The Intel® PXA250 Applications Processor

White Paper

February 2002

Contents

Overview

More than 98 percent of all microprocessors are used in "embedded" products, not PCs. Embedded products include all of the everyday items we see around us every day: cellular phones, microwave ovens, thermostats, video games, handheld PDAs, and more. Although PC processors seem to generate much of all the excitement in the press, it's the other 98 percent—the embedded processors—that are truly leading the way technologically.

The performance of embedded microprocessors rivals that of PCs of just a few years ago. With clock frequencies of 200, 300, and even 400 MHz, these chips offer better performance than ever before. More remarkably, embedded processors are becoming very frugal with electrical consumption. Their high performance levels, combined with low power requirements, are enabling a new generation of low-priced portable and handheld devices with unparalled features and usefulness. New PDAs, mobile telephones, communicators, wireless Web browsers, and other still-to-be-invented gadgets are being enabled by these new chips.

Balancing the three virtues of performance, price, and power consumption is no easy task. Super-fast microprocessors seem almost simple in comparison; how about one that's fast with low power and at a low cost? Merely supercharging a microprocessor with high-end features won't solve the problems of extending battery life, reducing weight, or lowering its cost. Nor will it simplify the engineers' job of improving time-to-market of new products.

Intel's newest generation of high-end 32-bit embedded microprocessors exhibits most of the features of performance-oriented machines combined with some innovative technologies for reducing power, increasing on-chip integration (without dramatically increasing silicon area), and preserving software compatibility—all at a low price.

A Tour of the Intel® PXA250 Applications Processor

It's easy to classify the Intel® PXA250 as just another 32-bit microprocessor but the processor core itself occupies less than 15 percent of the entire chip. Most of the silicon area is devoted to features that actively improve performance, integration, or reduce power consumption.

As the diagram below shows, the Intel PXA250 applications processor contains a number of peripheral features and functions. It includes integrated controllers for external memory, serial buses, parallel buses, wired and wireless communications links, expansion cards, and much more—all of which eases the engineering team's tasks of creating a complete product from a single chip. About the only things an engineer needs to add are a battery, memory, communications subsystem, and an LCD screen—and that last part is optional.

At the core of the Intel PXA250 is, of course, an Intel® XScale™ core-based microprocessor. Intel XScale is a 32-bit RISC microarchitecture based on architecture by Advanced RISC Machines (ARM*), now the most popular 32-bit embedded CPU family in the world. ARM-based and Intel XScale technology are completely binary compatible. So software and software-developed tools designed for older ARM processors also work on newer Intel XScale core-based processors. Of course, new Intel XScale technology-specific development tools work even better and take advantage of Intel XScale technology's newer features.

The Intel XScale RISC microarchitecture is noted for its efficiency. It obtains high performance, minimal number of silicon transistors, which requires less power to operate the chip. It also means the chip processor itself will take up less silicon, making it smaller and less expensive to manufacture in volume. This combination of high performance, small size, low power, and modest cost gives the Intel PXA250 applications processor some compelling advantages over competing processors.

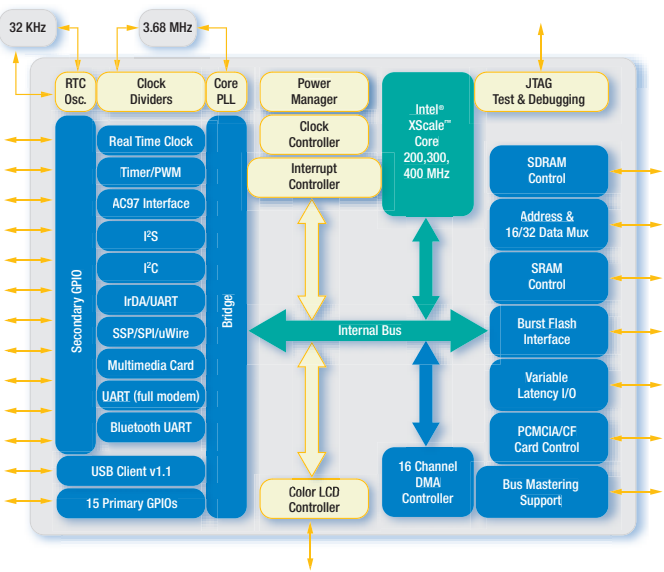

Figure 1 Intel® PXA250 Block Diagram

Intel® XScale™ Enhancements

The Intel XScale core in the Intel PXA250 applications processor is a generational improvement over the previous Intel® StrongARM* processors, the Intel® SA-1100 and the Intel® SA-1110. Unlike the SA chips, the Intel PXA250 applications processor includes the ARM®/Thumb® code-compression technology. For all its advantages, RISC architecture has one weakness: code density. Code density describes the space needed to store code (software) in memory. RISC chips generally have significantly larger programs than Complex Instruction Set Computing (CISC) processors requiring more memory to store the same amount of code. The Thumb technology improves this situation dramatically, compressing software density by about 30 percent over normal RISC code. This compression (and the corresponding real-time decompression) is handled automatically by the Intel PXA250 applications processor; engineers and programmers are generally unaware of its operation. They only notice the improved memory utilization.

Another improvement in the Intel XScale core is the dual-multiply/accumulate (dual-MAC) instruction. A MAC operation is a relatively new addition to most computers and microprocessors, borrowed from the Digital Signal Processor (DSP) world. MAC operations are so vital to many audio, video, and wireless applications, that one or two MAC instructions can dramatically benefit a chip's ability to run these multimedia applications.

Pipelining:

"Pipeline" sounds like plumbing, but it's really a high-performance instruction processing technique. Think "horsepower" and you begin to get an idea of how the pipeline improves the performance of a high-speed processor like the Intel PXA250 applications processor.

With several million transistors the Intel PXA250 applications processor has a lot going on underneath the hood. There are all kinds of advanced processes happening during every clock cycle—and at 400 million cycles (400 MHz) per second or more, that's not much time to do it in. The Intel XScale seven-stage pipeline, helps to make this keep the process efficient.

This concept is similar to that of an automotive assembly line, where the complex tasks of assembling the car are divided up into individual stages. In the Intel PXA250 applications processor's case, every function is evenly divided into seven stages, with different parts of the chip working in harmony to keep instructions flowing smoothly through the processor's execution units. The seven stages of the Intel PXA250 applications processor's pipeline are shown in Figure 2.

The more stages in a pipeline, the higher the frequency the processor can execute instructions. Having seven stages currently allows the Intel PXA250 applications processor to scale to frequencies of 400 MHz with greater headroom for later versions.

In our assembly line example, if an assembly line is shortened, each worker must do more work per unit of time, slowing the entire line down. If the Intel PXA250 applications processor's pipeline were any shorter, its clock rate would be limited to around 250 MHz because there would not be enough time for the chip to accomplish all the necessary work.

In addition, longer lines enables faster speeds but they also require more co-operation and coordination among the execution units. The PXA250 processor has a finely tuned "coordinator" function that ensures this intra-chip co-operation and allows it to run very efficiently compared to similar processors.

Intel® PXA250 Applications Processor Pipeline: Step By Step

The first two stages of the pipeline load a 16-bit or 32-bit Intel XScale core instruction from memory (or from the instruction cache if it's already there). This simple task requires two pipeline stages because memory chips are relatively slow compared to the Intel PXA250 applications processor. The processor also needs to look up branches in the Branch Target Buffer (BTB). Although the work isn't hard, it can be time-consuming.

Figure 2. The seven stages of the Intel® PXA250 applications processor's pipeline.

The next stage "decodes" the 32-bit RISC instruction by separating some of its bits and distributes these parts around the chip. This is similar to the order-entry process in a large manufacturer: incoming orders determine what the rest of the production line will be building henceforth.

The fourth stage loads data from the chip's onboard register file. This procedure executes more rapidly than the two-stage instruction fetch, because this data is already within the processor's registers.

The fifth stage is where all the real work happens. Surprisingly, almost all of the Intel PXA250 applications processor's instructions can be executed in a single cycle. This is the actual "computing" part of the chip where numbers are added, subtracted, or multiplied; where bits gets shifted left and right; or where logical comparisons are made. In a sense, all the other pipeline stages exist only to accommodate and support this stage.

The sixth pipeline stage is a "housekeeping" function, where certain internal variables are frozen, or committed, to memory. If a system error or fault occurs, this is where the Intel PXA250 applications processor will restore the proper operating condition of the program.

Finally, the seventh stage writes the results of the instruction back out to memory, cache, register—whatever destination is appropriate for that instruction.

At this point, the process is now complete and the chip can start on the next instruction.

In reality, it already has. The Intel PXA250 applications processor keeps seven different instructions "in the wings" at all times, meaning that its pipeline is always full, with seven instructions in different stages of execution. At any given moment the Intel PXA250 applications processor is fetching two instructions, decoding one, loading the operands for one, executing one, saving the state for another, and storing the results of the last one. All of this activity happens without a hitch, no matter what the program does.

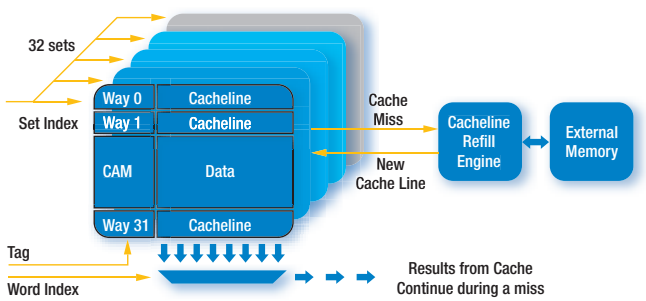

Figure 3. Hit-under-Miss-Data-Cache

Exceptional Cases

A few exceptional cases exist when executing instructions. One case is that of "branch" instructions, where the program changes the course of the software, similar to changing a product when it is still on the assembly line. (This is discussed under Branch Prediction.)

Another special case is "Multiply-Accumulate," or MAC, instructions. Because MAC instructions are complicated, they take longer than one cycle to execute. Rather than slow down the entire processor's operation with this one special case, MAC instructions use their own pipeline off to the side of the main seven-stage pipeline. After a MAC instruction finishes the fourth stage of the pipeline (register access), it moves to its own pipeline for calculations. When these are done, it rejoins the main pipeline to have its results stored.

Another exceptional case is a "cache miss"—a condition where the data the Intel PXA250 processor requires is not found in one of the on-chip caches. This is often a performance-sapping situation, which we will discuss in more detail.

Cache Performance Tweaks

We all know that processor caches improve overall execution performance. Some improve performance more than others. (For those who are new to computer or microprocessor terminology, a cache is a small memory bank kept on the chip itself. It duplicates some of the devices of the "real" off-chip memory, thereby increasing performance by keeping often-used data "close at hand." Generally speaking, the bigger the cache, the better the processor's given performance.)

Ideally, 100 percent of all the data a program requires would be in the cache. Often this data is not loaded in the cache, resulting in the performance-sapping "cache miss" condition: this requires a time-consuming fetch of the needed data out in the device's main (off-chip) memory.

Apart from increasing the size of the cache, or employing a few engineering tricks to improve the cache hit ratio, cache performance is mostly fixed. What isn't fixed, however, is a cache's behavior when it's being used heavily. Under heavy-loading conditions, caches are subject to a "miss penalty" that can unnecessarily reduce performance. The Intel PXA250 applications processor avoids this penalty with a clever "hit-under-miss" technique that keeps data moving as evenly as possible. See Figure 3.

In contrast to most microprocessors, which are forced to halt instruction execution until the data cache miss is processed, the Intel PXA250 processor's "hit-under-miss" function works by permitting it to continue doing other work while it is processing the data cache miss.

Its pipeline is likely filled with other instructions that do not require the missing data, and therefore do not need to be held up waiting for it. The Intel PXA250 applications processor opportunistically proceeds with other work while the offending instruction waits for its data.

This capacity works well most of the time, but what if one of those other instructions also needs data from the cache? In other chips, this would stall the microprocessor yet again—once the cache is tied up processing a miss, it becomes useless to other instructions, with its data, unavailable.

The Intel PXA250 processor, in contrast, doesn't take its cache off-line just because there was a previous cache miss; instead, it can still service other instructions that need the cache. Subsequent instructions that hit the cache are serviced even while there's an outstanding cache miss. This is called "hit-under-miss" and it satisfies a large number of conditions that occur frequently in real code. The entire program runs more smoothly (and predictably) when all instructions aren't penalized for one instruction that misses the cache.

Branch Prediction

One of the problems with very fast microprocessors is that they often lose performance when a program software requires an instruction branch. A fast processor is like a fast freight train: once it gets going, it's hard to make it change direction. With computer software, instruction branches are like switches in the railroad tracks. They cause the processor to lose momentum and waste performance.

To alleviate this problem, the Intel PXA250 applications processor uses an advanced technique called "branch prediction" to speed up the processor's "train of thought" and keep things on track and running smoothly. Branch prediction comes from high-end computer systems and is rarely seen on low-cost, high-volume microprocessors.

Branch prediction works by predicting which way the software will branch. Just as with railroad tracks, there are always two choices: programs can either continue straight ahead in the direction they are going (i.e., a branch is not taken) or they can branch to one side (a branch is taken). Microprocessors and trains both run far more smoothly if the direction of the switch is known ahead of time. Without branch prediction, the train—and similarly a microprocessor—has to come to a complete stop at every switch (or branch)—a significant waste of time and performance.

The Intel PXA250 applications processor can predict the outcome of up to 128 different branches in a program (the actual number might be smaller, depending on the program). It does this by maintaining a BTB, inside the processor itself. This buffer is a private reserved memory space that remembers what happened at each of the last 128 branches it encountered. If that branch is ever encountered again, the Intel PXA250 processor will have knowledge of the last outcome that it can use to predict the next one. See Figure 4.

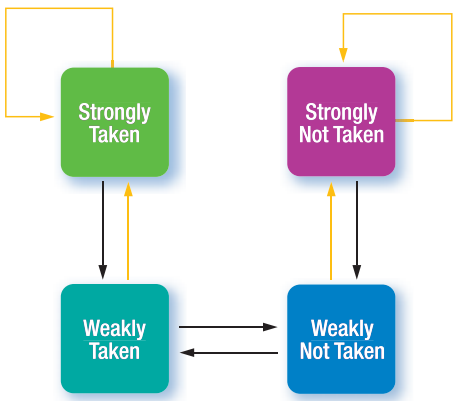

Figure 4. Software Branches Diagram

Statistics show that software branches tend to behave the same way every time. That is, if a particular branch was taken before, it will probably be taken again. If it was not taken before, then it probably won't be taken next time, either. Once again, programs behave like trains: they tend to follow the "main line" and visit outlying (branch) spurs only once in awhile. Every time the Intel PXA250 applications processor encounters a branch, it records whether or not that branch was taken in its BTB. Each pass over the same branch adds another "tally mark" to the BTB. Each tally mark reinforces the Intel PXA250 processor's opinion about that branch.

If the tally marks consistently show that a particular branch is rarely taken, the Intel PXA250 processor will predict that outcome for the next time and keep the train running straight ahead. On the other hand, if the BTB history shows that the branch has consistently been taken, then the processor will predict an upcoming branch. The applications processor will then prepare to move off the main line and head down that particular branch. This advanced preparation saves significant time and keeps the processor moving along at top speed. Correctly predicted branches consume no additional time. In effect, the Intel PXA250 processor handles the corners just as fast as it handles the straightaways.

What happens if a branch prediction is wrong? Absolutely nothing. Remember, most microprocessor chips don't have any branch prediction at all, so even an incorrect prediction is no worse than what most other microprocessors do all the time. The Intel PXA250 applications processor does its best to prevent this from happening.

Learning From Experience

For each one of the 128 branches stored in the BTB, the Intel PXA250 applications processor makes one of four predictions about that branch. The four predictions are shown in Figure 5.

The Intel PXA250 processor "learns" over time, refining its predictions for greater accuracy. The first two cases predict the branch will be taken; the second two predict it won't. Every time its prediction is correct, the Intel PXA250 applications processor reinforces that prediction for next time. For example, if the prediction was "weakly taken" and the branch was, indeed taken, the Intel PXA250 processor upgrades its prediction to "strongly taken." The same rule applies when upgrading "weakly not taken" to "strongly not taken" every time that prediction is correct.

Why four conditions, then? If the Intel PXA250 processor makes the wrong prediction, it can change its mind, but only with strong evidence that it's on the wrong track. If its prediction is "strong" but turns out to be wrong, it downgrades its next prediction (for that specific branch) to "weak." If it's still wrong the Intel PXA250 processor then reverses its prediction to "weak" in the opposite direction. If that turns out to be correct, it reinforces its new prediction to "strong." As you can see, a prediction has to be wrong twice in a row before the Intel PXA250 processor will change its mind; missing a prediction once in a while won't throw it off the track. This four-state prediction algorithm gives the Intel PXA250 applications processor flexibility to alter its predictions for changing conditions without vacillating unnecessarily.

Current Prediction	BTB Bits	Next Prediction If Correct	Next Prediction If Incorrect
Strongly Taken	00	Strongly Taken	Weakly Taken
Weakly Taken	01	Strongly Taken	Weakly Not Taken
Weakly Not Taken	10	Strongly Not Taken	Weakly Taken
Strongly Not Taken	11	Strongly Not Taken	Weakly Not Taken

Figure 5. The Predictions

Best of all, this process is completely transparent to the programmer, engineer, or developer. It's all completely automatic and internal to the Intel PXA250 processor.

Power-Saving Techniques

The real beauty of many high-end embedded microprocessors like the Intel PXA250 is not strictly their performance, but the fact that they extract this performance while consuming minute amounts of power. A pair of AA batteries can now power a 400 MHz microprocessor—and its dozens of attached peripherals—for hours, days, or even weeks. This level of efficiency doesn't come easily, and the Intel PXA250 applications processor employs some very advanced techniques to achieve it.

First and foremost among the power-saving weapons in the Intel PXA250 processor's arsenal is the company's 0.18-micron, six-layer-metal complementary metal-oxide semiconductor CMOS technology. This fabrication process has been proven in a number of other Intel high-volume silicon products; its tight transistor geometry increases speed and its low dielectric cuts power consumption. The Intel PXA250 processor's extremely low operating voltage (as low as 0.85V) slashes power consumption even further.

Another Intel PXA250 processor power management feature is its automatic clock gating. The chip has several million transistors but not all of them are used all the time. The chip aggressively works to temporarily shut down elements that aren't being used by "gating off," or disabling, their input signals. Transistors that aren't switching don't use energy, so the Intel PXA250 processor attempts to keep the entire chip as still as possible. Major portions of the chip can be idle for one clock cycle (one 400-millionth of a second) and fully active the next.

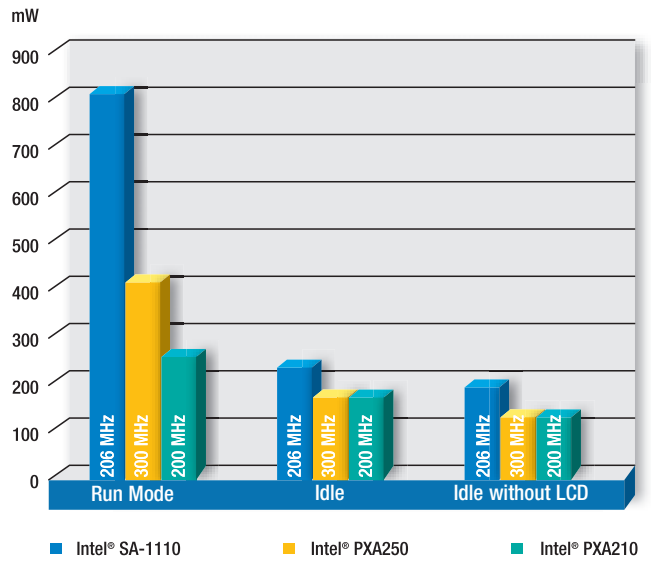

Figure 6. Low Power Targets

Clever programmers can also further reduce power consumption through software. If a program can detect when the Intel PXA250 processor won't be used for a short time, they can put part of, or the full processor to sleep. Even short naps of a few microseconds are worthwhile for maximizing energy saving. For example, a handheld computer can take naps in between each tap on the keyboard; even the fastest typists are slow compared to a microprocessor. The Intel PXA250 processor might be asleep for as much as 95 percent of the time, yet the user would notice no difference in the device's operation.

Lighter naps (called "idle mode") drop the Intel PXA250 processor's power consumption by a factor of two to three, to about 100 mW (one-tenth of a watt). Longer naps ("sleep mode") can slash power to just 50 µW—less power than can be obtained from two lemons and some copper wire.

Turbo Mode and the Phase Locked Loop

Finally, every microprocessor needs a crystal or two to keep it running smoothly. Just like the ones used in clocks or wristwatches, crystal-controlled clocks are inexpensive and extremely reliable. However, as more features and functions are added to a chip, they often require their own clocks to maintain consistent operation—causing a rapid proliferation of on-chip clocks. This problem worsens as chips get faster because high-frequency crystals cost more than low-frequency ones. And consume more power. See Figure 7.

One solution to this problem was discovered years ago in the form of the Phase Locked Loop (PLL). A PLL is a clever digital circuit that can "multiply" the frequency of a simple low-cost crystal, boosting it up to higher frequencies. Using multiple PLLs, the Intel PXA250 processor can generate all the clock signals its various features need from a single, inexpensive crystal.

The PLL has the added side benefit of allowing users to "downshift" the Intel PXA250 processor to a slower speed to save energy. If the full 400 MHz of performance isn't required, the Intel PXA250 applications processor can operate easily at 100 MHz. Almost any arbitrary speed is possible, and the associated power savings can be huge when the chip isn't running at full speed.

The drawback of every PLL is that, like a car's transmission, a PLL needs a small amount of time to shift from one gear to another. For a car, that means losing momentum. For a microprocessor, this can mean pausing processing and sometimes that's not an acceptable option. The Intel PXA250 processor helps overcome this issue. In addition to the programmable PLL, there's also a "turbo mode" clock when especially rapid changes are required.

The Intel PXA250 processor's turbo mode bypasses the usual PLL—and its associated switchover delay—by generating a second clock source from the same external crystal. The turbo mode clock can run at 1.5 or 2.3 times, the processor's normal speed. Best of all, users can switch the chip between the PLL clock and the turbo-mode clock in a single cycle (25 nanoseconds) without missing a beat. When "on the fly" frequency changes are required, turbo mode is a technically elegant solution to this long-standing problem.

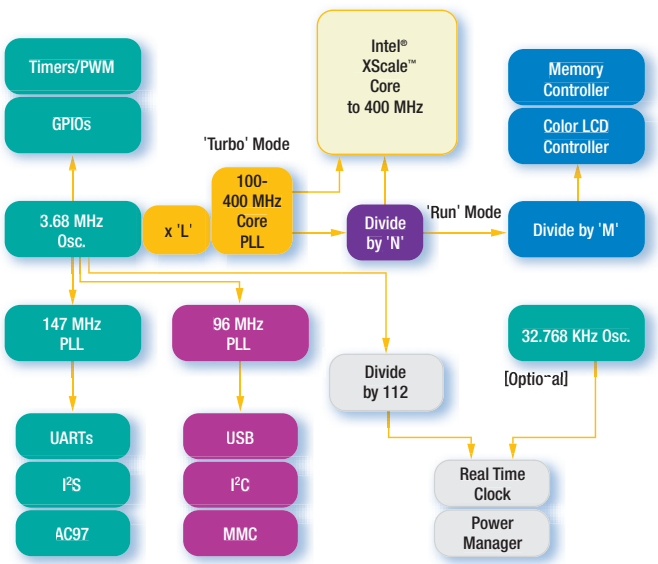

Figure 7. The Intel® PXA250 Clocks

As an example, you could design a portable battery-powered communicator that runs at full speed when it's downloading Web content or showing video, but then drops into low-speed, low-power mode when it's just working as a cell phone. Or imagine a small Walkman-like device that plays MP3 files while the Intel PXA250 applications processor (nearly) sleeps, but which wakes up instantly when the user presses a button to, say, play a video game. The PXA250 processor is as fast as you want it to be, but also as small, quiet and low power as you need it to be.

A Wealth of Peripherals

Even the fastest processor cannot stand alone, it needs memory, peripheral controllers, and support circuitry to bring all the elements of the package together. In small battery-powered consumer devices, these "accessories" can consume most of the precious room inside a small plastic case and complicate printed-circuit board layout—not to mention the design time required to locate, select, qualify, and design-in all these peripherals. Built-in peripherals are always simpler and more compact, assuming they do not enforce compromises in the designers' product vision. The Intel PXA250 applications processor spoils users with choices, providing all the basic and not-so-basic features and functions one could wish for.

Universal Serial Bus Controller

As its name implies, Universal Serial Bus (USB) connectivity standard has indeed become almost universal. USB ports appear on every PC and Macintosh* as well as on video games, digital cameras, home electronics, and computer add-ons such as keyboards, joysticks, mice, and others. Its elegance lies in its simplicity: it is truly "plug and play." If the USB cable fits, the USB standard ensures it will work.

USB was an obvious choice for the Intel PXA250 applications processor, allowing instant access to and compatibility with a growing array of fixed, portable and wireless devices already in the market. As a processor for PDAs, the Intel PXA250's USB interface provides a fast and easy way for users to synchronize key information—such as their address books and e-mail—with their home or office PCs. Two Intel PXA250 processor-based communicators can synchronize contact information with each other over a USB connection. Programming information, Web content, firmware upgrades, authentication codes—all of this can be transferred in seconds, easily and trouble-free, over USB connections to thousands of other devices.

Direct Memory Access Controller

Few elements of system design can seem as dull as a Direct Memory Access (DMA) controller, but nonetheless, some features shine through. The DMA control capability of the Intel PXA250 applications processor is a robust one, including 16 true independent channels, each with full access to external memory as well as all the on-chip (internal) peripherals.

The purpose of a DMA controller is to handle simple, but important, transfers between memory locations or between memory and a peripheral controller (for example, a device's USB controller). The processor core can be—and often is—used to do this. However, processing power is precious and not the most efficient means of doing this, when all the system designer wants to do is move a kilobyte of data from Point A to Point B. DMA controllers are geared for just this kind of "heavy lifting" and they can do it without disturbing the processor or taking time away from program code execution.

The secret of this capability lies in the DMA controller's descriptor rings and a feature known as chaining. A descriptor consists of a short table of instructions: "move 512 bytes from here to there and tell me when you're done." A descriptor ring is a sequence of these; when the DMA controller is finished with its first task it can automatically commence working on the next job without disturbing the main Intel XScale core-based processor for additional instructions. When the entire chain of tasks is complete, the DMA controller might be commanded to start the whole process again from the top, hence "descriptor ring."

With 16 DMA channels (plus the processor itself), the Intel PXA250 applications processor can simultaneously keep up to seven memory-to-memory or memory-to-peripheral transactions. This type of on-chip cooperation is what makes leading-edge products maintain high performance levels.

Baseband Interface, UARTs, and More

For wireless communications, the Intel PXA250 applications processor gives designers all the options they need without shoehorning them into a single solution. For Bluetooth*, 802.11b, infrared, and other wireless interfaces, the Intel PXA250 applications processor likely supports it. Central to all these wireless standards are serial controllers—called Universal Asynchronous Receiver/Transmitter (UARTs). The Intel PXA250 applications processor has two high-speed UARTs, supporting bit rates of up to 1.84 MHz—more than enough for Bluetooth and other high-end wireless standards. Full modem-control (Request To Send, Clear To Send, etc.) capabilities are also present if required.

And still there's more. Inter-Integrated Circuit (I2C) and Inter-Integrated Circuit Sound (I2S) both provide low-cost, low-pin-count interfaces to a wide variety of other chips in the system. Service Switching Point (SSP), Service Provider Interface (SPI), and mWire interfaces all call the Intel PXA250 applications processor. Even IrDA, the ubiquitous infrared standard for remote control and other popular devices, is handled by the processor with its own interface.

Finally, there are as many as 16 general-purpose I/O pins that are individually programmable. As inputs, these pins can even generate interrupts—a handy way to implement pushbuttons in the final product.

Liquid Crystal Display Controller

The most exciting feature of every handheld device is often the display. The Intel PXA250 applications processor supports a wide variety of displays that can range from small QCIF displays for cell phones to larger panel displays for PDAs and Web tablets. Integrating a quality LCD controller on a chip means that designers of these systems can reduce the chip count (and subsequent power requirements) of the device while having fast access to system memory. Speaking of system memory, what is great about the LCD controller on this part is that is has its own two Channel Direct Memory Access unit to allow it to off-load the memory moves from the processor core freeing up performance.

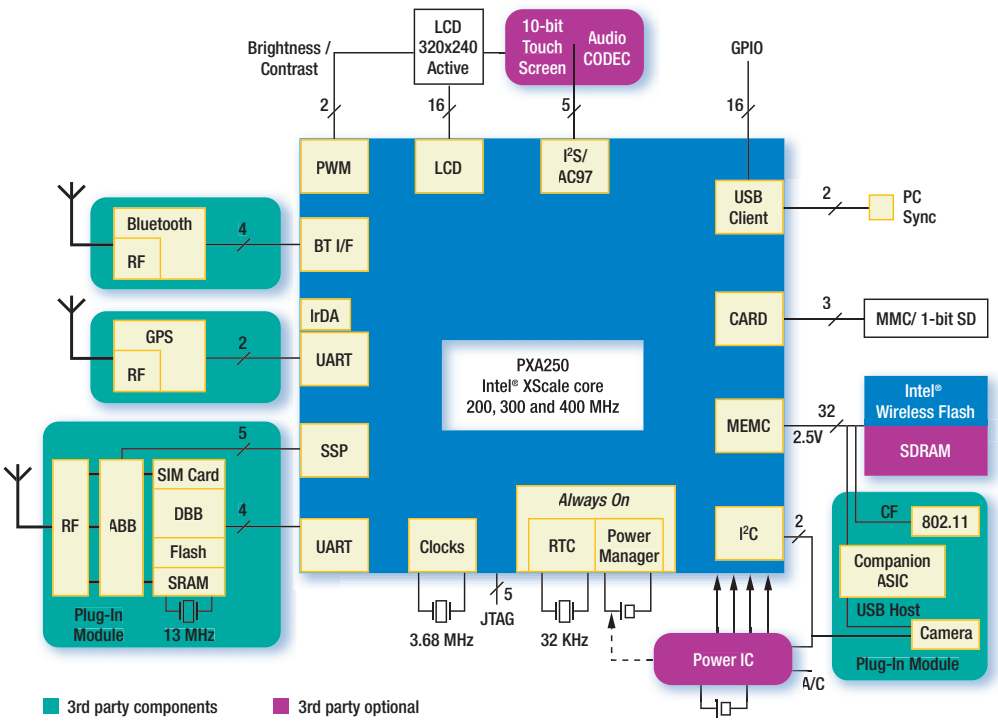

Figure 8. The Advance Wireless PDA

The Intel PXA250's LCD Controller supports a wide variety of single- and dual-panel LCD displays, with resolutions up to 800 x 600 pixels. It supports passive Monochrome panels with up to 256 gray-scale levels and both Active or Passive Color panels with up to 65,536 color values. For the best viewing experience with Passive panels, the high-quality TMED dithering algorithm is provided. To allow memory efficient storage of the Frame Buffer, picture data may be encoded into pixels ranging in size from 1 bit to 16 bits per pixel. The Intel PXA250 applications processor includes a 256-entry color palette Random Access Memory (RAM) that allows small encoded pixels to exploit the full 65,536 color values.

Memory Controller

A processor without memory can be likened to a fish without water, or a cake without frosting: an unhappy and unappealing state. Yet many (in fact most) processors don't include memory controllers, relying instead of generic bus interfaces for all their data-transport needs. Often the job of building a memory controller is left as an exercise for the system engineer however, this is, the case with the Intel PXA250 applications processor.

The 32-bit-wide memory controller used by the Intel PXA250 applications processor keeps operating at top speed. It handles Synchronous DRAM (SDRAM), Synchronous RAM (SRAM), Intel's Wireless Flash memory devices, or any combination of these. High-end systems often use copious quantities of commodity SDRAM, while smaller (or faster) systems might prefer SRAM.

Either way, Intel's Wireless Flash memory devices help provide nonvolatile storage for boot-up code, drivers, identification and customization information, and other infrequently updated data. The Intel PXA250's applications processor memory bus can even run in 16-bit mode for systems that are optimized for minimal space and power, with maximum performance.

MMC/SD & PCMCIA/CF Controller

Main memory doesn't stop at SDRAM, flash, or SRAM. For today's portable devices, users often want to add their own plug-in memory, or peripherals that emulate plug-in memory, such as PCMCIA cards. To this end, the Intel PXA250 applications processor supports practically all the current standards for consumer memory modules and peripherals.

To start, there's the combination PCMCIA/SD interface. PCMCIA is the oldest and still popular plug-in memory standard. Although it's hardly ever used for memory any more, PCMCIA is hugely popular for plug-in Ethernet cards, 802.11b wireless-networking cards, miniature digital cameras, and even hard disk drives. The same interface can also be used for CompactFlash™ cards, the popular standard for small nonvolatile memory, particularly favored by digital camera users.

There's still more. The chip also supports SD (secure digital) cards, just now emerging as the preferred way to distribute MP3 music content and other copyrighted material in a convenient, cost-effective, and secure format. And the same interface pins can be used for MMC (multimedia cards) plug-ins, yet another option for the high-tech consumer.

Conclusion

Embedded processors have certainly made dramatic advances in the past few years. From low-cost versions of "last year's model" they have now become technology leaders and volume leaders. The ingenuity and technology that go into today's high-end processors for consumer electronics are second to none. Indeed, developing a processor that's fast, low cost and miserly with power is a challenge few companies can successfully undertake.

These processors translate into products few customers can resist. Today, the average consumer can carry around more processing power in their pocket than NASA put on the moon in the 1970s. Power for communications, power for compressing and processing video, power for music and games—it's all there, tomorrow, in the palm of your hand.

For more information, please visit the Intel® Applications Processors Web site at: http://developer.intel.com/design/pca/applicationsprocessors/

For more information on the Intel® PCA Developer Network Web site, please visit: www.intel.com/pca/developernetwork

Intel Advantage

- The Intel® PCA is a wireless architecture that will enable a new generation of high-performance, ultra low-power mobile devices combining voice communications and Internet access. Intel PCA will accelerate the design and deployment of next-generation cell phones, personal digital assistants and other wireless Internet devices, as well as the services and applications that will make these devices attractive to end-users and telecom service providers alike.
- Intel supplies industry-leading processors, communications chipsets and flash memory, making it easier to build a scaleable, re-programmable device optimized for the wireless Internet. Intel roadmap will feature various integration options of these successful products.
- Beyond the products Intel is enabling the ecosystem building a community among OEMs, enterprise, carriers and ISVs to ease and speed the reality of next generation voice and data devices and applications
  - — Wireless Competency Centers
  - — PCA Developer's Network
  - — OS strategy
  - — ISV program to enable software spiral
  - — Driving communications standards

Intel Access

Intel® Personal Internet Client Architecture Home Page	http://www.intel.com/pca
Intel Literature Center	http://developer.intel.com/design/litcentr (800) 548-4725 7 a.m. to 7 p.m. CST (U.S. and Canada) International locations please contact your local sales office.
General Information Hotline	(800) 628-8686 or (916) 356-3104 5 a.m. to 5 p.m. PST

For more information, visit the Intel Web site at: **developer.intel.com**

UNITED STATES AND CANADA
Intel Corporation
Robert Noyce Building
2200 Mission College Blvd.
P.O. Box 58119
Santa Clara, CA 95052-8119
USA

EUROPE
Intel Corporation (UK) Ltd.
Pipers Way
Swindon
Wiltshire SN3 1RJ
UK

ASIA-PACIFIC
Intel Semiconductor Ltd.
32/F Two Pacific Place
88 Queensway, Central
Hong Kong, SAR

JAPAN
Intel Japan (Tsukuba HQ)
5-6
Tokodai Tsukuba-shi
300-2635 Ibaraki-ken
Japan

SOUTH AMERICA
Intel Semicondutores do Brasil LTDA
Av. Dr. Chucri Zaidan, 940-10° andar
04583-904 São Paulo, SP
Brazil

Order Number: 250621-001 Printed in USA/0202/5K/ASI/DC