To appear in Proceedings of the 1994 IEEE Computer Conference, San Francisco.

Low Power Hardware for a High Performance PDA

Michael Culbert,
System Architect,

Apple Computer Inc.
Cupertino, California 95014

Abstract

The first product in the Newton family operates under severe constraints in the areas of performance, cost, heat dissipation, power consumption, scalability, size and weight. This talk gives an overview of the Newton MessagePad system hardware, and focuses on the techniques and tradeoffs used to overcome these constraints. In particular, synergetic migration of functions from hardware to software and vice-versa will be discussed.

Introduction

The Newton family of products has one overriding goal: to provide the user with a fluid and simple man/machine interface. No Newton user should have to realize that they are using a computer. They should be able to use the Newton as they would use a piece of paper. As they become more accustomed to the Newton, new and powerful applications should become visible. In the pursuit of this goal, several key design choices must be considered, not the least of which are size, weight, cost, and power consumption. This paper concentrates on the power consumption of both the hardware and software of the current MessagePad system.

The Processor

Early in the design process, it became apparent that processor performance comparable to Intel 486 class of processor was required in order to give the user a smooth user experience, given the software environment that we had begun to create. Of more than a dozen processors that were evaluated, only a small number could meet most of the guidelines established for power consumption, cost and performance, and fully static operation (allowing the clock to be stopped at any time).

Only one processor, the ARM3 from Acorn Computers Limited (United Kingdom), met all of the goals. The ARM3, however, was not a perfect fit. The ARM3 had no integral memory management unit, and Acorn had no resources to develop one. Apple went to Acorn, and jointly with VLSI Technology Incorporated, created a new company now called Advanced Risc Machines, Limited (United Kingdom). The purpose of this company was to provide a development resource to keep the ARM architecture current and competitive in this new market arena.

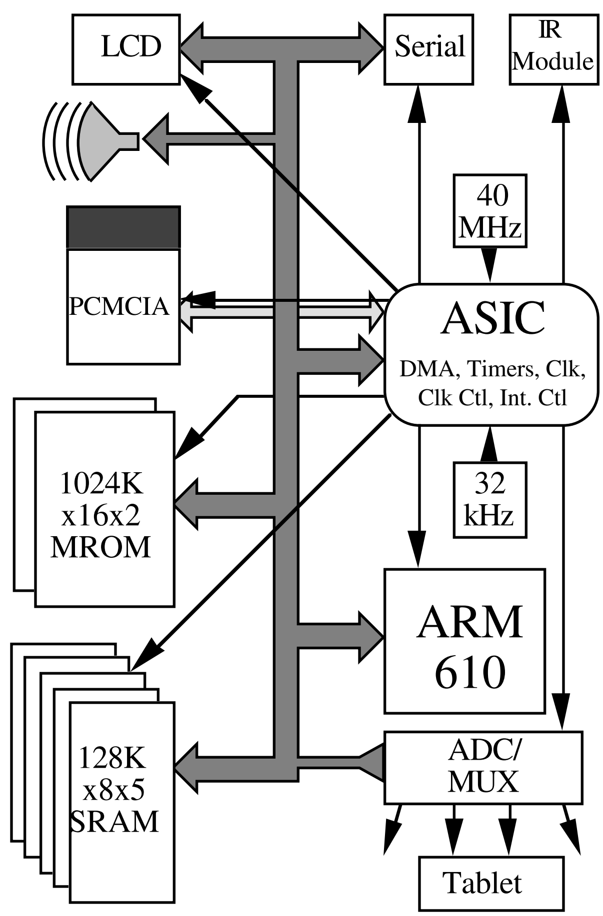

Fig. 1: MessagePad™ Simplified block diagram

Newton, MessagePad, AppleTalk and LocalTalk are trademarks of Apple Computer, Inc.

MIPS/Watt is an important metric for evaluating system performance over time. This term was coined early in our processor search, and most of our processor suppliers (as well as the rest of the industry) now recognize and use this new benchmark. The ARM processor has about twice the MIPS/Watt rating of its closest competitor.

The creation of Advanced Risc Machines made the future of the ARM architecture more certain, as well as creating a resource to design and implement a full 32 bit version of the ARM architecture with an integral MMU. The task was then to create a system around this processor that allowed us to take maximum advantage of the architecture.

The ARM610 processor, as it is known today, has several advantages over its current competition. The design consists of several flexible small blocks. There are three primary blocks: the processor core, known as the ARM6 core, the memory management unit that incorporates Apple’s domain memory management architecture, and a low power cache. ARM describes its processor architecture as not being a "pure" RISC. The departures from purity permit ARM code to be as dense as more traditional CISC architectures. This has several benefits, ranging from the obvious cost savings of being able to squeeze more code into fewer bytes of memory, to the slightly less obvious improvement in cache performance. This subtly improves both performance and power consumption of the overall system.

The memory management unit of the processor was carefully crafted to save system power and enhance performance. Permissions are assigned on 1K sub-pages while virtual mappings are done on more traditional 4K page boundaries. This permission mapping scheme allows effective management of small blocks of memory for independent processes. To decrease the power consumed by context switches, we took advantage of the fact that we have a large (32 bit) virtual address space within a small physical memory space. The cache of the ARM processor is a virtual cache, and in a more traditional operating system approach, the translation lookaside buffer (TLB) of the MMU and the cache would have to be flushed on each context switch. By assigning each process its own virtual address space within the OS, we have avoided flushing the TLB or the cache at every context switch. The other major cause of TLB flushes is permission changes in the MMU page tables. Apple's domain memory management allows us to avoid almost all of these flushes as well. The domain memory management allows the OS to assign pages or sections of memory an ID. This ID allows the OS to group pages or sections together as a domain. Permissions can be globally changed on domains by changing the contents of a single internal 32 bit register. Changing permissions requires no flushing of the TLB or reading of the page tables. Rather than having to switch almost the entire page table and then flush the TLB, certain tasks like garbage collection can simply write to a single 32 bit register to do its permission modifications.

Cache miss rate (as opposed to cache speed) is one of the important factors optimized for in the design of the core Newton operating system (OS). We chose to use the cache organization that had been designed for the ARM3, ARM610’s 26 bit predecessor. Though the cache had not been designed with our requirements in mind, it had nonetheless been originally designed with cost, power, and performance in mind. It needed to be small in order to meet the die area requirements of the processor, but it needed to be the most effective small cache that could support our system. It takes advantage of features of the ARM instruction set to keep power dissipation low. Its fairly unusual 64-way set associative organization, random replacement architecture was well suited to our requirements. This architecture ultimately led to a miss rate of less than 10% in our production system.

The Power System

The choice of a power supply voltage has major cost, performance and power tradeoffs. We initially chose a 3.3V ±10% main supply. This choice forced us to drive many of our silicon partners to produce 3V memories, communications peripherals, and ASICs earlier than they had originally planned. As the system design became more firm, and the cost implications of a 3V design became clear, we had to make a choice. The system performance would have be reduced significantly, the cost would be increased by more than 50%, or the battery life would be degraded by more than a factor of two. It was clear that we could not sacrifice performance or cost for this product, so we made the necessary choice to change the main power supply rail to 5V for the first generation products.

The choice of a 5V rail had a distinct impact on the choice of batteries. To achieve a low cost power supply we needed a minimum of 4 cells. Our original choice of AA batteries no longer fit within the physical product design. We had to choose between enlarging the product, or again reducing the battery runtime. The use of AAA size NICAD batteries could still provide one of our primary goals, which was to provide a typical user with at least one week of use of the product, so AAA batteries were chosen for the first product.

The power supply design was complicated because the loads in the system vary from about 10mA when the system is idle to about 400mA in the worst case. Additionally, the load can change virtually instantaneously from an idle state of 10mA to an active state of about 180mA . The task was to design a low cost converter that was very efficient in both cases. We achieved about 85% efficiency in the idle case (10mA) and over 90% in the active case (180mA).

Regulation directly impacts the cost of a power system. We had to trade regulation for power system cost. The power supply rail can vary almost 5% when the system is running, as the power supply struggles to cope with the changing load. This 5% change during operation meant that it was not possible to specify our power supply rail over a production run at 5%. We settled on 10%, and had to negotiate with the suppliers of all the major silicon components in our system to have them specify and test all their components at a –10% rail instead of the more typical –5%.

Power Savings and the User Interface

The liquid crystal display (LCD), the digitizing tablet, and the sound output hardware all comprise what Apple terms the user interface hardware. These elements are carefully chosen and implemented for low power and ease of use. The user interface is where the careful intertwining of hardware and software in this product is most visible.

When the digitizing tablet is in use, the OS constantly scans for tablet coordinates at a rate of about 80 points per second. Using a proprietary method, the validity of a particular point is dynamically determined. This automatic validity assessment prevents bad coordinate data, caused by more than one contact on the screen surface while writing, from being sent to the higher levels of the system for processing. The area which is being digitized is also continuously monitored. If the pen strays into an area of the screen which does not require precise coordinate monitoring, the sample rate of the tablet is slowed down to about 10 points per second. These screen areas are dynamic and are automatically defined by the Newton view architecture. This allows application developers to automatically take advantage of this power saving method without even knowing that it exists. The power savings from using these techniques are significant. The tablet circuitry consumes an average of about 17mA and 10% of the ARM CPU while active at full rate. Even when the digitizer is running at full rate, careful attention to power consumption is important; this is discussed further in the section on interrupts.

Even the LCD itself is a key power saving element in the MessagePad. In a typical LCD system, there is an LCD controller which uses its own RAM, or accesses main system memory. The LCD controller than drives a high frequency clock and data stream to the row and column drivers on the LCD panel. This leads to an LCD system that can consume over 100mW of power just to display a static image to the user. In the MessagePad, the LCD frame-buffers are integrated into the LCD row and column drivers. The drive edges are also carefully controlled, preventing losses from driver cross-conduction. The end result is an LCD panel which consumes less than 5mW to display a static image.

The sound system also benefits from a combined hardware and software effort designed to save power and cost. The linear amplifier that is used to integrate and playback the eight bit digital audio stream consumes about 17mA when active. In order to avoid a large waste of power, the audio amplifier must be shut down when not in use. To do this without pops and clicks, a ramp must be generated, both in the analog hardware and in the digital audio stream.

The World of Communications

Communications is a key element of the Newton platform. The system implements most communication functions through a very flexible serial port, as well as a full type II PCMCIA rev 2.01 compatible slot. Drivers can be dynamically loaded from the serial port or the PCMCIA interface. This enables many new functions for Newton users without them having to worry about installing or configuring software. Current examples of this are the Newton MessagingCard, and the Newton PrintPack. The Newton MessagingCard is a wireless data receiver which transparently loads a driver and user interface software into the MessagePad whenever it is installed. The Newton PrintPack is a serial cable that transparently loads drivers for several hundred Centronics compatible parallel printers. As elsewhere, all our communication protocols are implemented with power in mind.

The LocalTalk protocol stack designed for Newton is extremely low power. The user interface again plays an important role in saving power. This is seen in the method used for communicating with Macintoshes. One might think that the logical approach is to do what most LocalTalk devices do, and register the Newton MessagePad on the AppleTalk network, and then be able to choose it from any Macintosh or PC on the network for remote access. We chose to reverse this process to avoid having the serial engine and the ARM from constantly having to respond to name lookups and other incoming network traffic. In our user interface, the Macintosh registers itself on the network, and then the Newton user chooses the Macintosh from the list of Macs presented.

When the network chooser is displayed to the user, for selecting printers or other network devices, it sends name lookup requests rapidly at first. Over time, the software sends fewer and fewer requests. Eventually, the serial port is closed and all network traffic ceases. This prevents the user from inadvertently consuming their battery power when there is no real work to be done on the network. When the user finally does choose a network device, the serial engine is restarted, a new node id is obtained, and the address of the chosen device is confirmed, all transparently to the user.

Interrupts, Timers and Power Consumption

Internally, our interrupt and timer engines are a critical part of our power saving strategy. The system is entirely event driven. There is no concept of a spin loop within the OS. Whenever the processor must wait for a slow peripheral, and there is not enough time to make a context switch, or there are no other tasks waiting to execute, the OS stops the ARM processor in its tracks. An example of this is the previously discussed tablet driver. There are periods where the ARM processor is waiting for the completion of an analog to digital [ADC] conversion. The converter period is approximately 17µs. This is not enough time to complete a context switch and get any useful work done. So to conserve the maximum amount of power, the ARM processor is stopped until the ADC conversion is completed. The ADC completion signal is the event that restarts the clocks to the ARM processor. It then resumes execution and obtains the data from the conversion.

The ASIC Design

As the design of the system progressed, we approached the point where we realized that the custom logic chip (the ASIC) that controlled everything outside the ARM CPU was quickly becoming larger and more expensive than the processor. Our internal challenge was to prevent the ASIC from also consuming more power than the CPU. To this end we made several design decisions that greatly increased the complexity of the system.

The ASIC uses ripple counters for all event timers. These use less power than the more common synchronous counters. All internal and external bi-directional busses are driven during all operational states. The only bus which is tristated is the PCMCIA bus, and it uses nor gates in the input pads to prevent leakage from floating signals.

All internal subsystems within the ASIC use gated clocks. Each function within the ASIC is only clocked at the rate necessary, and is only clocked when the software activates it. This caused an unanticipated problem with the design methodology that was being used. The ASIC was entirely synthesized using Synopsys from Verilog source. Many parts of the ASIC had to be manually instantiated to prevent Synopsys from doing things like adding chains of 150 inverters in an attempt to balance timing where balancing was not required

Unfortunately, gating the clocks to infrequently used sections of the ASIC didn't provide an acceptable level of power savings. The power consumption was still almost twice that of the ARM 610 processor. Other measures were also needed to ensure that the battery life would be adequate, so another design improvement was made. Whenever the ASIC idles the system under software control, the internal ASIC clock to all blocks is reduced to 1.5Mhz. This clock is then used to run all the state machines that are waiting for an event to occur.

Since we were constrained by the available ASIC tools and processes for the first generation some tradeoffs were made. We look at them as opportunities for great future enhancement of the platform.

Robust Data Storage

System robustness is also of critical importance in a machine where critical personal data is stored. It is, of course, possible for a user to remove the batteries from the system at any time. The system must be able to recover from this event without losing any of the personal data.

To support this, a database that can have its operations interrupted at any point without damage must be created. The MessagePad has a complete transactional, object-oriented database system built into it. This system makes it impossible to lose any data that has been committed to the permanent object store, unless the memory backup battery is removed as well.

The Future

Despite the lead that Apple has in this class of products, there are design and process techniques that will allow us to improve the power/performance ratio of our system by more than eight times. The most obvious improvement is moving the main power supply to 3V. This alone will give us more than a factor of two savings in power at the same perceived performance.

In order to sustain these advantages in the long term, the system has been designed to be extremely flexible and portable. The applications development environment generates byte code interpreted NewtonScript™. This processor independent environment allows Newton licensees and Apple to choose the most appropriate processor platform at any point in time and with a minimal effort get the entire Newton OS and all applications running.

As we look to the future, we see many exciting new products providing two-way wireless communication, wireless faxing and, of course, improved free form recognition. We feel that we have built a platform that will allow us to be at the forefront in this class of products for a long long time.