A Power Metric for Mobile Systems

Thomas L. Martin[1], Daniel P. Siewiorek

Engineering Design Research Center[2]
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

The advent of mobile computing has made power consumption a critical design factor. There has been little systematic consideration of the power sources for mobile computing systems. This paper presents a class of system-level metrics intended to make a systematic study more feasible and to more accurately reflect the trade-off between battery life and performance. An example involving the clock frequency of a CPU shows the possible impact of the nonlinearity, an impact not predicted by existing metrics. Results of an initial attempt to verify the example are presented and explained. The paper concludes with drawbacks of the metric and possible extensions to overcome these drawbacks.

1. Introduction

Design of computer systems inherently involves trade-offs between constraints. Historically, the major constraints have been performance and cost. In addition, mobile computing systems are constrained by their power consumption. The techniques for reducing power have, so far, been ad hoc. Very little work has been done to study the relationship between power and performance at the system level.

One of the problems of current work is that it considers only the power of the systems, while the factor that matters to the user is battery life. The power of the system determines such characteristics as the internal temperature of the system, but it is the power together with the power source--usually batteries--which determine the useful working life of the system. Neglecting the power source may lead to design decisions that do not benefit the user.

Existing metrics assume that batteries are a source of constant energy. This assumption is the basis of statements such as "...if the energy source is a battery, halving [the power by halving the clock frequency] is equivalent to doubling the computation time, while maintaining constant computation per battery life" [1]. The available energy of a battery, however, is affected by the load current. The two effects of the load on the battery capacity that seem most likely to be exploitable by mobile systems are nonlinearity and recovery. Figure 1a shows the relationship between capacity and current for a typical NiCd battery. The current is measured in terms of the "C" rating, a standard notation of the battery industry in which the load current is normalized to the battery's capacity [2][3]. For example, a discharge current of 1C for a battery with a C rating of 500 mA-hours is 500 mA, while a discharge current of 1C for a battery with a C rating of 1000 mA-hours would be 1000 mA. The C rating is specified as the capacity for a given time of discharge. The capacity in Figure 1a was measured at the 2 hour rate, since 100% capacity occurs at 0.5C. Figure 1b shows the recovery effect, where a reduction of the load for periods of time results in an increase in battery capacity. The voltage rises while the load is reduced, and the overall time of discharge increases. This phenomena occurs because, during the time when load is reduced, reactants in the battery diffuse to the reaction location, allowing more of them to be used during the life of the battery. The degree to which the battery recovers depends on the discharge rate and the length of time the load is reduced, as well as the details of the battery construction.

Depending on the battery life of a mobile system, there are other battery phenomena which may be important, such as self-discharge. This paper, however, will focus on systems with battery lives of less than ten hours, where nonlinearity may be a factor.

The nonlinearity of batteries was determined empirically by Peukert in the late 1800's. Peukert's formula for the energy capacity of a battery is

$$C = k/I^{\alpha},$$

where k is a constant determined by the chemical family and physical design of the battery and I is the discharge current. For an ideal battery, $\alpha = 0$, i.e. the capacity is constant, but for real batteries, α ranges up to 0.7 for most loads, and like k is determined by the chemical family and physical design of the battery [2]. Peukert's formula does not capture the recovery effect, but it is useful as a first-order approximation of the battery life for most loads.

A metric which does not reflect the nonlinearity of the battery can be misleading in some cases. Assuming a constant load, then by using Peukert's formula it can be shown that a

1. This material is based upon work supported under a National Science Foundation Graduate Research Fellowship.

2. This work has been supported by the Engineering Design Research Center, A NSF Engineering Research Center, and the Advanced Research Project Agency.

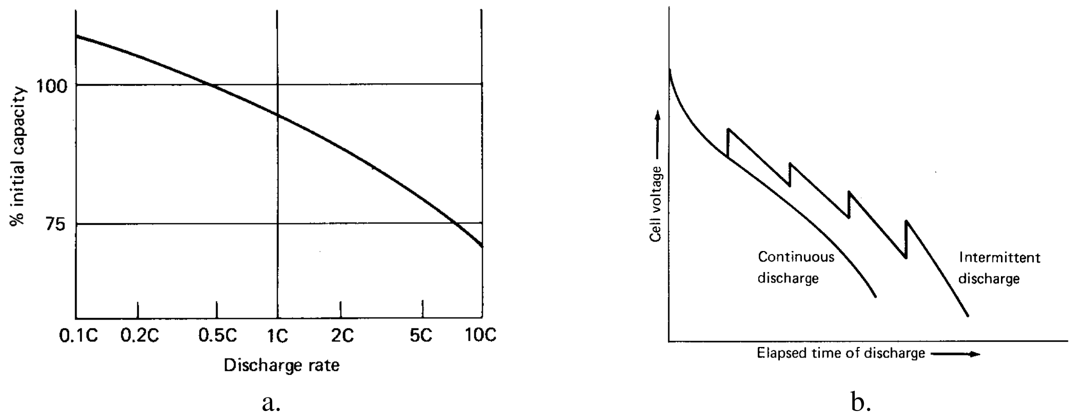

Figure 1: Battery nonlinearity and recovery (after [2])

modification which changes the system power by a factor of x will change the battery life by a factor of $x^{-(1+\alpha)}$. Battery life is important to the user of a mobile computer, but it does not reflect the performance requirements of the system. Performance is also important to the user, but by it does not reflect how long that performance can be sustained. The proposed class of metrics reflects the total amount of computation that the user can complete by multiplying performance, T, by the battery life, L, or

$$W = TL$$

If a proposed performance modification to a system changes the performance by a factor of n and the power by a factor of x, then ratio of the work completed by the modified system, W_n, to the work completed by the unmodified system, W, is

$$W_n/W = T_nL_n/TL = (nT)(x^{-(1+\alpha)}L)/TL = nx^{-(1+\alpha)}$$

In order for a modification to increase the amount of work completed, W_n/W must be greater than 1, so

$$nx^{-(1+\alpha)} > 1$$

Applying this to the earlier statement about halving the clock frequency and power of a system, then constant computation per battery life is not maintained. If the new system has half the power and half the performance (i.e., x = 0.5 and n = 0.5), the work ratio becomes

$$(0.5)\ (0.5)^{-(1+\alpha)} = 2^{\alpha}$$

which is greater than 1 for $\alpha > 0$. For non-ideal batteries, the slower system completes more computation per battery life in this situation.

The rest of the paper is organized as follows: Section 2 expands on this example of changing CPU clock frequency. Section 3 compares the work ratio to other metrics. Section 4 presents the results of an initial experiment to verify the example. Section 5 outlines our plans for future work, including the use of a battery model which reflects both the nonlinearity and recovery effects, and Section 6 summarizes the paper.

2. Derivation of the work ratio for CPU clock frequency

Generalizing the previous example of CPU clock frequency serves to show the usefulness of the work ratio, by examining a wider range of systems to determine what clock frequency allows a system to complete the most computation per battery life. The situation is an over-simplified version of most portable systems, because it assumes that the system is constantly active and cannot take advantage of any power management. There are systems where this may be the case, such as a remote data logger where the only requirement is that as many data points as possible be gathered, or a laptop computer which is being used to re-compute a large spreadsheet, with the computation time being nearly equal to the battery life. While the situation may be an oversimplified version of most portable systems, it shows that the work ratio gives a more useful answer than existing metrics.

To determine the amount of computation completed during a battery life, it is necessary to relate the battery life of the system to the CPU clock frequency, and the computation performed per unit time to the CPU clock frequency.

In the earlier example, it was assumed that the power consumed by the system was directly proportional to the CPU clock frequency. A more general assumption is that the current drawn by a system has two components, S and DF, where S is the component of the current that is independent of the CPU frequency F, and DF is the component that depends on the CPU frequency,

$$I = S + DF$$

The power of the system, P, is then

$$P = V(S + DF)$$

where V is the voltage of the battery. This more general relationship allows one to represent the full range of possibilities, from the system of the earlier example, where the power of the system is directly proportional to the CPU clock frequency (S=0), to systems where the system power does not depend on the CPU frequency (D=0). The energy capacity of a battery, C, is related to its discharge current I by Peukert's formula,

$$C = k/I^{\alpha},$$

The life of the battery, then, is

$$L = C/P = (k/I^{\alpha})/[V(S + DF)] =$$
$$k/[V(S + DF)I^{\alpha}] = k/[V(S + DF)^{1+\alpha}]$$

Assuming that V is constant, let k' = k/V and the battery life is then

$$L = k'/(S + DF)^{1+\alpha}$$

For this example, computation will be measured by the number of iterations of a loop that can be performed. Let j be the number of iterations of the loop executed per unit time. Then the total number of iterations the system can complete during a battery life is

$$W = jL = jk'/(S + DF)^{1+\alpha}$$

If the CPU frequency is changed by a factor n, n > 0, then the CPU frequency dependent portion of the power changes from DF to DFn, and, ideally, the number of iterations executed per unit time from j to jn. Then the total number of iterations the system can complete is

$$W_n = njL = njk'/(S + DFn)^{1+\alpha}$$

Taking the ratio W_n/W gives

$$W_n/W = [njk'/(S + DFn)^{1+\alpha}]/[jk'/(S + DF)^{1+\alpha}] =$$
$$n\left(\frac{S+DF}{S+DFn}\right)^{1+\alpha}$$

Letting $\rho = S/(S + DF)$, the percent of the system power that is independent of the CPU clock frequency, then

$$W_n/W = n\left(\frac{S/\rho}{(S/\rho) + (n-1)(S/\rho)(1-\rho)}\right)^{1+\alpha}$$

$$= n\left(\frac{1}{\rho + n(1-\rho)}\right)^{1+\alpha}$$

Figure 2 shows the contours of W_n/W for $\alpha = 0.2$, $0 < n < 10$. The dashed line shows the maximum value of W_n/W for a given ρ. There are two regions in the plot, one where increasing the clock frequency from n = 1 is initially beneficial and one where decreasing it from n = 1 is initially beneficial. The two regions are separated by the dashed line. The region where it is initially beneficial to increase the clock frequency is above the line; the region where it is initially beneficial to decrease the clock frequency is below the line. Only in the two ideal cases, $\rho = 0$ or 1, is it always better to decrease ($\rho = 0$) or increase ($\rho = 1$) the clock frequency. For all other values of ρ, a maximum occurs at $n = \rho/\alpha(1-\rho)$, as will be shown in Section 3.

For a given system, changing the clock frequency corresponds to a horizontal movement from $(1, \rho)$ to (n, ρ). ρ remains constant because it was defined using only the initial system. In order to complete the most computation per battery life, n should be set to the value which maximizes W_n/W. For n > 1, if this new, faster system is used to define a ρ', then $\rho' < \rho$. If $n = \rho/[\alpha(1-\rho)]$, then $(1, \rho')$ is the point where the 1.0 contours cross. At this value of ρ, the CPU frequency is optimal, because any change in n will result in a decrease in W_n/W.

The Navigator 1 wearable computer [4] will serve as an example of how the contour plot can be used. The power of the Navigator 1 was measured to be (5.36 + 0.15 x CPU frequency) Watts. The minimum operating frequency of the design is 8 MHz and the maximum is 32 MHz. Suppose the current design operates at 8 MHz and a designer would like to know if would be better to operate at some other frequency. At 8 MHz, then $\rho = 5.36/(5.36 + 0.15 \times 8) = 0.82$. At $\rho = 0.82$ and n = 1 in Figure 2, decreasing the frequency decreases the work ratio, and increasing the frequency increases the work ratio. The designer would then decide to increase the clock frequency. In this case, the designer would be limited by the maximum frequency of 32 MHz. 32 MHz corresponds to n = 4, with the work ratio being approximately 2.5, or a 150% increase in the number of iterations executed during a battery life.

Suppose the designer, in order to take advantage of W_n/W increases beyond 32 MHz, wanted to somehow increase the maximum frequency. The peak work ratio occurs at $n = \rho/[(1-\rho)\alpha] = 0.82/(0.18 \times 0.2) = 22.8$, or at 182 MHz. The system should not be operated beyond this point.

Suppose, however, that the design were modified to lower the static portion of the power from 5.36W to 1.00 W. The design still operates at 8 MHz, so $\rho = 1.00/(1.00 + 0.15 \times 8) = 0.45$. Figure 3 shows a close-up of the contour plot around $\rho = 0.45$, with an 'x' marking the current design and a line showing the possible increase in W_n/W by increasing the clock speed. The maximum occurs near n = 4.1, with Wn/W approximately 1.24. The close-up shows that W_n/W is very flat from n = 3.5 to n = 5 for $\rho = 0.45$. While the maximum occurs at n = 4.1, little could be gained by increasing the maximum frequency of the design beyond 32 MHz.

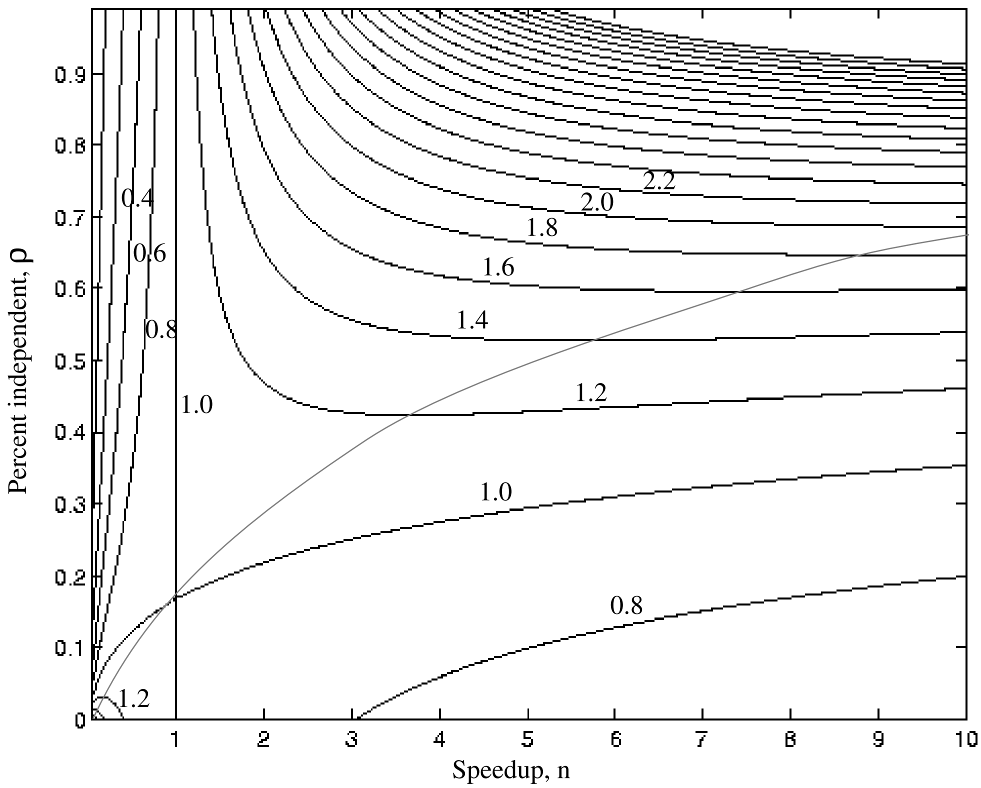

Figure 2: W_n/W contours, $\alpha = 0.4$

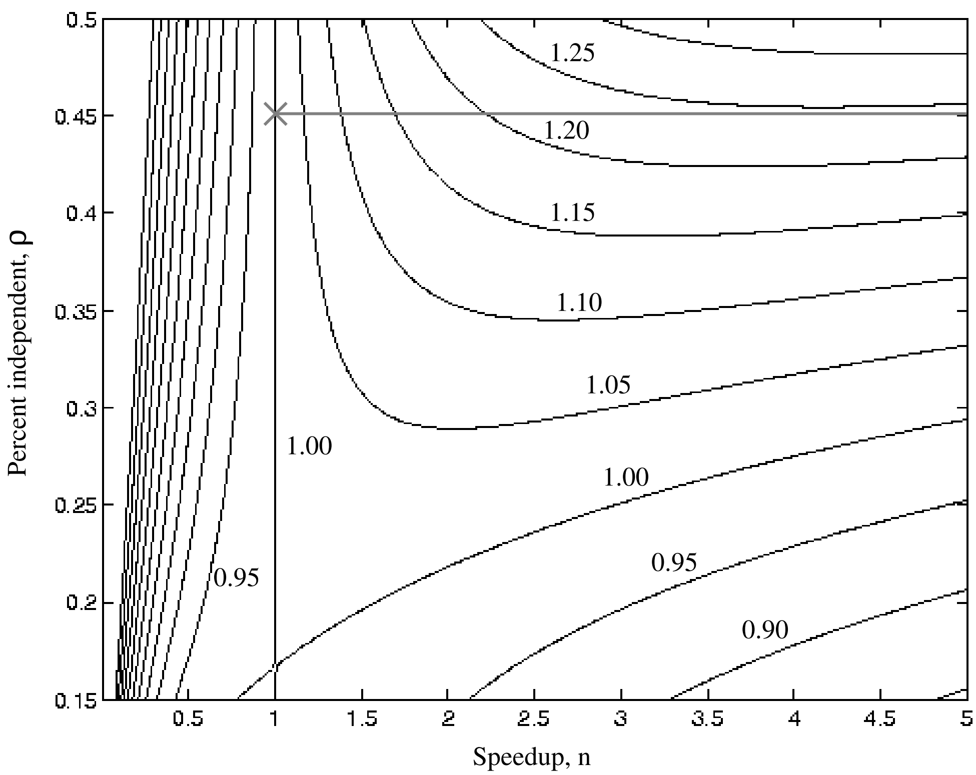

Figure 3: Close-up of work ratios around $\rho = 0.45$, showing the maximum possible increase for that ρ

3. Comparison of the work ratio to existing metrics for the case of CPU clock frequency

Unlike previous metrics relating performance and power, the work ratio shows that in some situations there is a useful trade-off between performance and battery life. An occasionally used power metric for CPU's intended for portable applications is MIPS/Watt [1][5]. An analysis of units shows that MIPS/Watt is actually (millions of instructions/s)/(J/s) = millions of instructions/J. For a given CPU, increasing its clock frequency from F to AF increases its MIPS rating by a factor of A, assuming that other factors such as I/O do not become bottlenecks. Increasing the clock frequency also increases the power, but by less than a factor of A because power has a static component and a dynamic component: $d + eF$. Because $F/(d + eF) < AF/(d + eAF)$ for $A > 1$, the CPU running at a higher frequency will always have a higher MIPS/W metric than it does when running at a lower frequency. MIPS/W, then, is monotonically increasing, and does not give any sense of diminishing returns.

Another metric in the literature power-delay, which for CPU's has the units W/Spec^2 [6]. Unlike MIPS/W, the power-delay metric has the property that "smaller is better". Again, increasing the clock frequency from F to AF increases its Spec rating by a factor of A, while the power increases by less than a factor of A. So a CPU running at a higher frequency will always have a lower W/Spec^2 rating than it does when running at a lower frequency. As with MIPS/W, W/Spec^2 does not give any sense of diminishing returns.

The work ratio gives a sense of diminishing returns. It does not always become better with increasing frequency. Taking the partial derivative with respect to the speedup factor,

$$\partial W_n/\partial W =$$
$$[-n(1 + \alpha)(1 - \rho) + (\rho + n(1 - \rho))]/[(\rho + n(1 - \rho))]^{-(2+\alpha)}$$

Setting the numerator equal to 0 and solving for n, one finds that the maximum occurs at $n = \rho/[\alpha(1 - \rho)]$

So, unlike the previous metrics, the work ratio has a point where increasing the CPU frequency is no longer beneficial. By substituting the above value of n into the equation for ρ, one finds that the optimal point is where $\rho = \alpha/(\alpha+1)$. Whether most systems can operate at this point is a question for further investigation.

4. Results

The Navigator 1 is a wearable computer intended for a campus tour application and is fully described in [4]. To test the work ratio, the Navigator 1 was modified to run in a continuous loop. The number of loops executed was counted by attaching a logic analyzer to the Navigator 1's address bus and triggering on an address occurring only once per loop. The Navigator 1 ran until the batteries were fully discharged at clock frequencies of 8 and 32 MHz. These correspond to n = 1 and 4, respectively. The experiment was performed five times at each frequency. Table 1 shows the results.

The first thing to note is the low value of α. The original Navigator 1 used a different set of batteries due to higher peak current requirements, which had an α of about 0.3. The α for a typical NiCd battery of the family used for the experiments is 0.1. Even with the low α, nonlinear behavior was predicted and observed. The calculated work ratio is within 2% of the measured work ratio. However, for the values of α, n, and ρ, the work ratio is very near the value for an ideal battery, as shown by the bottom row. Further testing is needed, but these early results are encouraging. With higher values of α and n, the difference from ideal behavior will be greater.

Clock frequency, MHz	8	32
Iterations	135 +/- 2	343 +/- 3
Experimental work ratio, W_n/W	1.00	2.53 +/- 0.05
Expected work ratio, W_n/W, ($\alpha = 0.1$)	1.00	2.49
Ideal battery work ratio, W_n/W, ($\alpha = 0.0$)	1.00	2.60

Table 1: Results for Initial Experiment with Navigator 1

5. Future work

There are two main drawbacks to the current approach. The first is that Peukert's formula, while easy to manipulate mathematically, is limited to constant loads within the range of loads for which α has been empirically determined. It is not useful for the variable loads typical of power management schemes, a prime concern of designers of mobile systems. The second drawback is that the assumptions about the performance break down at very small and very large values of n. For example, slowing the CPU below a given frequency may allow the memory wait states to be reduced, so that the performance is reduced by less than expected. However, over the middle range, the assumptions are adequate for an intuitive understanding of the trade-off.

The first drawback can be overcome by using a more complex battery model. We are currently investigating the Kinetic Battery Model, KiBaM[7], which allows variable loads over a wider range and provides voltage as a function of the discharge level. The added complexity of KiBaM increases the difficulty of mathematical manipulation, making simulation preferable for all but the simplest cases.

Simulation can also be used to overcome the second drawback. The system models available via simulation more accurately reflect the variation of performance with the variable of interest, in this case, CPU clock frequency. Effects which are difficult to include in an analytical model, such as the impact of memory wait states, are easily incorporated into a simulation.

Our future work includes the following:

1) Verifying the work ratio for a broader range of n. The current system allows the CPU frequency to vary by only a factor of 4. The initial results, while encouraging, are not sufficiently different from the ideal case to verify the work ratio. The contour plots show that the frequency should vary by up to a factor of 10 in order to verify the work ratio by demonstrating that the maximum exists. A platform that allows a wider variation is being sought. If a conventional computer system that allows this variation cannot be found, a simple oscillating circuit will be built that has the characteristics of the model to verify the work ratio. The question will then become whether or not there are interesting systems where the work ratio can be applied, and if not, how can existing systems be modified to take advantage of this trade-off between performance and battery life. A related question is the range of ρ for real systems. Both the Navigator 1 and the VuMan 2, another wearable computer described in [4], have a ρ of 0.8-0.9. High values of ρ means that systems will need to operate over a wide range of frequencies to take advantage of the battery nonlinearity.

2) Extending the work ratio to cases more complex than CPU frequency. While the CPU frequency example is interesting, the work ratio can be applied to performance-battery life trade-offs involving algorithms, memory hierarchy, and power management. A battery model allowing variable loads will be an essential part of examining the more complex cases. The more complex battery model should not only capture the nonlinear behavior of batteries, but should also reflect the recovery effect that may result from variable loads.

As new techniques are used to reduce power consumption, there will likely be more distinct periods of high and low power operation, allowing the recovery effect to be exploited. Existing power management schemes result in this variable power operation, and several proposed techniques may result in it as well. For example, one alternative for the memory system of mobile systems is to use Flash memory for all read-only memory and a small amount of DRAM for scratchpad memory [8][9]. Flash has lower access power [10], so in systems where memory power is a large portion of system power, periods of accessing Flash followed by periods of accessing DRAM will result in a power profile that might allow the recovery effect to be exploited. The areas of such a system that impact this profile are the operating system (access scheduling, either directly through power management or indirectly through process scheduling) and algorithm design (ordering of long periods of data access). Considering the power source may lead to different trade-offs in these areas than considering average power alone.

6. Summary and Conclusions

This paper has presented a new approach to the performance-power trade-off at the system level which involves modeling the power source. Using a first-order model of a battery and a simple example, we have shown that ignoring the nonlinear behavior of the batteries can lead one to make design decisions that do not benefit the user. The current approach has drawbacks due to its overly simple assumptions, and extensions to overcome these drawbacks have been proposed. We believe that the work ratio is an appropriate metric for system level performance-battery life trade-offs because it addresses the most important aspect for users of mobile systems--the amount of work they can perform during the battery life of the system.

References

[1] Burd, T. and Brodersen, R. “Energy Efficient CMOS Microprocessor Design,” Proceedings of the Twenty-Eighth Annual Hawaii International Conference on System Sciences; Vol.1; Wailea, HI, Jan. 1995.

[2] Linden, D. Handbook of Batteries and Fuel Cells. New York: McGraw-Hill, 1984.

[3] Hageman, S. “Simple PSpice Models Let You Simulate Common Battery Types,” EDN, October 28, 1993, pp. 117-129.

[4] Smailagic, A. and Siewiorek, D. “The CMU Mobile Computers: A New Generation of Computer Systems,” Proceedings of COMPCON ‘94; San Francisco, CA, 28 February-4 March 1994.

[5] NEC Electronics Inc. VR4100 Microprocessor Product Brief, 1995.

[6] Horowitz, M, Indermaur, T, and Gonzalalez, R. “Low-Power Digital Design,” Proceedings of the 1994 Symposium on Low Power Electronics, October 1994, pp. 8-11.

[7] Manwell, J. and McGowan, J. “Lead Acid Battery Storage Model for Hybrid Energy Systems,” Solar Energy Vol. 50, No. 5, pp. 399-405.

[8] Elnozahy, E. Private communication.

[9] Mayer, J. “Design Team Gives Thumbs Up to Portable Platform,” Portable Design, February 1996, pp. 47-51.

[10] Intel. Memory Products Databook, 1993.