# Performance Measurements of Typical Applications

May 1998, ver. 1 Application Note 96

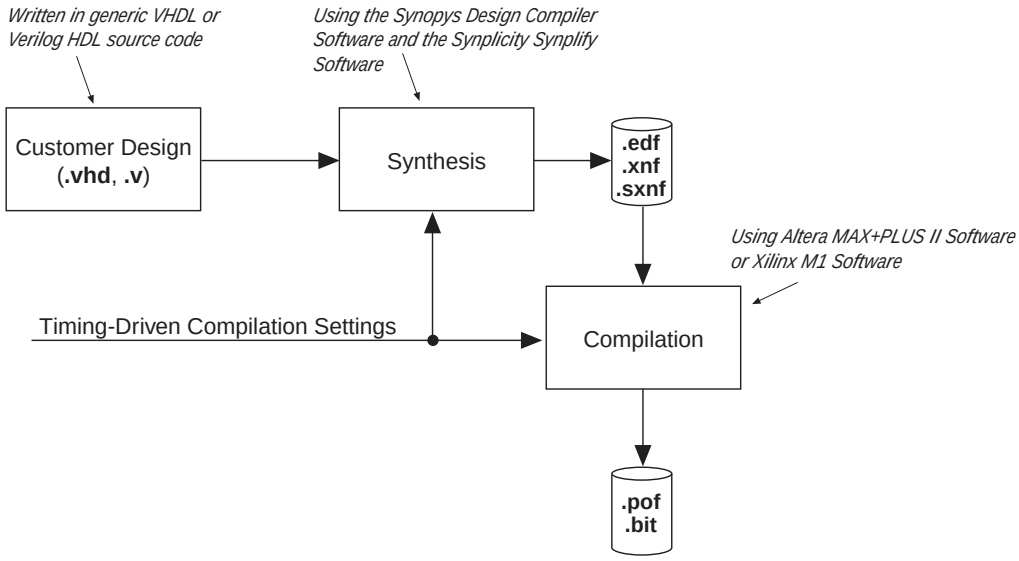
Benchmark tests are useful for comparing the performance of programmable logic devices (PLDs) from different vendors. However, they do not represent real-world designs. In some cases, benchmark tests can show results that are not attainable in real designs.

Altera® Applications recently conducted performance experiments using real customer designs that compared the fastest available devices from Altera and Xilinx, which are the -1 speed grade FLEX® 10KA devices (FLEX10KA-1) and -09 speed grade XC4000XL devices (XC4000XL-09). This application note describes the methodology of the experiment and discusses the results.

## Methodology

The experiment's methodology consisted of collecting customer designs, synthesizing the designs, compiling the designs, and obtaining performance results. Figure 1 describes the experiment flow.

***Figure 1. Experiment Flow***



In the experiment, timing-driven compilation was turned on in the synthesis and compilation tools, and for a given compilation, both tools were set to the same timing-driven compilation settings. When timing-driven compilation is enabled, the synthesis or compilation tool focuses on achieving a specified performance parameter for an overall design—sometimes at the expense of logic element (LE) utilization. The use of timing-driven compilation is discussed further in the synthesis and compilation sections.

## Determining Customer Designs

Altera collected about 50 actual customer designs for the experiment, but only 13 designs were selected because they compiled successfully for both FLEX 10KA-1 and XC4000XL-09 device families. To remove possible biases from the experiment, each designs required the following characteristics:

- Designs were written in generic VHDL or Verilog HDL source code, ensuring that they were unbiased toward either the Altera or Xilinx architecture.
- If a design used memory, the memory was mapped to logic, not to silicon-specific memory structures (e.g., embedded array blocks).
- None of the designs used floorplanning, instantiation, generic function blocks, MegaCore™ functions, or pipelining.

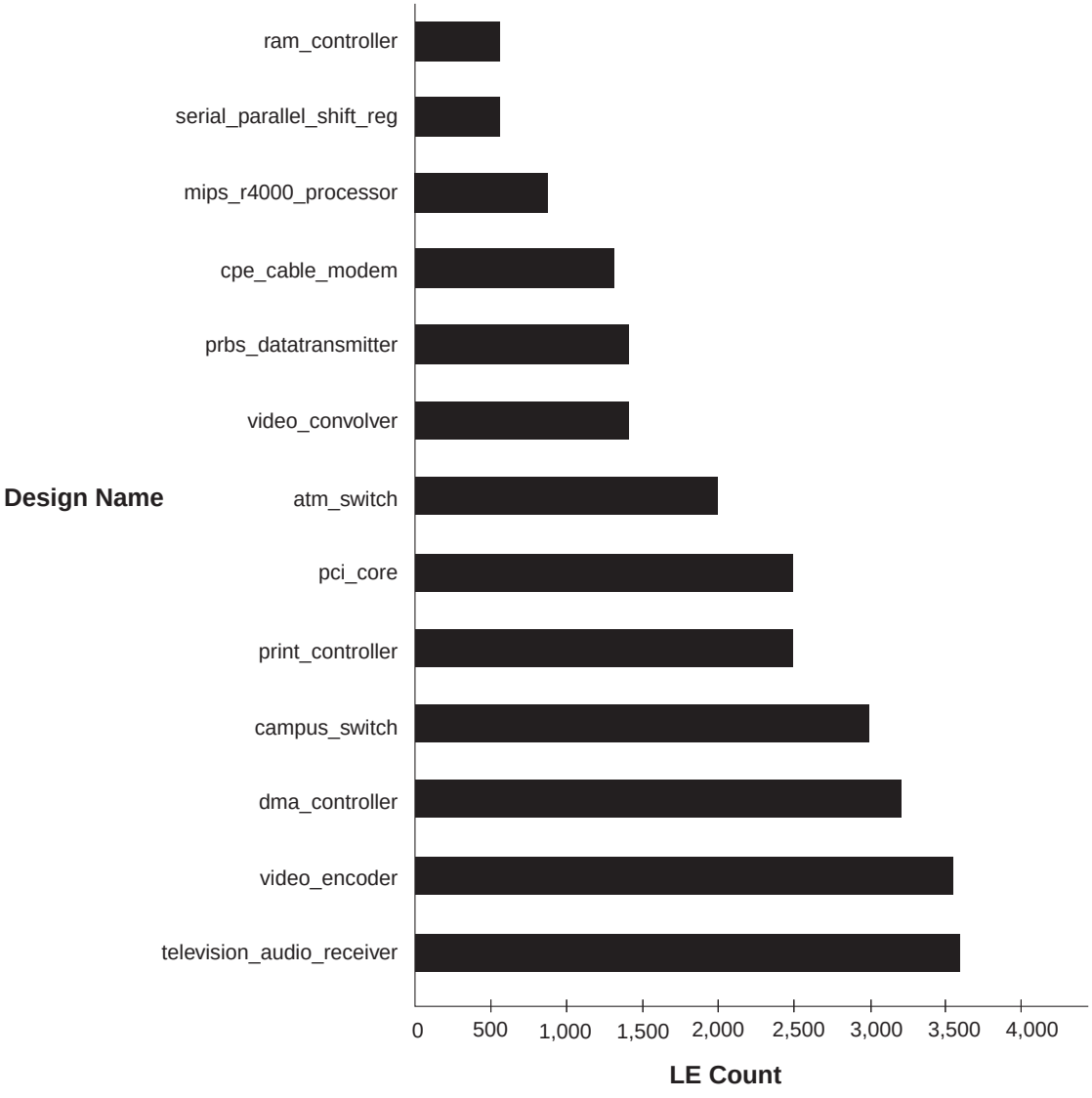
Table 1 describes the 13 designs used in the experiment.

*Table 1. Customer Designs Used in the Experiment*

| Name | LEs | I/O Pins | Design Entry Language |
|---|---|---|---|
| cpe_cable_modem | 1,300 | 97 | VHDL |
| dma_controller | 3,200 | 183 | Verilog HDL |
| prbs_datatransmitter | 1,400 | 150 | VHDL |
| ram_controller | 600 | 104 | VHDL |
| video_encoder | 3,600 | 223 | VHDL |
| video_convolver | 1,400 | 219 | VHDL |
| campus_switch | 3,000 | 125 | Verilog HDL |
| atm_switch | 2,000 | 60 | Verilog HDL |
| television_audio_receiver | 3,700 | 233 | VHDL |
| pci_core | 2,500 | 275 | VHDL |
| mips_r4000_processor | 850 | 132 | Verilog HDL |
| serial_parallel_shift_reg | 600 | 36 | VHDL |
| print_controller | 2,500 | 189 | Verilog HDL |

The 13 customer designs were generally large, averaging over 2,000 LEs. See Figure 2.

***Figure 2. Customer Design LE Counts*** *Note (1)*




*Note:*
(1) LE counts were determined during synthesis without timing-driven compilation.

## Synthesizing the Designs

The 13 customer designs were synthesized using the Synopsys Design Compiler software version 1997.01 and the Synplicity Synplify software version 3.0B. Each design was synthesized 5 times with each synthesis tool at various timing-driven compilation settings. Table 2 shows the timing-driven compilation settings used in the experiment.

*Table 2. Timing-Driven Compilation Settings Used*

| Synthesis Run # | Timing-Driven Compilation Setting (MHz) |
|---|---|
| 1 | 10 |
| 2 | 20 |
| 3 | 30 |
| 4 | 40 |
| 5 | 50 |

Each synthesis tool generated a Xilinx netlist file (**.xnf** or **.sxnf**) for XC4000XL-09 devices and an EDIF netlist file (**.edf**) for FLEX 10KA-1 devices. After synthesis was completed, the highest timing-driven compilation performance rate was used for each design, regardless of which synthesis tool produced it.

## Compiling the Designs

The resulting netlist files (**.edf**, **.xnf**, or **.sxnf**) were compiled in Altera's MAX+PLUS II version 8.2 software or Xilinx's M1 version 1.3.7 software, respectively. The same timing-driven compilation settings used in synthesizing the designs were used when compiling the designs. For example, if a design was synthesized in the Synopsys Design Compiler with timing-driven compilation set at 20 MHz, the design was compiled in the MAX+PLUS II or M1 software with the same setting.

## Obtaining Performance Results

After compilation, performance rates for the designs were measured using timing analysis in the MAX+PLUS II or M1 software. Each design's best performance rate was used in the comparison.

## Results

Figure 3 shows the results of the performance experiment.

***Figure 3. Performance Comparison of FLEX 10KA-1 vs. XC4000XL-09 Devices***

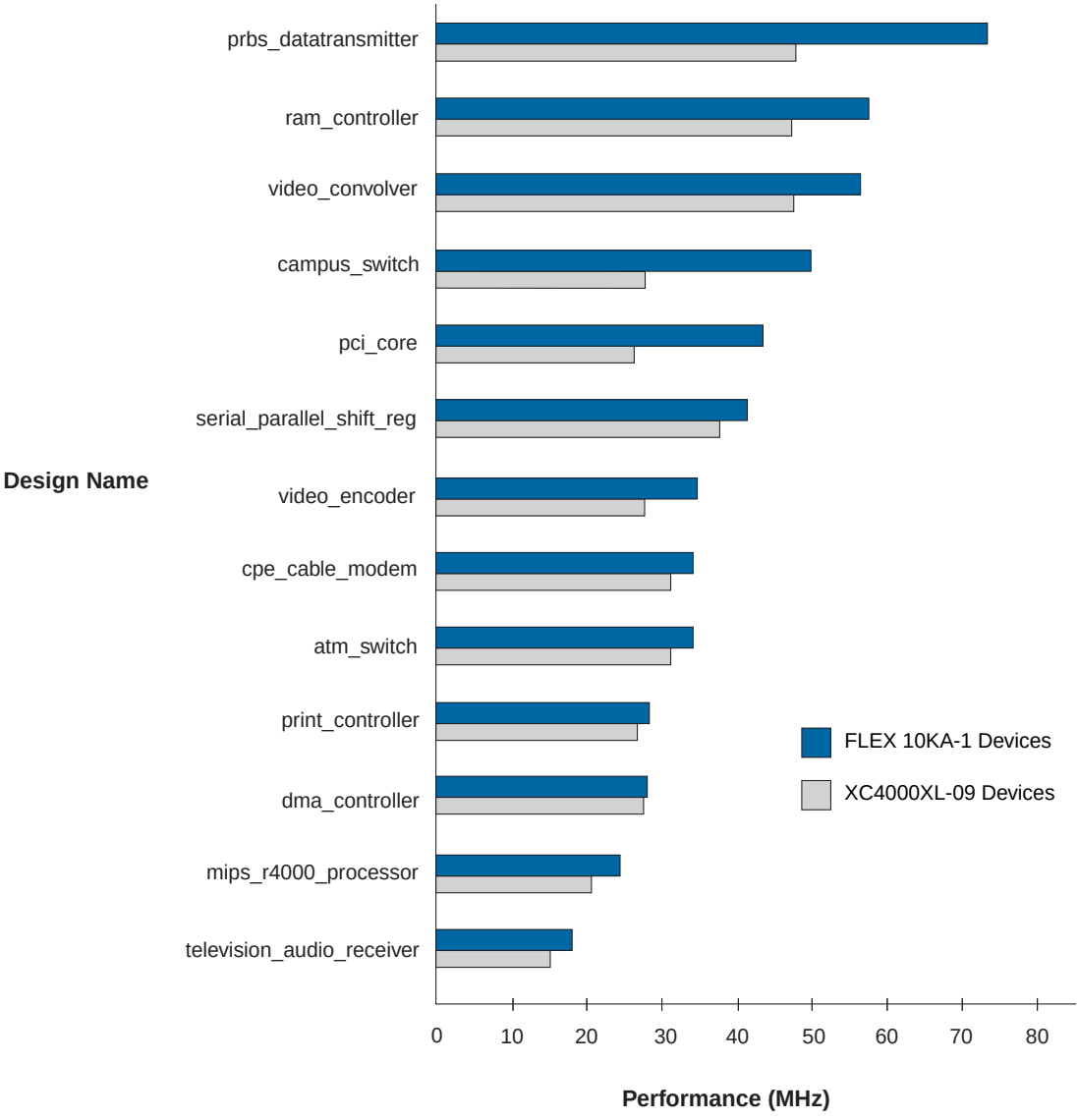
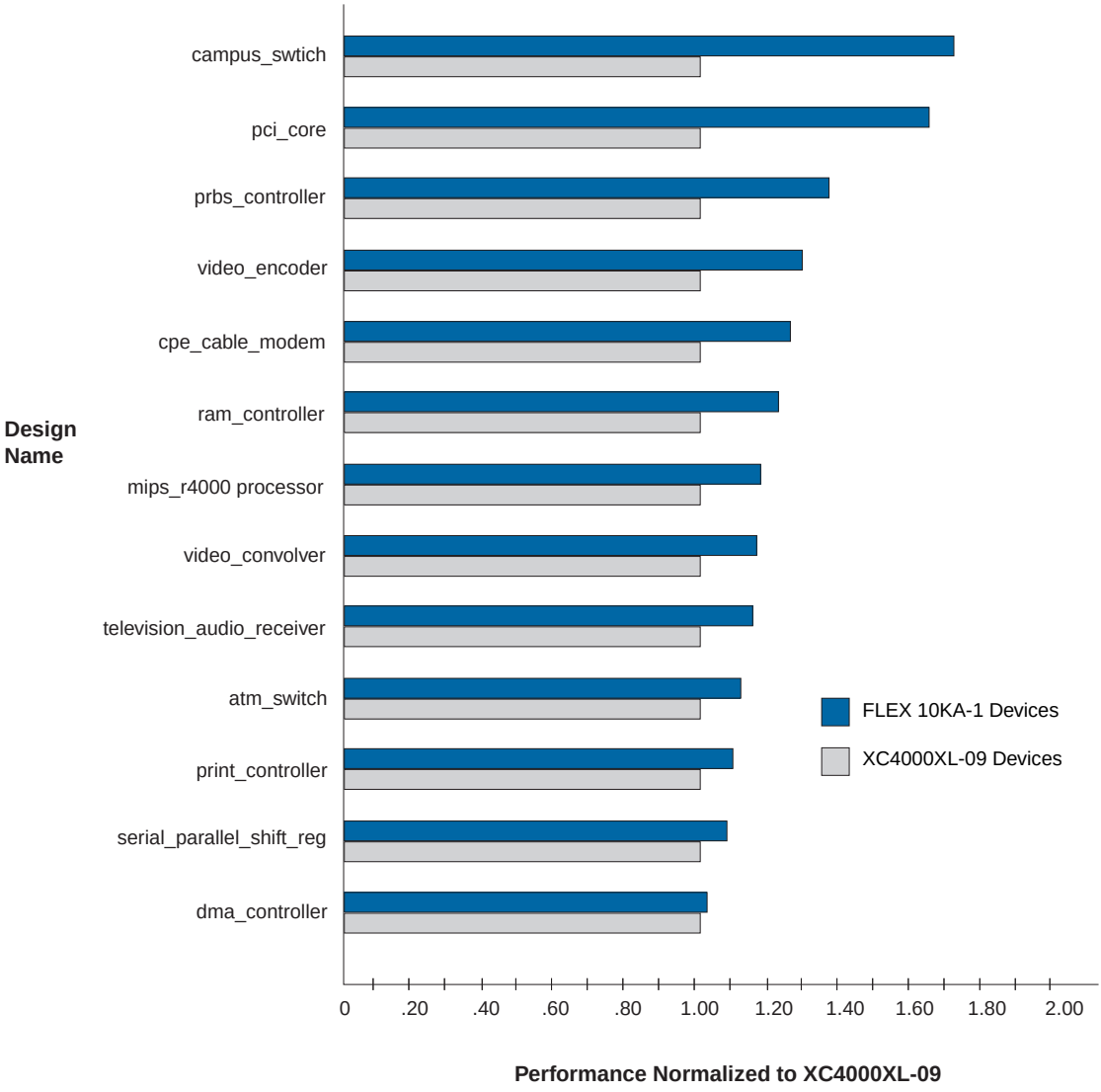


Figure 4 shows the same results normalized for the Xilinx performance results.

***Figure 4. Normalized Performance***



## Conclusion

This experiment showed that FLEX 10KA-1 devices outperform XC4000XL-09 devices in every design. For these designs, FLEX 10KA-1 devices outperformed XC4000XL-09 devices by a median of 25% and an average of 17%.

Because the designs used in the experiment were not optimized for the target device architecture, they provide a worst-case analysis. Even so, the results show that on average the FLEX 10K-1 devices operate at 40 MHz without special design techniques. In most cases, designers will use techniques to increase performance, such as:

- *Intellectual property*—Using megafunctions that are pre-optimized for a particular architecture.
- *Instantiation*—Placing functions in close proximity to reduce the overall timing.
- *Pipelining*—Breaking up logic with registers to increase system performance (this technique may require a bandwidth tradeoff).
- *Floorplanning*—Manual intervention on speed-critical paths to increase performance.

Using advanced design techniques can boost performance of the designs used in this experiment up to an average of 70 MHz (an increase of 75%). For more information on experiment results in which designs were optimized for performance, refer to *Application Note 98 (Comparing Performance Using Common Megafunctions)*.

## Additional Information

The following documents provide more detailed information about FLEX 10KA-1 devices:

- *FLEX 10K Embedded Programmable Logic Family Data Sheet*
- *Application Note 97 (Comparing Performance of High-Density PLDs)*

The documents are available by contacting Altera Literature Services at (888) 3-ALTERA; you can also download them from the Altera web site at **http://www.altera.com**.

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
lit_req@altera.com