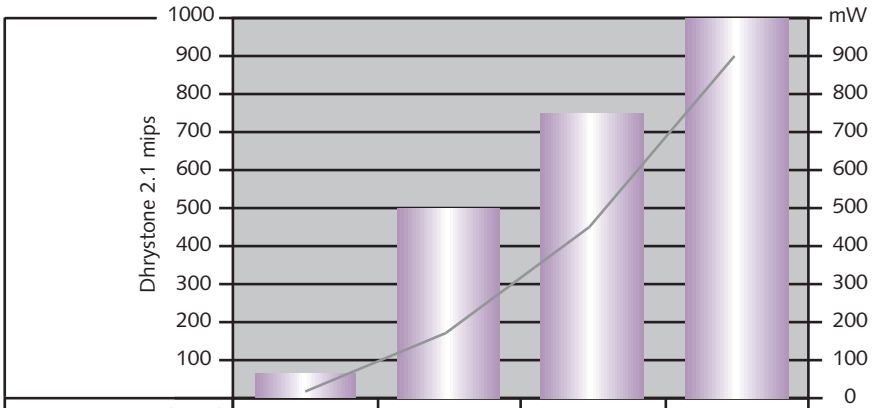
Cahners
MICROPROCESSOR REPORT
www.MPRonline.com
THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

XScale (StrongARM-2) Muscles In

1,000 mips at 900mW to 62 mips at 10mW, Sleeps at 0.1mW

By Steve Leibson {9/11/00-01}

OK, it's late. Intel disclosed substantial information about the StrongARM-2 microarchitecture at last year's Embedded Processor Forum (see *MPR 5/10/99-01*, "StrongARM Speed to Triple") and said that a processor based on the new microarchitecture would appear within a year. It's actually been a few more months than a year, but Intel finally demonstrated early processor silicon at the Intel Developer Forum (IDF) in San Jose last month. Intel now calls the processor XScale, for its extraordinary ability to scale over wide power and performance ranges. The name XScale is also a new brand that Intel can trademark separately from ARM. The XScale core is a 32-bit embedded processor that pushes both the lowest power dissipation (10mW) and highest performance (1,000 mips) extremes in the embedded world. Naturally, you don't get both low power dissipation and extremely high performance at the same time. You do get your choice. The XScale microarchitecture's low-power characteristics derive from design techniques that are becoming standard in processor design—extensive use of clock gating, dynamic voltage and frequency scaling—and some that are not so standard, like source-to-bulk back biasing. The processor's greatly enhanced performance stems from a high maximum clock rate (800MHz) and high-speed circuit- and logic-design techniques like differential-domino logic, use of pass gates, time borrowing, and source-to-bulk back biasing. In addition, it contains architectural enhancements: an enhanced ARM



Core Frequency (MHz)	50	400	600	800
Dhrystone 2.1 mips	62	500	750	1,000
mW	10	180	450	900
Core Voltage	0.75V	1.00V	1.30V	1.65V

Figure 1. A wide range of core voltages allows the XScale processor to achieve either very low power operation at reasonable performance (62 mips at 10mW) or very high performance (1,000 mips at 900mW). Although the graph shows four data points, power and frequency are continuously variable between the extremes shown. Also, the XScale processor continues to run while the core voltage and frequency change to the new set point. (Note: Power dissipation figures include the XScale core, MMU, 66K of caches, and the DSP coprocessor extensions. Peripheral circuits, not represented in the numbers listed here, frequently add substantially to the overall power dissipation of an integrated processor.)

V5TE (Thumb plus E DSP extensions) instruction set; larger caches; a branch-target buffer that adds dynamic branch prediction; and additional SIMD instruction enhancements. Five Intel divisions involved in the Internet backbone—telecom, storage, handheld PC, and cell phone arenas—are committed to using the XScale microarchitecture, assuring the processor of a critical mass of design wins, with volume production from the start.

And the Process You Rode In On

XScale rides in on a new process that gives the processor its extremely wide power and performance capabilities. The original StrongARM processors could be built on only a tweaked 0.35-micron process used in the Hudson fab Intel obtained when it purchased part of Digital Equipment Corp. in 1998. Intel will build XScale on a six-layer-metal, 0.18-micron process closely related to the P858 process used for Coppermine. Although P858 is rated for 1.5V operation, XScale operates on core voltages ranging from 0.75V to 1.65V. In fact, Intel pushed the XScale device it demonstrated at IDF to 1GHz, with a core voltage of 1.75V, but the company is unwilling to commit to that top-end figure on the official datasheet. As Figure 1 shows, the lowest core voltage allows the processor to run at 50MHz while dissipating 10mW. The official upper core-voltage limit allows the processor to run at a still-respectable 800MHz and reach 1,000 mips.

It's incredible that the original StrongARM processor, based on Digital's dated manufacturing process, maintained its power/performance leadership in the embedded world for so long. That it did so is a real testament to the design and process engineers who created the original StrongARM. But XScale is not saddled with an old fab process. It makes the most of its shiny new 0.18-micron process through design techniques that boost performance and squeeze out wasted power. The first of these techniques, clock gating, builds on those used by the original StrongARM developers at Digital. Clock gating was certainly employed in the original StrongARM design, but, according to Intel, it has been taken to an "unprecedented granularity level" for the XScale microarchitecture. Circuits with no clock dissipate no dynamic power, so it makes tremendous sense to cease clocking them when they're not doing useful work. However, logic circuits still dissipate static power (leakage), even when not clocked, so XScale designers employed source-to-bulk back biasing to reduce the XScale's static power dissipation. The normal leakage in the P858 process is 3nA/μm of gate length. Intel says the back bias substantially reduces leakage current below this figure but will not give an exact number.

The Power/Performance Slider

Circuit tricks are not the only techniques used to drop XScale's power levels. The new processor uses architectural techniques found in other processors to allow programmatic control over instantaneous power dissipation. XScale employs dynamic voltage and frequency scaling and sports one new low-power mode (standby). The standby mode takes advantage of XScale's pseudostatic design to stop the processor clock and, when invoked, drops power dissipation to 0.1mW. The processor's PLL requires only 20 μsec to relock when exiting standby mode. Consequently, standby mode should prove far more popular for saving power during idle periods than the original StrongARM's idle and sleep modes.

Dynamic frequency and voltage scaling allows the running program to set core voltage and frequency on the fly, to meet the transient processing-power requirements of the tasks at hand, while running at the lowest possible power levels. To achieve this, the program must have access to a power DAC driving the core's power supply and also to the on-chip PLL generating the core clock frequency. The core continues to run while the core operating frequency and voltage change. Most recently, this sort of power management has been touted by Transmeta, which incorporated its LongRun power-management system into its Crusoe line of x86-compatible processors (see *MPR 7/10/00-02*, "Top PC Vendors Adopt Crusoe"). LongRun relies on subroutines within Crusoe's VLIW code-morphing software to determine processor loading and to adjust the core power and frequency settings appropriately. Transmeta's code-morphing software runs beneath the application-level code, so the LongRun features are not directly accessible to the application.

This approach is necessary for the x86 world, where no other processors have this capability yet, and therefore no existing PC application software can exploit LongRun's abilities. However, systems designers employing XScale generally don't need to back-fit power-management code into an existing operating system like Microsoft Windows, because embedded environments are nowhere near as standardized as environments in the PC world. Consequently, XScale's dynamic voltage and frequency-scaling abilities are directly accessible to the application code through standard ARM coprocessor registers CP14 and CP15, unlike LongRun, which is accessed only by Transmeta's own VLIW code.

XScale's voltage-scaling abilities appear to exceed Crusoe's by a substantial amount. Transmeta's TM5400 processor can step its core voltage (using an external power DAC) from 1.1V to 1.6V in 32 steps, although Crusoe's developers admit that 5–7 steps are sufficient to realize most benefits of this technology (at least for PC-oriented applications). XScale's core-voltage range of 0.7–1.65V nearly doubles Crusoe's range. Although XScale's 0.45V increase in core-voltage operating range may not seem like much, power dissipation varies with the square of the operating voltage, so this wider core-voltage swing is one key to XScale's huge performance and power ranges. Like Crusoe's designers, XScale's design team leveraged the split core and I/O power planes and the power DAC technology developed for the Pentium generation of PC processors. Although Figure 1 shows only four voltage steps, XScale's core voltage can be varied continuously over the entire range. Intel also admits that it may combine the XScale core with on-chip power

DACs in some future XScale derivative chips. Some truly impressive mixed-signal semiconductor process (surely not just a modified P858) will be required to pull off an on-chip power DAC for XScale's full 10–900mW power range; however, the marriage of processor and power DAC promises to reduce overall system costs for many embedded applications that use the XScale processor.

Some of XScale's enhanced performance stems from circuit- and logic-design techniques, and some is due to architectural enhancements. For certain circuits that require high speed, XScale's designers employed differential-domino circuit design. Differential-domino circuitry uses precharged logic to speed processing through several asynchronous circuit stages within a clock period. The precharged domino circuits speed logic signals in much the same way that laser-pumped cesium cells (used in recent experiments conducted by the NEC Research Institute in Princeton, New Jersey) appear to accelerate the propagation of optical pulses beyond the speed of light (see "Gain-Assisted Superluminal Light Propagation," *www.neci.nj.nec.com/homepages/lwan/gas.htm*). Neither the differential-domino circuits nor the cesium cell actually produces FTL (faster-than-light) phenomena, but both inject energy into a physical system to set up triggered events with low-energy trigger thresholds, so that the far leading edge of a pulse can trigger an event much earlier than otherwise possible. The additional energy supplied by the precharge or optical pump effectively accelerates signal transmission, at least along a short path. Differential-domino circuits are fast, but they consume more power than normal, so the XScale designers used them sparingly, so as not to give back the low-power gains made with the low-power-design techniques listed above.

Seven or Ten Pipeline Stages—Who's Counting?

Circuit tricks, variable core voltages and clock frequencies, and advanced semiconductor process technology create a marvelous platform for the XScale architecture. As discussed in last year's article on the StrongARM-2, the new core sports many architectural features that boost performance beyond that delivered by raw clock speed. Figure 2 shows that XScale's integer pipeline is now ostensibly seven stages. Stretching the pipeline from five to seven stages allows a 50% jump in clock speed (in the same semiconductor process) and allows Intel to capitalize on the newer 0.18-micron fab process to an even greater extent than a five-stage pipeline would. In fact, Intel's remodeling of the StrongARM pipeline really goes beyond the seven-stage stretch. Using a technique called time borrowing, the pipeline designers stretched some operations to 1.5 pipeline stages (fetch, execute, and exception) and shrank another (write-back) to about half a pipeline stage. They did this by using transparent latches instead of D flip-flops in the pipeline's design. As long as the latch remains open, the operation continues to transpire, so clock edges are much less important in XScale's pipeline design than for many other processors. In some ways, XScale's pipeline behaves as if it has ten stages, although an instruction needs only seven clocks to move through the pipe.

However, stretching the pipeline invokes the long-pipeline tax: big delays when branches are mispredicted. Consequently, XScale adds a branch-target buffer to reduce the occurrence of missed-branch pipeline bubbles.

Microprocessor Report's earlier StrongARM-2 article also speculated on the instructions that might be added to the new processor, and these speculations have proved remarkably prescient. As predicted, a large number of XScale's instruction enhancements are DSP related. Intel's XScale employs ARM's V5TE instruction set, which was introduced at the Embedded Processor Forum in 1999 and is now offered in ARM9E-S family (see *MPR 6/21/99-03*, "ARM Refocuses DSP Effort"). These enhancements are meant to replace ARM's less than successful Piccolo DSP extensions and include

- Five one-cycle 16 x 16 and 32 x 16 MAC instructions
- Four zero-overhead saturation extensions to existing arithmetic instructions
- Instructions to load and store register pairs
- A CLZ (count leading zeroes) instruction
- A cache-preload instruction

The one-cycle MAC obviously will speed many DSP algorithms, which are almost universally based on the execution of many MACs. The zero-overhead saturation instructions improve the performance of stable control loops and bit-exact algorithms that require saturating arithmetic. Examples of applications that require saturating arithmetic include GSM implementations, FFTs, and state-space servo loops employed in hard-disk drives for positioning control.

However, Intel apparently doesn't believe that the v.5TE extensions go far enough, because XScale's designers have added a 40-bit accumulator and six new SIMD instructions that use this accumulator to XScale's bag of tricks. Table 1 shows these new SIMD coprocessor instructions. Intel added these instructions through ARM's coprocessor mechanism, but the actual implementation is part of the

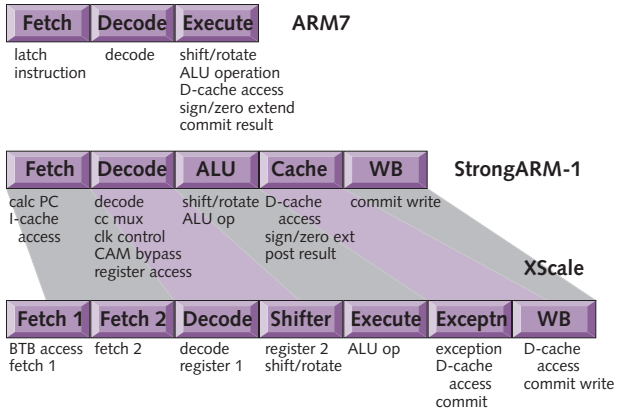


Figure 2. XScale's pipeline consists of seven stages, but in some stages, operations occur within half a clock cycle. Consequently, some versions of XScale may someday appear with ten pipeline stages.

Instruction	Operation	Comments
MIAPH acc0, Rm, Rs	(16 x 16) + (16 x 16) + 40 → 40	16-bit SIMD MAC
MIAxy acc0, Rm, Rs	(16 x 16) + 40 → 40	MIABB, MIABT, MIATB, MIATT
MIA acc0, Rm, Rs	(32 x 32) + 40 → 40	
MAR	40-bit acc → r1, r2	Move accumulator to register pair
MRA	r1, r2 → 40-bit acc	Move register pair to accumulator

Table 1. Intel added several SIMD instructions to the XScale as coprocessor instructions to boost performance in various audio applications.

processor core. Intel has employed the standard calling mechanism that ARM created to allow licensees to add architectural improvements, at the same time implementing the enhancements in a way that achieves single-cycle performance and also highlights the special nature of Intel's ARM license. The company says it plans to use these SIMD extensions for audio processing and has defined a coprocessing engine with eight accumulators; however, it has implemented just one in this incarnation of XScale.

Figure 3 shows the operation of the basic SIMD MAC, called MIAPH. Two registers are treated as two pairs of 16-bit numbers. The top 16 bits of each register are multiplied together, and the low 16 bits also are multiplied together. The results of these two multiplications are then added to the contents of the 40-bit accumulator, and the result is placed into the 40-bit accumulator in coprocessor space.

The MIAxy instruction is somewhat more complex. It involves the single multiplication of two 16-bit numbers, taken from either the top (high half) or bottom (low half) of the two source registers. There are thus four combinations of source operands, resulting in four different instructions: MIABB, MIABT, MIATB, MIATT. The selected 16-bit quantities are multiplied and then added to the contents of the accumulator, with the final result placed back into the 40-bit accumulator. The XScale microarchitecture can issue an MIAxy instruction every cycle, with a two-cycle latency.

Figure 4 illustrates the operation of the four MIAxy instructions. For symmetry, there is an MIA instruction that treats the two source registers as 32-bit quantities and uses the 40-bit accumulator, as Figure 5 shows.

Intel's addition of these coprocessor SIMD instructions seems to dovetail nicely with ARM's own E DSP extensions. In particular, the combination of instructions that load and store register pairs plus XScale's SIMD MACs and the 40-bit-accumulator move instructions allows a programmer to create tight code for handling media streams. In fact, Intel claims it worked with ARM to add the load-double instructions to the v.5TE definition expressly to augment XScale's SIMD extensions. The proof is in the performance. At 800MHz, Intel expects the XScale to execute 650 MMACs/s. Intel's own benchmarks indicate that the XScale enhancements triple speech-coding performance over that of the original StrongARM processor on a cycle basis alone.

For now, the configuration of the XScale SIMD extensions and the almost exclusive focus on 16-bit quantities limit these extensions to audio processing. Video streams really need 8-bit SIMD extensions and, although it's possible to use XScale's SIMD unit for video processing, the present SIMD unit is not very efficient. Intel claims that these SIMD extensions are the first of a planned series of multimedia extensions, so future versions of XScale could indeed incorporate video-oriented SIMD instructions.

In addition to the ARM and Intel DSP extensions, XScale now incorporates the ARM Thumb extensions (introduced in 1995, see *MPR 3/27/95-01*, "Thumb Squeezes ARM Code Size") that shoehorn 36 ARM instructions into a 16-bit instruction set. Thumb also limits programs to eight general-purpose registers. Although this shrunken instruction set and miniature register file truly represent reduced-instruction-set

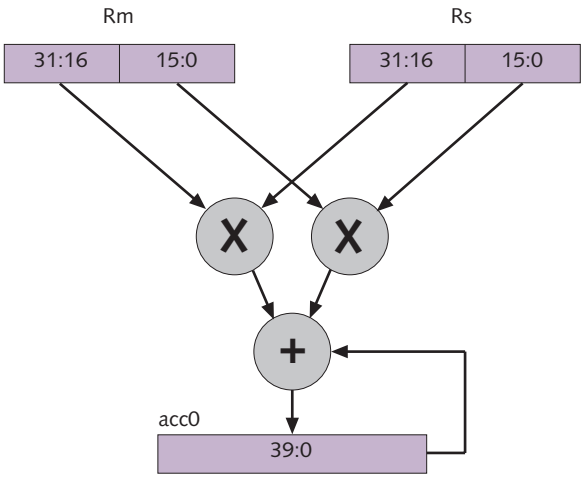


Figure 3. XScale's SIMD MIAPH instruction performs two 16 x 16-bit multiplications and a 40-bit addition in one clock cycle.

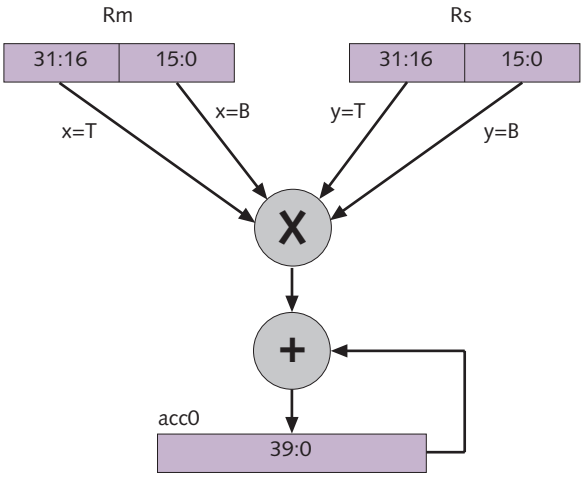


Figure 4. There are four combinations of 16-bit entities for XScale's SIMD MIAxy instruction, resulting in four different instructions.

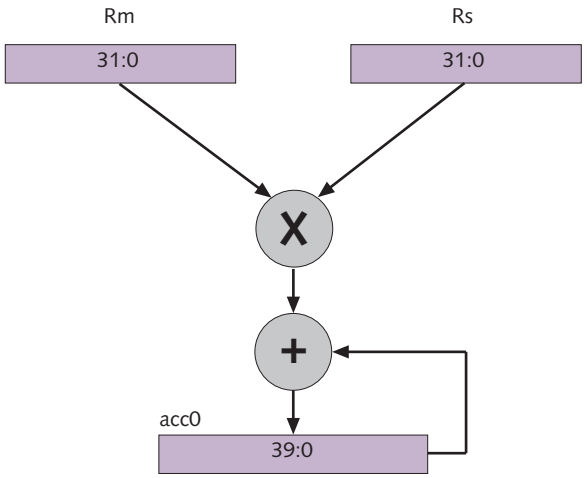


Figure 5. XScale's SIMD extensions can also perform a 32 x 32-bit multiply and 40-bit accumulate.

computing, their real purpose is code-size reduction; Thumb can reduce code size by 40% in some applications. By Intel's reckoning, the Thumb extensions can reduce power by allowing smaller caches and on-chip memories—which is true if most of the application runs in Thumb mode. Conversely, more Thumb code will fit in a cache of any given size, when compared to normal ARM code so there may be no performance hit from using the Thumb extensions if the instruction and register shortcomings don't overly crimp the program code.

ARM implements Thumb as a code preprocessor, and there is no reason to believe that Intel is doing anything differently. A mode bit turns the Thumb decompressor on and off, so Thumb and normal code cannot be conveniently intermixed. It's doubtful that Intel expects every XScale application to use the Thumb extensions, but the Thumb decompressor requires only about 3,000 transistors. The XScale design probably devotes far more transistors to clock gating than to its Thumb decompressor. The big reason for incorporating the Thumb instruction extensions to XScale, however, is simply market demand—the cell-phone industry has almost universally adopted the Thumb instruction set to save on memory costs.

Ultimately, the utility of these XScale instruction extensions can be realized only through efficient coding. Consequently, Intel plans to offer what it has dubbed Integrated Performance Primitives (IPP) to exploit various features in the XScale architecture. The purpose of these primitives is to provide better efficiency (higher performance at lower power) for audio, video, and graphics applications. Initial algorithms implemented under this program include filtering, MP3 encoding and decoding, and an H.263 video codec. Intel is also developing a performance-control library to exploit the XScale's dynamic voltage and frequency scaling. The XScale core contains performance-monitoring hardware, including counters and timers to measure performance-related characteristics such as cache-stall cycles, bus latency, and idle cycles. In all, the performance-monitoring hardware keeps tabs on 22 internal parameters and 8 external ones. An operating system or even application code can use these performance-measurement tools to dynamically adjust processor performance so that no energy is wasted executing tasks. IPP for the IA-32 and IA-64 architectures will be available first. The first version of IPP for XScale will be available from Intel late this year. Beta versions of IPP for StrongARM are available now. There will be no charge for this software.

Intel still has not announced any complete chips based on XScale. The IDF demo silicon was just the processor core—at least according to Intel. In the IDF demo, the core voltage was generated by an external programmable power supply instead of by an on-board or on-chip power DAC. So it's still not clear when Intel will ship XScale silicon, although the company has said it will launch products later this year. It is clear, however, that XScale is a truly formidable embedded processor. Alone, XScale covers *Microprocessor Report*'s entire embedded ChartWatch, from the lowest-power 32-bit embedded processor (at 10mW) to the highest-performance one (at 1,000 Dhrystone 2.1 mips). Meanwhile, others who aspire to the price/performance plateau created by the first StrongARM processors are eagerly trying to lock in as many design wins as possible before Intel ships XScale silicon. The real winners are the systems design teams that benefit from this intense competition for high performance at low power. Palm, one of the leading users of low-power 32-bit processors, announced months ago that it plans to migrate its PDA designs from Motorola's Dragonball to the StrongARM-2, now Intel's XScale. Because of XScale and the competitors sure to follow, it's unlikely that portable-equipment developers need ever again settle for a 4-mips processor just so their new widget can run off a pair of AAA batteries.

Price & Availability

Intel has not yet released price and availability information of processors based on the XScale microarchitecture.