Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

MicroBlaze Can Float

Xilinx Adds Floating-Point Logic to FPGA-Optimized Processor Core

By Tom R. Halfhill {5/17/05-02}

Only AMD and Intel share a greater rivalry than Altera and Xilinx. These companies are the Hatfields and McCoys of the semiconductor industry. While the world's leading PC-processor vendors are shotgunning each other with double-barreled cores and 64-bit extensions, the world's leading FPGA vendors are battling over who has the biggest programmable-logic chips, the coolest design-automation tools, and the best synthesizable processor cores.

Altera fired the last shot by introducing Nios II at Embedded Processor Forum 2004. (See *MPR 6/28/04-02*, "Altera's New CPU for FPGAs.") Nios II is the second generation of Altera's 32-bit RISC architecture. Although it is primarily intended for FPGA integration, Nios II also offers a migration path to structured ASICs and conventional SoCs for chip designs reaching higher volumes.

This week at **Spring Processor Forum**, Xilinx is blazing back with MicroBlaze v4.00, the newest version of its 32-bit RISC processor core for FPGAs. (See *MPR 11/5/01-03*, "FPGAs Catch Fire at MPF.") The most significant improvement is an optional tightly coupled floating-point unit (FPU)—a feature missing from Nios II. Other MicroBlaze v4.00 enhancements include new pattern-compare instructions, improved debug logic, and higher clock speeds when the core is synthesized for Xilinx's fastest FPGAs. Also, Xilinx is making the 32-bit integer multiplier from MicroBlaze v3.00 an optional feature in v4.00.

Of course, processor cores aren't the front line in the battle between Altera and Xilinx. Their main business is selling programmable-logic devices. Their low-cost processor cores exist largely to create more demand for their chips—especially the bigger, more profitable devices. In addition, the cores fill a hole left by other sources of licensable processors. The leading processor-core vendors—ARC International, ARM, MIPS Technologies, and Tensilica—don't aggressively pursue the FPGA market, partly because of its lower volumes but also because they fear losing valuable intellectual property (IP). A thief with a modicum of technical knowledge could extract the synthesized processor from an FPGA and reuse it for another design, or even distribute the code as pirated warez without paying licensing fees or royalties.

It's not a wholly unreasonable fear, which is the reason ARM forbids licensees to deploy a design in an FPGA. ARC, MIPS, and Tensilica are willing to license their cores for FPGA deployment, but they don't go out of their way to encourage the practice, and they haven't optimized their processors for better performance in programmable logic. (However, all processor-IP vendors allow licensees to test and debug their designs in FPGAs, and they even sell prototyping boards for that purpose.) Notable exception: ARM is collaborating with Actel to modify an ARM7TDMI-S core for integration in a new series of FPGAs based on Actel's ProASIC3 family. ARM's register-transfer-level (RTL) model will be encrypted to protect it from hackers and thieves. (See *MPR 4/4/05-02*, "ARM-Based MCUs Flex Muscles.")

The recent improvements to MicroBlaze and Nios II indicate that Xilinx and Altera remain committed to their soft processors. Indeed, as the cost of spinning a custom chip continues to soar, FPGA vendors are eager to lure more customers away from ASICs and SoCs in favor of programmable-logic

devices, whose prices are steadily declining. Easy-to-license 32-bit processor cores optimized for FPGAs are the loss leaders in this strategy.

Economical FPU Saves Gates

Full-blown FPUs are so complex that PC processors didn't integrate them on chip until Intel introduced the 486 in 1989—and for years afterward, they remained optional. In those days, third-party suppliers like Weitek thrived by offering separate floating-point coprocessor chips. In the embedded-processor market, most traditional applications don't require floating-point math, so hardware FPUs still aren't commonplace. For occasional floating-point tasks, function libraries written in software offer a painfully slow but cheap alternative.

Mindful of the burden a full-featured FPU would impose on a simple RISC core, Xilinx is sticking to the basics. The MicroBlaze v4.00 FPU is optional and limited to performing single-precision, 32-bit operations. It adds only 10 new instructions to the existing architecture, which is backward compatible with MicroBlaze v3.00. Limiting the FPU to single precision is an economical trade-off, because 32-bit operations are sufficient for the vast majority of embedded applications that require floating point, and single precision also allows the operands to use the existing 32-bit integer registers—eliminating the need for a new 64-bit register file.

Xilinx focused on implementing the most commonly used floating-point instructions in hardware: basic arithmetic operations and equality comparisons. In MicroBlaze v4.00, execution latencies for arithmetic operations range from 6 clock cycles for addition, subtraction, and multiplication to 30 clock cycles for division. Floating-point compare instructions (which are exact-match operations, not approximations) execute in three cycles. Table 1 shows all the new instructions in the MicroBlaze v4.00 instruction set, along with descriptions and execution latencies.

The tightly coupled FPU isn't independently pipelined, so floating-point instructions share the same three-stage pipe as integer instructions. This was another trade-off for the sake of economy. Sharing the same short pipeline and hazard-resolution logic shrinks the FPU but compromises performance, because the multicycle floating-point instructions stall the pipe while executing. Nobody in their right mind will complain, because the new instructions are a huge improvement over calling equivalent functions in the floating-point library. A single new instruction (fmul) can replace as many as 1,200 integer instructions that require 1,600 cycles to execute.

Although the performance improvement with the other new floating-point instructions is somewhat less dramatic, the savings always amount to hundreds of cycles. And because floating-point instructions share the integer pipeline, they execute at the same clock speed as any other instructions. In the fastest Xilinx Virtex-4 device, a MicroBlaze v4.00 processor can reach 200MHz, which yields peak throughput of 33 million floating-point operations per second (MFLOPS).

FPU Speeds Up Practical Applications

In embedded applications—such as motor control, industrial-machine control, multimedia, and printing—an FPU can make a big difference. Xilinx benchmarks show that JPEG decompression is six times faster; fast-Fourier transforms (FFT) are 50 times faster; and finite impulse response (FIR) filters are 120 times faster.

Recompiled software libraries can also use the new instructions to accelerate their complex functions. Xilinx says a MicroBlaze v4.00 processor with FPU can call as many as 82,000 library functions per second, compared with only 5,300 functions per second for the same processor without an FPU. (Unfortunately, there are no EEMBC benchmark scores for either MicroBlaze or Nios II. Xilinx doesn't belong to the benchmarking consortium, and although Altera is a board member, it hasn't published any scores.)

Xilinx cut a few corners with IEEE 754 compliance but supports the most common features. The MicroBlaze FPU performs signed-zero and infinity operations but supports only one rounding mode (round to nearest, not up or down). For indeterminate operations that generate a "quiet" not-a-number (NaN) value, the FPU substitutes a fixed NaN. A signaling NaN will trigger an exception if exceptions are turned on. Likewise, the FPU triggers an exception if

Instruction	Description	Latency	Notes
Single-Precision Floating-Point Instructions			
fadd	FP arithmetic add	6 cycles	Replaces 450 instructions (addsf3), 600 cycles
frsub	Reserve FP arithmetic subtraction	6 cycles	Replaces 450 instructions (subsf3), 600 cycles
fmul	FP arithmetic multiplication	6 cycles	Replaces 1,200 instructions (mulsf3), 1,600 cycles
fdiv	FP arithmetic division	30 cycles	Replaces 600 instructions (divsf3), 750 cycles
fcmp.lt	FP compare (less than)	3 cycles	Replaces 350 instructions (ltsf2), 450 cycles
fcmp.eq	FP compare (equality)	3 cycles	Replaces 350 instructions (eqsf2), 450 cycles
fcmp.le	FP compare (less or equal)	3 cycles	Replaces 350 instructions (lesf2), 450 cycles
fcmp.gt	FP compare (greater than)	3 cycles	Replaces 350 instructions (gtsf2), 450 cycles
fcmp.ne	FP compare (not equal)	3 cycles	Replaces 350 instructions (nesf2), 450 cycles
fcmp.ge	FP compare (greater or equal)	3 cycles	Replaces 350 instructions (gesf2), 450 cycles
Pattern-Compare Instructions			
pcmpbf	Pattern-compare byte find	1 cycle	Find first matching byte in two 32-bit words
pcmpeq	Pattern-compare equal	1 cycle	Return (a==b); replaces four instructions
pcmpne	Pattern-compare not equal	1 cycle	Return (a!=b); replaces four instructions

Table 1. The tightly coupled FPU, a new option, adds only 10 instructions to the MicroBlaze v4.00 architecture, which is backward compatible with v3.00. Note the vastly improved performance over functions in the floating-point software library. MicroBlaze v4.00 also has new pattern-compare instructions for faster string operations.

an input is denormalized, or it returns a fixed NaN if exceptions are off. If a result is denormalized, the FPU flushes the result to signed zero and sets the underflow bit.

Despite all the economy measures, the FPU still outweighs the base CPU. The number of programmable-logic cells—or, in Xilinx parlance, lookup tables (LUT)—required for the FPU varies slightly, depending on the device for which the FPU is synthesized. On average, the FPU needs 950 to 1,100 LUTs. In comparison, the base configuration of the MicroBlaze CPU needs at least 950 LUTs. The grand total of 1,900 to 2,000 LUTs—which grows fatter after more options are added to the base configuration—will undoubtedly deter some designers. Doubling the processor's size by adding the FPU might bump up the design to a larger-capacity programmable device, possibly busting the budget. Other designers who need fast floating-point performance will be happy to pay the price.

The smallest Xilinx FPGA that can accommodate a MicroBlaze v4.00 processor with FPU is the Spartan-3E XC3S250E, which has 4,896 LUTs and will cost about $2.95 in volume in 2H05. That's a tight fit, however. Designers usually want a larger device with plenty of room to augment the processor core with application-specific functions and peripherals. The largest FPGAs from Xilinx are those in the Virtex-4 series, which provide up to 178,176 LUTs. Volume prices of those devices range from $25 to several hundred dollars, depending on the number of LUTs.

In addition to new floating-point instructions, MicroBlaze v4.00 also has three new pattern-compare instructions, listed in Table 1. One new instruction (pcmpbf) finds the first byte that matches when comparing two 32-bit words, and the others (pcmpeq and pcmpne) compare pairs of 32-bit words for Boolean equality or inequality. These instructions accelerate string functions—such as strcmp(), strcpy(), and strlen()—that would otherwise require shift and mask operations. According to Xilinx, finding the length of a 1,024-byte string with strlen() is 40% faster and requires 30% fewer instructions with MicroBlaze v4.00 than with v3.00.

New debug logic in MicroBlaze v4.00 is only half as large as the debug logic in v3.00, but it allows designers to download data up to 15 times faster: 108KB/s over a parallel interface or 216KB/s over USB. The debugger can insert instructions directly into the processor's pipeline and access any register.

MicroBlaze v4.00 and previous MicroBlaze processors are compatible with GNU C Compiler (GCC) 3.4.1, a step up from GCC 2.9. All together, the hardware improvements and better compiler will boost MicroBlaze's performance to 0.92 Dhrystone mips per megahertz, compared with 0.79 Dhrystone mips per megahertz for MicroBlaze v3.00.

MicroBlaze Fares Well Against Competition

Although Nios II is the most obvious competition for MicroBlaze—both being synthesizable 32-bit RISC processors intended primarily for FPGAs—in a sense, they don't compete directly with each other at all. The reason is that each processor architecture confines the customer to a single FPGA vendor. Anyone licensing MicroBlaze must use it with a Xilinx chip, and anyone licensing Nios II must use it with an Altera chip. The cores aren't compatible with the integration tools for other FPGAs, and their licenses forbid cross-pollination.

Therefore, the factors driving a decision between MicroBlaze and Nios II aren't limited to the features of the processors themselves. Other factors, such as chip prices and tool preferences, may overshadow the processors. It's a little different from licensing any other soft processor core, which a customer can fabricate at virtually any foundry.

The arguments for choosing Altera over Xilinx, or vice versa, could fill a book (or start a war), so we'll focus on comparing the processors. MicroBlaze v4.00 has one significant advantage over Nios II: the new FPU. Nios II doesn't have one. So if an embedded application needs faster floating-point performance, MicroBlaze is a no-brainer, all other things being equal. Even their integer performance is similar: Xilinx claims 0.92 Dhrystone mips per megahertz for MicroBlaze v4.00, and Altera claims 1.1 Dhrystone mips per megahertz for Nios II. Given the well-known shortcomings of the moth-eaten Dhrystone benchmark, we call it a draw.

On the other hand, Nios II has one significant advantage over MicroBlaze: a user-configurable instruction set. Designers can create their own application-specific custom instructions for Nios II, whereas MicroBlaze is limited to its factory-equipped instructions. As ARC and Tensilica have demonstrated with their certified EEMBC scores, a few well-crafted custom instructions can be worth hundreds of megahertz. Although Altera hasn't published EEMBC scores for Nios II, the company does offer anecdotal evidence. One customer (which Altera says it cannot name yet) recently replaced 16 discrete DSPs with five Nios II cores on a single FPGA by creating application-specific instructions.

Instead of allowing developers to add custom instructions, Xilinx equips MicroBlaze with a fast interface for application-specific coprocessors. Called the Fast Simplex Link (FSL), this interface connects the core directly to the programmable-logic fabric. Developers can configure MicroBlaze with up to eight output (master) FSL channels and eight input (slave) FSL channels. Each channel is 32 bits wide, so it can transfer 32-bit words of data to and from the CPU's register file using simple get and put instructions. Implementing custom functions in external coprocessors is usually less efficient than adding new instructions to the CPU, but it has one advantage: it prevents multicycle operations from stalling the main pipeline or impairing the CPU's maximum clock frequency.

Configuration Options Add Flexibility

MicroBlaze has user-selectable options that allow designers to exercise some control over the size of the core as well as its performance. Options include the instruction and data caches (variable from 0KB to 64KB), a 32-bit integer multiplier, a barrel shifter, a hardware exception handler, the

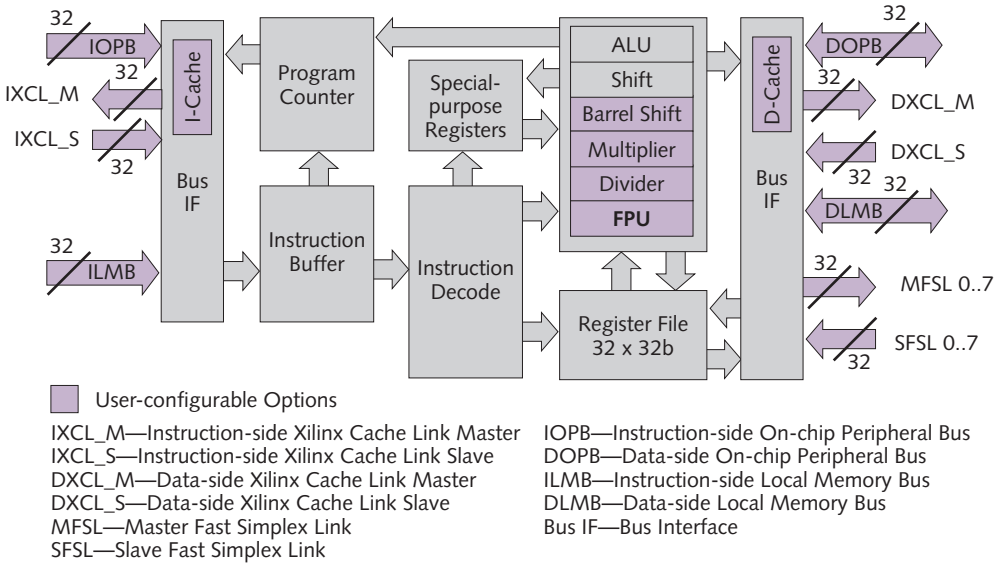

Figure 1. MicroBlaze v4.00 block diagram. Blocks shaded in purple indicate optional or user-configurable features. The base configuration of the processor requires 950 to 1,100 look-up tables (LUT) in a Xilinx FPGA. The new FPU requires about 1,300 LUTs. A fully configured MicroBlaze with FPU and 64KB instruction and data caches would require about 2,600 LUTs.

debug logic, the FPU, and the new pattern-compare instructions. Figure 1 shows a block diagram of MicroBlaze v4.00, highlighting the optional blocks.

Altera provides more or less the same flexibility by offering Nios II in three variations. Nios II/e is the economy model, lacking caches, a multiplier, a barrel shifter, and branch prediction. Nios II/s is the standard model, with a deeper pipeline (five stages), instruction cache (configurable from 0KB to 64KB), 32-bit multiplier, barrel shifter, and static branch prediction. Nios II/f is the fast model, offering everything in Nios II/s plus an even deeper pipeline (six stages), a data cache (0–64KB), and dynamic branch prediction. All Nios II processors have another feature missing from MicroBlaze: multiple privilege levels, allowing them to isolate user tasks from system tasks. Table 2 summarizes the features of the Xilinx and Altera processors and compares them with synthesizable processors from ARC and Tensilica, the two IP vendors least hostile to FPGA integration.

Xilinx offers yet another alternative: embedding one to four PowerPC 405 hard-core processors in a programmable-logic device. Of course, this option is entirely different from synthesizing a soft core like MicroBlaze, because the hard core doesn't run in the programmable fabric and can therefore attain much higher clock speeds. The PowerPC 405 is

Feature	Xilinx MicroBlaze v4.0	Xilinx MicroBlaze v3.0	Altera Nios II /f	Altera Nios II /s	Altera Nios II /e	ARC ARC 600	Tensilica Xtensa LX
Architecture	**MicroBlaze v4.0**	MicroBlaze v3.0	Nios II	Nios II	Nios II	ARCompact	Xtensa
Instr Length	**32 bits**	32 bits	32 bits	32 bits	32 bits	16–32 bits	16–24 bits*
Configurable ISA	**—**	—	Yes	Yes	Yes	Yes	Yes
Pipeline Depth	**3 stages**	3 stages	6 stages	5 stages	1 stage †	5 stages	5–7 stages ‡
I-Cache	**0–64KB**	0–64KB	0–64KB	0–64KB	—	0–32KB	0–32KB
D-Cache	**0–64KB**	0–64KB	0–64KB	—	—	0–32KB	0–32KB
32-bit Multiplier	**Optional**	Optional	Yes	Yes	—	Optional	Optional
Barrel Shifter	**Optional**	Optional	Yes	Yes	—	Optional	Optional
DSP Extensions	**—**	—	—	—	—	Optional	Optional
MMU	**—**	—	—	—	—	—	—
FPU	**Optional 32-bit**	—	—	—	—	Optional 32/64-bit	Optional 32-bit
Branch Predict	**—**	—	Dynamic	Static	—	Static	—
Privilege Levels	**1**	1	2	2	2	1	2
Core Freq (Max)	**205MHz****	150MHz	140–180MHz ††	140–180MHz ††	140–200MHz ††	290MHz ‡‡	350MHz ‡‡
Logic Cells	**950–2,400**	950	~1,800	~1,150	~600	n/a	n/a
Introduction	**May 2005**	2001	2004	2004	2004	2003	2004

Table 2. The new Xilinx MicroBlaze v4.00 has one significant advantage over MicroBlaze v3.00 and Altera's Nios II: an optional 32-bit FPU. Altera's most significant advantage is a user-configurable instruction set, although multiple privilege levels can also make its processors more suitable for secure applications. In general, the ARC 600 and Xtensa LX are more powerful and flexible than either MicroBlaze or Nios II, but the ARC and Tensilica processors aren't optimized for programmable logic, and their vendors are less enthusiastic about FPGA integration. *Tensilica's FLIX option permits shorter instructions. †Nios II/e has a six-stage pipeline, but it works like a one-stage pipe. ‡Xtensa LX pipeline depth is user configurable. **Maximum clock frequency in a Xilinx Virtex-4 device. ††Clock-frequency range in Altera Stratix and Stratix-II devices. ‡‡Approximate worst-case clock frequency in 0.13-micron CMOS, not programmable logic. n/a = not available.

available in some Virtex-II Pro and Virtex-4 FX devices at clock frequencies as high as 450MHz. (See *MPR 11/5/01-03*, "FPGAs Catch Fire at MPF.") Even at that frequency, however, the PowerPC 405 can be slower than a MicroBlaze core when executing floating-point operations, because the 405 lacks an FPU. Xilinx plans to add one later this year. (Third-party IP providers already offer FPUs for the PowerPC 405.)

The Price Is Right

One way in which Xilinx and Altera beat the licensable-IP competition hands down is pricing. Both companies offer synthesizable models of their processors, complete development kits, documentation, and royalty-free perpetual-use licenses for single- or multicore designs at the incredibly low price of $495. For $995, Altera even throws in a development board and interface cables. To put things in perspective: those prices are approximately three orders of magnitude lower than the cost of a single-use, single-core license from ARC, ARM, MIPS, or Tensilica. Multiple-use, multicore licenses from those companies can cost millions of dollars. And when the finished chip enters production, their customers owe per-chip royalties, as well.

No wonder the licensable processor cores from Altera and Xilinx have attracted so much interest. Altera claims to have shipped 14,000 Nios or Nios II development kits to more than 4,500 unique licensees. ARC, ARM, MIPS, and Tensilica boast of having dozens of licensees—or a hundred. Sure, many (if not most) of the Nios or MicroBlaze kits were probably sold to the merely curious, to engineering students, and to others who will never deploy an actual chip design in a commercial product. But the seeds are planted, and designs are starting to sprout.

Of course, Xilinx and Altera can afford to practically give away their processors, because their real strategy is to generate more demand for FPGAs. The processor-IP companies, in contrast, have built their business on selling licenses and collecting downstream royalties. As one might expect, a $495,000 license buys a more powerful processor, more compatibility with other IP, more-attentive technical support, and better development tools than a $495 license does. However, those differences don't loom as large as the price gap, especially with regard to the tools. The FPGA vendors are tossing in some surprisingly sophisticated development software.

As the cost of designing and manufacturing custom chips continues to rise and the prices of FPGAs continue to drop—two seemingly irreversible trends—the balance will tip further in favor of FPGA vendors. Their licensable processor cores, now mere loss leaders, will become more central to their mainline business strategies. Given the already lively competition between Xilinx and Altera, we expect to see them push the evolution of their processor cores more aggressively in the future. Today, their processors are still relatively simple microarchitectures having similar features. The competition will heat up before long. Meanwhile, the other processor-IP vendors will face the quandary of either ignoring a growing market for FPGA-based designs or modifying their business models so they can compete with processors that are practically free. ◇

Price & Availability

Xilinx is shipping the MicroBlaze v4.00 processor now. The Xilinx Embedded Development Kit 7.1i includes a synthesizable RTL model of the MicroBlaze core and the Platform Studio 7.1i development tools. The synthesizable model is encrypted; Xilinx synthesis tools produce a gate-level netlist. For $495, the license allows unlimited use of the processor in single- or multicore designs in Xilinx programmable-logic devices. For $995, customers can license the same package with an unencrypted VHDL model of the processor. For more information, see *www.xilinx.com/microblaze*.