



XAPP240 (v1.0) March 14, 2000

# High-Speed Buffered Crossbar Switch Design Using Virtex-EM Devices

Author: Vinita Singhal and Robert Le

## Summary

High-speed switches are increasingly required in high-bandwidth applications. In the face of constantly changing networking standards, FPGAs offer switch designers flexibility and adaptability. FPGAs with expanded memory capacity, such as Virtex™-E Extended Memory (Virtex-EM) devices, are ideally suited for scalable, fast switches. This document discusses a high-speed buffered crossbar switch that effectively addresses each of these concerns.

## Introduction

This document highlights features of the Virtex-EM architecture in the context of high-speed gigabit bandwidth network switch designs. In particular, Virtex-EM devices with a favorable ratio of memory to programmable logic are naturally suited to buffered crossbar switch implementations. To address the need for higher bandwidth created by the explosive growth of the Internet and corporate intranets, the Gigabit Ethernet has become the preferred standard in the LAN marketplace.

Another emerging standard is the Common Switch Interface Specification (CSIX), which defines a common interface between different switches from different vendors. More information about this standard is available at the CSIX web site (<http://www.csix.org>). Generally, FPGAs are an ideal choice for applications that are based on changing standards, and the Virtex-E architecture with extended memory offers an ideal solution.

A 16-port buffered crossbar switch design supporting an OC-192 line rate of 10 Gb/s per port is described to illustrate the Virtex-EM architecture and features.

## Switch Architectures

Of the three basic switch architectures (shared bus, shared memory, and crossbar), shared bus and shared memory architectures have physical and functional limitations on their scalability. Buffered crossbar switches are highly scalable and are the obvious choice for high-performance switching.

### Basic Crossbar Switches

An  $n \times n$  crossbar switch is organized as an  $n \times n$  matrix to connect  $n$  input ports to  $n$  output ports. Crossbar switches can transfer packets from multiple input ports to multiple outputs simultaneously. Each pair of input and output ports has a dedicated path through the switch, and additional ports can be incorporated by adding switching elements. A scheduler keeps a maximum number of output ports busy.

Although the architecture is inherently non-blocking, it is susceptible to head-of-line (HOL) blocking due to output contention. This occurs when the scheduler uses a single FIFO buffer at the input port for incoming packets. The scheduler considers the packet at the head of a single FIFO for transfer only when it reaches the head of the FIFO queue. Packets behind the head-of-line packet destined for other output ports are blocked. HOL blocking is reduced in the crossbar architecture by maintaining separate FIFO buffers for each input-output port pair.

## Buffered Crossbar Switch Architecture

The addition of buffers at the crosspoints to reduce HOL blocking and hence increase throughput is referred to as buffered crossbar switch architecture. This scheme of placing buffers at the crosspoints instead of pure input buffering reduces the HOL blocking problem caused by output contention. With buffer memory placed at each crosspoint, the scheduler can maximize total throughput and keep many output ports as busy as possible by storing the input packets in buffer memory.

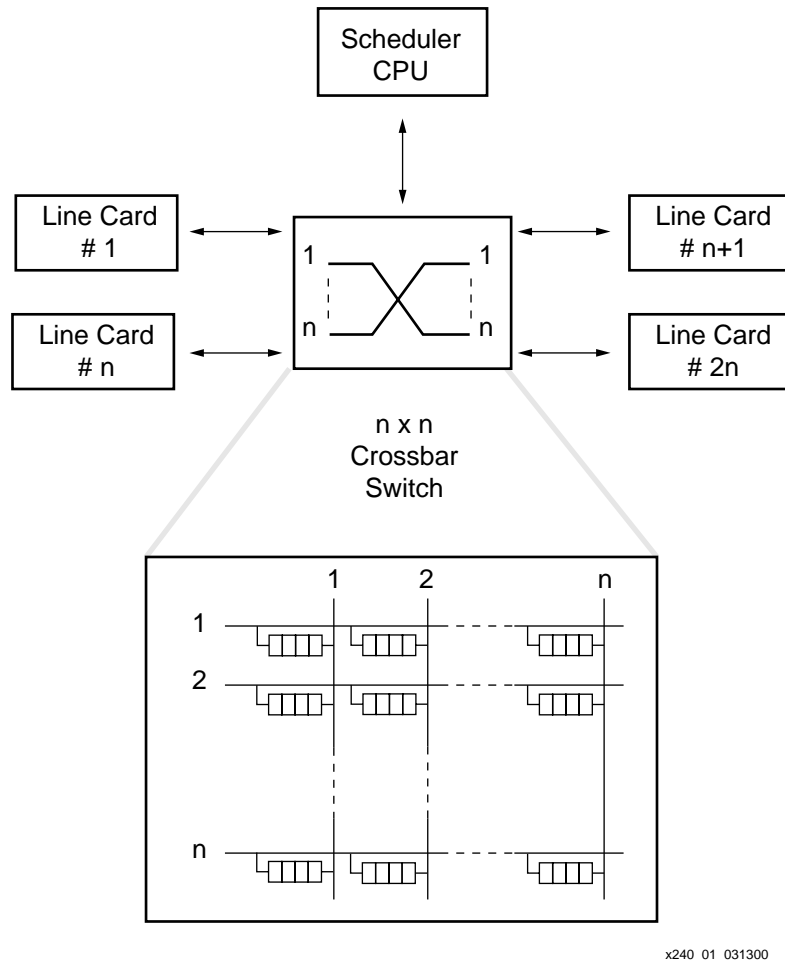


Figure 1: Buffered Crossbar Switch Architecture

## Design Example Using the XCV812E Device

### Design Overview:

The buffered crossbar switch architecture represented in Figure 1 is memory intensive. Buffer memory requirements for an  $n \times n$  switch with support for  $m$  number of priority or Quality-of-Service (QoS) classes =  $n \times n \times m$  buffers. For example, a  $16 \times 16$  switch connecting sixteen 8-bit input ports to sixteen 8-bit output ports with a single QoS class requires  $16 \times 16 \times 1 = 256$  buffers. The XCV812E device has over 1.12 Mb of block RAM, enabling each crosspoint to have 4 Kb of buffer space.

With more than 1 Mb of block RAM on the XCV812E, a fully populated  $16 \times 16$  buffered crossbar switch can be implemented with 4 Kb of high-speed True Dual-Port™ RAM buffers at each crosspoint. The optimum crosspoint buffer size and memory latency are important design considerations dictated by system bandwidth.

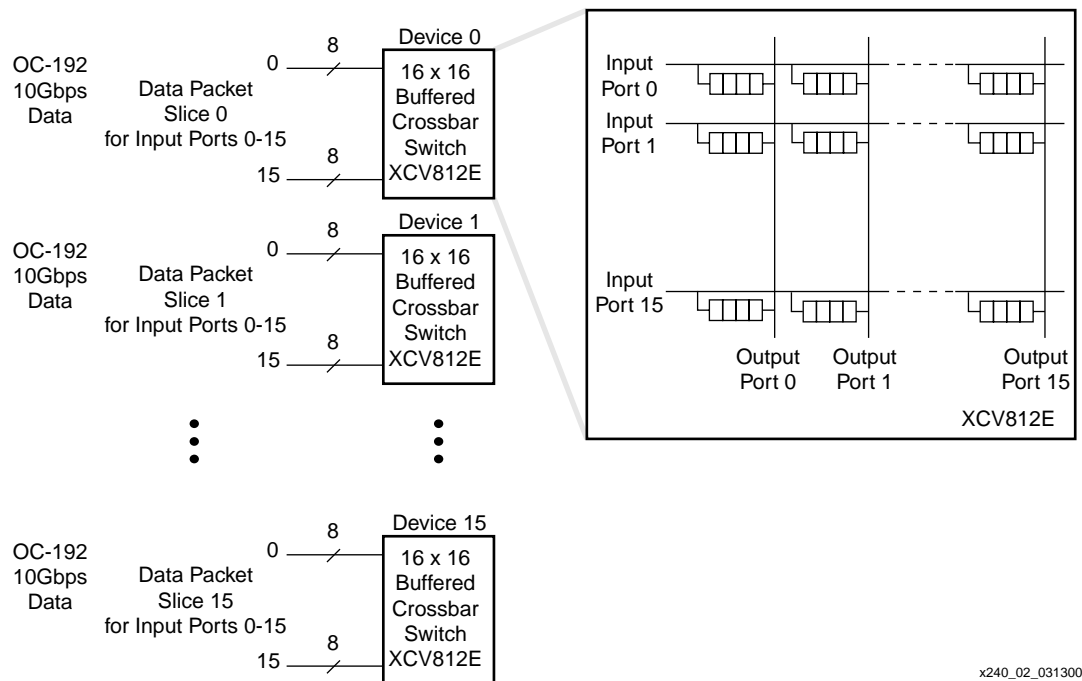
Higher aggregate bandwidths are obtained by breaking up a data packet into slices and switching them in parallel. Sixteen XCV812E devices, each implementing a byte-wide  $16 \times 16$

buffered crossbar switch at OC-192 rates of 10 Gb/s yield an aggregate bandwidth of 160 Gb/s. (See [Figure 2](#).)

The buffer size needed for the crossbar switch data chip per crosspoint per QoS class should be at least equal to 2 x port speed per slice x round-trip delay in the control feedback signal.

For a port speed of 10 Gb/s and 16 data slices (16 crossbar switch chips) and a propagation delay through the line card to the switch of approximately 800 ns, the smallest buffer size needed per crosspoint per QoS class is calculated as follows:

$$\text{Buffer Size} = 2 \times 10 \text{ (Gb/s)} \times 800 \text{ (ns)} / 16 = 1000 \text{ bits}$$



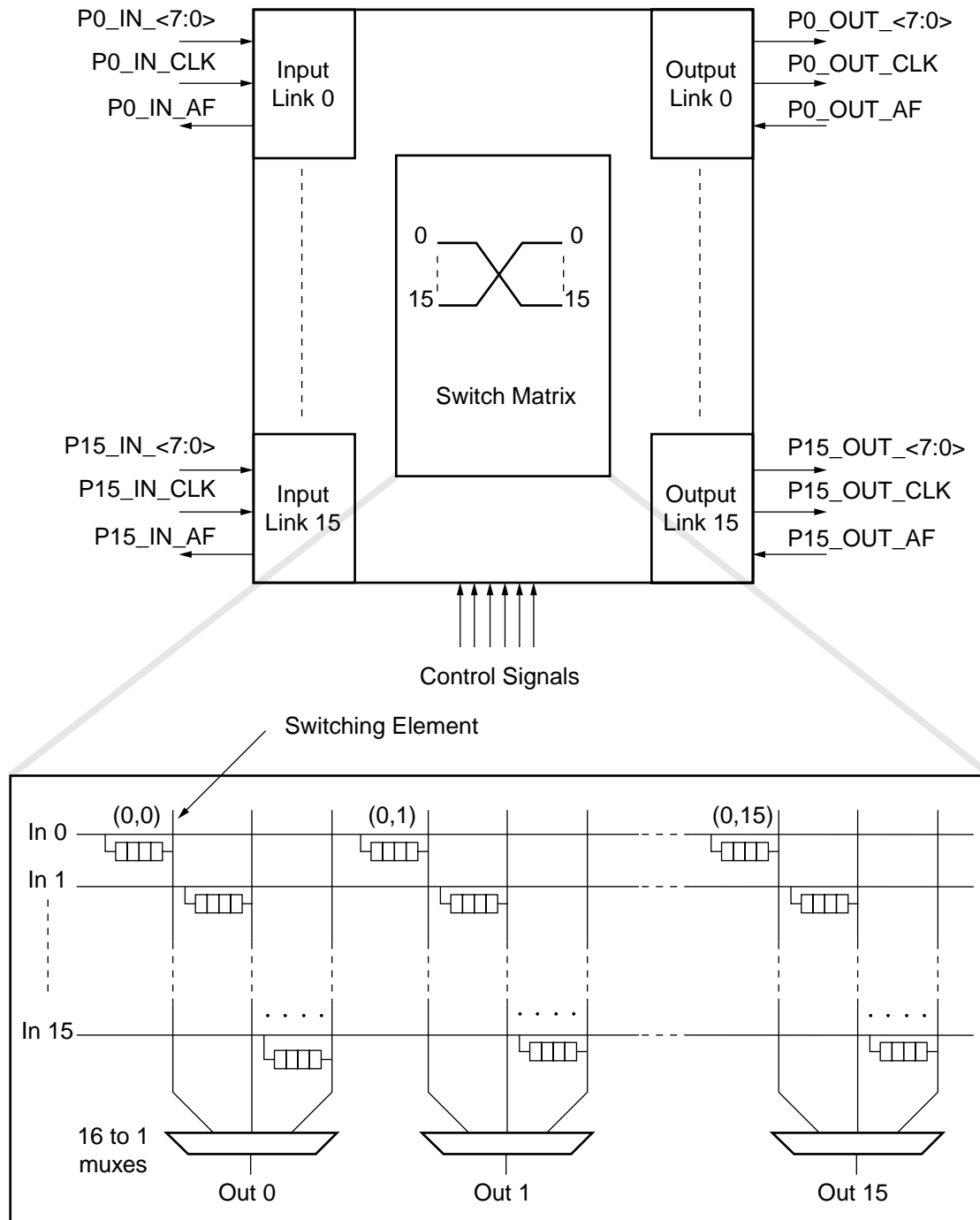
x240\_02\_031300

Figure 2: OC-192 16 x 16 Buffered Crossbar Switch

### The Physical Link

At each crossbar implementation, a block of True Dual-Port RAM is the link from input bus to output bus. In a byte-wide partition scheme, an 8-bit bus transfers data at a clock frequency of 100 MHz. The bandwidth of each port is 800 Mb/s (100 MHz x 8 bits). The total theoretical bandwidth of the 16-port byte-wide partition is 12.8 Gb/s (800 Mb/s x 16).

[Figure 3](#) illustrates the crossbar link in detail. Internally the crossbar switch buffer is a synchronous FIFO running at 100 MHz. Synchronous designs eliminate problems associated with different timing delays through different parts of the logic. They are also more reliable over variations of temperature, voltage and process.



x240\_03\_031300

Figure 3: Buffered Crossbar Switch Internal Detail

There are multiple levels of flow control and support. At the lowest level, link-level control is used to ensure that buffers do not overflow and packets are not dropped. The flow control signal  $P_x\_AF$  for each port becomes true when the FIFO is nearly full. The line card control logic sends data only when this signal is false. Since latencies due to board trace lengths and flow control signal detecting logic might be large, the FIFO should be capable of accepting a reasonable amount of data even after sending the flow control signal to indicate a near full condition. With more than 1 Mb of block RAM, the XCV812E device is more than adequate.

Crossbar links are not limited to particular electrical interfaces. Virtex SelectI/O+™ technology provides support for 20 I/O standards, including three differential signaling standards (LVDS, LVPECL, and BLVDS). For chip-to-chip interfaces, a high-speed single-ended standard, such as SSTL3 Class II, is a good choice. For better noise immunity and lower EMI, the LVDS differential standard with narrower data bus width is recommended.

For more details on Virtex SelectI/O+ technology, see Xilinx application note XAPP133 (<http://www.xilinx.com/xapp/xapp133.pdf>). Further information on I/Os based on differential signalling can be obtained from the following application notes:

- XAPP230 (<http://www.xilinx.com/xapp/xapp230.pdf>)
- XAPP231 (<http://www.xilinx.com/xapp/xapp231.pdf>)
- XAPP232 (<http://www.xilinx.com/xapp/xapp232.pdf>)
- XAPP233 (<http://www.xilinx.com/xapp/xapp233.pdf>)

## Data Path and Buffering in Crossbar Switches

**Figure 3** illustrates the data path of each node at the crosspoint. The switching element is multiplexer based. Wide muxes are used to implement the data path of the crossbar switch. The cascade/carry feature in the Virtex-EM device enables very efficient and fast implementation of high-speed muxes. Synchronous FIFOs are used at each crosspoint of the crossbar switch to provide buffering. Virtex-E True Dual-Port FIFOs allow incoming data to be written at one end of the port and transmitted data to be read from the other end. This feature significantly increases memory bandwidth and system performance (see <http://www.xilinx.com/xapp/xapp131.pdf>).

XCV812E devices provide 280 blocks of 4 Kb block RAM on chip with independent control signals for each port. The data widths of the two ports can be configured independently, providing built-in bus-width conversion. The block RAM also includes dedicated routing for an efficient interface with both Configurable Logic Blocks (CLBs) and other blocks of RAM (see <http://www.xilinx.com/xapp/xapp130.pdf>).

In Virtex-EM devices, the block RAM is organized in columns, starting to the left of column 0 and at the right outside edge and inserted at every fourth CLB column. Each memory block is four CLBs high, and each memory column extends the full height of the chip, immediately adjacent to the CLB column locations indicated in **Table 1**.

**Table 1: CLB/Block RAM Column Locations**

Virtex-E Device	0	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	
XCV405E	√	√	√	√	√	√	√			√	√	√	√	√	√	√							
XCV812E	√	√	√	√	√	√	√	√	√	√			√	√	√	√	√	√	√	√	√	√	√

The True Dual-Port feature coupled with efficient memory block placement is ideally suited to the implementation of data path and buffering at each crosspoint of the buffered crossbar switch. **Figure 4** illustrates a floorplan of the crossbar switch. The inputs are located at the upper and lower edges of the chip, with the outputs at the left and right edges of the chip.

The 4 Kb of block RAM buffer memory at each crosspoint is used to build high-performance synchronous 511 x 8 synchronous FIFOs. To provide distinct Empty/Full conditions, one address is dropped from the FIFO, so the FIFO depth is 511 instead of 512. Binary counters are used for the read/write addresses that address the block RAM. FIFO performance is in excess of 150 MHz as explained in detail in application note XAPP131 (<http://www.xilinx.com/xapp/xapp131.pdf>).

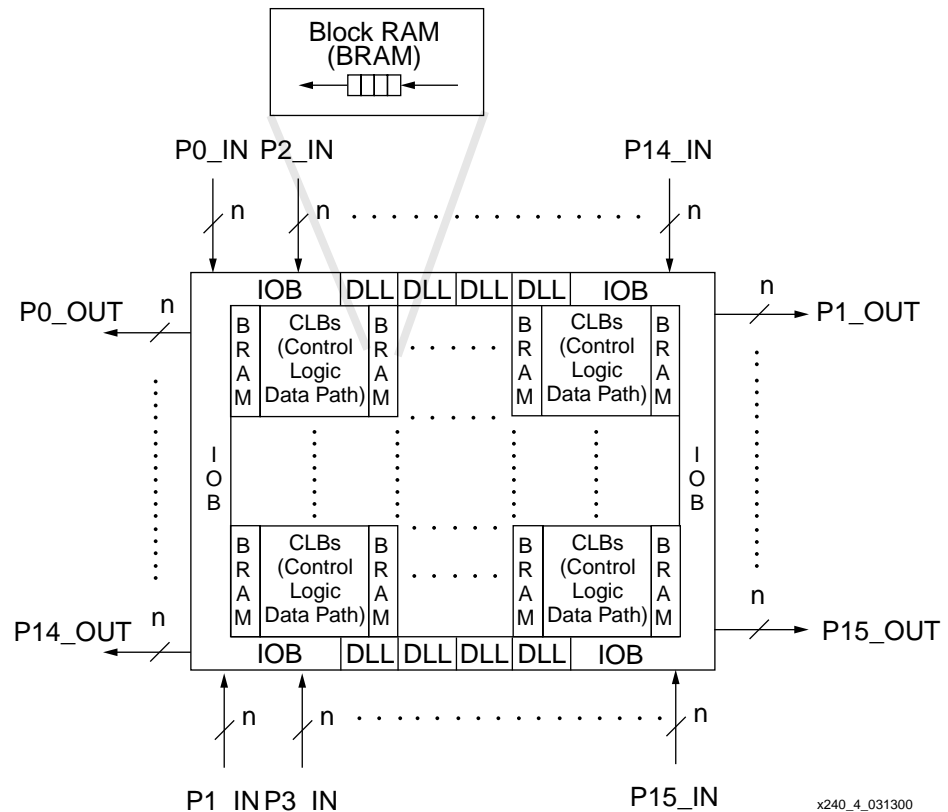


Figure 4: Floorplan Layout of the Reference Design

## Achieving Maximum Switch Performance

Optical interface standards such as Synchronous Optical Network (SONET) have made high-speed data rates possible. The XCV812E buffered crossbar switch example described above needs to keep up with the OC-192 line rate of 10 Gb/s. Having determined that 128 bits of data (partitioned or sliced into sixteen 8-bit busses) are processed in parallel per port, the input FIFOs need to operate at a 78 MHz clock rate (10 Gb/s / 128). A 100 MHz FIFO clock rate is required to maintain the incoming data rate after considering packet overhead. The XCV812E device can implement 170 MHz FIFOs that easily meet these requirements. An OC-192 16 x 16 buffered crossbar switch design can be implemented with sixteen XCV812E devices to deliver up to 160 Gb/s (16 ports x 10 Gb/s per port) aggregate bandwidth.

Switch performance can be further enhanced by operating the switch buffer memory at a rate faster than the required line rate, known as “speed-up.” A speed-up factor of two implies that two packets are transferred from each input port and two packets are destined for each output port during each scheduler cycle or packet time. Determining the optimum speed-up factor to fully utilize the buffer queues is critical in switch design. Stanford Professor Nick McKeown cites recent research that suggests a crossbar switch with a speed-up factor of two behaves almost identically to a crossbar switch with a speed-up factor of  $n$ , independent of the size of  $n$ . Hence, a speed-up factor of two seems to be the optimum implementation.

## Conclusion

This application note illustrates how the Virtex-EM device with its abundant True Dual-Port RAM memory makes it the ideal solution for high-speed crossbar switches. Using sixteen XCV812E devices, system designers can easily achieve an aggregate bandwidth of 160 Gb/s for a 16 x 16 buffered crossbar switch application.

## References

“Distributed Processing is on top,” Charlie Jenkins EE Times, Jan 24 2000.

“Gigabit Ethernet has arrived in force, with at least 40 vendors shipping products of various stripes. And just in time.” David Axner, Network World, July 20 1998.

“A Fast Switched Backplane for a Gigabit Switched Router,” White paper by Nick McKeown, Business Communications Review, December 1997 Vol. 27.

“A Survey of ATM Switching Techniques,” Sonia Fahmy, Department of Computer and Information Science, Ohio State University.

---

---

## Revision History

The following table shows the revision history for this document.

Date	Version	Revision
3/14/00	1.0	Initial Xilinx release.