Accurate Battery Lifetime Estimation Using High-Frequency Power Profile Emulation

Farhan Simjee
Center for Embedded Computer Systems
University of California- Irvine
Irvine, CA 92697-2625 USA
fsimjee@uci.edu

Pai H. Chou
Center for Embedded Computer Systems
University of California- Irvine
Irvine, CA 92697-2625 USA
phchou@uci.edu

ABSTRACT

For accurate estimation of battery lifetime, researchers have developed analytical and empirical models and applied them to representative load profiles. However, accurate battery models are not available for most batteries on the market. Although high-accuracy simulation models exist for certain battery chemistries, they are computationally intensive and still require calibration through trial and error. To address this problem, this paper presents a low-cost load emulation platform for automated, accurate battery estimation. By draining a battery with high-frequency emulation of a system power profile, all of the battery characteristics are accounted for, including the discharge rate and recovery effects. A designer can then accurately observe how the system effects battery life, quantify lifetime performance for multiple batteries, and ultimately optimize the system's power scheduling around a particular battery.

Categories and Subject Descriptors: B.8.2 [Performance and Reliability]: Performance Analysis and Design Aids

General Terms: Measurement, Performance, Verification.

Keywords: Load emulation, Battery lifetime estimation, Power profiling.

1. INTRODUCTION

One of the key problems in battery-powered systems is the estimation of battery lifetime. Because batteries are non-ideal power sources, different discharge patterns can result in different levels of battery efficiency. A variety of power estimation methods have been proposed, but ultimately all of them must be verified and validated. Today's approaches can be divided into three categories: battery measurement, battery simulation, and load emulation.

Battery measurement entails connecting an actual battery to a system that consumes power by executing some workload. Often one would script CPU tasks, but actual workload on these battery-powered systems are often reactive to many sources of inputs, including network activities, keyboard and mouse clicks, and possibly multimedia ports. These events are hard to reproduce precisely. Manually generating inputs by typing on the keyboard or clicking the mouse would be quite tedious if many types of batteries must be tested, or if the same battery is tested for aging or temperature effects. If a system is still being designed and no physical system exists, it may be possible to obtain an estimated power profile by running SPICE/VHDL simulations, but there still needs to be a way to drain a battery according to this load.

Researchers have used digital scopes to collect power profiles and use them as input to a battery simulation program. However, detailed, electrochemical battery simulation programs are very computationally intensive and can simulate at 20 ms timestep for a real-time simulation run [1, 2]. At this resolution, higher-frequency power fluctuations will not be taken into account. Faster simulators have been proposed [3] though not formally validated with actual batteries. Further, as the simulation time resolution is decreased to speed up the simulation, the accuracy of the battery life estimation severely decreases [4]. These simulators are then limited to coarse-grained, piecewise constant power profiles with little or no fluctuation and thus do not correspond to realistic load.

A third approach is to play back the power profile not to a battery simulator but to the battery as load. A load emulator is a hardware device that draws the specified amount of power over time according to the power profile. It discharges the battery exactly the same way as the actual system would. This not only makes the experiments fully reproducible but also makes it possible to automate many runs for the purpose of testing different types of batteries, as well as testing the aging or temperature effects. Commercial instruments are available to serve as load emulators however most are rather expensive and are intended for much higher power draw than the small power fluctuation in most battery-powered systems.

This paper first describes the theory and design of our load emulator that not only has all the same speed and reproducibility advantages of commercial load emulators, but also many more essential features that were possible previously only when running the actual system. Our load emulator supports not only automated batch testing of various battery effects, but also stitching together actual and synthetic power profiles conditionally. Second, analysis from the experimental data shows how the battery behavior is influenced by load characteristics including the dynamic range, switching frequencies, and fluctuation density. This load emulation system is being integrated into a power-aware system design framework that will use these battery testing results to improve and refine the power management capabilities of a system under design.

2. RELATED WORK

Load emulation has been used to validate techniques that maximize energy draw from the battery based on analytical models [5]. They emulate only the average power consumption of a profile rather than detailed fluctuations that are observable in high-frequency measurements. Electronic loads have been on the market

ISLPED'05, August 8–10, 2005, San Diego, California, USA

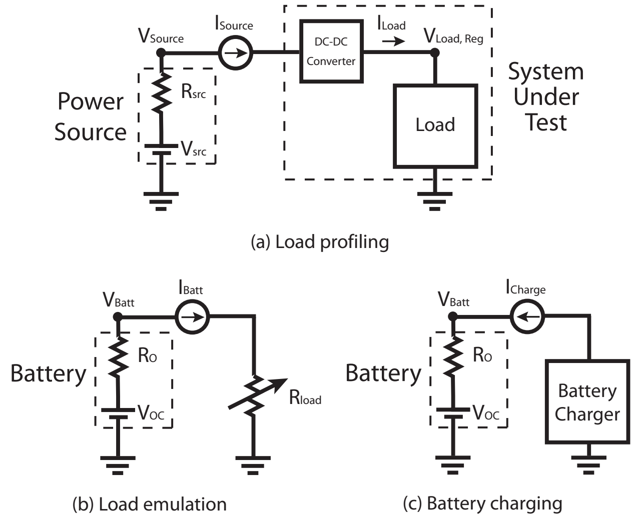

Figure 1: Load emulator operating modes

for quite some time [6]. The primary application is to test power supplies for accuracy, response time, and efficiency. At first glance they may appear ideal for emulating a system load profile, but one caveat is that their operating range is from one watt to several hundred. In contrast, most low-power electronics can operate around the single milliwatt range. Other work has been done on emulating the transient response of low-power systems [7]. This groundwork for load emulation provides insight for a more general purpose system that will not only emulate but also automate the whole process of battery life estimation.

3. PROBLEM STATEMENT

The problem is to build a high-frequency load emulator to discharge numerous batteries, using a power profile obtained by measurement or simulation. As shown in Fig. 1, the platform has three modes of operation: capturing system power profiles from the system under test (SUT), discharging batteries with load profile emulation, and charging the battery to automate scripted emulation runs. We define the following variables:

V_{oc} open-circuit voltage of battery

R_o internal battery impedance dependent on charge capacity, load current, temperature, etc.

V_{batt}, I_{batt}, P_{batt} volt., curr., power dissipated at batt. terminal

$V_{Sys,Reg}$, I_{Sys}, P_{Sys} voltage, current, and power supplied from DC-DC converter to system load

R_{load} resistive load emulating system power dissipation

$\eta_{converter}(t, V_{batt}(t), I_{Sys}(t), V_{Sys,Reg})$ converter efficiency

V_{cutoff} battery output voltage at which system shuts off

V_{source}, I_{source}, $P_{batt,measured}$ voltage, current, and power measured for load profile

$P_{batt,corrected}$ power dissipation corrected for converter efficiency

To emulate a system, the emulator requires a sampled load profile curve, henceforth known as the system power profile. A setup similar to Fig. 1(a) can perform measurement of V_{source} and I_{source} when a physical SUT is powered. One roadblock to accurately profiling the system load is the voltage converter that regulates the varying input voltage, V_{source}, and supplies the system a steady supply voltage, $V_{Sys,Reg}$. Due to the high efficiency and popularity of the DC-DC converter in battery-powered systems, this regulator will be used for modeling purposes. Thus, the battery power of this system can be described as (1) [4].

$$P_{batt} = \frac{P_{Sys}}{\eta_{converter}\left(t, V_{batt}(t), I_{Sys}(t), V_{Sys,Reg}\right)} \quad (1)$$

A problem now arises since the efficiency is not a constant value, but rather time-varying, due to the response time of the converter, and dependence upon V_{batt} and output power, both of which are time-varying as well. Fortunately, the switching frequency of these regulators are at least 100kHz. With a target sampling rate in the lower kilohertz range, higher frequency transients are averaged out and thus cause little change in efficiency. The time-varying system power consumption, P_{sys}, will influence the efficiency but since V_{source} and I_{source} are being measured at the input of the converter, these variations will already be accounted for in the measurements. Lastly, the input battery voltage, V_{batt}, will decline during a discharge and thus directly affect the efficiency level. In general, systems that have varying regulator efficiency across the discharge of the battery will observe P_{batt} multiplied by a scalar across the length of a battery discharge and thus the visible dynamic power range shifts either up or down, because of lower or higher efficiency, respectively. An efficiency correction factor $\eta_{correction}(V_{batt})$, which is solely a function of V_{batt}, can be extrapolated from the measurements and incorporated into the load profile using (2).

$$P_{batt,corrected} = P_{batt,measured}\eta_{correction}(V_{batt}) \quad (2)$$

So far, numerous measurements on the iPaq have shown no measurable difference in a profile's dynamic power range across a complete cycle of battery discharge. This probably has to do with the fact that the iPaq battery has a narrow dynamic operational voltage range from fully charged, V_{oc}, to V_{cutoff}. Lithium-ions for instance have an operational voltage range of ~30%, (e.g. 4.2V down to 3.0V). If a particular system is observed to have a varying power range, then a corrective factor should be used; otherwise, a battery of the same chemistry and capacity or a power source with similar voltage should be used for measuring the load profile to maintain similar regulator efficiency (3). The converter efficiency is then approximated as (4).

$$V_{source} \approx V_{batt} \quad (3)$$

$$P_{batt,measured} \approx \frac{P_{Sys}}{\eta_{approx}} \quad (4)$$

The third problem is automated charging as shown in Fig. 1(c). Repeated battery discharging is needed for measuring a battery over multiple profiles. Even for a given load pattern, the battery lifetime can still vary due to the random behavior in chemical reactions and thus require multiple discharges cycles. To speed up these emulation runs, an integrated battery charger is used to recharge the battery to a preset voltage level (V_{oc}). This enables the designer to interleave recharge cycles in a sequence of load emulations.

4. LOAD EMULATOR DESIGN

Fig. 2 shows the entire load emulator system. It combines the three components in the circuit model controlled by an embedded microcontroller. Profiling is taken care of by attaching the system under test and using the on-board ADC and current sensor to

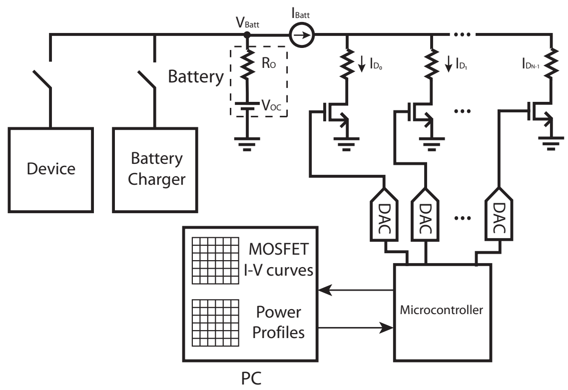

Figure 2: Power emulator circuit diagram

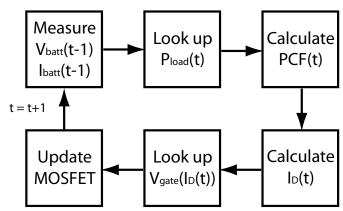

Figure 3: Emulation functional steps

measure the voltage and current. The integrated battery charger enables automated load emulation runs with recharging capability in between each cycle.

Taking inspiration from [7], load emulation entails adjusting MOSFET gate voltages in the saturation region so that the desired current flows through a resistor and transistor channel for dissipation. In general, most MOSFET saturation regions have at least two decades of current range, and so just a few branches of monotonically-larger transistors operating in the saturation region can easily source a wide and continuous range of current as shown in Fig. 2 and described by (6). At any given interval then, the current $I_{batt}(t)$ is calculated by a lookup in the power profile table and the measured battery voltage (7). Assuming that only one branch is sourcing current at a time, and given I_{batt}, the correct gate voltage is acquired from a MOSFET lookup table and programmed to the DAC by the microcontroller (8). The MOSFET lookup table is generated by sweeping the gate voltage and measuring the current drain for each transistor used in the system. Although a first-order MOSFET equation such as (5) or SPICE models provided by manufacturers can be used, they simply do not provide the level of accuracy necessary for a calibrated load emulation.

$$I_{D_i} = \frac{\mu_{n,i} C_{ox,i}}{2} \frac{W_i}{L_i} \left(V_{gate,i} - V_{t,i}\right)^2 \quad (5)$$

$$I_{D_0} < I_{D_1} < \ldots < I_{D_{N-1}} \quad (6)$$

$$I_{batt}(t) = I_{D_0}(t) + I_{D_1}(t) + \ldots + I_{D_{N-1}}(t)$$
$$= \frac{P_{load}(t)}{V_{batt}(t-1)} \quad (7)$$

$$V_{gate,i}(I_{D_i}(t)) = V_{gate,i}(I_{batt}(t)) \quad (8)$$

The interdependence of the battery parameters requires I_{batt} to be calculated iteratively by a battery model, but since the idea of load emulation is to avoid the need for models, this option is not viable. The best method to counter the deviation is through using memory so that the system remembers the local deviation amount and automatically multiplies the current by a power correction factor, $PCF[t]$ in Eqn. (12). If the deviation continues to increase, then the system will ramp the correction factor after every iteration until the correct power level is met. That is why $PCF[t]$ takes into account the overall power correction across a window of w previous iterations using a running average (10) as well as the immediate correction (11). Once this correction factor is calculated, it is simply multiplied to the current (12). Each iteration can be summed up as a looped series of functional blocks as shown in Fig. 3.

$$V_{batt} \propto \frac{1}{I_{batt} R_o} \quad (9)$$

$$PCF[t] = \frac{1}{w} \sum_{j=t-w-1}^{t-1} PCFarray[j] \quad (10)$$

$$PCFarray[t] = \frac{P_{load}(t-1)}{P_{batt}(t-1)} PCF[t] \quad (11)$$

$$I_{batt,PCF}(t) = \frac{P_{load}(t)}{V_{batt}[t]} PCF[t] \quad (12)$$

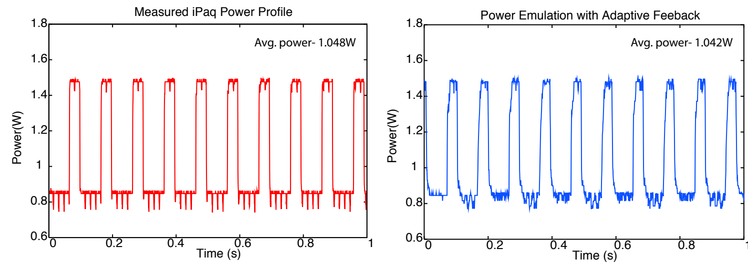

Figure 4: iPaq power profile and emulation

Profile	CPU Core Frequency (MHz)	CPU Core Voltage (V)	Avg. Itsy Power (W)
p0	206.4	1.452	.5830
p1	176.9	1.304	.4375
p2	147.5	1.215	.3398
p3	118.8	1.126	.2550
p4	88.5	1.037	.1875
p5	59.0	0.919	.1245

Table 1: Itsy power profiles captured for emulation

5. EXPERIMENTAL RESULTS

As many power schedulers today use the average power of each power mode to calculate energy usage, comparing the power distribution of the profile and at 0.2 Hz emulation makes it hard to realize that the battery lifetime will be similar. On the left diagram of Fig. 5, the average power profile is generated from averaging each power mode. As discussed earlier, the battery lifetime with the averaged power profile emulation will theoretically be less affected by real-world battery characteristics such as discharge rate effect and recovery effect. Emulation at lower frequencies, such as the case with averaged power profiles, should then have a larger deviation in battery life measured from a real system. Given a set of monotonically-increasing emulation frequencies, f, and the battery lifetime as a function of each frequency, $T(f)$, the deviation in

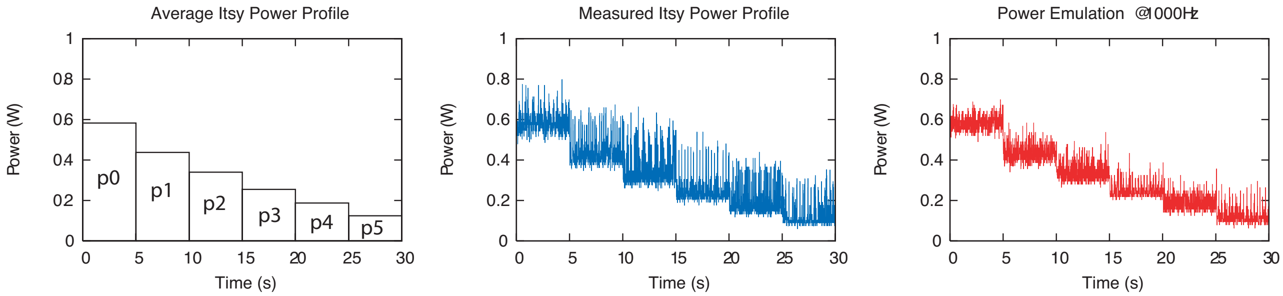

Figure 5: Comparison of Itsy power profile to load emulation waveform

battery life from the real system to time-averaged load profiles can be described as follows:

$$f \in \{f_0, f_1, \ldots, f_\infty\} \quad (13)$$

$$T(f) \in \{T(f_0), T(f_1), \ldots, T(f_\infty)\} \quad (14)$$

$$f_\infty = f_{system} \quad (15)$$

$$T(f_\infty) = T(f_{system}) \quad (16)$$

$$|T(f_\infty) - T(f_0)| > |T(f_\infty) - T(f_1)| > \ldots > |T(f_\infty) - T(f_\infty)| \quad (17)$$

As the emulation frequency approaches f_∞, inclusive of all frequencies, with $T(f_\infty)$ defined as the battery lifetime under the real system by Equations (15)(16), then the deviation in battery lifetime between the emulation and the real device decreases. Since no discrete-time sampled system can capture all possible frequencies, perfect emulation remains impossible, but at high enough frequencies a very close approximation is achievable. To further illustrate this concept, the measured Itsy profile is then emulated at different frequencies ranging from 0.2 Hz to 1 kHz, the operational frequency limitation of the load emulator prototype. Table 2 shows the result of numerous emulation runs and the experimental data suggests that as the emulation frequency is increased, the battery lifetime approaches that of the Itsy system. Since the power profiles for lower frequencies are time-averaged and their total average powers are the same, the average measured power dissipation for each emulation run should then statistically be similar, as validated by the experimental data.

The data gathered from this prototype reveals two attributes of batteries and load emulation: first, a battery's lifetime performance is heavily dependent upon the pattern, frequency, and dynamic range of the load waveform; and second, high-frequency load emulation can provide a very good approximation of load on the battery, allowing accurate estimation of battery performance and benchmarking power schedulers.

6. CONCLUSIONS

This paper presents a programmable load emulator system for validating battery lifetime estimation techniques. This system automates the discharge according to collected or synthetic power profiles and supports various resolutions. This then provides lower development time and cost as the testing process would otherwise be difficult if not impossible to reproduce accurately. Experiments showed that this load emulator is able to accurately emulate the discharge of batteries using the iPaq and Itsy system power profiles. These profiles are available either from circuit/system simulation software or through voltage and current measurement from a physical system. Future work will entail improving the circuit design to increase the emulation frequency, so that an even higher level of precision and accuracy is attainable. Plans are also underway to integrate the load emulation circuitry into battery emulation instruments to emulate not only the supply but also charging of rechargeable batteries. Together, these features will be indispensable for designers who need to more aggressively utilize battery capacity.

Board	$f_{emulation}$ (Hz)	Battery lifetime (min)	Lifetime Deviation %	Avg. Power (W)
Emu.	0.2	222.8	−11.87	0.3198
”	10	230.1	−8.98	0.3193
”	100	236.1	−6.61	0.3193
”	1000	246.0	−2.69	0.3184
Itsy	-	252.8	-	0.3216

Table 2: Battery lifetimes from varying emulation frequencies and Itsy

7. REFERENCES

[1] P. Chou, C. Park, J. Park, K. Pham, and J. Liu. B#: a battery emulator and profiling instrument. In *Proc. of Intl. Symposium on Low Power Electronics and Design (ISLPED)*, pages 288–293, Aug. 2003.

[2] J.S. Newman. FORTRAN programs for simulation of electrochemical systems. `http://www.cchem.berkeley.edu/~jsngrp/`.

[3] D. Panigrahi, C. Chiasserini, S. Dey, , R. Rao, A. Raghunathan, and K. Lahiri. Battery life estimation of mobile embedded systems. In *Proceedings of VLSI Design*, pages 57–63, 2001.

[4] K. Lahiri, A. Raghunathan, and S. Dey. Efficient power profiling for battery-driven embedded system design. *IEEE Trans. on Computer-aided Design of Integrated Circuits and Systems*, 23(6):919–932, June 2004.

[5] Daler Rakhmatov and Sarma Vrudhula. Energy management for battery-powered embedded systems. *ACM Trans. on Embedded Computing Systems*, 2(3):277–324, Aug. 2003.

[6] Agilent Technologies. *Agilent DC Electronics Loads Models N3300A-N3307A*.

[7] J.A. McNeill, M. Layler, G. Levesque, J. Ruiter, and J. Noon. A 50 A, 1-us-rise-time, programmable electronic load instrument for measurement of microprocessor power supply transient performance. In *Proceedings of Instrumentation and Measurement Technology Conf.*, pages 410–414, May 2000.