XILINX®

Xilinx XC5200 vs. Altera FLEX 8000A FPGAs

January 1996

White Paper

Table of Contents

Executive Summary

At first glance, there are similarities between the Xilinx XC5200 and Altera® FLEX 8000A™ FPGA architectures. Both have the raw gate density required for many applications. However, the lack of system features and routing flexibility in the FLEX 8000A limits its capabilities. The added features of the XC5200 provide a simpler and more efficient means to implement designs on a single device.

XC5200 Advantages

The XC5200 architecture has the following advantages over the FLEX 8000A architecture:

Logic Cell

- Higher effective XC5200 gate density than a similarly-priced FLEX 8000A device.
- Independent block outputs from both the combinatorial logic function and the flip-flop provide increased density and flexibility.
- Logic cells can be multiplexed together to provide any function of five inputs or a four-to-one multiplexer.
- The flip-flop can be either a D-type flip-flop or latch.
- A clock enable input to the flip-flop that provides higher logic block density than FLEX 8000A logic element for synchronous designs.

Table 1. Xilinx XC5200 Family Device Features

Feature	XC5202	XC5204	XC5206	XC5210	XC5215
Typical Usable Gates	2,200-2,700	3,900-4,800	6,000-7,500	10,000-12,000	14,000-18,000
Altera FLEX "gates"	3,076	5,714	9,333	15,428	23,901
VersaBlock Array	8 x 8	10 x 12	14 x 14	18 x 18	22 x 22
Logic Cells (LC)	256	480	784	1,296	1,936
Flip-Flops	256	480	784	1,296	1,936
Max. User I/O pins	84	124	148	196	244

Note: Altera FLEX "gates" = $\frac{\text{FLEX Claimed Gates}}{\text{\# of LEs in FLEX Device}}$ x # of LCs in XC5200

Table 2. Altera FLEX 8000A Family Device Features

Feature	EPF8282A	EPF8452A	EPF8636A	EPF8820A	EPF81188A	EPF81500A
Altera FLEX "gates"	2,500	4,000	6,000	8,000	12,000	16,000
Logic Elements (LE)	208	336	504	672	1,008	1,296
Flip-Flops*	282	452	636	820	1,118	1,500
Max. User I/O pins	78	120	136	152	184	208

* Includes flip-flops in I/O blocks

- Flip-flop control signals can be sourced from any internal or external signal. FLEX 8000A is limited to designated pins.
- A direct input to the flip-flop bypassing the combinatorial function allowing the function generator and flip-flop to be used independently.
- The direct input and output from a logic cell can be used to:
  - Hop on and off the carry chain, or
  - To provide additional routing through a cell.

I/O Cell

- Inversion control on inputs, outputs, and output-enable signals.
- Zero hold time to flip-flops adjacent to I/O, simplifying system timing for input registers.
- Fast direct connections between the I/O block and the outer ring of logic cells to provide input and output flip-flops. This method provides extra features on I/O flip-flops or latches such as clock enable.
- More independent sources for control signals such as output-enable, clock enable, clock, and clear.
- Plentiful independent output-enables for building busses and open-drain outputs.
- Specified and guaranteed 50 pF pin-to-pin timing numbers.

Interconnect

- More abundant and more flexible routing than in FLEX 8000A. XC5200 logic cells can drive any I/O pin. FLEX 8000A LEs can only drive a few pins.
- Internal three-state capability for building internal bi-directional data busses.
- Extra ring of routing resources, called VersaRing, allows design changes while maintaining original pinout. No need to modify the PC-board layout after every logic change.
- Global buffers can drive any logic, control, or clock input. FLEX 8000A buffers have limited connectivity.

General

- Lowest cost per gate of any programmable logic device currently available.
- Footprint compatibility between members of the XC5200 family and between devices in the XC4000 and XC8100 families.
- Proven migration path to Xilinx HardWire™ gate arrays for high-volume cost reduction.
- Boundary scan (JTAG) support on all members of the XC5200 family.
- Special global features such as global reset, global three-state, internal oscillator, and readback.

FLEX 8000A Advantages

The FLEX 8000A architecture has the following advantages over the XC5200 architecture:

Logic Cell

- Preset input for applications requiring simultaneous asynchronous set/clear.

I/O Cell

- Faster input register set-up time, but with a non-zero hold time.
- Faster clock-to-output timing.

Interconnect

- Faster execution time for placement and routing software on simple designs, partially due to limited connectivity.
- Applications that map into groups of logic array blocks have good performance.

Architecture Overview

At first glance, there are similarities between the Xilinx XC5200 and Altera FLEX 8000A architectures. Both include an interior array of logic blocks, surrounded by a perimeter of I/O blocks, with programmable routing resources between the blocks. Both use static memory cells to control how the resources on the device are configured, providing in-system reprogrammability.

Both the Xilinx and Altera logic blocks use look-up-table (LUT) based function generators to build combinatorial logic functions and dedicated D-type flip-flops for registered functions. The FLEX 8000A devices have dedicated circuitry to provide carry logic and dedicated circuitry to provide logic cascading. The XC5200 devices have a dedicated carry multiplexer that can be used for either carry logic or to generate functions of more than four input variables. Likewise, both architectures include a hierarchy of programmable routing resources.

However, further examination of the subtleties of the architectures reveals features in the Xilinx architecture that offer better system-level performance, increased functionality, higher capacity, and better flexibility.

Logic Block Comparison

The basic combination of a look-up-table (LUT) and a flip-flop is called a Logic Cell (LC) in the Xilinx nomenclature. Four LCs are grouped together to form the logic blocks of the array, called CLBs (Configurable Logic Blocks). Altera calls their logic blocks LEs (Logic Elements). Groups of eight LEs form a Logic Array Block (LAB) within the FLEX architecture. Figure 1 includes representative diagrams of a Xilinx XC5200 logic cell (LC) and an Altera logic element LE.

XC5200 Logic Cell

Each XC5200 LC contains a 4-input look-up-table (LUT) to implement combinatorial functions, a storage element, and control logic. There are five independent inputs and three outputs to each LC. Both the combinatorial function and the storage element outputs are available to other logic. Likewise, the data input to the storage element can come from the combinatorial function or directly from a dedicated LC input called Direct In. The Direct In input also provides a way to initialize the carry chain.

The independence of the inputs and outputs allows the software to maximize the resource utilization within each LC. Each Logic Cell also contains a direct feedthrough path as an additional routing resource that does not sacrifice the use of either the function generator or the register.

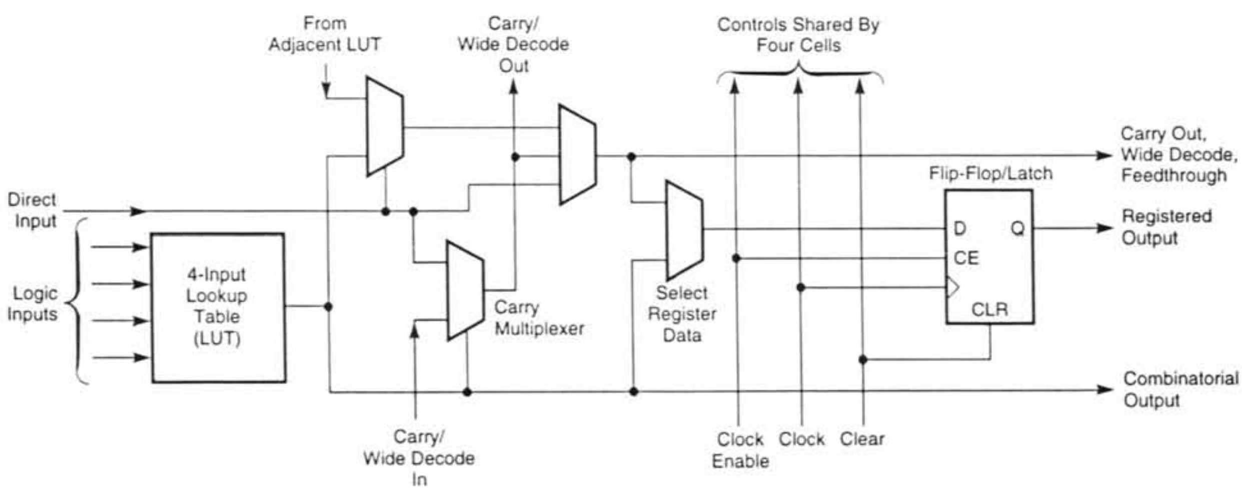
The control logic consists of carry logic for fast and efficient arithmetic functions originally pioneered in the XC4000 FPGA and XC7200A families. The carry logic can also be configured to decode very wide input functions.

The XC5200 LC storage element is configurable as either a D-type flip-flop or a latch. The storage element also features:

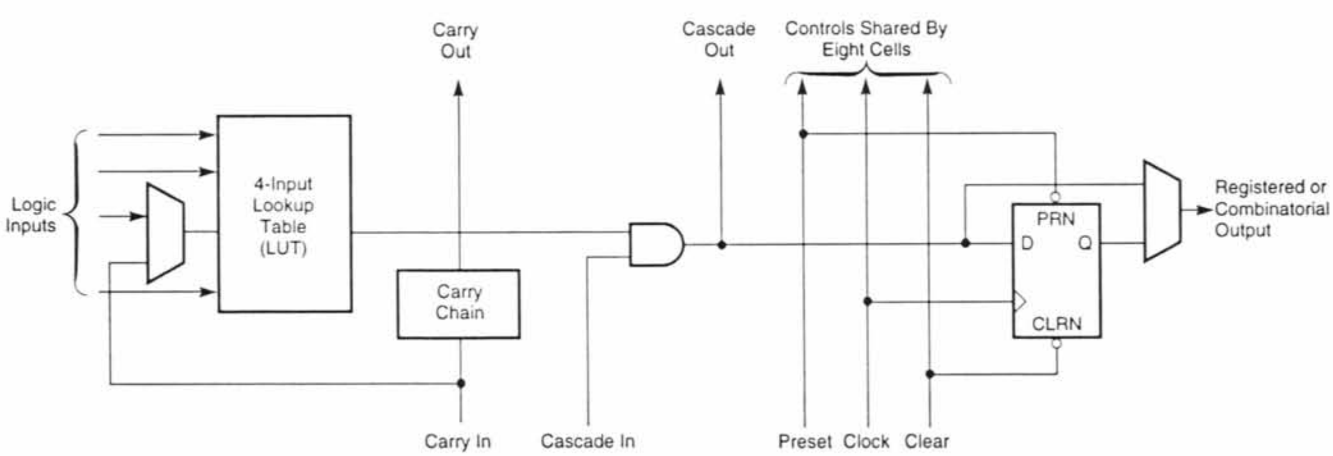
- A direct input, bypassing the LUT, that increases the effective density of register-intensive applications
- A dedicated, independent output that increases the effective density of register-intensive applications
- A shared clock input with individually selectable polarity control
- A shared, but optional, clock-enable input
- A shared, but optional, asynchronous clear input

FLEX 8000A Logic Element

The FLEX LE is similar to XC5200 LC in that it provides a 4-input LUT and a flip-flop (no latch option). It offers carry



a). Xilinx XC5200 Logic Cell (LC)



b). Altera FLEX 8000A Logic Element (LE)

Figure 1. Basic Logic Blocks For Xilinx XC5200 and Altera FLEX 8000A FPGAs.

logic and logic cascade capabilities. Note that the LE's single output come from either the combinatorial logic or the register but not both. Also, the flip-flop's data must come from the 4-input combinatorial logic function.

The FLEX LE offers an asynchronous Preset which the XC5200 LC does not. Few applications require both Preset and Clear simultaneously. Asynchronous Preset can be implemented in the XC5200 by inverting both the input to and the output from the flip-flop.

Clock Enable

Clock enables are almost always needed in synchronous designs. Significantly more designs require a flip-flop clock enable than require simultaneous set and reset. The XC5200 device provides a valuable clock enable input not available in the FLEX architecture. Clock enables are essentially free in the XC5200 LC in that they do not steal resources from the combinatorial logic function.

By contrast, clock enables in the FLEX architecture use two of the four LUT inputs available in an LE. This reduces the effective capacity of the LUT by 50%. Furthermore, using the LUT to build a clock enable introduces extra delay that reduces system performance. Xilinx learned the need for clock enable from its first FPGA architecture, called the XC2000 family, introduced in 1985.

Table 3. Logic Cell or Element Feature Summary

Feature	XC5200	FLEX 8000A
Lookup Table Inputs	4	4
Logic Cell Inputs	***5***	4
Logic Cell Outputs	***3***	1
Storage Element	***Flip-Flop or Latch***	Flip-Flop
Independent logic and flip-flop outputs	***Yes***	No
Direct input to storage element	***Yes***	No
Clock sharing	1 clock for 4 LCs	2 clocks for 8 LEs
Clock Enable	***Yes***	No
Clock Polarity	***Yes***	No
Asynch. Clear	Yes	Yes
Asynch. Preset	No	***Yes***

Note: ***Bolded italics*** text indicates advantage.

A simple circuit demonstrates the benefits of both the direct input and clock enable found in the XC5200 architecture. The circuit in Figure 2 shows a simple, clock-enabled register implemented in both the XC5200 and FLEX logic blocks. The XC5200 family flip-flops have both a direct data input and a clock enable input. The

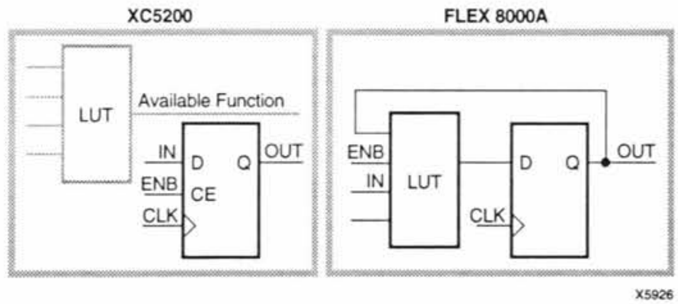


Figure 2. A Clock-enabled Flip-flop Built in Both XC5200 and FLEX FPGAs. Note that the XC5200 block has additional resources available.

combinatorial function (LUT) is still available for any 4-input function. Also, because the XC5200 LCs have both a combinatorial and a flip-flop output, the 4-input function generator in that LC can be used independently of the flip-flop.

By contrast, the FLEX architecture lacks either a clock enable or a direct data input to the flip-flop. The clock enable would be implemented in the LUT function by feeding back the output from the flip-flop. This occupies three of the four available LUT inputs–one for the feedback, one for the clock enable, and one for the flip-flop input. Only one LUT input remains to perform a useful function.

The lack of a clock enable in the FLEX architecture can be overcome by gating clock signals, which is a poor design practice because of possible glitches on the clock line. Also, the limited number of clock lines available within the FLEX device constrains this approach.

Multiplexed Block Outputs

The XC5200 logic cell has an added advantage for some applications. The outputs of two logic cells can be multiplexed together to provide larger, more complex functions. The multiplexer that performs this function is shown in the top, left-hand corner of Figure 1a. All see page 8 in [1] for more details.

Multiplexing the logic cell outputs provides a fast, efficient mechanism to build some common logic functions such as a four-to-one multiplexer or any arbitrary function of five inputs.

Table 4. Implementing a Four-To-One Multiplexer or Any Arbitrary Five-input Logic Function

	XC5200	FLEX 8000A
Logic cells required	***2***	3
Layers of logic	***1***	2

Note: ***Bolded italic*** text indicates advantage

Input/Output Block Comparison

Figure 3 illustrates the I/O blocks of the XC5200 and FLEX architectures. As with the logic block, the XC5200 I/O block offers superior features and flexibility.

Both architectures offer three-state output buffers and programmable slew-rate controls. However, Table 5 through Table 8 demonstrate the significant differences.

Table 5. Basic I/O Capabilities

Feature	XC5200	FLEX 8000A
Slew-rate control	Yes	Yes
Programmable Pull-up resistor	***Yes***	No
Invertable inputs and outputs	***Yes***	No

Note: ***Bolded italic*** text indicates advantage

Input/Output Registers

Each FLEX I/O block has a single edge-triggered flip-flop that can be used as either an input or an output register. By contrast, the XC5200 provides a high-speed direct connection between the I/O block and the nearest logic cell. The XC5200 approach thus offers additional control signals on input or output registers as shown in Table 6.

I/O Performance

Calculating I/O Performance

The individual I/O timing elements are specified in the data sheet for both the XC5200 and FLEX. However, pin-to-pin timing must be considered when interfacing the device to the rest of the system. The XC5200 pin-to-pin I/O timing values are specified and guaranteed, as per the XC5200 data sheet.

The FLEX I/O timing values, however, must be calculated from formulas found in Altera's 1995 Data Book (See Notes on page 13 for methods used to calculate the values shown in Table 7).

Table 6. Input or Output Register Options

Feature	XC5200	FLEX 8000A
Both input and output register	***Yes***	Yes
I/O register options	**Flip-flop or latch**	Flip-flop
Zero hold time	***Yes***	No
Clock inversion	***Yes***	No
Clock sources	***Independent – any I/O or LC***	2* total
Clock enable	***Yes***	No
Asynchronous clear	***Yes***	Yes
Clear sources	***Independent – any I/O or LC***	2 total

Note: ***Bolded italic*** text indicates advantage
*Clock line shared with OE1 output-enable signal

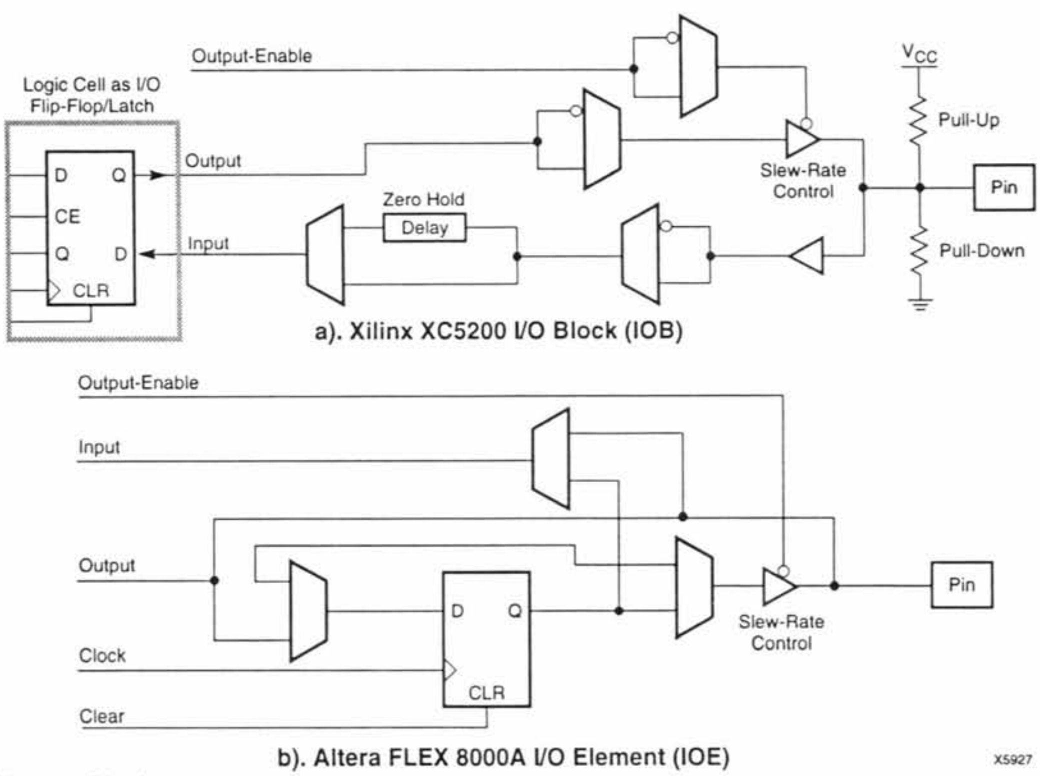


Figure 3. Input/Output Blocks

Table 7. I/O Timing Parameter Comparison

Parameter	Device	Speed Grade -6	Speed Grade -5	Speed Grade -4
Clock-to-clock (fast mode), C=50 pF	XC5210	17.2	15.4	14.2
	EPF81188A	***14.8***	***11.8***	***10.9***
Clock-to-output (slew-rate), C=50 pF	XC5210	21.7	19.0	17.1
	EPF81188A	***18.8***	***15.3***	***14.4***
Set-up (with hold)	XC5210	2.2	1.5	1.2
	EPF81188A	neg.	neg.	neg.
Hold (non-zero)	XC5210	***3.5***	***3.0***	***2.8***
	EPF81188A	8.2	6.2	5.7
Set-up (zero hold)	XC5210	***8.5***	***7.4***	***6.6***
	EPF81188A	Not available		
Hold (zero hold)	XC5200	***0***	***0***	***0***
	FLEX	Not available		

Note: ***Bolded italic*** text indicates advantage

Input Hold Time

One key advantage of the XC5200 architecture simplifies I/O timing. As shown in Figure 3a, the XC5200 I/O block contains an additional, programmable delay in the input path. This delay closely matches the global buffer routing delay and effectively provides zero hold time on input registers implemented in the outer ring of logic cells.

Zero hold time simplifies I/O timing requirements and allows data and clock to change simultaneously. If hold time is non-zero, then additional, external logic needs to be added or data race conditions can occur, regardless of the system clock frequency.

FLEX devices do not contain this programmable delay. Consequently, all FLEX I/O registers have significant hold time requirements, further complicating interface timing.

Output Enables

Another key difference between the XC5200 and FLEX families is the number of independent output enable controls for I/O pins. Table 8 describes the output enable features for each architecture. The source for an output enable on the XC5200 can come from any internal or external source. There can be as many independent output enables as output pins in the design.

Table 8. Output Enable Features

Features	XC5200	FLEX 8000A
Maximum global output enable signals	***5****	4
Output enable sources	***Independent – any I/O or LC***	Shared global signals
Output enable polarity control	***Yes***	No
Number of output enable signals per device	***Up to 1 per I/O pin. Unlimited***	4 total in most 10 in EPF81500A

Note: ***Bolded italic*** text indicates advantage

* Includes device-wide global three-state control.

In contrast, the FLEX architecture supports only four output enables, except for the EPF81500A which supports ten. Two of the four OE signals are shared with other control signals. Of these signals, OE1 is shared with one of the global clock lines, CLK1. So, if the control signal is used as a global clock, then it is not available as an output enable, and vice versa. Similarly, OE0 is shared with a global clear signal, called CLR1.

Output enables are used in a wide variety of applications, especially for bus interfaces. One particular application that stresses the number of output enables is an interrupt controller. Typically, the outputs from an interrupt controller drive open-collector or open-drain outputs to the rest of the system. Open-drain outputs are created by connecting the input of the three-state output buffer to ground and driving the output-enable pin as shown in Figure 4.

Table 9. Summary of I/O Block Features

Feature	XC5200	FLEX 8000A
Output drive	-8/8 mA	-4/12 mA
Slew-rate control	Yes	Yes
Pull-up resistor	***Yes***	No
Pull-down resistor	***Yes***	No
Flexible I/O routing for pin-locking flexibility	***Yes***	No
Selectable TTL or CMOS input thresholds	***Yes***	No
JTAG on all members	***Yes***	No
ESD protection	***>5 kV***	>2kV

Note: ***Bolded italic*** text indicates advantage

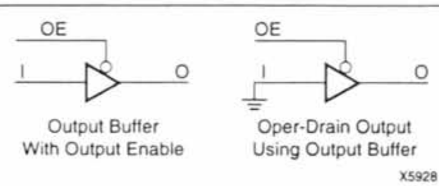


Figure 4. Building Open-drain Outputs Using Output Buffers With Enable.

Figure 5 shows a small example circuit using open-drain outputs. Such circuits are commonly found in interrupt request circuits used in most bus standards.

FLEX devices do not have enough output-enables to implement this simple circuit. By contrast, the circuit in Figure 5 fits in every member of the XC5200 family–including the smallest. However, the only FLEX device capable of implementing this circuit is the EPF81500A.

Boundary Scan (JTAG) Support

Industry-standard JTAG boundary scan logic (IEEE 1149.1) is integrated into all XC5200 devices. The JTAG logic eases system and board testing when using Xilinx programmable logic. It also provides a means to program XC5200 devices. JTAG is available on all members of the XC5200 family–from smallest to largest. JTAG not available on all members of the FLEX family.

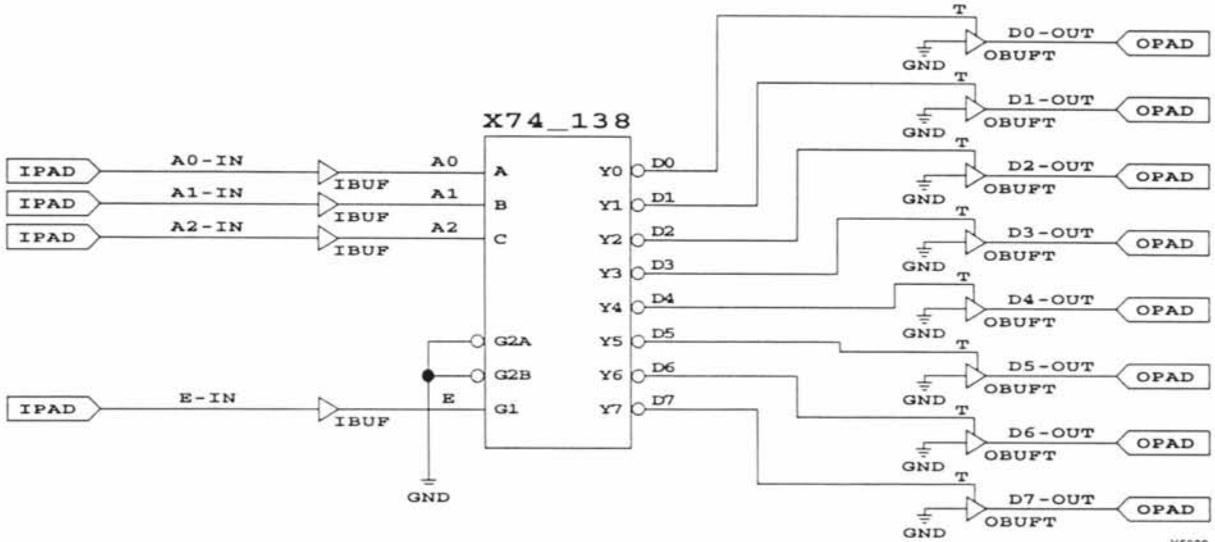


Figure 5. Interrupt Decoder Illustrating Advantage of Independent Output Enables.

Fixed Pinout Flexibility

One inevitable aspect of design is change. A typical design goes through numerous revisions before going to production.

The ideal programmable logic device tolerates significant design changes while maintaining a fixed pinout. Without this capability, a printed circuit board needs to be modified every time the logic inside the device changes. Each design modification adds cost and time to the project.

The XC5200 and FLEX families have widely different capabilities to accommodate such changes. The XC5200 is designed for change while the FLEX family is overly rigid.

XC5200 VersaRing™ – Fixed Pinout Flexibility

The exterior ring of the Xilinx XC5200 architecture contains extra routing resources. This additional routing provides the ability to route to a fixed PC board pinout between design revisions.

An independent study by the University of Toronto studied the effect of I/O pin placement on the routability and speed of FPGAs. The study examined the Xilinx XC4000 and FLEX 8000A architectures. The XC5200 was not available at the time of the study.

The study showed that the XC4000 devices were able to route to a fixed pinout in all 16 benchmark cases (though fixed pin assignment does impact routability because the amount of routing resources used was increased).

The XC5200 VersaRing, shown in Figure 6, is even more flexible than the edge routing around the XC4000 devices. In a more extensive Xilinx study of over 100 customer designs, the XC5200 was able to route to a fixed pinout in all cases. In over 60% of the designs, the effect of a non-optimal pin placement had little or no effect on performance.

Please note that Xilinx still recommends against pre-assigning I/O pins. Allowing the place and route software to create the initial pin placement makes it easier to route to a fixed pinout on subsequent design iterations.

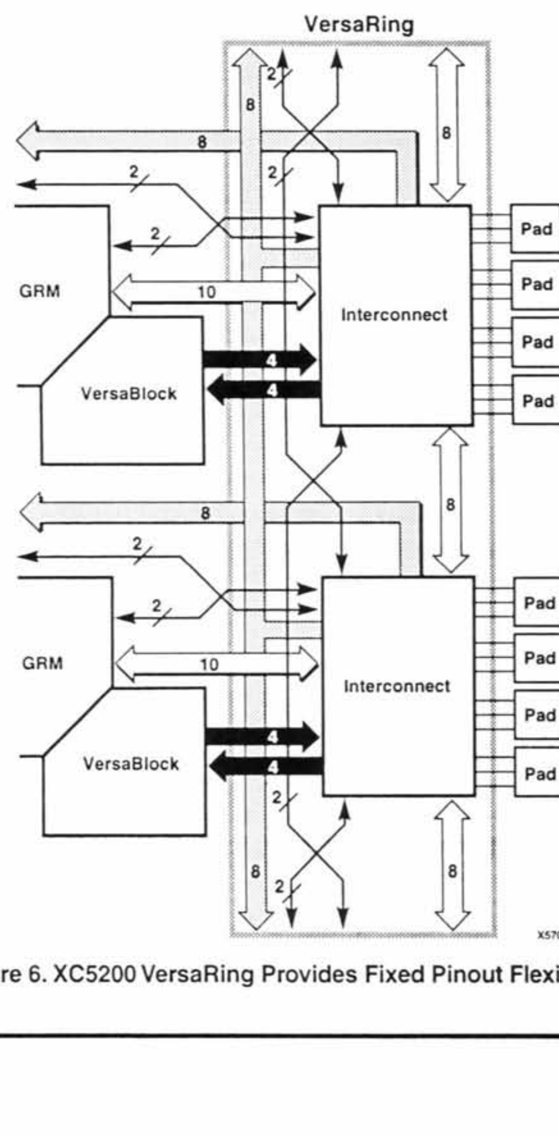


Figure 6. XC5200 VersaRing Provides Fixed Pinout Flexibility

Altera FLEX Routing Inflexibility

Because of the FLEX routing architecture, FLEX devices cannot tolerate minor internal logic placement changes with fixed I/O placement.

The architecture exhibits serious limitations on the total number of I/O pins that a FLEX logic element (LE) can reach. The FLEX data sheet lacks specific connectivity information that describes the actual number of output pins that each LE can reach. However, exhaustive tests show that a given LE can only directly drive six possible I/O pins. On the EPF8452A device, this limitation means that an LE can only drive 5% of the available I/O pins.

Two separate studies indicate a problem routing FLEX 8000A designs to a fixed pinout.

University of Toronto Study

In the University of Toronto study, the FLEX 8000A devices failed to route in 21% of the benchmark cases as shown in Table 10. The study further concluded that the FLEX architecture was susceptible to routing failures in designs where the I/O pin or logic utilization was close to 100%.

Table 10. Summary of University of Toronto Study on the Effect of Fixed Pinout

Devices	Bench-marks	Per Cent Routed to Fixed Pinout	Per Cent Failed to Route to Fixed Pinout
XC4000	16	100%	0%
FLEX 8000A	14	79%	21%

Xilinx Study

Xilinx studied the effect of pinout and routability using a test suite of 57 designs. These designs use no system-level features such as on-chip RAM, tri-state buffers, or I/O flip-flops. All 57 designs included only random logic and I/O.

For each test case, the same randomly-selected pin ordering was applied to an EPF81188, an EPF81500, an XC5210, an XC5215, and an XC4013 device. The designs were then processed using the appropriate placement and routing software without any constraints other than the pin ordering.

All 57 of the designs placed and routed to completion in the both the XC5215 and XC4013. All but two of the designs routed successfully in an XC5210. The 17 of 57 failed designs are summarized in Table 11 and Table 12. The failed design are shaded. The values in parentheses indicate the number of Logic Elements or Logic Cells required to implement the design.

Table 11. 17 of 57 Test Designs That Failed to Route in FLEX 8000A or XC5200. (Failed cases are shaded, number of LCs or LEs in design shown in parentheses).

No.	Altera FLEX 8000A		Xilinx XC5200	
	EPF81188	EPF81500	XC5210	XC5215
1	FAIL (238)	FAIL (238)	OK (239)	OK (239)
2	FAIL (658)	FAIL (658)	OK (807)	OK (807)
3	FAIL (249)	FAIL (249)	OK (212)	OK (212)
4	FAIL (611)	FAIL (611)	FAIL (744)	OK (744)
5	FAIL (238)	OK (238)	OK (234)	OK (234)
6	FAIL (524)	OK (524)	OK (588)	OK (588)
7	FAIL (429)	OK (429)	OK (423)	OK (423)
8	FAIL (445)	OK (445)	OK (268)	OK (268)
9	FAIL (510)	OK (510)	OK (459)	OK (459)
10	FAIL (548)	OK (548)	OK (584)	OK (584)
11	FAIL (589)	OK (589)	OK (599)	OK (599)
12	FAIL (458)	OK (458)	OK (388)	OK (388)
13	FAIL (683)	OK (683)	OK (621)	OK (621)
14	FAIL (495)	OK (495)	OK (534)	OK (534)
15	FAIL (668)	OK (668)	OK (702)	OK (702)
16	FAIL (729)	OK (729)	OK (969)	OK (969)
17	OK (335)	OK (335)	FAIL (1,027)	OK (1,027)

Table 12. Summary of 57 Routing Test Designs

Device	Per Cent Successfully Routed	Per Cent Failed to Route
EPF81188A	72% (41/57)	28% (16/57)
EPF81500A	93% (53/57)	7% (4/57)
XC5210	96% (55/57)	4% (2/57)
XC5212	***100%*** (57/57)	***0%*** (0/57)
XC4013	***100%*** (57/57)	***0%*** (0/57)

Interconnect Comparison

As with other aspects of the architectures, there are basic similarities between the XC5200 and FLEX interconnect structures, but also many differences. Both architectures employ a hierarchy of routing resources. There are two main types of interconnect in FLEX devices and six types in the XC5200 devices, distinguished by the relative length and connectivity of their segments.

XC5200 General Interconnect

Figure 7 and Figure 8 shows the routing hierarchy of the XC5200 family.

Advanced simulation tools were used during the development of the XC5200 architecture to determine the optimal level of routing resources required. The XC5200 family contains six levels of interconnect hierarchy –single-length lines, double-length lines, and long lines in the General Routing Matrix (GRM) plus direct connects, Local Interconnect Matrix (LIM), and logic-cell feedthroughs within each VersaBlock. Throughout the XC5200 interconnect, an efficient multiplexing scheme, in combination with triple-layer metal fabrication, was used to improve the overall efficiency of silicon usage.

Four XC5200 LCs and their associated interconnect, are grouped together to form a Configurable Logic Block (CLB), also called a VersaBlock. The LIM provides 100% connectivity of the inputs and outputs of each LC in a given CLB. The benefit of the LIM is that no general routing resources are required to connect feedback paths within a CLB. The LIM connects to the General Routing Matrix (GRM) via 24 bi-directional nodes.

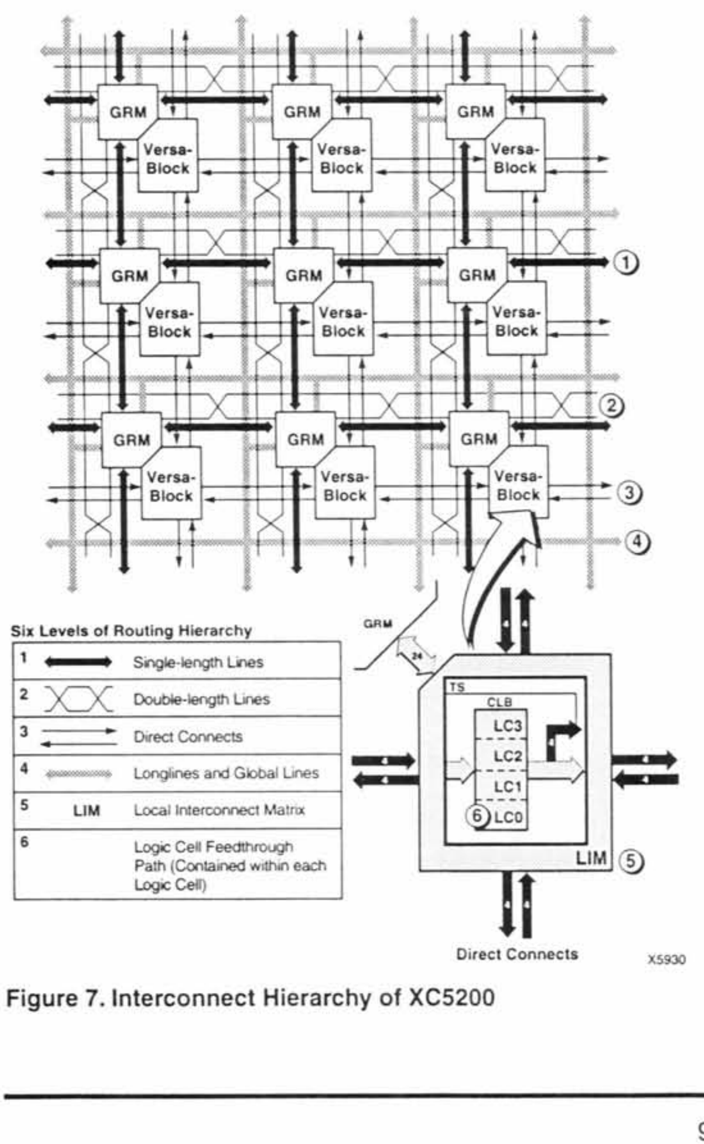


Figure 7. Interconnect Hierarchy of XC5200

The direct connects allow immediate connections to neighboring CLBs, once again without using any of the general interconnect. These two layers of local routing resource improve the granularity of the architecture, effectively making the XC5200 family a "sea of logic" cells. Each VersaBlock has four three-state buffers that share a common enable line and directly drive horizontal long lines, creating robust on-chip bussing capability. The VersaBlock allows fast, local implementation of logic functions, effectively implementing user designs in a hierarchical fashion. These resources also minimize local routing congestion and improve the efficiency of the general interconnect, which is used for connecting larger groups of logic. It is this combination of both fine-grain and coarse-grain architecture attributes that maximize logic utilization in the XC5200 family. This symmetrical structure takes full advantage of the third metal layer, freeing the placement software to pack user logic optimally with minimal routing restrictions.

The GRM is functionally similar to the switch matrices found in other architectures, but it is novel in its tight coupling to the logic resources contained in the VersaBlocks.

FLEX General Interconnect

Figure 9 shows the routing architecture of the FLEX 8000A family.

The FLEX architecture contains two basic types of routing: the local interconnect within a Logic Array Block (LAB), and long line interconnects running in the rows and columns between the LABs. The local interconnect within a LAB has connections to all four inputs and the single output of each of the eight LEs in that LAB, and also connects to the horizontal long lines. LE outputs can also be driven out to the horizontal or vertical long lines.

The FLEX architecture is very asymmetrical. The local interconnect in each LAB connects only to the horizontal long lines in the row above it, and LE outputs connect to long lines only in the row above or column to the right of the LAB. Thus, the architecture heavily favors a horizontal data flow that confines I/O placement.

The FLEX LABs themselves are arranged asymmetrically. For example, the EPF8452A has a 2-by-21 array of LABs while the EPF81188A has a 6-by-21 array. In large FLEX devices, there are far more long lines in the horizontal direction than in the vertical direction (1,008 horizontal lines vs. 336 vertical lines in the EPF81188A). This asymmetry has two effects on the design, in general:

1. I/O pins at the top and bottom of a FLEX device have limited connectivity. I/O-intensive designs are difficult to route. Furthermore, it is difficult to route to a fixed pinout between design iterations in the FLEX architec-

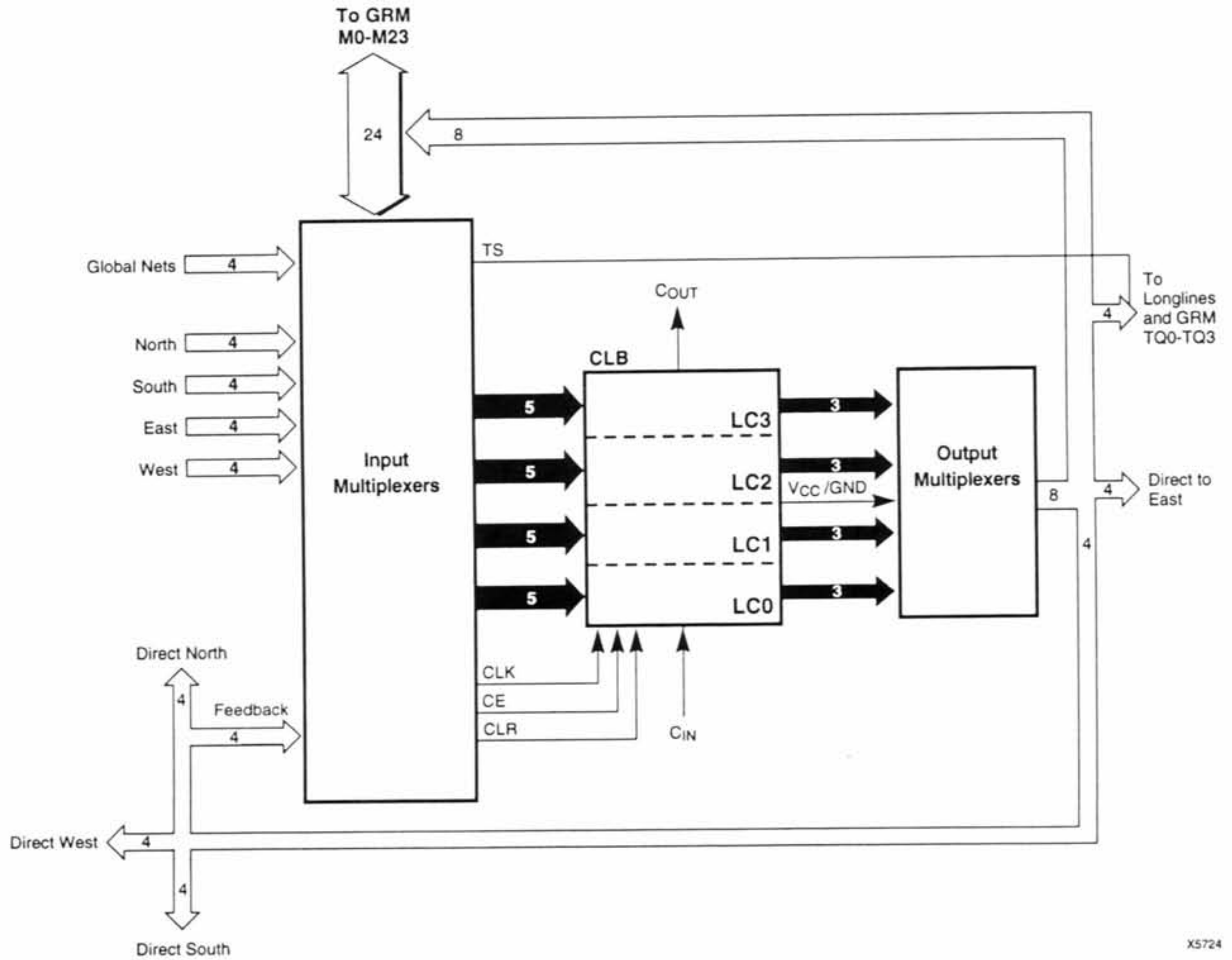


Figure 8. Details of Routing Within a VersaBlock

ture. The I/O placement may change between design iterations forcing changes to the printed circuit board.

2. Designs must be partitioned into rows for optimal performance. Complex designs that do not partition easily can be very difficult, if not impossible, to place and route. The relative shortage of vertical routing resources will be a more significant factor in the higher-density family members than in the smaller FLEX devices. This is demonstrated using Altera's PREP designs for multiple 16-bit counters. The software was able to utilize 81% of the FLEX device when manually-entered CLIQUE statements forced this simple design into left-to-right rows. However, utilization dropped to only 46% when the software was left to perform the task automatically, without the CLIQUE statements. In complex designs, CLIQUE statements cannot compensate for the severely limited FLEX interconnect architecture.

The interconnect limitations affect overall utilization. Designs exceeding 80% utilization are notoriously difficult to place and route in the FLEX architecture, especially for the larger members of the family.

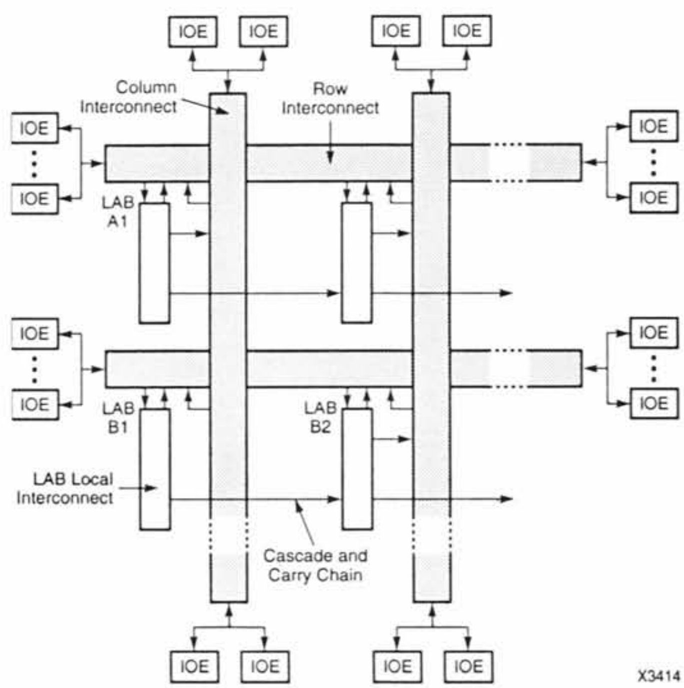


Figure 9. FLEX 8000A Routing Architecture

One positive consequence of the limited routing architecture is faster place and route run-times for simple, lower-utilization designs. Because there is less placement and routing flexibility in FLEX, the Altera software can fit low-utilization designs extremely quickly. However, at high utilization, runtimes increase dramatically.

Internal Three-State Capability to Build On-Chip Busses

One feature available only within the XC5200 family is on-chip three-state capability as shown in Figure 10. This capability provides fast and efficient bi-directional bussing within the device.

The FLEX architecture cannot implement bi-directional busses directly. They must be built using unidirectional busses and multiplexers. This approach consumes double the routing resources and cannibalizes logic resources to control data flow.

The positive affect of internal three-state capability can be graphically demonstrated using a simple example. The circuit shown in Figure 11 consists of eight 16-bit data registers connected via a common 16-bit bus. The data from the I/O pins can be individually read and written from any of the eight registers. In the XC5200 architecture, all eight registers are accessible via a single 16-bit bidirectional data bus using internal three-state buffers as shown in Figure 11.

In the FLEX architecture, by contrast, all eight registers would be written from one unidirectional bus and read via a multiplexer using another bus as shown in Figure 12.

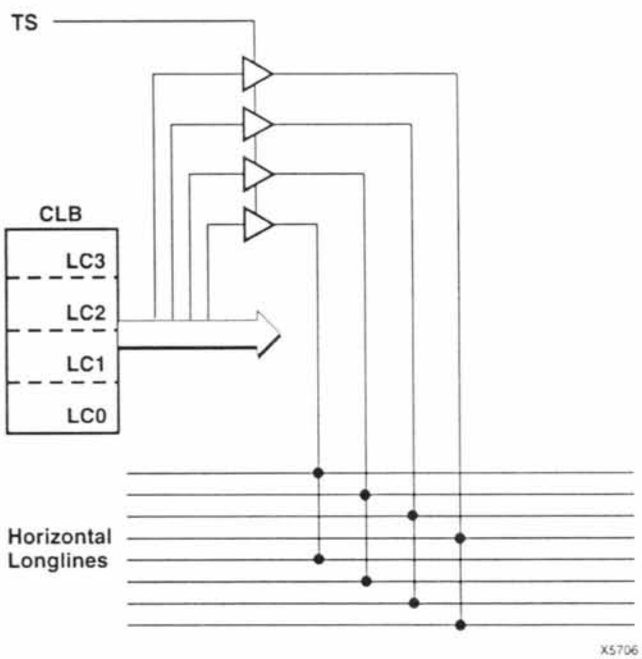


Figure 10. Three-state Capability of XC5200 Logic Element

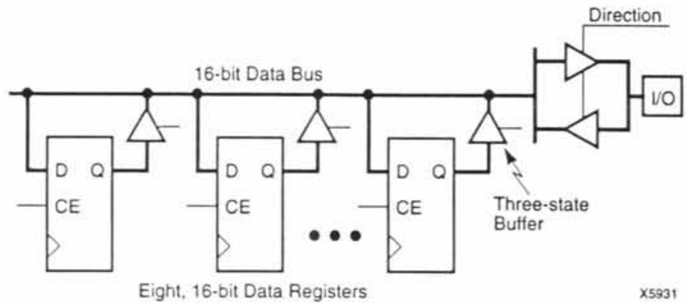


Figure 11. Eight 16-bit Registers Share An Internal Bidirectional Data Bus in XC5200 FPGA

Writing the registers requires a clock enable signal, which consumes an extra input on the Altera LUT (see related *Clock Enable* section on page 4). Reading the registers requires a 16-bit, 8-to-1 multiplexer on the output. This overhead consumes additional routing, decreases performance, and lowers overall utilization. The effect on larger devices is even more dramatic. Typically, larger devices need to implement wider data busses (i.e.–32-bits wide instead of 16-bits) with even more bus sources.

Global Signals

Both the XC5200 and FLEX families have four high-fanout global signals that may be used for clock or control signal distribution. However, there are restrictions on how the global lines are used in FLEX 8000A:

- One line is dedicated to clocking
- One line is dedicated to presetting/clearing flip-flops
- One line is shared between clocking and output enables
- One line is shared between presetting/clearing flip-flops and output enables.

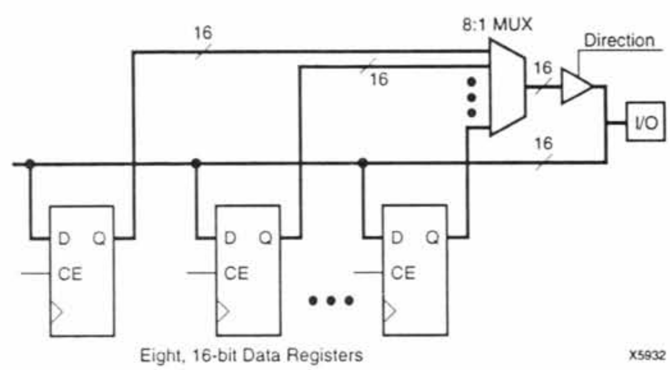


Figure 12. Eight 16-bit Registers Require Separate Unidirectional Busses in FLEX FPGAs

By contrast, the XC5200 global signals can drive practically any logic function as shown in Table 13.

Table 13. Features of Global Signals

Feature	XC5200	FLEX 8000A
Global signals	4	4
Buffers drive ...	**Any logic, control, or clock input**	Flip-flop clocks Flip-flop preset or clear Output enables

Note: **Bolded** text indicates advantage

Global Resources

The XC5200 architecture also provides some unique resources not available in the FLEX 8000A architecture including:

Global Reset – provides a chip-wide asynchronous reset. Can be sourced from any input or logic cell.

Global Three-state – provides a way to tri-state all of the device outputs; useful for device or board testing.

Internal Oscillator – provides a flexible clock source for applications such as watchdogs, simple timers, etc.

Readback – provides a way to readback the internal register values; especially useful with the XChecker download/readback cable for debugging.

Packaging Options and Flexibility

Both the XC5200 and FLEX 8000A offer a variety of packaging options. However, as shown in Table 14, the XC5200 provides additional flexibility for the design engineer including:

- **A wider density range in a given package style.** Choose the package and the capacity for the specific application.
- **Footprint compatibility between other members of the XC5200 family** in the same package style (indicated by ↕ in Table 14). If a design grows beyond a specific device, there are probably bigger devices in the same footprint. Likewise, a design can use a larger device for easier prototyping while the production design can be optimized to fit into a smaller, pin-compatible device.
- **Footprint compatibility with other Xilinx device families** including the XC4000/A/D/E/H SRAM-based FPGAs and the XC8100 OTP FPGAs (indicated by both ↕ and → in Table 14). A → in the table indicates that no other XC5200 device has a similar footprint. However, other XC4000 or XC8100 family members are pin-compatible. This allows a designer to choose specific Xilinx device family attributes without having to modify the printed circuit board. These device attributes include:

Table 14. Packaging and Footprint Compatibility of XC5200 Family (August 1995)

Pins	84	100	100	144	156	160	191	208	223	240	299	304
Style	PLCC	PQFP	VQFP	TQFP	PGA	PQFP	PGA	PQFP	PGA	PQFP	PGA	HQFP
XC5202	↕	↕	↕	↕	↕							
XC5204	↕	↕	↕	↕	↕	↕						
XC5206	↕	↕	↕	↕		↕	→	↕				
XC5210	↕			↕		↕		↕	→	↕		
XC5212								↕		↕	→	→

Table 15. Packaging and Footprint Compatibility of FLEX 8000A Family (March 1995)

Pins	84	100	160	160	192	208	225	232	240	280	304
Style	PLCC	TQFP	PGA	PQFP	PGA	RQFP	BGA	PGA	RQFP	PGA	RQFP
EPF8282A	X	X									
EPF8452A	↕		X	X							
EPF8636A	↕			X	↕	X					
EPF8820A				X	↕	X	X				
EPF81188A						X		X	X		
EPF81500A									X	X	X

LEGEND:

↕ Footprint compatibility between various devices in the family. For Xilinx devices, also indicates footprint compatibility between XC4000, XC5200, and XC8100 families.

→ No direct footprint compatibility between various devices in the family. However, indicates footprint compatibility between Xilinx XC4000, XC5200, and XC8100 families.

X No footprint compatibility.

- **XC5200** – General logic applications. Provides lowest cost per gate of any programmable logic device. SRAM-based. In-system reprogrammable.
- **XC4000/A/D/E** – General, higher-performance logic applications. PCI compliant. On-chip RAM for building FIFOs, register stores, etc. Higher output drive capability. SRAM-based. In-system reprogrammable.
- **XC8100** – General logic applications. Logic synthesis friendly design flow. Higher output drive capability. Single-chip solution. One-time programmable (OTP). Extremely high design security.

The pin-locking flexibility provide by the XC5200 VersaRing makes these benefits even more meaningful.

Table 15 shows the package and footprint compatibility of FLEX 8000A devices. Again, a ↕ indicates a footprint compatible package offering. Note that there are only two places in the table where there is footprint compatibility between two device sizes. A 'X' indicates a unique package footprint, incompatible with other package footprints.

Benefits of Common Footprint

There are various scenarios where a common device footprint is a big advantage. These include:

- **The design grows** to exceed the gate density of a device but the I/O remains the same. The design can be migrated to a larger device without re-spinning the printed circuit board layout, saving both time and money.
- **The design is optimized into a smaller device** through extra engineering. The smaller device costs less reducing the price of the board.
- **Inventory flexibility.** Use devices on hand for either prototyping or initial production.

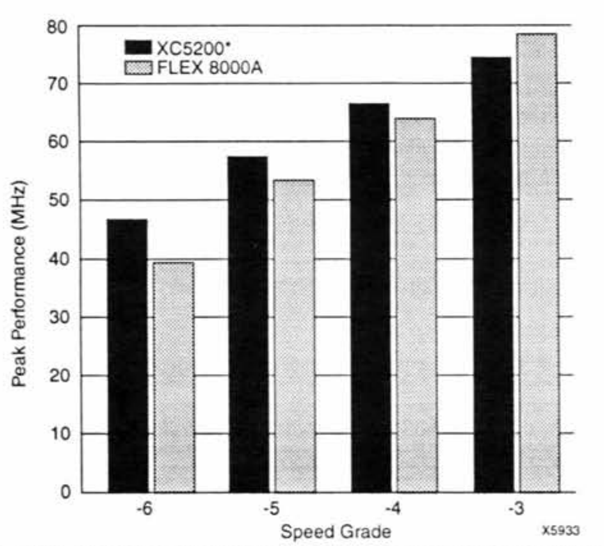


Figure 13. Relative Peak Performance of XC5200 and FLEX 8000A FPGAs. (*Note that -3 value for the XC5200 is a design migrated to the footprint-compatible XC4000D-3).

- **Reduce manufacturing line downtime** caused by a product shortage. Should the device designed into the product have long lead-times or become unavailable, you have various options:
  - Use a larger device from the same FPGA family.
  - Use a similar-sized device from another footprint-compatible FPGA family.

Benchmarks

Various industry-wide benchmarks provide a rough indication of speed and density of a device. The benchmarks indicated below are based on the PREP test suite. All nine benchmarks, except #5, where implemented for the XC5200 devices. The results for benchmark #5 was not available as of this writing. The results for the FLEX 8000A are from the PREP World-Wide Web site [4].

Performance

As shown in Figure 13, the XC5200 family is roughly equivalent in performance to the FLEX 8000A for a similar speed grade. For faster designs, a XC5200 design may be migrated to the footprint-compatible XC4000D/E-3 for added performance.

Density

Most of the existing density benchmark schemes are designed around the lowest common denominator of functions–those available to all programmable logic devices on the market. Xilinx devices have a variety of system-level features not found in competing devices. These features, like internal three state, significantly boost both density and performance in system level designs.

Consequently, the values shown in Figure 14 show the effective density of the XC5200 for "gates-only" applications, i.e., those not using any special features. For

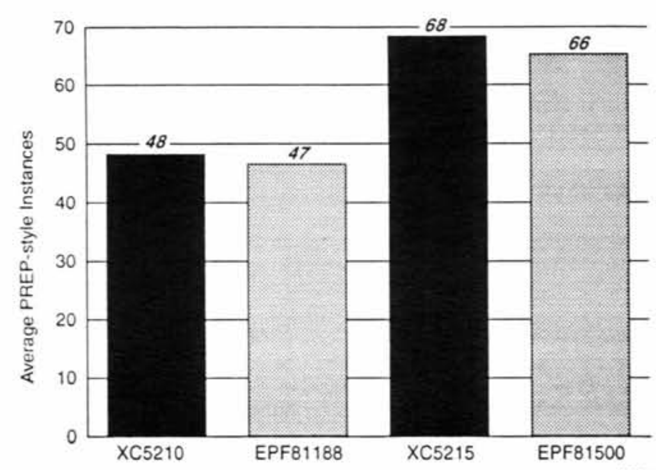


Figure 14. Average Density Benchmarks of XC5200 and FLEX 8000A. (PREP benchmarks #5 not included, data not available for XC5200).

example, benchmarks #6 through #8 connect in a very artificial way. Most designs do not have loadable binary counters being loaded from the inputs of another loadable binary counter. Usually, these functions would be bussed together. Consequently, the XC5200 numbers would be significantly higher.

See also the results presented earlier in the *Altera FLEX Routing Inflexibility* section. Those results indicate that an XC5210 is sufficiently large enough to replace nearly every EPF81188 device and some EPF81500 designs.

Real-World Design Conversions

The Xilinx Design Center has converted a number of FLEX 8000A devices into Xilinx XC5200 devices. Most of the conversions were done automatically using Exemplar Logic's CORE logic synthesis package. Unfortunately, this approach does not automatically convert between the FLEX 8000A carry logic and the XC5200 carry logic, nor does it use the XC5200's internal tri-state capability.

Nearly every FLEX 8000A converted into a smaller or comparably-size XC5200 device as shown in Table 16. If re-entered, bus-oriented FLEX 8000A designs fit into much smaller XC5200 devices due to the XC5200's internal tri-state capability. FLEX 8000A-3 and -2 applications required an XC4000E-3 or XC3100A-2 device for higher performance. The XC5200 designs that did not automatically meet the customer's performance requirements could meet the requirements with Xilinx-specific optimization. In some cases, this required re-entering the design specifically for Xilinx.

Table 16. Examples of FLEX 8000A Designs Converted to XC5200 Devices by Xilinx Design Center

FLEX 8000A Device	→	XC5200 Device	Method/ Comment
EPF8188A	→	XC5206	Exemplar
EPF8452A	→	XC5206	Exemplar
EPF81188A	→	XC5206	Exemplar
EPF81188A	→	XC5210	Exemplar
EPF81500A-2	→	XC5210 XC4010D-3	Used XC4010D-3 for better speed
EPF81188A	→	XC5210	Needed PQ208 package
EPF8636A-2	→	XC5204 XC4005E-3 XC3164A-1	Used XC3164A-1 for better speed
EPF8282A	→	XC5202	Schematics
EPF81188	→	XC5206	92% utilization

The Xilinx Design Center provides design conversions as a service. Conversions can be arranged by calling the local Xilinx sales office. For best quality conversions, the following files are requested (if applicable):

- `*.tdf` AHDL files
- `*.gdf` schematic files
- `design.rpt` report files
- `design.edo` EDIF output files

High-Volume Cost Reduction Strategies

The XC5200 FPGAs provide low production costs for very high volumes. For even higher production volumes, Xilinx provides a path approaching ASIC-like cost levels.

All Xilinx FPGAs have a proven, high-volume cost reduction path. Once a design has stabilized and is in volume production, the costs can be reduced by migrating the design to a Xilinx HardWire gate array. A HardWire gate array provides:

- A low-risk, high-volume, low-cost solution.
- No need to write simulation or test vectors. 100% fault coverage is ensured through automatic test vector generation and internal test circuits.
- 100% footprint-, timing-, and design-compatibility with the corresponding FPGA device.
- Fast time-to-volume.
- Little, or no engineering involvement required.

The Xilinx HardWire gate array design flow is completely owned and controlled by Xilinx. All Xilinx HardWire devices are manufactured at the same facilities as Xilinx FPGAs. Consequently, Xilinx HardWire gate arrays have the same high quality plus there is no need to qualify a second production facility.

Altera offers an FPGA to gate array conversion service that utilizes a third-party gate array vendor. This type of flow requires significantly more effort to generate simulation and test vectors to ensure 100% fault coverage.

Notes

Calculating I/O Performance for FLEX 8000A FPGAs

IOE Clock-to-Output Time

$t_{DIN\_IO} + t_{IOC} + t_{IOCO} + t_{OD1}$ as described on page 484 in [2]. Note that t_{OD1} refers to the output buffer delay for outputs without slew-rate limiting, while tOD3 refers to outputs with slew-rate limiting.

IOE Input Set-Up Time

t_{IN} - (t_{DIN_IO} + t_{IOC}) + t_{IOSU} as described on page 483 in [2].

Input Hold Time

(t_{DIN_IO} + t_{IOC}) - t_{IN} + t_{IOH} as described on page 483 in [2]. Note that there is an error in the Altera equation shown in [2]. The I/O clock value is t_{DIN_IO}, not t_{DIN_C} and the hold time parameter is t_{IOH}, not t_H.

Derating for 50 pF Loads

Data sheet specifications for I/O performance should be examined carefully. While Xilinx conforms to industry standards by characterizing I/O performance using 50 pF loads, the FLEX data sheet provides timing for 35 pF loads, making direct comparisons difficult. While no specific derating data is available for FLEX, a 0.75 ns value was added to the FLEX numbers so that both XC5200 and FLEX could be compared with similar loads (C=50 pF). The 0.75 ns value was derived by assuming an additional 0.05 ns/pF derating factor.

References

[1] Xilinx, Inc., "XC5200 Logic Cell Array Family Technical Data", Version 2.0, May 1995.

[2] Altera Corporation, "FLEX 8000A Programmable Logic Device Family" data sheet, 1995 Data Book, pages 37 through 93.

[3] Khalid, Mohammed A.S. and Rose, Jonathan, "The Effect of Fixed I/O Pin Positioning on the Routability and Speed of FPGAs", in *Proceeding of The 3rd Canadian Workshop on Field-Programmable Devices (FPD'95)*, 1995, pages 94 through 102.

[4] Programmable Electronics Performance Corp. (PREP) Web site at `http://www.prep.org`.

North America

Corporate Headquarters

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
Tel: 1 (408) 559-7778
Fax: 1 (408) 559-7114
NET: hotline@xilinx.com
WEB: http//www.xilinx.com

XILINX®
The Programmable Logic Company℠

Printed in U.S.A.

PN 0010270-01