Comparing Performance of Common Megafunctions

August 1998, ver. 2 Application Note 98

Introduction

The maximum performance capability of programmable logic devices (PLDs) has increased dramatically over the last few years. In the past, high-density PLDs were limited to 33-MHz operation for most applications. Today, devices with 20,000 to 100,000 gates typically operate up to 66 MHz and beyond.

The performance of high-density PLDs varies greatly depending on the device architecture, the design tools used, and the design complexity. As a result, the only way to fairly test performance between competitive architectures is to use benchmark designs that eliminate the variability associated with design complexity and tool efficiency.

This application note summarizes the results of performance tests comparing Altera® FLEX® 10KA-1 devices against Xilinx XC4000XL-09 devices when implementing pre-optimized functions. Because each function was optimized for the target architecture, biases associated with design tool efficiency, routing, and synthesis were eliminated from the experiment.

Proprietary Megafunctions

This experiment tested the performance of Altera FLEX 10KA-1 and Xilinx XC4000XL-09 devices when implementing pre-optimized functions created by each company. Altera chose two functions that are commonly used in typical applications, a multiplier and finite impulse response (FIR) filter, to use in the experiment. Table 1 shows the devices tested in the experiment.

Table 1. Devices Used in the Experiment

Company	Device	Total Logic Elements
Altera	EPF10K100A-1	4,992
	EPF10K50V-1	2,880
	EPF10K30A-1	1,728
Xilinx	XC4062XL-09	4,608
	XC4036XL-09	2,592
	XC4013XL-09	1,152

Functions

The multiplier and FIR filter function parameters were set to the same values for the FLEX 10KA-1 and XC4000XL-09 devices. Table 2 shows the parameter settings used in the multiplier experiment.

Table 2. Multiplier Parameter Settings

Parameter	Value
Inputs	16 bits
Outputs	32 bits
Inputs and outputs	Registered
Pipelined latency	3

Table 3 shows the parameter settings used in the FIR filter experiment.

Table 3. FIR Filter Parameter Settings

Parameter	Value
Number of taps	16
Input width	16 bits
Coefficient width	12 bits
Output precision	16 bits
Inputs and outputs	Registered
Pipelining	On

Altera Functions

The multiplier function was implemented in FLEX 10KA-1 devices using the `lpm_mult` function from the library of parameterized modules (LPM). The FIR filter function was implemented in FLEX 10KA-1 devices using Altera's `fir_16tp` reference design.

For more information on these functions, go to the *LPM Quick Reference Guide* and *Functional Specification 1 (FIR Filters)*.

Xilinx Functions

The multiplier and FIR filter functions were implemented in XC4000XL-09 devices using functions available from the Xilinx CoreGEN library of optimized functions. Both functions were compiled using Verilog HDL source code that was generated with the Xilinx CoreGEN version 1.4 software.

Lab Setup

This section describes the methods used to implement and compile the multiplier and FIR filter functions in Altera FLEX 10KA-1 and Xilinx XC4000XL-09 devices. The software setup used for each device family was chosen to achieve maximum design performance.

Altera FLEX 10KA Devices

The multiplier and FIR filter functions were implemented and synthesized in FLEX 10KA-1 devices with the MAX+PLUS® II version 8.3 development tool. Timing-driven compilation and the *Fast Synthesis* style were turned on for all design compilations. In addition, the speed / area optimization setting in the MAX+PLUS II software was set to maximize speed for the FIR filter function. Table 4 shows the difference in performance when timing-driven compilation was turned on in the EPF10K30A-1 device.

Table 4. EPF10K30A-1 Performance

Function	Timing-Driven Compilation	
	On (MHz)	Off (MHz)
Multiplier	81.96	74.07
FIR Filter	121.95	120.48

Xilinx XC4000XL Devices

The multiplier and FIR filter functions were implemented in XC4000XL-09 devices with the XACTstep M1 version 1.4 development tool and synthesized with Synplicity's Synplify 3.0B1 software. All designs were compiled with a specified clock period of 10 ns. For optimal performance, timing-driven compilation was turned on for all designs; without timing-driven compilation, the XC4000XL-09 devices had extremely low performance. Table 5 shows the difference in performance when timing-driven compilation was turned on in the XC4036XL-09 device.

Table 5. XC4036XL-09 Performance

Function	Timing-Driven Compilation	
	On (MHz)	Off (MHz)
Multiplier	71	25
FIR Filter	84	18.5

Initially, Xilinx's FIR filter function failed to fit in any XC4000XL-09 device even though adequate resources were available in all but the smallest device. For the designs that failed to fit, the Xilinx XACTstep M1 software produced the following errors:

```
Error: baste: 125 - The RLOC value of "R62C2" on CLB
u1/pdafir/symmetric_adder_result_6_0 in RPM "u1/hset"
creates a macro that is too large for the device. Use a
bigger device.

Error: baste: 125 - The RLOC value of "R78C18" on CLB
u1/pdafir/adder_tree_bus_1_6_1 in RPM "u1/pdafir_plan"
creates a macro that is too large for the device. Use a
bigger device.

FATAL_ERROR: baste: bastemacro.c: 1255:1.48 - Site already
used. Comp u1/ppp/pdafir/col_2b3_adders_0b2_b1_add_uO/
U0/newSim6053/cy4 and comp u1/ppp/pdafir/col_0b3_adders_
0b2_b1_add_uO/U0/newSim4907/cy4 are competing for same
site. Process will terminate. Please call Xilinx support.
```

Xilinx Applications was contacted for a workaround. According to Xilinx, the XACTstep M1 compiler tries to implement logic in a single horizontal row of configurable logic blocks (CLBs) to increase performance. The error was generated because the function was too big to fit in a single row of CLBs. The initial workaround provided by Xilinx failed to resolve the issue. The second solution was to use an "-ir switch" compilation setting, which permits logic to be implemented in multiple rows but degrades performance. The "-ir switch" compilation setting resolved the issue and was therefore used during the experiment.

Results

After compilation, performance rates and logic utilization for each design were measured using timing analysis in the MAX+PLUS II or XACTstep M1 software. See "Sample Files" on page 9 for sample Report Files and Timing Analyzer Output Files generated during the experiment. Table 6 shows the logic elements (LEs) consumed by each device in the experiment.

Table 6. Device Utilization Results

Device	Total LEs	LEs Used	
		Multiplier Function	FIR Filter Function, *Note (1)*
EPF10K100A-1	4,992	584	967
EPF10K50V-1	2,880	547	967
EPF10K30A-1	1,728	544	967
XC4062XL-09	4,608	416	1,652
XC4036XL-09	2,592	416	1,652
XC4013XL-09	1,152	416	*Note (2)*

Notes:
(1) For XC4000XL-09 devices, the FIR filter function results were obtained with the "-ir switch" compilation setting.
(2) The function was too large to fit in this device.

Figure 1 shows the performance results for the multiplier function.

Figure 1. Multiplier Performance Results

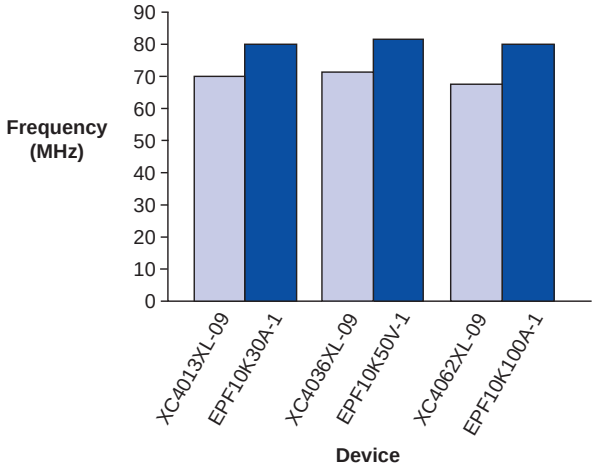
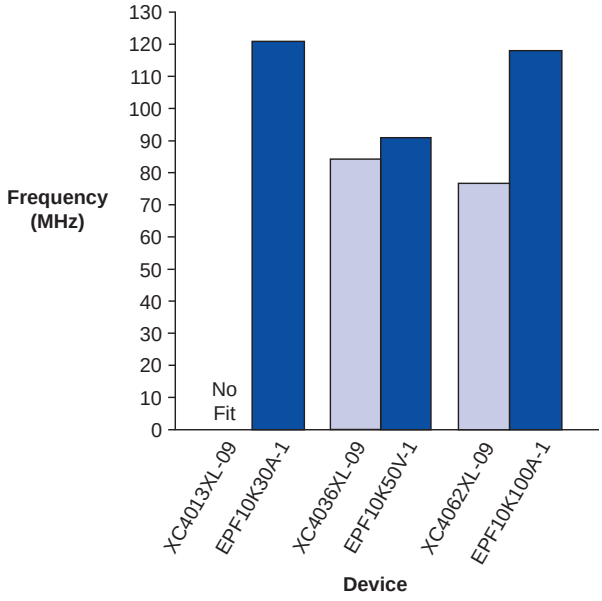


Figure 2 shows the performance results for the FIR filter function.

Figure 2. FIR Filter Performance Results



On average, the multiplier function performed 17% faster in FLEX 10KA-1 devices than in XC4000XL-09 devices, and the FIR filter function performed 31% faster in FLEX 10KA-1 devices than in XC4000XL-09 devices.

Third-Party Megafunction

This experiment tested the performance of FLEX 10KA-1 and XC4000XL-09 devices when implementing megafunctions from third-party vendors. To eliminate possible biases from the experiment, the vendor had to be both an Altera Megafunction Partners Program (AMPP^{SM}) partner and a Xilinx AllianceCORE partner, and offer functions optimized for both FLEX 10KA-1 and XC4000XL-09 devices. Using this criteria, Altera selected a commonly used function, the Viterbi decoder, from CAST, Inc.

Table 7 shows the density of each device tested in the experiment. The experiment used the smallest device from each family that could implement the Viterbi decoder megafunction.

Table 7. Devices Used in the Experiment

Device	Device LEs
EPF10K30A-1	1,728
XC4013XL-09	1,152

Megafunction

The Viterbi decoder megafunction is a complex function that is used to decode convolutional codes for a transmitted data stream. It can also produce the best estimate of a transmitted sequence over a channel with inter-symbol interference (ISI). The function has both complex combinatorial and sequential functionality. Figure 3 shows the block diagram for the Viterbi decoder megafunction.

Figure 3. Viterbi Decoder Megafunction Block Diagram

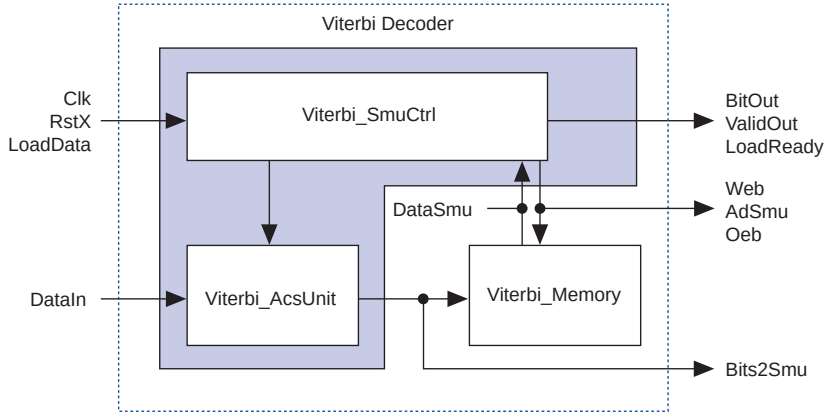


Table 8 shows the Viterbi decoder parameter values used for both devices.

Table 8. Parameters for Viterbi Decoder Megafunction

Parameter	Value
Number of states in trellis	16
Number of bits to represent transition values	8
Number of ACS cells	4
Length of trace-back	30
Length of the received burst	5
Initial path metric for state 0	–4
Survivor memory (RAM) word length	8

For more information on the Viterbi decoder megafunction, contact CAST, Inc. or go to the CAST web site at **http://www.cast-inc.com**.

Results

The XC4013XL-09 performance results were obtained from Xilinx's *Viterbi Decoder Product Specification*, and the EPF10K30A-1 performance results were obtained from Altera's *Solution Brief 33 (Viterbi Decoder Megafunction)*. See Table 9.

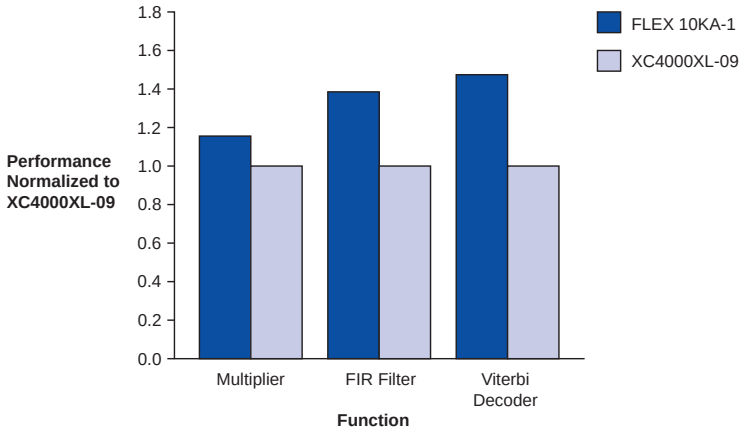
Table 9. Performance Results

Device	LEs Used	f_{MAX} (MHz)
EPF10K30A-1	960	31
XC4013XL-09	850	20

Conclusion

These experiments showed that FLEX 10KA-1 devices outperform XC4000XL-09 devices by an average of 55% when implementing pre-optimized functions. Figure 4 shows the results, normalized for the XC4000XL-09 performance results.

Figure 4. Normalized Common Megafunction Results



Pre-optimized megafunctions are used more often today to reduce design time and increase system performance. As a side benefit, megafunctions are an excellent source for benchmarking PLD performance between different product families, because factors associated with design tools and methodology are eliminated. Altera will continue using pre-optimized megafunctions to provide unbiased comparisons of competitive PLDs.

Sample Files

Figure 5 shows an excerpt from a sample MAX+PLUS II Report File (**.rpt**) for the FIR filter function.

Figure 5. MAX+PLUS II Report File for the FIR Filter Megafunction

```
***** Project compilation was successful

** DEVICE SUMMARY **

Chip/                Input     Output    Bidir     Memory    Memory            LCs
POF     Device       Pins      Pins      Pins      Bits%     Utilized   LCs    %Utilized

acf_122mhz
      EPF10K30AQC240-1  18 16       0         0         0%        967       55 %

User Pins:         18       16       0


Total dedicated input pins used:                5/6      ( 83%)
Total I/O pins used:                           29/183    ( 15%)
Total logic cells used:                       967/1728   ( 55%)
Total embedded cells used:                      0/48     (  0%)
Total EABs used:                                0/6      (  0%)
Average fan-in:                              1.84/4      ( 46%)
Total fan-in:                             1788/6912     ( 25%)

Total input pins required:                     18
Total input I/O cell registers required:        0
Total output pins required:                    16
Total output I/O cell registers required:       0
Total buried I/O cell registers required:       0
Total bidirectional pins required:              0
Total reserved pins required                    0
Total logic cells required:                   967
Total flipflops required:                     959
Total packed registers required:                0
Total logic cells in carry chains:            420
Total number of carry chains:                  35
Total number of carry chains of length  1-8 :  10
Total number of carry chains of length  9-16:  17
Total number of carry chains of length 17-24:   8
Total logic cells in cascade chains:            0
Total number of cascade chains:                 0
Total single-pin Clock Enables required:        0
Total single-pin Output Enables required:       0
Synthesized logic cells:                        0/1728   (  0%)
```

Figure 6 shows an excerpt from a sample Xilinx M1 Report File (**.par**) for the FIR filter function.

Figure 6. Sample Xilinx M1 Report File

```
*** Loading device database for application par from file "filter20.ncd".
"filter20" is an NCD, version 2.27, device xc4062xl, package bg432, speed -09
Loading device for application par from file '4062xl.nph' in environment
d:/xilinx.
Device speed data version:      x1_0.17 1.16 PRELIMINARY.
Overall effort level (-ol):     2 (default)
Placer effort level (-pl):      2 (default)
Placer cost table entry (-t):   1
Router effort level (-rl):      2 (default)
Device utilization summary:

Number of External IOBs           32 out of 352         9%
       Flops:                     32
       Latches:                    0
Number of Global Buffer IOBs       1 out of 8          12%
       Flops:                      0
       Latches:                    0
Number of CLBs                   826 out of 2304       35%
       Total Latches:              0 out of 4608        0%
       Total CLB Flops:         1468 out of 4608       31%
       4 input LUTs:            1135 out of 4608       24%
       3 input LUTs:             145 out of 2304        6%
Number of BUFGLSs                  1 out of 8          12%

The Number of signals not completely routed for this design is: 0
The Average Connection Delay for this design is:        3.825 ns
The Average Connection Delay on critical nets is:       0.000 ns
The Average Clock Skew for this design is:              0.587 ns
The Maximum Pin Delay is:                               8.280 ns
The Average Connection Delay on the 10 Worst Nets is:   7.456 ns
Listing Pin Delays by value: (ns)
d <= 10     < d <= 20   < d <= 30   < d <= 40   < d <= 50   d > 50
---------   ---------   ---------   ---------   ---------   ---------
4820        0           0           0           0           0
```

Figure 7 shows an excerpt from a sample timing analyzer file for the Xilinx device.

Figure 7. Sample Xilinx Timing Analyzer File

```
Design file:              D:\Test_DSP\new_16\4062\out.ncd
Physical constraint file: D:\Test_DSP\new_16\4062\filter20.pcf
Device,speed:             xc4062xl,-09 (x1_0.17 1.16 PRELIMINARY)
Report level:             error report, limited to 1 item per constraint

Timing constraint: NET "clk" PERIOD =  10 nS   HIGH 50.000 % ;
13529 items analyzed, 194 timing errors detected.
Minimum period is  13.011ns.

Slack:    -3.011ns path u1/pdafir/adder_tree_bus_3_1_4 to
u1/pdafir/adder_tree_bus_4_0_31 relative to 10.000ns delay constraint

Path u1/pdafir/adder_tree_bus_3_1_4 to u1/pdafir/adder_tree_bus_4_0_31
contains 12 levels of logic:
Path starting from Comp: CLB_R30C22.K (from clk_c)
To               Delay type      Delay(ns)     PhysicalResource
                                               Logical Resource(s)
-------------------------------------------------------------------------
CLB_R30C22.YQ    Tcko            1.470R        /pdafir/adder_tree_bus_3_1_4
CLB_R21C14.G1    net(fanout=1)   4.357R        /pdafir/adder_tree_bus_3_1_4
CLB_R21C14.COUT  Topcy           1.600R        _b1_add_uO/U0/newSim6325/cy4

Total (6.300ns logic, 6.711ns route)      13.011ns (to clk_c)
      (48.4% logic, 51.6%% route)

1 constraint not met.
Timing summary:
---------------

Timing errors: 194  Score: 146655

Constraints cover 15029 paths, 0 nets, and 4779 connections (100.0% coverage)
Design statistics:
     Minimum period:  13.011ns (Maximum frequency:  76.858MHz)
```

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