Energy Management of DVS-DPM Enabled Embedded Systems Powered by Fuel Cell-Battery Hybrid Source*

Jianli Zhuo
Dept. of EE
Arizona State University
Tempe, AZ, USA
jianli@asu.edu

Chaitali Chakrabarti
Dept. of EE
Arizona State University
Tempe, AZ, USA
chaitali@asu.edu

Naehyuck Chang
School of CSE
Seoul National University
Seoul, Korea
naehyuck@snu.ac.kr

ABSTRACT

Dynamic voltage scaling (DVS) and dynamic power management (DPM) are the two main techniques for reducing the energy consumption of embedded systems. The effectiveness of both DVS and DPM needs to be considered in the development of an energy management policy for a system that consists of both DVS-enabled and DPM-enabled components. The characteristics of the power source also have to be explicitly taken into account. In this paper, we propose a policy to maximize the operational lifetime of a DVS-DPM enabled embedded system powered by a fuel cell-battery (FC-B) hybrid source. We show that the lifetime of the system is determined by the fuel consumption of the fuel cell (FC), and that the fuel consumption can be minimized by a combination of a load energy minimization policy and an optimal fuel flow control policy. The proposed method, when applied to a randomized task trace, demonstrated superior performance compared to competing policies based on DVS and/or DPM.

Categories and Subject Descriptors:
C.3 **[Special-purpose and application-based systems]:** - *real-time systems and embedded systems*

General Terms: Algorithms, Design

Keywords: DPM, DVS, fuel cell, hybrid power, embedded system

1. INTRODUCTION

Energy minimization has always been a critical design criteria for portable embedded systems. DVS (dynamic voltage scaling) and DPM (dynamic power management) are the two main techniques to reduce the energy consumption for such systems. DVS saves energy by operating the system at a lower frequency and thus a lower voltage, while DPM saves energy by putting the system into a lower power state when the idle time is long enough.

There has been lots of work in the areas of DVS and DPM. Most DVS work is associated with task scheduling, and can be classified into offline task scheduling algorithms [1, 2, 3] or online task scheduling algorithms [4, 5]. Offline algorithms scale the voltage based on the difference between the worst case execution time and the deadline. Online algorithms utilize the variation of run-time execution times and scale the voltage even lower. Work in DPM has centered around prediction of future idle periods [6, 7], aggregation of small idle times to get longer idle durations [3, 8], stochastic control techniques based on Markov chain models [9, 10], etc. Techniques that combine the concepts of DVS and DPM policies to further reduce power usage have been proposed in [11, 12, 13]. The Markov chain-based stochastic approach is used to determine the voltage levels of the DVS processor in [11]. The work in [12] treats the different voltage levels of the DVS processor as multiple active power modes and then applies DPM based on a stochastic approach. The algorithm proposed in [13] looks at the trade offs between the DVS-enabled CPU and the DPM-enabled devices to choose the scaling level of the CPU. None of these approaches take into account the characteristics of the power source.

In this paper we propose a DVS-DPM policy that maximizes the operational lifetime of an embedded system powered by a fuel cell based hybrid power source. The embedded system consists of a processor (CPU) which supports DVS and peripheral devices such as memory, flash disk, ASIC, etc, some of which support DPM. We consider fuel cells (FC) as the power source since they have very high energy density and can meet the growing energy demand of embedded systems. Unfortunately, an FC has limited peak power capacity and can hardly track the frequent and large fluctuations in the load demand. So we use an FC-battery (FC-B) hybrid source which has the high energy density of FC and the high power density of battery [14]. FCs have different characteristics compared to traditional power sources (e.g., batteries) that have to be explicitly taken into account in the development of the energy management policies.

In our prior work, we have proposed FC-aware algorithms when the FC works at a fixed output level [15] as well as when the FC works at multiple output levels [16]. While in both cases, the voltage scaling level of the CPU is a function of the power model of the embedded system and the power state of the hybrid source, the work in [16] allows fuel flow control and results in additional fuel savings. Both [15, 16] considered the FC system efficiency to be a constant, which is true for our first hybrid power source prototype built with PWM DC-DC converter and constant fan speed control policy. In [17], we proposed a more efficient FC configuration with a PWM-PFM DC-DC converter and variable fan speed control policy that had higher efficiency. We proposed an efficient FC output control policy and applied this policy on top of a traditional DPM algorithm traditional DPM algorithm to derive a FC-aware DVS-

*This work was partly supported by the NSF grant (CSR-EHS 05059540). We sincerely acknowledge the help from Kyungsoo Lee at SNU and Aaron Williams at ASU.

ISLPED'07, August 27–29, 2007, Portland, Oregon, USA.

DPM policy. In this paper, we unify our prior work to handle both DVS and DPM. The main contributions of this paper are as follows:

- Proposed a system-wide energy-efficient DVS-DPM policy, which considers the CPU energy as well as the device energy during both the active period and the idle period.
- Formally proved that the proposed FC control policy is optimal in terms of minimizing the fuel consumption.
- Developed the FC-aware DVS-DPM algorithm, which minimizes both the energy consumption of the embedded system and the fuel consumption of the FC-B power source.

The rest of the paper is organized as follows: Section 2 introduces the characteristics of the system under consideration and the optimization framework. Section 3 illustrates our method through a motivational example. Section 4 presents the proposed algorithm which optimizes both the energy consumption and the fuel consumption. Experimental results based on random task traces are given in Section 5, and the paper is concluded in Section 6.

2. BACKGROUND

2.1 System overview

The system under consideration is shown in Figure 1. It consists of an embedded system that includes a DVS-enabled CPU and multiple DPM-enabled peripheral devices and is powered by an FC-B hybrid source. The FC-B source consists of an FC as the primary source that provides the average power, and a rechargeable battery that supports the fluctuations in the load demand. The charging and discharging of the battery is controlled by the CMS (charge management system). The control system implements a fuel-efficient energy management policy; it applies DVS/DPM policy to the embedded system and a fuel flow control policy to the FC system.

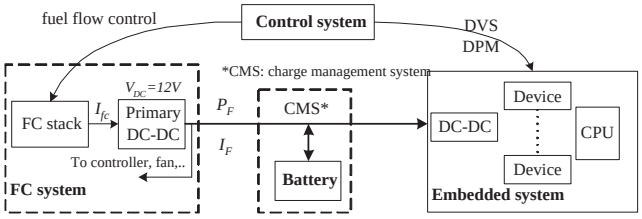

Figure 1: System overview.

2.1.1 Embedded system

DVS enabled CPU: The CPU power consumption consists of the dynamic power, P_{dyn}, and the static power (including the intrinsic power and the leakage power). This is modeled as

$$P_{CPU} = \underbrace{C_{eff} \times V_{dd}^2 \times f}_{P_{dyn}} + \underbrace{P_{on} + V_{dd} \times I_{static}}_{\text{static power}}, \quad (1)$$

where both P_{on} and I_{static} are assumed to be constant, C_{eff} is the effective capacitance determined by the switching activities, V_{dd} is the supply voltage and f is the operating frequency. Since $f \propto \frac{(V_{dd}-Vt)^{\sigma}}{V_{dd}}$, where V_t is the threshold voltage, and σ is a system parameter. We assume $V_{dd} \gg V_t$ and $\sigma = 2$, then $f \propto V_{dd}$.

We define s as the scaling factor which is the ratio of the highest frequency to the operating frequency. Thus $s \geq 1$ and $s \propto V_{dd}^{-1}$. We define κ_1 as the ratio of P_{dyn} to P_{CPU} and κ_2 as the ratio of P_{on} to P_{CPU} at the highest frequency. Then P_{CPU} is represented as

$$P_{CPU}(s) = P_{CPU}(1) \times (\kappa_1 s^{-3} + \kappa_2 + (1 - \kappa_1 - \kappa_2) s^{-1}). \quad (2)$$

If a task can be finished within time τ at the highest frequency, then for scaling factor s, its duration stretches to $s \times \tau$. The corresponding energy consumption is

$$E_{CPU}(s) = P_{CPU}(1) \times \tau \times (\kappa_1 s^{-2} + \kappa_2 s + (1 - \kappa_1 - \kappa_2)). \quad (3)$$

When the CPU is not executing any task, it is considered to be idle. In this mode, the CPU power consumption is a constant value, and is ignored in this paper.

DPM enabled devices: While some modern CPUs support dynamic power management (DPM), in this paper, we only consider DPM control of the peripheral devices. Without loss of generality, we assume that a device has three power modes: RUN, STANDBY and SLEEP. During the *active period* (the execution of a task), the enabled device is in the RUN mode and the corresponding power consumption is P_{run}. When the task finishes, the *idle period* is initiated and the device goes into the STANDBY mode; the corresponding power consumption is P_{sdb}. When the idle period is long enough, the device can be put into the SLEEP mode and the corresponding power consumption is P_{slp}. We assume that P_{run}, P_{sdb} and P_{slp} are constant values. It is obvious that $P_{run} > P_{sdb} > P_{slp}$.

We assume that there is no direct transition between the SLEEP mode and the RUN mode, i.e., the transition must go through the STANDBY mode. The transition overhead between the STANDBY mode and the RUN mode is assumed to be zero, and we only consider the transition overhead between the SLEEP mode and the STANDBY mode. The transition delay to the SLEEP mode is τ_{pd} (pd: power down) and the corresponding power consumption is P_{pd}. The transition delay from the SLEEP mode is τ_{wu} (wu: wake up) and the corresponding power consumption is P_{wu}. We put the device into the SLEEP mode when the idle period is longer than the break-even time, T_{be}. T_{be} is determined by P_{sdb}, P_{slp} and the transition overhead [9].

The device energy consumption during the active period is defined as E_{act}, and the energy consumption during the idle period is defined as E_{idle} which also includes the transition overhead. The total energy consumed by the device is then $E_{DVC} = E_{act} + E_{idle}$.

2.1.2 FC-B hybrid power source

FC system: As shown in Figure 1, the FC system consists of the FC stack, the primary DC-DC converter, and other components (not shown) such as the cooling fan, the air supply fan, the fuel pump, etc. In the FC stack, H_2 and O_2 react to produce electricity. The primary DC-DC converter is in charge of power regulation, and has a constant output voltage of 12 V.

The FC output power P_F can be set to a value in the *load following range*, $\Omega = [P_F^{min}, P_F^{max}]$, by varying the fuel flow rate. The corresponding FC output current is calculated by

$$I_F = \frac{P_F}{V_{DC}}. \quad (4)$$

The FC system efficiency, η_s, is determined by the FC stack efficiency, the primary DC-DC converter efficiency, and the power consumed by other components. Figure 2 plots η_s as a function of the FC output current, I_F. When the air supply fan and the cooling fan operate at the optimal speed determined by the output current, the FC system efficiency is shown in Figure 2(b). In the load following range, η_s can be represented by $\eta_s = \alpha - \beta \times I_F$ where $\alpha > 0$ and $\beta > 0$. This is the case that is considered in this paper. A simpler mechanism is to operate the fans at constant speed (regardless of the FC output) and has been considered in [15, 16]. In such a system, η_s appears to be almost constant within the load following range, as shown in Figure 2(c), and can be said to correspond to the case when $\beta = 0$.

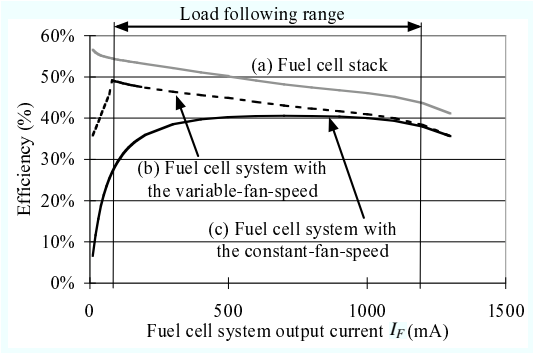

Figure 2: Measured FC stack efficiency and FC system efficiency of the BCS 20 W, 20 stack, room-temperature hydrogen fuel cell (@2 psig H_2 pressure).

The fuel flow rate is proportional to the FC stack current, I_{fc}. In our system, the relationship between I_{fc} and I_F is given by

$$I_{fc} = \frac{\gamma \times I_F}{\eta_s} = \frac{\gamma \times I_F}{\alpha - \beta \times I_F}, \quad (5)$$

where γ is a constant determined by the FC system.

Battery: The battery in Figure 1 acts as an energy buffer. The battery status is denoted as B which is the remaining energy in the battery, and the capacity of the battery is denoted as B^{max}. When P_F is higher than the load power, the charge management system (CMS) switches the battery to the charging mode, and the corresponding energy stored into the battery is denoted as E_{chg}. When P_F is lower than the load power, the battery works in the discharging mode to support the embedded system, and the energy pulled from the battery is denoted as E_{dis}. In this work, we assume that the CMS and the battery perform zero-loss charging/discharging operations.

2.2 Optimization framework

Our goal is to maximize the operational lifetime of the FC with a fixed amount of fuel, which is the same as minimizing the fuel consumption due to execution of a set of tasks. The fuel consumption is proportional to the integration of I_{fc} over time, and thus the cost function that we need to minimize is defined as

$$C = \int I_{fc} \cdot dt. \quad (6)$$

The following constraints have to be satisfied. First the FC output power has to be within the load following range, i.e.,

$$\textit{FC-constraint}: \quad P_F^{min} \le P_F \le P_F^{max}. \quad (7)$$

At any point in time, we cannot store energy into the battery after it has been fully charged, and we cannot pull energy from the battery if it has been exhausted. We refer to these as the hard constraints of the battery, i.e.,

$$\textit{Hard B-constraint}: \quad 0 \le B + E_{chg} - E_{dis} \le B^{max}. \quad (8)$$

To maintain a robust and stable operation, we add an additional battery constraint as shown in Equation (9). This constraint is a soft constraint, and could be violated when it conflicts with the *FC-constraint* or the *Hard B-constraint*.

$$\textit{Soft B-constraint}: \quad E_{chg} = E_{dis}. \quad (9)$$

Figure 3 illustrates the energy flow of the FC-B system. The fuel consumption in terms of C transforms to the FC output energy,

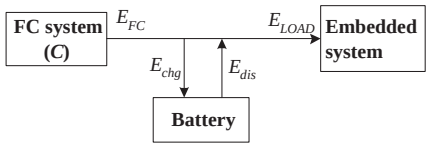

Figure 3: Energy flow of the system under consideration.

$E_{FC} = \int P_F \cdot dt$. The embedded system consumes energy E_{LOAD}. If all the above constraints are satisfied, we have $E_{FC} = E_{LOAD}$.

We summarize the definitions used in this paper in Table 1.

Table 1: Definitions

s	the scaling factor of the DVS-enable CPU, $s \ge 1$.
$P_{CPU}(s)$	CPU power when it is scaled by s.
κ_1, κ_2	the coefficients in CPU power model.
$P_{run}, P_{sdb}, P_{slp}$	device power in RUN/STANDBY/SLEEP mode.
P_{pd}, P_{wu}	device power when entering/exiting SLEEP mode.
τ_{pd}, τ_{wu}	delay when entering/exiting the SLEEP mode.
T_{be}	break-even time of the DPM component.
d	deadline of the task.
τ	execution time of the task when $s = 1$.
$E_{CPU}(s)$	CPU energy consumption.
$E_{act}(s), E_{idle}(s)$	device energy consumption during active/idle period.
$E_{DVC}(s)$	device energy consumption, $E_{DVC} = E_{act} + E_{idle}$.
$E_{LOAD}(s)$	load energy consumption, $E_{LOAD} = E_{CPU} + E_{DVC}$.
B	battery status, the remaining energy in the battery
B^{max}	battery capacity.
E_{chg}, E_{dis}	energy charged to/discharged from the battery.
P_F, I_F	FC system output power/current.
Ω=$[P_F^{min}, P_F^{max}]$	load following range of the FC system.
$V_{DC} = 12$ V	output voltage of the primary DC-DC converter.
η_s	FC system efficiency.
α, β, γ	the coefficients of FC system.
I_{fc}	FC stack current.
C	cost function, the integration of I_{fc} over time.

3. MOTIVATIONAL EXAMPLE

We illustrate the proposed FC-aware DPM-DVS method with a motivational example. To simplify the analysis, we consider a single task that requires one peripheral device coupled with the CPU.

The CPU power model is specified by $P_{CPU}(1) = 10$ W, $\kappa_1 = 0.8$ and $\kappa_2 = 0.1$. The CPU supports scaling factors from 1 to 2 with steps of 0.1. The power modes of the peripheral device are specified by $P_{run} = 8$ W, $P_{sdb} = 4$ W, $P_{slp} = 1.6$ W, $\tau_{pd} = \tau_{wu} = 0.05$ sec and $P_{pd} = P_{wu} = 6.4$ W. The break-even time is $T_{be} = 0.2$ sec. The task is specified by its deadline, $d = 2$ sec, and its execution time, $\tau =$ 1 sec. The FC system output can be varied between [4 15] W; its coefficients are $\alpha = 0.46$, $\beta = 0.13$ and $\gamma = 0.32$. The battery is pre-charged to half of its capacity 100mA-hr. The FC output only changes at the boundaries of the task.

As shown in Figure 1, the system-level management policy consists of a policy applied to the embedded system and a policy applied to the FC power source. In this example, we consider four different policies for the embedded system and two different policies for the FC. The embedded system policies are: (a) Policy *baseline* which does not apply any DVS or DPM technique, (b) Policy *DVS* which minimizes E_{CPU} by applying DVS on the CPU, (c) Policy *DPM* which minimizes E_{DVC} by apply DPM on the device, and (d) the proposed policy *DVS_DPM* which minimizes E_{LOAD} by applying both DVS and DPM on the embedded system. The FC policies are: (1) Policy *ASAP* which sets the FC output power as close to the load power profile as possible, and (2) proposed policy *OFC* which sets the FC output power to an average value in the task duration and lets the battery switch between discharging and charging

modes. The corresponding load profiles and the FC output levels for the eight combinations are shown in Figure 4.

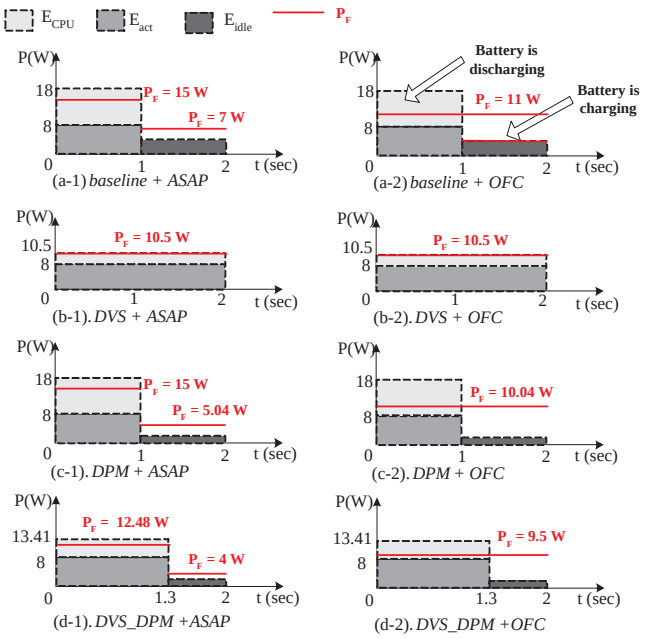

Figure 4: Motivational example - profiles of different policies.

First we check the energy metrics of the four different polices on the embedded system side. Table 2 lists the scaling factors and the values of E_{CPU}, E_{DVC} and E_{LOAD} under different policies. We see that *DVS* has the lowest value of E_{CPU} but the highest E_{DVC}, *DPM* has the lowest value of E_{DVC} but the highest E_{CPU}. *DVS_DPM* considers both DVS and DPM, and is successful at achieving the lowest total energy consumption, E_{TOTAL}.

Table 2: Motivational example - load energy metrics

	(a) *baseline*	(b) *DVS*	(c) *DPM*	(d) *DVS_DPM*
s	1	2	1	1.3
E_{CPU}	10 J	**5 J**	10 J	7.03 J
E_{DVC}	12 J	16 J	**10.08 J**	12 J
E_{LOAD}	22 J	21 J	20.08 J	**19.03 J**

To illustrate the difference between the two FC control policies, consider the profiles shown in Figure 4(a-1) and Figure 4(a-2). The load power during the active period ([0 sec, 1 sec]) is 18 W. Policy *ASAP* sets P_F to 15 W according to the *FC-constraint* (given in Equation (7)) and this causes the battery to discharge. To satisfy the *Soft B-Constraint* (given in Equation (9)), during the idle period ([1 sec, 2 sec]) P_F is adjusted from 4 W to 7 W. If instead we use Policy *OFC*, then for the entire duration (active period and idle period), FC output power is set to 11 W as shown in Figure 4(a-2).

After determining P_F, we calculate the fuel consumption in terms of $\mathcal{C}$ by using Equations (4) (5) and (6). The values of $\mathcal{C}$ for the eight policy-combinations are shown in Table 3. As we can see, the lowest fuel consumption is achieved by the combination of Policy *DVS_DPM* and Policy *OFC*.

Table 3: Motivational example - fuel consumption $\mathcal{C}$ (A-s)

	(a) *baseline*	(b) *DVS*	(c) *DPM*	(d) *DVS_DPM*
(1) *ASAP*	1.83	1.61	1.68	1.51
(2) *OFC*	1.72	1.62	1.52	**1.42**

4. ALGORITHM OFC_DVS_DPM

The proposed algorithm *OFC_DVS_DPM* consists of two main steps: *load energy minimization* by applying DVS and DPM together, followed by *optimal FC output control*. When we cannot control the FC output, as in our previous work [15], the minimal load energy consumption does not correspond to the minimal fuel consumption. However, if the FC has the capability to change its output power by controlling the fuel flow rate, the load energy minimization is an essential step for fuel minimization [16, 17]. This will be formally proved in Section 4.1.2.

We first describe our algorithm for the single task case, and then expand it to the multiple-task case.

4.1 Single task

4.1.1 Load energy minimization

We first describe how to minimize E_{LOAD} where there is only one peripheral device involved. Let the task have a deadline d and execution time τ. We define a break-even scaling factor, s_{be}, corresponding to T_{be}, as

$$s_{be} = \frac{d - T_{be}}{\tau}. \tag{10}$$

The system-level energy consumption of the embedded system is

$$E_{LOAD}(s) = E_{CPU}(s) + E_{DVC}(s) = E_{CPU}(s) + E_{act}(s) + E_{idle}(s), \tag{11}$$

where E_{CPU} is calculated by Equation (3), and the device energy consumptions are calculated by

$$E_{act}(s) = P_{run} \times s \times \tau, \tag{12}$$

$$E_{idle}(s) = \begin{cases} P_{slp} \times (d - s\tau - \tau_{pd} - \tau_{wu}) + P_{pd}\tau_{pd} + P_{wu}\tau_{wu} & \text{if } s \le s_{be}, \\ P_{sdb} \times (d - s\tau) & \text{otherwise.} \end{cases} \tag{13}$$

Since E_{LOAD} in Equation (11) is convex in the range of $s \le s_{be}$ as well as in the range of $s > s_{be}$, we can find the optimal scaling factor by comparing the two minimum values in these two ranges. For instance, when $s \le s_{be}$, the energy function can be expanded to

$$E_{LOAD}(s) = P_{slp} \times (d - \tau_{pd} - \tau_{wu}) + P_{pd}\tau_{pd} + P_{wu}\tau_{wu} + \left((\kappa_1 s^{-3} + \kappa_2 + (1 - \kappa_1 - \kappa_2)s^{-1}) \times P_{CPU}(1) + P_{run} - P_{slp}\right) \times s\tau,$$

and its minimal value occurs at $s = \sqrt[3]{\frac{2\kappa_1 \times P_{CPU}(1)}{\kappa_2 \times P_{CPU}(1) + P_{run} - P_{slp}}}$. Note that the scaling factor has to be bounded by s_{be} by definition. Similarly, the minimal value of $E_{LOAD}(s)$ when $s > s_{be}$ occurs at $s = \sqrt[3]{\frac{2\kappa_1 \times P_{CPU}(1)}{\kappa_2 \times P_{CPU}(1) + P_{run} - P_{sdb}}}$.

The power model of a real system might be more complicated and cannot be expressed by simple convex functions. In that case, the optimal scaling factor has to be determined by numerical search.

Next we consider the case when the task execution requires n devices, and each device has a different break-even time $T_{be,k}$. We sort the devices in decreasing order of their break-even times, so that $s_{be,1} \le s_{be,2} \le \cdots \le s_{be,n}$. Next we determine the value of s which minimizes $E_{LOAD}(s)$ by searching in each of the interval $(s_{be,k-1}, s_{be,k}]$. In a real system, the CPU only supports a few discrete scaling levels, so the scaling factor corresponding to the minimum energy consumption can be easily calculated by numerical search.

Note that our approach is similar to [13], but is more general in that it considers more accurate power models for both the CPU and the peripheral devices, multiple devices and devices with multiple power states.

4.1.2 Optimal FC output control

After determining the load power profile, we set the FC output power such that the fuel consumption is minimized. In our previous work [17], we have shown that setting an average FC output power during the whole *task slot* (active period + idle period) corresponds to the lowest fuel consumption. This is formally proved below.

THEOREM 1. *When a fixed amount of energy is to be drawn from the FC in a fixed amount of time, less fuel is consumed by operating the FC at a single output level than operating it at multiple output levels.*

PROOF. We only need to prove that operating the FC at one single power output level consumes less fuel than operating it at two power output levels.

Assume that we want to draw energy $\mathcal{E}$ from the FC in time $\mathcal{T}$. We consider two policies.

- Setting I: We first operate the FC at output level $\tilde{I}_{F,1}$ in a duration with length $\lambda\mathcal{T}$, where $0<\lambda<1$. Then we operate it at output level $\tilde{I}_{F,2}$ in the remaining duration of length $(1-\lambda)\mathcal{T}$. The two output levels have to satisfy

$$V_{DC}\times(\tilde{I}_{F,1}\times\lambda\mathcal{T}+\tilde{I}_{F,2}\times(1-\lambda)\mathcal{T})=\mathcal{E}. \quad (14)$$

  According to Equations (5) and (6), the corresponding cost function is

$$\begin{aligned} C_I &= \frac{\gamma\times\tilde{I}_{F,1}}{\alpha-\beta\times\tilde{I}_{F,1}}\times\lambda\mathcal{T}+\frac{\gamma\times\tilde{I}_{F,2}}{\alpha-\beta\times\tilde{I}_{F,2}}\times(1-\lambda)\mathcal{T} \\ &= f(\tilde{I}_{F,1})\times\lambda+f(\tilde{I}_{F,2})\times(1-\lambda), \end{aligned} \quad (15)$$

  where function $f(x)$ is in the form of $f(x)=\frac{\gamma\times x\times\mathcal{T}}{\alpha-\beta\times x}$.

- Setting II: We operate the FC at the output level $\tilde{I}_F$ in the whole duration. Similarly we have

$$V_{DC}\times\tilde{I}_F\times\mathcal{T}=\mathcal{E}, \quad (16)$$

  and the cost function is

$$C_{II}=\frac{\gamma\times\tilde{I}_F}{\alpha-\beta\times\tilde{I}_F}\times\mathcal{T}=f(\tilde{I}_F). \quad (17)$$

Function $f(x)$ is convex in the load following range, because $f''(x)>0$. From Equations (14) and (16), we know that

$$\tilde{I}_{F,1}\times\lambda+\tilde{I}_{F,2}\times(1-\lambda)=\tilde{I}_F. \quad (18)$$

Based on the definition of the convex function,

$$C_I>C_{II}. \quad (19)$$

Thus Setting II with a single output level consumes less fuel. Note that if the FC system has constant efficiency as shown in Figure 2(c), i.e., $\beta=0$, then $C_I=C_{II}$ and the single level output level is still a good choice. □

LEMMA 1. *If the FC operates at a single output level, lower FC output energy corresponds to lower fuel consumption.*

PROOF. In duration $\mathcal{T}$, assume that the single FC output $I_F=\hat{I}_{low}$ delivers energy $\mathcal{E}_{low}$, and that the single FC output $I_F=\hat{I}_{high}$ delivers energy $\mathcal{E}_{high}$. If $\mathcal{E}_{high}>\mathcal{E}_{low}$, $\hat{I}_{high}>\hat{I}_{low}$ because I_F is a linear function of $\mathcal{E}$. $C_{high}>C_{low}$ since the cost function C is monotonically increasing as I_F increases. That is,

$$\mathcal{E}_{high}>\mathcal{E}_{low}\Rightarrow\hat{I}_{high}>\hat{I}_{low}\Rightarrow C_{high}>C_{low}. \quad (20)$$

□

Note that Lemma 1 also shows that lower E_{LOAD} results in lower fuel consumption.

From Theorem 1 and Lemma 1, we see that for a certain E_{LOAD}, the minimum fuel consumption is achieved by setting the FC output level to a single value given by

$$P_F=\frac{E_{LOAD}}{d}. \quad (21)$$

If the desired value of P_F is beyond the FC load following range, i.e., the *FC-constraint* in Equation (7) is violated, we set P_F to P_F^{max} (if $\frac{E_{LOAD}}{d}>P_F^{max}$) or P_F^{min} (if $\frac{E_{LOAD}}{d}<P_F^{min}$). If the desired value of P_F cannot satisfy the *Hard B-constraint*, we set P_F to the closest boundary value determined by Equation (8). In these cases, we might have to sacrifice the *Soft B-constraint* for the current task slot, and the difference between E_{dis} and E_{chg} is compensated for in the following task slot (slots) by appropriate choice of P_F.

4.2 Algorithm OFC_DVS_DPM

Figure 5 describes the flow chart of Algorithm *OFC_DVS_DPM*. For each task, we first do load energy minimization, and then find the optimal FC operating level. This is performed sequentially for each task in the task queue.

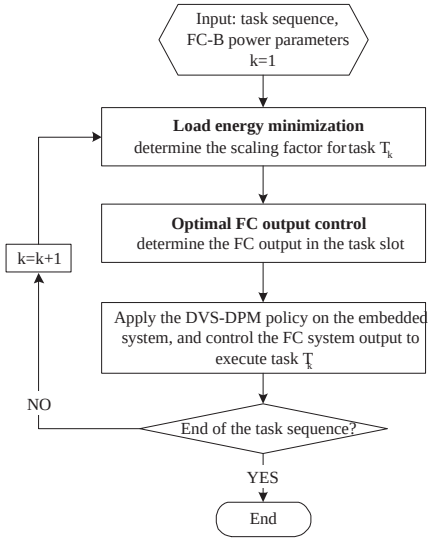

Figure 5: Flow chart of Algorithm *OFC_DVS_DPM*.

5. EXPERIMENTAL RESULT

In this section, we compare the performance of the proposed algorithm, *OFC_DVS_DPM*, with the competing policies for a synthetic random task trace.

5.1 Experimental settings

The FC-B power source consists of an FC with load following range [4 W, 15 W] and coefficients $\alpha=0.46$, $\beta=0.13$ and $\gamma=0.32$, and a battery which has been pre-charged to its capacity, $B^{max}=$ 100 mA-hr.

The embedded system consists of one CPU and two devices, D_1 and D_2. The power specifications of the CPU (at highest frequency) are as follows: the intrinsic power is 2 W, the static power is 1 W, and the dynamic power is determined by the task since different task has different switch activities. The scaling factors supported by the CPU are in the range of 1 to 2 with steps of 0.1. The device power specifications are given in Table 4.

Table 4: Device power specification

	P_{run}	P_{sdb}	P_{slp}	P_{pd}	τ_{pd}	P_{wu}	τ_{wu}	T_{be}
D_1	8 W	4 W	0.5 W	4 W	0.05 s	4 W	0.05 s	0.1 s
D_2	5 W	2 W	0.5 W	1 W	0.05 s	4 W	0.1 s	0.25 s

There are four groups of tasks, T_1, T_2, T_3 and T_4, executed sequentially. For each task group T_k, there are multiple task instances of the same task type (i.e., the same CPU power model and the same device set). The number of task instances is randomly generated between 50 to 100. The deadline of a task instance is chosen from in a uniform distribution between 5 sec and 10 sec; and the task execution time is the product of the deadline and processor utilization which again is randomly chosen from the range 30 % to 100 %. The CPU dynamic power and the required device set of the tasks in each task group are specified in Table 5.

Table 5: Task specification

Task	T_1	T_2	T_3	T_4
$P_{dyn}(1)$	8W	10 W	8W	6W
Device Set	D_1	-	D_1, D_2	D_2

5.2 Experimental results

We apply four embedded system policies (*baseline*, *DVS*, *DPM*, *DVS_DPM*) in combination with two FC control policies (*ASAP*, *OFC*) and compare the performance.

Table 6 shows the E_{LOAD} metrics of the four policies for the embedded system. As we can see, by jointly applying DVS on the CPU and DPM on the devices, we can achieve around 14 % savings in E_{LOAD}. The energy savings compared to DVS only policy or DPM only policy are also up to 5 % and 8 %, respectively. Note that the proposed policy, *DVS_DPM*, could be applied on the embedded systems powered by other sources as well (other than FC-B mentioned in this paper).

Table 6: Experimental result - load energy consumption

Policy	*baseline*	*DVS*	*DPM*	*DVS_DPM*
E_{LOAD} (KJ)	33.70	30.60	31.49	**29.19**

Table 7 shows the fuel consumption when we combine the above four policies with different FC control policies, *ASAP* and the proposed policy *OFC*. The results show that:

- When a certain energy policy is used on the embedded system, applying Policy *OFC* always consumes lower fuel consumption compared to application of Policy *ASAP*.
- When the optimal FC control policy is used, the fuel consumption due to different embedded system policies follows the same trend as that of E_{LOAD}.

The minimum fuel consumption is achieved by the combination of *DVS_DPM* and the optimal FC control policy *OFC*. The fuel savings of Algorithm *FC_DVS_DPM* is as high as 25% over the worst combination (*baseline* + *ASAP*).

The operational lifetime of the FC-B system is the inverse of the fuel consumption. For this example, when the optimal FC control is employed, the lifetime achieved by Algorithm *OFC_DVS_DPM* is 18% longer than the baseline policy, 6.2 % longer than the DVS only policy, and 7.9 % longer than the DPM only policy.

6. CONCLUSION

In this paper we consider an embedded system powered by an FC-B hybrid source, where the embedded system consists of a DVS-enabled CPU and DPM-enabled devices. Our goal is to maximize the lifetime of the FC, i.e., minimize the fuel consumption. This is achieved by applying a combination of an embedded system load energy minimization policy and an optimal FC output control policy. The load energy minimization finds the CPU scaling factor which minimizes E_{LOAD}, defined as the sum of the CPU energy and the device energy. The optimal FC control policy explicitly takes into account the characteristics of the FC system efficiency. It achieves the lowest fuel consumption for a given E_{LOAD}. Simulations on random task traces demonstrates the superiority of the proposed algorithm compared to other competing algorithms.

Table 7: Experimental result - fuel consumption, C (A-s)

Policy	*baseline*	*DVS*	*DPM*	*DVS_DPM*
ASAP	3332.3	2723.9	3186.0	2746.1
OFC	2960.8	2659.8	2700.8	**2504.3**

7. REFERENCES

[1] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced cpu energy," *IEEE Annual Foundations of Computer Science*, pp. 374–382, 1995.

[2] G. Quan, and X. Hu, "Energy efficient fixed-priority scheduling for real-time systems on variable voltage processors," in *Proc. of DAC*, 2001, pp. 828–833.

[3] R. Jejurikar and R. Gupta, "Leakage aware dynamic voltage scaling for real-time embedded systems," in *Proc. of DAC*, June 2004, pp. 275–280.

[4] Y. Shin, K. Choi, and T. Sakurai, "Power optimization of real-time embedded systems on variable speed processors," in *Proc. of ICCAD*, 2000, pp. 365–368.

[5] W. Kim, J. Kim and S. Min, "A dynamic voltage scaling algorithm for dynamic-priority hard real-time systems using slack time analysis," in *Proc. of DATE*, 2002, pp. 788–794.

[6] C.-H. Hwang and A. Wu, "A predictive system shutdown method for energy saving of event-driven computation," in *Proc. of ICCAD*, Nov. 1997, pp. 28–32.

[7] EY Chung, L. Benini, and G. De Micheli, "Dynamic power management using adaptive learning tree," in *Proc. of ICCAD*, Nov. 1999, pp. 274–279.

[8] Y. Lu, L. Benini, and G. De Micheli, "Low-power task scheduling for multiple devices," in *Proc. of CODES*, 2000, pp. 39–43.

[9] L. Benini, A. Bogliolo, and G. De Micheli, "A survey of design techniques for system-level dynamic power management," *IEEE TVLSI*, vol. 8, pp. 299–316, 2000.

[10] P. Rong and M. Pedram, "Battery-aware power management based on Markovian decision processes," *IEEE TCAD*, vol. 25, pp. 1337–1349, 2006.

[11] M. Kargahi and A. Movaghar, "A stochastic dvs-based dynamic power management for soft real-time systems," in *Proc. of IWCMC*, 2005, pp. 63–68.

[12] T. Simuinic, L. Benini, A. Acquaviva, P. Glynn, and G. D. Micheli, "Dynamic voltage scaling and power management for portable systems," in *Proc. of DAC*, June 2001, pp. 524–529.

[13] M. Kim and S. Ha, "Hybrid run-time power management technique for realtime embedded system with voltage scalable processor," in *Proc. of LCTES*, June 2001, pp. 11–19.

[14] L. Gao, Z. Jiang, and R. A. Dougal, "Evaluation of active hybrid fuel cell/battery power sources," *IEEE Trans. on Aeorspace and Electronic Systems*, vol. 41, no. 1, pp. 346–355, Jan. 2005.

[15] J. Zhuo, C. Chakrabarti, N. Chang, and S. Vrudhula, "Extending the lifetime of fuel cell based hybrid systems," in *Proc. of DAC*, July 2006, pp. 562–567.

[16] J. Zhuo, C. Chakrabarti, N. Chang, and S. Vrudhula, "Maximizing the lifetime of embedded systems powered by fuel cell-battery hybrids," in *Proc. of ISLPED*, October 2006, pp. 424–429.

[17] J. Zhuo, C. Chakrabarti, K. Lee, and N. Chang, "Dynamic power management with hybrid power sources," in *Proc. of DAC*, June 2007, pp. 871–876.