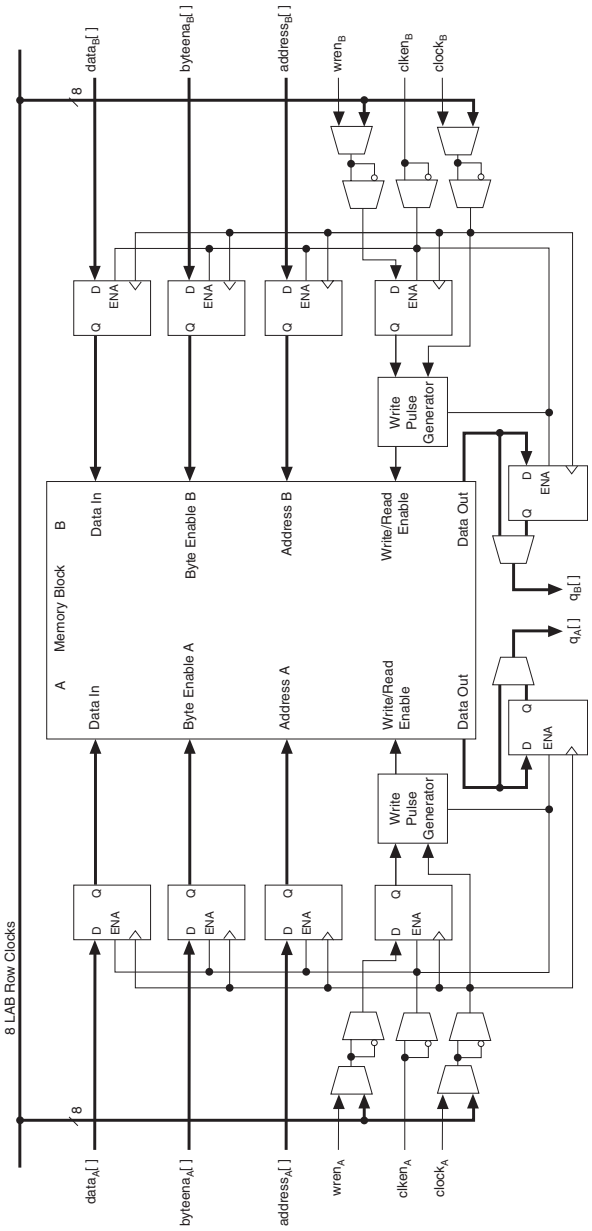
Table 7–6. M4K Memory Clock Modes

Clocking Mode	True-Dual Port Mode	Simple Dual-Port Mode	Single-Port Mode
Independent	✓	—	—
Input/output	✓	✓	—
Read/write	—	✓	—
Single-port	—	—	✓

Independent Clock Mode

M4K memory blocks can implement independent clock mode for true dual-port memory. In this mode, a separate clock is available for each port (A and B). Clock A controls all registers on the port A side, while clock B controls all registers on the port B side. Each port also supports independent clock-enable signals and asynchronous clear signals for port A and B registers. Figure 7–9 shows an M4K memory block in independent clock mode.

Figure 7–9. Independent Clock Mode *Note (1)*



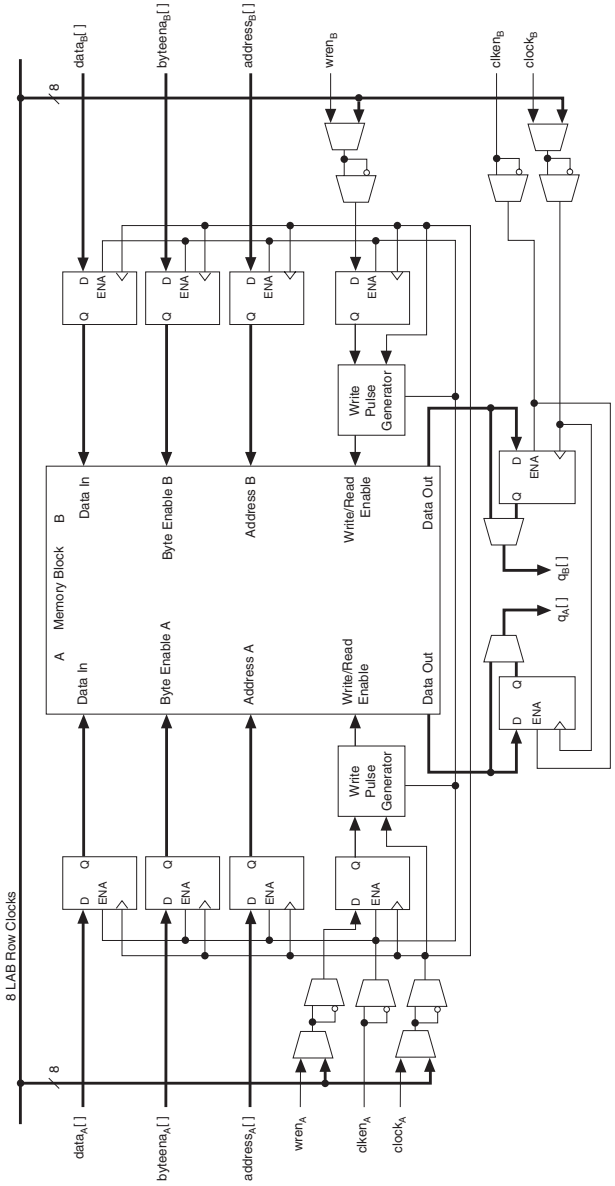
Note to Figure 7–9:

(1) Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Input/Output Clock Mode

M4K memory blocks can implement input/output clock mode for true and simple dual-port memory. On each of the two ports, A and B, one clock controls all registers for inputs (data input, `wren`, and `address`) into the memory block. The other clock controls the block's data output registers. Each memory block port also supports independent clock enables and asynchronous clear signals for input and output registers. Figures 7–10 and 7–11 show the memory block in input/output clock mode for true and simple dual-port modes, respectively.

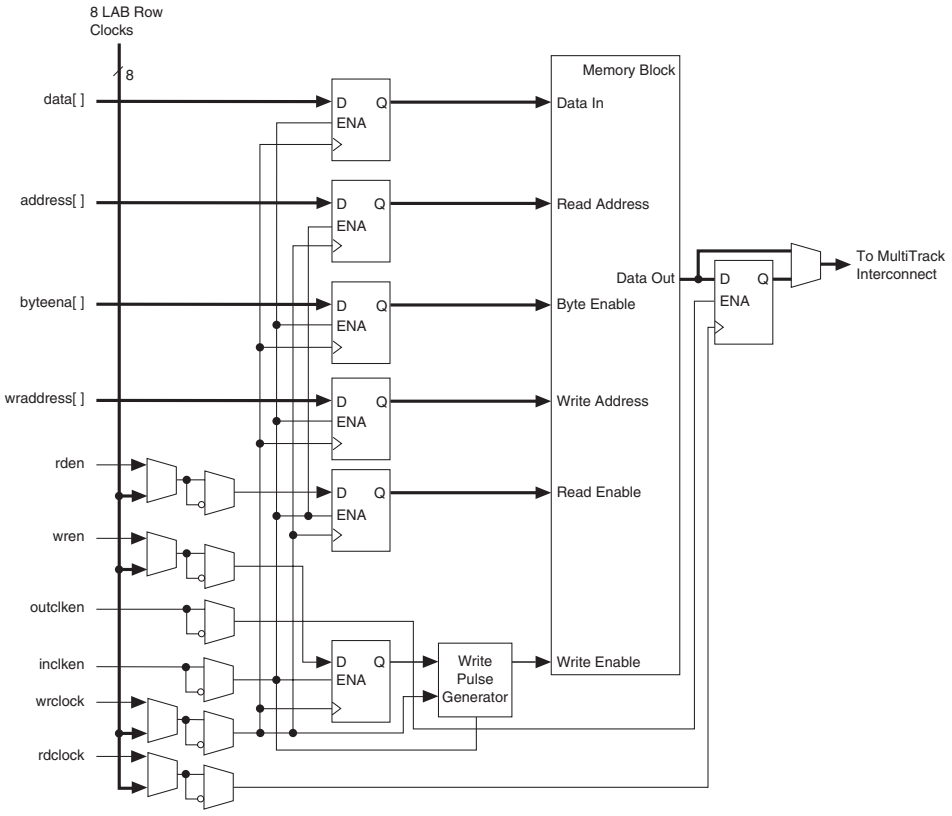
Figure 7–10. Input/Output Clock Mode in True Dual-Port Mode *Note (1)*



Note to Figure 7–10:

(1) Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Figure 7–11. Input/Output Clock Mode in Simple Dual-Port Mode *Notes (1), (2)*



Notes to Figure 7–11:

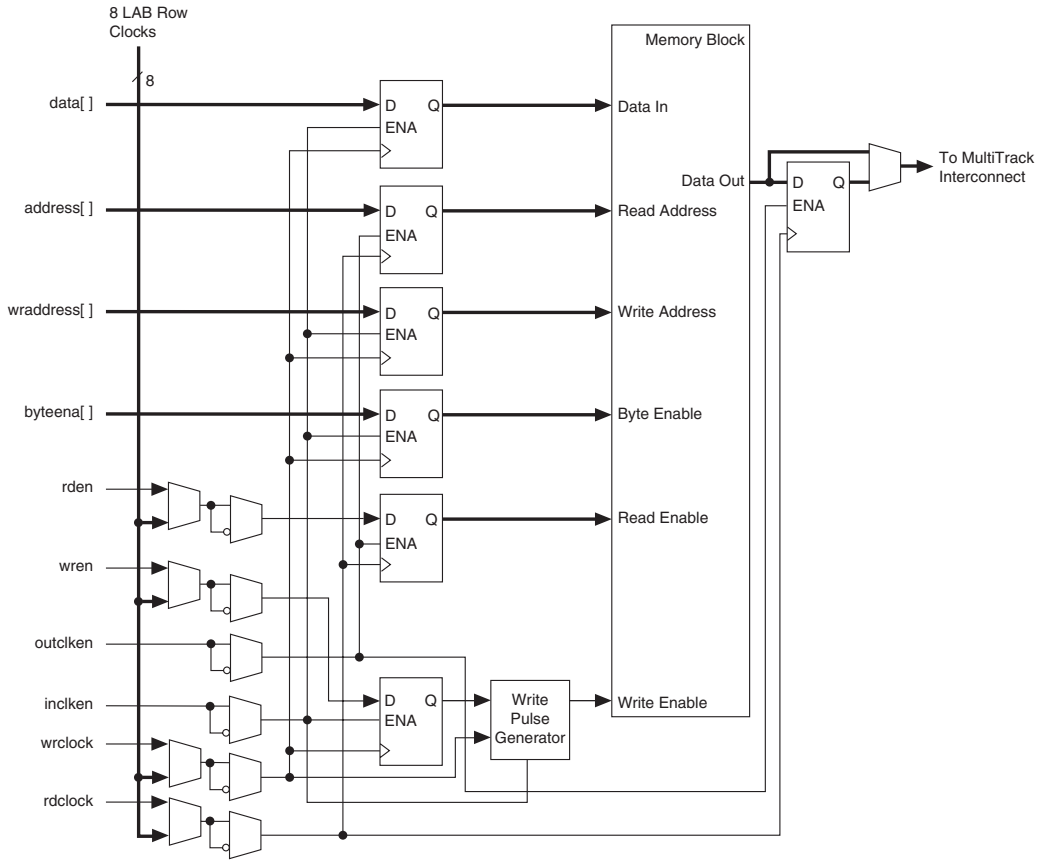
(1) For more information on the MultiTrack™ interconnect, refer to the *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*.

(2) Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Read/Write Clock Mode

M4K memory blocks can implement read/write clock mode for simple dual-port memory. This mode can use up to two clocks. The write clock controls the block's data inputs, `wraddress`, and `wren`. The read clock controls the data output, `rdaddress`, and `rden`. The memory blocks support independent clock enables for each clock and asynchronous clear signals for the read- and write-side registers. Figure 7–12 shows a memory block in read/write clock mode.

Figure 7–12. Read/Write Clock Mode in Simple Dual-Port Mode *Notes (1), (2)*



Notes to Figure 7–12:

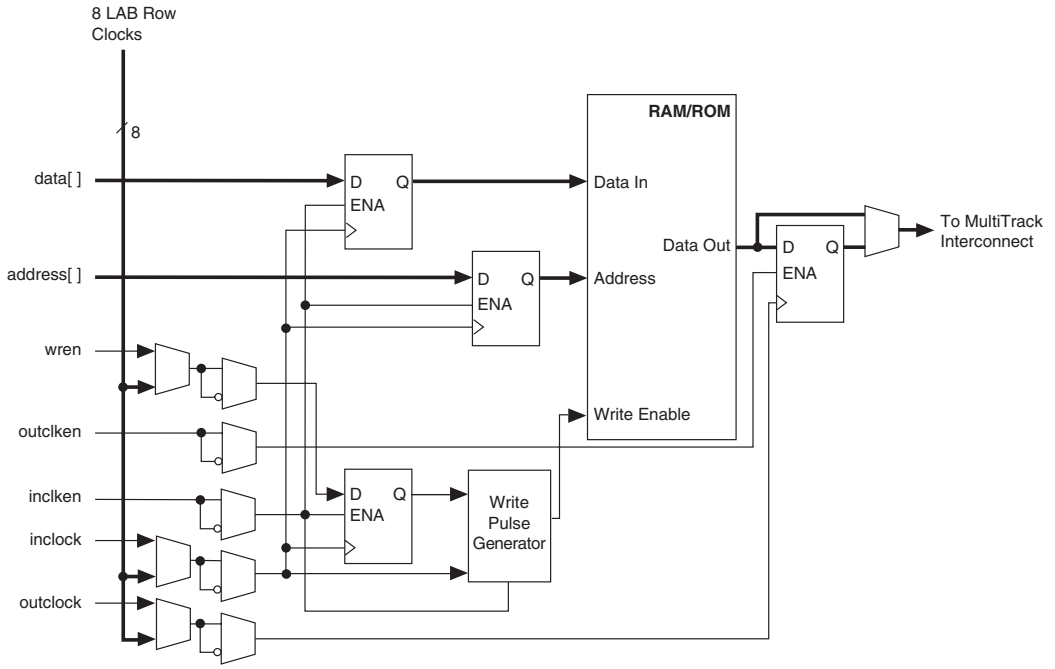
(1) For more information on the MultiTrack interconnect, refer to the *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*.

(2) Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Single-Port Mode

The M4K memory blocks can implement single-port clock mode when simultaneous read and write operations are not required (see Figure 7–13). A single block in a memory block can support up to two single-port mode RAM blocks in M4K blocks.

Figure 7–13. Single-Port Mode *Notes (1), (2)*



Notes to Figure 7–13:

(1) For more information about the MultiTrack interconnect, refer to the *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*.

(2) Violating the setup or hold time on the address registers could corrupt the memory contents. This applies to both read and write operations.

Synchronous and Pseudo-Asynchronous Modes

The M4K memory architecture implements synchronous, pipelined RAM by registering both the input and output signals to the RAM block. All M4K memory inputs are registered, providing synchronous write cycles. In synchronous operation, an M4K block generates its own self-timed strobe write enable (`wren`) signal derived from the global or regional clock. In contrast, a circuit using asynchronous RAM must generate the RAM `wren` signal while ensuring its data and address signals meet setup and hold time specifications relative to the `wren` signal. The output registers can be bypassed.

In an asynchronous memory, neither the input nor the output is registered. While Cyclone devices do not support asynchronous memory, they do support a pseudo-asynchronous read operation where the output data is available during the same clock cycle as when the read address is driven into it. Pseudo-asynchronous reading is possible in the simple and

true dual-port modes of the M4K blocks by clocking the read enable and read address registers on the negative clock edge and bypassing the output registers.

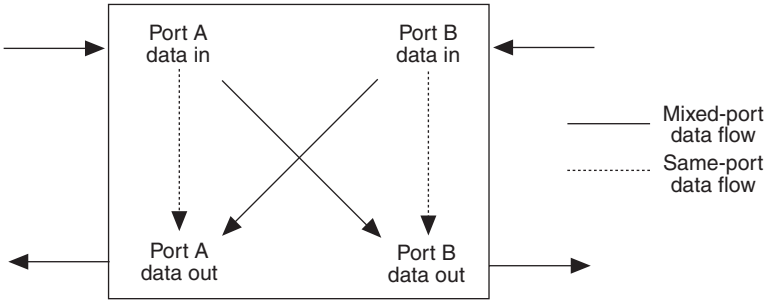
The clear signal for both asynchronous and synchronous mode for the memory are treated similarly in Cyclone devices. All inputs to the memory must be synchronous, therefore, the time it takes a clear signal to reset the input or output registers is synchronous to the clock.

For more information, refer to *AN 210: Converting Memory from Asynchronous to Synchronous for Stratix and Stratix GX Designs*.

Read-during-Write Operation at the Same Address

The following two sections describe the functionality of the various M4K memory configurations when reading from an address during a write operation at that same address. There are two types of read-during-write operations: same-port and mixed-port. Figure 7–14 illustrates the difference in data flow between same-port and mixed-port read-during-write.

Figure 7–14. Read-during-Write Data Flow

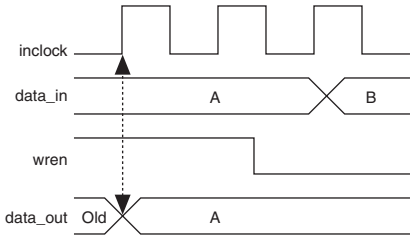


Same-Port Read-during-Write Mode

For read-during-write operation of a single-port RAM or the same port of a true dual-port RAM, the new data is available on the rising edge of the same clock cycle it was written on. See Figure 7–15 for a sample functional waveform.

When using byte-enable signals in true dual-port RAM mode, the outputs for the masked bytes on the same port are unknown. (See Figure 7–1.) The non-masked bytes are read out as shown in Figure 7–15.

Figure 7–15. Same-Port Read-during-Write Functionality *Note (1)*



Note to Figure 7–15:
(1) Outputs are not registered.

Mixed-Port Read-during-Write Mode

This mode is used when a RAM in simple or true dual-port mode has one port reading and the other port writing to the same address location with the same clock. You can configure the M4K memory block to operate in this mode and modify the parameter shown below using the MegaWizard® Plug-In Manager included with the Quartus II software.

The READ_DURING_WRITE_MODE_MIXED_PORTS parameter for M4K memory blocks determines whether or not to output the old data at the address. Setting this parameter to OLD_DATA outputs the old data at that address. Setting this parameter to DONT_CARE outputs an unknown value. During the instantiation of an ALTSYNCRAM or LPM_RAM_DP+ storage megafunction using the Quartus II software, the MegaWizard plug-in manager asks "How should the q output behave when reading a memory location that is being written from the other port?" Clicking "I don't care" assigns the DONT_CARE value to the parameter, and clicking "Old memory contents appear" assigns the OLD_DATA value to the parameter.

Altera recommends using the MegaWizard Plug-In Manager to create these memory megafunctions rather than directly creating instances. Once a storage megafunction is created using the MegaWizard Plug-In Manager, use the MegaWizard Plug-In Manager to make any necessary changes.

See Figures 7–16 and 7–17 for sample functional waveforms showing mixed-port read-during-write mode operation. These figures assume that the outputs are not registered.

Figure 7–16. Mixed-Port Read-during-Write: OLD_DATA

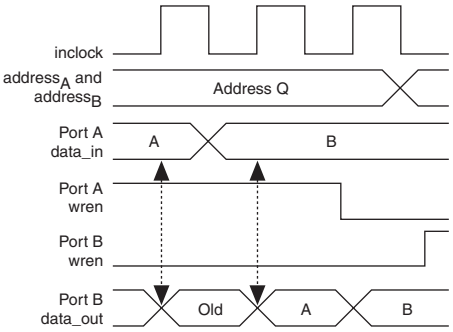
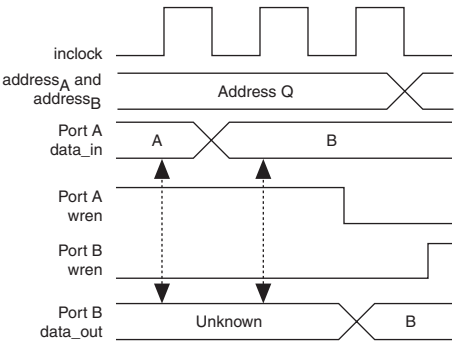


Figure 7–17. Mixed-Port Read-during-Write: DONT_CARE



Mixed-port read-during-write is not supported when two different clocks are used in a dual-port RAM. The output value is unknown during a mixed-port read-during-write operation.

For the minimum synchronous-write-cycle time, refer to the *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*.

Conclusion

M4K memory blocks are a flexible memory solution available in Cyclone devices that provide advanced features such as byte-enable capability, parity bit storage capability, and shift-register mode, as well as mixed-port width support and true dual-port mode. This flexibility makes these embedded memory blocks well suited for a wide range of applications including ATM cell packet processing, header/cell storage, channelized functions, and program memory for processors.

Referenced Documents

This chapter references the following documents:

- *AN 210: Converting Memory from Asynchronous to Synchronous for Stratix and Stratix GX Designs*
- *Cyclone FPGA Family Data Sheet* section of the *Cyclone Device Handbook*

Document Revision History

Table 7–7 shows the revision history for this chapter.

Table 7–7. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
May 2008 v1.4	Minor textual and style changes. Added "Referenced Documents" section.	—
January 2007 v1.3	Added document revision history.	—
August 2005 v1.2	Minor updates.	—
February 2005 v1.1	Updated notes for Figures 7-9 through 7-13.	—
May 2003 v1.0	Added document to Cyclone Device Handbook.	—