Crosspoint Switch Matrices in Altera MAX Series

2014.09.22

AN-294 Subscribe Send Feedback

You can use the Altera® MAX® 3000A, MAX II, MAX V, and MAX 10 devices to implement crosspoint switches. The supported Altera devices offer more flexibility to customize switches, meeting specific design goals with in-system programmability (ISP).

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.

Related Information

- **Design Example for MAX II and MAX 3000A: 16 x 16 Crosspoint Switch**
  Provides the design files for this application note (AN 294).
- **Design Example for MAX II and MAX 3000A: Customized Crosspoint Switch**
  Provides the design files for this application note (AN 294).
- **Design Example for MAX 10**
  Provides the MAX 10 design files for this application note (AN 294).

Crosspoint Switches Using Supported Altera Devices

16 × 16 Crosspoint Switch

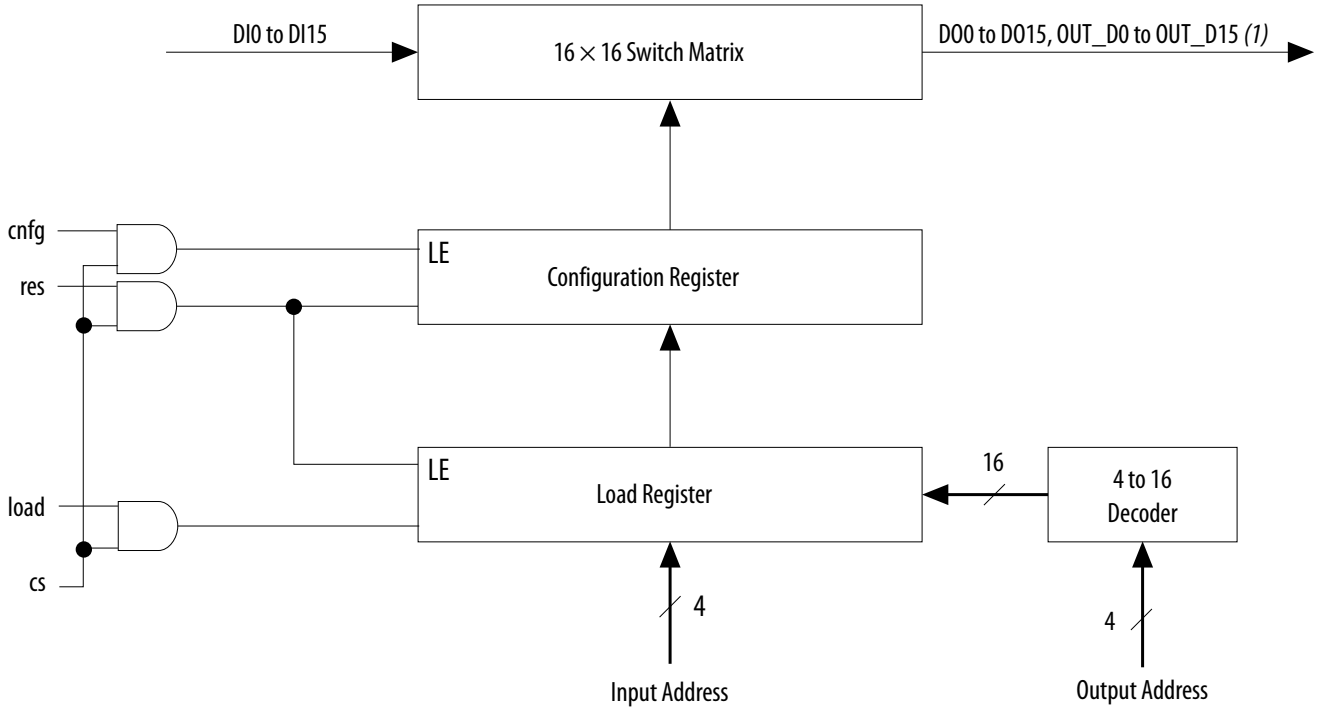
A 16 × 16 crosspoint switch is a non-blocking crosspoint switch. You can independently connect any output to any input and any input to any or all outputs.

This crosspoint switch is divided into three major blocks:

- Switch matrix
- Address decoder
- Configuration

16 × 16 Crosspoint Switch Architecture

Figure 1: 16 × 16 Crosspoint Switch Architecture Using Supported Altera Devices



Note:
(1) DO0 to DO15 is applicable to MAX II and MAX 3000A devices. OUT_D0 to OUT_D15 is applicable to MAX V and MAX 10 devices.

Signals

Table 1: Input and Output Signals for 16 × 16 Crosspoint Switch

Signal	Description	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 to be reconfigured.	—
out_add	Output address to be reconfigured.	—
di	Input data to the switch matrix.	—
do	Output data from the switch matrix. This is a dedicated signal name, cannot be used as a user defined singal name.This is applicable to MAX II and MAX 3000A devices.	—

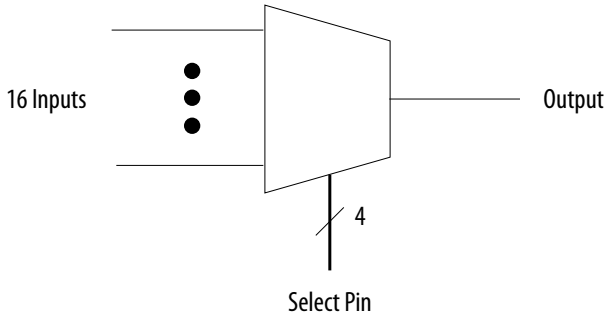
Signal	Description	Active High/Low
out_d	Output data from the switch matrix. This is applicable to MAX V and MAX 10 devices.	—

Switch Matrix

The switch matrix 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.

You can always add features to the output cell. For example, you can add another tri-state buffer or register at the output of the multiplexer.

Figure 2: Switch Matrix Circuit Output Cell



Address Decoder

You can use an address decoder to decode the output address so that only one output cell is enabled for configuration.

Table 2: Input and Output of a 4-Bit Decoder

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

Decoder Input [3..0]	Decoder Output [15..0]	Enabled Output Cells
1011	0000100000000000	Output 11
1100	0001000000000000	Output 12
1101	0010000000000000	Output 13
1110	0100000000000000	Output 14
1111	1000000000000000	Output 15

The Quartus® II software synthesizes the decoder. You can use the product-term logic in MAX 3000A device or in the look-up table (LUT) in MAX II, MAX V, and MAX 10 devices to combine the decoding with other functions.

Configuration

Configuration is the main feature in a crosspoint switch. 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 switch matrix circuit is controlled by data in two sets of 16, 4-bit registers —`LOAD REGISTERS` and `CONFIGURATION REGISTERS`. You can use four bits of each register to store the input address that identifies the input which 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.

Table 3: Input Connection to a Specific Output

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. 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 input0.

Crosspoint Switch Expansion

You can expand the switch size using the following methods:

- Source code modification
- Multiple device implementation

Source Code Modification

You can modify the HDL source code to increase the size of the crosspoint switch. For example, a few modifications on the HDL source code can expand the 16 × 16 crosspoint switch to a 32 × 32 crosspoint switch.

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.

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

Figure 3: Input Port Expansion

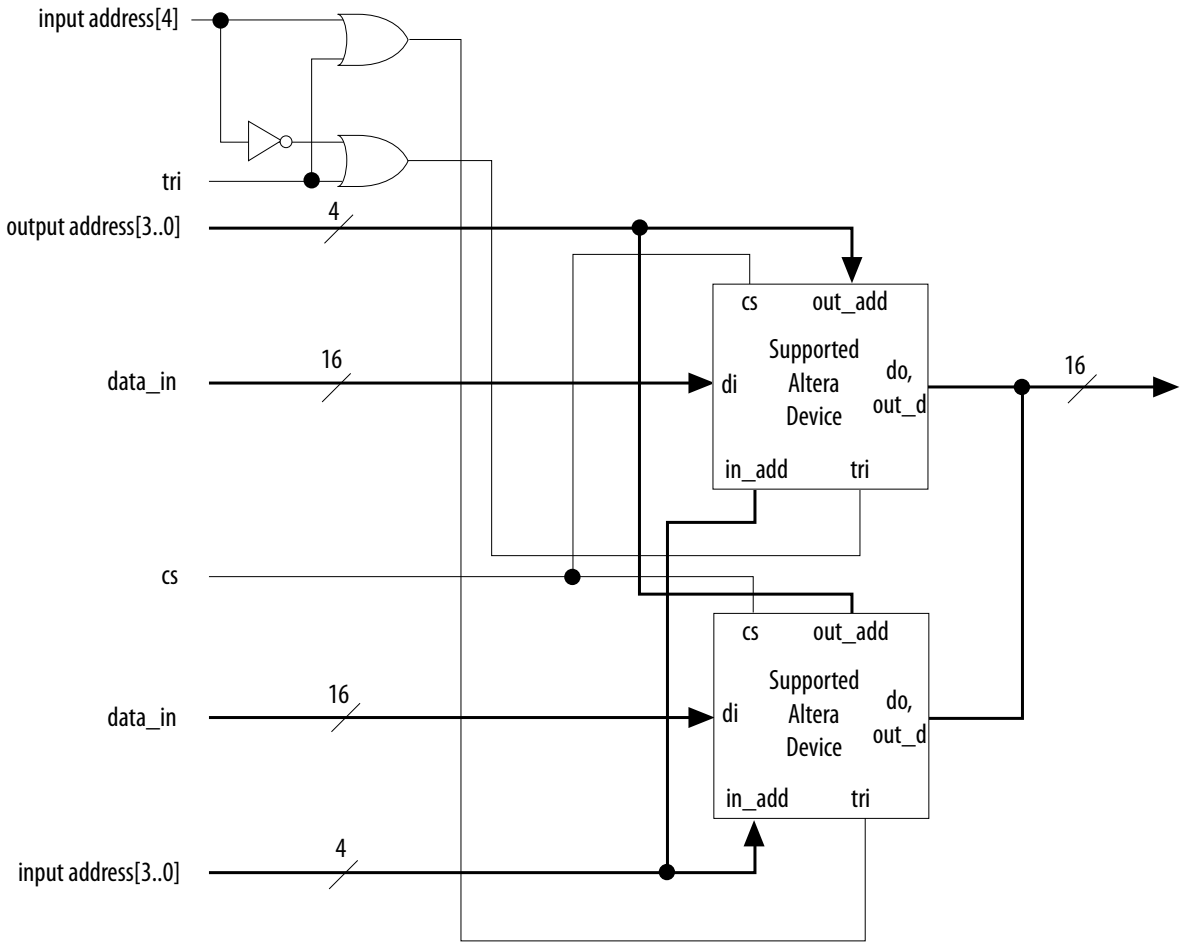
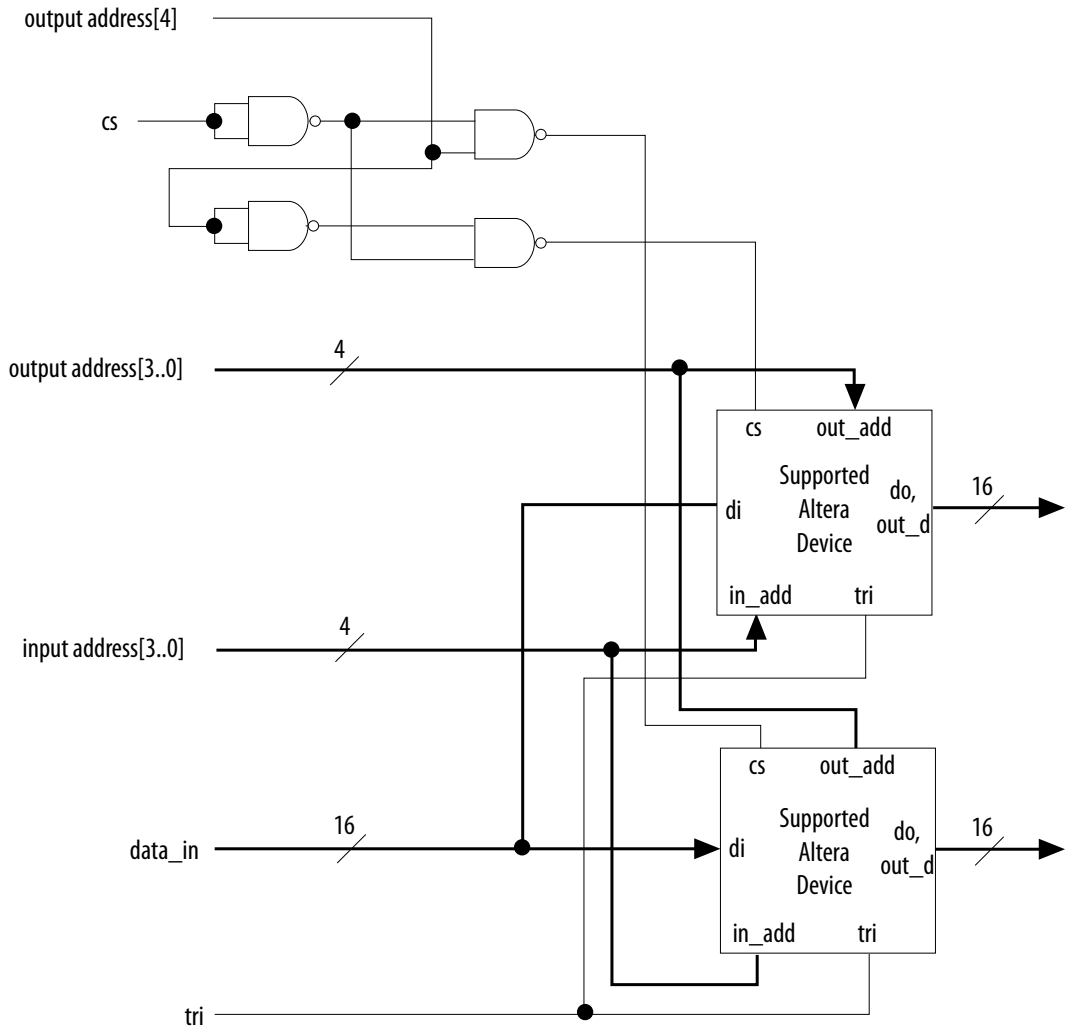


Figure 4: Output Port Expansion

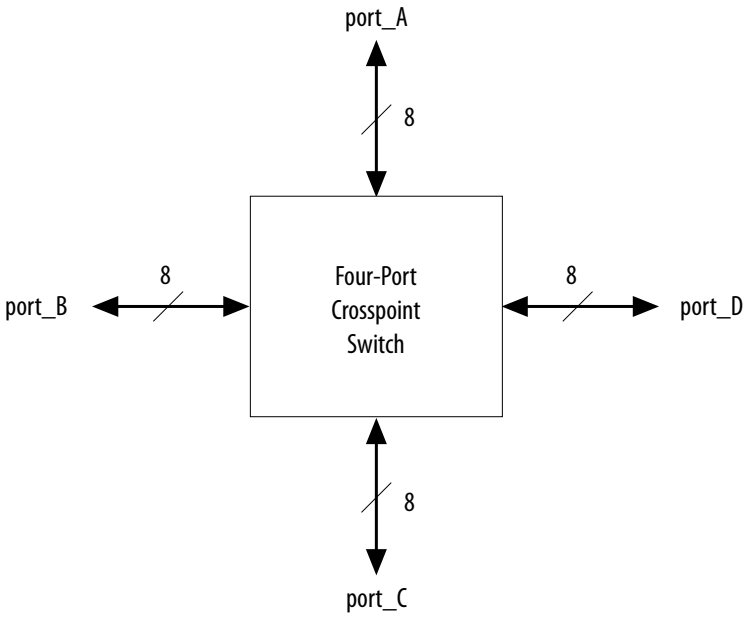


Customized Crosspoint Switch

The customized crosspoint switch implements a 4-port bidirectional crosspoint switch. Each port is eight bits wide and can be in tri-state mode, becoming an input port. You can connect each bidirectional port to any other port as an output or input. 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.

Customized Crosspoint Switch Architecture

Figure 5: 4-Port Crosspoint Switch Architecture



Configuration

The process of switch configuration is exactly the same as a 16 × 16 crosspoint switch. You can add a register to `LOAD REGISTERS` and `CONFIGURATION REGISTERS` to store the tri-sate 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.

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

Implementation

This application note describes two reference designs:

- 16 × 16 crosspoint switch
- Customized crosspoint switch

The detailed description of the implementation is based on the MAX 10, MAX II, and MAX 3000A devices. This application can also be implemented in MAX V devices.

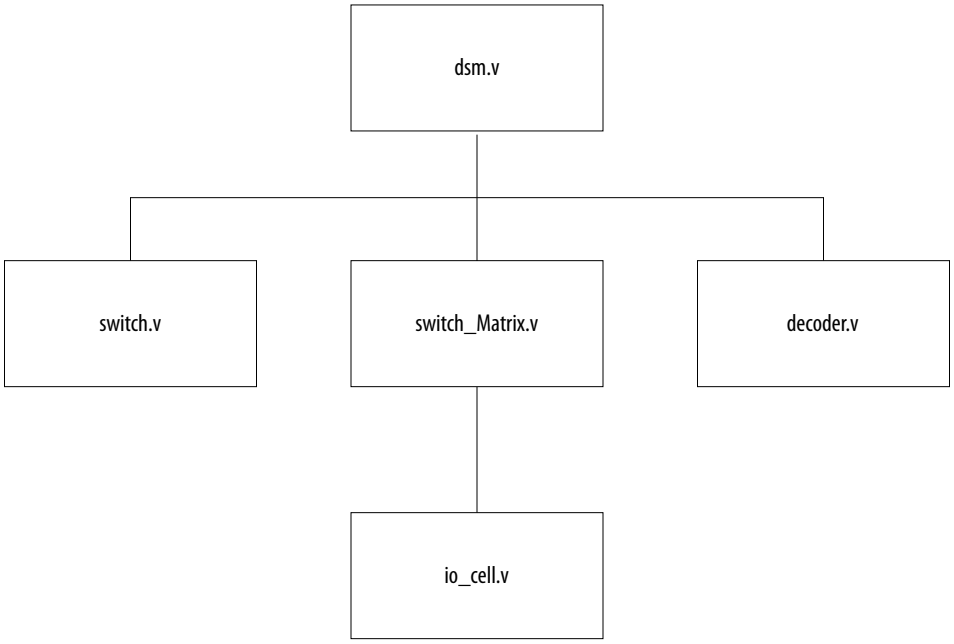
Design Example: 16 × 16 Crosspoint Switch

The 16 × 16 crosspoint switch design can be targeted for the following devices using the Quartus II software:

- MAX 10—10M08SAE144C8GES
- MAX II—EPM570F256C3
- MAX 3000A—EPM3256ATC144

Reference Design Files Hierarchy

Figure 6: 16 × 16 Crosspoint Switch Reference Design Files Hierarchy



Reference Design Files

Table 4: Reference Design Files for 16 × 16 Crosspoint Switch

16 × 16 Crosspoint Switch Block	Reference Design File
Switch matrix	**switch_matrix.v** and **io_cell.v**
Address decoder	**decoder.v**
Configuration	**switch.v**

Design Utilization

Table 5: 16 × 16 Crosspoint Switch Design Utilization for 10M08SAE144C8GES Device

Resource	Available	Used	Utilization
Logice Elements	8064	262	3%

Resource	Available	Used	Utilization
Registers	8064	128	3%
I/O pins	101	44	43%

Table 6: 16 × 16 Crosspoint Switch Design Utilization for EPM570F256C3 Device

Resource	Available	Used	Utilization
Logic Cells	570	258	45%
Flipflops	570	128	22%
I/O pins	160	44	27%

Table 7: 16 × 16 Crosspoint Switch Design Utilization for EPM3256ATC144 Device

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

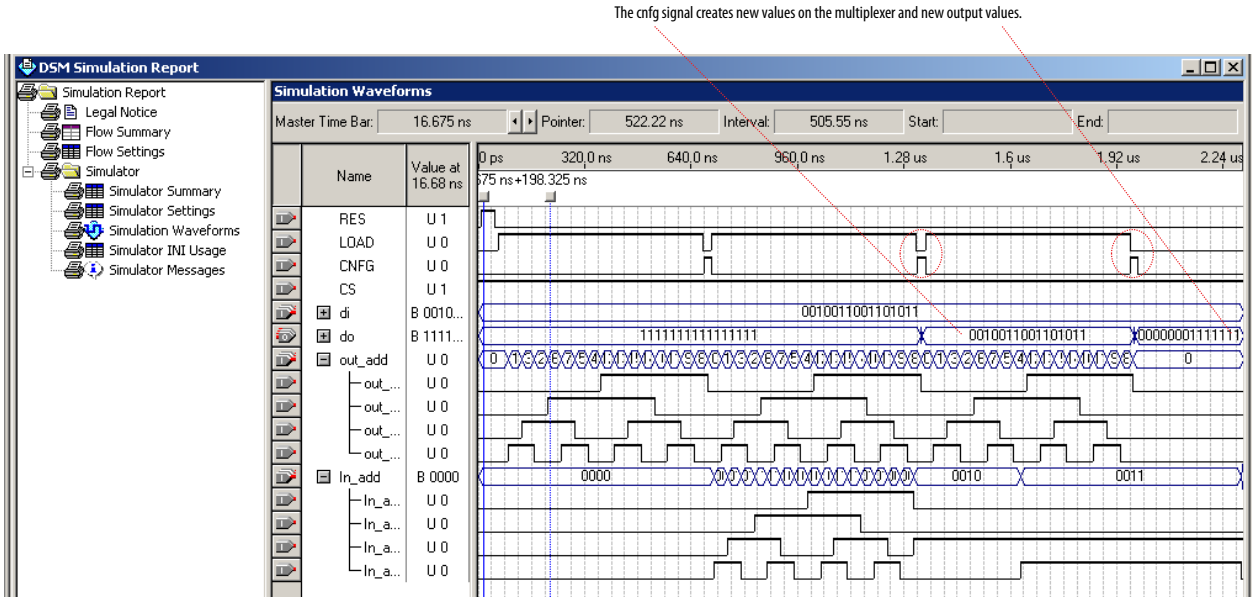
Table 8: Verilog HDL File Modification for a 32 × 32 Crosspoint Switch Reference Design

File Name	Modification
io_cell.v	Increase the multiplexer input pins and the `select` pins.
switch_matrix.v	Increase the number of inputs and outputs, and the `select` pins of the switch matrix.
decoder.v	Increase the number of inputs and outputs of the decoder.
switch.v	Increase the `LOAD REGISTERS` and `CONFIGURATION REGISTERS` size.
dsm.v	Increase the number of input and output pins to the switch matrix.

Design Verification

You can verify the design for a 16 × 16 crosspoint switch using the Quartus II software. The design verification occurs in both functional and timing simulations.

Figure 7: 16 × 16 Crosspoint Switch Timing Simulations



Initialization occurred for all outputs connected to `in_add[0]`. After the initialization, output configuration occurs resulting in the same input and output value (`0010011001101011`). In the final configuration, signals `out_add[15]` to `out_add[8]` are connected to `in_add[2]`; and signals `out_add[7]` to `out_add[0]` are connected to `in_add[3]` (input: `0010011001101011`, output: `1111111100000000`).

Design Example: Customized Crosspoint Switch

The customized crosspoint switch design can be targeted for the following devices using the Quartus II software:

- MAX 10—10M08SAE144C8GES
- MAX II—EPM240T100C3
- MAX 3000A—EPM3064ATC100

Cutomized 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.

Design Utilization

Table 9: Customized Crosspoint Switch Design Utilization for 10M08SAE144C8GES Device

Resource	Available	Used	Utilization
Logice Elements	8064	86	1%
Registers	8064	24	<1%
I/O pins	101	44	43%

Table 10: Customized Crosspoint Switch Design Utilization for EPM240T100C3 Device

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

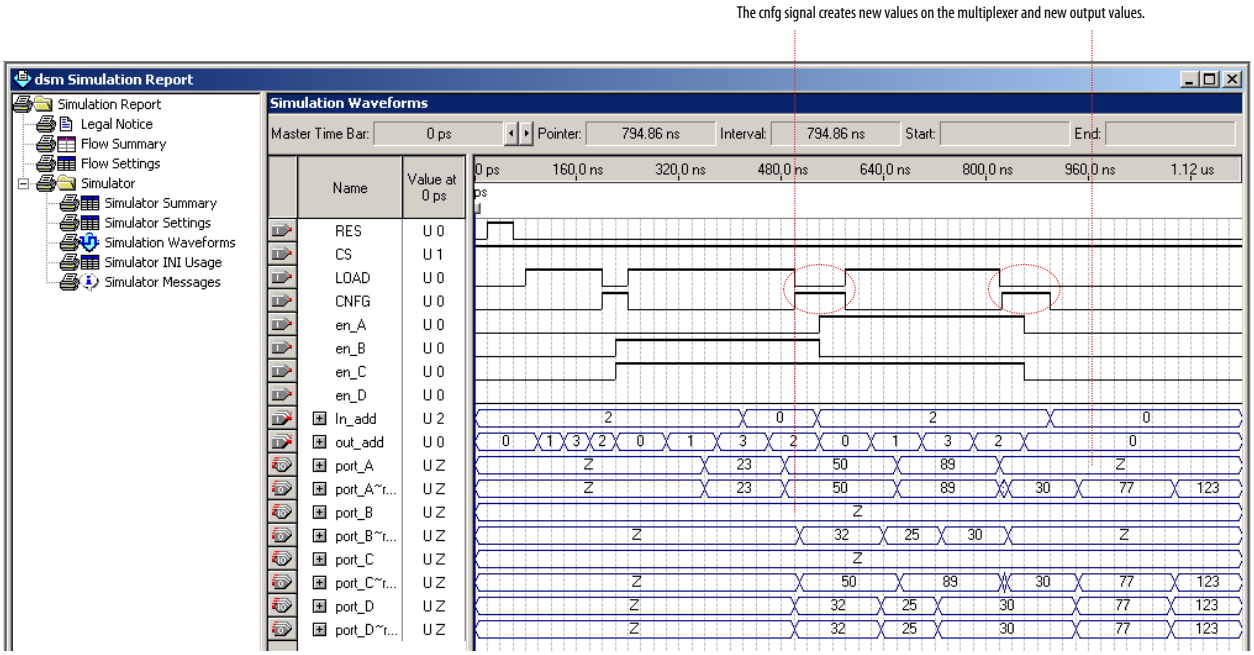
Table 11: Customized Crosspoint Switch Design Utilization for EPM3064ATC100 Device

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 Verification

You can verify the design for a 4-port customized crosspoint switch using the Quartus II software. The design verification occurs in both functional and timing simulations.

Figure 8: 4-Port Customized Crosspoint Switch Timing Simulations



With all the I/O pins in tri-state mode, each port was initialized. After initialization, output configuration 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.

Document Revision History

Date	Version	Changes
September 2014	2014.09.22	Added MAX V and MAX 10 devices.
March 2004	1.0	Initial release.