Comparing Performance of High-Density PLDs

May 1998, ver. 1 Application Note 97

Introduction

When selecting a programmable logic device (PLD), most designers compare density, price, and performance to decide which device meets their design requirements. Designers can compare density and price through industry-standard logic cell counts and comparative price quotes. Comparing performance, however, is a more challenging process. To determine performance, designers must consider numerous factors, such as signal routing, logic complexity, type of memory interface, and fan-out. However, by breaking down performance into individual metrics, designers can better compare device architectures between different vendors.

This application note summarizes the results of performance tests comparing the fastest Altera® FLEX® 10KA device against the fastest Xilinx XC4000XL device using benchmarks published in Xilinx's *Application Brief XBRF015 (Speed Metrics for High-Performance FPGAs).*

Lab Setup

In the experiments, designs were implemented in Altera EPF10K100A-1 devices using the MAX+PLUS® II version 8.2 development tool and a combination of schematic and Altera Hardware Description Language (AHDL) design entry. The results of these experiments were compared to the results published in Xilinx's *Application Brief XBRF015 (Speed Metrics for High-Performance FPGAs)*. In Xilinx's experiments, the designs were written in VHDL, synthesized in the Synopsys FPGA Express software, and compiled in the XACT Step M1 version 3.7 software. All benchmarks discussed in this application note represent maximum frequency.

Performance Metrics

Experiments were performed using the benchmarks listed below.

- I/O Frequency
- Average Routing Delay
- *N*-Level Combinatorial Logic
- *N*-to-1 Multiplexer
- *N*-Bit AND Gates
- Chained Adders

I/O Frequency

The I/O frequency (f_{IOEXT}) benchmark measures the maximum frequency that data can be transferred to and from a programmable logic device (PLD) with its inputs and outputs registered. The equation for calculating f_{IOEXT} is shown below, assuming the hold time (t_H) is zero:

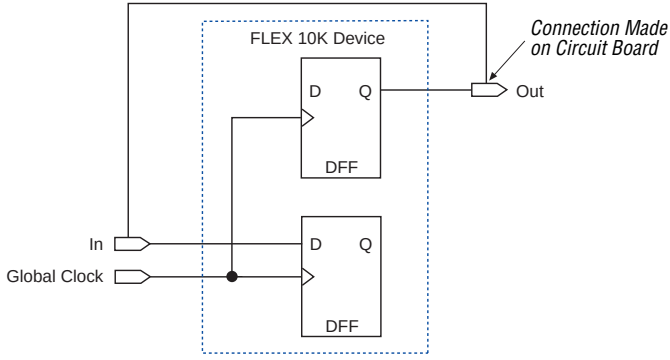
$$f_{IOEXT} = \frac{1}{t_{SU} + t_{CO}}$$

Where: t_{SU} = Input setup delay
t_{CO} = Clock-to-output delay

In this experiment, the t_{CO} and t_{SU} delays were measured on the rising edge of a dedicated global clock signal feeding an I/O cell register, as shown in Figure 1.

Figure 1. f_{IOEXT} Circuit

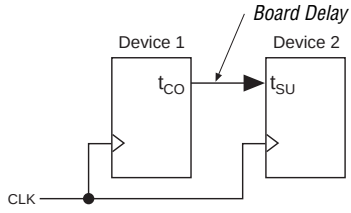
The f_{IOEXT} circuit used to measure the t_{CO} and t_{SU} delays of a device.



To accurately determine the f_{IOEXT} between two devices on a printed circuit board (PCB), designers need the individual t_{SU} and t_{CO} values for each device. Designers can then determine the system's frequency by adding the t_{CO} of device 1, the t_{SU} of device 2, and the board delay (see Figure 2.) Thus, a shorter t_{SU} and t_{CO} delay leads to a faster system frequency.

Figure 2. System Frequency Circuit

To determine a system's frequency, designers must add the t_{CO} of device 1, the t_{SU} of device 2, and the board delay.



In contrast, the device $\mathbf{f_{IOEXT}}$ values published in *Application Brief XBRF015 (Speed Metrics for High-Performance FPGAs)* only apply when an output pin feeds an input pin on either the same device or another identical device on the PCB.

Although it is more accurate for a designer to determine a device's $\mathbf{f_{IOEXT}}$ using individual $\mathbf{t_{SU}}$ and $\mathbf{t_{CO}}$ values, for this experiment, Altera calculated the $\mathbf{f_{IOEXT}}$ of EPF10K100A-1 and EPF10K100A-2 devices assuming they interface with devices of equivalent performance. This calculation allows Altera to compare the results directly to those published in *Application Brief XBRF015 (Speed Metrics for High-Performance FPGAs)*. Table 1 shows that the I/O frequency of EPF10K100A-1 devices was 95% faster than that of XC4085XL-09 devices. Further, the I/O frequency of EPF10K100A-2 devices was also faster than that of XC4085XL-09 devices.

Table 1. External I/O Frequency Results

Parameter	EPF10K100A-1	EPF10K100A-2	XC4085XL-09
External $\mathbf{f_{IOEXT}}$ (MHz)	111	94	57
$\mathbf{t_{SU}}$ (ns)	3.7	4.5	8.4
$\mathbf{t_{CO}}$ (ns)	5.3	6.1	9.0
$\mathbf{t_{H}}$ (ns)	0.0	0.0	0.0

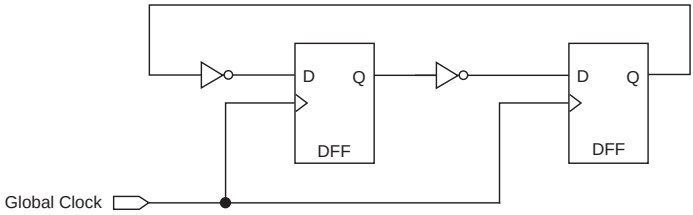
Average Routing Delay

The average routing delay benchmark measures the maximum achievable clock frequency when two registers are directly connected at various locations within a PLD. Using the average routing delay benchmark, designers can best compare the routing performance between two different device architectures.

In this experiment, register pairs were connected as circular shift registers fed by a global clock, as shown in Figure 3.

Figure 3. Average Routing Delay Circuit

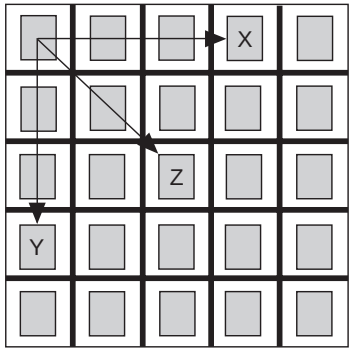
Pair of circular shift registers used to measure routing delays over horizontal, vertical, and diagonal distances within a device.



These registers were placed in specific FLEX logic elements (LEs) through assignments in a MAX+PLUS II Assignment & Configuration File (**.acf**). The maximum frequency was then measured between two registers located in the same row, the same column, and diagonally across the PLD (shown as the x-axis, y-axis, and z-axis, respectively, in Figure 4).

Figure 4. Average Routing Delay Measurement

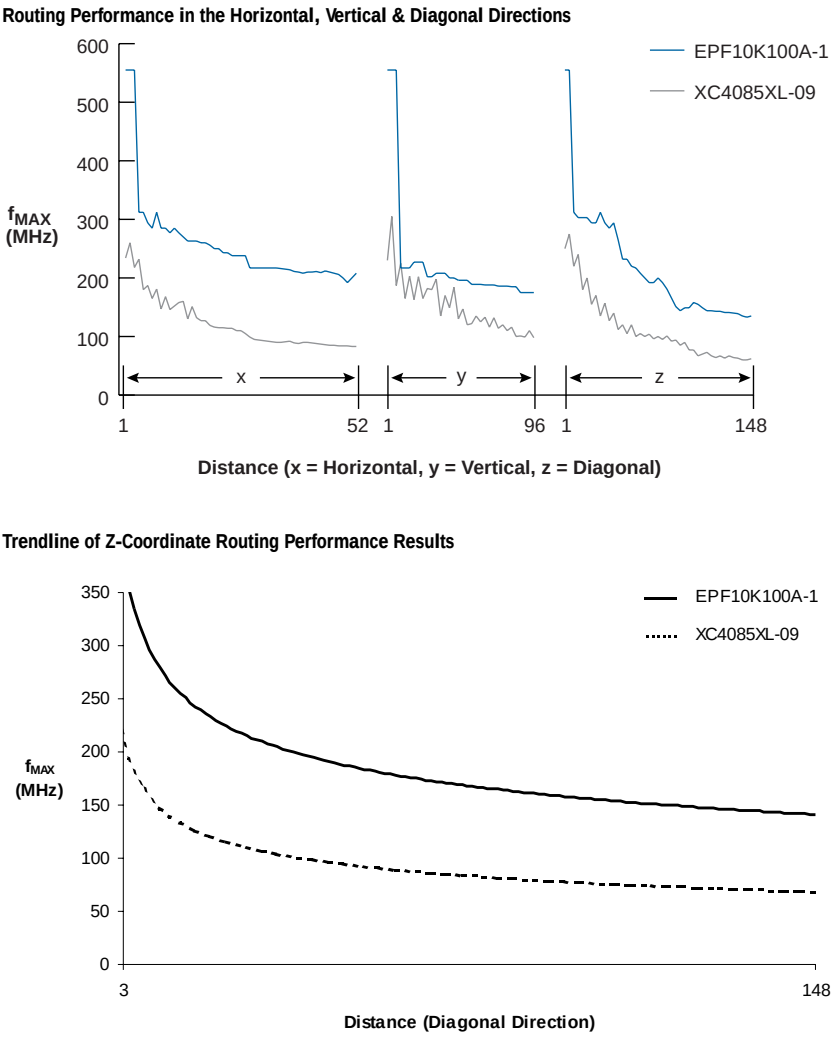
Measurements were made at incremental distances in the horizontal (x-axis), vertical (y-axis), and diagonal (z-axis) directions.



In the average routing delay experiment, measurements were made at incremental distances that spanned the entire distance across an EPF10K100A device. The range was from 1 to 52 logic cells apart in the horizontal direction, 1 to 96 logic cells apart in the vertical direction, and 1 to 148 logic cells apart (i.e., the sum of the x and y coordinate values) in the diagonal direction. The critical path (i.e., the slowest path in the design) is almost always diagonally across a PLD; thus, measurements taken in the diagonal direction will represent worst-case routing delays most closely.

Figure 5 shows the results of Altera's experiment as compared to the results published in *Application Brief XBRF015 (Speed Metrics for High-Performance FPGAs)*. Because the EPF10K100A device has 52 columns, 12 rows, and 8 logic cells per logic block, Figure 5 compares the EPF10K100A-1 and XC4085XL-09 device only up to 52 logic cells apart in the horizontal direction, 92 logic cells apart in the vertical direction, and 148 logic cells apart in the diagonal direction.

Figure 5. Routing Performance Results



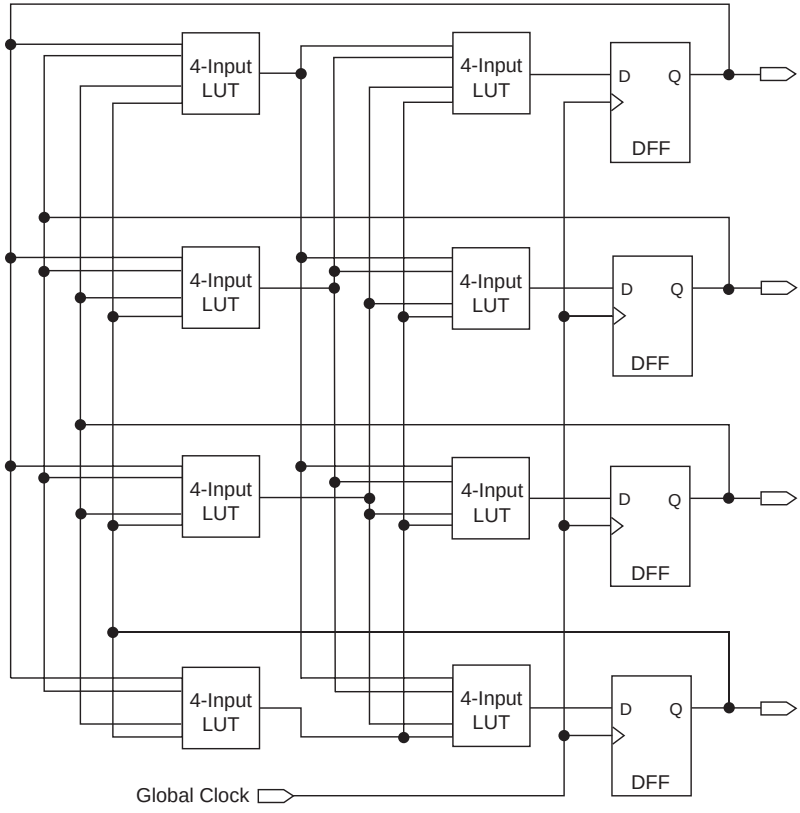
The results clearly show that the continuous interconnect routing structure of FLEX 10KA-1 devices is faster than the segmented interconnect structure of XC4000XL-09 devices. For example, in the worst case scenario when the registers are 148 logic cells apart in the diagonal direction, Altera's EPF10K100A-1 devices are 98% faster than XC4085XL-09 devices.

N-Level Combinatorial Logic

The *n*-level combinatorial logic benchmark measures the maximum combinatorial logic performance of look-up table (LUT) chains from 1 to 6 levels deep. In this experiment, 4-input LUTs were examined with the chains ranging from 1 to 6 levels, with each level consisting of four 4-input LUTs that are fully loaded and fully interconnected. Figure 6 shows a circuit used to test the 2-level chain of the 4-input LUT.

Figure 6. N-Level Combinatorial Logic Circuit

A 2-level chain of 4-input LUTs used to measure combinatorial performance results.



Designs implemented in EPF10K100A-1 devices were written in AHDL and used LCELL primitives to implement the LUTs. Figure 7 shows the results of the experiment.

Figure 7. Combinatorial Performance Results

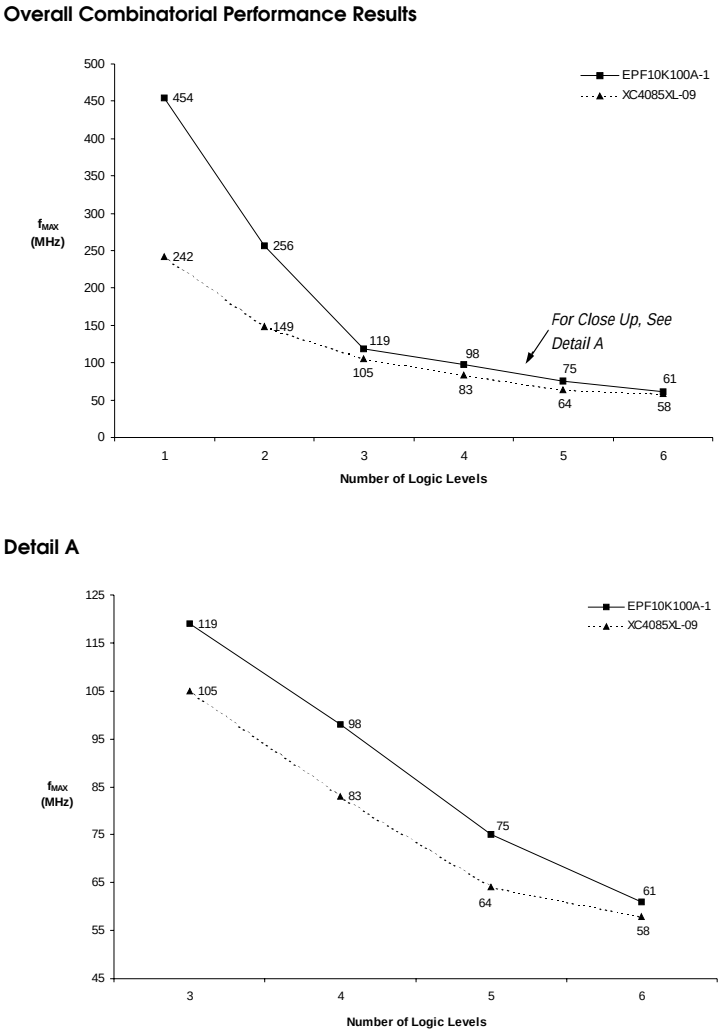


Figure 7 shows that EPF10K100A-1 devices consistently out-perform XC4085XL-09 devices at each logic level. For instance, an application with 4 levels of combinatorial logic runs 18% faster in EPF10K100A-1 devices than in XC4085XL-09 devices.

The combined results of the average routing delay and *n*-level combinatorial logic benchmarks are the best tool for comparing different PLD architectures. For real applications, the internal performance is determined by the signal path with the largest sum delay associated with the levels of logic and the routing between the levels of logic. Because the FLEX 10KA-1 performance was higher than the XC4000XL-09 performance for both benchmarks, it is reasonable to determine that the internal performance of FLEX 10KA-1 devices is faster than that of XC4000XL-09 devices.

N-to-1 Multiplexers

In this experiment, 2-to-1, 4-to-1, 16-to-1, 32-to-1, and 64-to-1 multiplexers were implemented in EPF10K100A-1 devices with their inputs and outputs registered. All of the multiplexers had 64 inputs (i.e., thirty-two 2-to-1 multiplexers, sixteen 4-to-1 multiplexers, four 16-to-1 multiplexers, two 32-to-1 multiplexers, and one 64-to-1 multiplexer). All of the multiplexers were implemented in the same design and shared the same inputs, thus introducing logic and loading delays. Designs implemented in EPF10K100A devices used the `lpm_mux` function from the library of parameterized modules (LPM). See Figure 8.

Figure 8. lpm_mux Function

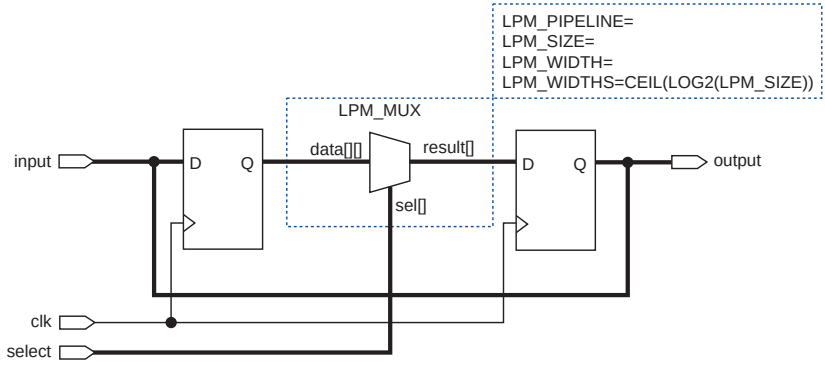
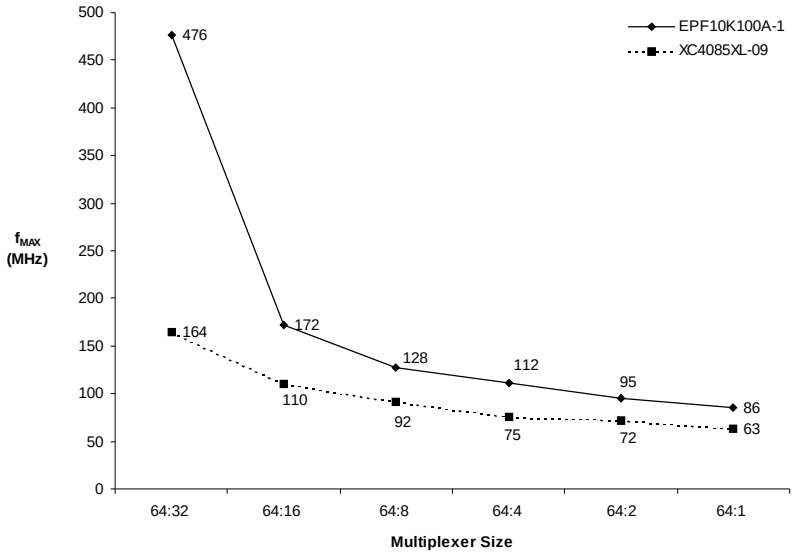


Figure 9 shows that EPF10K100A-1 devices were faster than XC4085XL-09 devices for all 6 multiplexer sizes.

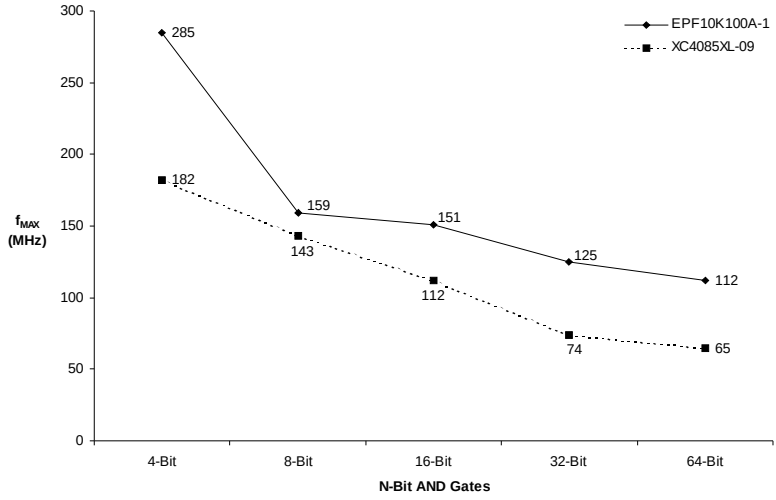
Figure 9. Multiplexer Performance Results



N-Bit AND Gates

The *n*-bit AND gate benchmark measures the performance of large combinatorial functions such as wide comparators. In the experiment, 4-bit, 8-bit, 16-bit, 32-bit, and 64-bit AND gates were implemented in EPF10K100A-1 devices. The inputs and outputs of the AND gates were registered, so performance could be measured in MHz. Using a common 64-bit input data path, the device was populated with sixteen 4-bit AND gates, eight 8-bit AND gates, and four 16-bit AND gates. Figure 10 shows that the EPF10K100A-1 device was 72% faster for 64-bit AND gates than for XC4085XL-09 devices.

Figure 10. AND-Gate Performance Results



Chained Adders

The chained adders benchmark measures the maximum frequency of a chained adder placed between two registers. This benchmark is important in determining the performance of common digital signal processing (DSP) functions that consist of large, complex mathematical operations, such as multipliers, counters, adders, and comparators. In this experiment, 8-, 16-, 24-, and 32-bit adders were placed in 1-, 2-, and 4-level chains. Figure 11 shows the circuit used to test a chain of two 8-bit adders.

Figure 11. Chained Adders Circuit

The circuit used to measure the performance of a chain of two 8-bit adders.

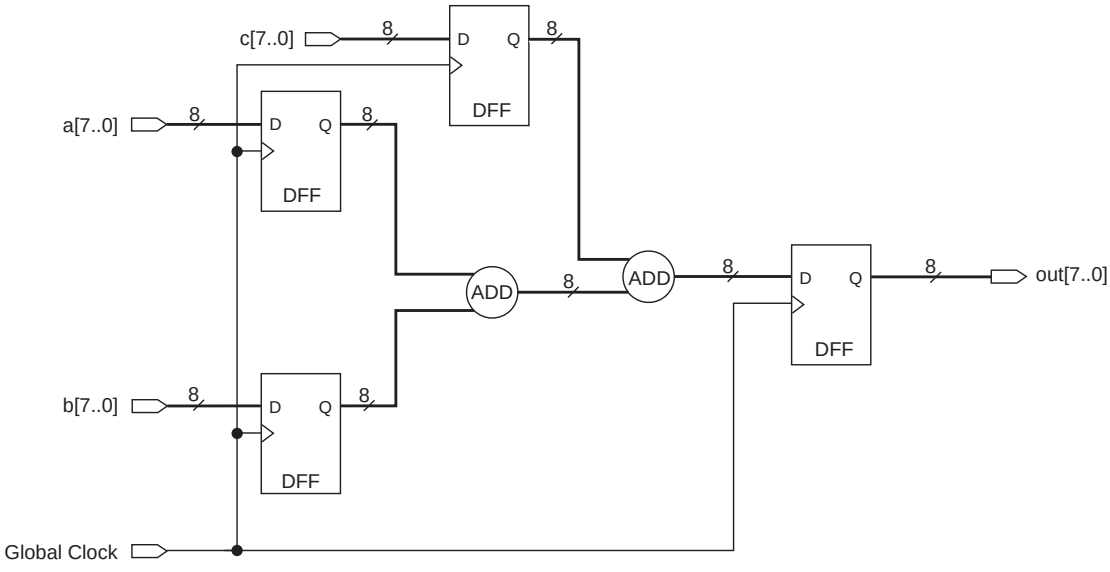
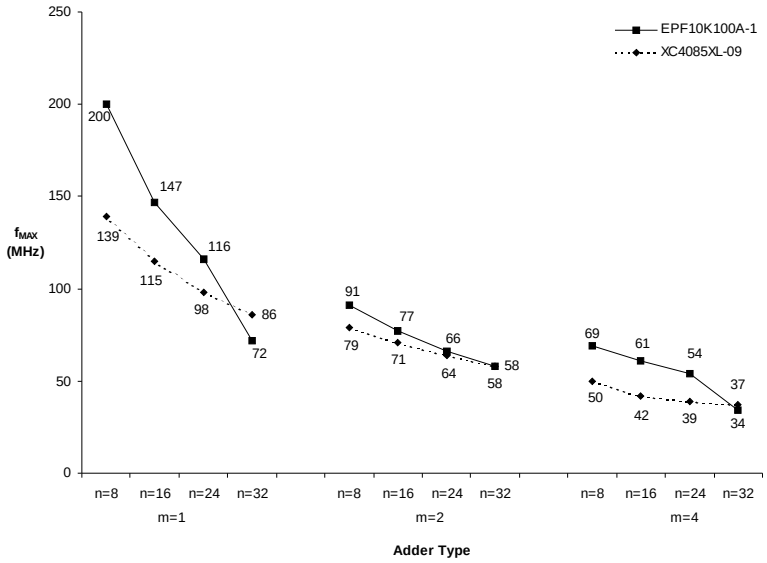


Figure 12 shows the results for this experiment, where *n* was the number of bits in the adder and *m* was the number of cascaded adders. EPF10K100A-1 devices were faster than Xilinx XC4085XL-09 devices in all instances, except for one chain of 32-bit adders and four chains of 32-bit adders. Part of Altera's performance advantage is due to Altera's fast carry chain architecture, which links LEs within logic array blocks (LABs) and between multiple LABs. These carry chains reduce the number of combinatorial logic levels required to implement adders.

Figure 12. Adder Performance Results



Conclusion

The performance of a design is dependent on the routing delay of an architecture and the number of logic levels in a design. Using the average routing delay and *n*-level combinatorial logic benchmarks, designers can directly compare the performance of two device architectures. The other benchmarks measure the performance of specific functions that are implemented in PLDs, but add little insight beyond their specific functionality. Even though some of the metrics do not relate to typical designs, these experiments prove that Altera EPF10K100A-1 devices consistently out-perform XC4000XL-09 devices for each benchmark.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com

Applications Hotline:
(800) 800-EPLD
Customer Marketing:
(408) 544-7104
Literature Services:
(888) 3-ALTERA