

Introduction

With a high level of flexibility, performance, and programmability, you can use crosspoint switches in applications such as digital cross switching, telecommunications, and video broadcasting. Although there are off-the-shelf devices available to implement switches, Altera® MAX® II and MAX devices offer more flexibility to customize switches, meeting specific design goals with in-system programmability (ISP).

There are two reference designs described in this application note:

- 16 × 16 crosspoint switch
- Customized crosspoint switch

A 16 × 16 crosspoint switch is a non-blocking crosspoint switch, allowing you to independently connect each output to any input and any input to be connected to any output.

The customized crosspoint switch implements a 4-port bidirectional crosspoint switch. You can connect each bidirectional port to any other port. For example, if the port acts as an output, you can connect it to any one of the other ports that acts as an input. With this customized crosspoint switch, Altera MAX II and MAX devices offer more flexibility to meet specific goals rather than only implementing a standard non-blocking crosspoint switch.

16 × 16 Crosspoint Switch

Figure 1 shows the reference design for a 16 × 16 crosspoint switch architecture. This crosspoint switch is divided into three major categories: switch matrix, configuration, and address decoder.

Figure 1. 16 × 16 Crosspoint Switch Architecture Reference Design

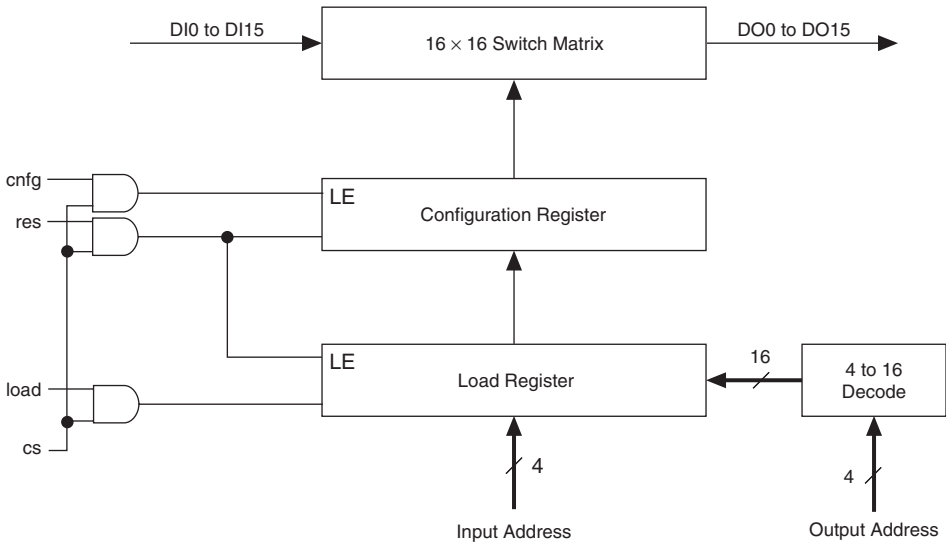


Figure 2 shows the hierarchy of design files that implement the 16 × 16 crosspoint switch.

Figure 2. 16 × 16 Crosspoint Switch Reference Design Hierarchy

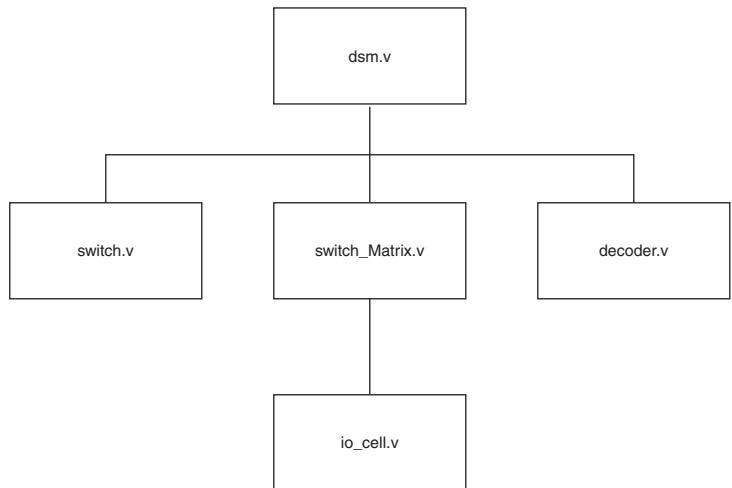


Table 1 lists all the functions of the input and output pins.

I/O Name	Function	Active High/Low
cnfg	Input signal to enable configuration	High
res	Input signal for resetting the switch matrix	High
load	Input signal to load the configuration data	High
cs	Input signal to select the chip	High
in_add	Input address that will be reconfigured	-
out_add	Output address that will be reconfigured	-
di	Input data to the switch matrix	-
do	Output data from the switch matrix	-

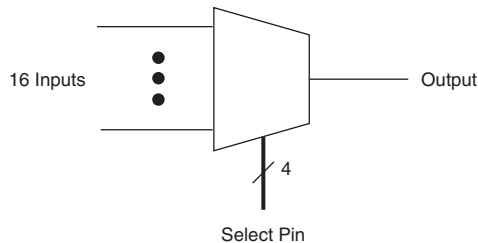
Switch Matrix

The switch matrix only has input, output, and multiplexer select pins. You can connect all 16 inputs to the multiplexer of each output, allowing you to independently connect any output to any input and any input to any or all outputs. Figure 3 shows the basic circuit of each output cell.



The names of the switch matrix reference design files are `switch_matrix.v` and `io_cell.v`.

Figure 3. Switch Matrix Circuit Output Cell



You can always add features to the output cell (e.g., adding another tri-state buffer or register at the output of the multiplexer).

Address Decoders

You can use an address decoder to decode the output address so that only one output cell is enabled for configuration. [Table 2](#) shows the output of the decoder and the enabled output cells for a 16 × 16 crosspoint switch.

Decoder Input [3..0]	Decoder Output [15..0]	Enabled Output Cells
0000	0000000000000001	Output 0
0001	0000000000000010	Output 1
0010	0000000000000100	Output 2
0011	0000000000001000	Output 3
0100	0000000000010000	Output 4
0101	0000000000100000	Output 5
0110	0000000001000000	Output 6
0111	0000000010000000	Output 7
1000	0000000100000000	Output 8
1001	0000001000000000	Output 9
1010	0000010000000000	Output 10
1011	0000100000000000	Output 11
1100	0001000000000000	Output 12
1101	0010000000000000	Output 13
1110	0100000000000000	Output 14
1111	1000000000000000	Output 15



The name of the address decoder reference design file is **decoder.v**. The Quartus® II software synthesizes the decoder, allowing you to use the product-term logic in the MAX 3000A device or in the look-up table (LUT) in the MAX II device to combine the decoding with other functions.

Configuration

Configuration is the main feature in a crosspoint switch. As shown in [Figure 1 on page 2](#), the configuration module consists of a double row register architecture, which allows reconfiguration of input to output connections during operation. Activation of the new configuration occurs with a single configuration pulse.



The name of the configuration reference design file is **switch.v**.

The switch matrix circuit is controlled by data in two sets of 16, 4-bit registers — the `LOAD REGISTERS` and `CONFIGURATION REGISTERS`. You can use the four bits of each register to store the input address that identifies the input that you can connect to a particular output. [Table 3](#) shows the connection of an input to a particular output when you select a different input address.

<i>Table 3. Input Connection to a Specific Output</i>	
Input Address (Multiplexer Select Pin) [3..0]	Output
0000	Input 0
0001	Input 1
0011	Input 2
0010	Input 3
0110	Input 4
0111	Input 5
0101	Input 6
0100	Input 7
1100	Input 8
1101	Input 9
1111	Input 10
1110	Input 11
1010	Input 12
1011	Input 13
1001	Input 14
1000	Input 15

You can select one of the 16, 4-bit registers in the first set of `LOAD REGISTERS` by placing a 4-bit word on the output address bus (see [Figure 1 on page 2](#)). You can place data that is written into the load register on the input address bus. The load register contains the 4-bit address of the input that connects to that output. The load register stores input data at the low-to-high transition of the `load` input pin with the chip-select (`cs`) signal set to high. The contents of the load registers are then transferred to the second set of `CONFIGURATION REGISTERS` at the low-to-high transition of the `cnfg` input signal with the `cs` signal set to high. This transition sets the state of the entire switch matrix to the chosen configuration.

Reset mode is also supported in this 16×16 crosspoint switch. When you assert the reset (`res`) signal (with `cs` set to high), the entire crosspoint switch is in its initial state where all outputs are connected to `input 0`.

The 16×16 crosspoint switch design can be targeted for MAX II (EPM570F256C3) or MAX 3000A (EPM3256ATC144) devices using the Quartus II software. The design utilization of MAX II and MAX 3000A devices is shown in Tables 4 and 5.

Table 4. 16×16 Crosspoint Switch EPM570 Utilization			
Resource	Available	Used	Utilization
Logic Cells	570	258	45%
Flipflops	570	128	22%
I/O pins	160	44	27%

Table 5. 16×16 Crosspoint Switch EPM3256A Utilization			
Resource	Available	Used	Utilization
Macrocells	256	192	75%
Flipflops	256	128	50%
I/O pins	116	44	38%
Shareable Expanders	256	0	0%
Parallel Expanders	240	48	20%

Crosspoint Switch Expansion

There are two ways of expanding the switch size: modification of the HDL source code or using more than one MAX or MAX II device.

Source Code Modification

You can use source-code modification to increase the size of the crosspoint switch. For example, a few modifications on the HDL source code of the reference design allows a 32×32 crosspoint switch implementation. Table 6 shows the modification required to increase crosspoint switch size for every Verilog HDL file from the reference design.

Table 6. Modification of Verilog HDL Files for a 32×32 Crosspoint Switch Reference Design (Part 1 of 2)	
File Name	Modification
<code>io_cell.v</code>	Increase the multiplexer input pins and the <code>selects</code> pins.
<code>switch_matrix.v</code>	Increase the number of inputs and outputs, and the <code>selects</code> pins of the switch matrix.

Table 6. Modification of Verilog HDL Files for a 32 × 32 Crosspoint Switch Reference Design (Part 2 of 2)

File Name	Modification
decoder.v	Increase the number of inputs and outputs of the decoder.
switch.v	Increase the <code>LOAD_REGISTERS</code> and <code>CONFIGURATION_REGISTERS</code> size.
dsm.v	Increase the number of input and output pins to the switch matrix.

Multiple Device Implementation

To implement multiple devices in hardware, you need an optional tri-state pin. This tri-state pin disables the output buffers so that you can maintain the number of outputs while the number of inputs is increased. [Figures 4](#) and [5](#) show how to expand the number of inputs and outputs to increase the switch matrix size using more than one MAX or MAX II device.



The switch matrix can be asymmetric when the number of inputs and outputs are different.

Figure 4. Input Port Expansion

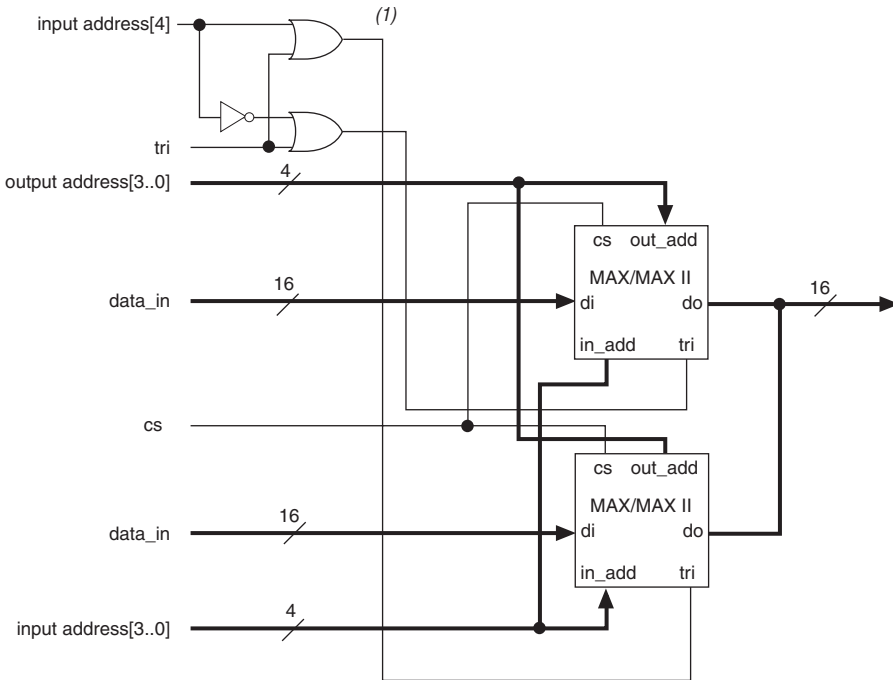
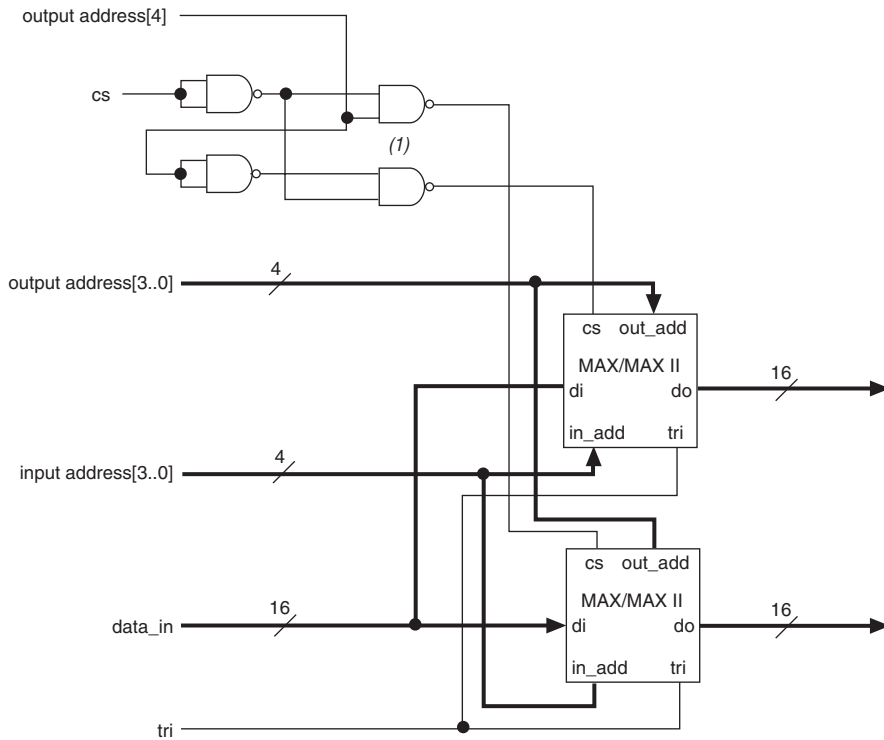
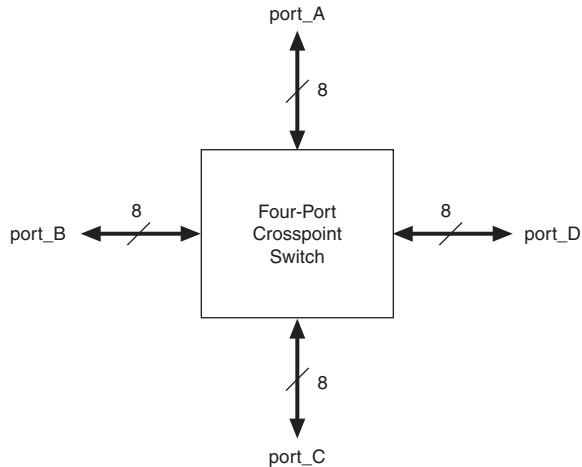


Figure 5. Output Port Expansion**Notes to Figures 4 & 5:**

(1) You can implement this design in either the EPM240, EPM3064A, or EPM3256A device.

Design Verification

You can achieve design verification for a 16×16 crosspoint switch by using the Quartus II software. MAX II (EPM570F256C3) and MAX 3000A (EPM3256ATC144-7) design verification occurs in both functional and timing simulations. Figure 6 shows the timing simulation of the 16×16 crosspoint switch.

Figure 7. Four-Port Crosspoint Switch Architecture

Customized Crosspoint Switch Configuration

To configure a 16×16 crosspoint switch to an 8-bit, 4-port crosspoint switch, make the following modifications:

1. Substitute the 16×16 crosspoint switch matrix with a 4-port crosspoint switch matrix.
2. Change the 4 to 16 decoder to a 2 to 4 decoder.
3. Add a register to `LOAD REGISTERS` and `CONFIGURATION REGISTERS` to store the tri-state bit.

The circuit used for configuration is same as the 16×16 crosspoint switch.

Switch Configuration

The process of switch configuration is exactly the same as a 16×16 crosspoint switch. As mentioned earlier, you can add a register to `LOAD REGISTERS` and `CONFIGURATION REGISTERS` to store the tri-state bit. This results in a total of three bits for a 4-port customized crosspoint switch: two bits for an input address and one bit for a tri-state. The two most significant bits (MSBs) in each register identify the input that connects to that output, and the least significant bit (LSB) controls whether the output is active or tri-state.

Since all ports are bidirectional, the reset mode for this customized design is different from the 16×16 crosspoint switch. When in reset mode, all the bidirectional ports are in an initialized state and all the ports are in tri-state mode.

The customized crosspoint switch design can be targeted to MAX II (EPM240T100C3) or MAX 3000A (EPM3064ATC100) devices using the Quartus II software. The design utilization in MAX II and MAX 3000A devices is shown in [Tables 7 and 8](#).

Table 7. Customized Crosspoint Switch EPM240 Utilization

Resource	Available	Used	Utilization
Logic Cells	240	86	35%
Flipflops	704	24	3%
I/O pins	80	44	55%

Table 8. Customized Crosspoint Switch EPM3064A Utilization

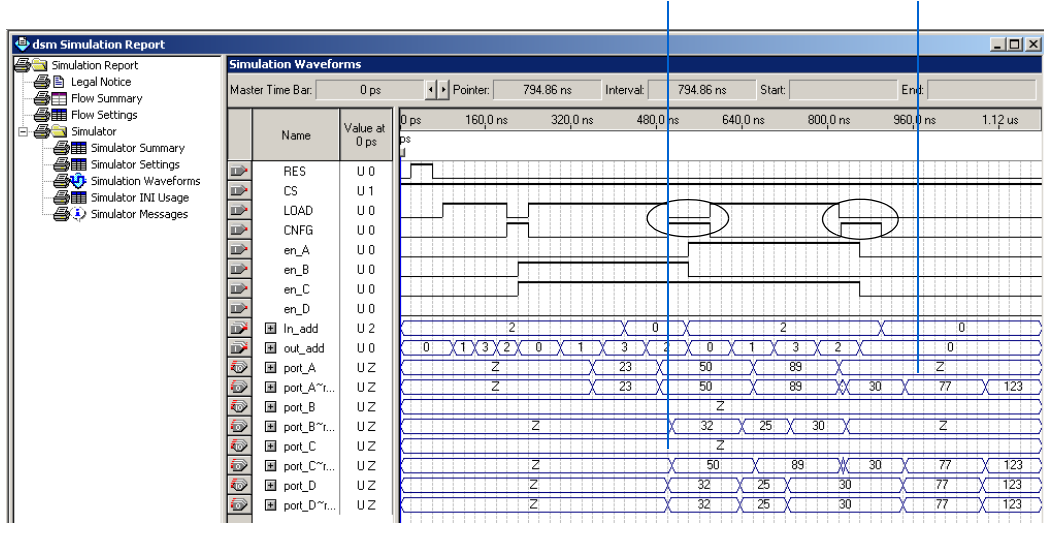
Resource	Available	Used	Utilization
Macrocells	64	56	87%
Flipflops	64	24	37%
I/O pins	66	44	66%
Shareable Expanders	64	0	0%
Parallel Expanders	60	0	0%

Design Verifications

The Quartus II software allows design verification of a 4-port customized crosspoint switch. MAX II (EPM240T100C3) and MAX 3000A (EPM3064ATC100-4) design verification occurs in both functional and timing simulations. [Figure 8](#) shows the timing simulation of the 4-port customized crosspoint switch.

Figure 8. Four-Port Crosspoint Switch Timing Simulations

The *cnfg* load signal creates new values on the multiplexer and new output values.



With all the I/O pins in tri-state mode, each port was initialized. After initialization, output configuration again occurs for port_B and port_C, while input configuration occurs for port_A and port_D. Input port_A is connected to port_C and input port_D is connected to port_B. In the last configuration, output port_A and port_C are connected to input port_D, while port_B is tri-stated.

Conclusion

This application note shows how to design a crosspoint switch using MAX or MAX II devices and provides two reference designs. You can design other customized crosspoint switches other than the examples provided.



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

Copyright © 2004 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.

