

# Designing High-Speed ATM Switch Fabrics by Using Actel FPGAs

The recent upsurge of interest in Asynchronous Transfer Mode (ATM) is based on the recognition that it represents a new level of both speed and simplification in telecommunication networks. The most significant characteristic of ATM is that it requires minimum cell processing in network nodes and in links such as repeaters, bridges, and routers. This means that ATM allows systems to operate at rates much higher than current packet-switching systems allow. This improved performance is due to higher media quality and to ATM operation in a connection-oriented mode that guarantees minimum packet loss. This low packet loss is the result of not granting entrance to the network until completion of a setup phase that allocates all necessary network resources.

To reduce the size of the internal buffers in switching nodes, and thus to reduce the queuing delays in these buffers, the information field length in ATM packets is kept relatively small. As a result, as packet size goes down, the speed requirement for each switching node on the network goes up. In general, to keep packet loss to a minimum, the throughput of ATM switching nodes must be in the 1-gigabit-per-second range.

This application note describes how to design typical high-speed switch fabrics that route ATM packets on broadband networks. *Switch fabric* is a term used to denote a large group of basic switching building blocks connected in a specific topology. The design, analysis, and implementation of these building blocks will be described.

## ATM Switching Applications

One of the main tasks of an ATM switching node is to transport ATM cells at high speed from its input ports to its destination output ports. This task is performed by the switch fabric. The switch fabric establishes a connection between an arbitrary pair of input and output ports. Switch fabrics usually consist of identical basic units called *switching elements*. The switching elements are interconnected in a specific topology to create the switch fabric.

The Actel ACT 3 family of FPGAs, with their high-speed multiplexer-based architectures, are, as will be seen in this application note, an excellent fit for applications, such as ATM switching, that stress the heavy use of multiplexing. This

application note will describe in detail the designs of two typical high-speed ATM switches:

- A pipelined 16:16 switch fabric
- A 16:16 multipath interconnect (MIN) switch fabric

## Pipelined 16:16 FPGA Switch Fabric

One of the simpler FPGA approaches to ATM switch fabric design is shown by the straightforward 16:16 multiplexing scheme in Figure 1. This switch fabric design is known as a *single-path network*, because the same path is always used from any given input to a given output. In this example, the switch fabric has 16 input ports and 16 output ports. To achieve connectivity, it employs 16 16:1 multiplexers called FFMX16's (Figure 2). Each of the 16 FFMX16s uses five Actel DFM6A basic 4:1 multiplexed flip-flops connected in two pipelined stages.

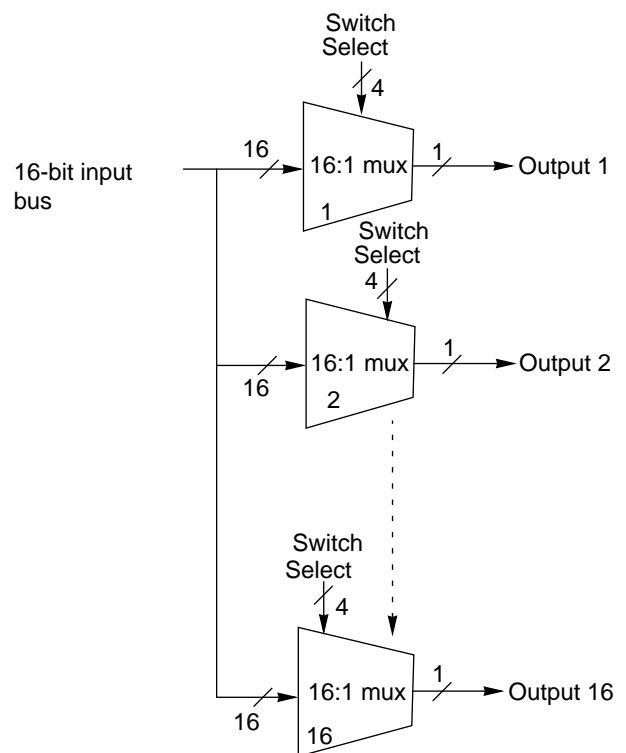
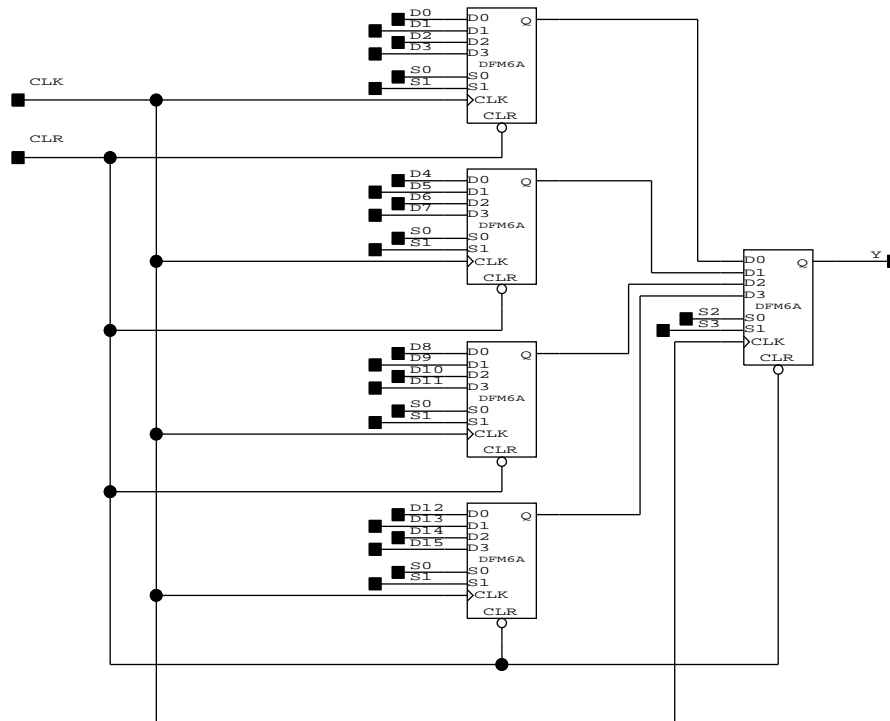


Figure 1 • 16:16 Basic Multiplexer Switch Fabric



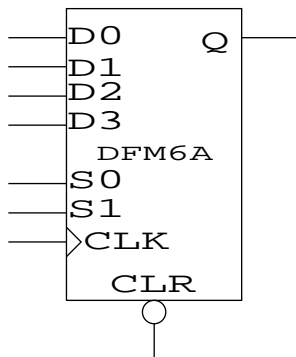
**Figure 2 • Two-Stage Pipelined 16:1 Multiplexer (FFMX16)**

Each DFM6A (Figure 3) is a 4:1 multiplexer driving a flip-flop and occupies a single Actel ACT 3 sequential logic module. This multiplexed flip-flop introduces only one level of logic delay in the design. In this implementation, once each of the two stages of the 16:1 multiplexer is full, the pipeline outputs data on every subsequent clock cycle. Thus, these 16:1 multiplexers are effectively implemented in one logic level, providing improved throughput.

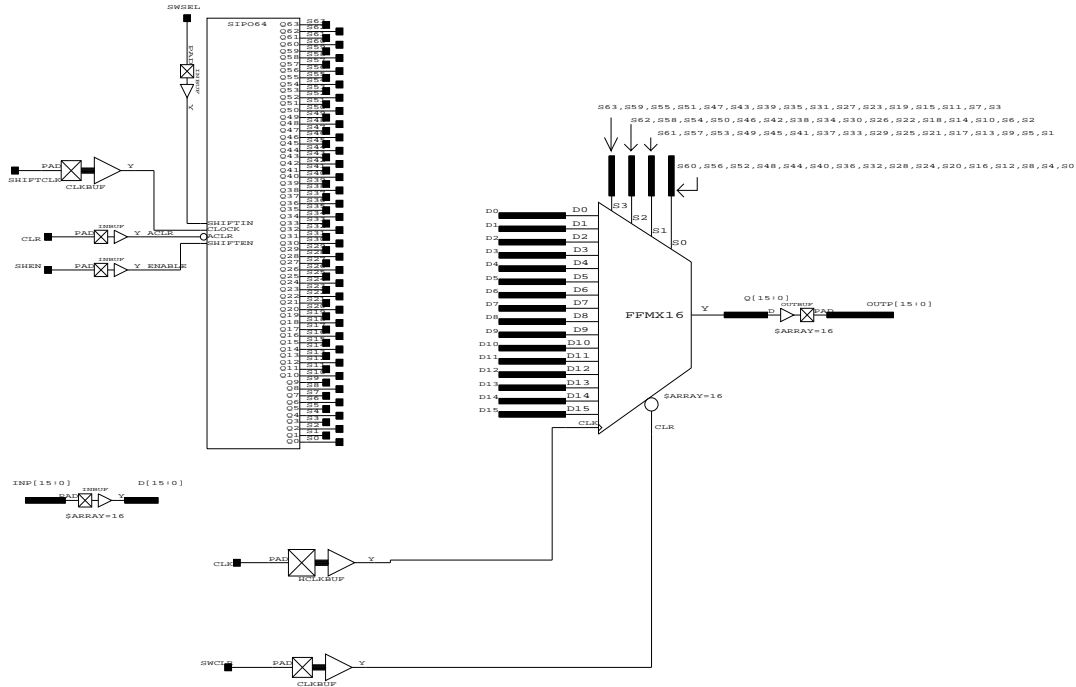
#### Switch Fabric Driving Circuit

The driving circuit for each of the FFMX16s in the 16:16 switch fabric is shown in Figure 4. As shown in the figure, selecting data for the output of each FFMX16 requires 64 signals. This number is based on the need for four switch-select inputs (S0, S1, S2, and S3) for each of the FFMX16s. Switch-select signals S0 and S1 operate with the first stage of each multiplexer, and select signals S2 and S3 operate with the second stages. A drive signal is received as a serial bit stream (SWSEL) that is converted to parallel form by a 64-bit serial-to-parallel shift register (SIPO64). Input data to the switch is received on the lines D[15:0]. Notice that the FFMX16 shown in Figure 4 represents 16 FFMX16s, so the inputs are 16 bits wide.

It takes two clock cycles to fill the FFMX16 pipeline, after which data is present on the OUTP[15:0] bus at each clock cycle. The 64-bit data selection word S[63:0] from the shift register is divided into four groups of 16 bits each, which are used to select the appropriate routing through the switch fabric. Note that there are separate clock lines for both the shift register and for the switch fabric so that data can be clocked to the output bus at a rate different from that of the shifted control bits.



**Figure 3 • Multiplexed Flip-Flops DFM6A**



**Figure 4** • Top Level View of the 16:16 Switch Fabric Design

Using ACTgen Macro Builder

The 64-bit serial-to-parallel shift register (SIP064) is generated by the Actel macro generator, *ACTgen* included with Designer Series software packages. With ACTgen’s graphical user interface, you can build structured macros (counters, adders, etc.) by simply clicking on a few menu choices. The ACTgen Macro Builder then creates functions that effectively use the Actel architecture. Each macro is developed with the goal of limiting module count, maximizing performance, and restricting loading to acceptable levels.

In this design, the SIP064 is generated by simply choosing the desired parameters from ACTgen’s graphical user interface (64-bits, serial-to-parallel, active-low clear, active-high shift enable, and positive-edge triggered clock). The created shift register is then instantiated in the design with no need for simulation. The ACTgen macros are already tested to guarantee correct functionality.

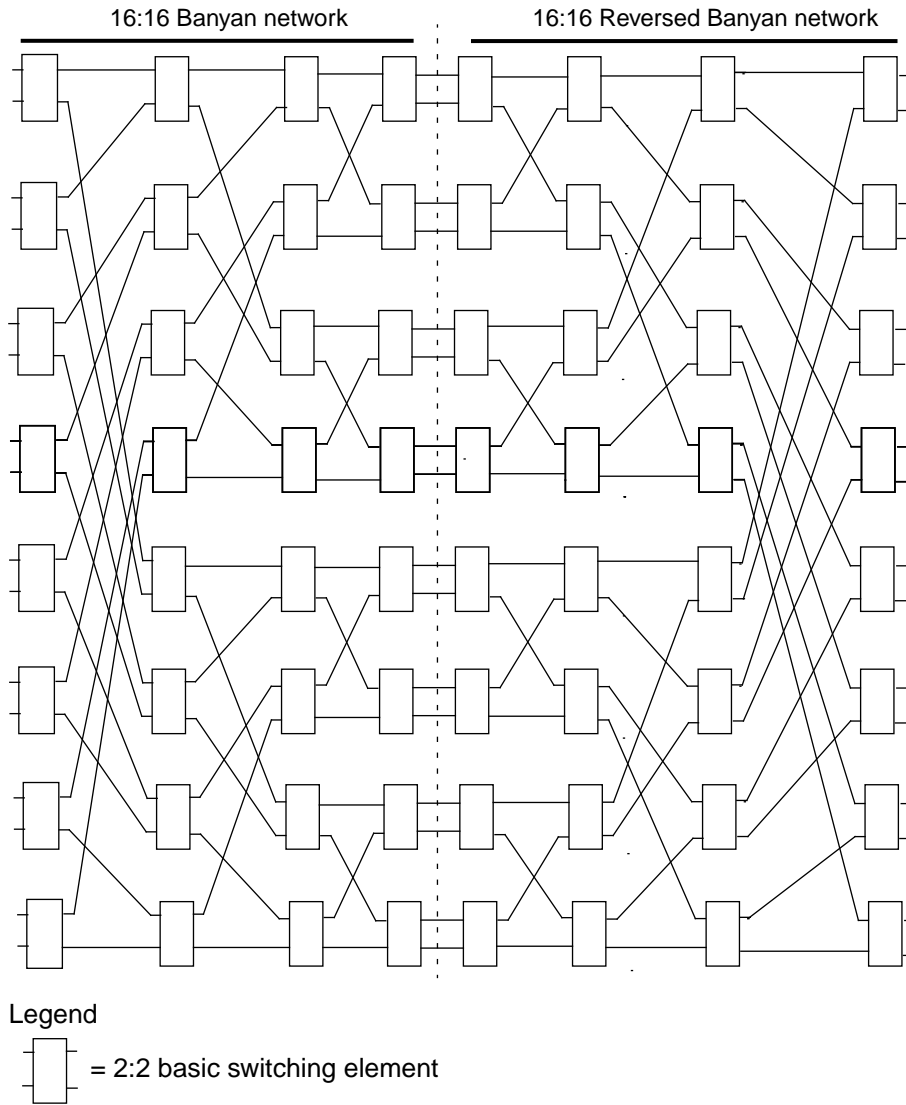
**Note:** *In the N:N multiplexed structure shown here, any given input may be broadcast to all outputs simultaneously. Also, an advantage of this approach is that after only two clock cycles, the pipeline is full and ready to output data. A disadvantage of this scheme is that it allows only one possible path, and no alternative paths, between each input and output. To implement a multiple-path capability, multiples of this switch fabric can be cascaded together. However, a better solution is the MIN switch fabric.*

### 16:16 Multipath Interconnect (MIN) Switch Fabric

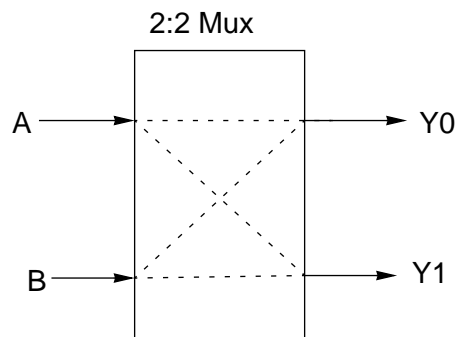
The primary advantage of a multipath interconnect network (MIN) is that it permits the creation of alternative paths between a given input and a given output in order to avoid possible packet collisions. One implementation of a MIN switch is the two-section Banyan network shown in Figure 5. This network consists of four stages that drive a second group of four. The second group is made up of the first four stages with a reversed topology—it is the mirror image of the first. Adding this second half produces a complete MIN switch fabric in a minimum number of stages.

The first four stages (see Figure 5) enable any output to be reached from any input via one specific path. This is the standard Banyan configuration. The second four stages use a reversed Banyan topology. Together, the two sections provide the multiplicity of paths required for a MIN switch fabric. That is, in an N:N MIN switch fabric, N internal paths are available to reach any output from an arbitrary input.

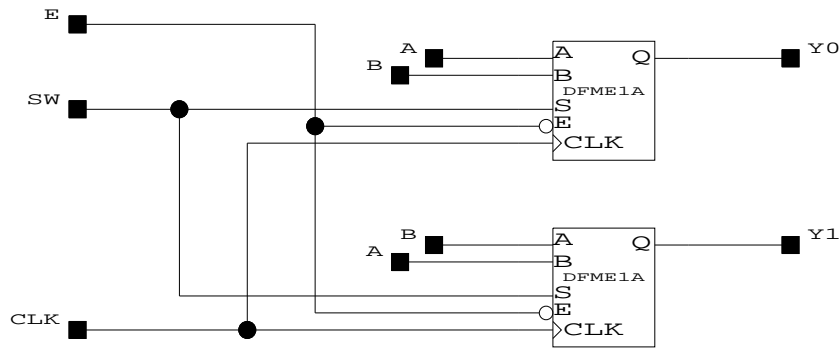
Each basic switching element used in this switch fabric is a 2:2 switch. (See Figure 6.) Depending on the value of switch line SW, the data will be either passed or crossed between the input and output lines. Figure 7 shows the implementation of the basic switching element using Actel DFME1A ACT 3 multiplexed flip-flops. As shown in the figure, two multiplexed flip-flops are required to implement the switching element. The truth table for the basic switching element shown is given in Table 1.



**Figure 5** • 16 X 16 Multipath Interconnect Network (MIN) Switch Fabric



**Figure 6** • Possible Signal Paths in a 2:2 Basic Switching Element



**Figure 7** • 2:2 Basic Switching Element SWCELL

**Table 1** • Truth Table for 2:2 Basic Switching Element SWCELL

Clock CLK	Enable E	Switch SW	Y0	Y1
X	1	X	As in previous state	As in previous state
↑	0	0	A	B
↑	0	1	B	A

Notes: ↑ = Triggered on positive edge of clock  
X = Don't care

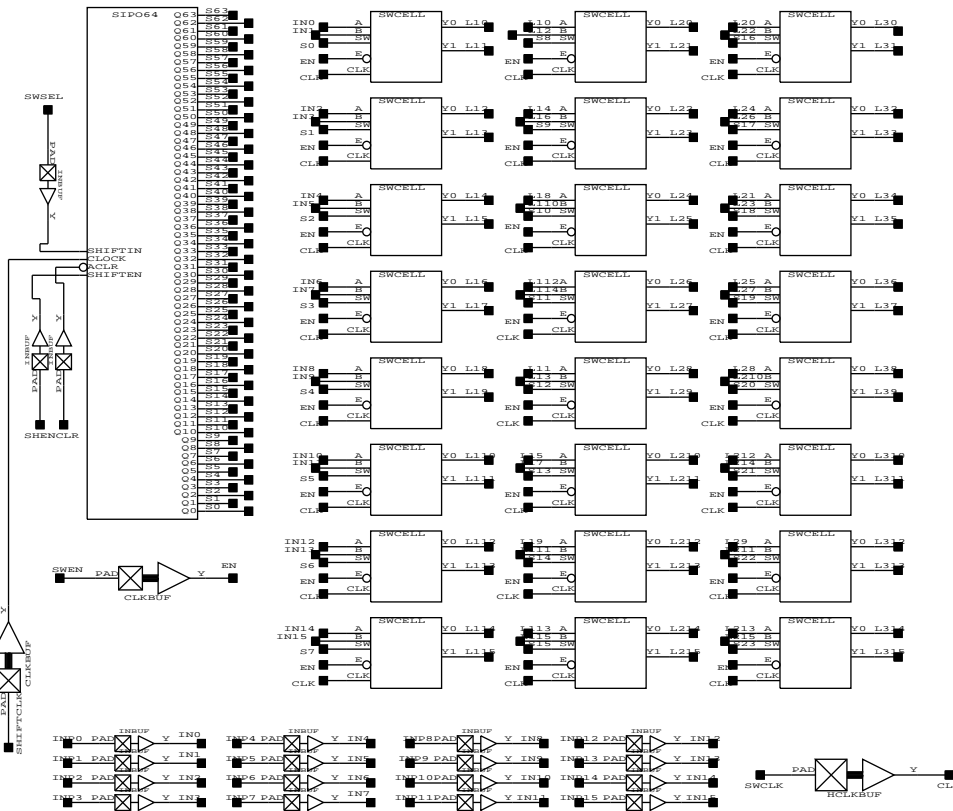
Assuming an N:N MIN switch fabric, the number of stages in the network is  $2\log_2 N$ . If  $N = 16$  (as in the present case), the MIN switch fabric is implemented in eight stages. Thus, a 16:16 MIN can be constructed of eight stages, with each stage consisting of eight 2:2 basic switching elements.

#### Switch Fabric Driving Circuit

The driving circuit for the MIN network is similar to the one used for the multiplexer-based switch fabric described in the previous section. As shown in Figure 8, the switching elements (SWCELL) are connected to each other to implement the topology shown in Figure 5. The SW lines of the switching elements are driven by the outputs (S[63:0]) of a 64-bit shift register. The S[63:0] signals are received as a serial bit stream (SWSEL) and are converted to parallel form by a 64-bit serial-to-parallel shift register (SIPO64). Input data to the switch is received on the lines IN[15:0] and is clocked to the outputs once the SWCELLs are enabled.

It takes eight clock cycles to fill the switch fabric pipeline, after which data is present on the OUTP[15:0] bus at each clock cycle. The 64 bits of the data selection word (S[63:0]) from the shift register are used to select the appropriate routing through the switch fabric. Note that there are separate clock lines for the shift register and for the switch fabric so that data can be clocked to the output bus at a rate different from that of the shifted control bits S[63:0].

**Note:** The complete multipath interconnect switch fabric is implemented by using 128 ( $8 \times 8 \times 2$ ) multiplexed flip-flops of the type DFME1A. The straight multiplexed switch structure discussed in the previous section requires 80 ( $5 \times 16$ ) DFM6A multiplexed flip-flops. However, this size differential does not translate for larger values of  $N$  (where  $N$  is the number of input and output ports). As  $N$  gets larger, the number of modules required to implement the MIN network does not increase as rapidly as it does for the simple mux-structured network. Also, notice that the multiplexed flip-flop used in the MIN network is a 2:1 type, whereas the straight mux switch fabric requires a 4:1 type. The 2:1 multiplexed flip-flop offers the advantage that, because of its lower fanin, it is easier to route on the Actel software.



**Figure 8 • Top-Level View of the MIN Switch Fabric Design**

### Timing Analysis

The MIN switch fabric discussed here can be implemented in most Actel ACT 3 devices, such as the A1425A, the A1440A, A1460A, and the A14100A. The timing analysis given in this section was obtained from the A1440A-2.

The MIN switch fabric can be operated as fast as the slowest switching element can switch its data from input to output. The basic switching element has a 5.7 ns clock-to-q (input-to-output) delay, along with 0.7 ns of setup time. Hence, the maximum frequency of the switch fabric clock is 156 MHz. In a 16:16 switch, this provides a maximum throughput of 2.5 gigabits per second. Note that it takes eight clock cycles for data to move from the switch fabric input to its output (about 51.2 ns). However, as in all such pipelines, once all stages of the network are filled up, data is output at every subsequent clock cycle.

### Conclusion

The basic concept behind switch fabrics is multiplexing data from input ports to output ports. The multiplexer-based architecture of Actel FPGAs fits this requirement. High-speed switching networks of almost any topology can be implemented efficiently using the multiplexed flip-flops in the Actel library. Each of these flip-flops is mapped to only one sequential module within the FPGA to take maximum advantage of the die area within the chip.

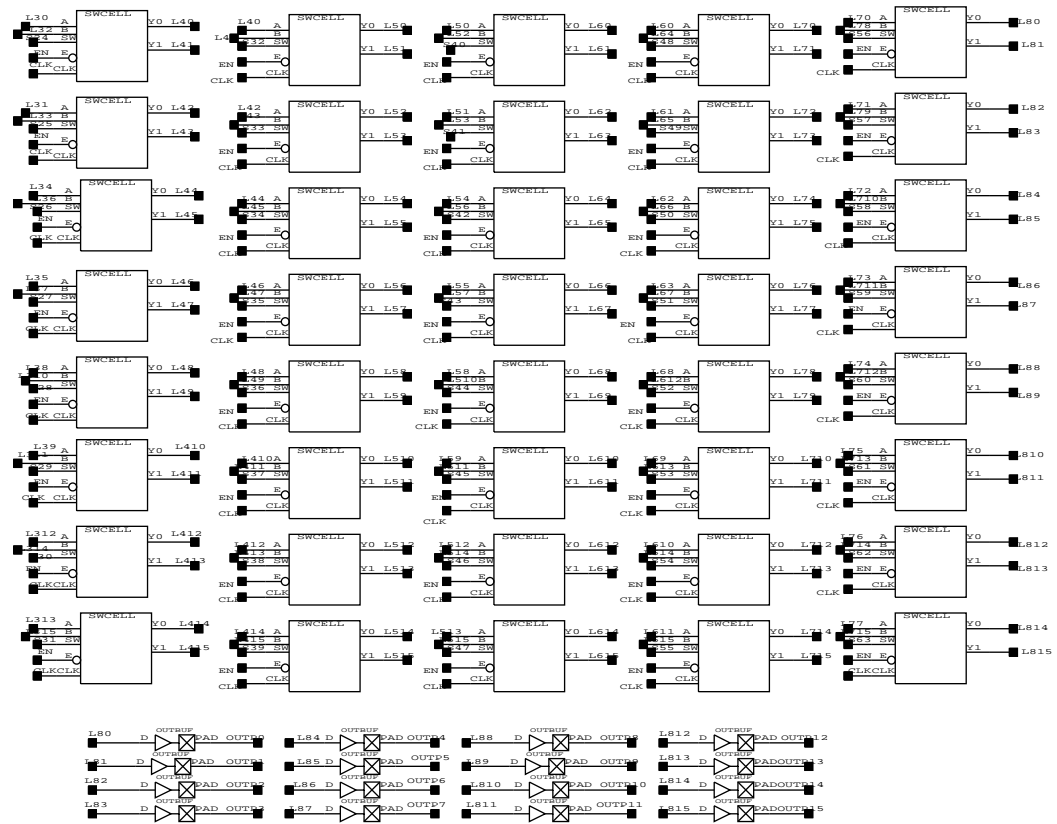


Figure 8 • Top-Level View of the MIN Switch Fabric Design (continued)

