Discrete-Time Battery Models for System-Level Low-Power Design

Luca Benini, Giuliano Castelli, Alberto Macii, Enrico Macii, Massimo Poncino, and Riccardo Scarsi

***Abstract*—For portable applications, long battery lifetime is the ultimate design goal. Therefore, the availability of battery and voltage converter models providing accurate estimates of battery lifetime is key for system-level low-power design frameworks. In this paper, we introduce a discrete-time model for the complete power supply subsystem that closely approximates the behavior of its circuit-level continuous-time counterpart. The model is abstract and efficient enough to enable event-driven simulation of digital systems described at a very high level of abstraction and that includes, among their components, also the power supply. The model gives the designer the possibility of estimating battery lifetime during system-level design exploration, as shown by the results we have collected on meaningful case studies. In addition, it is flexible and it can thus be employed for different battery chemistries.**

***Index Terms*—Batteries, digital systems, energy management, power demand.**

I. Introduction

During the development of a low-power digital system, the attention of designers is focused on the minimization of the power dissipated by the circuits and interfaces that perform computations, storage and data transfer/communication. Accurate and efficient power models for digital circuits at various levels of abstraction have been developed to support design space exploration [1]. Unfortunately, much less attention has been dedicated to power supply models. In many cases, it is implicitly assumed that the power supply provides a constant voltage and delivers a fixed amount of energy. This assumption is not valid in the case of battery-operated devices.

Even though power dissipation is a primary concern in the design of portable electronic devices, top-level specifications are not given in terms of maximum average power (or energy), but rather in terms of *minimum battery lifetime*. Furthermore, the portability requirement imposes tight constraints on maximum battery weight. For these reasons, successful portable applications combine low-power design techniques with careful battery selection and power-supply design [2].

As observed in [3], [4], a battery is not an ideal finite-charge power supply. The energy stored in a fully charged battery cannot be supplied to the digital circuitry to its full extent and the usable energy cannot be supplied at a constant rate. This is because the amount of energy a battery can provide depends on the current drawn from the battery itself. In other words, the higher the discharge current, the higher the energy waste of the battery [5], [3].

Design and optimization of digital circuits in portable systems ask for a careful understanding of battery behavior. Accurate simulation models for battery and dc–dc conversion circuitry are required to properly tune system power demands. Furthermore, the explosive growth of the portable electronics market has spurred numerous research efforts targeting the development of new and improved battery technologies, capable of providing more energy without increasing weight [6]. As a result, when designing a portable product, designers are faced with the nontrivial challenge of selecting a battery that delivers sufficient energy at low cost, with small weight and volume. Postponing battery selection and power distribution design to late phases of the development flow (or even to prototyping) may lead to serious violations of lifetime/weight specifications, which may impose redesign. Overestimation of battery lifetime with a given battery chemistry imposes the selection of a more expensive chemistry, which seriously impacts cost and can cause market failure [7]. It is therefore important to study the interaction of active devices with as many battery alternatives as possible in the early steps of the design process, when a system prototype is not yet available [8].

Battery simulation models [5], [9]–[11] have been developed to help designers estimating the discharge characteristics of common batteries, much before such characteristics can be measured by connecting the actual battery to a system prototype. Battery models have traditionally been formulated at the circuit level, because of the analog, continuous-time nature of battery discharge phenomena. Unfortunately, a continuous-time circuit-level battery model requires a load model at the same level of abstraction. Obviously, modeling the entire system loading the battery at the circuit level is a challenging task. Furthermore, circuit-level simulation of a system over the typical lifetime of a battery would require an enormous amount of time.

On the other hand, battery-conscious power metrics such as those introduced in [3], [4] suffer two limitations. First, they relate battery lifetime to the average current absorbed by active circuits. However, lifetime of actual batteries does not depend only on average current, but also on the profile of the time-domain current waveform. Second, they neglect the presence of voltage converters which can be responsible for a significant fraction of the total power. These inaccuracies may prevent accurate design-space exploration, especially in the case of power-managed systems, which exhibit highly nonstationary current waveforms.

Manuscript received August 14, 1999; revised February 16, 2001.

L. Benini is with the Dipartimento di Elettronica, Università di Bologna, Informatica e Sistemistica, Bologna 40136, Italy.

G. Castelli, A. Macii, E. Macii, M. Poncino, and R. Scarsi are with the Dipartimento di Automatica e Informatica, Politecnico di Torino, Torino 10129, Italy.

Publisher Item Identifier S 1063-8210(01)03690-3.

In this paper, we propose discrete-time battery and dc–dc converter models that help in bridging the efficiency gap between electrical-level and high-level simulation, without incurring in the accuracy losses which are normally imposed by simplified battery-conscious power metrics. The model takes into account first-order effects like dependence of battery voltage on its state of charge, discharge rate, and discharge frequency. Second-order effects like battery output resistance and dependencies on the temperature are also considered. Accounting for the latter is needed for accurately modeling modern high-capacity batteries that can provide high currents and are therefore subject to *IR* drops and self-heating.

The complete discrete-time model can be implemented as a VHDL entity (or as a Verilog module) that well approximates the continuous-time behavior but, at the same time, is fast and efficient enough to enable high-level, event-driven simulation of a complete system description. Therefore, it can be used for the purpose of battery lifetime estimation of systems described at a very high level of abstraction.

Model development and validation is first carried out for Lithium–Ion batteries (i.e., batteries that find very wide usage in portable electronic systems); then, it is extended to other battery chemistries, both for primary (i.e., nonrechargeable) and secondary (i.e., rechargeable) cells. This is a first step toward the construction of a "battery library" that can be employed by designers to compare several chemistries and cell types during the early stages of design exploration.

Usage of the model in the context of system-level design is illustrated through case studies. In the first one, we consider an abstract description of a small personal digital assistant (PDA), and we show how the availability of an high-level battery model is important in evaluating the effectiveness of dynamic power management techniques, when applied to battery-operated applications. In addition, we exploit this example to illustrate how battery modeling can be of help to system engineers in the task of battery selection. The second benchmark we analyze is an MPEG 2-Layer 3 (MP3) digital audio player. Here, we demonstrate how system-level battery modeling capabilities are key for the development and testing of dynamic power management policies which are dependent on the state of charge of the battery.

The rest of the manuscript is organized as follows. Section II summarizes the key characteristics of continuous-time battery models, available in the literature, that we have used as reference for the development of our discrete-time model. The latter is described in details in Section III. Section IV provides experimental results concerning the validation of the discrete-time model. Section V discusses model extensions to other chemistries, e.g., nickel–cadmium, alkaline, lead–acid, while Section VI illustrates, through realistic case studies, how high-level battery models can be fruitfully exploited in the context of system-level design exploration. Finally, Section VII concludes the paper.

II. Continuous-Time Power Supply Model

Several circuit-level battery models have been proposed in the past [5], [9]–[11]. In this section, we will identify the key features of continuous-time battery models that need to be reproduced in a discrete-time setting to achieve accurate lifetime estimation.

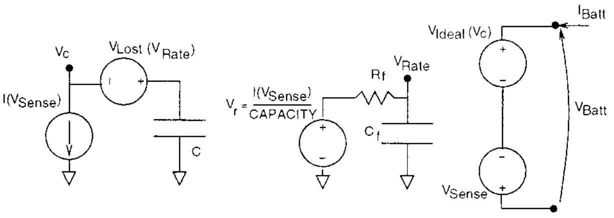


Fig. 1. First-order continuous-time battery model.

We will also describe the principle of operation and a modeling technique for dc–dc converters, which are the main components of the power distribution circuitry in portable systems.

A. Battery

Charge storage in a battery can be modeled as a capacitor with capacitance $C = 3600\cdot$ *CAPACITY*, where *CAPACITY* is the nominal capacity in Ahr, which is usually provided in the battery's data sheet. By setting the initial voltage across the capacitor $V_C = 1$, we initialize the battery to its fully charged state. Unfortunately, the simple linear capacitor model is not accurate enough to model complex phenomena observed during battery discharge. We will briefly describe these phenomena and a circuit-level model that takes them into account. Refer to [9], [11] for a complete treatment.

1) First-Order Effects: There are three first-order effects that a model of a battery must take into account.

- *Battery voltage depends nonlinearly on its state of charge (SOC)*: Voltage V_{Batt} decreases monotonically as the battery is discharged, but the rate of decrease is not constant.
- *The actual usable capacity of a battery cell depends on the discharge rate*: At higher rates, the cell is less efficient at converting its chemically stored energy into available electrical energy.
- *The "frequency" of the discharge current affects the amount of charge the battery can deliver*: The battery does not react instantaneously to load changes, but it shows considerable inertia, caused by the large time constants that characterize electrochemical phenomena.

These three first-order effects can be modeled at the circuit level, as shown in Fig. 1.

Dependency on the SOC ($V_{\text{Ideal}}(V_C)$) is realized by storing several points of the curve into a lookup table (LUT) addressed by the value of the state of charge (V_C). The model is accurate up to a minimum *cutoff voltage*, after which the battery is considered fully discharged.

Dependency on discharge rate is modeled with a voltage source V_{Lost} in series with the charge storage capacitor. Voltage V_{Lost} reduces the apparent charge of the battery [which controls battery voltage (V_{Batt})]. The value of V_{Lost} is a nonlinear function of the discharge rate (which can be modeled by another LUT). Dependency on the discharge frequency, and the time-domain transient behavior of the battery are modeled by averaging the instantaneous discharge rate used to control V_{Lost} through a low-pass filter (R_f, C_f). The low-pass filter

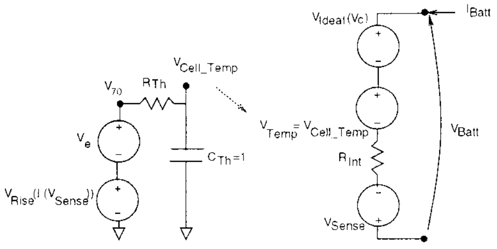


Fig. 2. Second-order continuous-time battery model.

models the relative insensitivity of batteries to high-frequency changes in discharge current.

Notice that $V_{\rm Sense}$ is a zero-valued voltage source added in series with the output voltage functions as the discharge-current ($I_{\rm Batt}$) sensor. *CAPACITY* is the total capacity of the battery. According to [9], [11], this model fits measured data fairly well (within 15%). This accuracy is acceptable, since the actual capacity of any group of cells may vary as much as 20% between identical units, when we take into account manufacturing variances [9].

2) Second-Order Effects: Among the various secondary phenomena that affect battery voltage [9], two are nonnegligible: External temperature and battery internal resistance. Properly taking them into account is key to ensure model accuracy for large-capacity batteries, where the high currents delivered can cause RI drops and self-heating.

Temperature may impact cell behavior in many ways. The most sizable effect is due to the offset in the output voltage caused by the heat released by the cell. This effect is particularly evident for high discharge currents. The effect of temperature can be modeled as a voltage loop similar to that of $V_{\rm Rate}$ in Fig. 1, as shown in Fig. 2 [11]. The state variable in the thermal loop on the left (V_{Cell_Temp}) causes an offset $V_{\rm Temp}$ in the cell output voltage. V_{Cell_Temp} is obtained as the sum of the equivalent voltage (V_e) of the environmental temperature and the voltage source ($V_{\rm Rise}$).

$V_{\rm Rise}$ is proportional to the temperature rise due to the resistive drop $R_{\rm Int} \cdot I^2(V_{\rm Sense}) \cdot \theta$, where θ is the temperature rise of the cell per Watt dissipated in free air [9]. R_{Th} lumps the effects of both thermal resistance and thermal capacitance.

The effect of the internal resistance amounts to subtracting its resistive voltage drop from the effective output cell voltage. The resistance $R_{\rm Int}$ in the model of Fig. 2 is used to account for such voltage drop. Other second-order phenomena influencing cell internal resistance such as the dependence of the resistance on the cell temperature or the state of charge, can be neglected.

B. dc–dc Converter

The output voltage of a battery depends on its chemistry and its state of charge. During operation, battery voltage is not well controlled. Thus, the battery cell cannot be connected directly to active circuits, but it requires the presence of a dc–dc *converter* for shifting and stabilizing the voltage supply. The most common dc–dc converter circuits for battery-operated devices are *switching converters* [12]. A basic switching down-converter known as *buck converter* [12] is shown in Fig. 3.

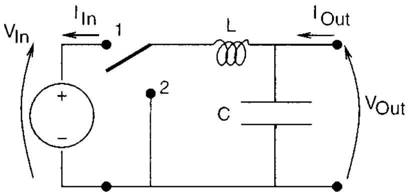


Fig. 3. Simple dc–dc buck converter structure.

A single-pole, double-throw switch is alternatively connected to the dc input voltage and to ground. The switch output is connected to a LC low-pass filter. If the switch position is changed periodically, at a frequency $f \gg 1/2\pi\sqrt{LC}$ and with duty-cycle $D \leq 1$, the output voltage of the converter is nominally $V_{\rm Out} = DV_{\rm In}$; thus, the buck converter performs voltage down conversion.

We observe that the buck converter is only one of the many switching converters described in the literature. We focus on a specific topology for the sake of explanation; however, our high-level model can be used for generic converters. All real-life dc–dc converters have sizable losses, usually collapsed in a single figure of merit called *efficiency*:

$$\eta = \frac{P_{\rm Out}}{P_{\rm In}}.$$

Typical efficiencies are within the range [0.8, 0.9]. For a given fixed output voltage value $V_{\rm Out}$, we can represent η as a nonlinear function $\eta(V_{\rm In}, I_{\rm Out})$. Efficiency curves are usually plotted in the data sheets of commercial dc–dc converters, and are used by system designers to choose among different converters and to set the operating point.

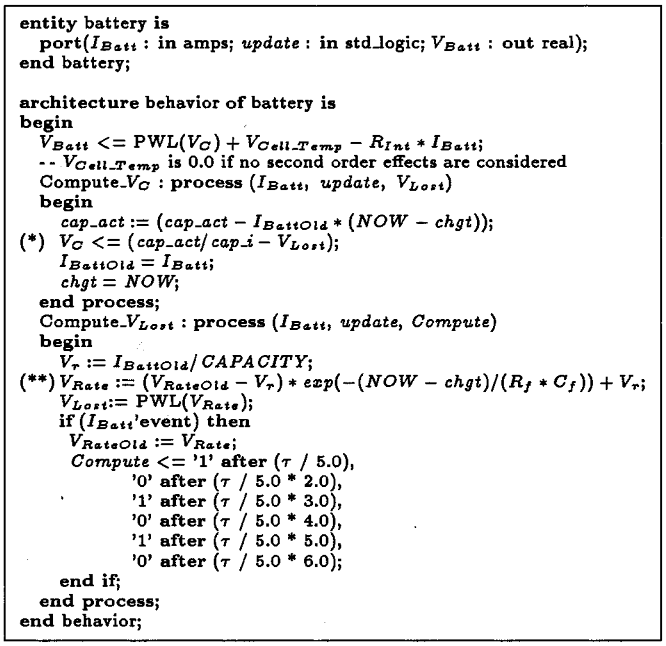
We can envision three classes of circuit-level dc–dc converter models: 1) transistor-level models; 2) behavioral white-box models; and 3) behavioral black-box models.

Transistor-level models contain the complete switch-level structure of the converter. These models are too complex to be simulated in reasonable time for long simulation periods. Behavioral white-box models employ simplified behavioral component models. An example of such models is the circuit in Fig. 3. Even the simulation of simple white-box behavioral models requires excessive time, because the switching period of typical dc–dc converters (a few microseconds) is several orders of magnitude smaller than the time required to estimate battery lifetimes (several hours).

Behavioral black-box models do not contain any information on the internal structure of the converters, but they just mimic their I/O characteristics. A behavioral black-box model is a two-port circuit; the output port is connected to the load, and it appears as a voltage source with fixed $V_{\rm Out}$ and small output impedance. The input port is connected to the battery and it appears as a current sink (i.e., a current source that absorbs current); notice that in this case, $V_{\rm In}$ and $I_{\rm In}$ correspond to $V_{\rm Batt}$ and $I_{\rm Batt}$ of Fig. 2, respectively.

The input current is expressed as the following:

$$I_{\rm Batt} = \frac{V_{\rm Out} I_{\rm Out}}{\eta(V_{\rm Batt}, I_{\rm Out}) V_{\rm Batt}} \tag{1}$$



```
entity battery is
   port(I_Batt : in amps; update : in std_logic; V_Batt : out real);
end battery;

architecture behavior of battery is
begin
   V_Batt <= PWL(V_C) + V_Cell_Temp - R_Int * I_Batt;
   -- V_Cell_Temp is 0.0 if no second order effects are considered
   Compute_V_C : process (I_Batt, update, V_Lost)
   begin
      cap_act := (cap_act - I_BattOld * (NOW - chgt));
(*)   V_C <= (cap_act/ cap_i - V_Lost);
      I_BattOld = I_Batt;
      chgt = NOW;
   end process;
   Compute_V_Lost : process (I_Batt, update, Compute)
   begin
      V_r := I_BattOld/ CAPACITY;
(**)  V_Rate := (V_RateOld - V_r) * exp(-(NOW - chgt)/(R_f * C_f)) + V_r;
      V_Lost:= PWL(V_Rate);
      if (I_Batt'event) then
       V_RateOld := V_Rate;
       Compute <= '1' after (τ / 5.0),
                '0' after (τ / 5.0 * 2.0),
                '1' after (τ / 5.0 * 3.0),
                '0' after (τ / 5.0 * 4.0),
                '1' after (τ / 5.0 * 5.0),
                '0' after (τ / 5.0 * 6.0);
      end if;
   end process;
end behavior;
```

Fig. 4. High-level VHDL code of the battery model.

which is obtained directly from the definition of efficiency. This black-box model has two key advantages. First, it can be simulated very efficiently, because all information on internal high-frequency switching are abstracted away. Second, it can be inferred directly from data-sheet information and does not require disclosure of the internal structure of the converter. For these reasons, we adopted the black-box model.

III. Discrete-Time Power Supply Model

This section describes a discrete-time power supply model that can be easily implemented within any system-level design environment. For the sake of concreteness, we will assume that system-level simulation is performed in VHDL. Hence, VHDL is the language of choice for the implementation of our abstract model.

A. Battery

The battery is defined as a VHDL entity detailed in Fig. 4. Its interface has two inputs: I_{Batt}, representing the current absorbed by the dc–dc converter, and *update*, a periodical signal used to update the values in the model. The output of the `battery` entity is V_{Batt}, which represents the voltage supplied by the cell to the dc–dc converter (see Fig. 1). Signals have analog values. A specialized resolved type has been defined for current signals, with the purpose of allowing multiple current loads connected to the same current signal. The resolution function simply sums over all currents.

The internal structure is based on the circuit-level model of Fig. 1 and consists of two concurrent, communicating processes. The first one (`Compute_`V_C) computes the value of node V_C in Fig. 1, the instantaneous state of charge of the battery (taking into account losses due to high discharge rate).

The second process (`Compute_`V_{Lost}) computes the value of V_{Lost}, i.e., it implements the low-pass filter shown in Fig. 1. The output voltage of the battery V_{Batt} is a function of V_C. It is implemented in VHDL with a continuous assignment: $V_{\mathrm{Batt}} = F(V_C)$, where F is realized by a LUT with linear interpolation (PWL).

The main challenge in the implementation of the voltage update processes is caused by the fact that they model voltages, which are changing in a continuous fashion over time, thus, some form of discretization is required to simulate them in an event-driven setting. To address this challenge, we implemented an autonomous source of events (signal *update* in Fig. 4) that generates events at a fixed frequency.

The state of charge V_C and the value of V_{Rate} are updated when the autonomous source generates an event. The change in SOC is obtained by integrating the differential equations of the continuous-time model over the update period. Notice that, in our case, the differential equations governing the evolution of V_C and V_{Rate} over time can be integrated exactly.

The differential equation for V_C, with fixed $V_{\mathrm{Lost}}(V_{\mathrm{Rate}})$ and I_{Batt} is

$$C\frac{d(V_C + V_{\mathrm{Lost}}(V_{\mathrm{Rate}}))}{dt} = -I_{\mathrm{Batt}} \tag{2}$$

which has the solution $V_C(t) = V_C(0) - V_{\mathrm{Lost}}(V_{\mathrm{Rate}}) - I_{\mathrm{Batt}} \cdot t/C$. The equation above appears in the code in Line (∗), where *cap_i* and *cap_act* correspond to C and $(V_C(0) \cdot C - I_{\mathrm{Batt}} \cdot t)$, respectively. Notice that we have assumed $V_C(0) = 1$ and $C = 3600 \cdot$ *CAPACITY*, as in [9].

The differential equation for V_{Rate} with fixed V_r is

$$V_{\mathrm{Rate}} + C_f R_f \frac{dV_{\mathrm{Rate}}}{dt} = V_r \tag{3}$$

which can be integrated in closed form, giving $V_{\mathrm{Rate}}(t) = (V_{\mathrm{Rate}}(0) - V_r)e^{-t/R_f C_f} + V_r$. This equation is used in Line (**) of Fig. 4, where $V_{\mathrm{Rate}}(0)$ corresponds to V_{RateOld}.

One last complication arises if we observe that the two equations are not mutually independent, and that both depend on I_{Batt} (i.e., the current absorbed by the load). To solve this problem, whenever I_{Batt} changes, new integration constants are computed for the two differential equations. The value of $V_{\mathrm{Rate}}(0)$ is known because the voltage across a capacitor must be a continuous function. Consequently, the value of $V_C(0)$ is known as well. In other words, the battery model reacts to an event on I_{Batt} by recomputing the boundary conditions in the solutions of the differential equations. After the change, the voltage update rule is re-established for both V_C and V_{Rate}. Note that the functional dependence of V_C from V_{Rate} imposes the re-evaluation of V_C even when V_{Rate} changes. The coupling of the two differential equations is explicit in the VHDL model of Fig. 4. The sensitivity list of process Compute_V_C contains V_{Lost}, which is computed in process Compute_V_{Lost}, as a function of V_{Rate}. Additionally, both processes are re-executed on I_{Batt} events, which impose new boundary conditions.

The changes in V_{Rate} (and V_{Lost}) in response to a variation in I_{Batt} are not instantaneous, but follow a transient with the time constant of the battery's low-pass filter $\tau = R_f C_f$ (for real-life batteries this interval is in the order of one second). Hence, a $\Delta t = \tau/5$ is sufficient to model the transient behavior of node V_{Rate} in response to changes of I_{Batt}. Observe

```
Compute_V_Cell_Temp : process(I_Batt, Compute_Temp)
begin
    V_70 := (V_e + I^2_BattOld * R_Int * θ);
    V_Temp := (V_TOld - V_70) * exp(-(NOW - chgt)/(R_Th * C_Th)) + V_70;
    if (I_Batt'event) then
     V_TOld := V_Temp;
     Compute_Temp <= '1' after (τ_Th / 5.0),
                     '0' after (τ_Th / 5.0 * 2.0),
                     '1' after (τ_Th / 5.0 * 3.0),
                     '0' after (τ_Th / 5.0 * 4.0),
                     '1' after (τ_Th / 5.0 * 5.0),
                     '0' after (τ_Th / 5.0 * 6.0);
    end if;
end process;
```

Fig. 5. High-level VHDL code of the secondary effects.

that we do not need to generate events for modeling the transient behavior of V_{Rate} if there are no new events on I_{Batt} and the time after the last I_{Batt} event is $t \geq 4\tau$. In this case, V_{Rate} is very close to its asymptotic value, and it does not need to be updated. This behavior is obtained in VHDL by scheduling a finite number of events of signal `Compute` in the future, whenever I_{Batt} changes, and by including *Compute* in the sensitivity list of process `Compute` V_{Lost}. The spacing of these events in time is proportional to τ. If V_{Rate} has settled and there is no I_{Batt}, state of charge updates are controlled exclusively by *update* events.

Concerning the second-order effects of Section II-A2, temperature is taken into account through a process, shown in Fig. 5, that computes the value of V_{Cell_Temp} in response to a variation of I_{Batt}. As for V_{Rate} in (3), V_{Cell_Temp} is obtained by solving the differential equation expressing its value and it is then added (with its relative sign) to the battery output voltage V_{Batt}.

Finally, the discrete-time model of the internal resistance effect is quite straight-forward and consists of an additional VHDL statement that subtracts $R \cdot I_{\text{Batt}}$ from the battery output voltage V_{Batt}.

B. dc–dc Converter

The dc–dc converter is modeled as a VHDL entity, as shown in Fig. 6, with two input ports, I_{Out} (coming from the load) and V_{Batt} (coming from the battery) and one output port I_{Batt} (connected to the battery). There is no V_{Out} port because the output voltage is kept constant by the dc–dc converter. The value of I_{Batt} is computed with (1). Efficiency η is a function of both V_{Batt} and I_{Out}. Its value is obtained by lookup table and linear interpolation (`PWL`). The computation of I_{Out} is stateless and it is implemented in VHDL as a continuous signal assignment with zero delay. The interaction between dc–dc converter and the battery is based on pairs of (V_{Batt}, I_{Batt}) events. Whenever the battery generates a new V_{Batt} event, the dc—dc converter responds with an I_{Batt} event with zero delay. The generation of zero-delay event loops is avoided because V_{Batt} cannot change in zero time in response to a I_{Batt} variation (remember that V_{Batt} is a function of voltages across capacitors that cannot change in zero time). A new I_{Batt} event is also generated in response to an I_{Out} event; by doing this, changes on the load are propagated to the battery.

In summary, the models for battery and dc–dc converter have limited complexity; in addition, they generate a limited number of events over a battery life (a few events per second, in the worst case). Therefore, they are well suited to work with system-level descriptions without sizable simulation overhead.

```
entity DCDC is
   port(I_Out : in amps; I_Batt : out amps; V_Batt : in real);
end DCDC;

architecture behavior of DCDC is
begin
   Efficiency <= PWL(I_Out);
   I_Batt <= (V_Out * I_Out/Efficiency)/V_Batt;
end behavior;
```

Fig. 6. High-level VHDL code of the dc–dc converter.

IV. Model Validation

In this section, we present data concerning the validation of the discrete-time model of Section III against the continuous-time model of Section II. For the experiments we consider Lithium–Ion batteries, since they have a dominant position as the chemistry of choice for notebook and laptop computers. The main reason for this fact is that Li–Ion batteries have the best volumetric energy density (as well as the best gravitometric energy density) among all competing cells. In simple terms, Li–Ion batteries provide more energy than other cells for a given battery volume (or weight). On the other hand, Li–Ion batteries have higher cost than traditional secondary batteries, like Nickel–Cadmium, because their fabrication technology is still quite expensive and not fully optimized. High performance and cost have limited the market expansion of Li–Ion batteries to high-end products, where energy drain is high and high capacity is required to obtain acceptable lifetime with low weight. Safety of Li–Ion batteries is an additional issue. If Li–Ion cells are overcharged, they may be damaged, leak, or even explode. Hence, smart battery chargers are required, thereby further increasing ownership costs. Li–Ion battery producers are steadily improving their technologies, both by cell chemistry optimization and by embedding low-cost electronic controllers within the battery package in order to reduce the external support needed for safe battery operation.

Currently, Li–Ion batteries are expanding their market dominance in the portable computer arena, and thanks to lower production costs, they are also extending their competitiveness to low-end products.

The ultimate target of our experiments is to show how closely the discrete-time model is able to track the continuous-time one. To this purpose, we have simulated both models under various conditions, corresponding to a set of different output loads, characterized by different maximum currents and time-domain behaviors. More specifically, we have considered a total of ten types of current load stimuli:

1) Type *CC*: Three constant current loads of magnitude 0.1, 0.2, and 1.0 A.
2) Type *SW*: Three square waves with 50% duty-cycle, with average value of 1.0 A and different current levels: {(1.1A,0.9A),(1.3A,0.7A),(1.8A,0.2A)}.
3) Type *STEP*: Periodic waveform with six different levels of current loads (average value 1.083 A).
4) Type *SP*: Three pulses with durations 1, 5, and 20 s.

TABLE I
MODEL VALIDATION RESULTS (LI–ION BATTERY)

Current Load	*LT* *HSpice*	*LT* *VHDL*	ΔLT [%]	*RMSRE* [%]
CC1	51700	51800	0.193	0.048
CC2	26000	26000	0.000	0.080
CC4	5050	5100	0.990	0.365
SW1	5010	4990	0.399	0.271
SW2	4930	4920	0.203	0.347
SW4	4790	4710	1.670	1.108
STEP	4560	4550	0.219	0.804
SP1	–	–	–	0.949
SP2	–	–	–	1.311
SP4	–	–	–	1.667
Average			0.525	0.695

Table I reports the results of the simulations. For each type of load, we give the battery lifetime (*LT*), in seconds, obtained by simulating both the HSpice model (column *HSpice*) and the VHDL model (column *VHDL*), and the corresponding relative errors (column ΔLT). Column *RMSRE* reports the root mean square relative error of the HSpice versus the VHDL battery output voltage waveforms.

Lifetime estimates are very accurate (error, on average, is 0.525%). The void entries for the *SP* loads are due to the fact that the maximum pulse duration is 60 s, for which only a negligible lifetime degradation was observed. The purpose of the *SP* loads is to evaluate the short-time behavior of the model. Battery output voltage results are also very satisfactory, since the *RMSRE* ranges from 0.048% to 1.667% (0.695% on average), depending on the type of load.

The errors are mainly due to the intrinsic difference of the implementation of the two models. In the VHDL model, the nonlinear relationship between the two quantities is obtained by piece-wise linear approximation of values tabulated in an array. Conversely, in the HSpice model the interpolation of the tabulated values is obtained by imposing the continuity of the first derivative.

The results are in agreement with the expected behavior of the battery/dc–dc converter system. For example, the comparison of the *CC4* type (1.0 A constant current load) to the *SW* loads (1.0 A average, but different levels) clearly shows that battery lifetime is strongly affected by the current variations, and not only by average current values. Periodically changing the load from 0.2 A to 1.8 A (SW4) results in a 5% decrease in lifetime with respect to a constant 1.0A load (4790 s versus 5050 s).

To better understand the transient behavior of the two models, in Figs. 7 and 8 we present two HSpice diagrams containing three curves each: The battery voltage, V_{Batt}, represented by the solid line with crosses, the battery discharge current, I_{Batt}, represented by the dashed line with triangles, and the output current of the dc–dc converter, I_{Out}, represented by the dashed line with circles. The diagrams for the discrete-time model are almost coincident with the ones shown in the figures and are thus not reported. The scale on the left is relative to voltage waveforms, and the one on the right is relative to the current. Notice that the curve for I_{Out} is shown as a negative current in order to increase readability.

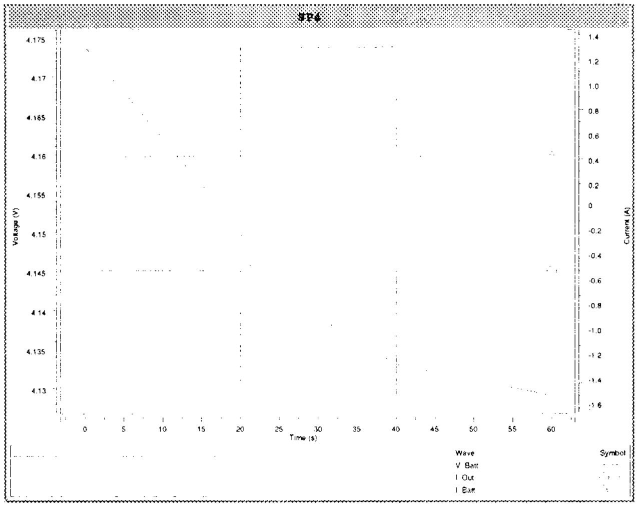


Fig. 7. Voltage and current plots for *SP4*.

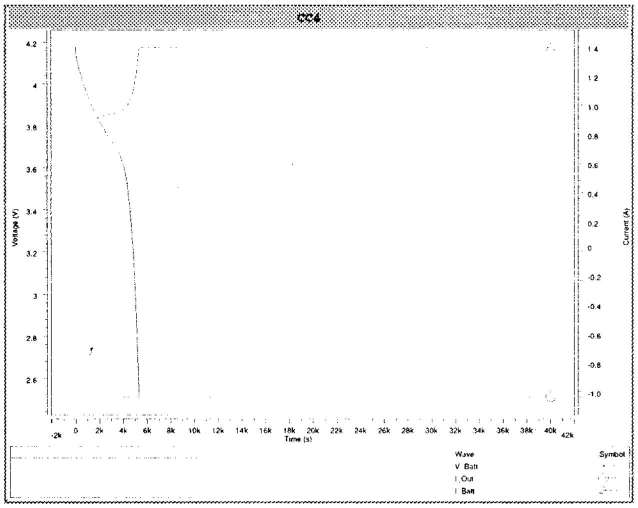


Fig. 8. Voltage and current plots for *CC4*.

Fig. 7 refers to the *SP4* stimulus. We can notice how the output voltage follows the current transitions; in particular, at the beginning and for the whole duration of the pulse, the output voltage decreases quite steeply; after the pulse is finished, the voltage exhibits a smoother decrease. Obviously, the output current of the dc–dc converter is insensitive to voltage variations.

Fig. 8 refers to the *CC4* stimulus. In this case, we can see that under the effect of a 1.0 A constant load current, the output voltage decreases very quickly; this affects the current provided by the battery, that tends to increase accordingly.

Figs. 9 and 10 show the impact of second-order effects (i.e., temperature and internal battery resistance, respectively) on the battery output voltage. In particular, the plot of Fig. 9 reports the output voltage at different environment temperatures (curves, from bottom to top, correspond to temperature values of -20 °, 0 °, 20 °, and 40 °C). As it can be clearly observed, the lower the temperature, the lower the output voltage, hence, the shorter the lifetime. This is due to the fact that the dc–dc converter draws higher currents in response to lower input voltages.

The plot of Fig. 10 shows how the internal resistance affects the output voltage. In particular, the top curve refers to the case

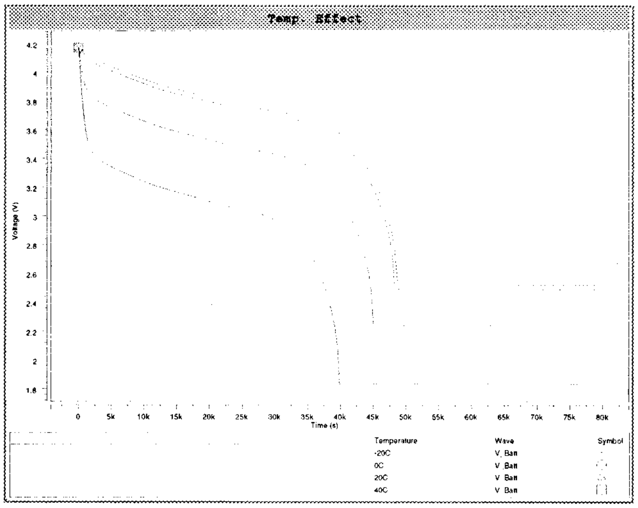


Fig. 9. Temperature effects.

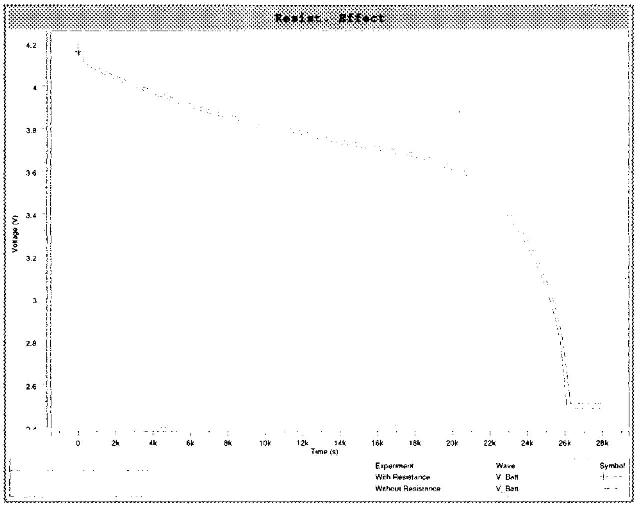


Fig. 10. Internal resistance effects.

in which the internal resistance is neglected, while the bottom curve considers the resistive effect. As expected, taking into account internal resistance yields a lower output voltage.

V. Model Extension to Other Battery Types

The results of Section IV refer to a Lithium–Ion battery. In this section, we prove that the proposed discrete-time model can be easily and conveniently adapted to other types of batteries. We start by briefly outlining the specific features of the battery types we consider; then, we report comparative simulation data.

A. Nickel–Cadmium Battery

Nickel–Cadmium batteries have been on the market for a long time. They are rechargeable and they have a well-established technology with very low cost. Also, they are robust and do not need maintenance. Their performance at low temperature is good and they perform fairly well under high-rate discharge. Ni–Cd batteries dominate the market for low-cost portable electronic appliances, but they are losing ground with respect to more energy-efficient chemistries (such as Li–Ion) for high-end applications.

B. Alkaline Battery

Alkaline batteries are an example of primary batteries. Their cost is much lower than that of rechargeable batteries, but they cannot be re-charged. In the portable electronic market they are a viable alternative to Ni–Cd batteries for low-cost applications. Several studies have been performed on how to choose between primary and secondary batteries for a given application [13]. A primary battery will be the best choice if the power requirement is low or in applications where the battery will be used infrequently over a long period of time. For applications, where the battery will be used frequently, the choice depends on user preference. In some cases, the user will opt for the convenience of a primary battery and the freedom from the charger. In other cases, the user may choose for the possibly lower operating cost of rechargeable batteries. In general, for high-end applications (such as laptop computers), primary batteries are not a viable alternative.

From the electrical point of view, alkaline batteries show a remarkable increase in cell resistance when the battery is close to full discharge. Furthermore, the energy efficiency of alkaline batteries decreases sharply at high discharge currents, independently from the capacity of the battery. In other words, these battery do not scale well with size for what concerns the peak output current.

C. Lead–Acid Battery

Lead–acid batteries are mainly used for vehicles and are very cost-effective for large cells. Their use in portable equipment has been limited to niche markets. They become a viable alternative to Ni–Cd when relatively large batteries are needed, as in the case of power tools. Furthermore, they tend to perform better than Ni–Cd batteries at high temperatures. Recently, a new fabrication technology, called *lead–acid thin metal film* has shown good promise. These new lead–acid batteries seem capable to deliver a large amount of energy in a relatively short time, but they are hard to fabricate and mass production of thin metal film batteries is still not viable.

D. Simulation Data

We have run the same experiments as in Section IV using typical parameters, taken from [9], for the three battery types just described. This with the objective of validating the robustness of the discrete-time model, for both accuracy and efficiency, when different batteries and different load conditions are chosen.

Table II summarizes the results of the simulations for the Nickel–Cadmium, Alkaline, and Lead–Acid batteries, respectively. The meaning of the columns is the same as in Table I: LT indicates the battery lifetime, ΔLT is the relative error in battery lifetime between the two models, and RMSRE is the root mean square relative error of the battery output voltage waveforms as generated through HSpice and VHDL simulations.

The data prove clearly that the model scales well with respect to different battery types: As for Li–Ion batteries (see Table I), lifetime estimation errors are still very small (the average over

TABLE II
MODEL VALIDATION RESULTS (OTHER TYPES OF BATTERIES)

Current Load	*Nickel-Cadmium*				*Alkaline*				*Lead-Acid*			
	LT		ΔLT	*RMSRE*	*LT*		ΔLT	*RMSRE*	*LT*		ΔLT	*RMSRE*
	HSpice	*VHDL*	[%]	[%]	*HSpice*	*VHDL*	[%]	[%]	*HSpice*	*VHDL*	[%]	[%]
CC1	20050	20050	0.000	0.265	59800	58650	1.923	0.313	59950	59900	0.083	0.006
CC2	10150	10100	0.493	0.428	25750	26250	1.942	0.503	33900	33950	0.147	0.284
CC4	1900	1900	0.000	0.787	1600	1700	6.250	0.746	4600	4650	1.087	0.362
SW1	1910	1910	0.000	0.070	5570	5690	2.154	0.471	4450	4490	0.899	1.226
SW2	1870	1880	0.535	0.099	5440	5510	1.287	2.305	4310	4340	0.696	1.321
SW4	1810	1820	0.552	0.190	4820	4850	0.622	2.859	3700	3680	0.540	2.144
STEP	1750	1750	0.000	0.149	5600	5540	1.071	2.296	3430	3420	0.291	0.604
SP1	–	–	–	0.050	–	–	–	0.296	–	–	–	2.967
SP2	–	–	–	0.039	–	–	–	0.313	–	–	–	2.938
SP4	–	–	–	0.029	–	–	–	0.240	–	–	–	2.118
Average			0.226	0.211			2.434	1.034			0.535	1.397

all battery types is 1.065%). Voltage estimation results are also in line with the results of Section IV since the average RMSRE ranges from 0.211% to 1.397%, depending on the type of battery (the total average is 0.880%).

VI. MODEL EXPLOITATION DURING DESIGN EXPLORATION

In this section, we demonstrate the importance of a system-level battery model in the context of design space exploration. The investigation is carried out on two realistic digital applications. The first one is a personal digital assistant (PDA) with power management capabilities. Our objective here is, on the one hand, to illustrate that the adoption of an accurate and efficient high-level battery model is key for properly quantifying the impact of dynamic power management (DPM) on battery lifetime. On the other hand, we are interested in proving that high-level battery models can be very helpful to system engineers who are in charge of selecting the best-suited battery for a given system.

The second application we consider is an MPEG 2-Layer 3 (MP3) digital audio player. Our objective, in this case, is to confirm the results obtained for the PDA regarding the usage of the battery model to assess the benefits of DPM on battery lifetime extension. In addition, we aim at illustrating how the capability, offered by the high-level model, of monitoring the SOC of the battery during system simulation can be exploited to develop DPM policies that are not simply work-load dependent, but also battery-dependent.

A. Case Study I: Personal Digital Assistant (PDA)

We consider the system-level description of a real-life personal digital assistant (PDA) with power management capabilities, whose conceptual block diagram, depicted in Fig. 11, is inspired to commercial products such as the Palm IIIx organizer [16]. In the bottom-left corner of the diagram the model of the battery system can be clearly identified.

The PDA consists of a CPU, with an embedded power management unit that can be selectively disabled; a memory block *MEM* (RAM and flash); some glue logic implemented as FPGA's (*Xilinx*); an LCD display. Two other blocks are contained in the system: STATIC, that emulates the static power dissipation (i.e., not power-manageable) of the PDA, and a block REQ that dispatches the incoming requests from block DRIVER, that emulates the user. Components are abstractly described using a state-based model similar to that described in [15]. States correspond to modes of operation, and transitions are taken in response to external events. The system model is built as an interconnection of behavioral state machines that communicate using abstract events.

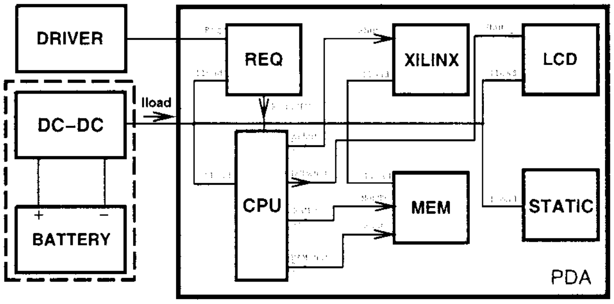


Fig. 11. Block diagram of the PDA.

The power model of the system components associates a current load with each mode of operation. During simulation, components change state of operation, thereby changing the current load experienced by the power supply. Every block of the PDA has two signals: *Iload*, that denotes the current drawn from the power supply, and *Shut*, driven by the CPU, that is used to indicate when the module is to be shut down. The model of the system written in VHDL can be simulated together with the power supply model for estimating battery lifetime.

1) Impact of DPM on Battery Lifetime: In a first experiment, we have applied a sequence of input requests to mimic the typical usage of the PDA for an approximate duration of one day. We measured the battery output voltage under the application of such sequence, both with and without power management. Needless to say, system simulation that accounts for the presence of battery and dc–dc converter was possible thanks to the availability of our model. In fact, HSpice simulation of the low-level synthesized description of the whole system would have been just infeasible (the netlist would be too complex and the duration of the simulation too long).

Results are shown in Fig. 12, and refer to the case of a Li–Ion battery with nominal capacity of $0.5Ahr$; as expected, power

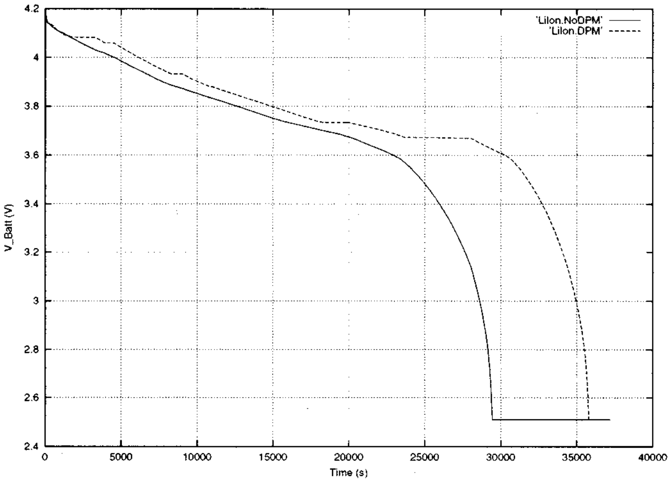


Fig. 12. Impact of DPM on Lithium–Ion battery lifetime.

managing the PDA extends battery life of about 26.4% (from 28 400 s to 35 900 s). In case the battery is replaced by a constant power supply, on the other hand, average power savings would be around 34.3%, indicating that average power reduction and battery lifetime extension are not numerically the same.

The importance of a detailed battery model for accurate lifetime estimation is further confirmed by the following experiment. Drawing from the battery a constant current exactly equal to the average value $I_{\rm avg}^{DPM}$ of the current drawn in the power-managed case yields an increase in estimated battery life of 17.1% (from $35\,900$ s to $42\,039$ s). This result shows that the model is extremely useful to evaluate the actual impact of power management; the time-dependent load determined by DPM cannot be accurately estimated by simply applying the relative average load to the battery *off line*. In fact, assuming an equivalent constant load may lead to sensible overestimates of battery life.

2) Battery Exploration: We have analyzed the performance of the various battery types described in Section V. To compare the behavior of the four batteries we have individually connected each of them to the PDA, we have applied at the inputs of the system a given service request trace, and measured the output voltage under the application of such sequence, both with and without power management. Obviously, all batteries had the same nominal capacity, that is, $0.5Ahr$.

Results for the Lithium–Ion battery have been shown in Fig. 12; results for the Nickel–Cadmium, Alkaline and Lead–Acid batteries are summarized in Table III.

From the experimental data we can observe that the Lithium–Ion battery provides the longest lifetime. However, the longest lifetime extension due to power management occurs when the Nickel–Cadmium battery is chosen. Regarding the Alkaline battery, we have noted that it presents a large variation (about 0.5 Volts) of the output voltage in response to current variations. This phenomenon is due to the high internal resistance and may impair the functionality of the dc–dc converter.

For completeness, also the Lead–Acid battery has been considered for comparison, although it has very different size, weight, and application domain with respect to Li–Ion, Ni–Cd and Alkaline batteries.

TABLE III
BATTERY EXPLORATION SUMMARY

Battery Type	*LT*		ΔLT
	No DPM	*DPM*	[%]
Lithium-Ion	28400	35900	26.4
Nickel-Cadmium	24960	33100	32.6
Alkaline	26000	32075	23.3
Lead-Acid	26000	32800	26.1

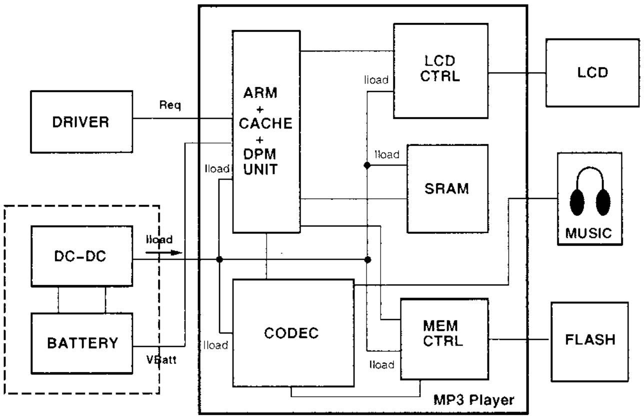


Fig. 13. Block diagram of the MP3 player.

B. Case Study II: MPEG 2-Layer 3 (MP3) Digital Audio Player

We consider the system-level description of an MPEG 2-Layer 3 (MP3) Digital Audio Player, whose block diagram, shown in Fig. 13, meets the specification of a commercially-available product by Diamond [17]. System components can be power-managed through signals issued by a DPM unit in accordance with the selected DPM policy.

The MP3 player consists of a core processor (*ARM720T*) with 8 KB of cache and a DPM unit, 32 KB of static RAM (*SRAM*), an LCD controller (*LCD CTRL*), an MP3 codec (*CODEC*) and a memory controller (*MEM CTRL*); all these devices, together with some additional functionalities (e.g., interrupt controller), are contained in the EP7209 Ultra-Low-Power Audio Decoder System-on-Chip by Cirrus Logic [18]. External to the system there are a 32 MB flash memory (*FLASH*), the battery subsystem, a block (*DRIVER*) that emulates the inputs provided by the user, an LCD display and an headset.

1) Workload-Driven and Battery-Driven DPM: The first DPM policy we have experimented with is *workload-driven* (WDDPM), that is, the criterion used by the DPM unit to decide whether a component of the system needs to enter its low-power mode is based on a time-out mechanism. This policy is similar to the one we adopted for the PDA of Section VI-A1 and does not consider the state of charge of the battery.

A more appealing policy, which fully exploits the capability of the battery model in providing on-line, battery SOC monitoring, is based on the ability of the MP3 codec in trading off the quality of the decompressed audio signal for a reduced power required by the decompression. More specifically, the DPM unit monitors the state of charge of the battery and selects the operation mode of the codec accordingly. When the SOC is above 50% (which corresponds to a battery voltage around 3.7 V), full

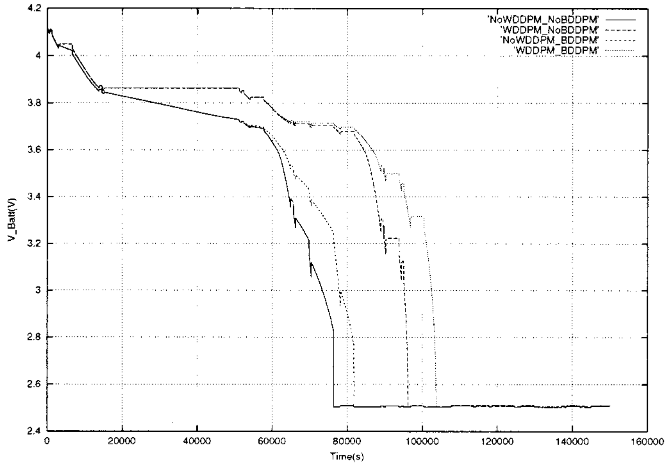


Fig. 14. Battery lifetime with different DPM policies.

TABLE IV
BATTERY LIFE-TIME WITH DIFFERENT DPM POLICIES

	No WDDPM	*WDDPM*
No BDDPM	66683	89570
BDDPM	74340	100457

decompression is performed (44 KHz samples/s, 128 kb/s, 87 mW). When it is between 50% and 20% (that is, $V_{\text{Batt}} \approx 3.3\,\text{V}$), a looser decompression rate is chosen (22.05 *KHz* samples/s, 64 kb/s, 50 mW), thus implying a lower quality of the audio signal but also a reduced power consumption (approximately 55% of power required by full decompression). Finally, when the SOC is smaller than 20%, the codec enters its sleep state (dissipation of 1 mW due stand-by currents). We call this type of DPM policy *battery driven* (BDDPM).

In order to study the impact on battery lifetime of the two DPM policies described above, we have supplied a few audio files as input to the MP3 player, interleaved with idle periods in order to emulate a typical usage of the appliance. We note that WDDPM and BDDPM are not mutual exclusive and can thus be used simultaneously. The results we have obtained are depicted in Fig. 14 and, in condensed form, in Table IV (battery lifetimes are expressed in seconds); they refer to the case of a Li–Ion battery with nominal capacity of $0.5Ahr$ (the same we used for the PDA experiments).

As expected, all DPM policies extend the battery lifetime, but the advantage obtained using a combination of the two policies is as high as 50.6%, while it is only 34.3% for WDDPM alone and 11.5% for BDDPM alone. Obviously, considering the usage of a battery-driven policy has been possible thanks to the availability of the high-level battery model.

We complete our experimentation with the MP3 player by replacing the battery with a constant power supply. In this case, the estimated power savings due to the usage of the WDDPM policy is around 44.2%. This value is sensibly higher than the percentage of lifetime extension we observe by applying the same policy when the system is battery-operated (34.3%); this result confirms the conclusion we have drawn for the case of the PDA application: Neglecting the presence of the battery in the system may cause mistakes in the estimation of the advantages that power optimization solutions may guarantee.

VII. CONCLUSION

We have presented a discrete-time model for batteries and power conversion circuits to be used in system-level design environments. The model is efficient enough to enable simulation-based battery lifetime estimation. Experimental results show that the accuracy of the estimates obtained by the discrete-time model is very close to that of Spice-level simulations.

We have first validated the model by assuming Lithium–Ion battery technology. Then, we have extended it to the case of other battery types, including both nonrechargeable and rechargeable cells. This is a first step toward the construction of a "battery library" that can be employed by designers to compare several chemistries and cell types when the task is the selection of the most appropriate battery.

Examples of design space exploration carried out on realistic systems have been provided to demonstrate the usefulness of the proposed discrete-time battery model in the context of system-level design.

REFERENCES

[1] E. Macii, M. Pedram, and F. Somenzi, "High-level power modeling, estimation, and optimization," *IEEE Trans. Comput.-Aided Design*, vol. 17, pp. 1061–1079, Nov. 1998.

[2] C. Smaltz, *et al.*, "Extending battery life of portable devices," in *Proc. IEEE WESCON'93*, San Francisco, CA, Sept. 1993, pp. 384–387.

[3] T. Martin and D. Sewiorek, "A power metric for mobile systems," in *Proc. ISLPED-96: ACM/IEEE Int. Symp. Low-Power Electronics and Design*, Monterey, CA, Aug. 1996, pp. 37–42.

[4] M. Pedram and Q. Wu, "Battery-powered digital CMOS design," in *Proc. DATE-99: IEEE 1999 Design Automation and Test in Europe*, Munich, Germany, Mar. 1999, pp. 72–76.

[5] M. Doyle, T. F. Fuller, and J. Newmann, "Modeling of galvanostatic charge and discharge of the lithium/polymer/insertion cell," *J. Electrochem. Soc.*, vol. 140, no. 6, pp. 1526–1533, June 1993.

[6] D. P. Arduini *et al.*, "Extending battery life and reliability," in *Proc. 1998 Int. High Frequency Power Conversion Conf. Exhibit*, Santa Clara, CA, Nov. 1998, pp. 245–256.

[7] C. McKinnon, "Battery selection isn't an afterthought," *Portable Design*, vol. 4, no. 4, Apr. 1998.

[8] D. Linden *et al.*, "Battery options for portable applications," in *Proc. IEEE WESCON'92*, Anaheim, CA, Nov. 1993, pp. 534–537.

[9] *Simple PSpice Models Let You Simulate Common Battery Types*, pp. 117–132, 1993.

[10] M. Glass, "Battery electro-chemical nonlinear/dynamic SPICE model," in *Proc. Energy Conversion Engineering Conf.*, Washington, DC, Aug. 1996, pp. 292–297.

[11] S. Gold, "A PSPICE macro-model for lithium-ion batteries," in *Proc. 12th Annu. Battery Conf. Applications and Advances*, Long Beach, CA, Jan. 1997, pp. 215–222.

[12] R. Erickson, *Fundamentals of Power Electronics*. New York: Chapman Hall, 1997.

[13] *Handbook of Batteries*, II ed., D. Linden, Ed., McGraw Hill, New York, 1996.

[14] M. Pedram and Q. Wu, "Design considerations for battery-powered electronics," in *Proc. DAC-36: ACM/IEEE Design Automation Conf.*, New Orleans, LA, June 1999, pp. 861–866.

[15] L. Benini, R. Hodgson, and P. Siegel, "System-level power estimation and optimization," in *Proc. ISLPED-98: ACM/IEEE Int. Symp. Low-Power Electronics and Design*, Monterey, CA, Aug. 1998, pp. 173–178.

[16] Palm, "Palm IIIx,", http://www.palm.com/products/palmiiix.

[17] Diamond Multimedia, "Rio PMP300,", http://www.diamondmm.de/eng/products/rio/rio300.htm.

[18] Cirrus Logic. EP7209 Ultra-low-power audio decoder SoC. [Online]. Available: http://www.cirrus.com/products/overviews/ep7209.html.

Luca Benini received the Dr.Eng. degree in electrical engineering from Università di Bologna, Italy, in 1991, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1994 and 1997, respectively.

Currently, he is an Assistant Professor of electrical engineering at the Università di Bologna. His research interests are in all aspects of computer-aided design of digital circuits, with special emphasis on low-power applications.

Giuliano Castelli received the Dr.Sc. degree in physics from Università di Trieste, Italy, in 1996. He is currently working toward the Ph.D. degree at the Dipartimento di Automatica e Informatica, Politecnico di Torino, Italy.

Since October 1998, he has been with the Dipartimento di Automatica e Informatica, Politecnico di Torino. His actual research interests are on modeling and design and optimization of battery-powered digital systems.

Alberto Macii received the Dr.Eng. and the Ph.D. degrees in computer engineering from Politecnico di Torino, Italy, in 1996 and 2001, respectively.

Currently, he holds a Postdoctoral position at the same institution. His research interests include several aspects of the development of algorithms, methodologies, and tools for the design of low-power digital systems.

Enrico Macii received the Dr.Eng. degree in electrical engineering from the Politecnico di Torino, Italy, the Dr.Sc. degree in computer science from the Università di Torino, and the Ph.D. degree in computer engineering from the Politecnico di Torino, in 1990, 1991, and 1995, respectively.

From 1991 to 1994, he was an Adjunct Faculty at the University of Colorado at Boulder. Currently, he is an Associate Professor of computer engineering at the Politecnico di Torino. He is an Associate Editor of the *ACM Transactions on Design Automation*. His research interests include several aspects of the computer-aided design of integrated circuits and systems, with particular emphasis on synthesis, optimization and formal verification.

Dr. Macii is an Associate Editor of the IEEE Transactions on Computer-Aided Design. He was the Technical Program cochair of the *IEEE Alessandro Volta Memorial Workshop on Low Power Design*, in 1999 and the Technical Program Co-Chair and the General Chair of the *ACM/IEEE International Symposium on Low Power Electronics and Design* in 2000 and 2001, respectively.

Massimo Poncino received the Dr.Eng. degree in electrical engineering in 1989 and the Ph.D. degree in computer engineering in 1993, both from the Politecnico di Torino, Italy.

From 1993 through 1994, he was a Visiting Faculty Member at the University of Colorado at Boulder. Currently, he is an Assistant Professor at the Politecnico di Torino. His research interests include synthesis, verification, simulation, and testing of digital circuits and systems.

Riccardo Scarsi received the Dr.Eng. degree in electrical engineering and the Ph.D. degree in computer engineering, both from Politecnico di Torino, Italy, in 1997 and 2001, respectively.

Currently, he holds a Postdoctoral position at the same institution. His research interests include several aspects of the development of algorithms, methododologies, and tools for the design of low-power digital systems.