WP218 (v1.2) February 14, 2005

Achieving Breakthrough Performance in Virtex-4 FPGAs

By: Frédéric Rivoallon

Virtex-4™ FPGAs are the first multi-platform FPGA family based on the revolutionary Advanced Silicon Modular Block or ASMBL™ architecture. Between its different platforms (LX, FX, and SX), the Virtex-4 family offers a programmable logic solution closely matching designers needs. With several new architectural elements designed for maximum throughput, higher integration, and lower power consumption, Virtex-4 FPGAs can attain new performance levels.

Maximum system performance requires a balanced mix of performance-efficient components in your system (logic, on-chip memory, I/O, etc.).

This paper first shows the level of performance that can be reached with Virtex-4 design building blocks. Then, it illustrates how to take advantage of this performance in actual designs or benchmarks by looking at coding style, tool settings, and constraints.

Performance benchmarks based on customer designs show Virtex-4 designs have up to 43% better logic performance than other 90 nm FPGAs, and are on average one speed grade faster.

© 2004-2005 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and disclaimers are as listed at http://www.xilinx.com/legal.htm. All other trademarks and registered trademarks are the property of their respective owners. All specifications are subject to change without notice.

500-MHz Clock Rates From Fabric to the Blocks

Virtex-4 devices are the platform of choice for high-performance design; its logic fabric and fixed blocks can all operate at 500-MHz clock rates. For example, in the logic fabric, many functions, such as counters, adders, and storage (RAM/ROM) implemented with lookup tables (LUTs), are capable of performing at that clock rate. The fixed blocks (memory and DSP) have been designed to also operate at the same speed.

Logic Fabric

The basic Virtex-4 logic element is composed of a 4-input LUT and a flip-flop, as well as additional elements, such as a function expander (MUXF) and an arithmetic cell (MULT_AND). The function expander enables larger LUT structures (such as 5-input LUT or 6-input LUT) to be built.

In RAM mode, the Virtex-4 LUT can be used as a 16-bit memory element, a 16-bit shift register, or even a loadable LUT whose content can be changed during operation. This RAM mode, unique to Virtex-4 FPGAs, offers very-efficient small memories.

DSP Performance

Virtex-4 FPGAs offer more DSP specific blocks than any other device, resulting in an 256 GMAC/s DSP bandwidth. The 500-MHz XtremeDSP™ slice provides the highest performance, lowest power, and most versatile arithmetic unit. It can implement a multiply accumulate unit and can be cascaded with other similar blocks to implement larger design structures such as a Finite Impulse Response (FIR) filter, while still operating at nominal speed. This performance and expandability enable Virtex-4 FPGAs to implement DSP critical functions with an unprecedented level of performance and integration, without requiring logic from the fabric. In addition to DSP functions, the DSP slices can be configured as counters, barrel shifters, adders, subtracters, accumulators, and other functions operating at 500 MHz.

On-Chip RAM

Each Virtex-4 block RAM stores 18K bits of data. Compared to the previous generation, the enhancements of the new block include:

- New built-in level of pipeline with an additional register at the data read output enabling 500-MHz clocking.
- Two adjacent block RAMs combined to one deeper 32K x 1 memory without any external logic or speed loss.
- Individual byte write enable.
- A unique built-in multi-rate FIFO mode providing address sequencing and control circuitry without additional logic. The FIFO mode provides Full and Empty outputs, as well as programmable Almost Full and Almost Empty flags, and guarantees glitch-free generation of these flags.
- Built-in Error-Correction Control (ECC) where the block RAMs provide optional Hamming code error correction and checking capability to improve effective system performance and save space.

I/O Bandwidth

Maximum system performance requires high-performance interactive links between the FPGA and the other components of the system. Virtex-4 FPGAs is the only programmable device providing high-speed serial transceivers capable of 11 Gb/s.

Embedded Processing

Virtex-4 FPGAs offer a built-in, enhanced PowerPC™ 405 core that delivers 680 DMIPS performance at 450 MHz and a new Auxiliary Processor Unit (APU) controller. This APU simplifies and streamlines the connection to custom co-processors and hardware accelerators.

Memory Interface

The ChipSync™ technology enables flexible and high performance memory interface applications. It provides a unique run time data to clock centering capability for increased design margins, with a resolution of 80 ps. The adjustment compensates for devices and parameter variations: process, voltage, and temperature.

Virtex-4 FPGAs can achieve 600 Mb/s per singled ended I/O with buses as wide as 432 bits.

Built-in Technologies for Improving Effective Performance

In addition to the highest performance logic, embedded blocks, and design tools, the effective real-world performance you can obtain depends on clock/data signal integrity, power integrity, power budgets, etc. Virtex-4 FPGAs provide built-in technologies to provide you with the highest effective performance:

- A low skew, low-jitter, 500 MHz differential clock structure with greatly expanded flexibility.
- Advanced packaging technology and flip-chip assembly techniques, enabled by the proprietary ASMBL technology and abundant power and ground pins, improves signal integrity by minimizing package and PCB inductance.
- The XCITE™ technology provides on-chip signal termination with digitally controlled impedance.
- Lower power consumption per MHz means you can achieve more performance within your power budget. Virtex-4 FPGA reduces dynamic power consumption with 90 nm technology while reducing static power consumption with triple-oxide technology. The net result is a 50% reduction in power consumption compared to previous generation FPGAs, and a significant advantage over any other 90 nm competitor.

Performance Advantage

Table 1 shows the performance of some different design elements running at the speed grade listed (using ISE 6.3 service pack 3). It also compares values to the closest competitive offering referenced by FPGA-90 nm.

Table 1: **Virtex-4 Performance by Speed Grade**

	Virtex-4 -10	FPGA-90 nm Slow	Virtex-4 -11	FPGA-90 nm Middle	Virtex-4 -12	FPGA-90 nm Fast
Multiplier (9 x 9 or 18 x 18)	400 MHz	268 MHz	450 MHz	322 MHz	500 MHz	370 MHz
16-tap FIR filter with 8-bit data	400 MHz	165 MHz	450 MHz	203 MHz	500 MHz	237 MHz
16K x 32-bit RAM	527 MHz	306 MHz	610 MHz	326 MHz	643 MHz	370 MHz
16K x 64-bit RAM	311 MHz	195 MHz	311 MHz	226 MHz	335 MHz	272 MHz
4K x 144-bit RAM	400 MHz	289 MHz	450 MHz	309 MHz	500 MHz	350 MHz
64-bit Adder	205 MHz	173 MHz	221 MHz	208 MHz	245 MHz	239 MHz
48-bit Magnitude Compare	335 MHz	170 MHz	364 MHz	205 MHz	401 MHz	238 MHz
High-Speed Serial I/O	6.25 Gb/s	Not Applicable	10.3 Gb/s	Not Applicable	11 Gb/s	Not Applicable

The performance advantage of Virtex-4 FPGAs over the closest competitor can be as high as 142% in the case of the FIR filter example or 97% in the fabric with the magnitude comparator.

Design Entry, Tool Settings, and Constraints

Some competitors' claims of performance leadership are based on logic benchmarks using methodology, settings, and constraints to misrepresent the performance attainable in Virtex-4 FPGAs. A sound methodology for performance benchmarks requires the following:

- Compare similar device speed grades.
- Apply constraints in synthesis until slack is negative.
- Apply tight, but realistic constraints in place and route (performance might degrade if constraints are too tight).
- Use the High-effort choice as applicable in the synthesis and place and route tool.
- Use comparable settings. Using retiming in only one tool does not provide a sound comparison.
- Constrain the important clock(s). Using general global constraints only improves the slowest clock in the design.

The next sections provide some guidelines for getting great results with Virtex-4 FPGAs.

Design Entry

Design entry is a critical stage in the quest for a high-performance design. A hardware description language, such as Verilog or VHDL, provides an efficient way of designing for Virtex-4 FPGAs. To get the best possible results, Xilinx recommends following some coding-style guidelines for yielding the best implementation in the device.

First and foremost, pipelining is highly recommended; lots of registers in the design keep the critical path short and fill the pipeline stages in the dedicated blocks. An appropriate level of pipelining ensures maximum throughput.

In Table 2, the code on the left uses the new block RAM with one level of pipeline embedded in the block versus two levels for the code on the right. The latter code improves performance by 28%.

Table 2: **Using Fully Pipelined Block RAM(1)**

One Level of Pipeline	Two Levels of Pipeline

`//Always at positive edge clock`
`if (en & we)`
`mem[addr] <= din;`
`else`
`dout <= mem[addr];`
`end`
	`//Always at positive edge clock`
	`if (en & we)`
	`mem[addr] <= din;`
	`else begin`
	`dout_piped <= mem[addr];`
	`dout <= dout_piped;`
	`end`

Clock Frequency: 350 MHz	Clock Frequency: 450.0 MHz (+28%)

Notes:
1. Using a Virtex-4 device with -11 speed grade.

One additional important recommendation is to avoid functionality not natively supported in blocks being targeted. For example, using asynchronous reset for registers before and after the multiplier prevents these registers from being merged into the XtremeDSP slice as the block uses synchronous reset for its registers. As shown in Table 3, the impact on performance can be quite dramatic.

Table 3: **Impact of Reset on Performance(1)**

With Asynchronous Reset	With Synchronous Reset

`// Requires fabric flip-flops always at positive edge clock or negative edge reset`
`if (~rst) prod <= 32'd0;`
`else prod <= coeff * sample;`
	`// Efficient use of XtremeDSP slice always at positive edge clock`
	`if (~rst) prod <= 32'd0;`
	`else prod <= coeff * sample;`

Three levels of pipeline (16x16 mult): Clock Frequency: 206.4.4 MHz	Three levels of pipeline (16x16 mult): Clock Frequency: 450.0 MHz (+118%)

Notes:
1. Using a Virtex-4 device with -11 speed grade.

Tool Settings

To get the best possible performance it is important to follow the guidelines of the tool vendors. For example, synthesis tool vendors recommend constraining clocks that use separate clock groups when cross domain timing is not critical. The most important recommendation is to make sure the clock is constrained enough.

Retiming (a.k.a. register balancing) is an option that can boost the performance of a design by moving registers across levels of logic. Depending on the design, it can substantially improve timing. Although Xilinx does not use retiming in its own performance comparison tests, this technology can help meet performance targets.

Other important settings to consider:

- The resource sharing option of the synthesis tool, once turned off, can generate better results. If operators are in the critical path, it is a good idea to turn it off.
- If state machine logic is in the critical path, the finite state machine (FSM) optimization of the synthesis tool might greatly improve performance. This optimization should then be turned on.
- When running a small test case as a benchmark, where only the internal performance is of concern (not considering I/O performance), it is recommended to make sure that I/O flip-flops are placed in the fabric.

Constraints

Even with adequate tool settings and coding style, synthesis, and place and route only provide their best results with timing constraints. Using timing driven synthesis tools, only a negative slack is an indication that the design is sufficiently constrained. Generally, it is a good idea to over constrain synthesis, as long as the constraint is still realistic. It is also recommended to specifically target a clock (or clocks) of interest by constraining them. General performance constraints tend to work hard on the slowest clock of the design only; this is not necessarily what the design requires.

After synthesis, the Xilinx ISE place and route dramatically improves performance once constraints are applied. The effort level also impacts the result, and so it is recommended to use the High effort for best results.

Let's take a look at the impact of a constraint based on a Verilog Reed-Solomon design (rs_5_3_gf256) from **www.opencores.org**.

The first bar in Figure 1 shows the result obtained without any constraints applied in either synthesis or place and route used in its lowest effort.

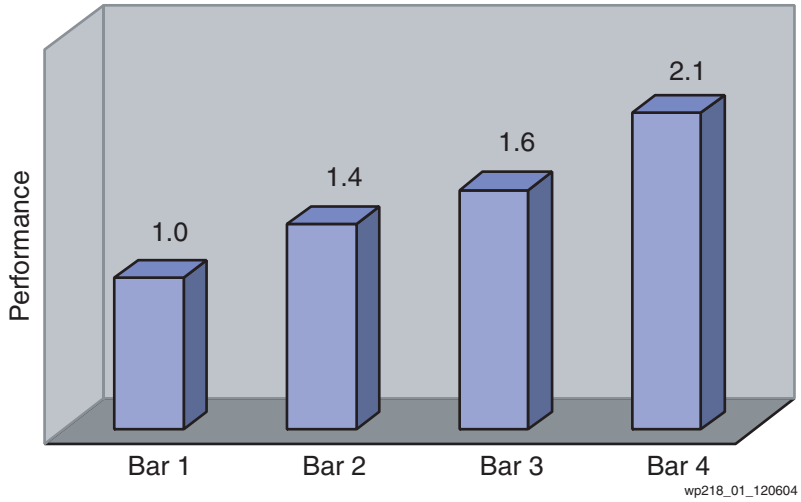

Figure 1: **Performance Variation**

The second bar shows the result after applying constraint in place and route and setting the effort level to High. Synthesis is not constrained. This shows a 40% improvement compared to original settings.

The third bar shows a new result after both synthesis and place and route are constrained with a High effort in place and route. At this point, the original design is improved by 60%.

Finally, the last bar on the right represents the result after constraints are applied and synthesis with its retiming option activated is used .

As we can see on this small test case, results can be vastly improved after tools are constrained. In this example, performance more than doubles compared to the original design without constraints.

To further improve performance with ISE, special options, such as timing-driven map or multi-pass place and route can help. These options can bring substantial improvement to designs with a higher level of utilization.

Benchmark Methodology

Among the many approaches to benchmarking FPGAs, this methodology makes design intent as a centerpiece of the flow. Alterations to user RTL code are kept to a minimum. Special HDL pragmas such as "parallel case" are preserved. The changes to the code pertain to inherent design conversion requirements as different FPGA technologies are used. In rare cases, code is also altered to remove latches, as latches produces different measurements based on the behavior of the timing analysis tools.

The benchmark methodology pushes the tools to provide the best possible logic fabric performance. To this end, based on user requirements, the clock of interest is identified. This clock is constrained during both synthesis and place and route.

Synthesis Flow

- Constraining specific clocks identified by the user is preferred to global performance constraints, as the latter would focus on slowest clocks or I/Os.
- When using Synplify, a clock group is created to isolate the clock and prevent interference from cross-clock domain paths if they exist, as in a vast majority of cases, these paths are not timing critical.

- Relaxing I/O constraints is also important to prevent interference with measuring internal paths. Constraining I/Os can yield a sub-optimal mapping in the FPGAs.
- Retiming is turned ON. When using the Stratix-II flow, the retiming option is turned ON in Quartus, see **Implementation (Place and Route)**.

The clock constraint is tightened by increments of 15% until the performance estimate of the synthesis tool does not indicate further improvement. The netlist associated with the best-estimated performance is kept for implementation. To generate the best result for synthesis, perform several timing driven runs.

Implementation (Place and Route)

The constraints are based on the same clock used at the synthesis stage. However, the estimated performance from synthesis is not used as a constraint for place and route as estimates can differ widely compared to actual results.

There are two place and route phases. In the first phase, a frequency sweep is performed using a high effort in the tools ("high" for ISE 'par' and "auto fit" in Quartus). Constraints are tightened by 5% increments until timing is not met.

In the second phase, designs are implemented using a 5% tighter constraint above the best result of first phase. During this new series of runs, ISE is used with different switches. Five implementations are performed with the following options:

- One run with timing driven map (`map -timing -ol high`) followed by high effort in par (`par -ol high`)
- One run with extra effort in par (`-xe n`)
- Three runs with high effort using cost tables 1, 2, and 3

To reproduce similar high performance results, Xilinx provides upon request the Xplorer™ script to combine the two phases together and provide comparable results. The script also offers a timing closure mode where user constraints are evaluated with the different Xplorer options.

Using Quartus, the Design Space Explorer tool is used in the "Physical Synthesis with Retiming Space" settings. This turns on retiming for the Quartus flow.

Xplorer Script

The benchmarks performed at Xilinx use comparable speed grades and settings for both Xilinx FPGAs and competing technologies. The chosen flow preserves most of the code and applies consistent constraints between synthesis and place and route. Results can be easily reproduced using the Xplorer script.

Conclusion

Virtex-4 FPGAs are the first FPGAs capable of reaching clock speeds of 500 MHz across many design building blocks (multipliers, adders, memory blocks, etc.).

Taking advantage of this unique level of performance in today's complex designs requires sound coding style practices, combined with appropriate tool settings and constraints. The Virtex-4 family represents a significant performance boost compared to any other FPGA architectures. Through its different families, the Virtex-4 platform offers a solution to closely match design requirements.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
12/06/04	1.0	Initial Xilinx release.
12/07/04	1.1	Switched the second row of Table 2. Edited Table 3.
02/14/05	1.2	Added **Benchmark Methodology**.