Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

Intel's Tiny Atom

New Low-Power Microarchitecture Rejuvenates the Embedded x86

By Tom R. Halfhill {4/7/08-01}

For 15 years, Intel has watched ARM dominate wireless communications and embedded computing by licensing low-power 32-bit processor cores. Now, finally, Intel is striking back. Last week, at the Intel Developers Forum in Shanghai, Intel officially announced its new Atom family of low-power x86 microprocessors, formerly known under the code names Silverthorne and Diamondville.

Although Atom still uses too much power for most traditional embedded systems, by x86 standards it's a power-performance landmark. At launch, Atom's clock frequency will range from 800MHz (at the minimum core voltage of 0.75V) to 1.86GHz (at the maximum core voltage of 1.2V). Yet thermal design power (TDP) is a mere 0.65W–2.4W over that range. TDP is a worst-case metric, so typical workloads will draw much less wattage. Intel estimates the "average" power at 160–220mW and idle power at 80–100mW.

In contrast, Intel's Celeron mobile-PC and embedded processors have TDPs around 25W for the standard-voltage parts and 8.0W–12W for ultralow-voltage (ULV) parts. Even the new Isaiah microarchitecture from VIA Technologies—formerly the low-power x86 leader—can't match Atom's TDPs. Atom completely redefines the low-power x86 landscape.

It's also a major departure for Intel—or perhaps, a return to its roots. When Intel introduced the first x86 microprocessor in 1978, PCs were still the playtoys of hobbyists. The vast majority of Intel's 8086 chips found their way into embedded systems, mainly industrial. Not until 1981, when IBM chose the 8088 processor for the first IBM PC, did Intel gain a significant foothold in the fast-growing PC market. Attracted by the high profitability of PC processors, Intel has focused most of its energies on that market ever since. Embedded x86 became an afterthought. Indeed, virtually all of Intel's "embedded" x86 chips since the 1980s have been hand-me-down PC processors demoted from the starting team.

Meanwhile, the embedded market has radically changed since the 1970s. No longer are industrial and military systems the major applications. Instead, consumer-electronics products are devouring embedded processors by the billions. Even pocket-sized devices like cellphones, digital cameras, and MP3 players contain two or three processors. A luxury car might have 50 or more. Almost every year, new product categories for embedded processors emerge. (*Microprocessor Report* considers any microprocessor not designed for desktop PCs, servers, or conventional laptop PCs to be an embedded processor.)

Since the 1990s, ARM and numerous other companies have dominated these markets with small, low-power 32-bit processors and licensable processor cores. Last year, ARM shipped its ten-billionth core. For the most part during this revolution, Intel x86 processors have been a no-show.

New Systems Emphasize Mobility

Now, with Atom, Intel is rejuvenating the x86 as a low-power processor suitable for some embedded applications. Atom is a clean-sheet microarchitecture sharing little in common with Intel's PC processors—except full-featured x86 compatibility. That legacy is crucial. Although Atom is Intel's simplest x86 design since the original Pentium in 1993, it's not crippled. It's a 64-bit microarchitecture supporting the

Feature	Intel Atom Z500	Intel Atom Z510	Intel Atom Z520	Intel Atom Z530	Intel Atom Z540
Core Freq	800MHz	1.10GHz	1.33GHz	1.60GHz	1.86GHz
FSB Freq	400MHz	400MHz	533MHz	533MHz	533MHz
Hyper-Threading	—	—	2 threads	2 threads	2 threads
L2 Cache	512K	512K	512K	512K	512K
TDP	650mW	2.0W	2.0W	2.0W	2.4W
Avg Power	160mW	220mW	220mW	220mW	220mW
Idle Power (C6)	80mW	100mW	100mW	100mW	100mW
Die Size	7.8mm x 3.1mm	7.8mm x 3.1mm	7.8mm x 3.1mm	7.8mm x 3.1mm	7.8mm x 3.1mm
Package	mFCBGA	mFCBGA	mFCBGA	mFCBGA	mFCBGA
Price*	$45	$45	$65	$95	$160

Table 1. Intel's Atom lineup at launch. All five parts are based on the same die, sorted by clock speed. Hyper-Threading is disabled on the slowest parts, which saves a little power. TDP is thermal design power, a worst-case specification for system designers who must anticipate cooling requirements. Intel measured "average power" while running the BAPCo MobileMark 2005 Office Productivity Suite on Windows XP for 90 minutes at 50°C. Intel measured "idle power" in the deep power-down C6 state, the processor's lowest power state. Note that idle power for the Z500 is 20mW lower than for the other parts; one reason may be that Intel tested the Z500 while running Linux, whereas the others were running Windows Vista. *Price includes the "Poulsbo" system-controller chip. Intel is shipping some parts now and says the others will ship in coming months.

latest x86 virtualization extensions, Supplemental SSE3 (SSSE3) media instructions, and chip-level multithreading. To the outside world of software, Atom mimics a Core 2 Duo.

Inevitably, Intel had to trade some performance to reduce power consumption and cut costs. This report will delve deeply into those trade-offs. But despite the compromises, Atom is still capable of running a heavyweight operating system like Windows Vista, and it does even better with a leaner OS such as Linux.

It's no surprise that Intel is pushing the x86 further toward the embedded market at this moment. Embedded processors outsell PC processors by about 50 to 1, and the embedded market is a fertile breeding ground for new applications. More importantly, consumer-electronics devices are absorbing functions that were once the exclusive domain of desktop and laptop PCs, which are Intel's bread and butter. Email, text messaging, appointment scheduling, contact management, web browsing, music, video, telephony, gaming—all these applications and more are migrating onto highly mobile devices. Today's desktop PCs are becoming tomorrow's dinosaur mainframes. The real "personal computers" will be those that people carry. To keep up, the x86 must make the leap into pockets and purses.

At first, Atom will target low-cost subnotebook computers, low-cost desktop PCs, and mobile Internet devices (MIDs). Intel refers to the subnotebooks as "netbooks" to emphasize their integrated wireless Internet connectivity. Linux may be more common than Windows on these systems, and some will cost less than $300. Similar subnotebooks based on other processors began appearing last year. Examples are the Asus Eee PC (which uses an Intel Celeron processor) and the Everex CloudBook (which has a VIA Centaur C7-M processor).

Likewise, Intel refers to the mini desktop PCs as "nettops." They will cost about the same as Atom-based netbooks and are reminiscent of the "network computers" unsuccessfully promoted by Sun Microsystems ten years ago. Nettops will be capable of operating offline, like conventional PCs, but they are primarily intended for an environment of ubiquitous Internet connectivity. Their low prices will open new markets. In developing nations, they will attract first-time buyers, and in mature markets, they may supplement conventional PCs. Netbooks, nettops, and universal networking mesh perfectly with Intel's big push for WiMAX wireless Internet.

MIDs Are a Bigger Gamble

MIDs are a greater departure from today's systems than netbooks and nettops are. Intel envisions MIDs as lively handheld computers with wireless Internet connectivity but without conventional keyboards. They will be small enough to carry everywhere, yet they will be capable of delivering an Internet experience second only to PCs. Over time, Intel foresees MIDs becoming handier and more versatile. Although similar territory has been unsuccessfully explored by a myriad of personal digital assistants (PDAs) and other hand-held computers, ubiquitous Internet access could make the difference this time.

MIDs, netbooks, and nettops are merely the start. First-generation Atom processors will lead to even lower-power parts that are suitable for smaller consumer-oriented embedded systems, such as smartphones. Eventually, MIDs and smartphones may converge into a single device, which would make them much more compelling. That convergence also would put Intel on a collision course. As Atom processors descend the power-consumption ladder, they will collide with embedded processors from ARM, MIPS, and other companies that are climbing the performance ladder.

Intel is shipping the first Atom processors and system-controller chips to OEM customers now. Silverthorne CPUs are shipping first, to be followed by Diamondville CPUs later this year. Essentially, these chips are the same die in different packages. Silverthorne is a micro flip-chip ball-grid array (mFCBGA) and Diamondville is a more conventional BGA.

As Table 1 shows, Silverthorne prices will range from $45 for 800MHz parts to $160 for 1.86GHz parts. Prices include a new system controller, which integrates the north-bridge and south-bridge functions in a single 22mm x 22mm chip. Code-named Poulsbo, this chip is officially branded the Intel System Controller Hub for Atom. (See the sidebar, "Atom's System Controller Slashes Power, Too.") Intel will announce Diamondville prices and clock frequencies later.

Silverthorne-based MIDs and netbooks will likely appear this summer. Their processor technology will be branded as "Intel Atom" or "Intel Centrino Atom," depending on their platform specifications. Diamondville-based netbooks and nettops are scheduled to appear later this year, based on another new system controller that Intel may announce in June. Smaller, better MIDs will follow in 2009 or 2010 when second-generation Atom chips and system controllers are ready.

Management's Mandate: x86 Forever

A new design team at Intel's Texas Development Center in Austin launched the Silverthorne project in 2004. That same year, a different team began working on the Poulsbo system controller. Although the Texas Development Center was led by Elinora Yoeli, who came from the Israel Design Center in Haifa, Intel says the Silverthorne team was largely populated with U.S. engineers and wasn't directly connected with the Israeli team that designed another low-power x86 microarchitecture earlier in this decade. In the early 2000s, an Israeli team led by Mooly Eden designed the Pentium M mobile PC processor, code-named Banias.

Banias stunted Transmeta's ambitions to capture a profitable share of the notebook PC market. (See *MPR 11/25/02-01*, "Intel Spills the Beans About Banias.") More important, the Banias core and a later version, code-named Dothan, replaced Intel's power-hungry Netburst core in desktop and server processors. Dothan enabled Intel to introduce its first multicore x86 processors in step with AMD, which had enjoyed a head start. (See *MPR 5/31/04-02*, "Intel's PC Roadmap Sees Double.") Without Banias and Dothan, Intel would have lagged years behind AMD and lost much more market share. Atom has similar potential to lift Intel's fortunes, because it carries Intel much further into the low-power realm.

When Intel launched the Silverthorne project, principal architect Belli Kuttanna received three mandates from upper management. First, the new microprocessor had to dramatically reduce power consumption, to about 10% of a ULV Dothan core. Second, the processor had to retain enough performance to give users a full Internet experience, meaning desktop-caliber web browsers and plug-ins running on a sophisticated OS. Third, the processor had to be fully x86 compatible, supporting the latest x86 extensions. Even a partially compatible x86 design was unacceptable, much less a completely new CPU architecture.

The first two requirements were aggressive but understandable. The third requirement is the most interesting, because it ruled out any option of saving power by overhauling the complicated x86 architecture at the expense of software compatibility. Intel could have followed Motorola's example with the 68000 (68K) architecture, which first appeared in 1979, a year after the x86. In 1994, after newer RISC architectures supplanted 68K processors in workstations and servers, Motorola streamlined the 68K for greater efficiency in embedded systems, producing the derivative ColdFire architecture. (See *MPR 10/24/94-05*, "Motorola Redefines 68K Instruction Set.") ColdFire inherited the best features of the 68K but isn't fully 68K compatible. It's debatable whether this compromise contributed to the subsequent decline of 68K/ColdFire. In any case, Intel didn't want to take the same chance.

Intel's decision eliminated many tempting opportunities to improve power-performance efficiency by discarding unneeded baggage in the 30-year-old x86 architecture. As a result, Atom's irreducible atomic weight tilts the scale more toward uranium than toward hydrogen. Long-suffering engineers praying for relief from the x86's notorious complexity will be disappointed. However, Intel's decision allows Atom-based systems to offer 100% software compatibility with the latest Windows PCs, Apple Macintoshes, and x86-based Linux systems. Potentially, the world's largest base of desktop software can go ultramobile. In particular, existing web browsers, browser plug-ins, and Java virtual machines can run without modification on Atom processors.

Software compatibility will be Intel's loudest marketing pitch against ARM. Although ARM's efficient 32-bit RISC architecture is widespread in the embedded world, it doesn't have a body of application software in the same class as the software available for the x86. Indeed, only the latest, fastest ARM processors—like the Cortex-A8 and Cortex-A9—have the performance required to run that kind of software, so relatively little exists. Intel's x86-forever strategy almost certainly cedes a low-power advantage to ARM and other RISC architectures, but it preserves the essential element that Intel believes Atom needs to succeed.

Alternative Approaches to Saving Power

Committed to full x86 compatibility, the Atom design team resorted to numerous other means of saving power while retaining adequate performance. Almost immediately, the engineers discarded the idea of modifying an existing x86 microarchitecture to derive a lower-power design. The problem was that all of Intel's existing x86 processors—even Banias and Dothan—were optimized primarily for high throughput, with low power a secondary goal. Atom had to be optimized primarily for low power; adequate throughput was the secondary goal. Reversing priorities affected every part of the design process.

Starting with a clean slate, the Atom team reverted to a very basic microarchitecture with a single-issue in-order instruction pipeline. Not since 1989, when Intel introduced the 486, has an Intel x86 processor relied on such a simple

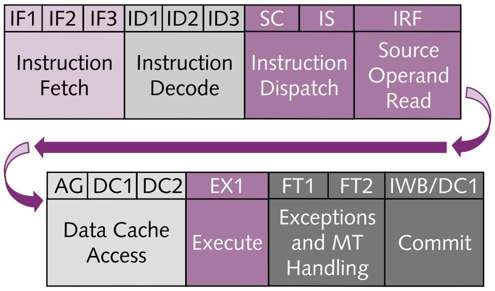


Figure 1. Atom pipeline diagram. This 16-stage integer pipeline is duplicated for two-way superscalar execution. It's deep enough to be a superpipeline but is still much shorter than the 30-plus stages of the power-hungry Hyper-Pipeline in the Pentium 4. The branch-misprediction penalty is 13 cycles. Note that some rare or complex x86 instructions will detour into a microcode sequencer for decoding, necessitating two additional clock cycles.

pipeline. Unfortunately, it wasn't good enough. The 486 was a top performer in its day, but those were the days when Microsoft shipped Windows on a few 5.25-inch floppy disks. Vista and even Linux are quite a bit heavier. So the Atom designers began adding stuff back. Whereas the 486 had a simple five-stage pipeline, Atom has a much deeper 16-stage pipeline capable of reaching much higher clock speeds. And there's a second integer pipe, allowing two-way superscalar execution.

Another modern enhancement, out-of-order execution, was evaluated and rejected. The processor would have needed too much control logic to shuffle the instructions, execute them out of program order, and restore them to their original order before retiring the results. The extra transistors required for that logic would worsen both active and passive power. Likewise, the designers rejected aggressive speculative execution. Atom does predict branches, so it may speculatively execute a few instructions beyond a predicted branch, but it discards the more-aggressive speculation of most other x86 processors. The drawback of speculation is that sometimes the program won't take a predicted branch, forcing the processor to discard the speculative results and waste all the power spent on them.

Still another advanced feature that Atom discards—in most cases, at least—is x86 instruction transformation. All Intel x86 microprocessors since the Pentium Pro in 1995 have decoded standard x86 CISC instructions into RISC-like "micro-ops." So do the x86 processors from AMD and VIA. It's easier for the CPU to handle these transformed operations, especially when dispatching multiple operations through superscalar pipelines. Usually, the processor divides long x86 instructions into two or three micro-ops. Simpler x86 instructions may undergo little or no transformation. Still others are so complex that they detour through a microcode ROM for special firmware decoding, instead of normal hardware decoding. (See *MPR 2/16/95-02*, "Intel's P6 Uses Decoupled Superscalar Design.")

Surprisingly, Atom reverts to handling almost all x86 instructions in their original form. The design team concluded that transforming the instructions into micro-ops used too much power. A key factor in this evaluation was Atom's relatively simple pipelines. Shorn of wide superscalar execution, out-of-order instruction processing, and aggressive speculation, Atom can get by with digesting x86 instructions whole. So did the original Pentium of 1993, which also had a dual-issue in-order pipeline without speculation.

Cute x86 Tricks

Nevertheless, Atom can play some tricks with x86 instructions. Wickedly complex instructions for which the x86 is infamous are diverted into a microcode sequencer for special decoding. They emerge as born-again RISC-like operations, cleansed of their CISC sins. This blessing requires two additional clock cycles.

In a few other cases, x86 instructions combining multiple operations—such as an ALU/memory instruction—are issued in pairs through the dual pipelines, as if they were separate micro-ops. They execute simultaneously, in lockstep. They can't bypass each other, as true micro-ops can do in other x86 processors. However, the paired operations occupy only one slot in the 32-entry instruction-dispatch table, because they're still a single x86 instruction.

Atom has two instruction decoders. Each can decode complex types of x86 instructions that other Intel x86 processors would transform into three micro-ops. To save a little power, Atom predecodes x86 instructions by tagging them with a one-bit marker indicating the end of each instruction. (Unlike RISC architectures, which typically have fixed-length 32-bit instructions, the x86 has variable-length instructions ranging in size from 8 bits to 120 bits. Finding the boundaries in an instruction stream is half the battle.)

In two predecode stages of the pipeline, Atom marks the instruction boundaries and stores the instructions in the L1 cache. Afterward, instructions fetched from the cache are already marked and can bypass those two stages. The instruction cache is 36KB, but the effective size is about 32KB, because the boundary tags occupy about 4KB. Except for this tagging, cached instructions are undecoded—they must pass through three additional pipe stages for full decoding.

When Intel says Atom has a 16-stage basic integer pipeline, note that the two predecode stages are excluded from the count (and from Intel's official pipeline diagram, which *MPR* has adapted in Figure 1). In practice, the pipeline varies from 16 stages to 19 stages, depending on circumstances. Sometimes, a cache miss forces the processor to spend three clock cycles finding the end of an instruction whose boundary hasn't been tagged yet. In some of those cases, the processor can simultaneously decode a preceding instruction without a penalty, preserving the nominal 16-stage pipeline. And in any case, after a cache hit, predecoded instructions pay no penalty. For most purposes, it's a 16-stage pipeline.

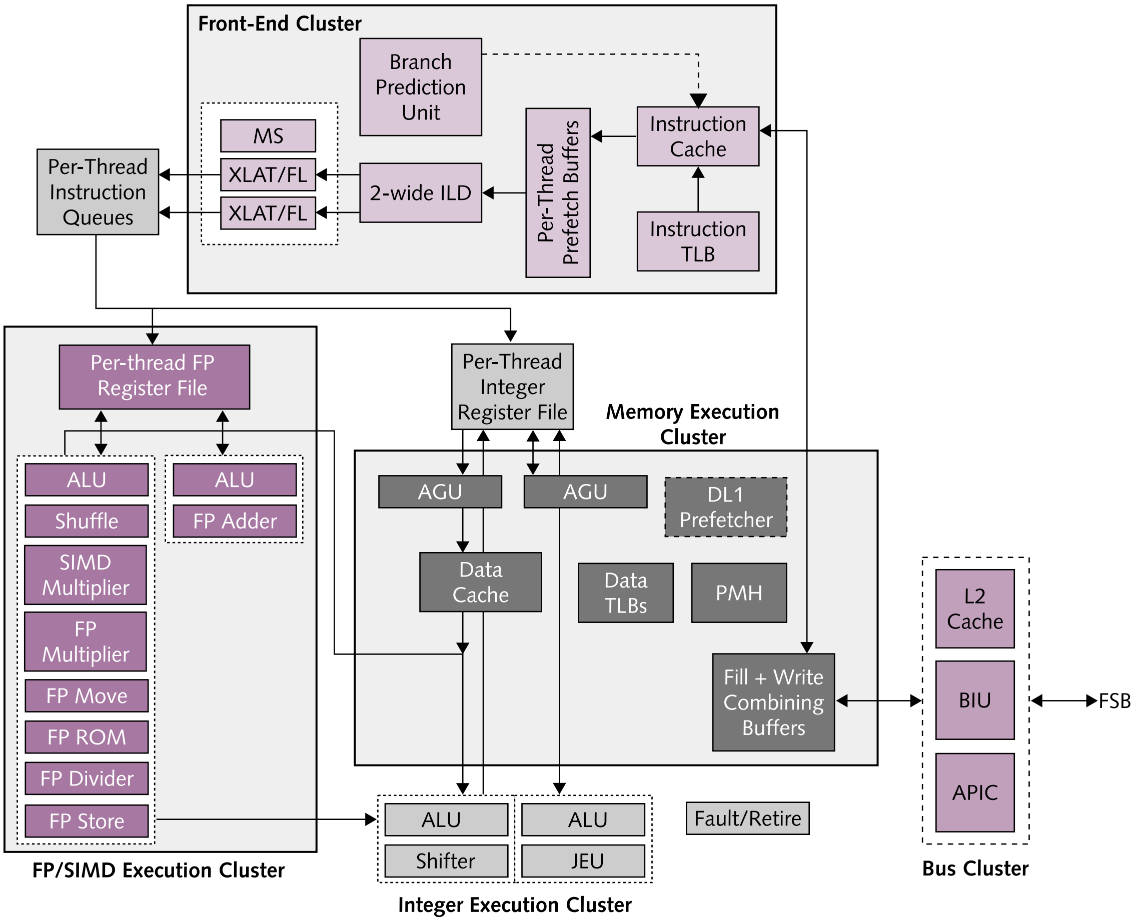


Figure 2. Atom block diagram. Although the two-way superscalar Atom processor is Intel's simplest x86 design since the original Pentium in 1993, it's definitely not your father's Pentium. Most of this design would be familiar to CPU architects of the early 1990s—for example, the dual ALUs, dual FPUs, dual address-generation units (AGUs), branch predictor, caches, and TLBs. But additional resources for chip-level multithreading would be unfamiliar to most architects of that era. Atom has duplicate copies of the integer and floating-point register files, instruction queues, and prefetch buffers. These resources allow the processor to maintain state information for two different software processes at the same time, and to switch contexts in a single clock cycle.

Atom's decoding hardware is one example of the tradeoff between power efficiency and software compatibility. In some cases, Atom spends five clock cycles decoding x86 instructions, spreading the complex decoding logic across five pipeline stages to avoid bogging down any particular stage with too many gate delays. If Intel had simplified the x86 architecture for the embedded world, decoding would have required less logic. But modifying the architecture would have compromised compatibility with existing x86 software, so Intel decided to suffer the power penalty instead. In the overall scheme of things, the penalty probably isn't severe, but it does give rival RISC architectures a potential advantage.

Multithreading Makes the Cut

Another surprise, in some respects, is that Atom supports Hyper-Threading, Intel's marketing name for simultaneous multithreading (SMT), also known as chip multithreading (CMT). This technique allows a processor to mix instructions from two different threads of execution in the same pipeline at the same time. The instructions may originate from two lightweight threads within a single program or from two completely different programs. Context switching can happen instantly, stage by stage. The processor maintains duplicate register files and other resources for storing the state of each thread, so a context switch doesn't require dumping and restoring the registers, status flags, stack pointers, and so forth. (See *MPR 9/17/01-01*, "Intel Embraces Multithreading.")

Other companies pioneered SMT, and Intel introduced its first dual-threaded implementation with the Pentium 4 processor in 2003. Since then, however, Intel has omitted Hyper-Threading from several of its processors. SMT requires additional control logic plus an extra register file for each thread. More logic and more registers translate into more transistors, which occupy more silicon and use more power.

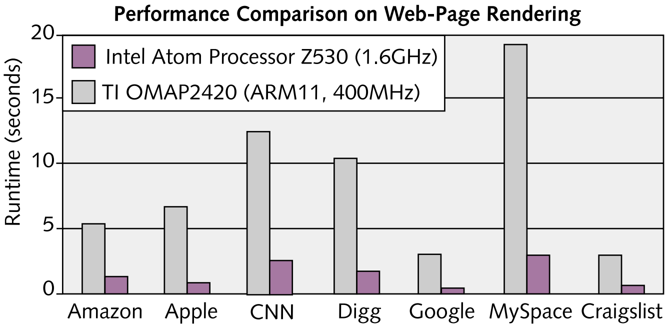


Figure 3. Intel's page-rendering benchmark tests for seven popular websites. These benchmarks compare a 1.6GHz Atom processor with a Texas Instruments OMAP2420 chip, which has a 400MHz ARM11 processor core. (Lower bars are better. To eliminate network latencies, both systems accessed all web pages on a solid-state flash drive.) Although Atom wins easily, the ARM11 doesn't fare too badly, considering its numerous handicaps. In clock frequency alone, Atom has a 4x advantage. Besides that, Atom is a 64-bit superscalar multithreaded processor, whereas the ARM11 is a 32-bit uniscalar single-threaded processor. Also, Atom is an optimized microprocessor chip, whereas the ARM11 is a fully synthesizable processor core. The ARM11's biggest advantage over Atom is lower power consumption. Even when an ARM11 is fabricated in a speed-optimized 90nm bulk CMOS process, it will use less power than an Atom processor fabricated in Intel's two-generations-better 45nm process with metal-gate transistors and high-*k* dielectrics.

The reappearance of Hyper-Threading in a low-power design like Atom is surprising at first, but the decision makes sense. Intel's tests revealed that dual threading boosts Atom's performance by 36% to 47% across several popular benchmarks. (Although Intel isn't releasing specific benchmark results at this time, *MPR* viewed the scores under a nondisclosure agreement, and they include SPEC benchmarks as well as EEMBC's embedded benchmark suites.) In return for this impressive improvement in throughput, Hyper-Threading worsens power consumption by 17% to 19% across the same benchmarks. Overall, it's a worthwhile trade-off. For customers who don't like the bargain, Intel offers some Atom processors with Hyper-Threading disabled.

Hyper-Threading's effect on Atom's die area was relatively minor—it enlarges the logic portion of the design by about 8%. There's an additional but less visible price on cache utilization and other structures. Instructions from two different threads must occupy the same instruction cache, which doesn't grow any larger, so cache misses on a particular thread are more likely than if one thread monopolizes the whole cache. Negative effects on the 24KB L1 data cache and the 512KB L2 cache are similar.

Both threads must share the same 32-entry instruction-dispatch table, so each thread gets only 16 entries. (The scheduler can pick two instructions from either thread per clock cycle.) Likewise, both threads share the same 64-entry translation lookaside buffer (TLB), which is two-way set-associative. A supplemental micro-TLB has 16 entries for each thread. To reduce other resource contentions during multithreading, Intel doubled the size of the instruction cache's prefetch buffer, so each thread has three 16-byte buffers. Several of these structures are visible in Figure 2, a block diagram of the Atom processor.

The Atom designers didn't feel compelled to enlarge every structure related to Hyper-Threading. For one thing, they didn't want to pay the power and die-area penalties. Another reason is that multithreading allows the processor to tolerate a little more latency during instruction execution, so some resource contentions aren't as bad. All things considered, the improved throughput from Hyper-Threading is significant and requires less logic and power than other tactics, such as adding a third superscalar pipeline or enabling out-of-order execution. *MPR* editorial-board member Don Alpert, a former Intel x86 architect, anticipated these trade-offs in a 2003 article. (See *MPR 11/17/03-03*, "Will Microprocessors Become Simpler?")

Different Designs for Different Goals

VIA takes a very different approach with its new Isaiah low-power x86 microarchitecture. (See *MPR 3/10/08-01*, "VIA's Speedy Isaiah.") Isaiah implements three-way superscalar execution, out-of-order instruction processing, and speculation, but omits multithreading. In almost every respect, the Atom and Isaiah engineers made opposite decisions and trade-offs.

Interestingly, the Atom and Isaiah design teams—which work only a few miles apart in Austin—offer the same explanations for these differences. Consider multithreading. A simpler in-order machine like Atom can derive more benefit from threading than a more complex out-of-order machine like Isaiah can. When a multithreaded in-order pipeline stalls, it can quickly switch threads to continue working. When a single-threaded out-of-order pipeline stalls, it can speculatively execute other instructions to continue working. With Atom, Intel needed to significantly reduce power consumption. With Isaiah, VIA needed to significantly improve performance. Engineering is all about making trade-offs.

Another example is the FPUs in Atom and Isaiah. For years, critics have bashed VIA's Centaur processors for their laggard floating-point and multimedia performance. Isaiah silences those critics by incorporating two very fast FPU/SSE units. Their 128-bit-wide datapaths can handle double-precision floating-point operations and SIMD media operations with aplomb. Meanwhile, across town, the Atom engineers were whacking at their FPUs with a machete. Although Atom has two FPUs, as Isaiah does, the SIMD floating-point adder has narrower 32-bit datapaths, so double-precision operations must be double-pumped. (For other SIMD operations, Atom has 128-bit-wide datapaths, like Isaiah's, so it can sustain two SSE operations per clock cycle.) Again, Intel needed to slash power consumption, whereas VIA needed to boost throughput. Both teams achieved their goals.

VIA's Isaiah and Centaur C7 processors have security features not found in Atom processors, as far as Intel has publicly disclosed. VIA's processors have true random-number generators and hard-wired acceleration logic for cryptography. These features are critical for some of VIA's customers. *MPR* believes all microprocessors will eventually incorporate similar features.

One trade-off that didn't make sense, according to Intel, was adding enough resources so that Atom could execute more than two chip-level threads at the same time. Indeed, no Intel processor has ever ventured beyond a dual-thread implementation of Hyper-Threading. The Atom design team explored the possibility of a three- or four-thread implementation, but concluded that the diminishing returns weren't worth the costs. Other CPU architects have reached different conclusions, though certainly under different circumstances. The MIPS32 34K embedded-processor core from MIPS Technologies supports up to five threads, and Ubicom's tiny IP3023 embedded processor handles eight threads. These processors were designed for more-specific embedded applications. (See *MPR 2/27/06-01*, "MIPS Threads the Needle," and *MPR 4/21/03-01*, "Ubicom's New NPU Stays Small.")

It's important to note that Hyper-Threading doesn't jeopardize Atom's x86 software compatibility. A popular misconception is that SMT forces developers to write multithreaded programs. Although multithreaded code can certainly benefit from SMT, chip-level threads can be instruction streams from two entirely different processes, which may be single threaded.

As an example, Intel says that Atom can render web pages 15% to 30% faster with Hyper-Threading enabled, even when running single-threaded web browsers. Part of this improvement may come from a multithreaded processor's ability to juggle instructions from the application program and the OS at the same time. Another possibility is that browser plug-ins like Adobe's Flash animator can execute as a separate thread, alongside the browser's usual page-rendering code. Figure 3 shows the results of Intel's page-rendering benchmarks with Hyper-Threading enabled on an Atom Z530 processor running at 1.6GHz. Intel ran the same benchmark tests on an ARM11-based Texas Instruments OMAP2420 processor—which, in most respects, was outclassed in this comparison. (See *MPR 3/22/04-01*, "Dial the Future.")

Additional Power-Saving Features

The Atom design team discovered no miracle cures for high power consumption. Beyond the approaches already described—mostly, trading away some throughput—the designers saved small amounts of power in numerous ways. Some of these techniques are widely used throughout the industry, while others are innovative and clever. Added together, lots of little power savings become significant.

One option the Atom designers ruled out (or that management ruled out for them) was tweaking the chip-fabrication process. Intel is making Atom processors in exactly the same 45nm process used for the highest-performance Itanium and x86 PC and server processors. In theory, Intel could run wafers mixing Atom dies with those of any other Intel processors. Even for a power-optimized processor like Atom, Intel doesn't want to veer from its famous "copy exactly" manufacturing philosophy. A fabrication process that proves successful at a pilot fab is replicated without change at every other Intel fab running the same process, anywhere in the world. Rigorous duplication ensures higher yields and faster ramp-ups to high-yield mass production.

However, adhering to the same design rules doesn't rule out using circuit-level and even transistor-level optimizations. Today's automated design tools allow engineers to choose lower-threshold, faster-switching transistors for critical signal paths, or higher-threshold, slower transistors for less-critical circuits. The most important circuits may be manually improved or laid out from scratch, even when using automated tools. Dynamic logic is avoided outside arrays. In wide datapaths, such as the 128-bit integer SIMD pipelines, wide fanouts between stages can quickly inflate the number of flip-flops. Manually optimizing those pipe stages can eliminate thousands of transistors. Engineers on the Atom team used all these techniques.

Nevertheless, *MPR* was surprised to learn that 91% of Atom's CPU core is based on standard cells, albeit with some manual tweaks. Intel says the only full-custom blocks in the CPU core are the on-chip thermal sensor and the microsequencer engine that decodes rare or complex x86 instructions. (Outside the CPU core, there are several full-custom blocks in the L2 cache, PLL, I/O logic, and fuses, among other things.) Although standard-cell synthesis is commonplace today, it's unusual for Intel to design an x86 processor in this manner—especially a new breed of x86 processor that must meet stringent performance requirements. Urgency to get the design out the door was undoubtedly one factor, but the Atom project is also a testament to modern design tools and Intel's expertise.

Price & Availability

Intel is shipping Atom ("Silverthorne") processors and "Poulsbo" system-controller hubs (the Atom Centrino platform) to customers in 2Q08. The first parts are shipping now, with others to follow in coming months. Speed grades for the first Atom processors range from 800MHz to 1.86GHz, and OEM prices range from $45 to $160 in 1,000-unit quantities. Prices include the system controller. OEMs expect to ship the first systems with Atom processors this summer. "Diamondville" Atom processors are scheduled to ship later this year, along with another new system controller. For more information, visit www.intel.com/technology/atom/.

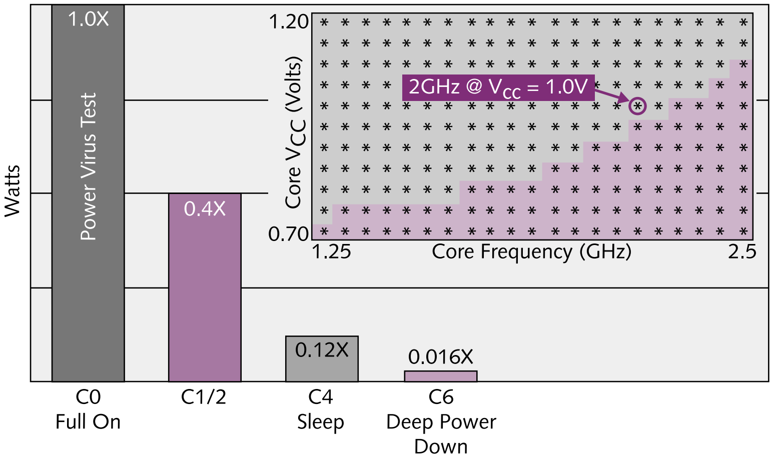


Figure 4. Shmoo plot of Atom's voltage-frequency curve in two low-power states. In reduced-power C1 and C2 states, after lowering the core voltage to 1.0V (V_{cc}), Atom can sustain a clock rate of 2.0GHz, which is 80% of its maximum target clock frequency. In the new C6 mode, Atom can enter a deep power-down sleep state that stops the clocks altogether. The "power virus test" referenced in the bar chart is an Intel program that pushes the processor to its worst-case thermal design power (TDP).

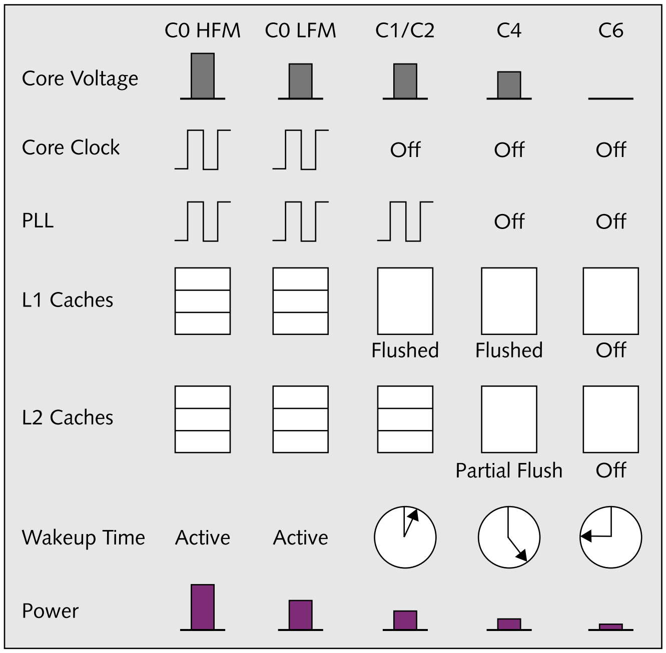


Figure 5. Atom's power states. C0 is the most active state, and it has high-frequency and low-frequency modes, using Intel's Enhanced SpeedStep technology. The C1 and C2 states are the least disruptive sleep modes with the shortest wakeup times. C4 state is a deeper slumber, and the new C6 state is the deepest power-down mode that allows the processor to maintain the critical software state of a running process.

The Atom microarchitecture is modular, too. Starting with the same die, Intel can short out built-in fuses to disable various functional blocks, then sell several different parts with different features and specifications. Among other things, Intel can disable Atom's 64-bit x86 extensions, virtualization extensions, and Hyper-Threading. For example, disabling the 64-bit extensions cuts power by about 10%.

Even the I/O transceivers on the front-side bus (FSB) are fused. Other Intel x86 processors use Advanced Gunning Transceiver Logic (A/GTL+) on the FSB. Normally, Atom uses CMOS transceiver logic instead. However, Intel built both types of transceivers into Atom processors and the Poulsbo system-controller chip. The transceivers are fused, so Intel can offer the chips either way. In CMOS mode, Atom's FSB can run at effective clock rates of 400MHz to 533MHz. (The bus is quad-pumped, so the base clock rates are 100MHz or 133MHz.) CMOS transceiver logic has longer latencies but draws power only during voltage transitions, not while idle. The power savings depend on the amount of bus traffic. Intel estimates that CMOS transceivers save a combined 200mW to 500mW in the processor and system controller when the FSB runs at an effective frequency of 400MHz.

If a customer prefers A/GTL+, Intel can disable the CMOS transceivers. One reason for preferring A/GTL+ is compatibility with other chipsets besides Poulsbo. Another reason may be performance. Intel admits including both types of transceivers in the Atom design because it wasn't clear at first if CMOS could reach the necessary bus speeds. Atom and Poulsbo have been successfully tested at effective bus speeds up to 533MHz. Intel hasn't publicly disclosed if the CMOS transceiver can reach higher speeds, but *MPR* suspects that A/GTL+ will be necessary to reach 667MHz and higher effective bus speeds.

Multiple Sleep Modes Save Power

Other power-conservation features are programmable, not fused, so the processor can dynamically adapt to operating conditions. For instance, the number of outstanding I/O requests buffered on the FSB is programmable. Parts of the L2 cache can be shut down to reduce both active power and current leakage. And, of course, Atom has Enhanced SpeedStep technology, Intel's version of dynamic voltage/frequency scaling. It works even better on the 45nm metal-gate process, which permits very low operating voltages. *MPR* saw production silicon running in the lab at 0.875V, and the devices were drawing less than 1.0W while enduring the torture of running Windows Vista.

The L1 instruction and data caches are somewhat unorthodox. They resemble register files built with eight-transistor bit cells, and they have only one read port and one write port. The caches are larger than small-signal memory arrays built with six-transistor bit cells, but they operate at a lower voltage and use less dynamic power. The 512KB L2 cache uses standard 6T cells. It transfers 256 bits per transaction and wards off soft errors with inline ECC protection (single-error correction, double-error detection). Set-associativity is programmable, from two ways to eight ways, affording another way to save a little power.

Atom supports multiple x86 power states, from full-power C0 state to the new C6 deep-sleep state. If C0 represents 100% power for an Atom device, then C1 is 40%, C4 is 12%, and C6 is 1.6%. (See Figure 4.) C6 voltage is determined by the amount of core leakage the design can tolerate and by how quickly the voltage regulator can restore the core power supply when the processor wakes up. When entering C6 mode, the processor saves its state information (registers, status flags, stack pointers, program counters, etc.) in a small amount of internal memory, then stops the clocks and goes to sleep. Atom also shuts down the FSB, which saves even more power, because the I/O voltage (1.05V) may be higher than the core voltage. To wake up, the processor must restart its clocks, restore the state information, and reprime the pipelines. To conserve energy, Atom gradually refills the caches on demand, not all at once.

To save the most power, a processor should spend as much time as possible in C6 state, but only if the slumber offsets the surge of energy required to wake up. Fast wakeup is important. If the processor can't return to C0 promptly, the system will seem sluggish and unresponsive. Intel says Atom can almost always wake up in less than 100 microseconds, which is pretty fast. (Wakeup from C4 takes only about 30 microseconds.) Figure 5 compares the various power states that Atom supports. These states include the high-frequency and low-frequency modes in the C0 power state, which correspond to the similar voltage/frequency modes in other x86 processors having Intel's Enhanced SpeedStep technology. (See *MPR 8/6/01-01*, "Intel Debuts 1.13GHz Tualatin.")

Intel estimates that a typical system could keep an Atom processor in C6 state about 80% of the time. If so, the power savings would be significant. Naturally, Intel's estimate depends greatly on the type of system. A mobile Internet-access device could enter C6 state when idle for only a few seconds without noticeably bothering the user with slow wakeup later. On the other hand, 100 microseconds might be too long to wait for an embedded-control system that needs hard real-time availability. Atom is designed primarily for the first type of system.

Entry into C6 state is regulated by the processor, system controller, and OS. Sometimes, the chips may override a software-driven request to enter C6, because OSes and application programs typically have no concept of power efficiency. If the chips deem a request inappropriate, they can decline altogether or demote the request to a higher power state, such as C4, which saves less power but recovers faster.

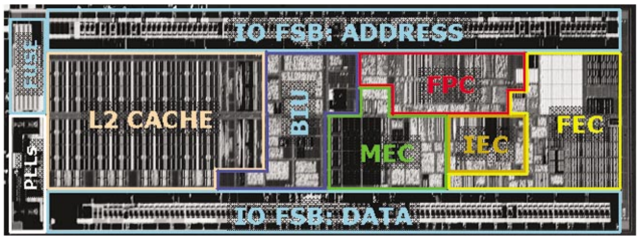


Figure 6. Atom die plot. The die measures only 3.1mm x 7.8mm, amazingly small for a full-featured 64-bit x86 microprocessor. The CPU core has only 13.8 million transistors. The on-chip 512KB L2 cache accounts for 30.6 million transistors (22% of the area), and miscellaneous other blocks contain an additional 2.7 million transistors. Notice that the front-side bus (FSB) data and address interfaces occupy relatively large blocks along two sides of the chip—35% of the die, in all. These I/O structures require analog logic that doesn't shrink as well as digital logic does. Intel will sell this die in a 441-ball micro-FCBGA (flip-chip ball grid array) package, or—as the "Diamondville" variant—in a BGA package. Abbreviation key: BIU (bus interface unit); FPC (floating-point cluster); MEC (memory execution cluster); IEC (integer execution cluster); FEC (front-end cluster, including the instruction decoder, branch predictor, I-cache, prefetch buffers, and instruction TLB).

Using feedback from an integral thermal sensor, Atom can dynamically adjust its core voltage and clock frequency in response to fluctuating chip temperatures and operating conditions. This mechanism is similar to the thermal-management technology in VIA's Isaiah microarchitecture. An emergency protection mechanism automatically slows the processor if heat damage seems imminent. Intel says better thermal-feedback performance management is coming in future versions of the chip.

A Small Chip, Ready to Grow

Much can be learned by studying the Atom die plot, shown in Figure 6. In the first place, Atom is remarkably tiny. Measuring only 3.1mm x 7.8mm (24.2mm^2), it's less than half the size of Isaiah (63mm^2) and is even smaller than VIA's Centaur C7-M (30mm^2), the previous low-power x86 record holder. Remember, Atom is a fully x86-compatible 64-bit processor capable of running a desktop OS. It has 512KB of L2 cache on chip, yet there are only 47.1 million transistors, including caches. (The CPU core has a mere 13.8 million transistors.) Obviously, Intel's leading-edge 45nm process was a major factor in shrinking the silicon, but careful design was a factor, too.

The most striking physical feature of the die, besides its small size, is its unusually wide (2.5:1) aspect ratio. Most microprocessors are laid out nearly square, mainly to minimize signal delays through long wires across the die. Atom looks like it was created as a larger die and then sliced in half. It's obvious that a dual-core version is coming. By replicating the layout and grafting them together, Intel can easily produce a dual-core Atom that will still be one of the smallest x86 processors on the market. (Depending on where the two halves are joined, either the address or data

Decoding the Code Names

Use this handy guide to decipher Intel's cryptic code names and brands. (Listed in approximate order of relevance to this article.)

Atom: brand name for a new low-power x86 microprocessor, formerly code-named Silverthorne, shipping 2Q08. Offered in a micro-FCBGA (flip-chip ball grid array) package. Intended for subnotebook PCs and mobile Internet devices.

Diamondville: code name for an Atom-based microprocessor in a BGA package, scheduled for release later this year. Intended for subnotebook PCs and Intel's Classmate educational PCs.

Poulsbo: code name for a new low-power system controller for the Atom microprocessor, shipping 2Q08. Official name is Intel System Controller Hub for Atom. Integrates the north bridge and south bridge in a single chip. Atom + Poulsbo = Atom Centrino platform.

Menlow: former code name for the low-power x86 platform now branded Atom Centrino.

netbooks: generic name for low-priced subnotebook PCs enabled by the Atom microprocessor. Retail prices may start below $300. May run Linux or Windows. (The Asus Eee PC, which currently has an Intel Celeron processor, is the general model for these systems.)

MIDs: generic name for mobile Internet devices. These will be hand-held computers with wireless network connectivity (WiMAX and 802.11), but without conventional keyboards. May run Linux or Windows. Atom is designed for MIDs.

Moorestown: code name for an improved Atom-based platform scheduled for release in 2009 or early 2010. It will further reduce power consumption and improve support for VoIP telephony. May have integrated memory control and stacked memory. Includes the Langwell, Briertown, and Evans Peak devices.

Langwell: code name for an I/O-hub south bridge, intended for the Moorestown platform. Will have a solid-state disk controller and I/O blocks licensed from third parties.

Briertown: code name for a power-management chip, intended for the Moorestown platform.

Evans Peak: code name for a single package containing two or three chips, including support for Wi-Fi, WiMAX, Bluetooth, GPS, 3G, and mobile TV. Intended for the Moorestown platform.

Tolapai: code name for a semicustom SoC that will integrate an Atom processor core, DSP microengines, north-bridge functions, and some I/O interfaces on a single chip. Intended for MIDs, digital-home products, and general embedded applications. Expected in 2009–2010.

Lincroft: code name for a semicustom SoC that will integrate an Atom processor core, graphics, video, display interfaces, and north-bridge functions on a single chip. Intended for netbooks, MIDs, and digital-home products. Expected in 2009–2010.

Cherrypoint: code name for Intel's next-generation Classmate educational PCs. Will use Atom (Diamondville) processors and introduce mesh networking, as found in the One Laptop Per Child (OLPC) computers.

Canmore: code name for an SoC that integrates an x86 Dothan processor core, 3D graphics, audio, video, north-bridge functions, and some I/O interfaces on a single chip. Intended for "digital home" products. Available now.

portion of the FSB will be relocated to the new edge of the die.) In other words, Atom is ready for fusion.

Another clue of a multicore design is Atom's clock-distribution network. The chip has separate PLLs for the CPU core and I/O blocks, and the clock tree doesn't follow the usual grid layout. Instead, Intel says Atom has a three-way branching clock tree with smaller "twigs" to distribute clock signals to various blocks throughout the chip. Blocks unneeded at the moment can be individually turned off. The clock network is designed to be more power efficient and is flexible enough to accommodate a dual-core implementation.

Does Atom really need two (or more) cores? For MIDs, a multicore chip is probably overkill and could burst the power envelope. In netbooks and nettops, multiple cores might be a godsend, if the OS is Windows Vista. Intel frankly prefers to see Linux running on mobile devices, because the open-source OS is a lighter burden on the CPU and needs less memory. (And it's free, or nearly so.) But if the marketplace demands greater compatibility with Windows PCs, then XP or Vista will become the de facto standard, and system developers will have to please their customers. These systems may need a multicore CPU to deliver the expected performance, even at the cost of reduced battery life.

Comparing Atom's Performance

Evaluating Atom's real-world performance is difficult at this point. Intel has been benchmarking the simulators and early silicon for months but is keeping most results under nondisclosure for now. *MPR* has seen some of these test scores. We can disclose general conclusions, but not detailed numbers. Two of the most interesting comparisons pitted Atom against ARM's Cortex-A8 and one of Intel's own ULV Celeron processors, the 800MHz Dothan-based A110 "Stealy" chip.

For the Cortex-A8 comparison, Intel ran EEMBC's embedded benchmark suites, an appropriate choice. (The scores aren't official, because Intel used a prerelease compiler, and EEMBC certification mandates a publicly available compiler.) In the first phase of these tests, a 1.2GHz Atom with Hyper-Threading disabled ran the EEMBC benchmarks not quite twice as fast as a 600MHz Cortex-A8. That result suggests that Atom's advantage over the Cortex-A8 merely scales with clock frequency. However, with Hyper-Threading enabled, a 1.6GHz Atom ran the benchmarks significantly more than twice as fast as a 1.0GHz Cortex-A8. This result supports Intel's claims that Hyper-Threading boosts performance by 36% to 47%.

Notably, Intel didn't report the results of EEMBC's power benchmarks when running these performance tests. One reason may be that EEMBC's power measurements require actual silicon, and Cortex-A8 silicon is only beginning to appear. Intel probably tested the Cortex-A8 in simulation.

Comparing Atom with Intel's own ULV Celeron A110 processor is more illuminating. This is the microprocessor in Intel's "McCaslin" platform for ultramobile PCs, which Atom could replace. Under simulation, Intel ran a large battery of benchmarks on both processors, including 3DMark05_CPU, SPECint2000, SPECfp2000, SysMark2004, and a variety of games and business software. At 2.0W TDP, Atom delivers about 15% better throughput. At 0.6W TDP—a low-power specification the ULV Celeron can't match—Atom is still 66% to 75% as fast.

Overall, it appears that Intel achieved its goals of significantly reducing power consumption in a fully compatible x86 processor without sacrificing too much throughput. Moreover, Atom is a very small x86 chip that will be inexpensive to manufacture. Future dual-core implementations will improve performance while keeping the die relatively small. Future process shrinks offer opportunities for even lower power and/or greater throughput. By x86 standards, certainly, Atom is a winner.

The larger question is whether x86 compatibility really matters. If it does, Intel has the smallest, lowest-power, cheapest-to-manufacture x86 chip capable of meeting the performance requirements. If x86 compatibility is not vital, then embedded RISC architectures still have an advantage.

Oh, No, Another RISC–CISC War!

Longtime readers of *MPR* no doubt remember the spirited RISC vs. CISC war of the 1980s and 1990s. Conventional wisdom holds that CISC won. The x86 not only defended its PC turf against RISC challengers, but also drove RISC processors out of most workstations and made serious inroads into servers. RISC's Waterloo defeat came in 2005, when Apple abandoned the Power Architecture (PowerPC) for the x86.

However, that's a PC-centric history of the war. In sheer volumes of 32- and 64-bit processors, RISC massacred CISC. For every PC or server processor that Intel sells, ARM's army of licensees sells five or ten ARM-based chips. Adding all the other RISC architectures—ARC, MIPS, SPARC, Tensilica, the Power Architecture, and more—makes the RISC victory look overwhelming. Furthermore, RISC processors rule the fastest-growing, most innovative markets. Mobile phones, iPods, TiVos, videogame consoles, portable video players, digital cameras, digital TVs, set-top boxes, DVD players, VoIP phones, flash-memory drives, memory cards, and numerous other products have RISC controllers. The x86 is found mostly in traditional PCs and servers.

Not that Intel completely surrendered the embedded market. Intel mounted a minor counterattack with XScale, but that was an ARM-based architecture acquired from Digital Semiconductor in a legal settlement. It was rather like hiring enemy mercenaries to fight your battles. In 2005, Intel CEO Paul Otellini announced the low-power x86 project that has developed into Atom. Otellini promised to deliver a 500mW Vista-capable x86 processor by 2010. In 2006, Intel sold most of its XScale business to Marvell Technology Group. At that time, *MPR* speculated that Intel might modify the x86 to make a more power-efficient version for embedded applications. *MPR* also urged Intel to consider licensing x86 cores to chip developers. (See *MPR 7/31/06-01*, "Intel's Embedded Future.")

Atom shows that Intel isn't willing to go that far—not yet. Atom is a great start toward reducing power consumption, but the ability to run x86 software remains the pivotal point. If tomorrow's ultramobile computing devices require no-compromise x86 compatibility, Atom is virtually a nuclear weapon. It's the only weapon RISC can't match. But if ultramobile devices don't need x86 compatibility, system designers can find many smaller, lower-power processors among the existing RISC architectures. The best RISC cores are perfectly capable of running sophisticated OSes, web browsers, MPEG codecs, and other relevant software. Indeed, they're doing it now.

Moreover, the RISC architectures are available as licensable processor cores, and some are extremely configurable. Developers can license these cores and customize them to build specialized SoCs highly optimized for their applications. The embedded world is rife with multicore SoCs that integrate CPUs, DSPs, peripherals, and memory. These custom chips are the key to reducing total system power consumption, total system costs, and system form factors.

In contrast, Intel will offer only a few standard parts based on Atom. Although Intel plans to introduce a few SoCs for some market segments, they will be semicustom standard parts, limiting the OEM's options for differentiation. (These future SoCs are code-named Tolapai and Lincroft; see the sidebar, "Decoding the Code Names.") Standard parts are ideal for high-volume commodity products like PCs, but they are less suitable for embedded systems, in which differentiation and innovation are everything. In particular, when battery life and mobility are paramount concerns, highly integrated SoCs enjoy a clear advantage over less-integrated standard or semicustom chips.

A Different Kind of War

In the first RISC vs. CISC war, RISC was the challenger. The x86 defended the high ground, entrenched with the world's

Atom's System Controller Slashes Power, Too

Cutting power consumption in the CPU is only a start. To offer an effective low-power platform, Intel must deliver the CPU with a system-controller hub (SCH) that reduces total system power. Therefore, Intel is introducing Atom microprocessors with an entirely new controller that integrates the north-bridge and south-bridge functions in a single chip. The code name for this device was Poulsbo, and the official name is the "Intel System Controller Hub for Atom." Intel's brand name for the Atom–SCH combo is the Intel Centrino Atom platform, formerly code-named Menlow.

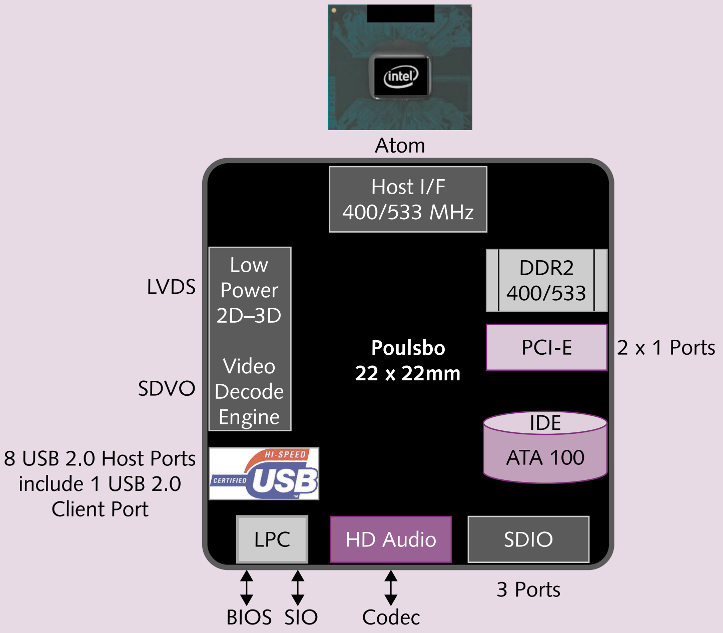
The Poulsbo chip measures 22- x 22mm and provides integrated graphics, a digital-audio interface, a main-memory interface, and numerous peripheral I/O interfaces. System designers will add DRAM, codec chips, and physical-layer (PHY) chips for the features they wish to support. Intel designed Poulsbo primarily for first-generation mobile Internet devices (MIDs). Later this year, Intel plans to introduce a similar system controller for the Diamondville variation of the Atom microprocessor. That controller will be intended for netbook and nettop computers.

Intel isn't publicly disclosing the precise range of Poulsbo's thermal design power (TDP) at this time, but it's approximately 2.0W, depending on the number of features enabled and I/O interfaces used. Intel estimates that "average power" for an Atom processor and Poulsbo chip is about 1.0W, depending on the workload. In comparison, the TDP for Intel's McCaslin mobile-PC chipset is about 6.0W, and the Alviso mobile-PC chipset has a TDP of about 12W. Poulsbo's power consumption is low enough for first-generation MIDs. For the future ultramobile products that Intel foresees, there will be even lower-power Atom processors and system controllers.

At first, Intel thought it could design Poulsbo by merging two existing chips, the 945GSE north bridge (code-named Calistoga) and ICH7-M south bridge. This idea was almost immediately discarded as impractical. Indeed, the project demanded such a fresh approach that Intel spun off the design team as a "new business initiative"—a separate entity funded like a startup by Intel Capital, the company's venture-capital arm. This team was led by principal engineer Joseph A. Bennett. Later, as the Atom project gained steam, Intel pulled Bennett and the other Poulsbo engineers back into the company proper. As a result, the north-bridge portion of the chip is a completely new design. The south-bridge portion retains only about half the elements in the ICH7-M south bridge.

To save power, Poulsbo must make compromises. It supports single-channel DDR2 memory, not dual-channel memory. (System designers can save additional power by using 1.5V SDRAM instead of 1.8V SDRAM, but the 1.5V memory is less common.) Front-side bus (FSB) speed is 400MHz or 533MHz. Maximum memory is 2GB, not 4GB. There are eight USB 2.0 host ports and one USB 2.0 client port. The parallel ATA interface has only one channel, not the usual two, so it supports only two disk drives. The north-bridge block runs at only 100MHz or 133MHz, and the south-bridge block runs at 33MHz. To keep costs down, Intel is fabricating Poulsbo in a very mature 0.13-micron technology.

Despite its compromises, Poulsbo has many features unexpected in a relatively low-power system controller. The integrated 2D/3D graphics engine can drive a 1,366- x 768-pixel display in 24-bit color. The integrated video engine can decode 1080p HDTV streams at 30 frames per second, using only 150mW in the process. It supports MPEG1, MPEG2, MPEG4, and H.264 video, and it's compatible with Microsoft's DirectX 9 and DirectX 10 graphics. Intel says Poulsbo can even handle the



Poulsbo block diagram. Together with an Atom microprocessor, this system controller comprises the Atom Centrino platform. Poulsbo uses less power than Intel's existing mobile-PC chipsets but is still too power-hungry for small handheld systems. Future versions will probably dispense with some PC-oriented features, such as parallel ATA and PCI Express.

Aero graphics in Windows Vista, though not as snappily as Intel would like. (Intel is lobbying Microsoft to make a lighter-weight version of Vista, more suitable for ultramobile computing.) To provide these graphics and video features, Intel licensed the PowerVX SGX graphics core and PowerVR VXD HD-video core from UK-based Imagination Technologies.

The block diagram shows a few more Poulsbo features. It doesn't show the additional steps taken to reduce power consumption. One is a new on-chip network that carries almost all the signals without using the numerous sideband channels found in other Intel chipsets. Actually, there are three fabrics: one for memory traffic, a second for I/O traffic, and a third message-based network that handles almost everything else. To manage these fabrics, the north bridge integrates an 8051 eight-bit microcontroller.

All memory transactions share a single content-addressable memory (CAM) buffer instead of using the separate buffers common in other Intel chipsets. Separate buffers reduce bottlenecks but use more power. Also, the memory controller optimizes transactions for power efficiency, not for best throughput. Writes to memory are allowed to accumulate for a time, keeping the DRAM in self-refresh mode as long as possible. When the transactions are released, specially optimized PLLs in the memory controller reduce the latency of exiting self-refresh mode.

Another power-saving feature is the FSB's CMOS signaling logic. Unlike the Advanced Gunning Transceiver Logic (A/GTL+) in other Intel chipsets, CMOS logic draws power only during voltage transitions, not when idle. Poulsbo, like Atom, supports both types of transceivers on the FSB. Intel says the CMOS transceiver saves 200mW to 500mW—a worthwhile trade-off for the slightly worse latency and slower effective bus speeds (400–533MHz). To reach higher bus speeds, system designers will probably have to sacrifice the power savings of the CMOS transceivers in favor of the A/GTL+ transceivers. The transceiver logic is fused, so Intel can offer devices with either type of transceiver.

Thanks to all these economies, the Poulsbo chip is only 22- x 22mm, about half the size of a Calistoga north bridge. TDP is three to five times lower than a traditional mobile-PC chipset. The low power consumption and low manufacturing cost of this chip will help Atom microprocessors penetrate the consumer-electronics market.

largest base of PC software. Now the x86 is the challenger. RISC defends the high ground, reinforced with countless design wins, lower power, lower costs, and greater design flexibility.

Much depends on a second battlefront raging between Windows and Linux. In the war for the desktop, the x86 had a crucial ally in Windows, which was uncommon or unavailable on RISC architectures. In ultramobile systems, Linux is less beholden to any particular CPU architecture than Windows is to the x86, and Linux has undeniable advantages over Windows. If Linux dominates the ultramobile market, Atom becomes just another processor that can run the multiplatform OS. Atom would have to compete with RISC processors on their terms.

For these reasons, *MPR* views x86 software compatibility as the difference that will prove decisive again. Pundits can debate benchmarks and power measurements forever, but if x86 compatibility becomes as indispensable in pockets and purses as it is on desktops, Atom will be good as gold. If it turns out that x86 compatibility doesn't matter, Atom will eventually decay into lead—or at least be relegated to the same relatively small market segments now claimed by VIA. Don't assume this RISC vs. CISC war will end as the first one did. ◇