

# Opportunities and Obstacles in Low-Power System-Level CAD

Andrew Wolfe  
Department of Electrical Engineering  
Princeton University  
Princeton, NJ 08544  
awolfe@princeton.edu

## Abstract

A case study in low-power system-level design is presented. We detail the design of a low-power embedded system, a touchscreen interface device for a personal computer. This device is designed to operate on excess power provided by unused RS232 communication lines. We focus on the design and measurement procedures used to reduce the power requirements of this system to less than 50mW. Additionally, we highlight opportunities to use system-level design and analysis tools for low-power design and the obstacles that prevented using such tools in this design.

## 1. Introduction

Embedded system design and hardware-software codesign have been hot topics for research for the past several years. Embedded systems form an interesting area for research not only due to their strategic and economic importance, but also because they require balancing numerous competing implementation properties including size, cost, performance, power, reliability, and design time. Researchers have been developing tools to analyze and optimize these properties. These include tools such as [3] that aid in hardware synthesis and tools such as [2] that aid in software design. Some of our own work [6] has involved tools to analyze and optimize power consumption. A more comprehensive survey of power estimation techniques is available in [4].

As developers of embedded system CAD tools, we have noted that many designers are still reluctant to use CAD for embedded system designs. This presents a stark contrast to VLSI design efforts where tools dominate most of the design process. Many practicing designers have voiced the opinion that current CAD tools do not address the critical system-level issues that often dominate in an embedded system design.

Recently, we were presented with the opportunity to design and document the design of a low-power commercial product. The evolution of the design through four generations, shows how requirements and priorities change to meet customer expectations. This type of case study can be valuable to illustrate the relative importance of various problems within the design process[8]. In this case, we focus on the methods used to reduce power requirements in the design of a computer peripheral. In documenting this design, we highlight opportunities where new or existing design automation tools for embedded systems could have improved the design process. We also attempt to identify the obstacles that prevented the use of design tools, whether these were shortcomings in our design methodology or deficiencies in existing tools. Some of the early data from this design is also available in [9].

## 2. Design Requirements

The case-study system is the electronic controller for a resistive-overlay touch-sensitive sensor for a CRT-based computer display. These systems are commonly called *touchscreens* in the computer industry. When the touchscreen is integrated into a computing system and supported by application software, a user

interacts with the computer by touching objects displayed on the underlying display.

Resistive-overlay touch sensors are a relatively mature technology; however, they are still rather expensive when compared to a keyboard or mouse. The new product is a simple-to-use, very-low-cost version of a touchscreen for the home market. It attaches to the front of a computer monitor with a Velcro-based hinge and plugs into the serial port of an IBM-style personal computer[1]. In order to reduce cost and simplify installation, the device uses no external power supply. The entire system must operate on excess power supplied by unused RS232 serial communication lines. This technique has been used to power the mouse on many computers and is familiar to consumers. Unfortunately, a touchscreen is inherently much more complicated and power-hungry than a mouse. Therefore, the functional demands of this system dictate an aggressive approach to low-power design.

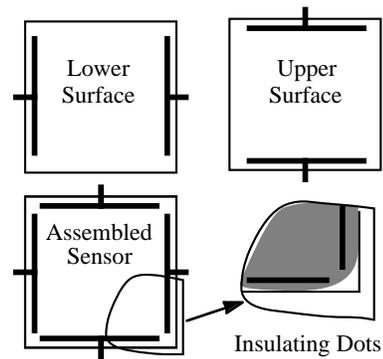


Fig. 1: Resistive-Overlay Touch Sensor.

The resistive-overlay sensor consists of two sheets of transparent plastic material coated on the inner surfaces with a transparent thin-film, uniformly resistive material (Indium-tin oxide). These two resistive surfaces are separated by insulator dots to prevent contact. Each surface includes a conductor at two ends as is shown in Fig. 1. A voltage is placed between the conductors of one surface, establishing a uniform electric field and creating a linear voltage gradient across the resistive film from one conductor to the other. Finger pressure on the sensor causes contact between the two surfaces. The passive surface acts as a probe to measure the voltage on the active surface at the point of contact. This voltage is proportional to the X coordinate of the touched location. The entire procedure is then repeated with the alternate surface driven to measure the Y coordinate of the point of contact. In practice, this procedure is preceded by a touch-detect phase where the processor determines whether or not the sensor is being touched at all.

The overall system is straightforward but it does involve a number of moderately complex tasks. The system must sequentially acquire a number of high-resolution analog measurements and interpret the results. This is accomplished by an embedded microcontroller. The processor also filters the measurements, scales the data, formats the data and transmits it to the host. Concurrently, it must accept and process commands from the host controlling calibration, flow control, diagnostics, *etc.* In

addition to this central measurement and computation function, the system must include interfaces to the sensor and the RS232 communication line.

The initial controller for this sensor was designed many years ago without regard for power requirements. The primary design goal at that time was low cost in moderate volumes, thus common NMOS and bipolar components provided the best solution. It used 3 power supplies at +5V,  $\pm 12$ V and typically consumed 2.5W. A second generation product was developed several years ago, primarily to provide a solution for handheld, battery-powered PDA-type devices. The primary goal was high-integration and single-supply operation although power was also clearly a priority. This system, called AR4000, draws approximately 200mW from a single +5V supply. The AR4000 serves as a starting point for this new very-low-power design called LP4000.

Meeting the low-power goals for the LP4000 presents several obvious challenges. A new power supply circuit must be designed to extract usable, regulated power from the extra RS232 signals. An analysis is required to determine where, why, and when the existing controller is consuming power. Known power-hungry components such as the resistive-overlay sensor and the charge-pumps for the RS232 drivers must be carefully managed at the system level. Finally, the power consumption of the processor and its peripherals must be reduced despite the fact that the AR4000 is already a low-power CMOS design.

### 3. Establishing Specifications

Most CAD methodologies are based on the premise that precise, quantitative specifications are available as the input to the design tool and that these specifications are used to guide system synthesis by bounding the design space and establishing evaluation criteria. Unfortunately, in most real-world designs including this one, precise, formal specifications do not exist. This may change as designers adopt more formal design methods and more tools become available, but for the time being it is uncommon for embedded systems designs to be formally described. Instead, the design is described by a variety of quantitative and qualitative parameters that guide the designer. In some cases these specifications relate to specific quantitative performance requirements. For example, the LP4000 must provide 10-bits of resolution along each axis. Other specifications are far less precise. For example, the LP4000 must provide adequate user response when tested with typical existing applications. The designer must explore the range of correct designs in order to establish a design point where all quantitative and qualitative design requirements can be met. In low-power design this generally means that performance must be limited in order to meet power constraints.

As such, the initial task in designing the LP4000 was to establish a reasonable initial set of specifications and constraints given the stated design goals. In addition to the given resolution requirement, standard RS232 communication at 9600 baud and an 11-byte ASCII data reporting format that is supported by existing software will be used. The electrical specifications of the sensor are also fixed and there are established cost and size goals. The development schedule and cost constraints rule out the design of any custom or semicustom chips.

The overall system performance should be similar to the earlier product which samples the sensor at 150 samples/s then extensively filters the data before reporting it to the host at 75 or 150 reports/s. Between samples the CPU powers down to save energy; thus reducing the sampling rate reduces average power consumption. Applications-based testing shows satisfactory performance if the sampling and reporting rate is reduced to 40 samples/s with improved performance up to 75 samples/s.

Finally, as a low-power system design project, it is important to characterize the power constraints. Many low-power designs are

primarily concerned with energy consumption since this determines battery life. In this case, the energy supply is unlimited but the rate of power delivery is sharply constrained. Unfortunately, this power is not supplied at a fixed voltage or current. In practice, the RS232 electrical standards for signals are not strictly followed; however, in most personal computers one of a small number of interface chips is commonly used. We characterized the current/voltage response for the two most common RS232 drivers under various loads. The output capabilities of these two chips, the Motorola MC1488 and the Maxim MAX232 are shown in Fig. 2.

The interpretation of these results depends on the intended design of the power regulation circuitry and the requirements of the system components. Two different supply voltages are currently used for most off-the-shelf CMOS components, 5V and 3.3V. In digital CMOS systems, the reduced supply voltage (3.3V) can reduce power consumption by more than 50%. Unfortunately, this system has analog signals which are measured to 10-bit (.1%) accuracy. Reducing the entire system voltage would increase the noise on these signals and thus should be avoided. Furthermore, there is still a significant price premium for 3.3V components, thus we decided to attempt to meet the power goals with 5V logic throughout. A switching power supply can provide higher efficiency than a simple linear regulator, however; they are expensive and quite noisy, affecting the quality of sensor measurements.

The combined effect of these issues is that a 5V regulated supply must be delivered by a linear regulator. The regulator drops .4V and the required isolation diodes from the signal lines drop .7V so the incoming RS232 signal must supply at least 6.1V to maintain system operation. Analysis of the RS232 driver I/V response shows that either chip can supply up to about 7mA at this voltage. Since two unused RS232 signals are available for power (RTS & DTR), the system power must be safely under 14mA.

### 4. Analysis of Existing System

Simple measurements on the existing AR4000 controller show that its power consumption exceeds the available power from the RS232 signals. In order to understand how and when this circuit consumes power we experimentally measured the power consumption using instrumentation techniques discussed in [6][7].

Fig. 3 shows the various components of the AR4000 controller. At the time of this design, the primary factor determining the hardware partitioning was the potential for reducing the product size. As many critical functions as possible were combined into a single chip with the intention that customers could integrate this system into their product with minimum overhead. Many of the functions that are not on the CPU chip can be eliminated when an OEM uses this design. An 80C552 microcontroller from Philips semiconductor [5] provides most of the functionality. This chip

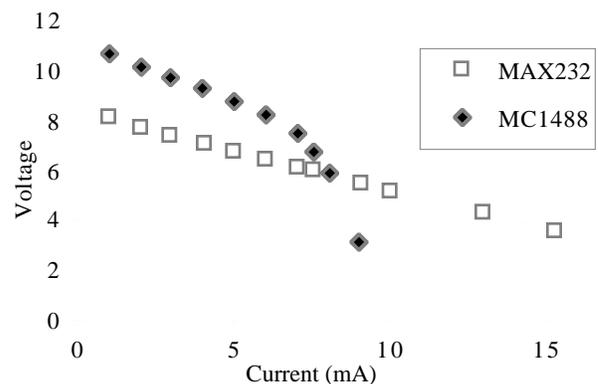


Fig. 2: I/V response of two common RS232 drivers.

includes an Intel 8051 compatible processor core, 256 bytes of RAM, a UART, a 10-bit A/D converter, multiple timers, and several digital I/O pins. An address latch (74HC573) and EPROM (27C64) provide program memory. A 74AC241 transceiver chip acts as a high-current buffer to drive the sensor. A 74HC4053 analog multiplexer selects which surface to measure. An open-drain output pin on the CPU provides the resistive load for touch detection. A Maxim MAX232 which includes an integrated charge pump to produce  $\pm 10V$ , buffers the RS232 communication signals. The processor is clocked at 11.0592MHz (1152 times the communication rate). The sampling rate for the sensor is 150 samples/s with 75 samples/s reported to the host at 9600 baud.

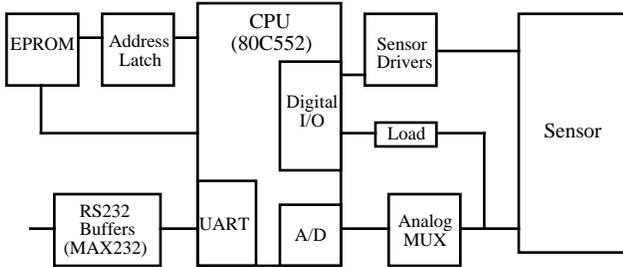


Fig. 3: AR4000 block diagram.

The system was characterized in two periodic operating modes. Every 6.7ms, a timer interrupts the processor from its *IDLE* mode (a low-power “sleep mode”). The processor drives a voltage onto the upper surface of the sensor and enables the resistive load on the lower surface. After a fixed settling time, the processor samples the voltage on the lower surface and determines if the sensor is being touched. If it is not, then the processor returns to *IDLE*. This sequence is referred to as *Standby* mode. If the sensor is being touched, then the processor must perform several additional steps. This is referred to as *Operating* mode. A voltage gradient is placed on each surface and the X and Y coordinates of the touched location are measured. This data is then filtered and scaled. Finally, the processor formats the data and transmits it to the host. The processor then powers down the sensor and returns to *IDLE* mode. Clearly this requires more power than standby operation.

	Standby	Operating
74HC4053	0.00 mA	0.00 mA
74AC241	0.00 mA	8.50 mA
74HC573	0.31 mA	2.02 mA
80C552	3.71 mA	9.67 mA
EPROM	4.81 mA	5.89 mA
MAX232	10.03 mA	10.10 mA
<b>Total of ICs</b>	<b>18.86 mA</b>	<b>36.18 mA</b>
<b>Total measured</b>	<b>19.6 mA</b>	<b>39.0 mA</b>

Fig. 4: Power measurements for the AR4000.

Fig. 4 presents the results of these power measurements for both modes. Each major component was measured as well as the total system current. Some minor discrepancies exist in the total current measurements [9]. Even allowing for some small error, several observations are clear:

- Operating mode consumes significantly more power than standby mode.
- The CPU and its memory use only about 50% of the power in operating mode.
- The DC load of the sensor (through the drivers) is a primary component of the increased power consumption during operating mode.
- The power consumption of the RS232 transceiver is large and unrelated to serial-port usage.
- A power reduction of approximately 75% is required.

Given the degree of power reduction required, it is unlikely that existing hardware-software codesign techniques such as repartitioning, logic optimization, or software optimization will be sufficient for this design. Circuit redesign and system-level power management will be required to reduce the power consumption of the sensor and communication drivers.

## 5. Low-power Redesign

The new low-power design provides an opportunity to reexamine the partitioning of the system functions into components. The new primary design goal of low-power may favor a different partitioning that the earlier design which valued flexibility and low chip count more heavily. Furthermore, the new system requires power management functions that were not required in earlier systems.

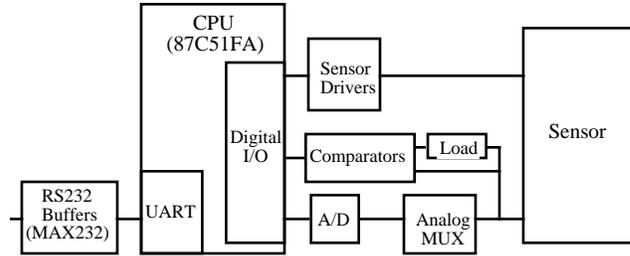


Fig. 5: LP4000 Initial Design Block Diagram.

A block diagram of the repartitioned system is shown in Fig. 5. The primary functions of the new LP4000 controller are:

- Power regulation and management
- Computation and control
- Sensor interface (drivers and quantization)
- Communications

The partitioning of these functions into chips is primarily dictated by the availability of low-power solutions off-the-shelf. The processor and memory draw about 17.5mA in the AR4000 system. Only processors that are binary compatible with the 80C552 were considered. The external program memory is convenient and flexible, but clearly consumes power. A processor with on-chip program memory is required. Philips supplies a masked ROM version of the 80C552 called the 83C552. This would provide a pin-compatible solution; however, it is risky to use a sole-source masked ROM microcontroller. The simpler 80C52 processors are available from several manufacturers. This chip includes all of the functions of the 80C552 that are used in this design except for the A/D converter and the open-drain outputs. It is not initially obvious why moving to a less integrated solution is advantageous when power consumption is critical, however; the 80C52 processor uses significantly less power than the 83C552. We believe that the reason for this is that manufacturers have been more aggressive at moving this component to newer process technologies. This is partially driven by the fact that this is the highest volume part in this processor family and thus profits most from die-size reduction. It is also likely that manufacturers hesitate to change processes for components that include analog functions but are anxious to migrate all-digital components. The effect is that the simpler, all digital components are currently manufactured in a more aggressive, lower-power process.

An 80C52 compatible processor, the Intel 87C51FA has been used for development of the LP4000. This determines the hardware partitioning for the remainder of the system. An external, serial, 10-bit A/D converter is used for sensor measurement. An LM393A dual comparator initially was used to provide touch detection and an open-drain output for the touch-detect load, however; it was replaced by a slightly more expensive CMOS equivalent, the TLC352, early in the development. The 74AC241

drivers and 74HC4053 multiplexers are retained. A low-power version of the MAX232 is selected, the MAX220. The software is then modified to support the new peripheral configuration. Current measurements for the new design are shown in Fig. 6 for both the original sampling rate and a reduced rate of 50 samples/s. This shows a significant improvement over the AR4000 but still exceeds the new specifications.

	Standby	Operating
150 samples/s	12.25 mA	21.94 mA
50 samples/s	11.70 mA	15.33 mA

Fig. 6: Power measurements for the initial LP4000 prototype.

The repartitioning of functionality for the LP4000 was performed without the benefit of any CAD tools. This is unfortunate, as it really only allowed the exploration of one system configuration. A far better solution would have been to use some type of system-level power modeling tool that would have allowed many different solutions to be compared. We do not know of any tools that are capable of predicting the power consumption of even a single system of this type, much less compare many systems. Such a tool would need to provide some framework for determining the total power of an embedded system based on a set of components and their interactions. In order to develop such a tool, we would need low-level models for the power consumption of individual components. In many cases, these are not currently available. Two clear problems are that detailed power models are not available for many off-the-shelf analog components and that there are no tools that model the interactions between software and hardware in the digital domain. Many designers have expressed a desire for these types of exploratory tools early in the design process, however; we are still a long way from providing useful systems.

The other tool that would have been beneficial at this stage in the design would have been an efficient, compatible, retargetable compiler. The code for this system was written in the PLM-51 language, a special embedded systems language for the 8051 family, and in 8051 assembly language. This restricted the choice of processors for the design and eliminated many lower-power alternatives. Even if the specification language had not been processor specific, embedded code written in portable languages such as C generally contain a great deal of processor-specific code. Retargetable compilers that can produce fast, small code from a portable specification, mapping requirements to specific processor resources rather than requiring programmers to deal with architecture-specific features, could prove to be an extremely powerful development tool. Binary to binary translation tools that can efficiently migrate a program from one processor to another would be equally effective.

## 6. Design Refinement

The system-level changes, repartitioning and revising the sampling rate, significantly reduced power but not by enough to meet the specifications. Additional improvements are required. Preferably, without further reducing performance. Another breakdown of current flow (Fig. 7) identifies which components still consume power. This analysis shows that the CPU, RS232 drivers, and voltage regulator are the primary consumers of power. A series of design refinements are required to reduce the power consumption of these components.

### 6.1. RS232 drivers

The MAX220 had been selected because it was widely advertised as a .5mA component; however, in this system the measured power consumption is much higher. Merely being connected to the host draws an additional 3-4mA whether or not any data is transmitted. One option would be to power down this chip when it is not being used; however, communications from the

host are unscheduled. The solution required a more sophisticated and of course more expensive transceiver chip, the LTC1384 from Linear Technologies. This transceiver includes integrated power management that can shut down the charge pumps and disable the transmitter while keeping the receivers enabled. In this mode the chip draws 35 $\mu$ A. When enabled it uses 4.77mA, similar to the MAX220. With software added to disable this chip when the processor's transmit buffer is empty, the LTC1384 requires only 35 $\mu$ A in standby and 2.97mA operating, reducing system power to 6.90mA standby and 13.23mA operating. This meets the required specifications, but leaves little margin for component variation.

	Standby	Operating
74HC4053	0.00 mA	0.00 mA
74AC241	0.00 mA	1.39 mA
A/D (TLC1549)	0.52 mA	0.52 mA
87C51FA	4.12 mA	6.32 mA
Comparator (TLC352)	0.13 mA	0.12 mA
MAX220	4.87 mA	4.85mA
Regulator	1.84 mA	1.84 mA
<b>Total of ICs</b>	11.48 mA	15.04 mA
<b>Total measured</b>	11.70 mA	15.33 mA

Fig. 7: Power breakdown for the LP4000 prototype.

### 6.2. Processor

Most of the remaining power is consumed by the processor. Reducing processor power is believed to be a straightforward hardware-software codesign problem. The traditional model of power consumption in CMOS microprocessors is that power is proportional to  $f \times \%T$  where  $f$  is the clock frequency and  $\%T$  is the average percentage of devices that switch each clock. This processor operates in two modes, IDLE mode where  $\%T$  is very low and normal mode where  $\%T$  is quite high. Each sampling period (20ms) a fixed computation is performed in normal mode then the processor waits in IDLE mode. The energy consumed by this computation is essentially constant since it requires a fixed number of clocks. The remaining clocks, those that occur during IDLE mode are overhead. Therefore, the time spent in IDLE mode should be eliminated.

	3.684MHz		11.059 MHz	
	Standby	Operating	Standby	Operating
87C51FA	2.27 mA	5.97 mA	4.12 mA	6.32 mA
74AC241	0.00 mA	3.52 mA	0.00 mA	1.39 mA
<b>Total meas.</b>	5.03 mA	15.5 mA	6.90 mA	13.23 mA

Fig. 8: Effect of reduced clock speed.

Using this standard model, we analyzed the software requirements during each sampling period. This was measured using an in-circuit emulator but could have been established using a cycle-level timing simulator if the actual hardware was not yet available. The computation per sample requires approximately 5500 machine cycles (66,000 clocks). This requires a minimum clock rate of 3.3MHz to complete in 20ms. The closest value that will permit the UART to operate at standard rates is 3.684MHz, thus this value was selected. All programmed timing delays were adjusted and the power consumption was measured once again. These results are shown in Fig. 8. The power required by the processor dropped as expected. The standby power, where much time is in IDLE mode, is especially improved. Unfortunately, the overall power increases significantly in operating mode. This is primarily due to a large increase in the sensor driver power.

This unexpected result illustrates two weaknesses in the commonly used power model. The common assumption is that power consumption is proportional to clock speed in digital CMOS. As found here, when there is essentially a fixed amount of computation to be performed, the number of clocks required to

	3.684MHz		11.059 MHz		22.118 MHz	
	Standby	Operating	Standby	Operating	Standby	Operating
87C51FA	1.82 mA	5.97 mA	3.54 mA	5.71 mA	6.14 mA	7.84 mA
74AC241	0.00 mA	3.66 mA	0.00 mA	1.27 mA	0.00 mA	0.71 mA
<b>Total: 2 components</b>	1.82 mA	9.63 mA	3.54mA	6.98 mA	6.14 mA	8.55 mA

Fig. 9: Effect of increased clock speed.

perform this computation is fixed regardless of clock speed, thus the energy requirement is essentially fixed. For a periodic computation like this one, this means that power reduction as a function of slowing the clock is highly sublinear. The traditional model also assumes that the load on the system is purely capacitive. In fact, this circuit, like many others, has resistive loads as well. These include the sensor, the touch-detect load, and the transmitter load. By slowing down processor operations, such as communication with the A/D converter or testing for an empty transmit buffer, these DC loads are driven for a longer time. This can increase power consumption. In this case, standby power is reduced while operating power is increased.

The obvious follow-up experiment is to determine whether speeding up the clock can actually reduce power consumption further in operating mode. In order to perform this experiment, we doubled the clock speed and repeated the measurements. We used a slightly different processor for just this test in order to permit higher speed operation. In fact these tests, shown in Fig. 9, were performed much later than the original design and reflect several minor software and hardware changes. Despite some minor variations, the results are fascinating. The original clock speed is more efficient than either higher or lower clock speeds. The combination of increased IDLE mode current and the fact that some portions of the code, such as timing loops, do not speed up when the clock is increased contribute to the high power consumption at 22MHz. One would assume from this data, that there is an optimal clocking rate, however, determining such without tools is very difficult. Each tested speed requires many timing-related modifications to the program. A tool to solve this type of problem would be very valuable. At a minimum, this requires expanding the scope of existing power modeling tools to consider DC power effects, fixed-time software delays, and variable-time computations in order to predict the power consumption of real systems.

Based on the power breakdown, some minor modifications were made to the hardware. The LM317LZ regulator requires an adjustment current of almost 2mA. Newer *micropower* regulators are available that are designed for increased efficiency at a somewhat higher cost. This appeared to be a good tradeoff. The LT1121CZ-5 regulator from Linear Technologies was substituted for the LM317LZ to reduce this adjustment bias current. This reduced current flow to 3.11mA in standby and 13.02mA operating. A further observation that the LTC1384 could reliably operate at 9600 baud (a small fraction of its specified peak rate) with smaller charge-pump capacitors reduced system current to 3.07mA in standby and 12.77mA operating.<sup>1</sup>

### 6.3. Design Problems

Following this power reduction effort, the system was operational on the vast majority of host systems; however, it would often lock up when power was first applied. The problem was that all of the power management was at least partly implemented in software. This software was not active immediately at startup; therefore, the system consumed too much power initially and never reached a valid supply voltage. This was a critical flaw in the

<sup>1</sup> Despite the advantages of a higher clock for the tested configuration, the 3.684MHz clock was retained for these experiments and those in the next section. The clock-speed experiments were repeated several more times throughout the design process and the results still indicated that 11.059MHz clock provided the best solution.

hardware-software partitioning. Some power management would need to be implemented in hardware alone to control the current demands until the system was stable and the software had initialized. The power switching circuit in Fig. 10 was added to assist during startup. Power is not supplied to the main circuit until after the reserve capacitor is charged and the regulator is stable at 5V.

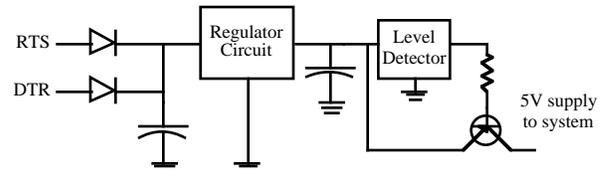


Fig. 10: Revised power-up circuit.

In practice, this was an extremely difficult type of problem to analyze and would have been an even more difficult problem to predict. This is an example of the type of problem where tools are particularly effective. Analytical solutions are often reasonably accurate for steady-state operation, but boundary conditions, like startup, are difficult to predict without simulation. Some type of system level modeling tool would have been very valuable in detecting and identifying this problem, however; it becomes clear that the development of a tool is not the primary obstacle. The primary issue is that detailed models are not available for the individual off-the-shelf components like the voltage regulator or the RS232 driver. Without accurate models supplied by either the tool vendor or the component vendor, tools are useless. It was also a challenge to design a reliable revised circuit. In this case, existing tools like SPICE would have been adequate if the component models had been available.

### 6.4. Beta Test Results

Given the final prototype design including the extra power management hardware, the system uses 3.5mA standby/12.6mA operating. Since operating power appears to be more critical than standby power, the decision to slow down the clock appears to have been incorrect. Restoring the clock speed to 11.059MHz increases standby current to 5.45mA but decreases operating current to 11.01mA. Several samples confirm that these are typical values.

The final design effort involved vendor qualification. The CPU is the most critical component in terms of power; therefore, several vendor's compatible chips were tested. The Philips 87C52 was selected for initial production. Using this chip, the system draws 4.0mA standby and 9.5mA operating. These are well within the design goals. The approximate distribution of power among the components and the improvement from the older AR4000 design is shown in Fig. 12 along with later data.

Several hundred of the systems were produced and sent to customers in a beta testing program. The majority of systems were highly successful, operating reliably and with satisfactory performance. Unfortunately, approximately 5% of the systems seldom or never worked on particular computers. It appeared that it was the particular computer that caused some incompatibility rather than some variation in this product. We asked several customers to send us the problematic systems and discovered that all were using non-standard RS232 drivers. In fact these computers all used RS232 drivers that had been combined into some larger system I/O ASIC. We characterized these drivers as

shown in Fig. 11 and discovered that they supply far less current than the initial components we had tested.

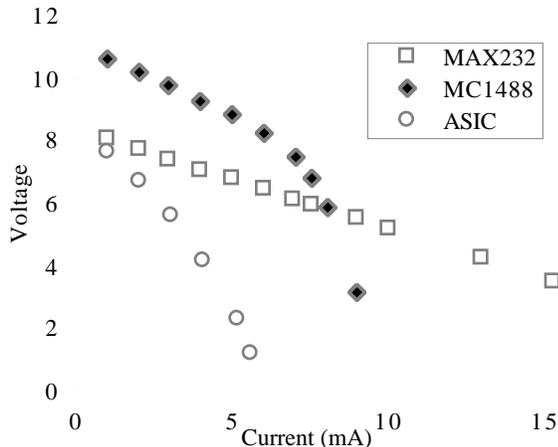


Fig. 11: Additional RS232 Driver Data.

## 7. Further Improvements

Following the beta test, several minor improvements to the power circuits improved reliability and reduced power by removing the bipolar transistor from the system and adding additional hysteresis to the reset circuit. Despite meeting the original goals, there would be great additional benefit to reducing the power further so that the computers that would not operate in the beta test could also be used. This would require reducing the operating current to less than about 6.5mA. It was determined that in order to accomplish this, the original design specifications would need to be revised. This was done in 3 major areas:

- RS232 activity was reduced. This was done by doubling the communications baud rate to 19200 bps and by reformatting the data from an 11-byte ASCII string to a 3-byte binary format. This reduces the active time of the RS232 drivers by about 86% but it required rewriting the device drivers for the host computer.
- The sensor drive voltage was reduced by adding resistors in line with the sensor. This reduces the S/N ratio on these measurements by about 1 bit.
- Some compute intensive functions such as scaling and calibration of data were moved from this system to the driver on the host system.

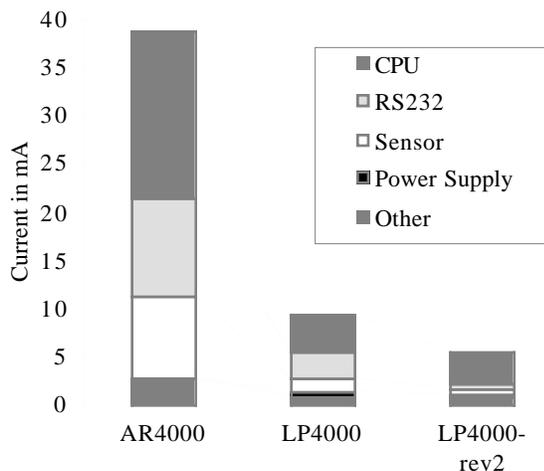


Fig. 12: Final Power Reduction.

The combined effect of these changes was an additional 35% savings in operating power from the beta test units. This includes

an 8.8% overall savings due to CPU power, a 5.5% savings due to sensor power, and a 20.8% savings due to communications power. The final system uses 3.59mA in standby and 5.61mA in operating mode. This represents an 86% reduction in power from the original AR4000 design. Depending on the characteristics of the host RS232 driver, this represents a total power consumption of around 35-50mW for the total system. Fig. 12 shows the final breakdown of power savings.

## 8. Conclusions

The primary goal of documenting and investigating this design was to discover issues in system-level design of low-power system that have not been properly addressed by the research community. In this regard, there have been several beneficial observations.

- Analog components often dominate low-power design decisions. These include sensors and actuators, communications, and power supplies. Few tools are available to manage the design of these types of components.
- Modeling the interactions between design domains is critical. Problems occur at the boundaries between analog and digital components. Also the interaction of hardware and software caused failures in this design. Boundary conditions are particularly problematic without tools.
- Tools are useless without accurate component models.
- Partitioning is often dominated by component availability.
- Switching activity models are inadequate for power modeling.
- Simple tools, such as circuit simulators, compilers, and software timing simulators, can be very valuable if well supported by component models.

These validate much of the current research into embedded system design tools and methodologies; however, they also indicate that in many cases designers need better ways to look at the big picture. Design optimization tools that optimize software or digital logic are useful during the design process, but designers are desperately in need of exploratory tools that permit system level simulation and analysis and synthesis tools that map specifications to existing components.

## 9. References

- [1] "Cyberscope", Newsweek, v. CXXV, no. 10, p. 12, March 6, 1995.
- [2] J. Gong, D. Gajski, and S. Narayan, "Software Estimation from Executable Specifications," *Journal of Computer and Software Engineering*, v. 2, no. 3, pp. 239-258, 1994.
- [3] E. D. Lagnese and D. E. Thomas, "Architectural Partitioning for System Level Synthesis of Integrated Circuits," *IEEE Transactions on Computer Aided Design*, v. 10, July 1991.
- [4] F. N. Najm, "A Survey of Power Estimation Techniques in VLSI Circuits," *IEEE Transactions on VLSI Systems*, v. 2, no. 4, pp. 446-455, Dec. 1994.
- [5] Philips Components, *Data Handbook: 80C51 and Derivative Microcontrollers*, 1990.
- [6] V. Tiwari., S. Malik, and A. Wolfe, "Compilation Techniques for Low Energy: An Overview," in *proc. IEEE Solid-State Circuits Council 1994 Symposium on Low-Power Electronics*, 1994.
- [7] V. Tiwari., S. Malik, and A. Wolfe, "Power Analysis of Embedded Software: A First Step Towards Software Power Minimization," *IEEE Transactions on VLSI Systems*, v. 2, no. 4, pp. 437-445, Dec. 1994.
- [8] W. Wolf, A. Wolfe, S. Chinatti, R. Koshy, G. Slater, and S. Sun, "TigerSwitch: A Case Study in Embedded Computing System Design," in *proc. International Workshop on Hardware-Software Codesign*, IEEE Computer Society Press, 1994.
- [9] A. Wolfe, "A Case Study in Low-Power System-Level Design", in *proc. of ICCD '95*, pp. 332-338, October, 1995.