ROM Megafunction

User Guide

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com

Quartus II Version:	7.1
Document Version:	1.0
Document Date:	May 2007

UG-01016-1.0

Contents

About this User Guide

Revision History ... v
Referenced Documents ... v
How to Contact Altera ... vi
Typographic Conventions ... vi

Chapter 1. About these Megafunctions

Device Family Support ... 1–1
Introduction ... 1–1
Features of ROM: 1-PORT MegaWizard Plug-In ... 1–2
General Description of ROM: 1-PORT MegaWizard Plug-In ... 1–2
Resource Utilization and Performance of Single-Port ROM ... 1–4
Features of ROM: 2-PORT MegaWizard Plug-In ... 1–4
General Description of ROM: 2-PORT MegaWizard Plug-In ... 1–4
Resource Utilization and Performance of Dual-Port ROM ... 1–6

Chapter 2. Getting Started

Software and System Requirements ... 2–1
MegaWizard Plug-In Manager Customization ... 2–1
Using the MegaWizard Plug-In Manager ... 2–1
The ROM: 1-PORT MegaWizard Plug-In Page Descriptions ... 2–3
The ROM: 2-PORT MegaWizard Plug-In Page Descriptions ... 2–10
Inferring Megafunctions from HDL Code ... 2–18
Instantiating Megafunctions in HDL Code or Schematic Designs ... 2–19
Generating a Netlist for EDA Tool Use ... 2–19
Using the Port and Parameter Definitions ... 2–20
Identifying a Megafunction After Compilation ... 2–20
Simulation ... 2–20
Quartus II Software Simulation ... 2–20
EDA Simulation ... 2–21
SignalTap II Embedded Logic Analyzer ... 2–21
In-System Updating of Memory and Constants ... 2–22
Design Examples for the ROM Megafunctions ... 2–22
Design Files ... 2–22
Example for ROM: 1-PORT ... 2–22
Generate the Single-Port ROM ... 2–23
Implement Single-Port ROM ... 2–26
Functional Results: Simulate the Single-Port ROM in the Quartus II Software ... 2–28
Understanding the Simulation Results ... 2–30
Functional Results: Simulate the Single-Port ROM in ModelSim-Altera ... 2–32

Example for ROM: 2-PORT 2–33
Generate the Dual-Port ROM 2–33
Implement Dual-Port ROM 2–36
Functional Results: Simulate the Dual-port ROM in the Quartus II Software 2–36
Understanding the Simulation Results 2–38
Functional Results: Simulate the Single-Port ROM in ModelSim-Altera 2–40
Conclusion 2–42

Chapter 3. Specifications

Introduction 3–1
Ports and Parameters for the altsyncram Megafunction 3–1

About this User Guide

Revision History

The following table shows the revision history for this user guide.

Date and Document Version	Changes Made	Summary of Changes
May 2007 v1.0	Initial release of document. Released with Quartus II v7.1	Initial release.

Referenced Documents

This user guide references the following documents:

- *TriMatrix Embedded Memory Blocks in Stratix III Devices* chapter of the *Stratix III Devices Handbook*
- *In-System Updating of Memory and Constants* chapter in volume 3 of the *Quartus II Handbook*
- *Recommended HDL Coding Styles* chapter in volume 1 of the *Quartus II Handbook*
- *Synthesis* section in volume 1 of the *Quartus II Handbook*
- *Quartus II Integrated Synthesis* chapter in volume 1 of the *Quartus II Handbook*
- *Simulation* section in volume 3 of the *Quartus II Handbook*
- *Design Debugging Using the SignalTap II Embedded Logic Analyzer* chapter in volume 3 of the *Quartus II Handbook*

How to Contact Altera

For the most up-to-date information about Altera® products, refer to the following table.

Contact (1)	Contact Method	Address
Technical support	Website	www.altera.com/mysupport/
Technical training	Website	www.altera.com/training/
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Altera literature services	Email	literature@altera.com
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note to table:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

This document uses the typographic conventions shown in the following table.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, filenames, filename extensions, and software utility names are shown in bold type. Examples: $\mathbf{f_{MAX}}$, **\qdesigns** directory, **d:** drive, **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75: High-Speed Board Design.*
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: <*file name*>, <*project name*>**.pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
"Subheading Title"	References to sections within a document and titles of on-line help topics are shown in quotation marks. Example: "Typographic Conventions."

Visual Cue	Meaning
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input.` Active-low signals are denoted by suffix `n`, e.g., `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword `SUBDESIGN`), as well as logic function names (e.g., `TRI`) are shown in Courier.
1., 2., 3., and a., b., c., etc.	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ● •	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
☞	The hand points to information that requires special attention.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or the user's work.
⚠ WARNING	A warning calls attention to a condition or possible situation that can cause injury to the user.
↵	The angled arrow indicates you should press the Enter key.
👣	The feet direct you to more information about a particular topic.

Chapter 1. About these Megafunctions

Device Family Support

The ROM megafunctions support the following target Altera® device families:

- Arria™ GX
- Stratix® III
- Stratix II
- Stratix II GX
- Stratix
- Stratix GX
- Cyclone® III
- Cyclone II
- Cyclone
- HardCopy® II
- HardCopy Stratix
- ACEX 1K® (single-port ROM only)
- APEX™ II
- APEX 20KC (single-port ROM only)
- APEX 20KE (single-port ROM only)
- FLEX 10K® (single-port ROM only)
- FLEX® 10KA (single-port ROM only)
- FLEX 10KE (single-port ROM only)

The dual-port ROM megafunction does not support ACEX 1K®, APEX™ 20KC, APEX 20KE, FLEX 10K®, FLEX® 10KA, or FLEX 10KE devices.

Introduction

As design complexities increase, the use of vendor-specific intellectual property (IP) blocks has become a common design methodology. Altera provides parameterizable megafunctions that are optimized for Altera device architectures. Using megafunctions instead of coding your own logic saves valuable design time. The Altera-provided functions offer more efficient logic synthesis and device implementation. You can scale the size of the megafunction by setting various parameters.

A ROM allows read operations that read predetermined data from specific memory addresses, but does not support write operations. The Quartus II software provides two ROM MegaWizard® plug-ins, accessed with the MegaWizard Plug-In Manager, to support single-port and dual-port ROM functionalities:

- ROM: 1-PORT
- ROM: 2-PORT

These plug-ins are user-view wizards and not the actual megafunctions. The Quartus II software uses the altsyncram megafunction to implement the ROM functionality you configure through the MegaWizard Plug-In Manager. This chapter describes the features and resource usage of the ROM MegaWizard plug-ins.

Features of ROM: 1-PORT MegaWizard Plug-In

The ROM: 1-PORT MegaWizard plug-in implements a single-port ROM function and offers many additional features, which include:

- Configurable memory block type
- Single-clock and dual-clock modes
- Clock enable, asynchronous clear, and address-clock enable features
- Read-enable support for Stratix III and Cyclone III devices

General Description of ROM: 1-PORT MegaWizard Plug-In

The ROM: 1-PORT MegaWizard plug-in is an easy-to-use GUI for configuring a single-port ROM.

It provides different memory block types for selection depending on the device you select. Refer to "Resource Utilization and Performance of Single-Port ROM" on page 1–4 for more details.

The ROM: 1-PORT MegaWizard plug-in allows you to specify either of two clocking modes: a single-clock or a dual-clock (input/output) mode. In single-clock mode, input and output signals are synchronous with the same clock. In the Stratix and Cyclone series of devices, a single clock with a clock enable controls all registers of the memory block. Dual-clock (input/output) mode operates with two independent clocks: `inclock` and `outclock`. The input clock controls all input registers, such as the address and read-enable registers, while the output clock controls the data output register. In the Stratix and Cyclone series of devices, you can select either single-clock or dual-clock (input/output) mode for your single-port ROM. In other devices, only dual-clock mode is available.

The single-port ROM supports a clock enable feature in the Stratix and Cyclone series of devices. You can create a clock-enable signal for each clock signal. All registered ports are controlled by this signal by default, but you can use a clock-enable bypass option (through the MegaWizard) to bypass the clock enable feature for input or output registers.

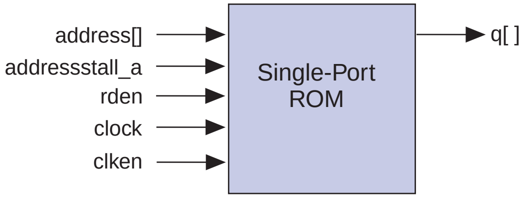
Asynchronous clear is another feature supported in the Stratix and Cyclone series of devices. When asynchronous clear is applied to an input register, the register clears immediately. However, the output of the memory block does not show the effects until the next rising edge of the clock. When asynchronous clear is applied to an output register, the register clears and the effects are immediately visible outside the memory block. Stratix III and Cyclone III devices have an additional feature in which asynchronous clear signals are available on output latches and output registers. This feature allows you to clear the ROM outputs via the output latch asynchronous clear even when your ROM is not using the output registers.

The single-port ROM features address clock enable support, which holds the previous address value for as long as the signal is enabled (addressstall = 1). By default, the address clock enable signal is turned off. This feature is supported in the Arria GX, Stratix III, Stratix II, Stratix II GX, Cyclone III, Cylone II, and HardCopy II device families only.

In addition to these ports, Stratix III and Cyclone III devices support the read enable port (`rden`) for single-port ROM. The only exception is that Stratix III devices do not support this port for MLABs.

Figure 1–1 shows the ports available on a typical single-port ROM in a Stratix III device.

Figure 1–1. Typical Single-Port ROM Block Diagram for Stratix III Devices



For more information about the features of ROM supported in Stratix III devices, refer to the *TriMatrix Embedded Memory Blocks in Stratix III Devices* chapter of the *Stratix III Device Handbook.*

Resource Utilization and Performance of Single-Port ROM

The ROM: 1-PORT MegaWizard plug-in uses the altsyncram megafunction to implement single-port ROM in the Stratix and Cyclone series of devices, and uses the lpm_rom megafunction to implement it in earlier device families. The single-port ROM uses the following device resources:

- Embedded System Blocks (ESB) in APEX II and APEX 20KC devices
- Embedded Array Blocks (EAB) in ACEX 1K, FLEX 10K, FLEX 10KA, and FLEX 10KE devices
- M4K in HardCopy II, Cyclone, and Cyclone II devices
- M9K in Cyclone III devices
- M512 or M4K in the Stratix series of devices, except Stratix III devices
- MLAB, M9K, or M144K in Stratix III devices

Features of ROM: 2-PORT MegaWizard Plug-In

The ROM: 2-PORT MegaWizard plug-in implements a dual-port ROM function and offers many additional features, which include:

- Configurable memory block type
- Different data width support for the output ports
- Providing different clock modes of operations
- Clock enable, asynchronous clear, and address clock enable features

General Description of ROM: 2-PORT MegaWizard Plug-In

The ROM: 2-PORT MegaWizard plug-in is an easy-to-use GUI for configuring a dual-port ROM.

It provides different memory block types for selection depending on the device you select. Refer to "Resource Utilization and Performance of Dual-Port ROM" on page 1–6 for more details.

The dual-port ROM also supports different data width for the output ports. This feature is only supported in the Stratix and Cyclone series of devices. The ROM: 2-PORT MegaWizard plug-in allows you to configure the two ports of the dual-port ROM to have different output data widths.

Through the ROM: 2-PORT MegaWizard plug-in, you can also configure different clock modes for your dual-port ROM. The following clock modes are available: single clock mode, independent clock mode, input/output clock mode, and asynchronous mode. The availability of the clocking modes depends on the device you select.

In single clock mode, the input and output signals are synchronous with the same clock. In independent clock mode, a separate clock is available for each port (A and B). Clock A controls all registers on the port A side, and clock B controls all registers on the port B side. In input/output clock mode, an input clock controls all input address registers, and the output clock controls data output registers. The input/output clocking mode is only supported in the Stratix and Cyclone series of devices. In asynchronous mode, no clock is required. Asynchronous mode is only supported by APEX II devices.

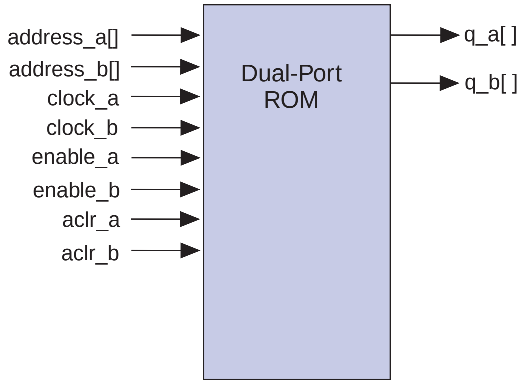
The dual-port ROM supports a clock-enable feature. You can create a clock enable signal for each clock signal and all registered ports are controlled by the enable signal. For the Stratix and Cyclone series of devices (except Stratix, Stratix GX, HardCopy Stratix, and Cyclone devices), you can use the clock enable bypass options (through the MegaWizard) to bypass the clock enable feature for either input register or output register.

Asynchronous clear is another feature supported in the Stratix and Cyclone series of devices. When asynchronous clear is applied to an output register, the register clears and the effects are immediately visible outside the memory block. Stratix III and Cyclone III devices have an additional feature in which asynchronous clear signals are available on output latches and output registers. This feature allows you to clear the ROM outputs via the output latch asynchronous clear even when your ROM is not using the output registers.

The dual-port ROM features address-clock enable support, which holds the previous address value for as long as the signal is enabled (addressstall = 1). By default, the address clock enable signal is turned off. This feature is only supported in the Stratix and Cyclone series of devices (except Stratix, Stratix GX, HardCopy Stratix, and Cyclone devices).

Figure 1–2 shows the ports available on a typical dual-port ROM in a Stratix III device.

Figure 1–2. Typical Dual-Port ROM Block Diagram for Stratix III Devices



Note to Figure 1–2:

(1) This figure shows only the common input ports for a typical dual-port ROM for Stratix III devices. Refer to "Ports and Parameters for the altsyncram Megafunction" on page 3–1 for all the input and output ports.

Resource Utilization and Performance of Dual-Port ROM

The ROM: 2-PORT MegaWizard plug-in uses the altsyncram megafunction to implement dual-port ROM in the Stratix and Cyclone series of devices. The dual-port ROM uses the following device resources:

- ESB in APEX II devices
- M4K in the Stratix and Cyclone series of devices except Stratix III and Cyclone III devices
- M9K in Cyclone III devices
- M9K or M144K in Stratix III devices

Chapter 2. Getting Started

Software and System Requirements

The instructions in this section require the following software:

- For operating system support information, refer to:

  //www.altera.com/support/software/os_support/oss-index.html

- Quartus® II software version 7.1 or higher

MegaWizard Plug-In Manager Customization

The MegaWizard Plug-In Manager creates or modifies design files that contain custom megafunction variations which can then be instantiated in a design file. The MegaWizard Plug-In Manager provides a wizard that allows you to specify options for single-port ROM and dual-port ROM, depending on the ROM MegaWizard plug-in you select.

You can use the wizard to set the features of the ROM megafunctions in the design. Start the MegaWizard Plug-In Manager using one of the following methods:

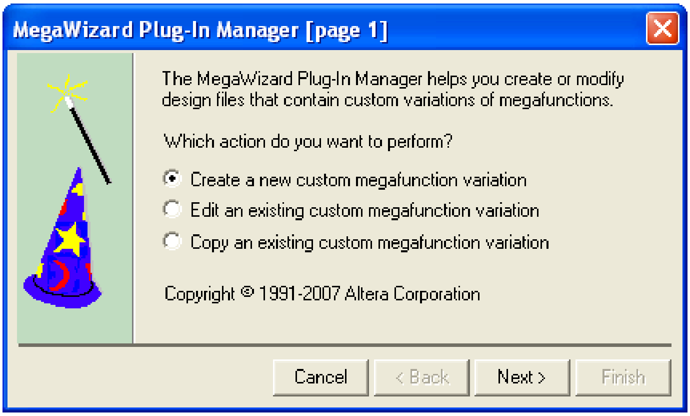
- On the Tools menu, click **MegaWizard Plug-In Manager**.
- When working in the Block Editor, in the Symbol window, click **MegaWizard Plug-In Manager**.
- Start the stand-alone version of the MegaWizard Plug-In Manager by typing the following command at the command prompt: `qmegawiz`↵

Using the MegaWizard Plug-In Manager

This section provides descriptions of the options available on the individual pages of the ROM: 1-PORT and ROM: 2-PORT MegaWizard plug-ins.

On page 1 of the MegaWizard Plug-In Manager, select **Create a new custom megafunction variation, Edit an existing custom megafunction variation,** or **Copy an existing custom megafunction variation**. This page identifies what you can do using the wizard. Figure 2–1 shows page 1 of a MegaWizard Plug-In Manager.

Figure 2–1. MegaWizard Plug-In Manager [page 1]



MegaWizard Plug-In Manager [page 1]

The MegaWizard Plug-In Manager helps you create or modify design files that contain custom variations of megafunctions.

Which action do you want to perform?

● Create a new custom megafunction variation

○ Edit an existing custom megafunction variation

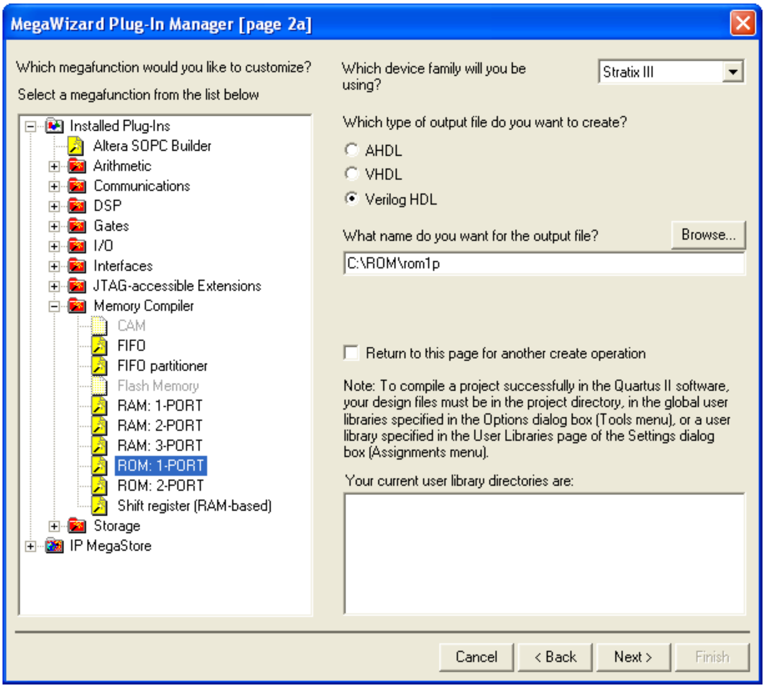
○ Copy an existing custom megafunction variation

Copyright © 1991-2007 Altera Corporation

Page 2a is where you specify which MegaWizard you want to use. The ROM: 1-PORT and ROM: 2-PORT MegaWizard plug-ins are located under **Memory Compiler** in the megafunctions list.

You can also specify the device family to use, type of output file to create, and the name of the output file from this page. You can choose AHDL (**.tdf**), VHDL (**.vhd**), or Verilog HDL (**.v**) as the output file type. Figure 2–2 shows page 2a of a MegaWizard Plug-In Manager.

Figure 2–2. MegaWizard Plug-In Manager [page 2a]



MegaWizard Plug-In Manager [page 2a]

Which megafunction would you like to customize?

Select a megafunction from the list below

Which device family will you be using? Stratix III

Which type of output file do you want to create?

○ AHDL
○ VHDL
● Verilog HDL

What name do you want for the output file? C:\ROM\rom1p

□ Return to this page for another create operation

Note: To compile a project successfully in the Quartus II software, your design files must be in the project directory, in the global user libraries specified in the Options dialog box (Tools menu), or a user library specified in the User Libraries page of the Settings dialog box (Assignments menu).

Your current user library directories are:

The ROM: 1-PORT MegaWizard Plug-In Page Descriptions

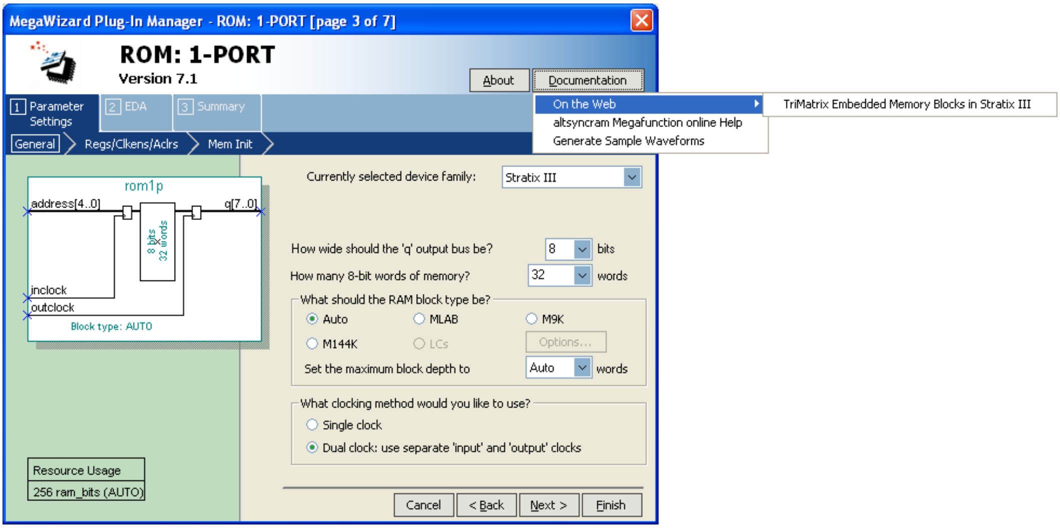
On page 2a of the MegaWizard Plug-In Manager, under **Memory Compiler**, you can select the **ROM: 1-PORT** MegaWizard plug-in.

Page 3 of the ROM: 1-PORT MegaWizard plug-in is where you specify the device family, set the width of the data output bus, set the number of words of memory, select the memory block type, and select the clock mode.

☞ Beginning with page 3, from the **Documentation** button, you can select the **On the Web** option to view documentation about TriMatrix Embedded Memory Blocks in Stratix III, or select the **Generate Sample Waveforms** option to view a sample waveform based on your configuration settings.

Figure 2–3 shows page 3 of the ROM: 1-PORT MegaWizard plug-in.

Figure 2–3. MegaWizard Plug-In Manager — ROM: 1-PORT [page 3]



Currently selected device family: Stratix III

How wide should the 'q' output bus be? 8 bits

How many 8-bit words of memory? 32 words

What should the RAM block type be?

● Auto ○ MLAB ○ M9K

○ M144K ○ LCs Options...

Set the maximum block depth to Auto words

What clocking method would you like to use?

○ Single clock

● Dual clock: use separate 'input' and 'output' clocks

Table 2–1 shows the features and settings of the ROM: 1-PORT MegaWizard plug-in page 3 options.

Table 2–1. ROM: 1-PORT MegaWizard Plug-In Page 3 Options (Part 1 of 2)

Function	Description
Currently selected device family:	Specify which Altera device family to use.
How wide should the 'q' output bus be?	Specify the width of the output data bus. Note that you can manually enter a number that is not in the pull-down list.
How many 8-bit words of memory?	Specify the number of 8-bit words in the memory. 8-bit represents the width of the output `q` data bus. You can set different widths of output data bus. Note that you can manually enter a number that is not in the pull-down list.
What should the RAM block type be?	Specify the memory block type. The options available depend on the device you select. *(1)*

Table 2–1. ROM: 1-PORT MegaWizard Plug-In Page 3 Options (Part 2 of 2)

Function	Description
What clocking method would you like to use?	Specify the clocking mode: single-clock mode or dual-clock mode.*(2)*

Notes to Table 2–1:

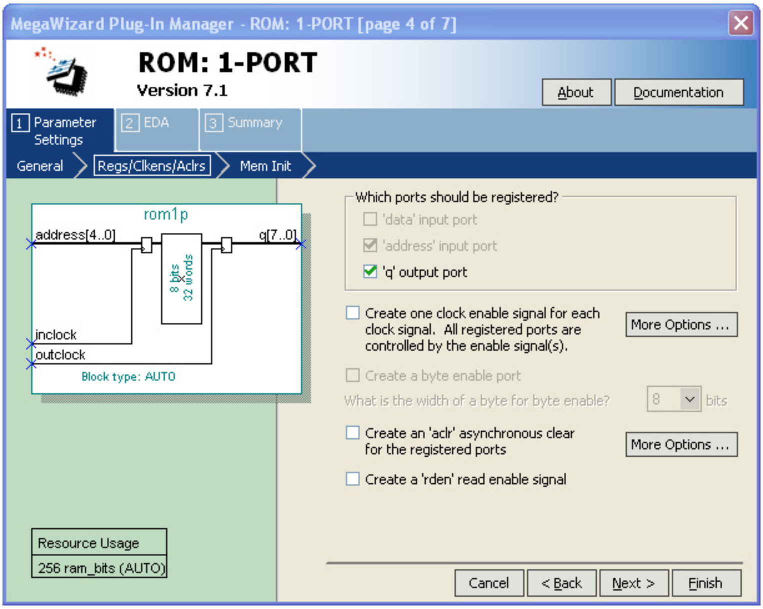
(1)

Available Memory Block Type Options	Associated Device or Devices
Auto / MLAB / M9K / M144K	Stratix III
Auto / M512 / M4K	Arria GX, Stratix II, Stratix II GX, Stratix, Stratix GX
Auto / M9K	Cyclone III
Auto / M4K	Cyclone II, Cyclone
Auto	Other legacy devices

(2) You must use dual-clock mode for older devices. Single- or dual-clock modes are available only in devices from the Stratix and Cyclone series of devices.

Page 4 of the ROM: 1-PORT MegaWizard plug-in is where you specify input and output ports for registration, and create a clock enable signal for each clock signal, an address-clock enable signal, and an asynchronous clear for the registered ports. Figure 2–4 shows page 4 of the ROM: 1-PORT MegaWizard plug-in.

Figure 2–4. MegaWizard Plug-In Manager — ROM: 1-PORT [page 4]



Which ports should be registered?

- □ 'data' input port
- ☑ 'address' input port
- ☑ 'q' output port

□ Create one clock enable signal for each clock signal. All registered ports are controlled by the enable signal(s).

□ Create a byte enable port

What is the width of a byte for byte enable? 8 bits

□ Create an 'aclr' asynchronous clear for the registered ports

□ Create a 'rden' read enable signal

Resource Usage
256 ram_bits (AUTO)

Table 2–2 shows the features and settings of the ROM: 1-PORT MegaWizard plug-in page 4 options.

Table 2–2. ROM: 1-PORT MegaWizard Plug-In Page 4 Options (Part 1 of 2)

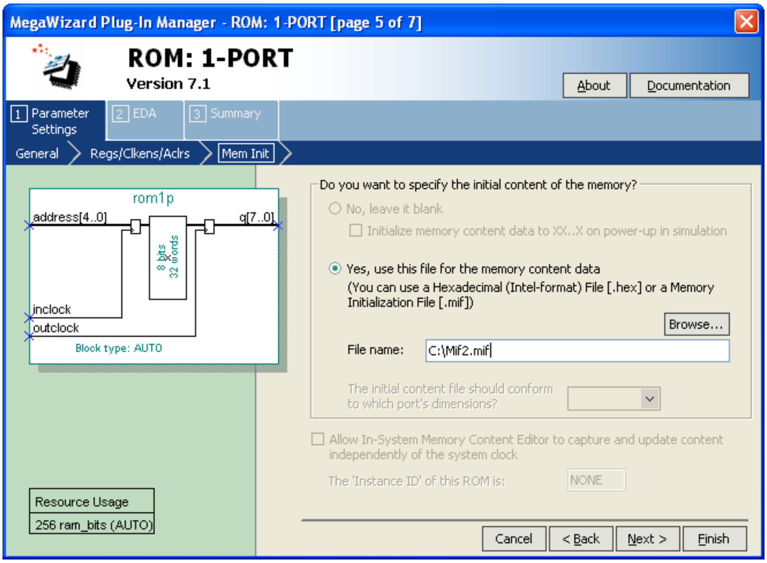
Function	Description
Which ports should be registered?	The ports available for registration are the address input port and the data output port. For Stratix III devices, you can unregister the read address port if you select MLAB as the memory block type.
Create one clock enable signal for each clock signal. All registered ports are controlled by the enable signal(s).	When turned on, a clock enable signal is created for each clock signal. This option is available in the Stratix and Cyclone series of devices only. You can bypass the clock enable signal for the input registers or for the output registers, or both, by clicking the **More Options** button to the right of this option. This **More Options** button also allows you to create a clock enable signal for the address register (`addressstall_a`). The address register holds the previous address value as long as the `addressstall_a` signal is high.

Table 2–2. ROM: 1-PORT MegaWizard Plug-In Page 4 Options (Part 2 of 2)

Function	Description
Create an 'aclr' asynchronous clear for the registered ports	Asynchronously clear the registered ports. You can select which registered ports are affected by the `aclr` signal by clicking the **More Options** button to the right of this option. All devices in the Stratix and Cyclone series of devices support asynchronous clear on ROM output registers, but only Stratix III, Stratix, Stratix GX, Cyclone III, Cyclone, and HardCopy Stratix devices support asynchronous clear on the read address input register. Asynchronous clears are available on ROM output latches and output registers in Stratix III and Cyclone III devices only.
Create a 'rden' read enable signal	Create a read enable signal to control read operations. This option is available only for Stratix III and Cyclone III devices with all memory block types except MLAB. Note that Cyclone III devices do not support the MLAB memory block type.

Page 5 of the ROM: 1-PORT MegaWizard plug-in is where you specify the initial content of memory, and select whether you want the In-System Memory Content Editor to capture and update content independently of the system clock. Figure 2–5 shows page 5 of the ROM: 1-PORT MegaWizard plug-in.

Figure 2–5. MegaWizard Plug-In Manager — ROM: 1-PORT [page 5]



Do you want to specify the initial content of the memory?

○ No, leave it blank

□ Initialize memory content data to XX..X on power-up in simulation

● Yes, use this file for the memory content data
(You can use a Hexadecimal (Intel-format) File [.hex] or a Memory Initialization File [.mif])

File name: C:\Mif2.mif

The initial content file should conform to which port's dimensions?

□ Allow In-System Memory Content Editor to capture and update content independently of the system clock

The 'Instance ID' of this ROM is: NONE

To complete the configuration of your single-port ROM, you must specify the file for the memory content data. If you do not already have one, you can create a Memory Initialization File (**.mif**) for the memory content data using the Quartus II Memory Editor. On the File menu, click **New**, and on the **Other Files** tab, select **Memory Initialization File**. Fill in the form to set the number of words and the word size, and click **OK**. Fill in the initial data for specific memory addresses, and save the file.

Table 2–3 shows the features and settings of the ROM: 1-PORT MegaWizard plug-in page 5 options.

Table 2–3. ROM: 1-PORT MegaWizard Plug-In Page 5 Options

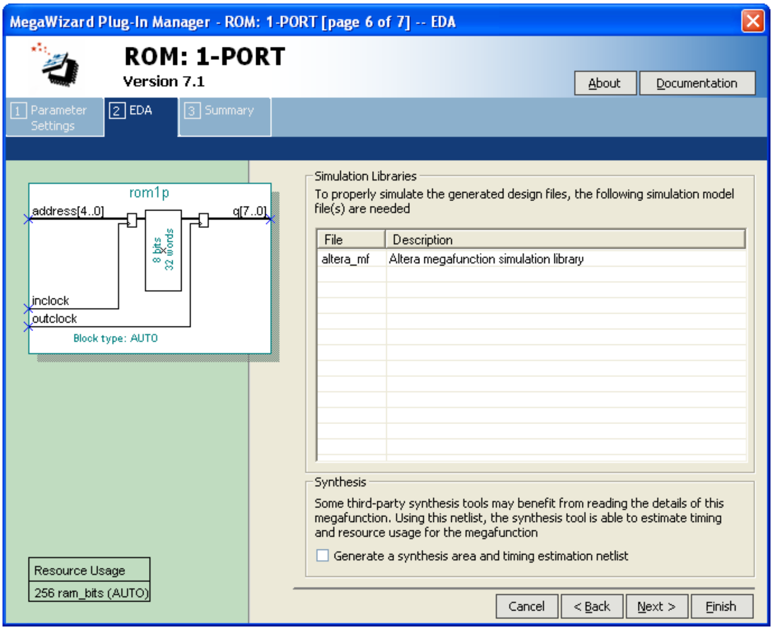
Function	Description
Do you want to specify the initial content of the memory?	Specify the initial content of the memory, leave it blank, or use a Hexadecimal File (**.hex**) or a Memory Initialization File (**.mif)** for the memory content data. You must specify the initial content of the ROM.
Allow In-System Memory Content Editor to capture and update content independently of the system clock	Turn on to enable In-System Memory Content. This option is not available for dual-clock mode.

For more information about viewing and modifying internal memories and constants, refer to the *In-System Updating of Memory and Constants* chapter in volume 3 of the *Quartus II Handbook*.

Page 6 shows the simulation libraries needed to properly simulate the generated design files and provides an option to generate a synthesis area and timing estimation netlist. By using the netlist, you enable some third-party synthesis tools to estimate timing and resource usage for the megafunction.

Figure 2–6 shows page 5 of the ROM: 1-PORT MegaWizard plug-in. No input is required for this page.

Figure 2–6. MegaWizard Plug-In Manager — ROM: 1-PORT [page 6]



MegaWizard Plug-In Manager - ROM: 1-PORT [page 6 of 7] -- EDA

ROM: 1-PORT
Version 7.1

Simulation Libraries

To properly simulate the generated design files, the following simulation model file(s) are needed

File	Description
altera_mf	Altera megafunction simulation library

Synthesis

Some third-party synthesis tools may benefit from reading the details of this megafunction. Using this netlist, the synthesis tool is able to estimate timing and resource usage for the megafunction

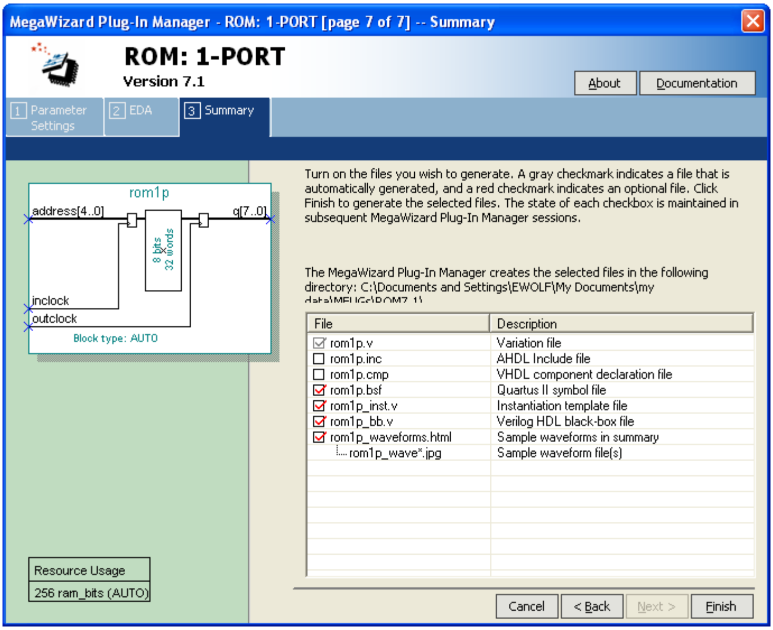
□ Generate a synthesis area and timing estimation netlist

Resource Usage
256 ram_bits (AUTO)

On page 7 of the ROM: 1-PORT MegaWizard plug-in, specify the types of files to be generated. Choose from the HDL wrapper file, <*function name*>**.v,** <*function name*>**.inc,** <*function name*> **.cmp**, <*function name*>**.bsf**, <*function name*>**_inst.v**, <*function name*>**_bb.v**, or <*function name*>**_waveforms.html**. A gray checkmark indicates a file that is automatically generated, and a red checkmark indicates an optional file.

Figure 2–7 shows page 7 of the ROM: 1-PORT MegaWizard plug-in.

Figure 2–7. MegaWizard Plug-In Manager — ROM: 1-PORT [page 7]



Turn on the files you wish to generate. A gray checkmark indicates a file that is automatically generated, and a red checkmark indicates an optional file. Click Finish to generate the selected files. The state of each checkbox is maintained in subsequent MegaWizard Plug-In Manager sessions.

The MegaWizard Plug-In Manager creates the selected files in the following directory: C:\Documents and Settings\EWOLF\My Documents\my data\MEUGs\ROM7.1\

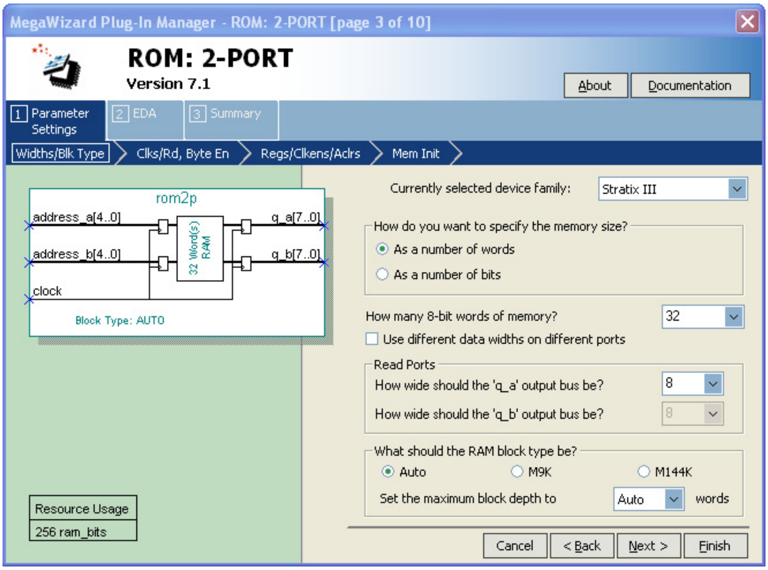
File	Description
☑ rom1p.v	Variation file
□ rom1p.inc	AHDL Include file
□ rom1p.cmp	VHDL component declaration file
☑ rom1p.bsf	Quartus II symbol file
☑ rom1p_inst.v	Instantiation template file
☑ rom1p_bb.v	Verilog HDL black-box file
☑ rom1p_waveforms.html	Sample waveforms in summary
rom1p_wave*.jpg	Sample waveform file(s)

The ROM: 2-PORT MegaWizard Plug-In Page Descriptions

On page 2a of the MegaWizard Plug-In Manager, under **Memory Compiler**, you can select the **ROM: 2-PORT** MegaWizard plug-in. Figure 2–2 shows page 2a of the MegaWizard Plug-In Manager.

Page 3 of the ROM: 2-PORT MegaWizard plug-in is where you specify the device family, specify memory size, set the width of the `q_a` and `q_b` output buses, and select the memory block type. Figure 2–8 shows page 3 of the ROM: 2-PORT MegaWizard plug-in.

Figure 2–8. MegaWizard Plug-In Manager – ROM: 2-PORT [page 3]



MegaWizard Plug-In Manager - ROM: 2-PORT [page 3 of 10]

ROM: 2-PORT

Version 7.1

Currently selected device family: Stratix III

How do you want to specify the memory size?

● As a number of words

○ As a number of bits

How many 8-bit words of memory? 32

□ Use different data widths on different ports

Read Ports

How wide should the 'q_a' output bus be? 8

How wide should the 'q_b' output bus be? 8

What should the RAM block type be?

● Auto ○ M9K ○ M144K

Set the maximum block depth to Auto words

Resource Usage

256 ram_bits

Table 2–4 shows the features and settings of the ROM: 2-PORT MegaWizard plug-in page 3 options.

Table 2–4. ROM: 2-PORT MegaWizard Plug-In Page 3 Options

Function	Description
Currently selected device family:	Specify which Altera device family to use.
How do you want to specify the memory size?	Specify the memory size as a number of words or as a number of bits.
How many 8-bit words of memory? or How many bits of memory?	Specify the number of 8-bit words or bits in the memory. 8-bit represents the width of the output `q_a` data bus. You can set different widths of output data bus. Note that you can manually enter a number that is not on the pull-down list.
Use different data widths on different ports	Select this option to set a different width for each output bus. *(1)*
Read Ports	Set the width of output data bus. If the option **Use different data widths on different ports** is enabled, you can set a different width for each of the output data busses `q_a` and `q_b`. Note that you can manually enter numbers that are not on the pull-down lists.
What should the RAM block type be?	Specify the memory block type. The options available depend on the device you select. *(2)*

Notes to Table 2–4:

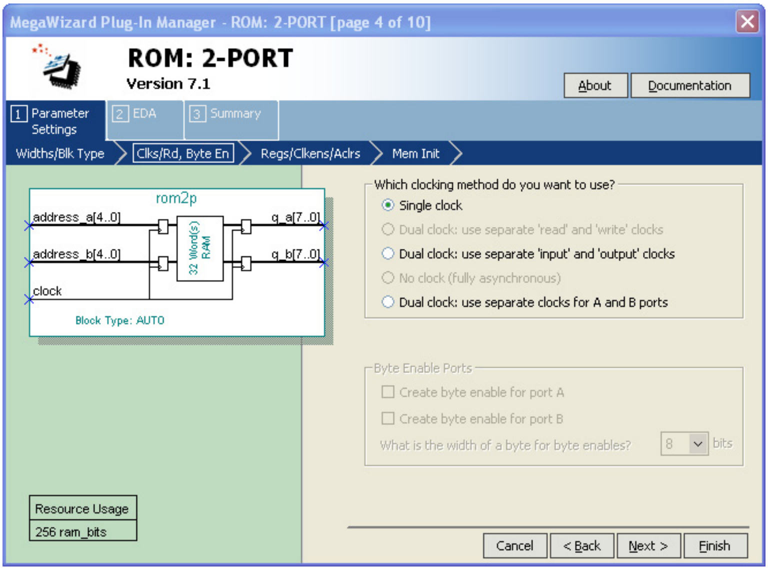
(1) Only the Stratix and Cyclone series of devices support this option.

(2)

Available Memory Block Type Options	Associated Device or Devices
Auto / M9K / M144K	Stratix III
Auto / M9K	Cyclone III
Auto / M4K	Arria GX, Stratix II, Stratix II GX, Stratix, Stratix GX, Cyclone II, Cyclone
Auto	Apex II

Page 4 of the ROM: 2-PORT MegaWizard plug-in is where you select the clocking mode for the dual-port ROM. Figure 2–9 shows page 4 of the ROM: 2-PORT MegaWizard plug-in.

Figure 2–9. MegaWizard Plug-In Manager — ROM: 2-PORT [page 4]



Which clocking method do you want to use?

- ● Single clock
- ○ Dual clock: use separate 'read' and 'write' clocks
- ○ Dual clock: use separate 'input' and 'output' clocks
- ○ No clock (fully asynchronous)
- ○ Dual clock: use separate clocks for A and B ports

Byte Enable Ports

- □ Create byte enable for port A
- □ Create byte enable for port B

What is the width of a byte for byte enables? 8 bits

Resource Usage: 256 ram_bits

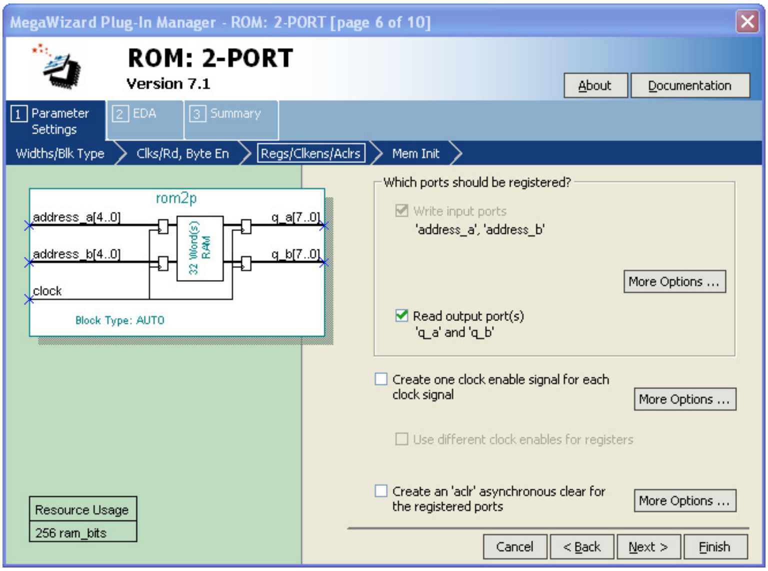
Table 2–5 shows the features and settings of the ROM: 2-PORT MegaWizard plug-in page 4 options.

Table 2–5. ROM: 2-PORT MegaWizard Plug-In Page 4 Options

Function	Description
Which clocking method do you want to use?	Three clocking modes are available: single-clock mode, dual-clock mode, and asynchronous mode. Two types of dual-clock mode are available: independent clock mode and input/output clock mode.
Single clock	In this clock mode, input and output are synchronous with the same clock.
Dual clock: use separate 'input' and 'output' clocks	In this type of dual-clock mode, one clock is dedicated to input signals and one clock is dedicated to output signals. Only the Stratix and Cyclone series of devices support this clocking mode.
No clock (fully asynchronous)	In this clock mode, no clock is required. Only APEX II devices support this clocking mode.
Dual clock: use separate clocks for A and B ports	In this type of dual clock mode, each of port A (input and output) and port B (input and output) has its own dedicated clock.

Page 6 of the ROM: 2-PORT MegaWizard plug-in is where you specify input and output ports for registration, create a clock enable signal for each clock signal, and create an asynchronous clear for the registered ports. Figure 2–10 shows page 6 of the ROM: 2-PORT MegaWizard plug-in

Figure 2–10. MegaWizard Plug-In Manager — ROM: 2-PORT [page 6]



Which ports should be registered?

☑ Write input ports 'address_a', 'address_b'

☑ Read output port(s) 'q_a' and 'q_b'

□ Create one clock enable signal for each clock signal

□ Use different clock enables for registers

□ Create an 'aclr' asynchronous clear for the registered ports

Resource Usage

256 ram_bits

Table 2–6 shows the features and settings of the ROM: 2-PORT MegaWizard plug-in page 6 options.

Table 2–6. ROM: 2-PORT MegaWizard Plug-In Page 6 Options

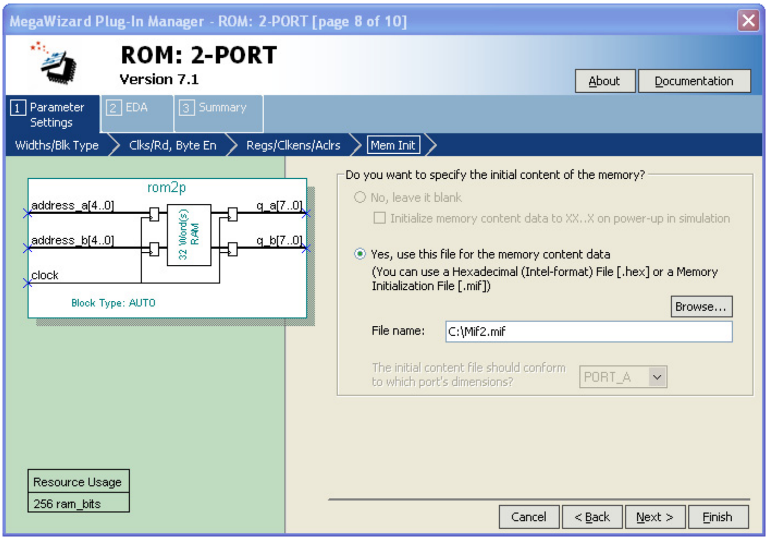
Function	Description
Which ports should be registered?	The ports available for registration are the two address input ports, and the two data output ports. In the Stratix and Cyclone series of devices, both address input ports must be registered.
Create one clock enable signal for each clock signal	When turned on, a clock enable signal is created for each clock signal. This option is available in APEX II devices and in all devices in the Stratix and Cyclone series of devices. In some devices, additional settings are available. If the button is available, you can bypass the clock enable signal for the input registers or for the output registers, or both, by clicking the **More Options** button to the right of this option. This **More Options** button also allows you to create address-clock enable signals for the address registers (`addressstall_a` and `addressstall_b`). The address register holds the previous address value as long as its `addressstall` signal is high.
Create an 'aclr' asynchronous clear for the registered ports	Asynchronously clear the registered ports. You can select which registered ports are affected by the `aclr` signal by clicking the **More Options** button to the right of this option. The availability of this button depends on the device you select. Asynchronous clears are available on ROM output latches and output registers in Stratix III and Cyclone III devices only.

Page 8 of the ROM: 2-PORT MegaWizard plug-in is where you specify the initial content of memory. You can use Hexadecimal File (**.hex**) or a Memory Initialization File (**.mif**) for the memory content data.

☞ To complete the configuration of your dual-port ROM, you must specify the file for the memory content data. If you do not have one, you can create a Memory Initialization File (**.mif**) for the memory content data using the Quartus II Memory Editor. On the File menu, click **New**, and on the **Other Files** tab, select **Memory Initialization File**. Fill in the form to set the number of words and the word size, and click **OK**. Fill in the form to set the initial data for specific memory addresses, and save the file.

Figure 2–11 shows page 8 of the ROM: 2-PORT MegaWizard plug-in.

Figure 2–11. MegaWizard Plug-In Manager — ROM: 2-PORT [page 8]



MegaWizard Plug-In Manager - ROM: 2-PORT [page 8 of 10]

ROM: 2-PORT

Version 7.1

Do you want to specify the initial content of the memory?

○ No, leave it blank

□ Initialize memory content data to XX..X on power-up in simulation

● Yes, use this file for the memory content data
(You can use a Hexadecimal (Intel-format) File [.hex] or a Memory Initialization File [.mif])

File name: C:\Mif2.mif

The initial content file should conform to which port's dimensions? PORT_A

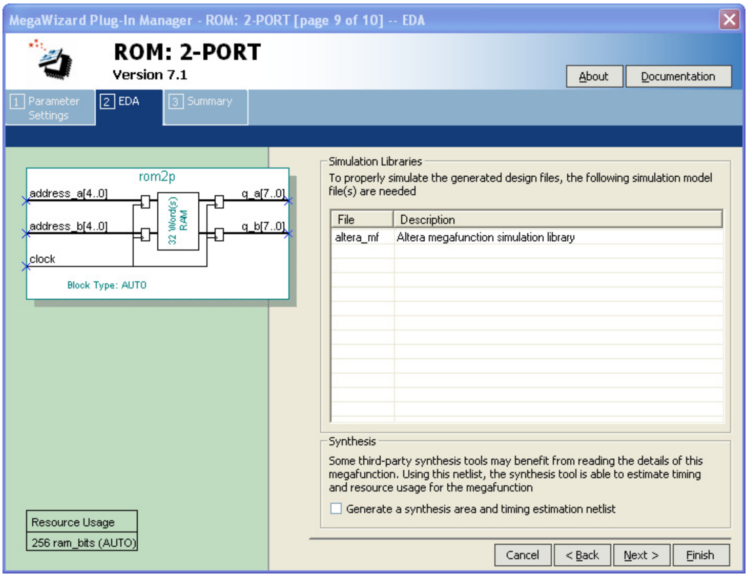
Resource Usage

256 ram_bits

Page 9 shows the simulation libraries needed to properly simulate the generated design files and provides an option to generate a synthesis area and timing estimation netlist. By using the netlist, you enable some third-party synthesis tools to estimate timing and resource usage for the megafunction.

Figure 2–12 shows page 9 of the ROM: 2-PORT MegaWizard plug-in. No input is required for this page.

Figure 2–12. MegaWizard Plug-In Manager — ROM: 2-PORT [page 9]



Resource Usage
256 ram_bits (AUTO)

Simulation Libraries

To properly simulate the generated design files, the following simulation model file(s) are needed

File	Description
altera_mf	Altera megafunction simulation library

Synthesis

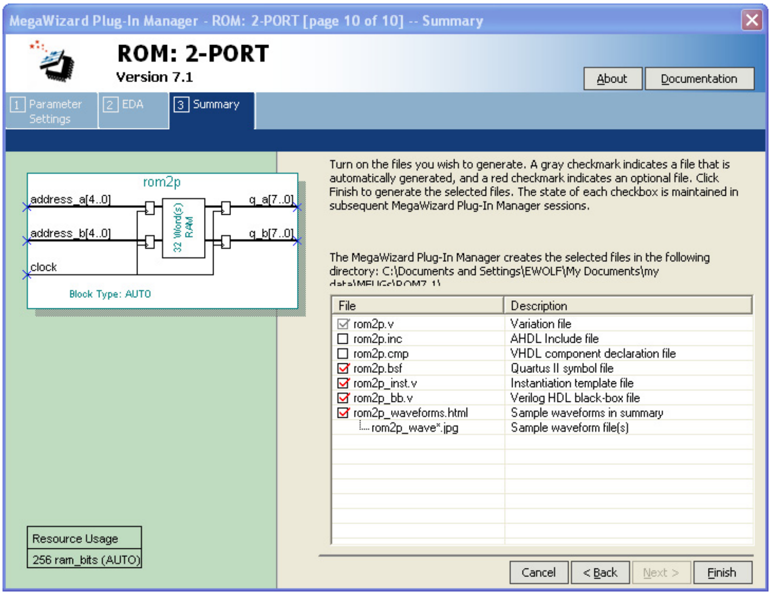
Some third-party synthesis tools may benefit from reading the details of this megafunction. Using this netlist, the synthesis tool is able to estimate timing and resource usage for the megafunction

□ Generate a synthesis area and timing estimation netlist

On page 10 of the ROM: 2-PORT MegaWizard plug-in, specify the types of files to be generated. Choose from the HDL wrapper file, <*function name*>**.v,** <*function name*>**.inc,** <*function name*> **.cmp**, <*function name*>**.bsf**, <*function name*>**_inst.v**, <*function name*>**_bb.v**, or <*function name*>_**waveforms.html**. A gray checkmark indicates a file that is automatically generated, and a red checkmark indicates an optional file.

Figure 2–13 shows page 10 of the ROM: 2-PORT MegaWizard plug-in.

Figure 2–13. MegaWizard Plug-In Manager — ROM: 2-PORT [page 10]



Turn on the files you wish to generate. A gray checkmark indicates a file that is automatically generated, and a red checkmark indicates an optional file. Click Finish to generate the selected files. The state of each checkbox is maintained in subsequent MegaWizard Plug-In Manager sessions.

The MegaWizard Plug-In Manager creates the selected files in the following directory: C:\Documents and Settings\EWOLF\My Documents\my data\MFUGs\ROM7.1\

File	Description
☑ rom2p.v	Variation file
☐ rom2p.inc	AHDL Include file
☐ rom2p.cmp	VHDL component declaration file
☑ rom2p.bsf	Quartus II symbol file
☑ rom2p_inst.v	Instantiation template file
☑ rom2p_bb.v	Verilog HDL black-box file
☑ rom2p_waveforms.html	Sample waveforms in summary
rom2p_wave*.jpg	Sample waveform file(s)

Resource Usage
256 ram_bits (AUTO)

For more information about the ports and parameters for the ROM megafunctions, refer to Chapter 3, Specifications.

Inferring Megafunctions from HDL Code

Synthesis tools, including Quartus II integrated synthesis, recognize certain types of HDL code and automatically infer the appropriate megafunction when a megafunction can provide optimal results. The Quartus II software uses the Altera megafunction code when compiling your design, even though you may not have specifically instantiated a megafunction. The Quartus II software infers the altsyncram megafunction because it is optimized for Altera devices, so area usage, performance, or both may be better than generic HDL code. Additionally, you must use megafunctions to access certain Altera architecture-specific features, including memory, DSP blocks, and shift registers. These features provide improved performance when compared to basic logic elements.

Refer to *Recommended HDL Coding Styles* in volume 1 of the *Quartus II Handbook* for specific information about your particular megafunction.

Instantiating Megafunctions in HDL Code or Schematic Designs

When you use the MegaWizard Plug-In Manager to customize and parameterize a megafunction, it creates a set of output files that allow you to instantiate the customized function in your design. Depending on the language you choose in the MegaWizard Plug-In Manager, the MegaWizard instantiates the megafunction with the correct parameter values and generates a megafunction variation file (wrapper file) in Verilog HDL (**.v**), VHDL (**.vhd**), or AHDL (**.tdf**), along with other supporting files.

The MegaWizard Plug-In Manager provides options to create the following files:

- A sample instantiation template for the language of the variation file (**_inst.v**, **_inst.vhd**, or **_inst.tdf**)
- Component Declaration File (**.cmp**) that can be used in VHDL Design Files
- ADHL Include File (**.inc**) that can be used in Text Design Files (**.tdf**)
- Quartus II Block Symbol File (**.bsf**) that can be used in schematic designs
- Verilog HDL module declaration file that can be used when instantiating the megafunction as a black box in a third-party synthesis tool (**_bb.v**)

For more information about the MegaWizard-generated files, refer to Quartus II Help or to the *Recommended HDL Coding Styles* chapter in volume 1 of the *Quartus II Handbook.*

Generating a Netlist for EDA Tool Use

If you use a third-party EDA synthesis tool, you can instantiate the megafunction variation file as a black box for synthesis. Use the VHDL component declaration or Verilog module declaration black-box file to define the function in your synthesis tool, and then include the megafunction variation file in your Quartus II project.

If you enable the option to generate a synthesis area and timing estimation netlist in the MegaWizard Plug-In Manager, the MegaWizard generates an additional netlist file (**_syn.v**). The netlist file is a representation of the customized logic used in the Quartus II software. The file provides the connectivity of the architectural elements in the megafunction but may not represent true functionality. This information enables certain third-party synthesis tools to better report area and timing estimates. In addition, synthesis tools can use the timing information to focus timing-driven optimizations and improve the quality of results.

For more information about using megafunctions in your third-party synthesis tool, refer to the appropriate chapter in the *Synthesis* section in volume 1 of the *Quartus II Handbook*.

Using the Port and Parameter Definitions

Instead of the MegaWizard Plug-In Manager, you can instantiate the megafucntion directly in your Verilog HDL, VHDL, or AHDL code by calling the megafunction and settings its parameters as you would any other module, component, or subdesign.

Altera strongly recommends that you use the MegaWizard Plug-In Manager for complex megafunctions. The MegaWizard Plug-In Manager ensures that you set all megafunction parameters properly.

Refer to Chapter 3, Specifications for a list of the megafunction ports and parameters.

Identifying a Megafunction After Compilation

During compilation with the Quartus II software, analysis and elaboration is performed to build the structure of your design. You can locate your megafunction in the Project Navigator window by expanding the compilation hierarchy and locating the megafunction by its name.

Similarly, to search for node names within the megafunction using the Node Finder, in the **Look in** box, click **Browse** and select the megafunction from the Hierarchy box.

Simulation

The Quartus II Simulator provides an easy-to-use, integrated solution for performing simulations. The following sections describe the simulation options.

Quartus II Software Simulation

With the Quartus II Simulator, you can perform two types of simulations: functional and timing. A functional simulation enables you to verify the logical operation of your design without taking into consideration the timing delays in the FPGA. This simulation is performed using only your RTL code. When performing a functional simulation, add only signals that exist before synthesis. You can find these signals with the Registers: Pre-Synthesis, Design Entry, or Pin filters in the Node Finder. The top-level ports of megafunctions are found using these three filters.

In contrast, the timing simulation in the Quartus II software verifies the operation of your design with annotated timing information. This simulation is performed using the post place-and-route netlist. When performing a timing simulation, add only signals that exist after place-and-route. These signals are found with the post-compilation filter of the Node Finder. During synthesis and place-and-route, the names of RTL signals change. Finding signals from your megafunction instantiation in the post-compilation filter therefore may be difficult.

To preserve the names of your signals during the synthesis and place-and-route stages, use the synthesis attributes `keep` or `preserve`. These are Verilog and VHDL synthesis attributes that direct analysis and synthesis to keep a particular wire, register, or node intact. Use these synthesis attributes to keep a combinational logic node so you can observe the node during simulation.

For more information about these attributes, refer to the *Quartus II Integrated Synthesis* chapter in volume 1 of the *Quartus II Handbook.*

EDA Simulation

For more information about EDA simulation, refer to the appropriate chapter (based on the tool you use) in the *Simulation* section in volume 3 of the *Quartus II Handbook*. The *Quartus II Handbook* chapters describe how to perform functional and gate-level timing simulations that include the megafunctions, with details about the files that are needed and the directories where the files are located.

SignalTap II Embedded Logic Analyzer

The SignalTap® II embedded logic analyzer provides a non-intrusive method of debugging the Altera megafunctions within your design. With the SignalTap II embedded logic analyzer, you can capture and analyze data samples for the top-level ports of Altera megafunctions while your system is running at full speed.

To monitor signals from Altera megafunctions, configure the SignalTap II embedded logic analyzer in the Quartus II software, and include the analyzer as part of your Quartus II project. The Quartus II software then embeds the analyzer in your design in the selected device seamlessly.

For more information about using the SignalTap II embedded logic analyzer, refer to the *Design Debugging Using the SignalTap II Embedded Logic Analyzer* chapter in volume 3 of the *Quartus II Handbook.*

In-System Updating of Memory and Constants

FPGA designs are growing larger in density and are becoming more complex. Designers and verification engineers require more access to the design that is programmed in the device to identify, test, and resolve issues quickly and accurately. The In-System Updating of Memory and Constants capability of the Quartus II software provides you with a non-intrusive method of accessing your ROM within the Altera FPGA. With the In-System Memory Content Editor, you can capture, analyze, and update ROM data while your system is running at full speed.

To gain access to your ROM megafunction, enable the In-System Updating of Memory and Constants feature within the MegaWizard Plug-In Manager. The Quartus II software will then modify your ROM (in the background) so you have access to it while the FPGA is processing. You can read, write, or update the contents of your ROM multiple times without having to reconfigure your FPGA. For ROM megafunctions, the In-System Updating of Memory and Constants feature is only supported by single-port ROM, and only with certain configuration settings. For example, this feature is not available when you select MLAB as your memory block type, or if you are using dual-clock mode.

For more information about viewing and modifying internal memories and constants, refer to the *In-System Updating of Memory and Constants* chapter in volume 3 of the *Quartus II Handbook.*

Design Examples for the ROM Megafunctions

This section presents two design examples that use the ROM: 1-PORT and ROM: 2-PORT MegaWizard plug-ins to generate single-port ROM and dual-port ROM, respectively. For each design example, a table specifies the configuration settings used to generate it. When you are finished with the examples, you can incorporate them into your overall project.

Design Files

The example design files are available with this user guide in the Quartus II Project section and in the User Guides section of the Altera website (**www.altera.com**).

Example for ROM: 1-PORT

The objective of this example is to implement and instantiate a single-port ROM using the ROM: 1-PORT MegaWizard plug-in. The ROM: 1-PORT example illustrates single-clock mode with unregistered output mode.

This example also illustrates the new feature supported by Stratix III devices for the ROM: 1-PORT MegaWizard plug-in, specifically the `read enable` signal to control read operation. Cyclone IIII devices also support this feature, although this fact is not illustrated by the example. Address clock enable (`addressstall`) is another feature of single-port ROM that is illustrated in this example. Verify the results you obtain at the end of this example with the expected simulation results provided.

In this example, you perform the following activities:

- Generate a single-port ROM using the ROM: 1-PORT MegaWizard plug-in
- Implement the single-port ROM by assigning the Stratix III device to the project and compiling the project
- Simulate the single-port ROM design

Generate the Single-Port ROM

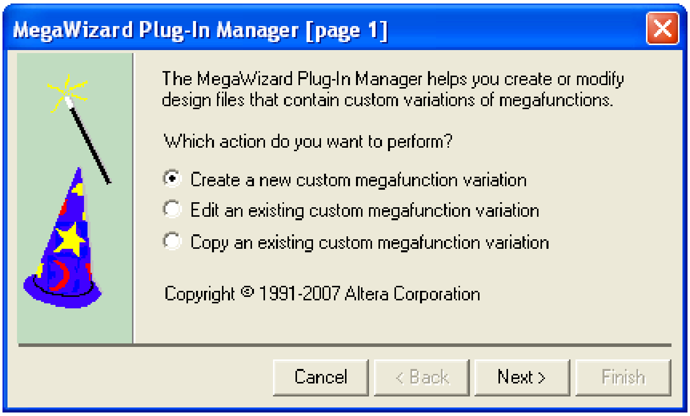
1. Open **rom1p_DesignExample_ex1.zip** and extract **rom1p.qar** to any working directory.

2. In the Quartus II software, open **rom1p.qar** and restore the archive file into your working directory.

   ☞ The **rom1p.qar** project includes a memory initialization file (**rom.mif**). You must specify this file while you configure the single-port ROM through the wizard for this example. This file defines the initial data at memory addresses 00, 01, 02, 03, 04, and 05 as 0xFA, 0xFB, 0xFC, 0xFD, 0xFE, and 0xFF, respectively. The **rom1p.qar** project also includes a vector waveforms file (**rom1p.vwf**) for running functional simulation.

3. On the Tools menu, click **MegaWizard Plug-In Manager**.
   Page 1 of the MegaWizard Plug-In Manager appears (Figure 2–14).

Figure 2–14. MegaWizard Plug-In Manager [page 1]



MegaWizard Plug-In Manager [page 1]

The MegaWizard Plug-In Manager helps you create or modify design files that contain custom variations of megafunctions.

Which action do you want to perform?

● Create a new custom megafunction variation

○ Edit an existing custom megafunction variation

○ Copy an existing custom megafunction variation

Copyright © 1991-2007 Altera Corporation

4. Select **Create a new custom megafunction variation**, and click **Next**. Page 2a appears.

5. On page 2a of the MegaWizard Plug-In Manager, make the following selections:

   a. In the **Which device family will you be using?** list, select **Stratix III**.

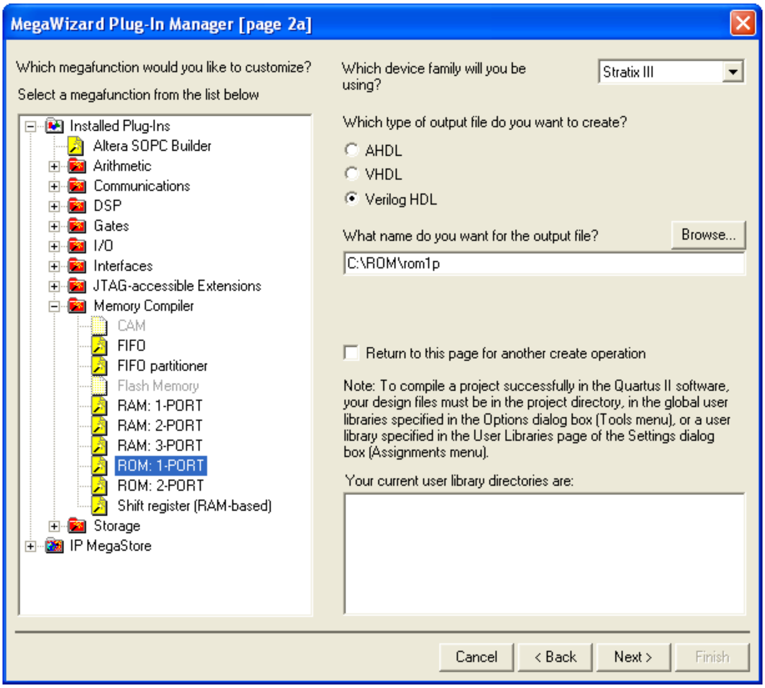
   b. Under **Which type of output file do you want to create?**, select **Verilog HDL**.

   c. Expand the **Memory Compiler** folder and select **ROM: 1-PORT**.

   d. For the name of the output file, type `rom1p`, or click **Browse** to select the file from the project folder.

Figure 2–15 shows page 2a after you have made these selections.

Figure 2–15. MegaWizard Plug-In Manager [page 2a]



Which megafunction would you like to customize?

Select a megafunction from the list below

Which device family will you be using? Stratix III

Which type of output file do you want to create?

○ AHDL

○ VHDL

● Verilog HDL

What name do you want for the output file?

C:\ROM\rom1p

□ Return to this page for another create operation

Note: To compile a project successfully in the Quartus II software, your design files must be in the project directory, in the global user libraries specified in the Options dialog box (Tools menu), or a user library specified in the User Libraries page of the Settings dialog box (Assignments menu).

Your current user library directories are:

6. Click **Next.** For the remaining pages, configure the single-port ROM using the options provided in Table 2–7.

Table 2–7. ROM: 1-PORT MegaWizard Plug-In Page 3, 4, and 5 Options

Function	Options
Currently selected device family: *(1)*	Stratix III
How wide should the 'q' output bus be? *(1)*	8 bits
How many 8-bit words of memory? *(1)*	32 words
What should the RAM block type be? *(1)*	Auto
Set the maximum block depth to *(1)*	Auto words
What clocking method would you like to use? *(1)*	Single clock
Which ports should be registered? *(2)*	The 'address' input port is selected. Deselect the 'q' output port
Create one clock enable signal for each clock signal. All registered ports are controlled by the enable signal(s). *(2)*	Click on **More Options** and, under Address options, select **Create an 'addressstall_a' input port**
Create a 'rden' enable signal *(2)*	Select this option
File name *(3)*	Browse to your project directory and specify the **rom.mif** file

Notes to Table 2–7:

(1) This option appears on page 3 of the ROM: 1-PORT MegaWizard plug-in.

(2) This option appears on page 4 of the ROM: 1-PORT MegaWizard plug-in.

(3) This option appears on page 5 of the ROM: 1-PORT MegaWizard plug-in.

7. Click **Finish.** Page 7 of the wizard appears. This page requires no input.

8. Click **Finish** again to complete the configuration settings.

The **rom1p** module is now built.

Implement Single-Port ROM

Next, assign the EP3SE50F484C2 device to the project and compile the project.

1. On the Assignments menu, click **Settings**. The **Settings** dialog box appears.

2. Under **Category**, select **Device**.

3. In the **Family** field, select **Stratix III**.

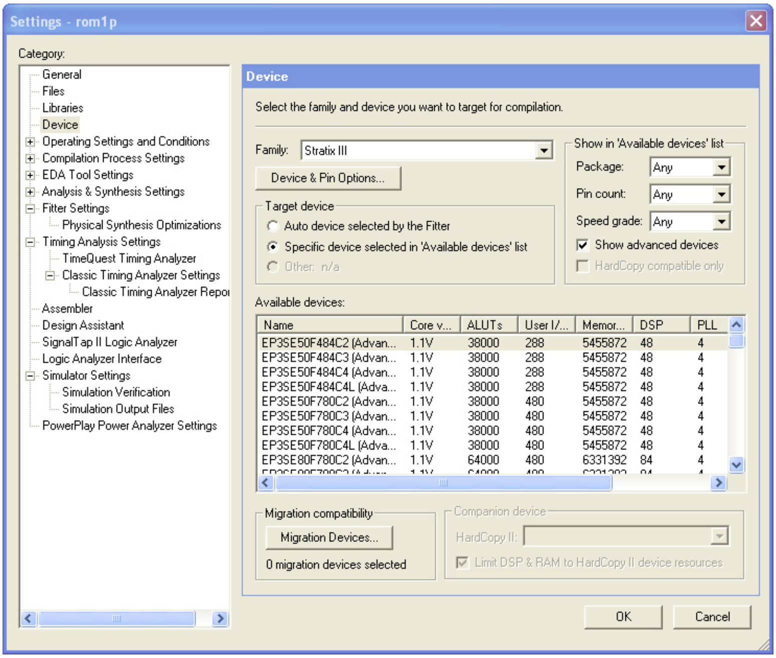
4. Under **Target Device**, ensure that **Specific device selected in 'Available devices' list** is selected.

5. In the **Available devices:** list, select **EP3SE50F484C2**.

6. Leave all other variables at their default values.

Figure 2–16 shows the **Settings** dialog box after you have made these selections.

Figure 2–16. Device Settings Dialog Box for ROM: 1-PORT Design Example



Select the family and device you want to target for compilation.

Family: Stratix III

Target device

○ Auto device selected by the Fitter
● Specific device selected in 'Available devices' list
○ Other: n/a

Show in 'Available devices' list

Package: Any
Pin count: Any
Speed grade: Any
☑ Show advanced devices
□ HardCopy compatible only

Available devices:

Name	Core v...	ALUTs	User I/...	Memor...	DSP	PLL
EP3SE50F484C2 (Advan...	1.1V	38000	288	5455872	48	4
EP3SE50F484C3 (Advan...	1.1V	38000	288	5455872	48	4
EP3SE50F484C4 (Advan...	1.1V	38000	288	5455872	48	4
EP3SE50F484C4L (Adva...	1.1V	38000	288	5455872	48	4
EP3SE50F780C2 (Advan...	1.1V	38000	480	5455872	48	4
EP3SE50F780C3 (Advan...	1.1V	38000	480	5455872	48	4
EP3SE50F780C4 (Advan...	1.1V	38000	480	5455872	48	4
EP3SE50F780C4L (Adva...	1.1V	38000	480	5455872	48	4
EP3SE80F780C2 (Advan...	1.1V	64000	480	6331392	84	4

Migration compatibility

0 migration devices selected

Companion device

HardCopy II:

☑ Limit DSP & RAM to HardCopy II device resources

7. Click **OK**.

8. To compile the design, on the Processing menu, click **Start Compilation**, or on the toolbar, click the **Start Compilation** button.

9. When the **Full Compilation was successful** message box appears, click **OK**.

Functional Results: Simulate the Single-Port ROM in the Quartus II Software

Finally, simulate the design to verify the results. Set up the Quartus II Simulator by performing the following steps:

1. On the **Processing** menu, click **Generate Functional Simulation Netlist**.

2. When the **Functional Simulation Netlist Generation was successful** message box appears, click **OK**.

3. On the **Assignments** menu, click **Settings**. The **Settings** dialog box appears.

4. Under **Category**, select **Simulator Settings**.

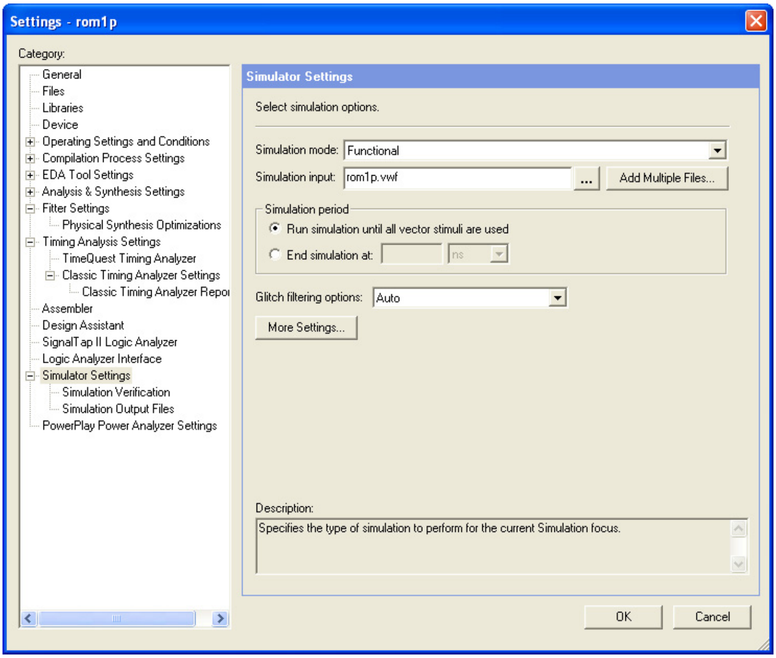
5. In the **Simulation mode** pull-down list, select **Functional**.

6. In the **Simulation input** box, type `rom1p.vwf`, or click **Browse** to select the file from the project folder.

7. Turn on **Run simulation until all vector stimuli are used**.

Figure 2–17 shows the **Simulation Settings** dialog box after you have made these selections.

Figure 2–17. Simulator Settings Dialog Box



Select simulation options.

Simulation mode: Functional

Simulation input: rom1p.vwf

Simulation period

● Run simulation until all vector stimuli are used

○ End simulation at: ns

Glitch filtering options: Auto

Description:

Specifies the type of simulation to perform for the current Simulation focus.

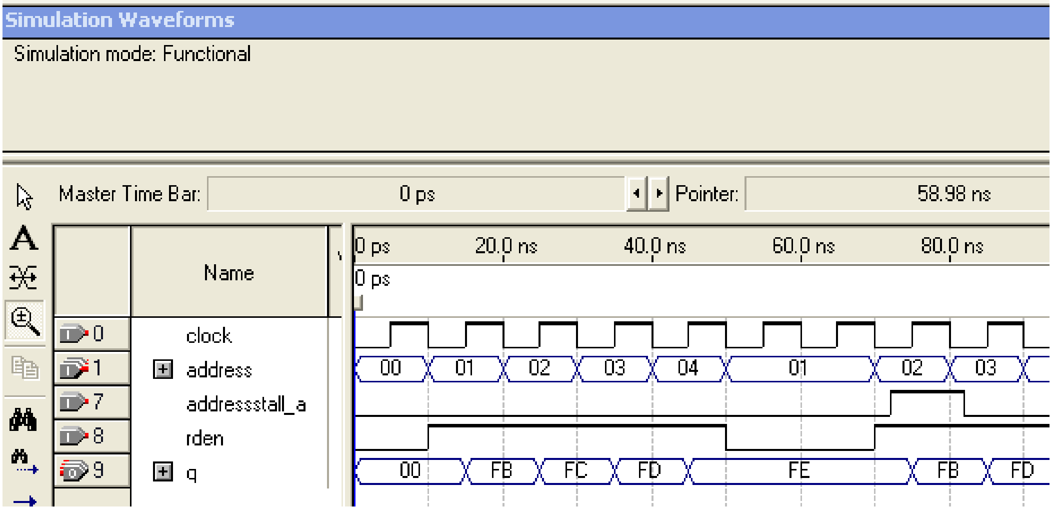
8. Click **OK**.

9. To run the simulation, on the **Processing** menu, click **Start Simulation,** or, on the toolbar, click the **Start Simulation** button.

10. When the **Simulator was successful** message box appears, click **OK**.

11. In the Simulation Report window, view the simulation waveforms to verify the results. Figure 2–18 shows the expected simulation results.

Figure 2–18. Functional Waveform for Single-Port ROM in Single-Clock Mode with Unregistered Output



Understanding the Simulation Results

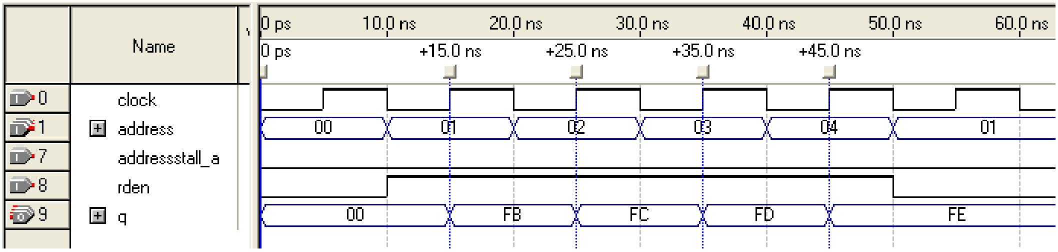
In this example, you configured the ROM: 1-PORT to have the following properties:

- Single clock
- Unregistered output port `q`
- `read enable` signal that is only supported by Stratix III and Cyclone III devices
- Address clock enable feature (`addressstall_a`)

The following section explains the simulation results corresponding to the configuration you set through the ROM: 1-PORT MegaWizard plug-in.

Figure 2–19 shows read operations with the read-enable feature for the single-port ROM.

Figure 2–19. Read Operations with Read-Enable Feature in Single-Port ROM

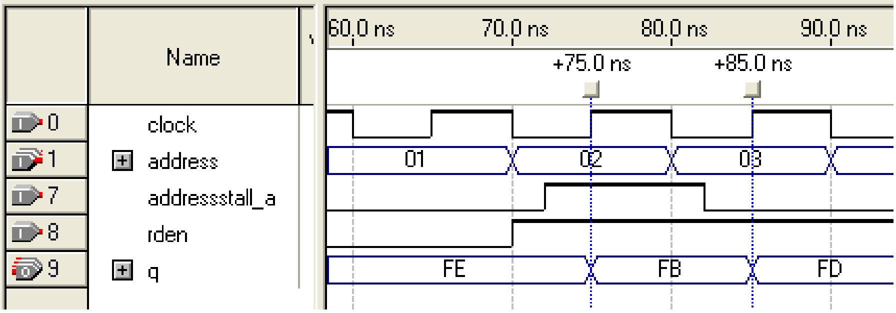


The memory addresses 00, 01, 02, 03, 04, and 05 hold data values 0xFA, 0xFB, 0xFC, 0xFD, 0xFE, and 0xFF, respectively, as set in the **rom.mif** file. Therefore, when `rden` is high, the `q` port successively outputs read data 0xFB from address 01 (at 15 ns), data 0xFC from address 02 (at 25 ns), data 0xFD from address 03 (at 35 ns), and data 0xFE from address 04 (at 45 ns). The read operations from address 00 at 5 ns and from address 01 at 55 ns are not successful because `rden` is low at those times.

Note that the read data appears on `q` immediately, because the ROM is configured to have unregistered output. If you want the data to be read out at the next rising edge of the clock, configure your ROM with registered output.

Figure 2–20 illustrates the address-clock enable feature in a single-port ROM.

Figure 2–20. Read Operations with Address-Clock Enable Feature in Single-Port ROM



At 75 ns, a read operation targets memory address 02. Because the `addressstall_a` signal is asserted, the internal address register holds its previous value, 0x01. Therefore, the output data is read from memory address 01 (which holds data value 0xFB) instead of from memory address 02 (which holds data value 0xFC). The address register maintains its previous value while the `addressstall_a` signal remains high. At 85 ns, a read from memory address 03 (which holds data value 0xFD) is successful when `addressstall_a` is deasserted.

Functional Results: Simulate the Single-Port ROM in ModelSim-Altera

Simulate the design in ModelSim to compare the results of both simulators. This User Guide assumes that you are familiar with using ModelSim-Altera before trying out the design example. If you are unfamiliar with ModelSim-Altera, refer to the support page for software products on the Altera website (**www.altera.com**). On the support page for ModelSim-Altera, there are various links to topics such as installation, usage, and troubleshooting.

Set up the ModelSim-Altera simulator by performing the following steps:

1. Unzip the **rom1p_ex1_msim.zip** file to any working directory on your PC.

2. Start ModelSim-Altera.

3. On the File menu, click **Change Directory**.

4. Select the folder in which you unzipped the files. Click **OK.**

5. On the Tools menu, select **Execute Macro**.

6. Select the **rom1p_msim.do** file and click **Open**.

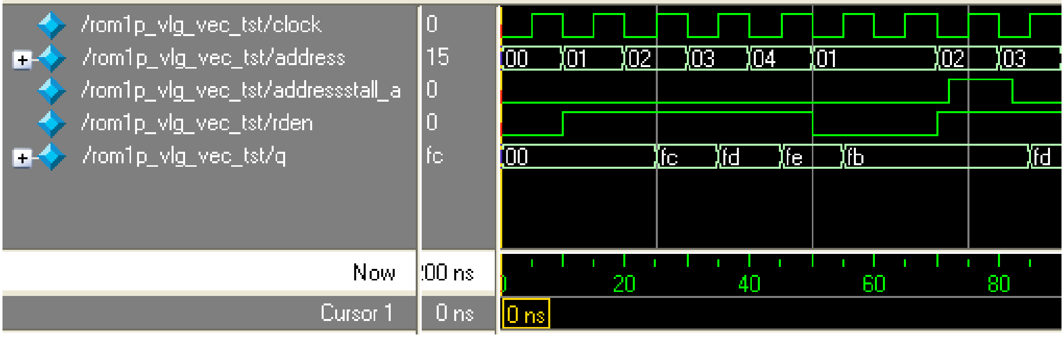
   The **rom1p_msim.do** file is a script file for ModelSim that automates all necessary settings for the simulation.

7. Verify the results shown in the Waveform Viewer window.

You can rearrange signals, remove signals, add signals, and change the radix by modifying the script in **rom1p_msim.do** accordingly to match the order of the results in the Quartus II simulator. Also, you may need to modify the script to specify your local path to the Stratix III library files.

Figure 2–21 shows the expected simulation results when running the design example in ModelSim-Altera. Compare the results obtained using the Quartus II simulator from Figure 2–18 on page 2–30. Refer to the section "Understanding the Simulation Results" on page 2–30 for an explanation of the simulation results.

Figure 2–21. ModelSim Simulation Waveform for Single-Port ROM



Example for ROM: 2-PORT

The objective of this example is to implement and instantiate a dual-port ROM using the ROM: 2-PORT MegaWizard plug-in. The ROM: 2-PORT example illustrates the independent clock mode (separate clocks for port A and port B) with unregistered output port A and registered output port B.

This example also illustrates the new feature, asynchronous clear on latch, that is supported only in Stratix III and Cyclone III devices. The asynchronous clear signal clears the output q even if the output is not registered. In this example, the output port A is unregistered to illustrate this feature. Verify the results you obtain at the end of this example with the expected simulation results provided.

In this example, you perform the following activities:

- Generate a dual-port ROM using the ROM: 2-PORT MegaWizard plug-in
- Implement the dual-port ROM by assigning the Stratix III device to the project and compiling the project
- Simulate the dual-port ROM design

Generate the Dual-Port ROM

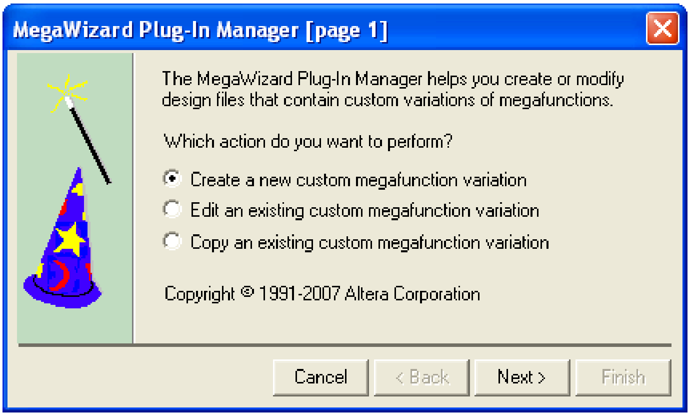
1. Open **rom2p_DesignExample_ex2.zip** and extract **rom2p.qar** to any working directory.

2. In the Quartus II software, open **rom2p.qar** and restore the archive file into your working directory.

☞ The **rom2p.qar** project includes a memory initialization file (**rom.mif**). You must specify this file while you configure the dual-port ROM through the wizard for this example. This file defines the initial data at memory addresses 00, 01, 02, 03, 04, and 05 as 0xFA, 0xFB, 0xFC, 0xFD, 0xFE, and 0xFF, respectively. The **rom2p.qar** project also includes a vector waveforms file (**rom2p.vwf**) for running functional simulation.

3. On the Tools menu, click **MegaWizard Plug-In Manager**. Page 1 of the MegaWizard Plug-In Manager appears (Figure 2–22).

Figure 2–22. MegaWizard Plug-In Manager [page 1]



MegaWizard Plug-In Manager [page 1]

The MegaWizard Plug-In Manager helps you create or modify design files that contain custom variations of megafunctions.

Which action do you want to perform?

● Create a new custom megafunction variation
○ Edit an existing custom megafunction variation
○ Copy an existing custom megafunction variation

Copyright © 1991-2007 Altera Corporation

4. Select **Create a new custom megafunction variation**, and click **Next**. Page 2a appears.

5. On page 2a of the MegaWizard Plug-In Manager, make the following selections:

   a. In the **Which device family will you be using?** list, select **Stratix III**.

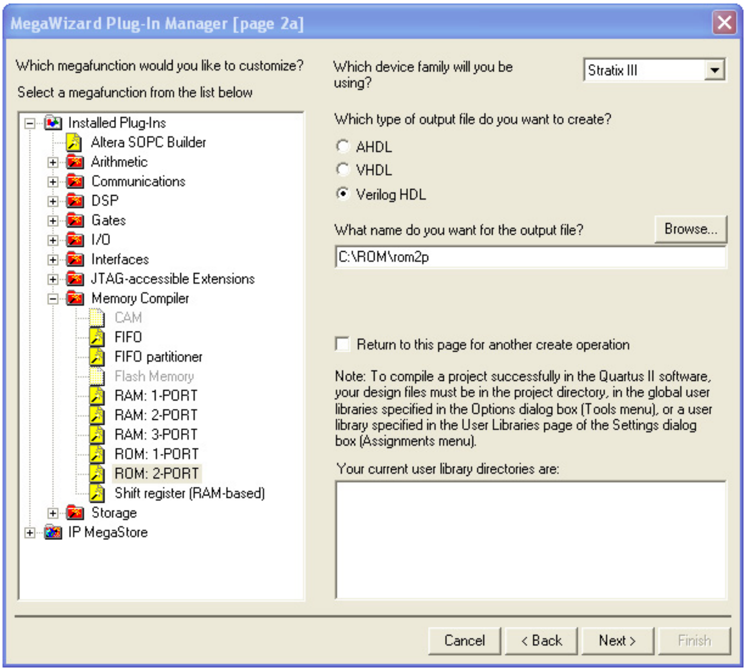
   b. Under **Which type of output file do you want to create?**, select **Verilog HDL**.

   c. Expand the **Memory Compiler** folder and select **ROM: 2-PORT**.

   d. For the name of the output file, type `rom2p`, or click **Browse** to select the file from the project folder.

Figure 2–23 shows page 2a after you have made these selections.

Figure 2–23. MegaWizard Plug-In Manager — ROM: 2-PORT [page 2a]



MegaWizard Plug-In Manager [page 2a]

Which megafunction would you like to customize?

Select a megafunction from the list below

Which device family will you be using? Stratix III

Which type of output file do you want to create?

○ AHDL
○ VHDL
● Verilog HDL

What name do you want for the output file? C:\ROM\rom2p

□ Return to this page for another create operation

Note: To compile a project successfully in the Quartus II software, your design files must be in the project directory, in the global user libraries specified in the Options dialog box (Tools menu), or a user library specified in the User Libraries page of the Settings dialog box (Assignments menu).

Your current user library directories are:

6. Click **Next.** For the remaining pages, configure the dual-port ROM using the options provided in Table 2–8.

Table 2–8. ROM: 2-PORT MegaWizard Plug-In Page 3, 4, 6, and 8 Options

Function	Options
Currently selected device family: *(1)*	Stratix III
How do you want to specify the memory size? *(1)*	As a number of words
How many 8-bit words of memory? *(1)*	32 words
Use different data widths on different ports *(1)*	Deselect this option *(5)*
How wide should the 'q_a' output bus be? *(1)*	8 *(5)*
What should the RAM block type be?*(1)*	Auto
Set the maximum block depth to *(1)*	Auto words
Which clocking method do you want to use? *(2)*	Select **Dual clock: use separate clocks for A and B ports**

Table 2–8. ROM: 2-PORT MegaWizard Plug-In Page 3, 4, 6, and 8 Options

Function	Options
Currently selected device family: *(1)*	Stratix III
How do you want to specify the memory size? *(1)*	As a number of words
How many 8-bit words of memory? *(1)*	32 words
Which ports should be registered? *(3)*	Click on **More Options**. The **'address'** input ports are selected. Deselect the **'q_a' port** and select the **'q_b' port** under **Q output ports**.
Create one clock enable signal for each clock signal *(3)*	Deselect this option
Create an 'aclr' asynchronous clear for the registered ports *(3)*	Select this option
File name*(4)*	Browse to your project directory and specify the **rom.mif** file

Notes to Table 2–8:

(1) This option appears on page 3 of the MegaWizard plug-in.
(2) This option appears on page 4 of the MegaWizard plug-in.
(3) This option appears on page 6 of the MegaWizard plug-in.
(4) This option appears on page 8 of the MegaWizard plug-in.
(5) In this example, the `q_a` and `q_b` output busses have the same width.

7. Click **Finish**. Page 10 of the wizard appears. This page requires no input.

8. Click **Finish** again to complete the configuration settings.

The **rom2p** module is now built.

Implement Dual-Port ROM

Implement the dual-port ROM and compile the design according to the instructions for implementing the single-port ROM. Refer to the section "Implement Single-Port ROM" on page 2–26 for these instructions.

Functional Results: Simulate the Dual-port ROM in the Quartus II Software

Finally, simulate the design to verify the results. Set up the Quartus II Simulator by performing the following steps:

1. On the **Processing** menu, click **Generate Functional Simulation Netlist**.

2. When the **Functional Simulation Netlist Generation was successful** message box appears, click **OK**.

3. On the **Assignments** menu, click **Settings**. The **Settings** dialog box appears.

4. Under **Category**, select **Simulator Settings**.

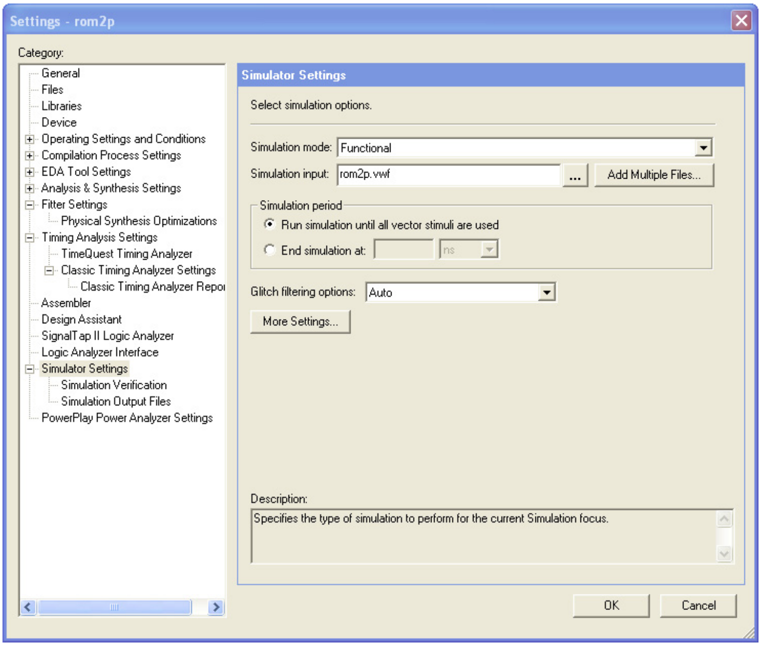
5. In the **Simulation mode** pull-down list, select **Functional**.

6. In the **Simulation input** box, type `rom2p.vwf`, or **Browse** to select the file from the project folder.

7. Turn on **Run simulation until all vector stimuli are used**.

Figure 2–24 shows the **Simulation Settings** dialog box after you have made these selections.

Figure 2–24. Simulator Settings Dialog Box for ROM: 2-PORT Design Example



Settings - rom2p

Category:

- General
- Files
- Libraries
- Device
- Operating Settings and Conditions
- Compilation Process Settings
- EDA Tool Settings
- Analysis & Synthesis Settings
- Fitter Settings
  - Physical Synthesis Optimizations
- Timing Analysis Settings
  - TimeQuest Timing Analyzer
  - Classic Timing Analyzer Settings
    - Classic Timing Analyzer Repor
- Assembler
- Design Assistant
- SignalTap II Logic Analyzer
- Logic Analyzer Interface
- Simulator Settings
  - Simulation Verification
  - Simulation Output Files
- PowerPlay Power Analyzer Settings

Simulator Settings

Select simulation options.

Simulation mode: Functional

Simulation input: rom2p.vwf

Simulation period

● Run simulation until all vector stimuli are used

○ End simulation at: ns

Glitch filtering options: Auto

Description:

Specifies the type of simulation to perform for the current Simulation focus.

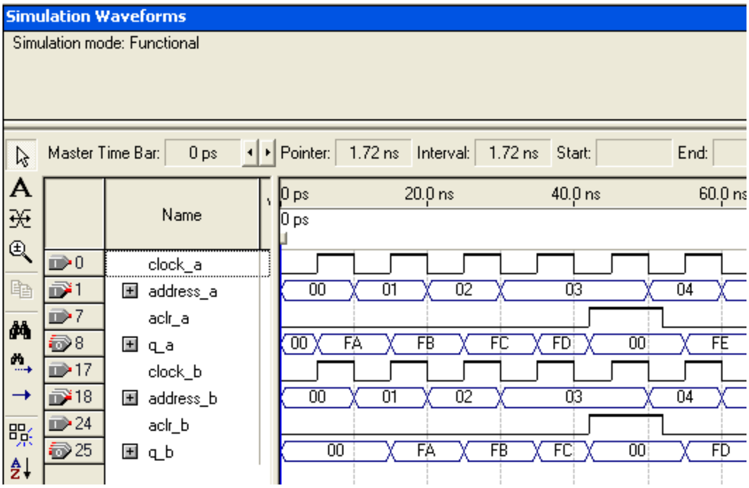
8. Click **OK**.

9. To run the simulation, on the **Processing** menu, click **Start Simulation,** or, on the toolbar, click the **Start Simulation** button.

10. When the **Simulator was successful** message box appears, click **OK**.

11. In the **Simulation Report** window, view the simulation waveforms to verify the results. Figure 2–25 shows the expected simulation results.

Figure 2–25. Functional Waveform for Dual-Port ROM Design Example



Simulation Waveforms

Simulation mode: Functional

Master Time Bar: 0 ps | Pointer: 1.72 ns | Interval: 1.72 ns | Start: | End:

Understanding the Simulation Results

In this example, you configured the ROM: 2-PORT to have the following properties:

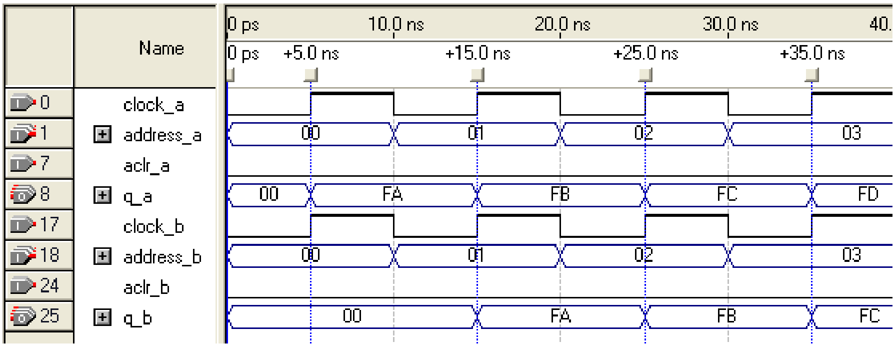
- Independent clock mode (separate clocks for port A and port B)
- Unregistered output port A and registered output port B
- Asynchronous clear on output latches and registers

The following section explains the simulation results corresponding to the configurations you set using the ROM: 2-PORT MegaWizard plug-in.

In this example, two independent clocks control the input and output of the two ports of the ROM. The input and output signals of port A, `address_a` and `q_a`, are synchronous with `clock_a`, and the input and output signals of port B, address_b and q_b, are synchronous with clock_b. The two clocks may operate at different frequencies, but in this example they operate at the same frequency.

Figure 2–26 shows the effects of read operations in a dual-port ROM with unregistered port A and registered port B.

Figure 2–26. Read Operations in Dual-Port ROM

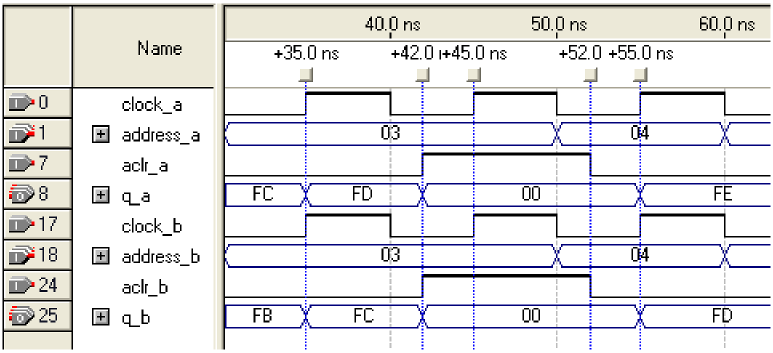


The memory addresses 00, 01, 02, 03, 04, and 05 hold data values 0xFA, 0xFB, 0xFC, 0xFD, 0xFE, and 0xFF, respectively, as set in the **rom.mif** file. Figure 2–26 shows the q_a port successively outputs read data 0xFA from address 00 (at 5 ns), data 0xFB from address 01 (at 15 ns), data 0xFC from address 02 (at 25 ns), and data 0xFD from address 03 (at 35 ns).

The read data appears immediately on output port q_a, before the next rising edge of clock_a, because q_a is unregistered. In contrast, on port B, the read data appears on output q_b on the next rising edge of clock_b, because q_b is registered.

Figure 2–27 illustrates the asynchronous clear on output latches and registers in a dual-port ROM. Each port has its own dedicated asynchronous clear signal (aclr_a or aclr_b). The asynchronous clear on output latches and registers is supported only in Stratix III and Cyclone III devices.

Figure 2–27. Asynchronous Clear on Output Latches and Registers in Dual-Port ROM



In this example, the behavior on output port `q_a` illustrates the effect of the asynchronous clear on an unregistered output latch, and the behavior on output port `q_b` illustrates its effect on a registered output port.

Figure 2–27 shows that both `q_a` and `q_b` clear immediately when `aclr_a` and `aclr_b` are asserted (at 42 ns). The output latches of `q_a` (an unregistered port) and `q_b` (a registered port) are affected by the asynchronous clear signal.

The outputs remain cleared while the asynchronous clear signal is high. In this example, the asynchronous clear signals are deasserted at 52 ns, but the outputs still show 00 until the next rising edge of the clock occurs at 55 ns.

At 55 ns, `q_a` outputs the data value 0xFE read from the current memory address, 04; in contrast, `q_b` outputs the data value 0xFD read from the previous memory address, 03.

Functional Results: Simulate the Single-Port ROM in ModelSim-Altera

Simulate the design in ModelSim to compare the results of both simulators. This User Guide assumes that you are familiar with using ModelSim-Altera before trying out the design example. If you are unfamiliar with ModelSim-Altera, refer to the support page for software products on the Altera website (**www.altera.com**). On the support page for ModelSim-Altera, there are various links to topics such as installation, usage, and troubleshooting.

Set up the ModelSim-Altera simulator by performing the following steps:

1. Unzip the **rom2p_ex2_msim.zip** files (except **readme.txt**) to any working directory on your PC.
2. Start ModelSim-Altera.
3. On the File menu, click **Change Directory**.
4. Select the folder in which you unzipped the files. Click **OK.**
5. On the Tools menu, select **Execute Macro**.
6. Select the **rom2p_msim.do** file and click **Open**.

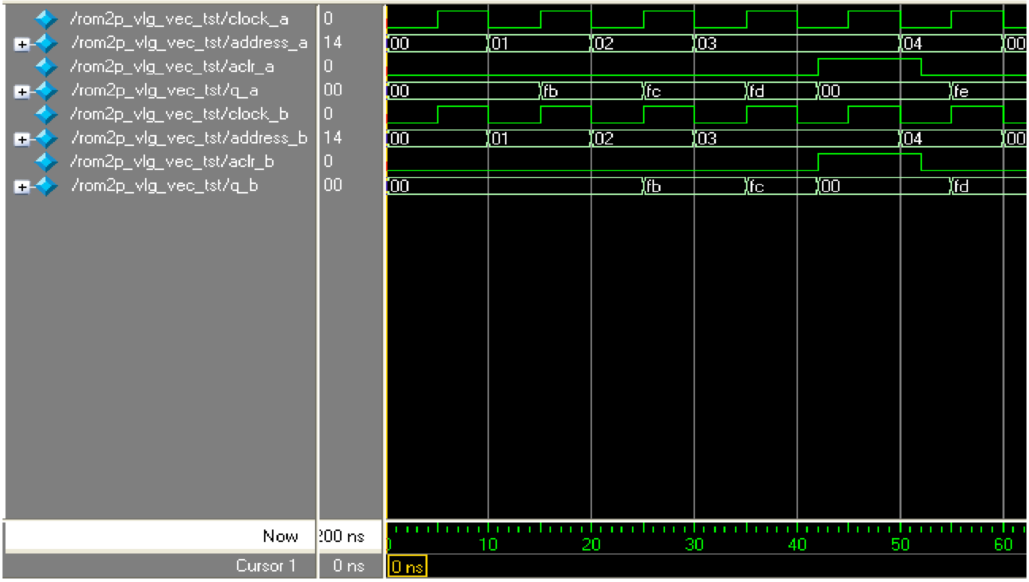
   The **rom2p_msim.do** file is a script file for ModelSim that automates all necessary settings for the simulation.

7. Verify the results shown in the Waveform Viewer window.

You can rearrange signals, remove signals, add signals, and change the radix by modifying the script in **rom2p_msim.do** accordingly to match the order of the results in the Quartus II simulator. Also, you may need to modify the script to specify your local path to the Stratix III library files.

Figure 2–28 shows the expected simulation results when running the design example in ModelSim-Altera. Compare the results obtained using the Quartus II simulator in Figure 2–25 on page 2–38. Refer to the section "Understanding the Simulation Results" on page 2–30 for an explanation of the simulation results.

Figure 2–28. ModelSim Simulation Waveform for Dual-Port ROM


/rom2p_vlg_vec_tst/clock_a
/rom2p_vlg_vec_tst/address_a
/rom2p_vlg_vec_tst/aclr_a
/rom2p_vlg_vec_tst/q_a
/rom2p_vlg_vec_tst/clock_b
/rom2p_vlg_vec_tst/address_b
/rom2p_vlg_vec_tst/aclr_b
/rom2p_vlg_vec_tst/q_b
0
14
0
00
0
14
0
00
00
01
02
03
04
00
00
fb
fc
fd
00
fe
00
01
02
03
04
00
00
fb
fc
00
fd
Now
200 ns
Cursor 1
0 ns
0
10
20
30
40
50
60
0 ns

Conclusion

The Quartus II software provides parameterizable ROM megafunctions through the ROM: 1-PORT and ROM: 2-PORT MegaWizard plug-ins that implement the ROM using the altsyncram or lpm_rom megafunction. With these megafunctions, you can easily configure your ROM design with other support features, such as different clocking modes, read enable, clock enable, address clock enable, asynchronous clear, and other features. The megafunction is performance-optimized for Altera devices and therefore, provides more efficient logic synthesis and device implementation, because it automates the coding process and saves valuable design time. In addition, it is easy to reconfigure the characteristics of your ROM design through the easy-to-use GUI. Altera recommends using these functions during design implementation so you can consistently meet your design goals.

Chapter 3. Specifications

Introduction

The Quartus® II software provides the altsyncram megafunction that supports single-port and dual-port ROM functionality. This chapter describes the ports and parameters of the altsyncram megafunction in ROM mode. Ports and parameters that only support write functionality are not supported in ROM mode and are not included in this chapter.

The parameter details are only relevant for users who bypass the MegaWizard Plug-In Manager interface and use the megafunction as a directly parameterized instantiation in their design. The details of these parameters are hidden from MegaWizard Plug-In Manager interface users.

Refer to the latest version of the Quartus II software Help for the most current information about the ports and parameters for this megafunction.

Ports and Parameters for the altsyncram Megafunction

Table 3–1 shows the input ports, Table 3–2 shows the output ports, and Table 3–3 shows the altsyncram megafunction parameters.

Table 3–1. altsyncram Megafunction Input Ports for ROM (Part 1 of 3)

Port Name	Required	Description	Comment
`wren_a`	No	Write enable input port	Not applicable for ROM.
`rden_a`	No	Read enable input port	This port is available for Stratix III and Cyclone III devices only in single-port ROM. Not available in MLAB mode.
`wren_b`	No	Write enable input port	Not applicable for ROM.
`rden_b`	No	Read enable input port	Not applicable for ROM. (Dual-port ROM does not support a read-enable signal.)
`data_a[]`	No	Data input port to the memory for port A	Not applicable for ROM.
`data_b[]`	No	Data input port to the memory for port B	Not applicable for ROM.

Table 3–1. altsyncram Megafunction Input Ports for ROM (Part 2 of 3)

Port Name	Required	Description	Comment
`address_a[]`	Yes	Address input to the memory for port A	This input port is `WIDTHAD_A` bits wide.
`address_b[]`	Yes	Address input to the memory for port B	This input port is `WIDTHAD_B` bits wide.
`clock0`	Yes	Clock input port to the ROM	This clock is used in single-clock or dual-clock mode. In dual-clock mode, it clocks all input registers in input/output clock mode, and all port A registers in independent clock mode.
`clock1`	No	Clock input port to the ROM	This clock is used in dual-clock mode only. It clocks all output registers in input/output clock mode, and all port B registers in independent clock mode.
`clocken0`	No	Clock enable for `clock0`	
`clocken1`	No	Clock enable for `clock1`	
`clocken2`	No	Additional clock enable for `clock0`	This port defaults to VCC. This port is available for Stratix III devices only, and is useful for reducing power usage.
`clocken3`	No	Additional clock enable for `clock1`	This port defaults to VCC. This port is available for Stratix III devices only, and is useful for reducing power usage.
`aclr0`	No	Asynchronous clear for single-port input registers or dual-port port A registers	This port is applicable only when the input port or port A is registered. Asynchronous clear is available on ROM output latches only in Stratix III and Cyclone III devices.
`aclr1`	No	Asynchronous clear for single-port output registers or dual-port port B registers	This port is applicable only when the output port or port B is registered. Asynchronous clear is available on ROM output latches only in Stratix III and Cyclone III devices.
`byteena_a[]`	No	Byte enable input port	Not applicable for ROM.
`byteena_b[]`	No	Byte enable input port	Not applicable for ROM.

Table 3–1. altsyncram Megafunction Input Ports for ROM (Part 3 of 3)

Port Name	Required	Description	Comment
addressstall_a	No	Address stall input for port A. The address_a register holds the previous address value while addressstall_a is held high.	For single-port ROM, this port is available in the Stratix and Cyclone series of devices. For dual-port ROM, this port is available in Arria™ GX, Stratix® III, Stratix II, Stratix II GX, Cyclone® III, Cyclone II, and HardCopy® II devices only.
addressstall_b	No	Address stall input for port B. The address_b register holds the previous address value while addressstall_b is held high.	This port is available in dual-port ROM only, and in Arria GX, Stratix III, Stratix II, Stratix II GX, Cyclone III, Cyclone II, and HardCopy II devices only.

Table 3–2 shows the output ports of the altsyncram megafunction.

Table 3–2. altsyncram Megafunction Output Ports for ROM

Port Name	Required	Description	Comment
q_a[]	Yes	Data output port from the memory	This output port is WIDTH_A bits wide. The q_a[] port is legal only when the OPERATION_MODE parameter is set to "ROM" or “BIDIR_DUAL_PORT".
q_b[]	No	Data output port from the memory	This output port is WIDTH_B bits wide. The q_b[] port is legal only when the OPERATION_MODE parameter is set to "BIDIR_DUAL_PORT"
eccstatus[]	No	Status output for the memory core clock enable (ECC)	Not applicable for ROM.

Table 3–3 shows the parameters of the altsyncram megafunction.

Table 3–3. altsyncram Megafunction Parameters for ROM (Part 1 of 6)

Name	Type	Required?	Comment
OPERATION_MODE	String	✓	Specifies the operation of the ROM. Values are "ROM" for single-port ROM and "BIDIR_DUAL_PORT" for dual-port ROM.
WIDTH_A	Integer	✓	Specifies the width of the q_a[] output port.
WIDTHAD_A	Integer	✓	Specifies the width of the address_a[] input port.
NUMWORDS_A	Integer	—	Number of words stored in memory. If omitted, the default is $2^{WIDTHAD\_A}$.
OUTDATA_REG_A	String	—	Specifies the clock for the q_a[] output port. Values are "CLOCK0", "CLOCK1", or "UNREGISTERED". If omitted, the default is "UNREGISTERED".
ADDRESS_ACLR_A	String	—	Specifies the asynchronous clear for the address_a[] port. Values are "CLEAR0" and "NONE". If omitted, the default is "NONE". For Arria GX, Stratix III, Stratix II, Stratix II GX, Cyclone III, Cyclone II, and HardCopy II devices in BIDIR_DUAL_PORT mode, the value must be set to "NONE". However, in Stratix III, Stratix, Stratix GX, HardCopy Stratix, Cyclone III, and Cyclone devices in ROM mode, a value of "CLEAR0" is available.
OUTDATA_ACLR_A	String	—	Specifies the asynchronous clear for the q_a[] output port. Values are "CLEAR0", "CLEAR1", or "NONE". If omitted, the default is "NONE". Specifies the asynchronous clear parameter for the output latch in Cyclone III and Stratix III devices when the OUTDAT_REG_A parameter is set to "UNREGISTERED".
INDATA_ACLR_A	String	—	Not applicable for ROM.
WRCONTROL_ACLR_A	String	—	Not applicable for ROM.
BYTEENA_ACLR_A	String	—	Not applicable for ROM.
WIDTH_BYTEENA_A	Integer	—	Not applicable for ROM.
WIDTH_B	Integer	—	Specifies the width of the q_b[] output port. When the OPERATION_MODE parameter is set to "BIDIR_DUAL_PORT" mode, the WIDTH_B parameter is required.
WIDTHAD_B	Integer	—	Specifies the width of the address_b[] input port.
NUMWORDS_B	Integer	—	Number of words stored in memory. If omitted, the default is $2^{WIDTHAD\_B}$.

Table 3–3. altsyncram Megafunction Parameters for ROM (Part 2 of 6)

Name	Type	Required?	Comment
RDCONTROL_REG_B	String	—	Not applicable for ROM. Dual-port ROM does not support the rden_b port.
ADDRESS_REG_B	String	—	Specifies the clock for the address_b[] port. Values are "CLOCK0" and "CLOCK1". If omitted, the default is "CLOCK1". This parameter is only applicable for dual-port ROM (OPERATION_MODE has value "BIDIR_DUAL_PORT").
INDATA_REG_B	String	—	Not applicable for ROM.
WRCONTROL_WRADDRESS_REG_B	String	—	Not applicable for ROM.
BYTEENA_REG_B	String	—	Not applicable for ROM.
OUTDATA_REG_B	String	—	Specifies the clock for the q_b[] port. Values are "CLOCK0", "CLOCK1", and "UNREGISTERED". If omitted, the default is "UNREGISTERED". This parameter is only applicable for dual-port ROM (OPERATION_MODE has value "BIDIR_DUAL_PORT").
OUTDATA_ACLR_B	String	—	Specifies the asynchronous clear for the q_b[] output port. Values are "CLEAR0", "CLEAR1", and "NONE". If omitted, the default is "NONE". Specifies the asynchronous clear parameter for the output latch in Stratix III and Cyclone III devices when the OUTDATA_REG_B parameter is set to "UNREGISTERED".
RDCONTROL_ACLR_B	String	—	Specifies the asynchronous clear for the rden_b input port. Not applicable for ROM. Dual-port ROM does not support the rden_b port.
INDATA_ACLR_B	String	—	Not applicable for ROM.
WRCONTROL_ACLR_B	String	—	Not applicable for ROM.
ADDRESS_ACLR_B	String	—	Specifies the asynchronous clear for the address_b[] port. Values are "CLEAR0" and "NONE". If omitted, the default is "NONE". For Arria GX, Stratix III, Stratix II, Stratix II GX, Cyclone III, Cyclone II, and HardCopy II devices, the value must be set to "NONE".
BYTEENA_ACLR_B	String	—	Not applicable for ROM.
WIDTH_BYTEENA_B	Integer	—	Not applicable for ROM.
BYTE_SIZE	Integer	—	Not applicable for ROM.
READ_DURING_WRITE_MODE_MIXED_PORTS	String	—	Not applicable for ROM.

Table 3–3. altsyncram Megafunction Parameters for ROM (Part 3 of 6)

Name	Type	Required?	Comment
RAM_BLOCK_TYPE	String	—	Specifies the memory block type.The values available depend on the device you select. Values are "M-RAM", "M4K", "M512K", "M9K", "M144K", "MLAB", and "AUTO". If omitted, the default is "AUTO".
INIT_FILE	String	—	Name of the Memory Initialization File (**.mif**) or Hexadecimal (Intel-Format) Output File (**.hex**) containing ROM initialization data ("<*file name*>"), or "UNUSED". You must specify a file name for ROM.
INIT_FILE_LAYOUT	String	—	Specifies the layout port used with the initialization file. For single-port ROM, the legal value is "PORT_A". For dual-port ROM, "PORT_B" is allowed when port A and port B have different widths.

Table 3–3. altsyncram Megafunction Parameters for ROM (Part 4 of 6)

<table>
<tr><th>Name</th><th>Type</th><th>Required?</th><th>Comment</th></tr>
<tr><td>MAXIMUM_DEPTH</td><td>Integer</td><td>—</td><td>Specifies the maximum segmented value of the ROM. The MAXIMUM_DEPTH parameter value depends on the RAM_BLOCK_TYPE parameter. If omitted, the default is 0. You can also use the MAXIMUM_DEPTH parameter to save power in Stratix devices; however, this parameter may increase the number of logic elements (LEs) and affect design performance. For example, the following table shows power usage settings for a 4K × 36 (M4K RAM block) design of a Stratix II EP2S15 device.
<table>
<tr><th>M4K Slice Type</th><th>Dynamic Power (mW)</th><th>ALUT Usage</th><th>M4Ks</th></tr>
<tr><td>4K × 1 (default)</td><td>136.93</td><td>0</td><td>36</td></tr>
<tr><td>2K × 2</td><td>100.15 (73%)</td><td>40</td><td>36</td></tr>
<tr><td>1K × 4</td><td>74.56 (55%)</td><td>62</td><td>36</td></tr>
<tr><td>512 × 9</td><td>58.87 (43%)</td><td>143</td><td>32</td></tr>
<tr><td>256 × 18</td><td>54.23 (40%)</td><td>302</td><td>32</td></tr>
<tr><td>128 × 36</td><td>58.42 (43%)</td><td>633</td><td>32</td></tr>
</table>
As the RAM is sliced shallower, the dynamic power usage decreases. For a 128-deep ROM block, the power used by the extra LEs starts to outweigh the power gain achieved by shallower slices. This is a sample case and the power usage and ROM/LE usage depends on the ROM configuration, parameter/port usage, and input vectors. You can also use the MAXIMUM_DEPTH parameter to reduce the total number of memory blocks used in Stratix devices (at the expense of LEs). This applies to RAMs that have widths that are multiples of 9.

For example, in the above table, the 4K x 36 ROM uses 36 M4K ROM blocks with a default slicing of 4K × 1 slices. By setting the MAXIMUM_DEPTH to 512, the 4K × 36 ROM can fit into 32 M4K blocks.</td></tr>
<tr><td>CLOCK_ENABLE_INPUT_A</td><td>String</td><td>—</td><td>Specifies the clock enable for all port A inputs. This parameter is available only in Arria GX, Stratix III, Stratix II, Stratix II GX, Cyclone III, Cyclone II, and HardCopy II devices. For Cyclone III and Stratix III devices, values are "NORMAL", "BYPASS", and "ALTERNATE". If omitted, the default is "NORMAL". For other supported devices, values are "NORMAL" and "BYPASS".</td></tr>
</table>

Table 3–3. altsyncram Megafunction Parameters for ROM (Part 5 of 6)

Name	Type	Required?	Comment
CLOCK_ENABLE_OUTPUT_A	String	—	Specifies the clock enable for the q_a[] output port. Values are "NORMAL" and "BYPASS". This parameter is available in Arria GX, Stratix III, Stratix II, Stratix II GX, Cyclone III, Cyclone II, and HardCopy II devices.
CLOCK_ENABLE_CORE_A	String	—	Specifies the clock enable for the core of port A. Values are “USE_INPUT_CLKEN”, "NORMAL", “ALTERNATE”, and "BYPASS". “NORMAL” and “USE_INPUT_CLKEN” are legal values for Stratix and Cyclone devices. In Arria GX, Stratix II, and Cyclone II devices, the default value is “USE_INPUT_CLKEN”. “ALTERNATE” is a legal value only in Stratix III and Cyclone III devices.
READ_DURING_WRITE_MODE_PORT_A	String	—	Not applicable for ROM.
CLOCK_ENABLE_INPUT_B	String	—	Specifies the clock enable for all port B inputs. For Stratix III and Cyclone III devices, values are "NORMAL", "BYPASS", and "ALTERNATE". If omitted, the default is "NORMAL". This parameter is available in Arria GX, Stratix III, Stratix II, Stratix II GX, Cyclone III, Cyclone II, and HardCopy II devices. For other supported devices, values are "NORMAL" and “BYPASS”. .
CLOCK_ENABLE_OUTPUT_B	String	—	Specifies the clock enable for the q_b[] output port. Values are "NORMAL" and "BYPASS". This parameter is available in Arria GX, Stratix III, Stratix II, Stratix II GX, Cyclone III, Cyclone II, and HardCopy II devices.
CLOCK_ENABLE_CORE_B	String	—	Specifies the clock enable for the core of port B. Values are “USE_INPUT_CLKEN”, "NORMAL", “ALTERNATE“, and "BYPASS". “NORMAL” and “USE_INPUT_CLKEN“ are legal values for Stratix and Cyclone devices. In Arria GX, Stratix II, and Cyclone II devices, the default value is “USE_INPUT_CLKEN“. “ALTERNATE” is a legal value only in Stratix III and Cyclone III devices.
READ_DURING_WRITE_MODE_PORT_B	String	—	Not applicable for ROM.
ENABLE_ECC	String	—	Not applicable for ROM.
LPM_HINT	String	—	Specifies Altera-specific parameters in VHDL Design Files. The default is UNUSED.

Table 3–3. altsyncram Megafunction Parameters for ROM (Part 6 of 6)

Name	Type	Required?	Comment
`LPM_TYPE`	String	—	Identifies the library of parameterized modules (LPM) entity name in VHDL Design Files.
`INTENDED_DEVICE_FAMILY`	String	—	This parameter is used for modeling and behavioral simulation purposes. Create the altsyncram megafunction with the MegaWizard Plug-In Manager to calculate the value for this parameter.