

## Introduction

TriMatrix™ memory resources in the Stratix™ and Stratix GX architecture provide a diverse and powerful set of memory functions that address the memory requirements of today's system-on-a-programmable-chip (SOPC) designs. The Altera® Quartus® II design software includes sophisticated tools that fully utilize the advantages of TriMatrix memory, while maintaining simple, easy-to-use procedures that accelerate the design process. The Quartus II software also provides mechanisms for automatic or manual conversion of memory functions to target the Stratix and Stratix GX architecture.

This application note describes the basic and advanced features in the Quartus II software that enable you to take full advantage of the TriMatrix memory in Stratix and Stratix GX devices. Using the information in this document requires a working knowledge of the Quartus II design software.



The information in this application note refers to the implementation of TriMatrix memory in the Quartus II software version 2.0 and higher. If you are using a different release of the software, refer to the Quartus II Help or the Altera web site (<http://www.altera.com>) to obtain the latest information.



For an introduction to the Quartus II software, refer to the Quartus II Tutorial (Help menu).

## Using TriMatrix Memory in the Quartus II Software

The Quartus II software supports a simple design methodology for instantiating memory functions in Stratix and Stratix GX designs. Using the MegaWizard® Plug-In Manager, you can specify in what type of TriMatrix block the compiler should implement memory functions in one of two ways:

- You can manually select the TriMatrix block type.
- The Quartus II software automatically selects the appropriate TriMatrix memory block type based on functionality and size.

Allowing the Quartus II software to automatically choose an implementation at compile time makes the design process fast, simple, and flexible. Specifying how you implement the memory before compilation provides more control over the actual hardware implementation of your design.

## Specifying a Type of TriMatrix Memory

Stratix and Stratix GX TriMatrix memory consists of three types of blocks: M512, M4K, and M-RAM blocks. The three TriMatrix memory block types provide you with the resources to efficiently address memory needs for your SOPC design. Each of the TriMatrix memory block types is well suited to a variety of memory functions. The most apparent difference between the M512, M4K, and M-RAM blocks is size (see [Figure 1 on page 4](#)).

[Table 1](#) summarizes the features supported by the three sizes of TriMatrix memory.

Feature	M512 Block	M4K Block	M-RAM Block
Performance	312 MHz	312 MHz	300 MHz
Total RAM bits (including parity bits)	576	4,608	589,824
Configurations	512 × 1 256 × 2 128 × 4 64 × 8 64 × 9 32 × 16 32 × 18	4K × 1 2K × 2 1K × 4 512 × 8 512 × 9 256 × 16 256 × 18 128 × 32 128 × 36	64K × 8 64K × 9 32K × 16 32K × 18 16K × 32 16K × 36 8K × 64 8K × 72 4K × 128 4K × 144
Parity bits	✓	✓	✓
Byte enable		✓	✓
Single-port memory	✓	✓	✓
Simple dual-port memory	✓	✓	✓
True dual-port memory		✓	✓
Embedded shift register	✓	✓	
ROM	✓	✓	
FIFO buffer	✓	✓	✓
Simple dual-port mixed width support	✓	✓	✓
True dual-port mixed width support		✓	✓
Memory initialization (.mif)	✓	✓	
Mixed-clock mode	✓	✓	✓
Power-up condition	Outputs cleared	Outputs cleared	Outputs unknown

Table 1. Summary of TriMatrix Memory Features (Part 2 of 2)

Feature	M512 Block	M4K Block	M-RAM Block
Register clears	Input and output registers (1)	Input and output registers (2)	Output registers
Same-port read-during-write	New data available at positive clock edge	New data available at positive clock edge	New data available at positive clock edge
Mixed-port read-during-write	Outputs set to unknown or old data	Outputs set to unknown or old data	Unknown output

**Notes to Table 1:**

- (1) The `rden` register on the M512 memory block does not have a clear port.
- (2) On the M4K block, asserting the clear port of the `rden` and byte enable registers drives the output of these registers high.

*M512 Blocks*

The M512 blocks incorporate high-performance dedicated shift register circuitry, and are suited for small, high-performance memory functions. Stratix and Stratix GX devices have a large number of M512 blocks, enabling efficient implementation of many separate memory functions.

*M4K Blocks*

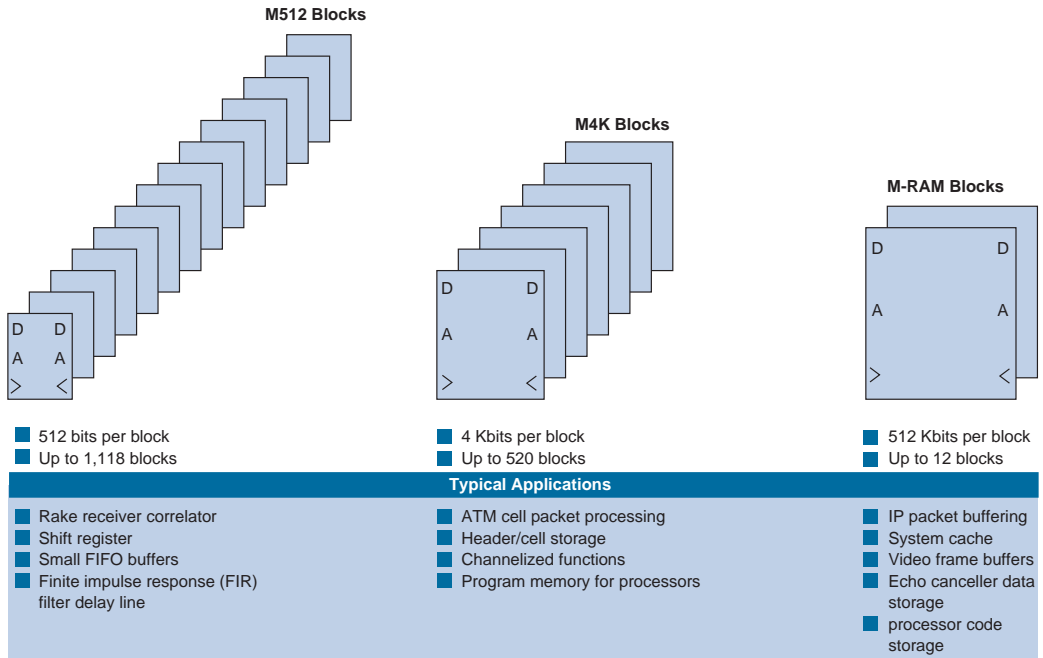
M4K blocks provide a larger capacity and wide configuration options, making them useful for storing larger data blocks. The M4K blocks also have dedicated high-performance shift register circuitry. They are also useful in conjunction with Stratix and Stratix GX digital signal processing (DSP) blocks, where you can use them to store data used in multiply-accumulate functions.

*M-RAM Blocks*

The high-density M-RAM blocks offer an ideal high-speed memory solution for an on-chip microprocessor, such as Altera's Nios™ embedded processor. M-RAM blocks are also useful in many applications requiring large amounts of data storage, such as payload storage of packetized data while packet headers are processed in networking applications.

Figure 1 shows the M512, M4K, and M-RAM memory blocks.

Figure 1. M512, M4K & M-RAM Block Sizes



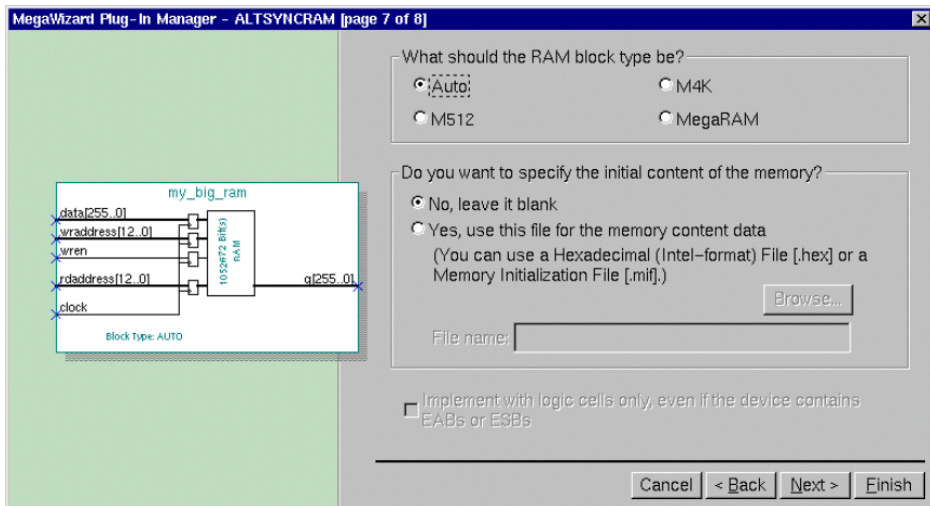
## Using the MegaWizard Plug-In Manager to Define Memory Functions

The MegaWizard Plug-In Manager is a graphical user interface that enables you to quickly and easily specify the parameters of Altera-specific functions. For Stratix and Stratix GX designs, it enables the parameterization of memory functions. For the Stratix and Stratix GX architecture, the `altsyncram` megafunction supports all Stratix and Stratix GX single- and multi-port RAM and ROM memory functions. For first-in first-out (FIFO) functions, use the `lpm_fifo` or `lpm_fifo+` megafunctions. The MegaWizard Plug-In Manager generates a design module that instantiates the `altsyncram` megafunction with the specified parameters. You can then instantiate this module in your design files.

You can use the `altsyncram` megafunction for single- and multi-port RAM and ROM Stratix and Stratix GX TriMatrix memory functions. However, you can use other MegaWizard Plug-Ins to generate the parameterized `altsyncram` megafunction instantiation. For example, the `lpm_fifo`, `lpm_ram_dp+`, and `lpm_rom` MegaWizard Plug-Ins are capable of producing memory functions that target a Stratix device or other Altera devices. Using the `lpm_rom` MegaWizard Plug-In, for example, to implement a ROM function in a Stratix device produces an `altsyncram` megafunction instance with `OPERATION_MODE = ROM`. Using the same MegaWizard Plug-In to implement a ROM function in another Altera device generates a `lpm_rom` megafunction instance.

When using the `altsyncram` megafunction, you are prompted to select the type of TriMatrix memory block to use to implement the memory function. Choosing **Auto** allows the Quartus II software to choose the TriMatrix block type during compilation. Alternately, you can specify the TriMatrix block type by selecting **M512**, **M4K**, or **M-RAM** in the MegaWizard Plug-In Manager. Allowing the Quartus II software to choose the memory type gives the compiler the flexibility to place the memory function in any available memory resource. [Figure 2](#) illustrates how you can select a TriMatrix block type or use the **Auto** setting.

Figure 2. TriMatrix Memory Block-Type Selection Using the `altsyncram` MegaWizard Plug-In Manager





Choosing a TriMatrix memory resource, or instructing the Quartus II software to do so in the MegaWizard Plug-In Manager, determines the value of the `RAM_BLOCK_TYPE` parameter in the resulting `altsyncram` instantiation. If you manually instantiate an `altsyncram` function, the settings for the `RAM_BLOCK_TYPE` parameter are **Auto**, **M512**, **M4K**, or **M-RAM**. If you are using the MegaWizard Plug-In Manager to instantiate `altsyncram`, it automatically sets this parameter.

### How Quartus II Compiles Memory

Feature differences between the various TriMatrix memory block types may restrict memory functions with specific functional requirements to certain TriMatrix memory resources. For example, because the M-RAM blocks do not support an initialized state at power up, selecting a MIF in the MegaWizard Plug-In Manager restricts the resulting memory function that is implemented to either a M512 or M4K block.

When designing functions that are intended to use specific TriMatrix memory resources, you must understand the functional differences among the different types of TriMatrix memory blocks.



For details on the functionality of each TriMatrix memory block, refer to [AN 203: Using TriMatrix Embedded Memory Blocks in Stratix & Stratix GX Devices](#), the [Stratix Programmable Logic Device Family Data Sheet](#), and the [Stratix GX FPGA Family Data Sheet](#).

#### *Intelligent TriMatrix Block Type Selection*

During compilation, the Quartus II software selects the appropriate TriMatrix memory resource for each memory function in the design. The MegaWizard Plug-in Manager restricts the TriMatrix memory block type choices to allow only valid combinations of functionality and block type. Similarly, if you do not specify a block type (e.g., `RAM_BLOCK_TYPE = AUTO`), the Quartus II Compiler automatically eliminates invalid block types.



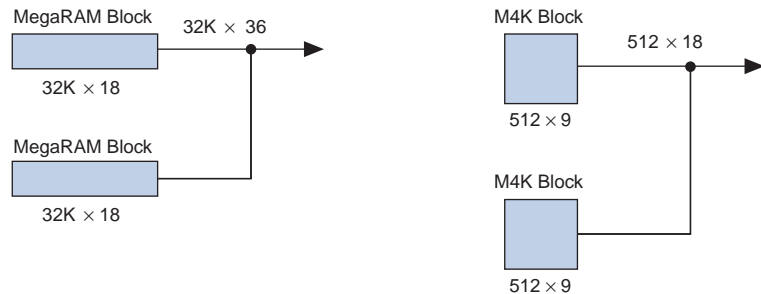
If you manually instantiate an `ALTSYCNRAM` megafunction with functionality that is not supported in the specified block type (e.g., an initialized memory function using a M-RAM block), the compiler generates an error message and compilation will not succeed.

Based on the size and configuration of a memory function, the Quartus II software attempts to implement the function in a single TriMatrix memory block. The Quartus II software first attempts to implement each memory function in the smallest memory block type that meets the size and functional requirements.

### *Memory Implementation, Flexibility & Efficiency*

If a memory function does not fit in a single memory block, the Quartus II software partitions the memory function into multiple pieces, implementing it in more than one TriMatrix memory block. The Quartus II software implements memory functions in TriMatrix memory blocks, partitioning a memory function among multiple blocks as necessary, to successfully fit all of the memory functions of a design into the targeted device. For example, if the requirement for a ROM function is  $512 \times 18$ , the Quartus II software can implement it using two M4K blocks. Similarly, the Quartus II software can implement a  $32K \times 36$  RAM function by combining two M-RAM blocks, each configured in  $32K \times 18$  mode. See [Figure 3](#).

**Figure 3. Examples of Combining TriMatrix Memory Blocks**



Additionally, the M4K blocks are capable of implementing multiple memory functions in a single M4K block for increased memory resource utilization.

## Advanced Features

For most designs, setting the memory function's block type to `AUTO` and allowing the Quartus II software to choose what block type and location to use produces acceptable results. However, for design optimization it is useful for you to instruct the compiler to use a specific TriMatrix memory block type and location before compilation. The Quartus II software provides simple, easy-to-use mechanisms for delivering these instructions to the compiler.

## Making LogicLock & Location Assignments

To optimize performance, the Quartus II software implements memory functions in the appropriate physical memory structures (locations). For more direct control on the memory placement, you can make LogicLock™ region or location assignments to memory function instantiations.

LogicLock regions give you more flexibility in the types of location constraints that are made and also enable powerful design techniques such as an incremental design flow. Location assignments are less flexible, but are useful for final timing closure in some designs. Altera recommends using LogicLock regions for constraining memory implementations.



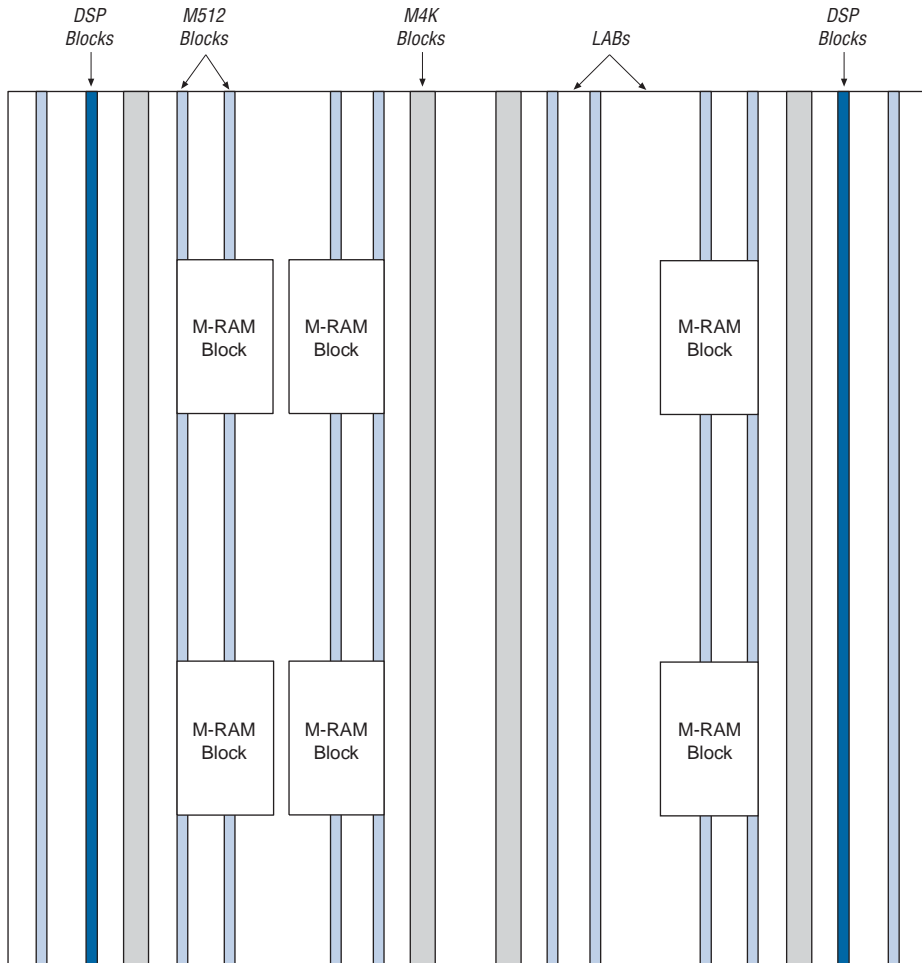
For more information on the LogicLock design methodology, refer to [\*AN 161: Using the LogicLock Methodology in the Quartus II Design Software\*](#).

M512 and M4k memory structures are organized in columns; M-RAM memory structures span multiple columns. [\*Figure 4\*](#) shows an example of the general layout of the EP1S60 device, including the relative positions of the M512, M4K, and M-RAM blocks.



For detailed descriptions about the physical layout of the Stratix or Stratix GX device family, refer to the [\*Stratix Programmable Logic Device Family Data Sheet\*](#) or the [\*Stratix GX FPGA Family Data Sheet\*](#), respectively.

Figure 4. EP1S60 Device



LogicLock region and location assignments use coordinate representations of the Stratix or Stratix GX device to restrict design entities to specific locations on the device. You can make location assignments directly in the **Current Assignments Floorplan** or through the **Assignment Organizer** by using a **Custom Region**. A LogicLock region or **Custom Region** should include the coordinate of the intended TriMatrix block for location assignments of memory functions.

TriMatrix memory block locations are represented by a single row/column coordinate, which signifies the bottom left-hand corner of the memory block. This coordinate represents the entire memory block that spans several other logical coordinates, because TriMatrix memory blocks are larger than logic array blocks (LABs) or logic cells. For example, the coordinate M512\_X4\_Y1 represents the M512 memory block located in column 4, row 1.



A LogicLock or custom region containing any portion of a M-RAM or M4K block contains the entire M-RAM or M4K block. The LogicLock or custom region does not necessarily need to include the bottom-left coordinate that is used to represent the block for location assignments.

For a visual representation of the physical layout of the device, use the Quartus II Floorplan Editor. For detailed information on location naming conventions and how to use location names in LogicLock region or location assignments, refer to the Quartus II Help.

## Migrating Non-Stratix Memory Functions to Stratix Devices

Because of the differences between the Stratix and Stratix GX TriMatrix memory architecture and the memory architectures of other Altera devices, conversion is necessary when migrating a design from another family's memory architecture to a Stratix or Stratix GX architecture. The Quartus II software provides tools that make this conversion easy.

### Using The MegaWizard Plug-In Manager for Memory Migration

For memory functions that were originally created for another architecture, you can migrate your existing functions to the Stratix or Stratix GX architecture by re-running the MegaWizard Plug-In Manager. The MegaWizard Plug-In Manager defaults to parameters that are compatible with the original function (if possible), simplifying the process. By using the MegaWizard Plug-in Manager to convert the memory function, you verify that Stratix or Stratix GX devices support all of the functionality of the original memory function.



For details on the differences between APEX™ and Stratix memories and complete instructions for converting APEX memory functions to Stratix memory functions, refer to [AN 206: Understanding the Differences Between Stratix and APEX Memory Architecture](#) and [AN 210: Converting Memory from Asynchronous to Synchronous for Stratix & Stratix GX Designs](#).

## Using Automatic Migration

If you attempt to compile a design targeting a Stratix or Stratix GX device that contains non-Stratix memory functions, the Quartus II software attempts to automatically implement these functions in Stratix or Stratix GX resources. This feature is useful for estimating the utilization and performance of a design that was originally targeted for another architecture.

In some cases, where the original functionality is fully supported by the TriMatrix memory architecture, the automatic conversion performed by the Quartus II software is sufficient for design migration. However, if a functional change is required, the Quartus II software generates an error or warning message and may require user intervention.

### *Performing a Fitting Evaluation on APEX Designs to Target Stratix or Stratix GX Devices*

Any memory function targeted to another Altera device architecture is automatically converted to a TriMatrix memory function for fitting-evaluation purposes if that memory function contains at least one clock. Because TriMatrix memory requires that you register all input signals, the Quartus II software modifies any memory functions with unregistered inputs so that all inputs are registered using an available clock. In this case, because the resulting memory is not functionally equivalent to the original memory, the Quartus II software issues a warning explaining that modifications to the memory function were made, and no programming file will be generated.



You must manually convert purely asynchronous memories (with no clock inputs) to target the Stratix or Stratix GX architecture. For more information, see [AN 210: Converting Memory from Asynchronous to Synchronous for Stratix & Stratix GX Designs](#).

You can use the results of a compilation where previously unregistered inputs are registered for fitting-evaluation purposes to estimate the likely utilization and performance of the Stratix or Stratix GX device design.



Fitting-evaluation results are estimates only. Consider that you can improve performance by pipelining memory functions. The design may also require some modifications to implement the memory function in TriMatrix memory blocks, which also affects the performance and utilization.

*Converting Memory Functions To Target TriMatrix Memory*

For synchronous memory functions with all input signals registered (or if only read-enable signals are unregistered), the Quartus II software can generate a compatible memory implementation in a Stratix device. The Quartus II software implements memory functions that are compatible with the original design and issues warnings about any differences between the functionality in the original implementation and the TriMatrix memory implementation. These differences are minor and will not affect most designs. Once you have confirmed that these functional differences do not affect your design, you can disable these warning messages by setting the `SUPPRESS_MEMORY_CONVERSION_WARNINGS` parameter to ON in the Assignment Organizer. You can apply this parameter to individual entities or for an entire project.

**Table 2** lists the functional differences that can affect a non-Stratix memory function implemented in Stratix and Stratix GX devices. Ensure that these functional differences will not adversely affect the functionality of your design.

Functionality	Affected in Memory Functions	Behavior in APEX Devices	Behavior in Stratix & Stratix GX Devices
Power-up conditions	All memory functions.	Memory powers up reading data at read address.	Memory powers up to the idle state. (1)
Mixed-port read-during-write (2)	Where the same memory location is written to on one port, and simultaneously read from a different port using the same clock for both.	Read port shows old data during the first half of the read cycle; read port shows new data on second half of the read cycle (after falling edge).	For M512 and M4K blocks, read port outputs old data; for M-RAM blocks, read port outputs unknown data.
Read enable	Where an unregistered Read-enable is used.	Read enable can be unregistered and can asynchronously enable the read port.	Read enable is registered and data is available on read ports after the next rising clock edge.

**Notes to Table 2:**

- (1) Registered outputs are cleared. Unregistered outputs on M512 and M4K blocks output zero. Unregistered outputs on M-RAM blocks output unknown values until a location that has been written to is read.
- (2) Using a single clock, this functionality results from the behavior of the `q_a` and `q_b` outputs when reading a memory location that is written from the other port.

## Conclusion

TriMatrix memory resources in the Stratix and Stratix GX architecture provide you with a diverse and powerful set of memory resources that address the memory requirements of today's SOPC designs. The Quartus II software gives you a complete set of tools to easily take advantage of the TriMatrix memory architecture of Stratix and Stratix GX devices.

## Revision History

The information contained in *AN 207: TriMatrix Memory Selection Using the Quartus II Software* version 2.1 supersedes information published in previous versions.

### Version 2.1

The following changes were made to *AN 207: TriMatrix Memory Selection Using the Quartus II Software* version 2.1:

- Updated text on [page 1](#).
- Changed figure on [page 4](#).
- Updated text on [page 6](#).
- Updated text on [page 8](#).
- Updated text on [page 11](#)
- Updates text on [page 12](#).



101 Innovation Drive  
San Jose, CA 95134  
(408) 544-7000  
<http://www.altera.com>  
[Applications Hotline:](#)  
(800) 800-EPLD  
[Literature Services:](#)  
[lit\\_req@altera.com](mailto:lit_req@altera.com)

Copyright © 2002 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

