

# A Case Study in Low-Power System-Level Design

Andrew Wolfe  
Department of Electrical Engineering  
Princeton University

## Abstract

A case study in low-power system-level design is presented. We detail the design of a typical low-power embedded system, a touchscreen interface device for a personal computer. This device is designed to operate only 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. Furthermore, we highlight opportunities to use system-level design and analysis tools for low-power design. Finally, we identify key issues in low-power system design that are not currently being explored by the design automation community.

## 1. Introduction

Embedded system design and hardware-software codesign have been hot topics for research for the past several years. A huge number of new embedded systems are designed each year. They 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].

Although we would like to believe that design tools are primarily motivated by the requirements of users, in fact they often reflect solutions to the problems with which tool designers are comfortable. It is useful to examine the design process from the designer's perspective and to consider to what degree the current approach to design automation for embedded systems is useful and whether it is moving in the right direction. Recently, we were presented with the opportunity to design and document the design of a low-power commercial product. 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. We search for points in the design process where existing embedded systems tools can be useful. More importantly we highlight opportunities where new types of design automation tools for embedded systems can improve the design process.

## 2. Design Requirements

The system under consideration is the electronic control system 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 properly integrated into a computing system and supported by application software, a user can interact with the computer by touching objects displayed on the underlying CRT screen.

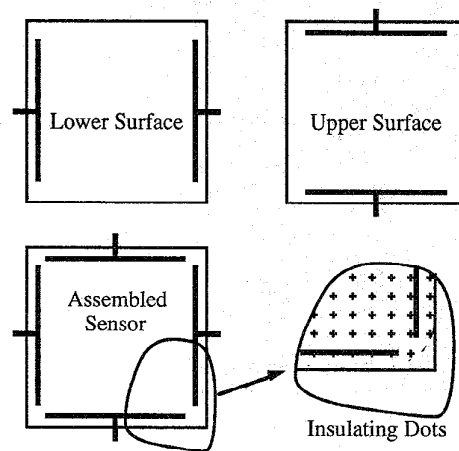


Figure 1: Resistive-Overlay Touch Sensor.

Touchscreen systems have been available for many years and 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 user's 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.

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 Figure 1. A voltage is placed between the conductors of one surface. This establishes a uniform electric field and creates 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 non-driven surface acts as a probe to measure the voltage on the driven surface at the point of contact. This voltage is proportional to the X coordinate of the touched location. This can be quantized with an A/D converter and reported to the host computer system. 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 upper surface is driven to a constant voltage rather than a gradient and a resistive load is placed between the lower surface and ground. If the sensor is being touched, a current will flow through this resistor and the X and Y coordinate measurements can be started.

The overall system is reasonably 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. It used 3 power supplies at +5V,  $\pm 12V$  and typically consumed 2.5W. The introduction of smaller, battery-powered systems such as laptop computers and personal organizers motivated a more integrated, low-power redesign several years ago. 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

In designing CAD tools for embedded systems, researchers often discuss specifications. The implication is that precise, quantitative specifications should be 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 of a complete formal specification, 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 is required to 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. Several specifications are derived directly from the functional specifications of the product. In addition to the resolution requirement, we know that we need to support standard RS232 communication at 9600 baud and that an existing 11-byte ASCII data reporting format that is well 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 is expected to be similar to the earlier AR4000 product. That system samples the sensor at approximately 150 samples/s then extensively filters the data before reporting it to the host. Between samples the CPU powers down to save energy. Reducing this sampling rate could significantly reduce average power consumption. The 150 samples/s rate had been established to support handwriting recognition applications in hand-held devices. This is not a common application in the LP4000's intended market. In fact, feedback from existing users shows that other than in handwriting-recognition applications, most users reduce the reported data rate to 75 samples/s (825cps) in order to allow the use of 9600 baud (960cps) communications. Applications-based testing shows satisfactory performance if the sampling and reporting rate is reduced to 40 samples/s with continuous improvements in performance up to 75 samples/s. This defines the design space for this parameter.

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 Figure 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. A mixed-voltage design is too costly for this type of product, thus we decided to attempt to meet the power goals with 5V logic throughout.

The option is also available to choose between a linear voltage regulator or a switching power supply. The linear regulator reduces the supply voltage to a set value, dissipating excess power in the form of heat. The switching regulator stores energy and delivers it to the system at a fixed voltage. Therefore, when supply voltage is being reduced as in this system, much of the excess power available from the drop in voltage can be supplied to the system as additional current rather than dissipated as heat. The improved efficiency of a switching regulator is attractive; however, they are significantly more complex and costly than a linear regulator. Also, switching regulators are quite noisy and affect the quality of the sensor measurements. For these reasons, a linear regulator is preferable if the other goals can be achieved.

The combined effect of these decisions 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

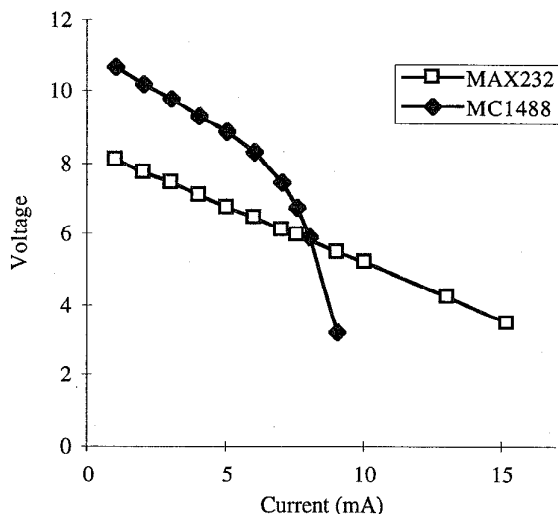


Figure 2: I/V response of two common RS232 drivers.

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. If power consumption can be further reduced, it is likely that the system will operate on a wider range on non-standard computers.

#### 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 an instrumentation technique discussed in [6]. This technique involves separately instrumenting the power supply pins for each component using a simple dual-slope, integrating digital ammeter. The current flow can thus be measured for a repetitive, periodic task. Further experiments in [7] show that when this period is short as compared to the ammeter sampling time, this method is reasonably accurate and that the results of various measurements can be linearly combined.

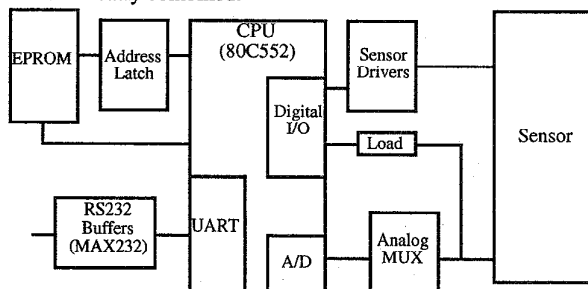


Figure 3: AR4000 block diagram.

Figure 3 shows the various components of the AR4000 controller. An 80C552 microcontroller from Philips semiconductor [5] provides most of the functionality. This chip includes an Intel 8051 compatible processor core, 256 bytes of RAM, a UART for serial communications, 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 digital 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.

The system was measured 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 mode. This 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>

**Figure 4: Power measurements for the AR4000.**

Figure 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. Some of this is due to leakage currents through capacitors, *etc.* that are not measured. Additional error may be caused by the fact that the CPU chip uses PMOS pull-ups on inputs that can source current back into the main  $V_{DD}$  net around the power pin. In fact, the CPU chip works perfectly well in this circuit without the power pin connected. Even allowing for some small error, several observations are clear:

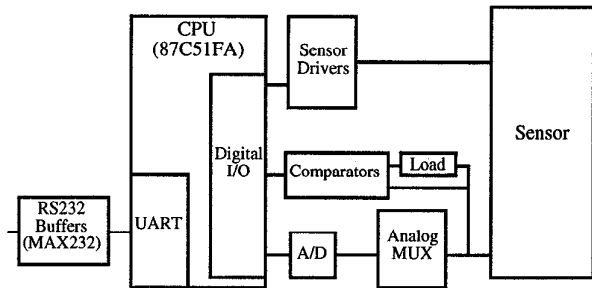
- Operating mode consumes significantly more power than standby mode.
- The CPU and its memory use only about 50% of the power in operating mode.
- The DC load of the sensor (through the drivers) is a primary component of the increased power consumption during operating mode.
- The power consumption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

Given this data, it is clear that existing hardware-software codesign techniques such as repartitioning, logic optimization, or software optimization, will be insufficient 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

### 5.1. Initial Low-power Design

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.

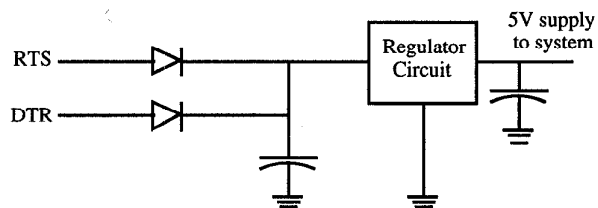


**Figure 5: LP4000 Initial Design Block Diagram.**

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

- Power regulation and management
- Computation and control
- Sensor Interface (Drivers and digitization)
- Communications

The partitioning of these functions into chips is primarily dictated by the availability of low-power solutions off-the-shelf. The power regulation circuit is shown in Figure 6. Since the voltages on the RS232 signal lines, RTS and DTR, can potentially change during operation, diodes are required to prevent reverse current flow. A moderately large capacitor is included to store unused energy. This permits peak currents in excess of the supply current; yet, the capacitor is small enough that it does not require a long time to initially charge. A small linear regulator, the LM317LZ, reduces the supply voltage to +5V.



**Figure 6: Power circuitry.**

The processor and memory draw about 17.5mA in the AR4000 system. Lower power processors are available, but it would be very difficult to predict the power consumption of any other processor given the same computing task. To do so would require rewriting all of the application software. The time and

expense of porting the software to another architecture was deemed unacceptable. This restricts the choice of processors to those that are binary compatible with the 80C552. The external program memory is convenient and flexible, but clearly consumes power. A processor with on-chip program memory is required. EPROM-based microcontrollers are too expensive for production use so a model that is available with a masked ROM is required. Philips supplies masked ROM and EPROM versions of the 80C552 called the 83C552/87C552. This would provide a completely pin-compatible solution; however, it is risky to use a sole-source masked ROM microcontroller. The simpler 80C52/87C52 processors are available from several manufacturers and in many pin-compatible derivatives. 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. Additionally, it requires less power than the 80C552. A compatible version, 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 provides touch detection and an open-drain output for the touch-detect load. The existing 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 Figure 7. This shows a significant improvement over the AR4000 but still exceeds the new specifications. The bipolar LM393A was replaced by the slightly more expensive CMOS equivalent, the TLC352. This only slightly reduced power consumption.

	Standby	Operating
with LM393A	12.73 mA	22.43 mA
with TLC352	12.25 mA	21.94 mA

Figure 7: Power measurements for the initial LP4000 prototype.

Since the power consumption exceeds the allowable limit by a significant amount, major system-level changes in the power management strategy may be required. Reducing the sampling rate would allow the processor to spend more time in idle mode and would reduce the amount of time that the sensor is driven. The sampling rate was reduced to 50 samples/s from 150 samples/s and the data reporting rate was reduced to 50 samples/s from 75 samples/s. Power consumption dropped noticeably as shown in Figure 8.

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

Figure 8: Power measurements at two sampling rates.

## 5.2. Design Refinement

The system-level changes, repartitioning and revising the sampling rate, significantly reduced power but not enough to

meet the specifications. Additional improvements would be required. Preferably, this can be done without further reducing performance. Another breakdown of current flow would identify which components were still consuming power. Figure 9 gives the results of these measurements. 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.

### 5.2.1. 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. In this case, the culprit is the host system. 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 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 only draws 35µA. When enabled it uses 4.77mA, similar to the MAX220. With software support added to disable this chip when the processor's transmit buffer is empty, the LTC1384 requires only 35µA in standby and 2.97mA operating. This reduces system power to 6.90mA standby and 13.23mA operating. This meets the required specifications, but leaves little margin for component variation.

### 5.2.2. Processor

Most of the remaining power is consumed by the processor. Reducing the processor power is a relatively 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

	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>	<b>11.48 mA</b>	<b>15.04 mA</b>
<b>Total measured</b>	<b>11.70 mA</b>	<b>15.33 mA</b>

Figure 9: Power breakdown for the LP4000 prototype.

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

Figure 10: Effect of reduced clock speed.

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.

Using this standard model, we analyzed the software requirements during each sampling period. 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 time constants were adjusted where possible and the power consumption was measured once again. These results are shown in Figure 10. The power required by the processor dropped as expected. The standby power, where much time is spent in IDLE mode, is especially improved. Unfortunately, the 74AC241 power increases greatly in operating mode. An additional .5mA is drawn by other components as well. This unexpected result illustrates a weakness in the commonly used power model. This model 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.

### 5.2.3. Voltage Regulator

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. 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. This reduced current flow to 3.07mA in standby and 12.77mA operating.

### 5.3. 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 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 Figure 11 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. This provides ample reserves to achieve stable operation.

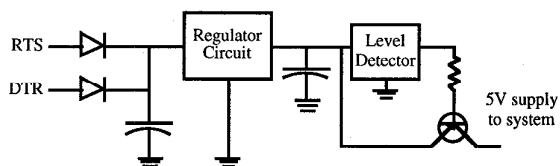


Figure 11: Revised power-up circuit.

## 5.4. Final Analysis

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 is shown in Figure 12.

## 6. Further Improvements

The LP4000 controller meets its stated design goals. It provides good performance without an external power supply. Although the design is operational, there are several possibilities for further improvement. A masked ROM CPU will be used for volume production. It is likely that this is slightly less power hungry than the EPROM-based version. It may also be possible to improve the power-switching circuit. The base current into the PNP transistor is about .5mA. A MOSFET based solution would not need to draw that current. Finally, it may be possible to make the application itself adapt to variations in available power. The sampling rate could be increased when there is reserve power and decreased when power is limited. Either a

reserve-power sensing circuit could be added or the processor could attempt to estimate reserve power based on its own usage patterns.

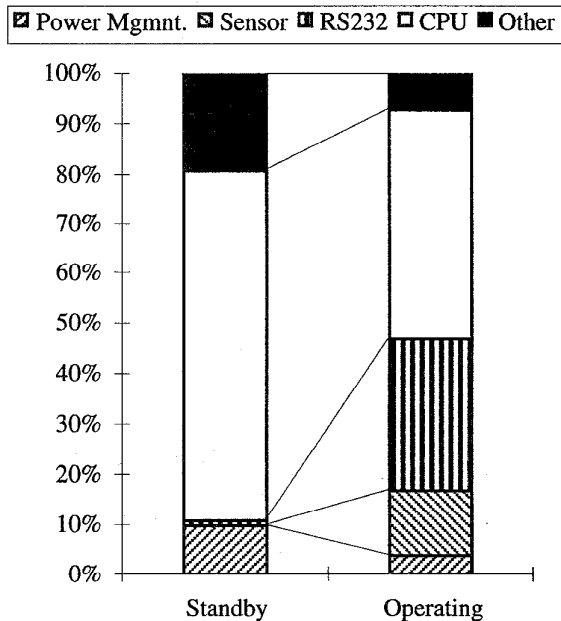


Figure 12: Power Distribution.

## 7. 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.

- Precise system specifications are often not provided at the beginning of the design cycle. Exploring specifications and requirements is often a part of the design process. Tools must be flexible enough to allow exploration of partially specified designs.
- CPU selection is a critical part of the hardware-software partitioning of the design but it is often limited by software compatibility issues. Portable high-level-language compilers and binary to binary translation tools could reduce the pressure to retain the same architecture from design to design.
- System-level power management is the most important, but most difficult step in low-power design. Traditional digital logic only represents a fraction of the hardware in many embedded systems. Tools that allow every component in the system to be modeled and assist the designer in exploring power management strategies would be valuable.

- Switching activity models are often inadequate for power estimation on real embedded systems. Real systems often consume DC power as well, particularly at interfaces. In these cases, reducing clock speed can increase power consumption.
- Current measurement techniques, although seemingly crude, can provide crucial feedback to the design process.
- Hardware partitioning is often dictated by component availability. Tools that partition hardware at the gate level are impractical in such a constrained design environment.
- Hardware/software partitioning often depends on boundary conditions such as startup. Tools that use approximations of hardware and software functionality may miss these critical design constraints.

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.

## 8. 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.