Stratix III RAM Feature Description

Version 1.0

March 24, 2009

by

Altera Corporation

Table of Contents:

1 Overview

This document is intended for users of the QUIP tools package working with Stratix III devices. It provides information about the Stratix III RAM block architectures to help with memory block usage.

2 Stratix III Memory Block Features

The following figure shows the high-level functional block diagram of a Stratix III memory block:

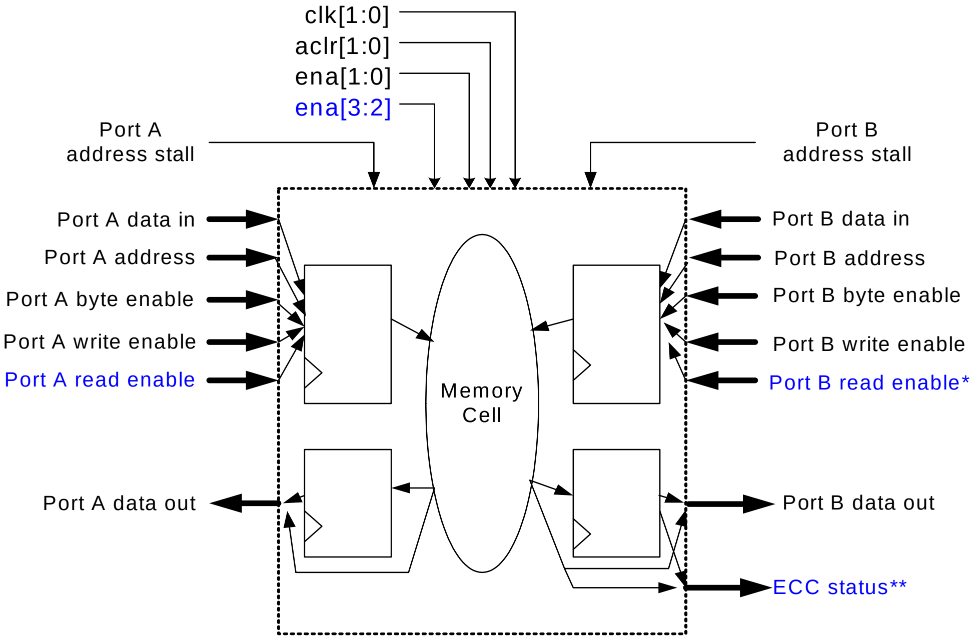


Figure 1: Stratix III RAM

* In Stratix II, port B read enable is shared with port B write-enable
** ECC feature is only available in M144K simple dual-port x64 mode

Similar to the Stratix II families, Stratix III retains the heterogeneous memory architecture. However, contrary to Stratix II, Stratix III only has two different sizes of hard memory blocks the M9K memory block and the M144K memory block. Both memory block types support the same set of features with the exception of ECC support which is limited to the M144K block. For the detailed features supported in each block, please refer to the following two tables (Table 1 and Table 2).

M512 blocks in Stratix II are replaced by the LUTRAM feature in Stratix III.

Operation Mode	**M9K**	**M144K**
Single-port	8Kx1 1Kx9 4Kx2 512x18 2Kx4 256x36[1]	16Kx9 2Kx72[1] 8Kx18 4Kx36
Simple dual-port **(dual_port)**	WxM/RxN or WxY/RxZ M,N is (1,2,4,8,16,32) Y,Z is (9, 18, 36)	WxM / RxN M is (9,18,36, 72) N is (9,18,36, 72)
True dual-port **(bidir_dual_port)**	AxM/BxN or AxY/BxZ M,N is (1,2,4,8,16)[2] Y,Z is (9, 18)[2]	AxM / BxN M is (9,18,36) N is (9,18,36)
ROM [3]	Same as single-port[4]	Same as single-port[4]

Table - Stratix III RAM Configurations

1. The widest single-port mode is actually supported by emulation through true dual-port mode. Quartus II will pack two atoms(each with half data width) in true dual-port mode to support the widest mode.
2. The M9K memory block is now fully symmetric in terms of mixed-width configurations in true dual-port mode unlike the MEAB in Stratix II.
3. Inputs to Stratix III RAM are always registered. The ROM mode really means a pipelined ROM.
4. The widest ROM mode will be implemented through simple dual-port mode as opposed to a packed mode implementation for single-port mode. This is mainly for power savings considerations since ROM only needs one read-port and we can save power by disabling the activities in the write-port.

	M9K	**M144K**
Physical Layout	256x36	2Kx72
Speed	600 MHz	600 MHz
Byte Enable Mask	Yes	Yes
Read-During-Write on Different Ports	Old data	Old data
Same-Port Read-During-Write	New Data,	New Data,

	Old Data, Enhanced BE Read	Old Data, Enhanced BE Read
Pre-initialized content	Yes	Yes
Packed Mode [1]	Yes	Yes
Address Stall	Yes	Yes
Parity Bit	Yes	Yes
Register Clear	Address Reg, Output Reg, Output latch	Address Reg, Output Reg, Output latch
Mixed data-width	Yes	Yes
Error Correction Code (ECC)	No	Yes

Table 2 Stratix III RAM Functionalities

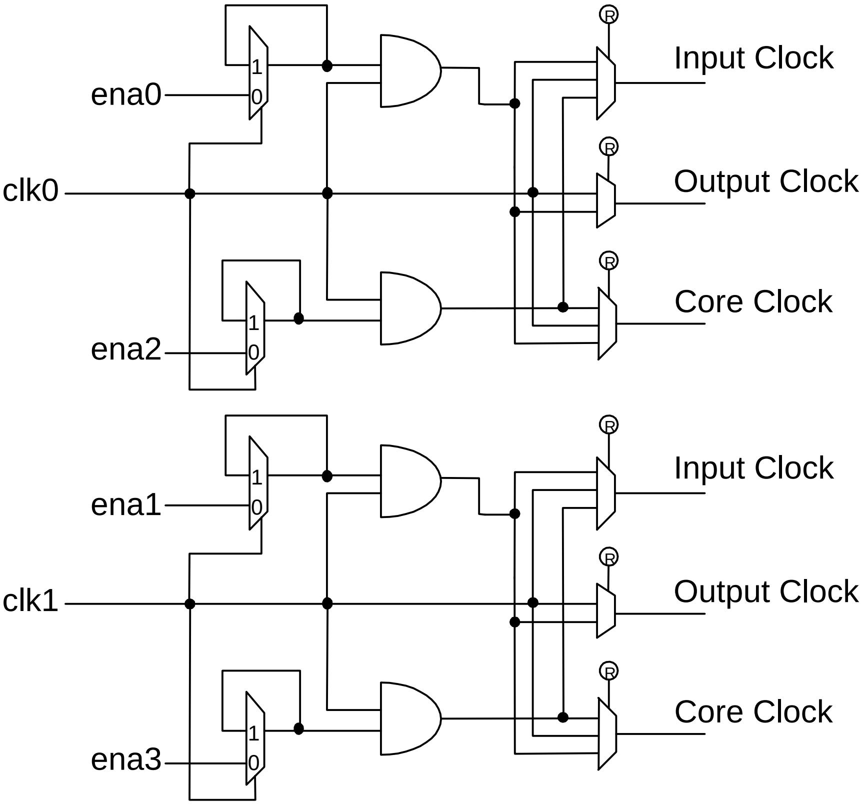
1. Packed mode is the feature to pack two independent single-port RAMs/ROMs into one block. It uses the true dual-port mode by a tie-off of the highest address bit to VCC for one RAM and GND for the other RAM. The size of each independent single-port RAM must not exceed the half of the target block size.

3 Stratix III vs. Stratix II

From the feature aspect, Stratix III is a superset of Stratix II. It supports all Stratix II features. In addition, the following enhancements are added to the Stratix III memory block.

3.1.1 Additional Clock Enables

An additional clock enable has been added to each clock to allow for a more flexible control of the RAM clocking. In Stratix III, each clock now has two clock-enable controls associated with it as opposed to one clock-enable in Stratix II. Therefore, the input registers, output registers, and core memory cells can have their own choice of using (i) one clock-enable, or (ii) the clock without a gating clock-enable. Note that the output registers only have the choice of using (i) clock-enable ena[1:0], or (ii) no clock-enable. They don’t have access to the additional clock-enables ena[3:2]. This new feature helps to reduce the power consumption of the memory block and provides more opportunities to absorb the output registers into the RAM block.

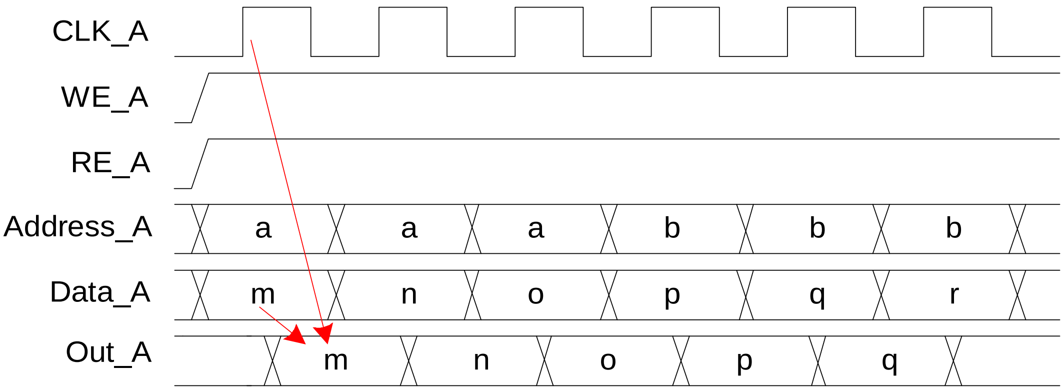


3.1.2 Additional Read-During-Write Modes

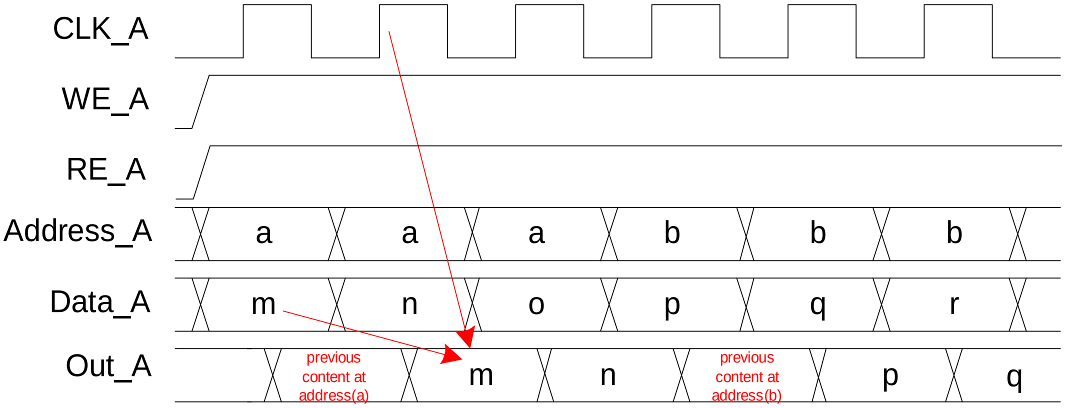
Stratix II only has one same-port read-during-write mode which is the new data mode (a.k.a. flow-through mode). It means that the data output of the read port can show the new written data during the write operation at the same clock cycle. In Stratix III, this feature has been enhanced to provide the old-data mode as well. Therefore, the RAM outputs can show the old data at the address before the write proceeds. This feature provides a better match to the behavior of the non-blocking assignment in HDL.

Mode	Data	Address	Output	Mem Content
New-data(flow-through)	DA	R	DA	Mem[R] = DA
Old-data	DA	R	Previous Mem[R]	Mem[R] = DA

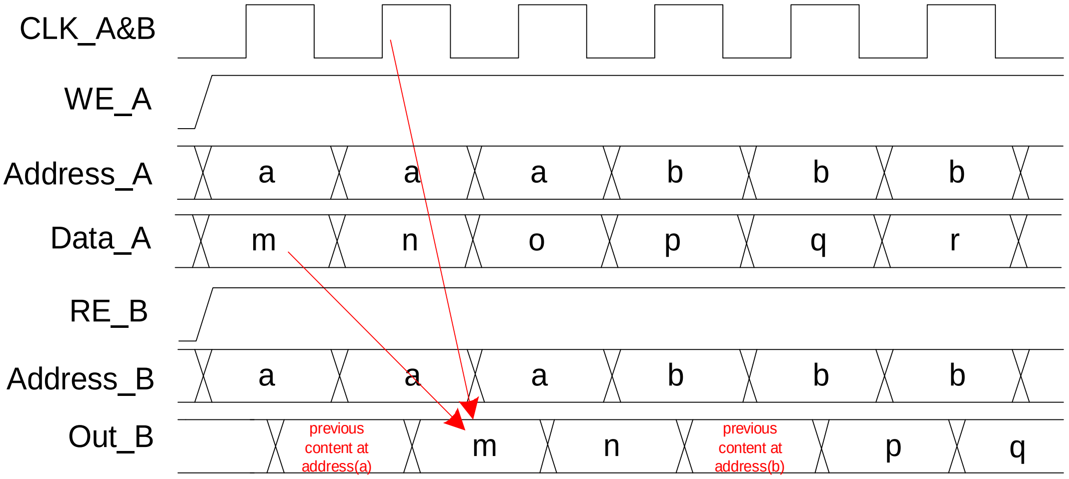
The mixed-port read-during-write mode remains the same as in Stratix II. The read-port can show the old-data when the write port is accessing the same location using the same clock.



Mode: Same-Port New Data with Unregistered Output


CLK_A
WE_A
RE_A
Address_A
a
a
a
b
b
b
Data_A
m
n
o
p
q
r
Out_A
previous content at address(a)
m
n
previous content at address(b)
p
q

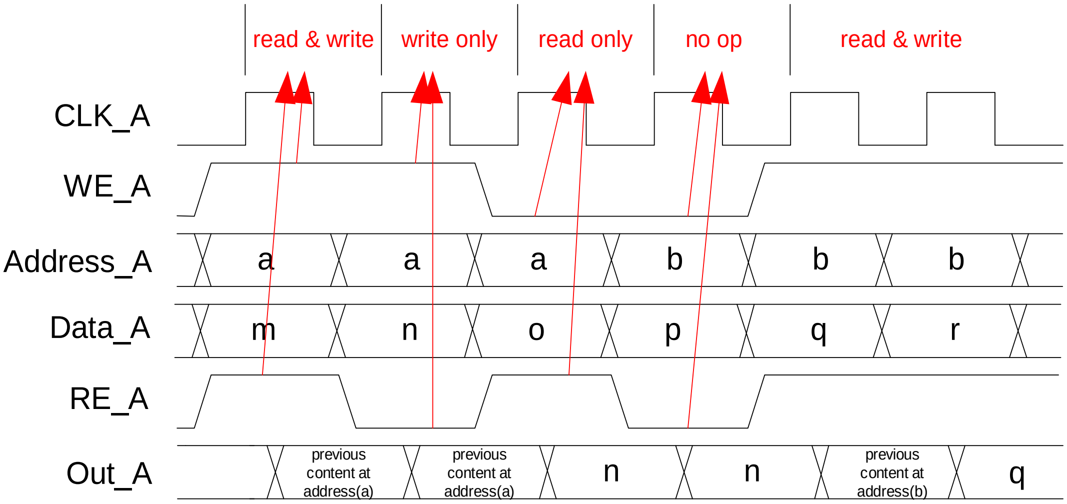
Mode: Same-Port Old Data with Unregistered Output



Mode: Mixed-Port Old Data with Unregistered Output

3.1.3 Independent Write Enable (WE) and Read Enable (RE) Controls

Stratix II only has one write enable port to control the write/read mode which has the side effect that read-during-write is always enabled. When the WE port is high, the memory block is in write mode. When the WE port is low, it is in read mode. The memory block in Stratix III has separate read-enable and write-enable controls. This is useful to reduce power if the user does not care about the data output during a write operation.



Mode: Same-Port Old Data with Unregistered Output

3.1.4 Read During Byte-enable Mask Write

The status of data outputs that have been masked out by a byte enable during a write operation is unknown in Stratix II. Indeed, they remain at the same state as they were during the previous operation. Therefore, they are modeled as unknown (X) for simplicity.

Stratix III has the additional option of reading out the memory content at the current write address location if the read-during-write mode is set to “New Data”. In this case, the resulting output is equivalent to what is actually written to the memory cell.

Assume Mem[R] = YZ before the write, and same-port read-during-write is set to “new data”

Mode	Data	BE	Address	Output	Mem Content
Read masked byte	AB	10	R	AZ	Mem[R] = AZ
No read masked byte	AB	10	R	AX	Mem[R] = AZ

3.1.5 ROM Mode for M144K

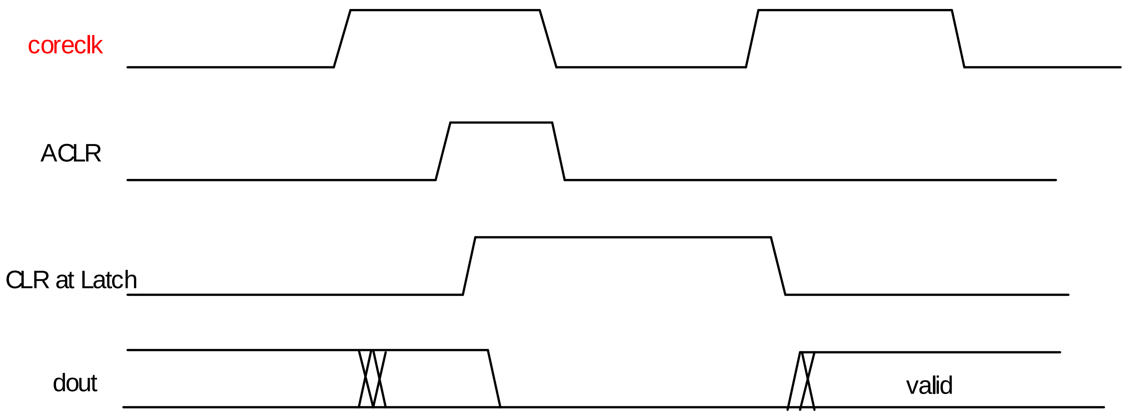
The MRAM in Stratix II does not support initialized content. The M144K block in Stratix III has been enhanced to allow user specified initial content.

3.1.6 Asynchronous Clears on Address Registers and Output Latch

The Stratix II memory blocks provide asynchronous clear signals only on output registers. The Stratix III memory blocks provide asynchronous clear controls (ACLR) to address registers and to the output latch as well. This implies that the output latch can have an asynchronous reset even if the output registers are not used. Note however that the output latch is based on the core clock (a variation of input clock) and not the output clock. Therefore, it will resume to pass data through on the next input clock edge once ACLR is released.

Note that performing an asynchronous clear on the address registers can cause data corruption of the memory array. This feature has been mainly added to the memory block to improve the FIFO application since the FIFO application does not care about the data after asynchronous reset.

Figure: ACLR Implementation for Output Latch and Output Register



Output Latch Asynchronous Clear Waveform

RAM Register Group	Clear Source	Clear Disable
Port A Address	A_CLR	Yes
Port B Address	A_CLR/B_CLR	Yes
Output Latch	A_CLR/B_CLR	Yes
Output Registers	A_CLR/B_CLR	Yes

Asynchronous Clear Option for RAM Registers

3.1.7 Rising-Edge Triggered Read/Write

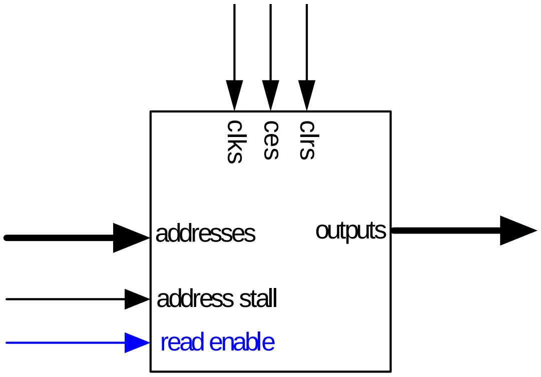
This change does not affect the functionality from a user's view point since the read-during-write behavior will define the RAM functionality that the user cares about. In Stratix III, both read and write operations are based on the clock's rising edge.

4 Operation Modes

Stratix III RAM blocks support the same four operation modes that Stratix II memory blocks supported. Those modes are ROM (read-only memory) mode, single-port RAM mode, simple dual-port RAM mode and true dual-port (bidir) RAM mode.

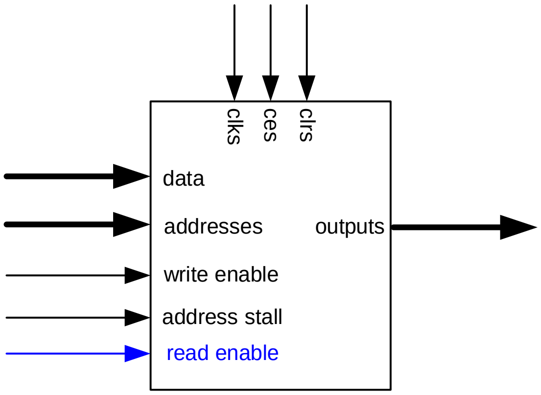
4.1 ROM (Read-Only Memory) Mode

ROM is a pre-initialized memory block that can only perform read operations. The following picture illustrates its I/O interfaces. Addresses are always registered by clk0 input. The outputs can be unregistered or registered. When the outputs are registered, they can be registered by clk0 or clk1. However, all outputs must be registered by the same clock. Only output registers can be cleared. And the clear source can be clr0 or clr1.



4.2 Single-Port RAM Mode

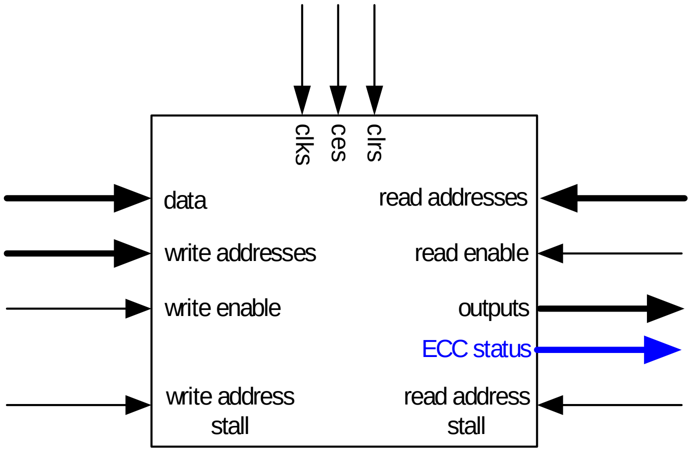
In a single-port RAM, only one location of the RAM can be accessed at a time. Read and write operations can be performed on the memory block. The following picture shows its I/O interfaces. Data, addresses and write-enable are always registered by clk0. The outputs can be unregistered or registered. Again, all outputs must be registered by the same clock (either clk0 or clk1). And only output registers can be cleared. The clear can come from clr0 or clr1.



4.3 Simple Dual-Port RAM Mode

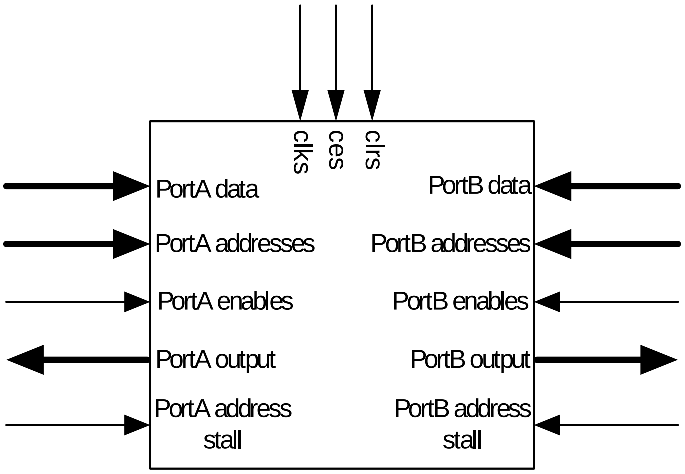
This mode is also known as dual-port mode. In simple dual-port mode, 1 read and 1 write operation (1R1W) can be performed on different locations at the same time. However, the RAM can't perform 2 reads or 2 writes at the same time. The following picture shows the I/O interfaces. The read port and the write port can be separately registered. The write-port must be registered by clk0, and the read-port can be registered by either clk0 or clk1. Only output registers can be cleared. The clear signal can come from either clr0 or clr1.

The Stratix III M144K memory block also supports the ECC feature in this mode. When the ECC feature is enabled, the byte-enable feature cannot be used.



4.4 True Dual-Port RAM Mode

This mode is also known as bidir dual-port. In true dual-port mode, 2W, 1R1W, or 2R operations can be performed at different locations at the same time. The following picture shows the I/O interfaces. Port A and Port B can be separately registered. Port A inputs (data, address and enable) must be registered by clk0. Port B inputs can be registered by either clk0 or clk1, but all port B inputs must be registered by the same clock. Both Port A and Port B outputs can be registered separately by either clk0 or clk1. They can be cleared by clr0 or clr1. Only output registers can be cleared.



5 RAM Block Inference

Similar to Stratix II, RAM blocks can be inferred from HDL source by synthesis tools. Since Stratix III supports all features of Stratix II, all modes inferred by Quartus II for Stratix II will also be inferred for Stratix III. Please see the corresponding Stratix II document for details.

The following sections will only discuss the new features that should be taken into consideration for Stratix III RAM block inference.

5.1 ROM

Synthesis should take advantage of mapping big ROM blocks to M144K memory blocks because they now support ROM mode.

5.2 More Read-During-Write Modes

In Stratix II, the RAM only supports the new data mode for same port read-during-write behavior. Stratix III can support both old-data and new-data mode. This provides more opportunities for RAM inference when non blocking assignments are used in HDL.

5.2.1 New-Data Mode

The following example shows the sample code that demonstrates the new-data read-during-write mode. The output will show the newly written data in the same clock cycle.

```
// Single-Port RAM with unregistered output
// Read-during-write mode is new_data
module single (clk, din, addr, wren, dout);

input clk, wren;
input [3:0] addr;
input [7:0] din;
output [7:0] dout;

reg [7:0] dout;
reg [7:0] mem[15:0];

always @ (posedge clk) begin
  if (wren)
    mem[addr] = din;    // Blocking assignment. The output will get
  dout = mem[addr];     // the written data in the same clock cycle.
end

endmodule
```

5.2.2 Old-Data Mode

The following example shows the sample code that demonstrates the old-data read-during-write mode. The output will show the old data at the write location in the same clock cycle. This is not available in Stratix II. Stratix II can implement the old-data read-during-write behavior only in dual-port RAM mode.

```
// Single-Port RAM with unregistered output
// Read-during-write mode is old_data
module single (clk, din, addr, wren, dout);

input clk, wren;
input [3:0] addr;
input [7:0] din;
output [7:0] dout;

reg [7:0] dout;
reg [7:0] mem[15:0];

always @ (posedge clk) begin
  if (wren)
    mem[addr] <= din;         // Note: This is non-blocking assignment.
                              // The output will get the previous data
  dout <= mem[addr];          // before the write happens.
end

endmodule
```

5.3 Additional Clock-Enables

Stratix II RAM inference fails if the HDL uses more than two clock-enables due to the limited availability of clock-enable ports in Stratix II. However, the synthesis module is able to take advantage of the extra flexibility in the clock-enable feature and infers RAM blocks even if there are more than two clock-enables. Below is an RTL code example that can be mapped to a dual-port RAM in Stratix III by using the additional clock-enable. As for Stratix II, the following example will be mapped to a RAM block without output registers since the output clock-enable can't be implemented in RAM and thus an extra register is needed in the core to implement the output register.

```
// Dual-port RAM with registered output
// Separate read/write clocks, separate read/write addresses
// With write clock-enable and separate input/output read port clock-
enable
```

```
// wclk -> clk0, wclk_ena -> ena0, rclk -> clk1, rdclk_out_ena->ena1,
rdclk_in_ena->ena3
// port_a_input_clock_enable = ena0, port_b_input_clock_enable = ena3,
// port_b_core_clock_enable = ena3, port_b_output_clock_enable = ena1
module dual(rclk, wclk, wclk_ena, rdclk_in_ena, rdclk_out_ena, din,
raddr, waddr, wren, dout);


input rclk, wclk, wren;
input rdclk_in_ena, rdclk_out_ena, wclk_ena;
input [3:0] raddr;
input [3:0] waddr;
input [7:0] din;
output [7:0] dout;


reg [7:0] tmpdout;
reg [3:0] reg_raddr ;
reg [3:0] reg_waddr ;
reg [7:0] dout;
reg [7:0] mem[15:0];


always @ (posedge wclk) begin
  if (wclk_ena) reg_waddr = waddr;
  if (wren && wclk_ena)
    mem[reg_waddr] <= din;
end


always @ (posedge rclk) begin
  if (rdclk_in_ena) begin
     reg_raddr = raddr;
     tmpdout <= mem[reg_raddr];
  end


  if (rdclk_out_ena)  dout <= tmpdout;
end


endmodule
```

5.4 Independent/Separate Read/Write Controls

Separate read/write controls have to be mapped to simple dual-port RAM blocks for Stratix and Stratix II. In Stratix III, separate read/write controls are provided on the same port. Therefore, a single-port RAM block can be used instead if there is only one address provided. This saves power since the other port can be shut-off.

```
// Single-Port RAM with unregistered output
// And separate read/write controls
// Read-during-write mode is new data
module single (clk, din, addr, wren, rden, dout);

input clk, wren, rden;
input [3:0] addr;
input [7:0] din;
output [7:0] dout;

reg [7:0] dout;
reg [7:0] mem[15:0];

always @ (posedge clk) begin
  if (wren)
    mem[addr] = din;          // Note: This is the blocking assignment.
                              // The output will get the written data
  if (rden) dout = mem[addr]; // in the same clock cycle.
end

endmodule
```