POWER EVALUATION OF A HANDHELD COMPUTER

IN A COMPREHENSIVE STUDY USING THE ITSY POCKET COMPUTER, THE AUTHORS MEASURE BOTH TOTAL SYSTEM POWER AND POWER DISSIPATED BY INDIVIDUAL SUBCIRCUITS FOR REPRESENTATIVE WORKLOADS. THE RESULTS SUGGEST POSSIBLE LOW-POWER DESIGN OPTIMIZATIONS AND POWER MANAGEMENT STRATEGIES.

Marc A. Viredaz
Hewlett-Packard

Deborah A. Wallach
Google

The computing requirements of battery-powered systems are increasing faster than battery capacity. Because large, heavy batteries are unacceptable for handheld computers, low-power design and power management are crucial for these systems. One of the main obstacles limiting these techniques is the lack of a detailed analysis of system power under representative workloads. With the understanding provided by such an analysis, device designers can improve the power consumption of next-generation systems and optimally manage the fixed amount of energy provided by the battery.

This article presents a comprehensive study of a handheld device's power consumption. We measured the total power as well as the power dissipated by six subcircuits. Using this information and running the same benchmarks on different configurations of the same hardware, we divided power consumption into nine domains, such as the processor core, dynamic RAM (DRAM), liquid crystal display (LCD), speaker, and power supply.

As the subject of our study, we used the Itsy pocket computer, developed at Compaq's Western Research Laboratory. The Itsy project's aim was to develop a flexible research platform for pocket computing.[1] Itsy developers built several versions of the hardware, all based on the StrongARM SA-1100 processor.[2] Our study concentrates on Itsy version 2.4, a complete handheld computer measuring 118 mm × 65 mm × 16 mm and weighing only 130 grams. Because one of the Itsy project's goals was to support research in power evaluation and management, several features in the system make it an ideal platform for our study.

Although it is easy to measure a system's power with expensive, highly sophisticated instruments, moderate-cost instruments (accessible in any commercial or academic environment) usually suffice. For meaningful results, however, an understanding of the limitations of the chosen instruments and methodology is necessary. Our study devotes special attention to this issue. (See the "Related work" sidebar for a summary of other studies.)

Methodology

To conduct a set of detailed measurements, we chose a basic strategy and then had to make a series of tradeoffs. We discuss the most important tradeoffs here.

Strategy

Because battery lifetime measurements are extremely time consuming, they are rarely performed. For example, in our study, the

Published by the IEEE Computer Society

fully automated run of the benchmark suite lasted two months. However, we considered a battery lifetime analysis necessary for several reasons. First, a battery-powered device's power consumption is not constant; rather, it varies as the battery voltage decreases. In addition, battery capacity varies with the load placed on the battery. Even with realistic loads, the capacity can decrease by 20 percent to 40 percent.[3]

Ideally, we would have liked to measure the power breakdown over the full battery lifetime for each benchmark. However, the available test equipment prevented us from running such experiments simultaneously on several units, and running them sequentially on several units would have taken too much time.

To avoid these problems, we conducted the Itsy power analysis in two phases. In the first phase, we measured battery lifetime, while continuously monitoring total power consumption.[4] Later, we replaced the battery with a power supply, determining its value from the first phase's data. These power-supply-based tests yielded an evaluation of the power breakdown among Itsy's building blocks. We conducted a careful error analysis of all measurements.

Our study did not address long-term battery aging. However, short-term battery aging—the aging that occurs during running of the full benchmark suite—is unavoidable, given our test strategy. In a previous study, using the same battery and a very similar system, we quantified this term and found a decrease in battery capacity of 1.0 percent to 2.9 percent.[5] This is smaller than typical system-to-system variations (up to 10 percent in the present study). Therefore, within these tolerances, the battery lifetime figures presented here are representative of relatively new batteries.

From the first-phase data, we can infer the effective battery voltage—the voltage at which power consumption is identical to average power across the battery lifetime. This voltage varies slightly with the load. Itsy's lithium-ion battery's voltage is 4.1 V when fully charged. Its effective voltage is between 3.75 V and 3.85 V in sleep mode, around 3.80 V while the processor is idle (idle mode), and between 3.60 V and 3.80 V while the processor is doing useful work (run mode). The average effective battery voltage across all the experiments was 3.77 V.

For the second phase, we measured the power breakdown with the battery replaced by a power supply set to 3.75 V. This voltage is close to the effective battery voltage for all benchmarks, with a slight bias toward run-mode benchmarks. Fixing the supply voltage near the effective battery voltage should yield a power breakdown similar to that of the complete battery lifetime. To minimize run-to-run variation, we ran each benchmark for the greater of 20 minutes or 25 iterations.

Related work

Jacob R. Lorch and Alan Jay Smith presented an extensive power analysis for several laptops in the Macintosh Powerbook Duo family.[1] Because the Duos were not designed for such a study, the authors had to rely on differential measurements and estimates from an Apple Computer engineer to determine such important figures as processor and display power.

Other researchers have published less comprehensive studies. Although it is not truly a handheld, the Linux Advanced Radio Terminal (LART) embedded device is also based on the StrongARM SA-1100 processor and thus has power characteristics similar to Itsy's. Although instrumented, the LART includes only three sense resistors, so researchers Johan Pouwelse, Koen Langendoen, and Henk Sips could separate the system into only three power domains and thus could take only limited direct measurements.[2,3]

Researchers Tajana Šimunić et al. published a limited power breakdown of the SmartBadge, another device using the StrongARM SA-1100.[4] Keith I. Farkas et al. developed power usage profiles for an earlier Itsy by running microbenchmarks and measuring overall system power.[5] They estimated measurement errors of approximately ±5 mW, much larger than the total power usage of many components. Finally, Carla S. Ellis performed system-level measurements on a Palm Pilot to obtain overall power usage for certain activities.[6]

References

1. J.R. Lorch and A.J. Smith, "Apple Macintosh's Energy Consumption," *IEEE Micro*, vol. 18, no. 6, Nov. 1998, pp. 54-63.
2. J. Pouwelse, K. Langendoen, and H. Sips, "Power Consumption Trade-Offs for Wireless Audio Access," *Proc. Int'l Workshop Mobile Multimedia Communications*, 2000, pp. P6.1-P6.6.
3. J. Pouwelse, K. Langendoen, and H. Sips, "Dynamic Voltage Scaling on a Low-Power Microprocessor," *Proc. Int'l Conf. Mobile Computing and Networking* (Mobicom 2001), ACM Press, 2001, pp. 251-259.
4. T. Šimunić et al., "Event-Driven Power Management," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 7, July 2001, pp. 1700-1742.
5. K.I. Farkas et al., "Quantifying the Energy Consumption of a Pocket Computer and a Java Virtual Machine," *Proc. ACM SIGMETRICS*, ACM Press, 2000, pp. 252-263.
6. C.S. Ellis, "The Case for Higher-Level Power Management," *Proc. Workshop Hot Topics in Operating Systems* (HotOS VII), IEEE CS Press, 1999, pp. 162-167.

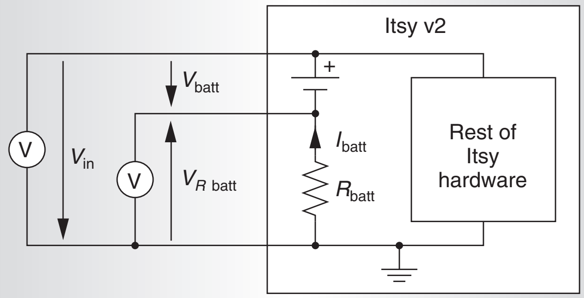

Figure 1. Experimental setup for the battery lifetime measurements.

Experimental setup

Figure 1 shows the experimental setup we used for the battery lifetime measurements. Itsy v2 features a precision sense resistor R_{batt} in series with the battery. Two external multimeters monitor input voltage V_{in} and the voltage drop across sense resistor V_{Rbatt}. Both Itsy and the multimeters connect to a host computer through serial links. A power supply and a few relays (not shown in Figure 1) let the host computer run a series of experiments without operator intervention, charging the battery between experiments.

For the power-breakdown measurements, we used a similar setup of 11 multimeters to measure all the voltages shown in Figure 2. From these measurements, we divided power consumption into the nine elementary power domains listed in Table 1.

We measured the nine domains in different ways. Power terms such as P_{DRAM} and P_{core} are direct measurements. Power supply terms $P_{main\ sup}$ and $P_{core\ sup}$ are differential measurements (the subtraction of two measured quantities) taken during the same run. Finally, we measured LCD power P_{LCD} and speaker power P_{spkr} by running the same workload twice: once with the LCD or speaker connected and again with it disconnected. These are also differential measurements, but we took them during different runs. Each term reflects only the power dissipated by the individual domain itself. For comparison with other numbers published in the literature, it is also possible to calculate the difference of the *total* power between two corresponding runs (for example, with and without a speaker), which also includes the overhead caused mainly by power supply inefficiencies.

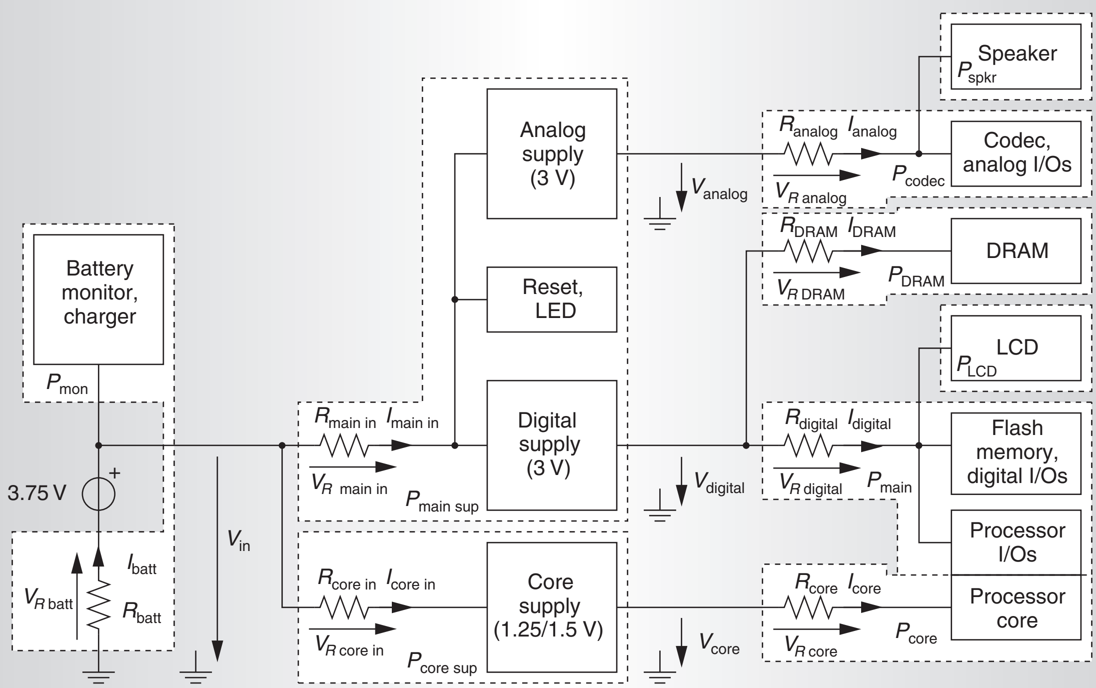

Figure 2. Itsy power domains. We directly measured each voltage and calculated each current from the corresponding sense-resistor voltage drop. Dashed lines delimit elementary power domains.

Battery lifetime

The host computer measured the battery lifetime. Itsy signals that it is still alive by sending a string of two characters (that is, a character and a new line) at regular intervals. The duration of this interval represents a tradeoff between the overhead to send characters and the battery lifetime measurement resolution. By measuring this overhead, we could estimate the resulting power and lifetime errors. To achieve the best tradeoff, we chose a different interval for each experiment.

Average power

An electrical system's average power is the time integral of the instantaneous power (voltage-current product) divided by the total time. Because the moderate-cost instruments we used cannot perform such an operation, we used an approximation.

We configured all multimeters to perform a predefined number of measurements N while calculating the average voltage and tracking the minimum and maximum values. At the end of every acquisition (set of N measurements), all multimeters upload the average, minimum, and maximum voltages to the host computer, which also records the acquisition's start and end times.

We calculated currents from the corresponding voltage drops across the sense resistors using Ohm's law: $I = V_R / R$. For each acquisition a, we calculated an approximation $\tilde{P}_{\text{avg}, a}$ of the average power $P_{\text{avg}, a}$ as

$$\tilde{P}_{\text{avg}, a} = V_{\text{avg}, a} \cdot I_{\text{avg}, a}$$

Finally, average power P_{avg} during an experiment is the time-weighted mean of all terms $\tilde{P}_{\text{avg}, a}$.

Error analysis

We cannot overemphasize the importance of a complete error analysis. We made a complete list of all possible errors and calculated or estimated their values. Some errors turned out to be negligible, and we subsequently ignored them. Calculating or estimating all errors is critical because the most important terms often are not those that intuitively seem most important.

Most of our error analysis was fairly conventional and came directly from the specifications of the sense resistors and the multimeters, as well as from the software instrumentation and initialization.[4,6] However, the most interesting error arose from using approximation $\tilde{P}_{\text{avg}, a}$ for average power. This error is bounded by the term

$$\varepsilon_{\text{avg}, a} = \frac{N-1}{2 \cdot N} \cdot \frac{\left(V_{\max, a} - V_{\min, a}\right) \cdot \left(I_{\max, a} - I_{\min, a}\right)}{\tilde{P}_{\text{avg}, a}}$$

Because this term is directly proportional to the difference between the per-acquisition maximum and minimum of the voltage and current, the multimeters must record the minimum and maximum values.

This error requires special attention because its upper bound can become arbitrarily large, even exceeding 100 percent, making the measurement meaningless. The most important parameter affecting this term is N, the number of measurements per average value uploaded to the host computer. An earlier study showed that large numbers of measurements per uploaded value can result in large error terms, even if the actual errors are probably much smaller.[5] For this study, we ran short experiments with different values of N and calculated the error's upper bound. We could then choose a value of N that would result in an acceptable error yet keep the overhead of uploading results low.

Table 1. Power domains.

Domain	Source of power dissipation
P_{mon}	Battery monitor and leakage through the charger circuit
$P_{\text{main sup}}$	Digital and analog power supplies, reset circuit, and white light-emitting diode (LED)
P_{spkr}	Speaker
P_{codec}	Codec's analog part, microphone, and touch screen
P_{DRAM}	DRAM
P_{LCD}	LCD and LCD backlight
P_{main}	I/O pins and part of the peripherals of the StrongARM SA-1100, as well as the rest of the digital logic: flash memory, daughter-card buffers, serial interfaces, codec's digital part, two-axis accelerometer, and buttons
$P_{\text{core sup}}$	Core power supply
P_{core}	StrongARM SA-1100 core

Benchmarks

We used the following benchmarks for our study:

Table 2. Itsy's average power and battery lifetime.

			Phase 1: Battery lifetime experiments					Phase 2: Breakdown experiments	
			Power		Lifetime			Power	
Benchmark	**Frequency (MHz)**	**V_{core} (V)**	**mW**	**Error (%)**	**Hours (days)**	**Error (%)**	**Capacity (W-hour)**	**mW**	**Error (%)**
Deep sleep	0	0	4.58	9.5	500.0 (20.8)	4.3	2.29	4.40	4.0
Sleep	0	0	7.40	5.8	308.5 (12.9)	3.5	2.28	7.18	2.9
Sleep, LCD	0	0	26.2	2.6	87.0 (3.6)	2.6	2.27	25.8	1.6
Idle	59	1.25	55.4	4.8	40.6	4.9	2.25	55.5	1.7
Idle	59	1.5	69.6	6.7	32.3	6.8	2.25	68.8	1.6
Idle	133	1.5	83.0	6.2	27.0	6.3	2.24	81.9	1.6
Idle	206	1.5	101	5.8	22.0	6.1	2.23	99.3	1.5
WAV	59	1.5	279	3.6	7.75	4.2	2.16	277	1.3
WAV	206	1.5	311	4.4	6.88	4.4	2.14	308	1.4
DECtalk	74	1.25	353	2.6	6.11	3.0	2.16	350	1.3
DECtalk	74	1.5	398	2.8	5.35	3.4	2.13	393	1.2
DECtalk	206	1.5	402	4.3	5.29	3.7	2.13	394	2.1
MPEG-1	206	1.5	826	5.3	2.42	3.8	2.00	808	1.5

- *Deep sleep.* The processor is in sleep mode.[2] All units that can be disabled or left unpowered are off.
- *Sleep.* This is the same as the deep-sleep benchmark, but the DRAM is in self-refresh mode.
- *Sleep, LCD.* This is the same as the sleep benchmark, but the LCD displays a static monochrome image.
- *Idle.* Itsy is running the Linux operating system without a workload. Thus, the processor is mostly in idle mode.
- *WAV.* Itsy is playing an audio file in Windows waveform (WAV) format with the speaker at nearly full power (using the same file and settings as for the MPEG-1 benchmark).
- *DECtalk.* Itsy is generating an audio stream from a text file (text-to-speech), using the DECtalk program.
- *MPEG-1.* Itsy is playing a video file in MPEG-1 format, using a modified version of the easympeg program and mpeg_lib libraries. Playing the matching audio WAV file produces the audio.

All idle- and run-mode experiments ran under the Linux operating system. The LCD was enabled with the backlight off. The touch screen and buttons were ready to accept input but never actually used. The audio was enabled only when used. The serial interfaces and the LED were always disabled, while the two-axis accelerometer was always on. This configuration simulates a system's typical operation when not connected to a host computer. We used no daughter-card during these experiments, so the daughter-card buffers stayed disabled.

Results

Table 2 presents selected results from both phases of the study. (We have published the complete results in a research report.[6]) All physical quantities (power and time) are rounded to the closest least-significant digit shown, and errors are rounded to the next highest least-significant digit shown.

For the Phase 1 experiments, Table 2 lists the average power and battery lifetime. As the second and third columns show, the benchmarks ran at various frequencies and with the processor core at two different voltages. The fourth column shows total power averaged over a complete battery discharge on three Itsy systems. The sixth column shows average battery lifetime for the same set of experiments. The eighth column shows effective battery capacity (power-lifetime product).

The table shows that Itsy's lifetime varies

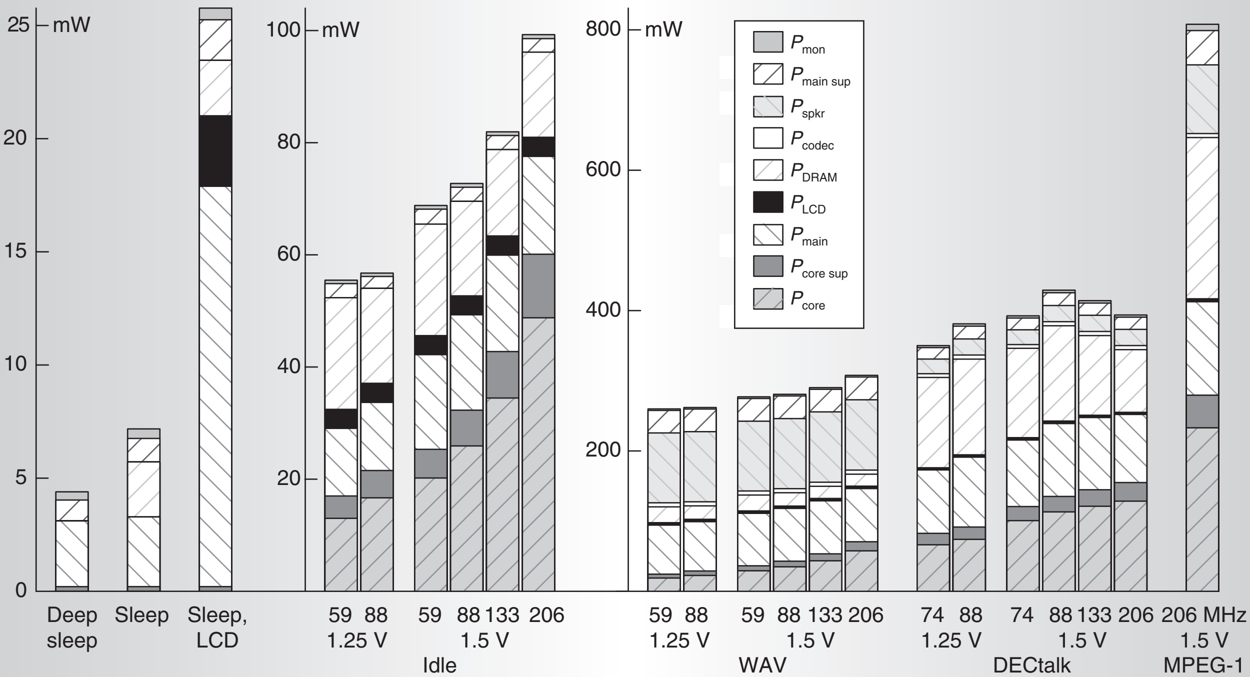

Figure 3. Average-power breakdown of an Itsy computer running with a 3.75-V power supply.

greatly depending on the workload. In sleep mode, Itsy can keep the memory refreshed for almost 13 days. When not in sleep mode, the battery lifetime varies from 40 hours, when idle, to 2.4 hours for our most power-hungry benchmark, the real-time MPEG-1 player. Real-world usage should result in a lifetime somewhere between these two values. We expect that the battery will last a full day for most users. Unsurprisingly, the battery capacity drops slightly with higher workloads.

In Phase 2, to understand exactly where Itsy was using power, we used the effective battery voltage calculated from the Phase 1 data to measure the power breakdown of all processor frequencies higher than the minimum required to run the benchmark. From 59 MHz to 88 MHz, we also ran experiments at the low core voltage (V_{core} = 1.25 V). Figure 3 shows a representative subset of this data.

The last two columns of Table 2 show the average power data taken during the second phase. As we expected, the average power for this phase was always within the average power error of the battery lifetime experiments, confirming that 3.75 V was appropriate as an effective battery voltage.

Hardware analysis

Despite being a research platform, Itsy has well-behaved power characteristics. For example, units unused during any or all of the sleep-mode (processor core, DRAM, LCD, and speaker) and idle-mode (speaker) experiments draw an unmeasurably small amount of power.

Units that don't depend on processor frequency have a constant power dissipation. P_{main}, which includes the 3-V part of the processor, stays approximately constant with each benchmark. P_{codec} hardly changes across all sound-producing experiments. P_{spkr} is constant for all WAV and MPEG-1 benchmarks (they use the same sound file). This is also true of all DECtalk benchmarks. However, DECtalk is quieter than WAV, so it uses less speaker power. P_{LCD} is the same for all Idle, WAV, and DECtalk benchmarks.

At constant voltage, P_{core} increases monotonically with the frequency. Decreasing V_{core} from 1.5 V to 1.25 V significantly reduces this term but doesn't affect units other than the processor. Unexpectedly, the power dissipated by the 3-V part of the processor also decreases at the low core voltage, as reflected by P_{main}.

The DRAM behavior requires explanation. Ideally, P_{DRAM} should depend only marginally on processor frequency, because the duration of memory accesses should be constant. However, the StrongARM SA-1100 implements DRAM timings as multiples of the processor clock period. Due to rounding, access times decrease from 59 MHz to 118 MHz and are approximately constant at higher frequencies. This trend is evident in the DRAM power consumption. This type of DRAM controller design makes using some frequencies potentially undesirable.

As typical of portable electronic systems, Itsy's power supply efficiencies are between 75 percent and 95 percent. Except in sleep mode, however, the combined overhead $P_{main\ sup}$ + $P_{core\ sup}$ is always less than 16 percent.

Battery monitor power P_{mon} increases with total power, mostly because of sense resistor R_{batt}. However, its overhead is about 1 percent, except in sleep mode.

Benchmark analysis

Except for DECtalk, all the benchmarks in this study are fixed-duration tasks. The time spent in sleep or idle mode depends on the user, whereas the time required by the WAV and MPEG-1 benchmarks depends on the content being played. In this important class of applications, energy is directly proportional to power.

The total idle-mode power varies significantly as the frequency decreases. The dramatic decrease of P_{core} largely offsets the slight increase in P_{DRAM} caused by longer access times (DRAM and LCD refresh). An obvious conclusion is that the processor preferably should switch to the lowest frequency and voltage before entering idle mode.

A similar behavior occurs when Itsy is playing a WAV file in real time. In this case, the system must perform only a small amount of computation per time unit and is mostly idle. However, frequency has a smaller relative effect on total power because the speaker draws 32 percent to 38 percent of the power, also increasing the power dissipated by the supply.

The DECtalk benchmark presents a very interesting behavior. As Figure 3 shows, its total power consumption is approximately equal at 74 MHz and 206 MHz but is higher for intermediate values. At 74 MHz, the processor cycles are fully utilized. As the frequency increases, the processor spends increasing time in idle mode. The total power's behavior is due to the interplay of P_{core}, which increases with frequency; P_{DRAM}, which decreases with frequency; and P_{main}, which has no direct relation to frequency.

Itsy can run DECtalk at various speeds. However, the higher the frequency, the faster the program runs (it reaches a plateau at 177 MHz). Therefore, we should consider the total energy consumed for running the task. The energy used decreases slightly from 74 MHz to 206 MHz (from 170 J to 146 J) but does not do so monotonically (for example, Itsy uses more energy at 147 MHz than at 133 MHz or 118 MHz). Itsy uses the least energy when running at 206 MHz, but 88 MHz at the low core voltage is a very close second at 149 J. These results are important because they show that the interplay of different units can produce entirely nonintuitive power and energy behavior.

Applying the results

Because Itsy is a realistic handheld computer, the findings presented here have implications for the low-power design and power management of many similar handheld and wearable devices. A hardware designer can use these results as data points for the development of a low-power system. A software designer can use them to devise power management strategies for a hardware system. Obviously, our study is relevant only if such a system is similar enough to Itsy. Therefore, we have compared Itsy with commercial handhelds.

Low-power hardware

Unsurprisingly, the processor (P_{core} and P_{main}) and the DRAM consume the bulk of the power in idle and run mode. In future systems, we expect most power savings to come from these components as VLSI technology improves. Figure 3 shows that the speaker can also be a significant power user. Because speakers transmit power mostly as sound, we do not expect much improvement in that domain unless the use of headsets becomes common.

Accessing the DRAM requires considerable power, but the percentage of power needed to keep the DRAM refreshed is small in idle and run modes. Therefore, increasing the number of DRAM banks will not increase system power, as long as the processor doesn't implement simultaneous accesses to multiple banks. However, this situation does not apply to sleep mode, in which the DRAM refresh cost is high. Thus, implementing a mechanism to selectively unpower some of the banks is important. DRAMs' monetary cost and package size may also be important considerations.

Power management

Power management techniques aim to optimally use the fixed amount of energy stored in a device's battery. The goal is to minimize the energy required to perform a given task. Hence, power management techniques must weight power consumption by the time required to perform the task. The nonideal characteristics of batteries further complicate the power management problem.

The significant variation of core power at different voltages and frequencies offers an opportunity for energy savings through a technique called voltage frequency scaling.[7-9] The large difference in power between idle and sleep modes suggests other savings. Projects in our laboratories are currently exploring both of these ideas, using Itsy as a platform.

Comparison with commercial handhelds

Itsy is a representative testbed for power evaluation of handheld devices. For example, in a 1997 study, researchers measured two commercially available PDAs at 164 mW and 312 mW while idle.[10] The same team also calculated that a PDA in "typical use" requires 700 mW to 1,200 mW. These figures are slightly higher than those for Itsy because they came from an earlier hardware generation.

Itsy also compares favorably with a more recent handheld. We measured nearly identical power in sleep mode (7 mW to 8 mW) on Compaq's iPAQ H36xx (released in 2000), which uses the StrongARM SA-1110, a successor to Itsy's SA-1100 processor. (We modified an iPAQ by adding a sense resistor in series with the battery, similar to the Itsy setup in Figure 1. We took these measurements over a 30-second period with a partially charged battery; they are thus imprecise. The iPAQ was running Microsoft's PocketPC operating system.) On the other hand, the iPAQ's idle-mode power at 206 MHz was approximately 250 mW, in contrast to Itsy's 99 mW.

One reason that Itsy's idle mode takes less power than that of the iPAQ is that we chose a passive-matrix gray-scale LCD for Itsy rather than a more power-hungry active-matrix color display. By taking differential measurements, Sukjae Cho determined that the iPAQ LCD uses about 39 mW and that the LCD controller uses an additional 88 mW.[11] (Cho used an iPAQ running Linux, over a short period of time, with a partially charged battery.) Itsy's equivalent numbers, taking power-supply overhead into account, are 3.5 mW and 19 mW, in idle mode at 206 MHz. As we expected, the Itsy screen's power consumption differs from iPAQ's by an order of magnitude. The factor of four in the power dissipated by the LCD controller is easy to explain—the iPAQ's frame buffer (12-bit pixels padded to 16 bits) is four times the size of Itsy's (4-bit pixels). A color display would have at least doubled Itsy's idle-mode power and would have been a significant factor in the other benchmarks.

At full power, Itsy's backlight consumes 324 mW (343 mW including power supply overhead). Modulating the backlight to decrease its intensity proportionally reduces power consumption. Under PocketPC, the iPAQ frontlight can assume any of four different settings, which use from 400 mW to 960 mW. Clearly, the power required for a frontlight or backlight quickly becomes a dominant cost.

Because we built Itsy as a research platform, it is an ideal testbed for power studies. The Itsy processor itself can monitor four of its eight sense resistors, including R_{batt}, $R_{main\ in}$, and $R_{core\ in}$. (The fourth measures the charging current so is not used in this study.) We placed a sense resistor at every natural power boundary but did not separate any power planes solely for power monitoring. At least one such division would have been useful—namely, isolating the 3-V part of the processor from the rest of the 3-V digital logic. This would have provided a better understanding of processor power consumption and would have been useful for debugging. Because the StrongARM SA-1100 processor has 50 3-V pins, and because the power input to a processor should be extremely stable, we could not add such a feature later without remanufacturing the printed circuit board.

An important result of our study is a deeper understanding of why the relationship between frequency and power is not intuitive. We found that measuring a system's power is necessary to understanding this relationship. For system software to manage a battery's energy optimally, on-board power evaluation capability is crucial. MICRO

Acknowledgments

The Itsy project was led by William Hamburgen, who codesigned the hardware with Marc Viredaz. Lawrence Brakmo, Carl Waldspurger, and Deborah Wallach ported the Linux operating system. Many other members of the Compaq Western Research Laboratory and the Compaq System Research Center made significant contributions to the project. We also acknowledge our management's enthusiastic support. Feedback from Parthasarathy Ranganathan and the anonymous reviewers significantly improved this article's quality.

References

1. W.R. Hamburgen et al., "Itsy: Stretching the Bounds of Mobile Computing," *Computer*, vol. 34, no. 4, Apr. 2001, pp. 28-36.
2. *Intel StrongARM SA-1100 Microprocessor: Developer's Manual*, Document No. 278088-004, Intel, 1999.
3. T.L. Martin and D.P. Siewiorek, "The Impact of Battery Capacity and Memory Bandwidth on CPU Speed-Setting: A Case Study," *Proc. Int'l Symp. Low Power Electronics and Design* (ISLPED 99), ACM Press, 1999, pp. 200-205.
4. M.A. Viredaz and D.A. Wallach, *Power Evaluation of Itsy Version 2.4*, Tech. Note TN-59, Western Research Laboratory, Compaq, 2001.
5. M.A. Viredaz and D.A. Wallach, *Power Evaluation of Itsy Version 2.3*, Tech. Note TN-57, Western Research Laboratory, Compaq, 2000.
6. M.A. Viredaz and D.A. Wallach, *Power Evaluation of a Handheld Computer: A Case Study*, Research Report 2001/1, Western Research Laboratory, Compaq, 2001.
7. K. Flautner, S. Reinhardt, and T. Mudge, "Automatic Performance-Setting for Dynamic Voltage Scaling," *Proc. Int'l Conf. Mobile Computing and Networking* (Mobicom 01), ACM Press, 2001, pp. 260-271.
8. T. Pering, T. Burd, and R. Brodersen, "The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms," *Proc. Int'l Symp. Low Power Electronics and Design* (ISLPED 98), IEEE Press, 1998, pp. 76-81.
9. J. Pouwelse, K. Langendoen, and H. Sips, "Dynamic Voltage Scaling on a Low-Power Microprocessor," *Proc. Int'l Conf. Mobile Computing and Networking* (Mobicom 01), ACM Press, 2001, pp. 251-259.
10. M. Stemm and R.H. Katz, "Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices," *IEICE Trans. Communications*, vol. E80B, no. 8, Aug. 1997, pp. 1125-1131.
11. S. Cho, "[iPAQ] Experiment Result about Power and To Do List," Feb. 2001, http://www.handhelds.org/pipermail/ipaq/2001-February/003962.html.

Marc A. Viredaz is a senior member of the technical staff at Hewlett-Packard Laboratories. He performed the work described in this article at Compaq before it merged with Hewlett-Packard. His research interests include handheld and mobile computing, low-power systems, multiprocessor systems, and computer architecture. Viredaz has a PhD in computer engineering from the Swiss Federal Institute of Technology at Lausanne. He is a member of the IEEE and the IEEE Computer Society.

Deborah A. Wallach is a software engineer at Google's System Laboratory. She was previously at Compaq, where she performed the work described in this article. Her research interests include mobile computing, operating systems, and power management. Wallach has a PhD in computer science from the Massachusetts Institute of Technology.

Direct questions and comments about this article to Marc A. Viredaz, viredaz@computer.org, and Deborah A. Wallach, kerr@alum.mit.edu.