DC–DC Converter-Aware Power Management for Low-Power Embedded Systems

Yongseok Choi, *Student Member, IEEE*, Naehyuck Chang, *Senior Member, IEEE*, and Taewhan Kim, *Member, IEEE*

***Abstract*—Most digital systems are equipped with dc–dc converters to supply various levels of voltages from batteries to logic devices. DC–DC converters maintain legal voltage ranges regardless of the load current variation as well as battery voltage drop. Although the efficiency of dc–dc converters is changed by the output voltage level and the load current, most existing power management techniques simply ignore the efficiency variation of dc–dc converters. However, without a careful consideration of the efficiency variation of dc–dc converters, finding a true optimal power management will be impossible. In this paper, we solve the problem of energy minimization with the consideration of the characteristics of power consumption of dc–dc converters. Specifically, the contributions of our work are as follows: 1) We analyze the effects of the efficiency variation of dc–dc converters on a single-task execution in dynamic voltage scaling (DVS) scheme and propose the DC_DVS technique for dc–dc converter-aware energy-minimal DVS. 2) DC_DVS is then extended to embed an awareness of the characteristics of dc–dc converters in general DVS techniques for multiple tasks. 3) We go on to propose a technique called DC_CONF for generating a dc–dc converter that is most energy efficient for a particular application. 4) We also present an integrated framework, i.e., DC-lp, based on DC_DVS and DC_CONF, which addresses dc–dc converter configuration and DVS simultaneously. Experimental results show that DC-lp is able to save up to 24.8% of energy compared with previous power management schemes, which do not consider the efficiency variation of dc–dc converters.**

***Index Terms*—DC–DC converter, low power, voltage scaling.**

I. Introduction

ALMOST all modern digital systems are supplied with power through dc–dc converters because high-performance CMOS devices are optimized to specific supply voltage ranges. DC–DC converters are generally classified into two types, namely: 1) linear voltage regulators and 2) switching voltage regulators, according to the circuit implementation. However, nontrivial power dissipation is unavoidable in both types of voltage conversion and directly affects battery life. Fig. 1 shows the path of current flow through a dc–dc converter from a battery. It is reported that there is always a nontrivial power loss in the converter, the amount of which is 10% to 40% of the total energy consumed in the system.

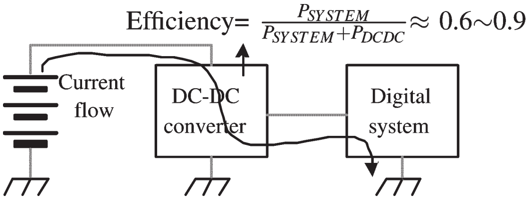

Efficiency= $\frac{P_{SYSTEM}}{P_{SYSTEM}+P_{DCDC}} \approx 0.6\sim0.9$
Current flow
DC-DC converter
Digital system

Fig. 1. Typical current flow path. Nontrivial power loss in the dc–dc converter results in a short battery life despite low power dissipation of the digital system.

It is generally known that switching regulators achieve better power efficiency than linear regulators, but linear regulators are much cheaper and produce less noise than switching regulators. For this reason, switching regulators are mostly used for low-power or high-current application, except when low noise or low cost is particularly important. There are several works that have addressed the problem of increasing the efficiency of switching dc–dc converters. Some [2], [3] have focused on more efficient circuit configurations, others [4], [5] on circuit modifications, and yet others [2], [6], [7] on investigating the sources of power loss in dc–dc converters and on developing a power dissipation model in terms of input/output characteristics and converter parameters. Simunic *et al.* [8] proposed a methodology for cycle-accurate simulation of performance and energy consumption in an embedded system with a dc–dc converter and pointed out that the energy loss in a dc–dc converter took a significant fraction of the total energy consumption.

On the other side, so far, a lot of power management techniques aimed at saving energy in embedded system design have been proposed. Nevertheless, almost all of them do not seriously take into account the efficiency of dc–dc converters, simply assuming dc–dc converter power efficiency as a constant value [9], [10]. If the efficiency of a dc–dc converter was constant over its entire operating range, we could ignore the dc–dc converter's effect on the total energy consumption of the system. However, in reality, the efficiency of a dc–dc converter has a close correlation with the level of output voltage and load current. Consequently, when a power management scheme such as dynamic voltage scaling (DVS), which involves varying the supply voltage, is implemented in an embedded system, it is also essential to properly schedule the output voltage of the dc–dc converter, so that the overall energy consumption of the system is minimized. Note that in the case of a switching regulator, in addition to the output voltage, its power efficiency is affected by the load current as well. The key concern of our

Manuscript received July 25, 2005; revised August 4, 2006. This work was supported by the Information Technology Research and Development Project funded by the Korean Ministry of Information and Communications. The work of T. Kim was supported in part by the Nano IP/SoC Promotion Group, Seoul Research and Business Development Program, and in part by the Ministry of Science and Technology/Korea Science and Engineering Foundation through the Advanced Information Technology Research Center. This paper was presented in part at the Design Automation Conference 2005, Anaheim, CA, June 2005. This paper was recommended by Associate Editor M. Pedram.

Y. Choi and N. Chang are with the School of Computer Science and Engineering, Seoul National University, Seoul 151-742, Korea.

T. Kim is with the School of Electrical Engineering and Computer Science, Seoul National University, Seoul 151-742, Korea.

Digital Object Identifier 10.1109/TCAD.2007.890837

0278-0070/$25.00 © 2007 IEEE

work is as follows: Although an effective power management scheme can reduce the power consumption of a device to a large extent, it does not always mean that it also reduces the power consumption of a dc–dc converter minimally, in some cases operating very inefficiently, resulting in a poor battery life enhancement. Consequently, it is quite necessary to solve the two problems, namely: 1) the problem of (output) voltage scaling of a dc–dc converter and 2) the problem of voltage scaling that is applied to the devices other than the dc–dc converter in an integrated fashion, so that the total energy consumption is globally minimized.

DVS is accepted as one of the most effective and well-studied power management techniques. Assuming that the processor supply voltage is dynamically and continuously variable, there are optimal algorithms for scheduling nonperiodic tasks and selecting the best voltage for each task [11], and there are also voltage scaling techniques with fixed priority scheduling, which are applicable to periodic tasks [12]. Essentially, most studies suggested DVS algorithms based on dynamic or static priorities. These algorithms are differentiated by how slack times are estimated and redistributed [13]–[17]. Some DVS schemes adjust the supply voltage within an individual task boundary (i.e., intratask), not on task-by-task basis [18]–[20]. In [21], practical DVS schemes with the consideration of discrete supply voltage and nonuniform load capacitances were suggested. However, while all these DVS schemes do save energy, none of them take into account the effects of voltage scaling on the dc–dc converter, the resulting changes in load current, and their effect on the efficiency of the dc–dc converter itself. To overcome this limitation in previous power management techniques, we will now address the issue of dc–dc converter-aware power management. Specifically, we approach the problem in two aspects to cover the core parts of the problem of dc–dc converter-aware power management.

1) *The converter-aware voltage scaling problem.* For a single task with execution cycles and a deadline, we derive the power consumption model of a dc–dc converter by analyzing how power consumption is related to output voltage and propose a robust voltage scaling technique[1] that minimizes the sum of the energy consumed by the execution of the task and the energy dissipated by the dc–dc converter. The proposed technique is then extended to handle multiple tasks. We then address the other problem.
2) *The application-driven converter optimization problem.* This involves finding the dc–dc converter configuration that is best suited to the application and system, in terms of minimizing the total energy consumption.

Section II starts with a brief summary of the function of dc–dc converters, followed by the development of a model of their power consumption and the derivation of power equations. In Section III, we present an integrated dc–dc converter-aware energy minimization algorithm, which essentially solves the two core problems, namely: 1) converter-aware voltage scaling and 2) application-driven converter optimization. Section IV presents a set of experimental results that show the effectiveness of the proposed techniques. Finally, concluding remarks are made in Section V.

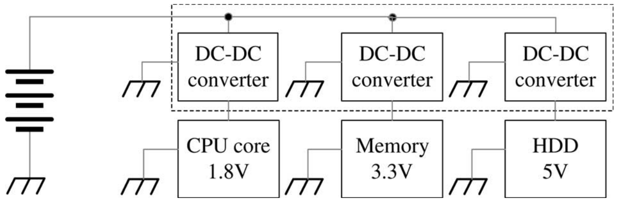


Fig. 2. DC–DC converters generate different supply voltages for the CPU, memory, and hard disk drive from a single battery.

II. DC–DC CONVERTERS

A. Voltage Regulation

The proliferation of digital devices and constant technological innovation make it impossible to use a single supply voltage for all devices, and often, multiple levels are required in a single device. Since all supply voltages are generally derived from a single battery, voltage regulators (dc–dc converters) are needed to control the supply voltage for each device, as indicated in Fig. 2, which shows a simplified power supply network for a typical battery-operated embedded system.

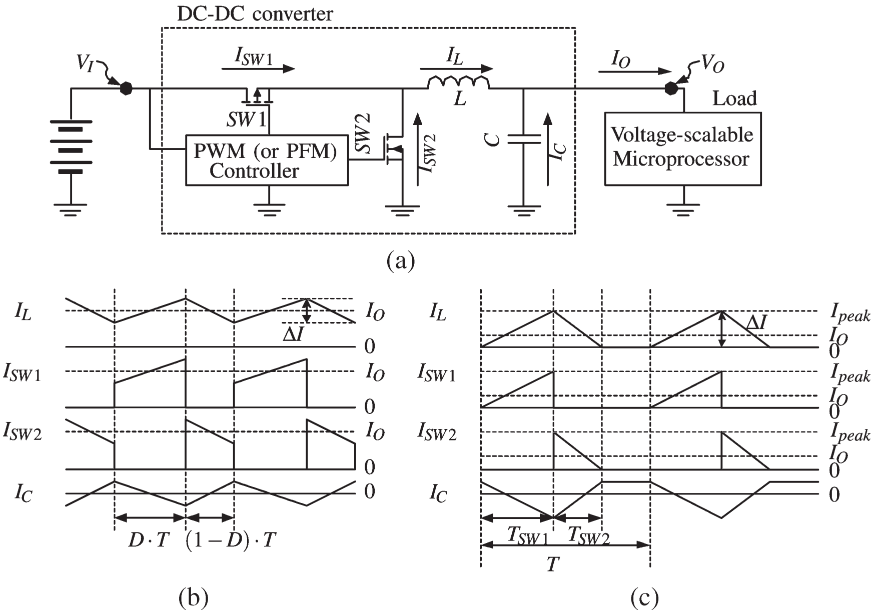
The primary role of a dc–dc converter is to provide a regulated power source. Unlike passive components, logic devices do not draw a constant current from their power supply. The power supply current changes rapidly with changes in the devices' internal states. An unregulated power supply is likely to suffer an IR drop corresponding to the load current, whereas a regulated power supply aims to keep the output voltage constant regardless of variation in the load current. The phenomenon of IR drop is caused by internal resistance of the power supply. Therefore, a dc–dc converter is still needed for voltage regulation even if there is only one supply voltage.

B. Switching Regulator Basics

We focus on minimizing the power dissipation of a step-down switching regulator, which is the type most frequently used in low-power applications. A switching regulator uses an inductor, a transformer, or a capacitor as an energy-storage element to transfer energy from the power source to the system. The amount of power dissipated by voltage conversion in a switching regulator is relatively low, mainly due to the use of low-resistance MOSFET switches and energy-storage elements. However, the amount of power dissipated by a linear regulator is rather high, mainly because there is an upper bound on the efficiency of a linear regulator, which is equal to the output voltage divided by the input voltage. In addition, switching regulators can increase (i.e., boost), decrease (i.e., buck), and invert input voltage with a simple modification to the converter topology, unlike linear regulators. Fig. 3(a) shows the basic structure of the step-down (buck) switching regulator.

A switching regulator contains a circuit, located on the path between the external power supply and the energy-storage

[1] Note that our proposed voltage scaling technique is flexible enough to be incorporated into most of the existing DVS methods with minimal modification. Section III-D covers such a general applicability.



(a)

(b)

(c)

Fig. 3. Simplified block diagram of a buck converter and the current flow thorough each component. (a) Structure of a buck dc–dc converter. (b) Current flow in a PWM dc–dc converter. (c) Current flow in a PFM dc–dc converter.

element, which controls two MOSFET switches. The switch control techniques most widely used in practical dc–dc converters are pulsewidth modulation (PWM), which controls the turn-on duty ratio of each MOSFET with a fixed switching frequency, and pulse-frequency modulation (PFM), which controls the switching frequency by constraining the peak current flowing through the inductor. Each control technique has its own advantages and shortcomings. DC–DC converters controlled by PWM generate less ripple in the output voltage, their switching noises are easier to filter out, and they are more efficient under heavy loading, whereas dc–dc converters controlled by PFM exhibit higher efficiency with light loads.

Most commercial dc–dc converters use either PWM or hybrid PWM/PFM control. The hybrid technique inherits higher light-load efficiency from the PFM control technique, but PWM is preferred for applications that require low cost, or small size, and for noise-sensitive systems including analog circuits and wireless communication subsystems. A dc–dc converter needs to be selected carefully to meet the constraints of a particular application. In this paper, we will consider both types of dc–dc converter.

C. Power Dissipation of DC–DC Converter

An ideal switching regulator consumes no power, unlike an ideal linear voltage regulator. However, practical dc–dc converters have nonideal characteristics that cause power to be lost. Generally, the major sources of power dissipation in dc–dc conversion are classified into three categories [7], which are to be amplified in the remainder of this section. Based on a number of previous studies of power loss in dc–dc converters [2], [7], [25], we will express the power dissipation due to each source in terms of manufacturing parameters and load conditions such as the output voltage and the output current, which can be controlled by DVS. In this paper, we will express the power dissipation of the dc–dc converter as the sum of the following three components:

$$P_{\text{dcdc}} = P_{\text{conduction}} + P_{\text{gate\_drive}} + P_{\text{controller}}. \tag{1}$$

1) Conduction Power Dissipation: All the elements of a dc–dc converter, such as switches, inductors, and capacitors, are nonideal and have their own resistive components R_{ESR}. This means that power dissipation $I^2 \cdot R_{\text{ESR}}$ due to the current I through these elements is unavoidable.

Although varied amounts of current flow through different components, as shown in Fig. 3, these currents are all positively related to the load current I_O of the system. Consequently, the conduction power dissipation of the dc–dc converter can be reduced by reducing the load current, which can be achieved by high-level power management that controls the load power.

Since the two types of dc–dc converter have different ways of switching their two MOSFETs, as shown in Fig. 3(b) and (c), their conduction power dissipation has different characteristics. Therefore, we use different conduction power models for the different types of dc–dc converter.

The power consumption of a PWM dc–dc converter can be formulated as

$$\begin{aligned} P_{\text{conduction(PWM)}} &= I_O^2 \cdot (D \cdot R_{SW1} + (1-D) \cdot R_{SW2} + R_L) \\ &+ \frac{1}{3} \cdot \left(\frac{\Delta I_{L(\text{PWM})}}{2}\right)^2 \cdot (D \cdot R_{SW1} + (1-D) \cdot R_{SW2} + R_L + R_C) \end{aligned} \tag{2}$$

where V_I, V_O, and I_O are the input voltage, output voltage, and output current (i.e., the load current) of the dc–dc converter, respectively; and R_{SW1}, R_{SW2}, R_L, and R_C are the turn-on resistance of the top MOSFET $(SW1)$, the turn-on resistance of the bottom MOSFET $(SW2)$, the equivalent series resistance of the inductor L, and the equivalent series resistance of the capacitor C, respectively. D and $\Delta I_{L(\mathrm{PWM})}$ are the duty ratio (time when the current actually flows through the component/total time) and the ripple of the current flowing through the inductor, respectively, which can be expressed as follows:

$$D = \frac{V_O}{V_I}, \qquad \Delta I_{L(\mathrm{PWM})} = \frac{V_O \cdot (1-D)}{L_f \cdot f_S} \tag{3}$$

where L_f is the value of the inductor, and f_S is the switching frequency, which is assumed to be constant in a PWM dc–dc converter.

$P_{\mathrm{conduction(PWM)}}$ consists of two terms. The first and second terms represent the conduction power consumptions due to the dc component and the ac component (or current ripple), respectively, of the current flowing through all components (i.e., $SW1$, $SW2$, L, and C) on the current path. In the first term of $P_{\mathrm{conduction(PWM)}}$ in (2), $D \cdot R_{SW1} + (1-D) \cdot R_{SW2} + R_L$ is the effective resistance of the current path of the dc–dc converter, considering the duty ratio of each component on that path. The duty ratios for $SW1$, $SW2$, and L are D, $(1-D)$, and 1, respectively. (Since the dc component of the current flowing through the C is zero, the term related to C is omitted.) It is well known that the conduction power consumption of some systems can be expressed by $I^2 \cdot R$, where I is the current flowing through the system and R is the resistive component of the system. Therefore, the product of this effective resistance and I_O^2, where I_O is equivalent to the dc component of the current flowing through each component, can be used to model the dc component of the conduction power consumption of the PWM dc–dc converter. In the second term, $D \cdot R_{SW1} + (1-D) \cdot R_{SW2} + R_L + R_C$ is the effective resistance, and $(1/3) \cdot (\Delta I_{L(\mathrm{PWM})}/2)^2$ is the square of the ac component (or current ripple) of the current flowing through the components.

A PFM dc–dc converter has a variable switching frequency that depends on the output current, the output voltage, and other factors. Therefore, the switching frequency should be characterized accurately to determine the amount of the conduction power dissipation of a PFM dc–dc converter. From [7], the switching frequency can be described as

$$f_{S(\mathrm{PFM})} = \frac{1}{T} = \frac{2 \cdot I_O}{I_{\mathrm{peak}} \cdot (T_{SW1} + T_{SW2})} \tag{4}$$

where I_{peak} is the peak inductor current allowed in a given PFM dc–dc converter, and T_{SW1} and T_{SW2} are the turn-on times of the top MOSFET $(SW1)$ and the bottom MOSFET $(SW2)$, respectively. T_{SW1} and T_{SW2} can be determined as follows:

$$T_{SW1} = \frac{I_{\mathrm{peak}} \cdot L_f}{V_I - V_O}, \qquad T_{SW2} = \frac{I_{\mathrm{peak}} \cdot L_f}{V_O}. \tag{5}$$

$P_{\mathrm{conduction(PFM)}}$ in (6) is modeled in the same way as $P_{\mathrm{conduction(PWM)}}$. In the first term, $((T_{SW1} + T_{SW2})/T) \cdot (I_{\mathrm{peak}}/2)^2$ is the square of the dc component of the current flowing through each component, and in the second term, $(1/3) \cdot ((T_{SW1} + T_{SW2})/T) \cdot (\Delta I_{L(\mathrm{PFM})}/2)^2$ is the square of the ac component of that current. The duty ratios for $SW1$ and $SW2$ are $(T_{SW1}/(T_{SW1} + T_{SW2}))$ and $(T_{SW2}/(T_{SW1} + T_{SW2}))$, respectively. These expressions for the current and duty ratios can be found in (or derived from) many references (e.g., [7] and [25]). Replacing T_{SW1} and T_{SW2} with the expressions from (5), we can also construct the alternative expression shown in the last two lines of the following equation:

$$\begin{aligned}
&P_{\mathrm{conduction(PFM)}} \\
&= \frac{T_{SW1}+T_{SW2}}{T} \\
&\quad \cdot \left(\left(\frac{I_{\mathrm{peak}}}{2}\right)^2 \cdot \left(\frac{T_{SW1} \cdot R_{SW1}}{T_{SW1}+T_{SW2}} + \frac{T_{SW2} \cdot R_{SW2}}{T_{SW1}+T_{SW2}} + R_L \right) \right. \\
&\quad \left. + \frac{1}{3} \cdot \left(\frac{\Delta I_{L(\mathrm{PFM})}}{2}\right)^2 \cdot \left(\frac{T_{SW1} \cdot R_{SW1}}{T_{SW1}+T_{SW2}} + \frac{T_{SW2} \cdot R_{SW2}}{T_{SW1}+T_{SW2}} + R_L + R_C \right) \right) \\
&= \frac{2 \cdot I_O}{I_{\mathrm{peak}}} \\
&\quad \cdot \left(\left(\frac{I_{\mathrm{peak}}}{2}\right)^2 \cdot \left(\frac{V_O \cdot R_{SW1}}{V_I} + \frac{(V_I - V_O) \cdot R_{SW2}}{V_I} + R_L \right) \right. \\
&\quad \left. + \frac{1}{3} \cdot \left(\frac{I_{\mathrm{peak}}}{2}\right)^2 \cdot \left(\frac{V_O \cdot R_{SW1}}{V_I} + \frac{(V_I - V_O) \cdot R_{SW2}}{V_I} + R_L + R_C \right) \right)
\end{aligned} \tag{6}$$

where $\Delta I_{L(\mathrm{PFM})}$ is the ripple of the inductor current, which is almost the same as I_{peak} in the PFM dc–dc converter.

2) Gate Drive Power Dissipation: The gate capacitance of two MOSFET switches is another source of power dissipation in dc–dc converters. A dc–dc converter controls the output voltage and maintains the required load current by opening and closing two switches alternately. This process requires repeated charging of the gate capacitances of the two switches. Thus, the gate drive power dissipation is directly affected by the amount of switching per unit time, which is the switching frequency. Consequently, PWM dc–dc converters with a constant switching frequency consume a fixed gate drive power that is independent of the load condition, whereas PFM dc–dc converters consume less gate drive power as the output current diminishes. Gate drive power dissipation is roughly proportional to the input voltage, the switching frequency, and the gate charge of MOSFETs, as shown in the following equation [25]:

$$P_{\mathrm{gate\_drive}} = V_I \cdot f_S \cdot (Q_{SW1} + Q_{SW2}) \tag{7}$$

where Q_{SW1} and Q_{SW2} are the gate charges of the top MOSFET and the bottom MOSFET, respectively.

This gate drive power model can be applied to both PWM and PFM dc–dc converters in the same way, except that f_S is a constant in the PWM model, but a variable in the PFM model.

3) Controller Power Dissipation: Besides the gate drive power dissipation of the control circuit, the static power dissipation of the PWM or PFM control circuit, and the power lost in miscellaneous circuits in a dc–dc converter should be considered. Generally, controller power dissipation is independent of the load condition, which makes this power dissipation a dominant one under light loads. We characterize the controller power dissipation as

$$P_{\text{controller}} = V_I \cdot I_{\text{controller}} \tag{8}$$

where $I_{\text{controller}}$ is the current flowing into the controller of the dc–dc converter, excluding the current charging the gate capacitance.

Almost all manufacturing parameters (i.e., R_{SW1}, R_{SW2}, R_L, R_C, f_S, I_{peak}, $I_{\text{controller}}$, etc.) can be obtained from datasheets provided by the manufacturer of each component. We can build a power consumption model for the dc–dc converter with this information. We then validate our power model by comparing the power estimated by the model with figures from the dc–dc converter manufacturer's datasheets or a circuit simulation.

If we have one parameter whose value is not known, for example, $I_{\text{controller}}$, we can estimate its value from the difference between the energy consumption calculated by our energy model excluding only $I_{\text{controller}}$ terms and the energy consumption obtained from the curve of load current versus efficiency (or power consumption), which is provided by the manufacturers for a specific condition, as follows:

$$\begin{aligned}
&P_{\text{dcdc(PWM)}}(v)\\
&\quad = i_O(v)^2 \cdot \left(\frac{v}{V_I} \cdot R_{SW1} + \left(1 - \frac{v}{V_I}\right) \cdot R_{SW2} + R_L\right)\\
&\qquad + \frac{1}{3} \cdot \left(\frac{1}{2} \cdot \frac{v}{L_f \cdot f_S} \cdot \left(1 - \frac{v}{V_I}\right)\right)^2\\
&\qquad \cdot \left(\frac{v}{V_I} \cdot R_{SW1} + \left(1 - \frac{v}{V_I}\right) \cdot R_{SW2} + R_L + R_C\right)\\
&\qquad + V_I \cdot f_S \cdot (Q_{SW1} + Q_{SW2}) + V_I \cdot I_{\text{controller}}\\
&\quad = P_{\text{dcdc\_except\_}I_{\text{controller}}\text{(PWM)}}(v) + V_I \cdot I_{\text{controller}}
\end{aligned} \tag{9}$$

$$\begin{aligned}
&I_{\text{controller}}\\
&\quad = \frac{\left(P_{\text{dcdc(PWM)}}(v) - P_{\text{dcdc\_except\_}I_{\text{controller}}\text{(PWM)}}(v)\right)}{V_I}.
\end{aligned} \tag{10}$$

Since all circuit parameters except $I_{\text{controller}}$ are given, we can obtain the value of $P_{\text{dcdc\_except\_}I_{\text{controller}}\text{(PWM)}}(v)$ from our power model and the value of $P_{\text{dcdc(PWM)}}(v)$ from the datasheet for a specific output voltage (v in the previous equations) and, thus, estimate the value of $I_{\text{controller}}$. Finally, to verify the validity of our power model and the estimated parameter, we incorporate this parameter value into our power model and then see whether it estimates the power consumption of the dc–dc converter accurately for different output voltages and changed values of other parameters.

To validate these power models, we compared the efficiency curves provided by manufacturers with the curves estimated by our model for two commercial dc–dc converters, namely: 1) the TPS40009 [26] and 2) TPS62100 [27], which use PWM and PWM/PFM hybrid control, respectively. All manufacturing parameters have been extracted from datasheets. As shown in Fig. 4, the power models for both the PWM and PFM converters are accurate enough to allow us to estimate the power dissipation of real dc–dc converters.

D. *Effects of MOSFET Gate Width Sizing*

As shown in previous sections, power dissipation in dc–dc converters is affected by various parameters. These consist of the manufacturing parameters, which cannot be changed at run time, and load-dependent parameters, such as the output voltage and current of the dc–dc converter, which can be changed to suit the run-time workload, or by high-level power management techniques.

Since we are trying to reduce the total system energy consumption by means of DVS when using a given dc–dc converter, rather than proposing modification to dc–dc converters themselves, we consider most manufacturing parameters to be fixed and focus on the effects of the run-time load variation in the power dissipation. However, unlike other manufacturing parameters, the gate width of the MOSFET switches shows interesting behavior, especially in systems equipped with a PWM dc–dc converter. As the gate width of the MOSFET gets smaller, the turn-on resistance of the MOSFET increases, whereas the gate charge is reduced [2], [7]. More specifically, the turn-on resistance R_{SW} and the gate charge Q_{SW} of the MOSFET switches with a gate width of W_{SW} can be estimated as

$$R_{SW} = \frac{W_0}{W_{SW}} R_0, \qquad Q_{SW} = \frac{W_{SW}}{W_0} Q_0 \tag{11}$$

where R_0 and Q_0 are the turn-on resistance and the gate charge, respectively, of a MOSFET with a gate width of W_0.

This means that the optimal value of the gate width W, in terms of the energy consumption, will vary with the load condition because the turn-on resistance and the gate charge affect the load-dependent power (i.e., conduction power dissipation) and the load-independent power (i.e., gate drive power dissipation), respectively, in a PWM dc–dc converter. Since DVS causes a drastic variation of the load condition from one application to another, the optimal gate width of the MOSFET switches may vary with the application in a DVS-enabled system. Fig. 5 shows that the change of W affects the conversion efficiency in opposite ways under light and heavy loads.

III. DC–DC Converter-Aware Energy Management Techniques

A. *Proposed Algorithm: An Overview*

Configuring a dc–dc converter and a DVS scheme to minimize the overall energy consumption is a complex problem, as it will become apparent in the following section. To make the problem more tractable, so that it can be involved in a

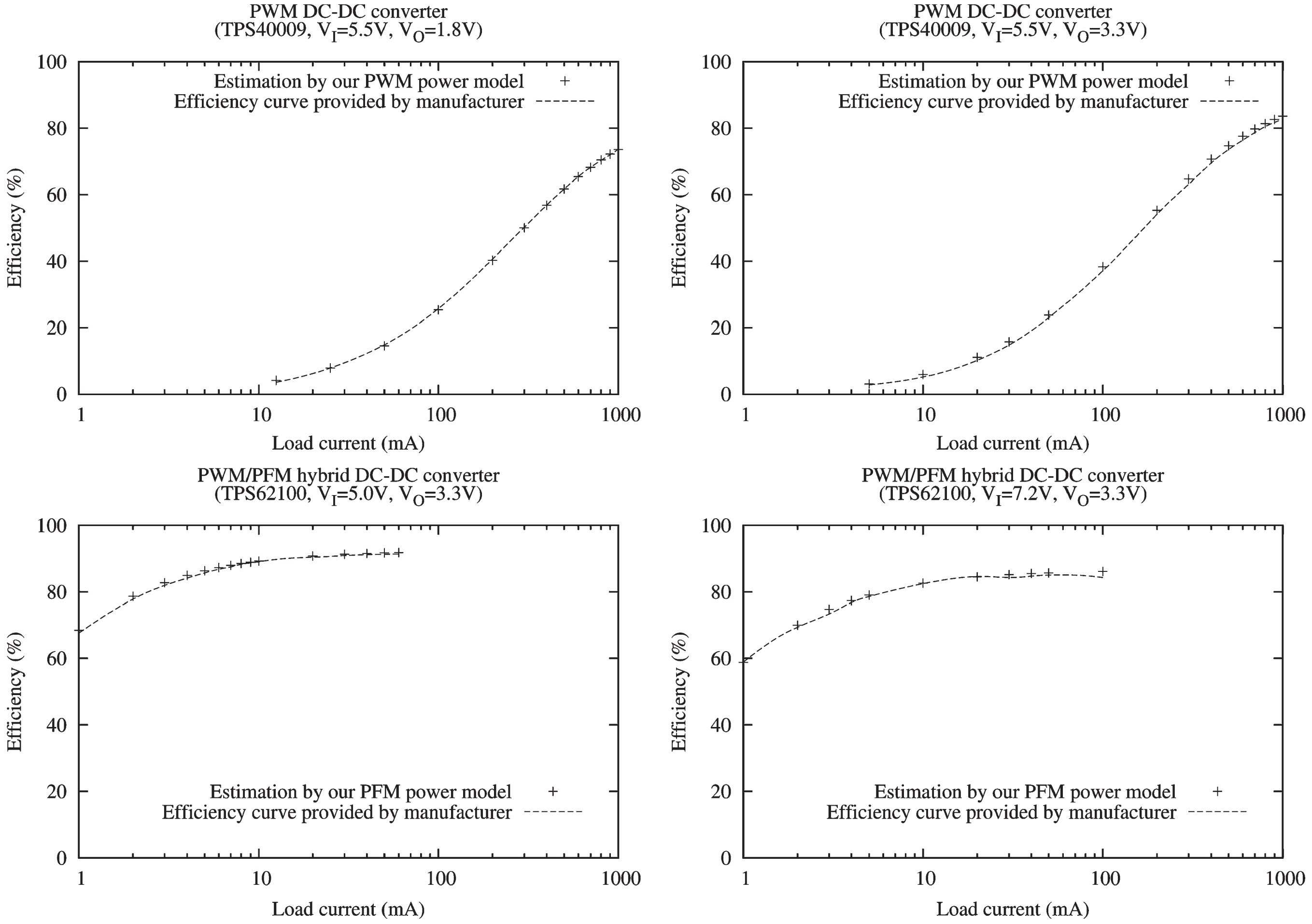


Fig. 4. Comparison of dc–dc converter efficiency curves provided by a manufacturer with the estimates from our dc–dc converter power model.

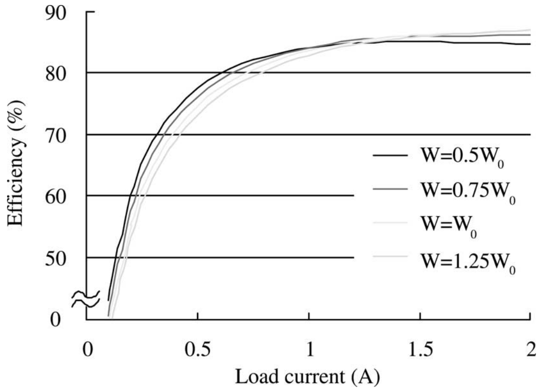


Fig. 5. Conversion efficiency of a dc–dc converter for different values of the parameter W. (The efficiency curve with $W = W_0$ is equivalent to that of a TPS40009 [26].)

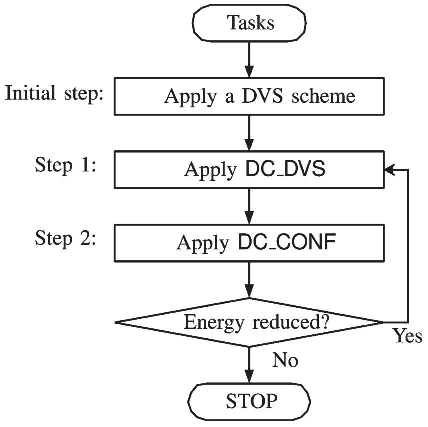


Fig. 6. Flow of our proposed iterative algorithm DC-lp.

systematic way, we propose a simple but robust framework, which is called DC-lp, for our converter-aware energy minimization algorithm. DC-lp essentially combines two core techniques, namely: 1) DC_DVS (Section III-B) and 2) DC_CONF (Section III-C). DC_DVS refines the DVS result by considering the energy efficiency of the dc–dc converter to be used, whereas DC_CONF refines the configuration of the dc–dc converter (i.e., determines the optimal value of the parameter W) from the updated DVS result. Fig. 6 shows the flow of the integrated algorithm. Initially, we are given a DVS result $\mathcal{A}$ for input tasks and a converter configuration $\mathcal{B}$. Then, the two steps in Fig. 6 are performed iteratively until there is no further reduction in the total energy consumption: In Step 1, DC_DVS is applied to $\mathcal{A}$, using $\mathcal{B}$ to produce a new DVS result $\mathcal{A}'$. In Step 2, DC_CONF is applied to $\mathcal{A}'$ to produce a new configuration $\mathcal{B}'$.

The following sections describe the two steps, each of which solves the dc–dc converter energy minimization problems. Step 1 is the converter-aware voltage scaling problem, which is to determine task and voltage schedules that minimize the total energy consumption of a system, including that of the dc–dc converter. Step 2 is the application-driven dc–dc converter optimization problem, which is to find the most energy-efficient configuration of the dc–dc converter for the application.

B. Converter-Aware Voltage Scaling Technique

For a CMOS circuit, it is well known that the CPU power P_i and the energy consumption E_i for a task J_i can be computed (assuming a fixed supply voltage) by

$$\begin{aligned} P_i &= C_{\mathrm{CPU},i} \cdot V_{dd,i}^2 \cdot f_i + V_{dd,i} \cdot I_{\mathrm{static}} + P_{\mathrm{on}} \\ E_i &= R_i \cdot P_i \end{aligned} \tag{12}$$

where $C_{\mathrm{CPU},i}$ is the average switched capacitance per clock cycle for the task, f_i is the operating frequency, $V_{dd,i}$ is the supply voltage used for the execution of the task, I_{static} is the frequency-independent static current (consisting mainly of the subthreshold leakage current), P_{on} is the inherent power consumption (which is independent of the scalable supply voltage of the CPU), and R_i is the total number of cycles required for the execution of task J_i.

However, supply voltage scaling incurs one crucial penalty, i.e., the reduced voltage increases circuit delay, which is approximately linearly proportional to the supply voltage since the circuit delay T_d can be expressed [22] as

$$T_d = \frac{C_L V_{dd}}{\mu C_{\mathrm{ox}}(D/L)(V_{dd} - V_t)^{\alpha}} \tag{13}$$

where C_L represents the total node capacitance, μ is the mobility, C_{ox} is the oxide capacitance, V_t is the threshold voltage, V_{dd} is the supply voltage for the task, α is a constant satisfying $1 < \alpha < 2$, and D and L represent the width and length of the transistors, respectively.

An instance of a task scheduling and a voltage allocation problem in a system consists of a set of tasks (or jobs) $\mathcal{J} = \{J_1, J_2, \ldots, J_N\}$ and a variable voltage range $[V_{\min}, V_{\max}]$, where N is the number of tasks.

Each task $J_i \in \mathcal{J}$ is associated with the following parameters:

a_i	arrival time of J_i;
d_i	deadline of J_i ($a_i \leq d_i$);
R_i	number of processor cycles required to complete J_i.

Since the supply voltage directly determines the processor's clock frequency [as implied in (13)], it is often convenient to consider the energy consumption to be a function of the clock frequency. Let $f_i(t)$ be the clock frequency assigned to task J_i at time t, and let $P_i(f_i(t))$ be the energy consumed in task J_i during a unit time period, starting at t. Then, the total energy consumed by the voltage scaling $\mathcal{A}_i$ of task J_i is given [11] by

$$E(\mathcal{A}_i) = \int_{t_{i,1}}^{t_{i,2}} P_i\left(f_i(t)\right) dt \tag{14}$$

where $t_{i,1}$ and $t_{i,2}$ are the starting and ending times, respectively, of the execution of task J_i. Thus, the total CPU energy consumption E_{CPU}, excluding that of the dc–dc converter, for N tasks $(J_1, J_2, \ldots, J_N)$ is

$$E_{\mathrm{CPU}} = \sum_{i=1}^{N} \int_{t_{i,1}}^{t_{i,2}} P_i\left(f_i(t)\right) dt. \tag{15}$$

Combining this equation with (1), the total energy consumption, including that of the dc–dc converter, for the tasks can be expressed as

$$E_{\mathrm{tot}} = E_{\mathrm{CPU}} + \sum_{i=1}^{N} \int_{t_{i,1}}^{t_{i,2}} P_{\mathrm{dcdc}} dt. \tag{16}$$

Note that the values of a_i, d_i, and R_i are given for task J_i, and the values of $f_i(t)$ and $P_i(f_i(t))$ vary with the dynamically scaled voltages used in running J_i and, thus, directly affect the energy consumption. A schedule of tasks is referred to as a *feasible schedule* if all the timing constraints of the tasks are satisfied. Then, the task scheduling and voltage scaling problem becomes as follows.

Problem 1: Given an instance of tasks, a dc–dc converter, and the voltage range of a processor, find a feasible task schedule and voltage scaling that minimizes E_{tot} in (16).

To reduce the complexity of Problem 1, we first propose a technique for solving a restricted version of the problem and then extend it to a full solution.

Solution to Problem 1 with a single task: From (1) and (12), we can derive the total power equation in terms of the supply voltage variable alone because a system with DVS has the maximum operating frequency, which is proportional to its operating voltage. That is, $f = \alpha V$, where α is a system-dependent constant, and thus, $P_{\mathrm{CPU}} = C_{\mathrm{CPU}} \cdot \alpha \cdot V_{dd}^3 + V_{dd} \cdot I_{\mathrm{static}} + P_{\mathrm{on}}$. Furthermore, since power consumption can also be expressed as a product of load current and supply voltage (i.e., $P = VI$), we have

$$\begin{aligned} I_{dd} &= C_{\mathrm{CPU}} \cdot \alpha \cdot V_{dd}^2 + I_{\mathrm{static}} + \frac{P_{\mathrm{on}}}{V_{dd}} \\ &= C_{\mathrm{CPU}} \cdot \alpha \cdot V_O^2 + I_{\mathrm{static}} + \frac{P_{\mathrm{on}}}{V_O} = I_O \end{aligned} \tag{17}$$

where I_{dd} is the supply current flowing into the CPU. I_O is the output current, and V_O is the output voltage of the dc–dc converter.

Fixing the value of W in (1) and (17), we can express the total power consumption P_{tot}, including that of the dc–dc converter, for each control technique, as

$$\begin{aligned}
&P_{\text{tot(PWM)}}(v)\\
&= P_{\text{CPU}}(v) + P_{\text{dcdc(PWM)}}(v)\\
&= C_{\text{CPU}}\cdot\alpha\cdot v^3 + I_{\text{static}}\cdot v + P_{\text{on}} + i_O(v)^2\\
&\quad\cdot\left(\frac{v}{V_I}\cdot R_{SW1} + \left(1-\frac{v}{V_I}\right)\cdot R_{SW2} + R_L\right)\\
&\quad + \frac{1}{3}\cdot\left(\frac{1}{2}\cdot\frac{v}{L_f\cdot f_S}\cdot\left(1-\frac{v}{V_I}\right)\right)^2\\
&\quad\cdot\left(\frac{v}{V_I}\cdot R_{SW1} + \left(1-\frac{v}{V_I}\right)\cdot R_{SW2} + R_L + R_C\right)\\
&\quad + V_I\cdot f_S\cdot(Q_{SW1}+Q_{SW2}) + V_I\cdot I_{\text{controller}}
\end{aligned}\tag{18}$$

$$\begin{aligned}
&P_{\text{tot(PFM)}}(v)\\
&= P_{\text{CPU}}(v) + P_{\text{dcdc(PFM)}}(v)\\
&= C_{\text{CPU}}\cdot\alpha\cdot v^3 + I_{\text{static}}\cdot v + P_{\text{on}} + \frac{2\cdot i_O(v)}{I_{\text{peak}}}\\
&\quad\cdot\left(\left(\frac{I_{\text{peak}}}{2}\right)^2\cdot\left(\frac{v}{V_I}\cdot R_{SW1} + \left(1-\frac{v}{V_I}\right)\cdot R_{SW2} + R_L\right)\right.\\
&\qquad\left. + \frac{1}{3}\cdot\left(\frac{I_{\text{peak}}}{2}\right)^2\cdot\left(\frac{v}{V_I}\cdot R_{SW1} + \left(1-\frac{v}{V_I}\right)\cdot R_{SW2} + R_L + R_C\right)\right)\\
&\quad + V_I\cdot\frac{2\cdot i_O(v)}{I_{\text{peak}}}\cdot\frac{(V_I - v)\cdot v}{I_{\text{peak}}\cdot L_f\cdot V_I}\\
&\quad\cdot(Q_{SW1}+Q_{SW2}) + V_I\cdot I_{\text{controller}}
\end{aligned}\tag{19}$$

where v is the scalable voltage, the only variable controlled by DVS, which is equivalent to both the supply voltage V_{dd} of the CPU and the output voltage V_O of the dc–dc converter, and $i_O(v) = C_{\text{CPU}}\cdot\alpha\cdot v^2 + I_{\text{static}} + (P_{\text{on}}/v)$.

For a task with an execution time T and a deadline D, the value of E_{tot} for the execution of the task can be obtained by simply multiplying the total power consumption $P_{\text{tot}}(v)$ by the execution time because the power loss in the dc–dc converter during standby state is negligible, i.e.,

$$E_{\text{tot}}(v) = \int_0^D P_{\text{tot}}(v)dt = \int_0^T P_{\text{tot}}(v)dt = T\cdot P_{\text{tot}}(v). \tag{20}$$

Then, applying $T = R/f = R/\alpha V$, where R is the number of cycles for the task and V is its supply voltage, to $E_{\text{tot}}(v)$ gives

$$E_{\text{tot}}(v) = \frac{R}{\alpha}\cdot\frac{P_{\text{tot}}(v)}{v}. \tag{21}$$

$E_{\text{tot}}(v)$ is not a monotonically increasing function of the output voltage. This means that using the lowest feasible voltage (or frequency) for a task does not always minimize the total energy consumption. Fig. 7 shows the relationship between $E_{\text{tot}}(v)$ and supply voltage for a system consisting of a PWM dc–dc converter and a general embedded CPU. The curve clearly indicates that the optimal voltage for $E_{\text{tot}}(v)$ is not always the lowest feasible voltage. (This system configuration, i.e., $C1$, is described in Section IV. The curve for $C2$ has a similar shape.)

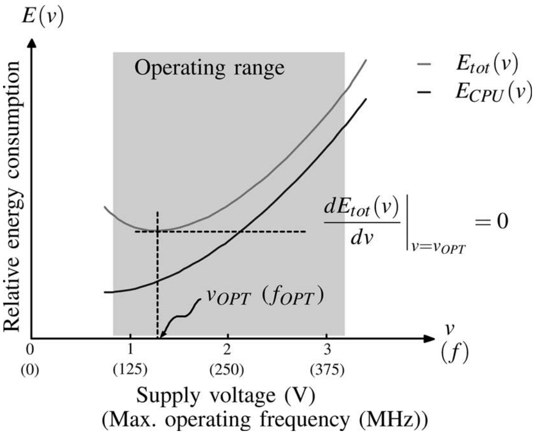


Fig. 7. Energy consumption against supply voltage for the system configuration $C1$, which is described in Section IV ($v \propto$ CPU clock frequency).

```
DC_DVS-1: DC-DC converter-aware DVS for a single task
Input: A task with arrival time $a_i$, deadline $d_i$,
       and $R_i$ cycles to execute, and processor with
       operating frequency range $[f_{min}, f_{max}]$.
Output: Frequency $f$ that minimizes $E_{tot}(\cdot)$.
• Set $f_{lowest} = \frac{R_i}{d_i - a_i}$;
• If $f_{min} < f_{lowest}$, set $f_{min} = f_{lowest}$;
• Derive $f_{OPT}$ from Eq. 21;
case
    $f_{OPT} < f_{min}$:   $f$ = $f_{min}$;
    $f_{OPT} > f_{max}$:   $f$ = $f_{max}$;
    otherwise:   $f$ = $f_{OPT}$;
endcase;
return $f$;
```

Fig. 8. Summary of the proposed algorithm for Problem 1 with a single task.

We can derive the voltage that corresponds to the lowest energy consumption by solving $dE_{\text{tot}}(v)/dv = 0$. Applying the inequality $V_{\min} \le v \le V_{\max}$, where $V_{\min}$ and $V_{\max}$ are the minimum and maximum supply voltages, respectively, allowable for a given CPU, to this solution, we obtain the voltage v_{OPT} that gives the lowest energy consumption, together with the corresponding operating frequency f_{OPT}.

Fig. 8 summarizes our procedure for dc–dc converter-aware energy-optimal DVS, which is called DC_DVS-1, to Problem 1 for a single task. DC_DVS-1 simply checks if the value of f_{OPT} in (21) is in the feasible frequency range $[f_{\min}, f_{\max}]$ for the processor and sets the energy-minimal frequency accordingly.

Solution to Problem 1 with multiple tasks: There are two approaches to solving Problem 1 for multiple tasks. One is a generic technique that is applicable to a broad class of DVS methods. The other is a fine-tuned technique, which is only applicable to a specific DVS method. Since we are interested in the problem of integrating the efficiency variation of a dc–dc converter into the existing DVS methods, we choose the former approach. To be more specific, for any (existing) DVS method, which makes no attempt to save power in the dc–dc converter,

```
DC_DVS-m: DC-DC converter-aware DVS for multiple tasks
Input: Tasks, DVS scheme, DC-DC-converter,
        and processor with operating frequency range [$f_{min}, f_{max}$].
Output: Frequency $f_i$ to each task $i$ that minimizes $E_{tot}(\cdot)$.
• Apply the DVS scheme to the input tasks and produce
  schedule $\mathcal{S}$ and voltage scaling for tasks;
foreach (schedule $\int$ of task $i$)     // $\int$: (start-time, end-time)
    • Apply DC_DVS-1 to task $i$ with [$a_i$, $d_i$] = $|\int|$
      and produce $f_i$;
endfor;
return ($f_i$ to each task, $\mathcal{S}$);
```

Fig. 9. Summary of the proposed algorithm for Problem 1 with multiple tasks.

we try to reduce the system energy consumption by allowing for the power consumption used by the dc–dc converter. The essence of our technique, which we call DC_DVS-m, is to decompose the schedule into individual tasks and then to apply DC_DVS-1, as summarized in Fig. 8, to each of partial schedule with the aim of reducing the total energy consumption of that particular task. Let E_i^{before} and E_i^{after} be the values of E_{tot} from (21) for task i before and after the application of DC_DVS-1 to that task, respectively. Then, the total amount of energy saved by DC_DVS-m compared with the existing DVS method is

$$\Delta E_{\text{tot}} = \sum_{\text{task } i} \left(E_i^{\text{before}} - E_i^{\text{after}} \right). \tag{22}$$

Note that the value of ΔE_{tot} is always positive because, for every i, $E_i^{\text{before}} - E_i^{\text{after}} > 0$. Fig. 9 summarizes the procedure. Note that DC_DVS-m preserves the task schedule produced by the original DVS method. It only updates the frequency (i.e., supply voltage) for each task. If the execution of a task spans more than one time interval (due to the preemption of the task), the intervals are merged into a single time interval, and DC_DVS-1 is applied to that interval. Precisely, note that "time interval" is an interval of time on which no voltage transition occurs to further save energy consumption [11]. The notion of time interval corresponds to that of critical interval in [11]. The merging process of intervals of a task is to find a better voltage, reflecting the power consumption of dc–dc converter as well. The assignment $[a_i, d_i] = |\int|$ in Fig. 9 merges the split into time intervals. When DC_DVS-1 has been applied to each task, the merged interval is split into the original intervals. Since DC_DVS-1 does not increase the length of the interval $[a_i, d_i]$, the restored intervals will still satisfy the deadline constraint of the task. This means that, essentially, since DC_DVS-m does not change the schedule of tasks at all, the selection order of tasks does not matter. The DVS scheme applied to the input task in Fig. 9 is preemptive. However, a DVS scheme, which is nonpreemptive, has fixed priorities, or other characteristics can also be modified in this way.

C. Application-Driven Converter Optimization

The problem of implementing a dc–dc converter that consumes the least energy consumption under the application of DVS is not simple since some parameters may conflict with each other. As mentioned in Section II-D, a parameter that has a critical effect on energy consumption is the gate width of the MOSFET switch W [see (11)] that controls the tradeoff between load-independent and load-dependent power in a significant way. (Fig. 10 shows two different energy curves obtained from experiments with a dc–dc converter using two different values of W). In this section, we show how W can be optimized to minimize the total energy consumption of the system. Note that our optimization procedure is general in that it is applicable to any of the parameters if the energy consumption can be expressed in terms of that parameter.

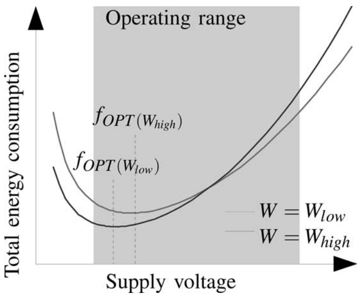


Fig. 10. Total energy consumption against W.

```
DC_CONF: Energy-minimal DC-DC converter configuration
Input: DVS voltage result for multiple tasks,
        range of $W$ for the converter [$W_{min}, W_{max}$].
Output: Value $w$ for $W$ that minimizes $E_{tot}$ under [$W_{min}, W_{max}$].
• Solve equation for $E_{tot}$ to find $w_{OPT}$;
case
    $w_{OPT} < w_{min}$:   $w = w_{min}$;
    $w_{OPT} > w_{max}$:   $w = w_{max}$;
    otherwise:   $w = w_{OPT}$;
endcase;
return $w$;
```

Fig. 11. Summary of the proposed algorithm for finding an energy-minimal configuration of a dc–dc converter.

```
DC_DISC_DVS-1: DC_DVS-1 for discretely variable voltages
Input: A task, a DC-DC-converter, a processor
        with the operating frequency range [$f_{min}, f_{max}$],
Output: Frequency $f$ that minimizes $E_{tot}(V(f))$.
• Derive $f_{OPT}$ from Eq. 21;
case
    $f_{OPT} < f_{min}$:   $f = f_{min}$;
    $f_{OPT} > f_{max}$:   $f = f_{max}$;
    otherwise,
        $E_{tot}(v_L) \geq E_{tot}(v_H)$:   $f = f_H$;
        otherwise,
            $f_L \geq f_{ideal}$:   $f = f_L$
          otherwise,
                $f = f_L$, for the period $\frac{f_{OPT}-f_H}{f_L-f_H} \cdot T$,
                and $f = f_H$, for the period $\frac{f_{OPT}-f_L}{f_H-f_L} \cdot T$,
                where $T$ is the execution time when $f = f_{OPT}$;
endcase;
return $f$;
```

Fig. 12. Summary of the algorithm for Problem 1 for a single task, using a set of discrete voltages.

Although it is not too difficult to find energy-optimal values of W and V for a single task in a specific application, in a practical point of view, it would be hard to find optimal values for multiple tasks. Solving the problem using a complicated mathematical tool would be very time consuming; thus, we simplify the problem in a way to find the best value of W after the application of DVS, independently of the dc–dc converter. In other words, for a given DVS result, we want to find a

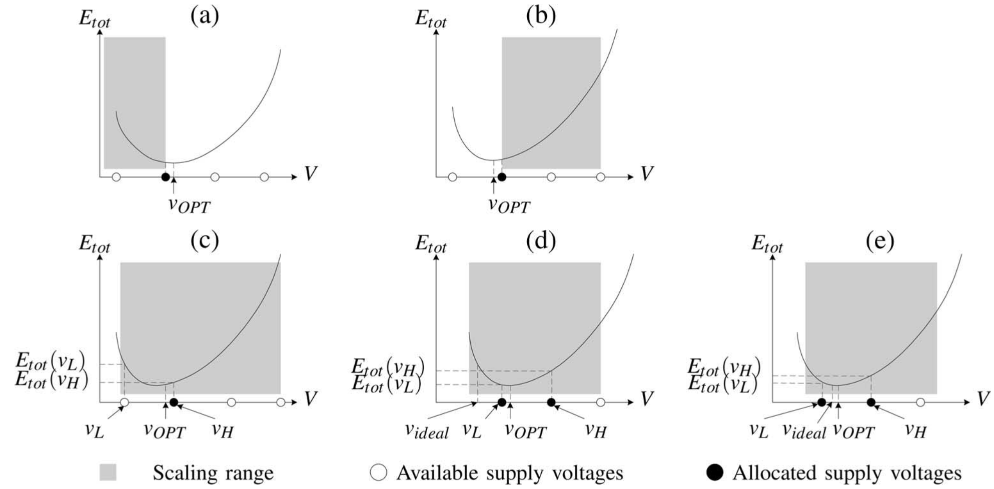


Fig. 13. Determination of the discrete voltages (frequencies) by DC_DISC_DVS-1. (a) $f_{\mathrm{OPT}} < f_{\min}$. (b) $f_{\mathrm{OPT}} > f_{\max}$. (c) $f_{\min} \leq f_{\mathrm{OPT}} \leq f_{\max}$ and $E_{\mathrm{tot}}(v_L) \geq E_{\mathrm{tot}}(v_H)$. (d) $f_{\min} \leq f_{\mathrm{OPT}} \leq f_{\max}$, $E_{\mathrm{tot}}(v_L) < E_{\mathrm{tot}}(v_H)$, and $f_L \geq f_{\mathrm{ideal}}$. (e) $f_{\min} \leq f_{\mathrm{OPT}} \leq f_{\max}$, $E_{\mathrm{tot}}(v_L) < E_{\mathrm{tot}}(v_H)$, and $f_L < f_{\mathrm{ideal}}$.

value of W in $[W_{\min}, W_{\max}]$ that minimizes the total energy consumption of the system. Note that W in the converter design must be kept within $[W_{\min}, W_{\max}]$ to avoid violating the constraints of worst-case power delivery and the maximum size of the MOSFETs. More particularly, let $v_1, v_2, \ldots, v_k$ be the voltages applied to a (scheduled) sequence of unit execution times of multiple tasks produced by a DVS scheme, and let $E_{\mathrm{tot}}(v_i, W)$ be the total energy consumption in the corresponding time for a supply voltage v_i. Then, the total energy can be expressed solely in terms of w, which represents the gate width W of the MOSFETs, as follows:

$$\begin{aligned} E_{\mathrm{tot}}(w) &= E_{\mathrm{tot}}(v_1, w) + \cdots + E_{\mathrm{tot}}(v_k, w) \\ &= \gamma_1 \cdot w + \gamma_2 \cdot \frac{1}{w} + \gamma_3 \end{aligned} \tag{23}$$

where γ_1, γ_2, and γ_3 are constants. Note that this equation is convex with respect to w. Consequently, to determine the energy-optimal value of W in $[W_{\min}, W_{\max}]$, we first derive a W value w_{OPT} that minimizes E_{tot} and then simply check whether w_{OPT} is in the range $[W_{\min}, W_{\max}]$ and finally set the minimum-energy value of W accordingly. This solution procedure, which is called DC_CONF, is shown in Fig. 11.

D. DC–DC Converter-Aware DVS for Discrete Supply Voltages

To show that DC-lp is applicable to a wide range of DVS problems, we will now consider the problem of dc–dc converter-aware DVS with *discretely variable* supply voltages (or operating frequencies). By examining the flow of DC-lp in Fig. 6, we can see that we need, at the least, to update Step 1 (i.e., DC_DVS), which solves the DVS problem for continuously variable supply voltages. In more detail, we need to introduce the constraint of discrete voltages into DC_DVS-1, which solves the dc–dc converter-aware DVS problem for a single task. Once we have developed a technique (which we will call DC_DISC_DVS-1) to solve the dc–dc converter-aware DVS problem for a single task using discrete voltages, DC_DVS-m can repeatedly use DC_DISC_DVS-1 for each of the scheduled tasks. (Note that the execution schedule of tasks will have already been obtained using an existing DVS technique.) We now explain DC_DISC_DVS-1, which is also summarized in Fig. 12.

Let $(f_1, f_2, \ldots, f_K)$ be the set of available operating frequencies, assuming $f_1 < f_2 < \cdots < f_K$. Also, let $f_{\min}$ and $f_{\max}$ refer to f_1 and f_K, respectively. For a task to be executed in a certain time interval, we determine the value of f_{OPT} in (21), which is the dc–dc converter-aware energy-optimal operating frequency for a continuously variable voltage processor. Based on that value of f_{OPT}, we try to find the most energy-efficient operating frequency (or frequencies) in the set $(f_1, f_2, \ldots, f_K)$. If $f_{\mathrm{OPT}} < f_{\min}$ or $f_{\mathrm{OPT}} > f_{\max}$, then we have no choice except to use the extreme frequencies [the cases in Fig. 13(a) and (b)].[2] However, if the value of f_{OPT} is in between $f_{\min}$ and $f_{\max}$, then we focus on the two frequencies, which are the neighbors of f_{OPT} in $(f_1, f_2, \ldots, f_K)$. Let f_L and f_H denote these two frequencies, where $f_L < f_{\mathrm{OPT}}$ and $f_{\mathrm{OPT}} < f_H$. We now compare the values of the energy consumption E_{tot} for v_L and v_H. If $E_{\mathrm{tot}}(v_L) \geq E_{\mathrm{tot}}(v_H)$, f_H is the best choice [the case in Fig. 13(c)]. Otherwise, f_L will be used. However, in this case, we have to check whether reducing the frequency from f_{OPT} to f_L will violate the task's timing constraint. Let f_{ideal} be the lowest frequency that can be applied to the task without violating the timing constraint. We now see whether the value of f_L is greater than f_{ideal}. If it is, f_L will be used [the case in Fig. 13(d)]. Otherwise [the case in Fig. 13(e)], a combination of f_L and f_H is required. The time periods for using f_L and f_H are simply determined from f_L, f_H, and the given time interval for the task.

IV. Experimental Results

We implemented our proposed dc–dc converter-aware power management techniques in C++ and tested them on a set

[2] v_x represents the voltage corresponding to the frequency f_x.

TABLE I
EXPERIMENTAL SYSTEM CONFIGURATIONS

System configuration	DC-DC Controller	MOSFET ($SW1$, $SW2$)	Inductor (L)	Capacitor (C)	Load (Refer Table II)
C1	TPS40009, PWM (Texas Instruments)	Si4946EY, N-type (Vishay Siliconix)	DR74-6R8, 6.8 μH (Coiltronics)	20TQC22M, 22 μF (Sanyo)	***P1***
C2	TPS62100, PWM/PFM hybrid (Texas Instruments)	Integrated with controller, P-type for $SW1$, N-type for $SW2$	DO1608C-153ML, 15 μH (Coilcraft)	JMK212BJ106MG, 10 μF (Taiyo Yuden)	***P2***

TABLE II
EXPERIMENTAL CPU CONFIGURATIONS

CPU configuration	Supply voltage (v)	Operating frequency (f)	Static current (I_{static})	Base power (P_{on})	Total power @ max. supply voltage (P_{CPU}@V_{max})
P1	0.8 ∼ 3.2 V	100 ∼ 400 MHz	100 mA	150 mW	5.85 W
P2	1.8 ∼ 3.6 V	4 ∼ 8 MHz	300 μA	500 μW	5.32 mW

TABLE III
COMPARISONS OF ENERGY CONSUMED BY NO DVS (NO_DVS), A CONVENTIONAL DVS SCHEME (DVS_ONLY), AND OUR DC–DC CONVERTER-AWARE DVS TECHNIQUE (DC_DVS) FOR BENCHMARK PROGRAMS. SYSTEM CONFIGURATION IS $C1$, AND w IS THE CONFIGURATION PARAMETER OF THE DC–DC CONVERTER USED

Design	Task deadline constraint	Normalized energy consumption with w=0.5W_0			Reduction over DVS_ONLY	Normalized energy consumption with w=0.75W_0			Reduction over DVS_ONLY
		NO_DVS	DVS_ONLY	DC_DVS		NO_DVS	DVS_ONLY	DC_DVS	
MPEG	D	1	0.447	0.395	11.7%	1	0.479	0.416	13.1%
	D×0.5	1	0.429	0.429	0%	1	0.445	0.445	0.0%
	D×2.0	1	0.447	0.395	11.7%	1	0.479	0.416	13.1%
VP	D	1	0.439	0.395	10.1%	1	0.471	0.416	11.5%
	D×0.5	1	0.437	0.437	0%	1	0.453	0.453	0%
	D×2.0	1	0.447	0.395	11.7%	1	0.479	0.416	13.1%
AVN	D	1	0.768	0.768	0%	1	0.773	0.733	0%
	D×0.9	1	0.908	0.908	0%	1	0.911	0.911	0%
	D×2.0	1	0.401	0.401	0%	1	0.420	0.420	0%
CNC	D	1	0.433	0.433	0%	1	0.449	0.449	0%
	D×0.6	1	0.749	0.749	0%	1	0.755	0.755	0%
	D×2.0	1	0.439	0.395	10.1%	1	0.470	0.416	11.4%

Design	Task deadline constraint	Normalized energy consumption with w=W_0			Reduction over DVS_ONLY	Normalized energy consumption with w=1.25W_0			Reduction over DVS_ONLY
		NO_DVS	DVS_ONLY	DC_DVS		NO_DVS	DVS_ONLY	DC_DVS	
MPEG	D	1	0.509	0.435	14.4%	1	0.537	0.452	15.7%
	D×0.5	1	0.459	0.459	0%	1	0.472	0.472	0%
	D×2.0	1	0.509	0.435	14.4%	1	0.537	0.452	15.7%
VP	D	1	0.499	0.435	12.8%	1	0.526	0.452	14.0%
	D×0.5	1	0.467	0.467	0%	1	0.479	0.479	0%
	D×2.0	1	0.509	0.435	14.4%	1	0.537	0.452	15.7%
AVN	D	1	0.778	0.778	0%	1	0.781	0.781	0%
	D×0.9	1	0.912	0.921	0%	1	0.914	0.914	0%
	D×2.0	1	0.437	0.437	0%	1	0.453	0.453	0%
CNC	D	1	0.464	0.464	0%	1	0.477	0.477	0%
	D×0.6	1	0.760	0.760	0%	1	0.764	0.764	0%
	D×2.0	1	0.498	0.435	12.7%	1	0.525	0.452	13.9%

of benchmark designs [18], [23], [24]. The evaluation was conducted in two parts: 1) For a given configuration comprising a dc–dc converter, a task set, and a voltage range, we want to know to what extent the voltage scaling achieved by DC_DVS (see Figs. 8 and 9) reduces the energy consumption of the system, including that of the dc–dc converter. 2) For a range of values of W in the dc–dc converter, a task set, and a voltage range, we want to know how effectively DC-lp (see Fig. 6) determines the converter configuration and voltage scaling to reduce the total energy consumption.

1) Target System Configurations: We performed our experiments for two system configurations. $C1$ corresponds to general embedded CPUs [e.g., PXA275 (Intel)], which have operating currents of a few hundreds of milliamperes to a few amperes and powered by a PWM dc–dc converter. $C2$ represents ultralow-power CPUs [e.g., MSP430C11x1 (Texas Instruments) and PIC12F629 (Microchip)], which have operating currents of a few hundreds of microamperes and powered by a PWM/PFM hybrid dc–dc converter. These configurations are summarized in Tables I and II.

2) Assessing the Effectiveness of DC_DVS*:* We tested DC_DVS on a number of real-time task sets, including a videophone application [18], an avionics application (AVN) [23], and a computerized numerical control (CNC) machine controller application [24]. For a fixed configuration of dc–dc converter ($w = 0.5W_0, 0.75W_0, W_0,$ or $1.25W_0$, where W_0 is the gate width of the original MOSFET) and target system ($C1$ or $C2$, which are described in Table I), Tables III and IV compare the energy consumed by a scheme (NO_DVS) that always applies the fastest clock frequency to every task, a

TABLE IV
COMPARISONS OF ENERGY CONSUMED BY NO DVS (NO_DVS), A CONVENTIONAL DVS SCHEME (DVS_ONLY), AND OUR DC–DC CONVERTER-AWARE DVS TECHNIQUE (DC_DVS) FOR BENCHMARK PROGRAMS. SYSTEM CONFIGURATION IS $C2$, AND w IS THE CONFIGURATION PARAMETER OF THE DC–DC CONVERTER USED

Design	Task deadline constraint	Normalized energy consumption with w=0.5W_0			Reduction over	Normalized energy consumption with w=0.75W_0			Reduction over
		NO_DVS	DVS_ONLY	DC_DVS	DVS_ONLY	NO_DVS	DVS_ONLY	DC_DVS	DVS_ONLY
MPEG	D	1	0.927	0.871	6.1%	1	0.925	0.871	5.9%
	D×0.5	1	0.927	0.871	6.1%	1	0.925	0.871	5.9%
	D×2.0	1	0.927	0.871	6.1%	1	0.925	0.871	5.9%
VP	D	1	0.927	0.871	6.1%	1	0.925	0.871	5.9%
	D×0.5	1	0.916	0.871	4.9%	1	0.914	0.871	4.8%
	D×2.0	1	0.927	0.871	6.1%	1	0.925	0.871	5.9%
AVN	D	1	0.913	0.913	0%	1	0.914	0.914	0%
	D×0.9	1	0.963	0.963	0%	1	0.964	0.964	0%
	D×2.0	1	0.927	0.871	6.1%	1	0.925	0.871	5.9%
CNC	D	1	0.918	0.871	5.2%	1	0.916	0.871	5.0%
	D×0.6	1	0.913	0.913	0%	1	0.913	0.913	0%
	D×2.0	1	0.927	0.871	6.1%	1	0.925	0.871	5.9%

Design	Task deadline constraint	Normalized energy consumption with w=W_0			Reduction over	Normalized energy consumption with w=1.25W_0			Reduction over
		NO_DVS	DVS_ONLY	DC_DVS	DVS_ONLY	NO_DVS	DVS_ONLY	DC_DVS	DVS_ONLY
MPEG	D	1	0.924	0.871	5.7%	1	0.922	0.871	5.6%
	D×0.5	1	0.924	0.871	5.7%	1	0.922	0.871	5.6%
	D×2.0	1	0.924	0.871	5.7%	1	0.922	0.871	5.6%
VP	D	1	0.924	0.871	5.7%	1	0.922	0.871	5.6%
	D×0.5	1	0.913	0.871	4.6%	1	0.911	0.871	4.5%
	D×2.0	1	0.924	0.871	5.7%	1	0.922	0.871	5.6%
AVN	D	1	0.914	0.914	0%	1	0.915	0.915	0%
	D×0.9	1	0.964	0.964	0%	1	0.964	0.964	0%
	D×2.0	1	0.924	0.871	5.7%	1	0.922	0.871	5.6%
CNC	D	1	0.915	0.871	4.8%	1	0.913	0.871	4.7%
	D×0.6	1	0.914	0.914	0%	1	0.914	0.914	0%
	D×2.0	1	0.924	0.871	5.7%	1	0.922	0.871	5.6%

TABLE V
COMPARISONS OF ENERGY CONSUMED BY NO DVS (NO_DVS), A CONVENTIONAL DVS SCHEME (DVS_ONLY), AND OUR INTEGRATED CONVERTER-AWARE DVS (DC-lp). SYSTEM CONFIGURATION IS $C1$

Design	Normalized energy consumption			Reduction over	Normalized energy consumption			Reduction over
	$w = W_0$		w=[0.5W_0, 1.5W_0]		$w = 1.25W_0$		w=[0.5W_0, 1.5W_0]	
	NO_DVS	DVS_ONLY	DC-lp	DVS_ONLY	NO_DVS	DVS_ONLY	DC-lp	DVS_ONLY
MPEG	1	0.509	0.403	20.7%	1	0.537	0.403	24.8%
VP	1	0.499	0.403	19.2%	1	0.526	0.403	23.3%
AVN	1	0.778	0.777	0.1%	1	0.781	0.777	0.6%
CNC	1	0.464	0.442	4.5%	1	0.477	0.442	7.2%

second scheme (DVS_ONLY [11]) that performs an energy-optimal voltage scaling without considering the energy consumption in the dc–dc converter, and our dc–dc converter-aware scheme (DC_DVS). Note that Motion Pictures Expert Group (MPEG) is a single task, but the rest of the applications involve multiple tasks. Therefore, DC_DVS-1 is applied to MPEG, and DC_DVS-m is applied to the rest. Each design was tested three times with a normal deadline D, a reduced deadline, and an extended deadline, with the values shown in Table III. The deadlines of AVN and CNC could not be reduced to 50% because the resulting schedules are infeasible, even using the highest voltage; thus, these changes are reduced to 10% and 40%, respectively. In summary, DC_DVS is able to reduce the total energy consumption by up to 15.7% for $C1$ and 6.1% for $C2$, when compared with conventional DVS optimization techniques, across four different configurations of dc–dc converters.

3) Assessing the Effectiveness of DC-lp*:* Table V shows the amount of energy consumed using NO_DVS and DVS_ONLY with fixed values of w ($w = W_0$ and $w = 1.25W_0$) and using DC-lp with $[W_{\min}, W_{\max}] = [0.5W_0, 1.5W_0]$. The system configuration is $C1$ throughout. The comparisons reveal that DC-lp performs well both in terms of voltage scaling and in the selection of a converter configuration to reduce the total energy consumption. It saves a maximum of 24.8% more energy than DVS_ONLY. This strongly implies that the problem of selecting a dc–dc converter configuration that is best suited to the target application program is at least as important as the problem of voltage scaling in reducing energy consumption.

4) Assessing the Effectiveness of DC_DISC_DVS *and* DC_DISC-lp*:* Tables VI and VII compare the energy consumed under DC_DISC_DVS (which augments DC_DVS with support for the discrete voltage constraint), with the energy consumption using conventional DVS [21], for configurations $C1$ and $C2$, respectively. Table VIII compares the energy consumed using DC_DISC-lp, which replaces DC-lp. The sets of voltages we used in the experiments were {0.8 V, 1.4 V, 2.0 V, 2.6 V, 3.2 V} for $C1$ and {1.8 V, 2.25 V, 2.7 V,

TABLE VI
ENERGY CONSUMPTION USING DC_DISC_DVS COMPARED WITH A CONVENTIONAL DVS METHOD [21], USING {0.8 V, 1.4 V, 2.0 V, 2.6 V, 3.2 V}, FOR THE CONFIGURATION $C1$

Design	Task deadline constraint	Normalized energy consumption with w=0.5W_0			Reduction over	Normalized energy consumption with w=0.75W_0			Reduction over
		NO_DVS	DISC_DVS	DC_DISC_DVS	DISC_DVS	NO_DVS	DISC_DVS	DC_DISC_DVS	DISC_DVS
MPEG	D	1	0.447	0.404	9.6%	1	0.479	0.423	11.7%
	D×0.5	1	0.441	0.441	0%	1	0.458	0.458	0%
	D×2.0	1	0.447	0.404	9.6%	1	0.479	0.423	11.7%
VP	D	1	0.445	0.404	9.2%	1	0.477	0.423	11.3%
	D×0.5	1	0.450	0.450	0%	1	0.467	0.467	0%
	D×2.0	1	0.447	0.404	9.6%	1	0.479	0.423	11.7%
AVN	D	1	0.774	0.774	0%	1	0.779	0.779	0%
	D×0.9	1	0.916	0.916	0%	1	0.918	0.918	0%
	D×2.0	1	0.407	0.404	0.7%	1	0.427	0.423	0.9%
CNC	D	1	0.441	0.441	0%	1	0.457	0.457	0%
	D×0.6	1	0.753	0.753	0%	1	0.759	0.759	0%
	D×2.0	1	0.444	0.404	9.1%	1	0.476	0.423	11.1%
Design	Task deadline constraint	Normalized energy consumption with w=W_0			Reduction over	Normalized energy consumption with w=1.25W_0			Reduction over
		NO_DVS	DISC_DVS	DC_DISC_DVS	DISC_DVS	NO_DVS	DISC_DVS	DC_DISC_DVS	DISC_DVS
MPEG	D	1	0.509	0.440	13.6%	1	0.537	0.455	15.2%
	D×0.5	1	0.472	0.472	0%	1	0.485	0.485	0%
	D×2.0	1	0.509	0.440	13.6%	1	0.537	0.455	15.2%
VP	D	1	0.506	0.440	13.1%	1	0.533	0.455	14.6%
	D×0.5	1	0.481	0.481	0%	1	0.493	0.493	0%
	D×2.0	1	0.509	0.440	13.6%	1	0.537	0.455	15.2%
AVN	D	1	0.784	0.784	0%	1	0.787	0.787	0%
	D×0.9	1	0.920	0.920	0%	1	0.921	0.921	0%
	D×2.0	1	0.444	0.440	1.0%	1	0.460	0.455	1.2%
CNC	D	1	0.472	0.472	0%	1	0.485	0.485	0%
	D×0.6	1	0.763	0.763	0%	1	0.768	0.768	0%
	D×2.0	1	0.505	0.440	12.9%	1	0.532	0.455	14.5%

TABLE VII
ENERGY CONSUMPTION USING DC_DISC_DVS COMPARED WITH A CONVENTIONAL DVS METHOD [21], USING {1.8 V, 2.25 V, 2.7 V, 3.15 V, 3.6 V}, FOR THE CONFIGURATION $C2$

Design	Task deadline constraint	Normalized energy consumption with w=0.5W_0			Reduction over	Normalized energy consumption with w=0.75W_0			Reduction over
		NO_DVS	DISC_DVS	DC_DISC_DVS	DISC_DVS	NO_DVS	DISC_DVS	DC_DISC_DVS	DISC_DVS
MPEG	D	1	0.927	0.874	5.7%	1	0.925	0.874	5.6%
	D×0.5	1	0.927	0.874	5.7%	1	0.925	0.874	5.6%
	D×2.0	1	0.927	0.874	5.7%	1	0.925	0.874	5.6%
VP	D	1	0.927	0.874	5.7%	1	0.925	0.874	5.6%
	D×0.5	1	0.927	0.874	5.7%	1	0.919	0.874	4.9%
	D×2.0	1	0.927	0.874	5.7%	1	0.925	0.874	5.6%
AVN	D	1	0.916	0.916	0%	1	0.916	0.916	0%
	D×0.9	1	0.967	0.967	0%	1	0.967	0.967	0%
	D×2.0	1	0.927	0.874	5.7%	1	0.925	0.874	5.6%
CNC	D	1	0.919	0.874	4.9%	1	0.918	0.874	4.8%
	D×0.6	1	0.916	0.916	0%	1	0.916	0.916	0%
	D×2.0	1	0.927	0.874	5.7%	1	0.925	0.874	5.6%
Design	Task deadline constraint	Normalized energy consumption with w=W_0			Reduction over	Normalized energy consumption with w=1.25W_0			Reduction over
		NO_DVS	DISC_DVS	DC_DISC_DVS	DISC_DVS	NO_DVS	DISC_DVS	DC_DISC_DVS	DISC_DVS
MPEG	D	1	0.924	0.873	5.4%	1	0.922	0.873	5.3%
	D×0.5	1	0.924	0.873	5.4%	1	0.922	0.873	5.3%
	D×2.0	1	0.924	0.873	5.4%	1	0.922	0.873	5.3%
VP	D	1	0.924	0.873	5.4%	1	0.922	0.873	5.3%
	D×0.5	1	0.917	0.873	4.8%	1	0.916	0.873	4.6%
	D×2.0	1	0.924	0.873	5.4%	1	0.922	0.873	5.3%
AVN	D	1	0.917	0.917	0%	1	0.917	0.917	0%
	D×0.9	1	0.967	0.967	0%	1	0.967	0.967	0%
	D×2.0	1	0.924	0.873	5.4%	1	0.922	0.873	5.3%
CNC	D	1	0.916	0.873	4.7%	1	0.915	0.873	4.5%
	D×0.6	1	0.916	0.916	0%	1	0.917	0.917	0%
	D×2.0	1	0.924	0.873	5.4%	1	0.922	0.873	5.3%

3.15 V, 3.6 V} for $C2$. From Table VIII, we can see that the energy reductions are comparable to those achieved by continuously variable voltages, as shown in Table V.

V. CONCLUSION

A dc–dc converter is an essential component in voltage scaling, but 10% to 40% of the total system energy is consumed

TABLE VIII
ENERGY CONSUMPTION USING DC_DISC-lp COMPARED WITH A CONVENTIONAL DVS METHOD [21], USING {0.8 V, 1.4 V, 2.0 V, 2.6 V, 3.2 V}, FOR THE CONFIGURATION $C1$

Design	Normalized energy consumption			Reduction	Normalized energy consumption			Reduction
	$w = W_0$		w=[0.5W_0, 1.5W_0]	over	$w = 1.25W_0$		w=[0.5W_0, 1.5W_0]	over
	NO_DVS	DISC_DVS	DC_DISC-lp	DISC_DVS	NO_DVS	DISC_DVS	DC_DISC-lp	DISC_DVS
MPEG	1	0.509	0.413	18.7%	1	0.537	0.413	23.0%
VP	1	0.506	0.413	18.3%	1	0.533	0.413	22.5%
AVN	1	0.784	0.783	0.1%	1	0.787	0.783	0.6%
CNC	1	0.472	0.451	4.5%	1	0.485	0.451	7.1%

by the converter itself. We have proposed an effective way to integrate the optimization of a dc–dc converter into well-known DVS power-saving techniques. Specifically, we have proposed a dc–dc converter-aware low-power DVS technique, which is called DC-lp, in which two core subproblems, namely: 1) the dc–dc converter-aware energy-minimal DVS problem and 2) the converter configuration selection problem, are effectively solved and integrated. Our experimental results show that DC-lp is able to restore up to 24.8% of energy loss compared to a conventional DVS approach. We believe that a dc–dc converter-aware power management scheme is essential in embedded systems equipped with variable voltage processors.

ACKNOWLEDGMENT

The authors would like to thank the ICT at Seoul National University for providing the research facilities for this study.

REFERENCES

[1] Y. Choi, N. Chang, and T. Kim, "DC-DC converter-aware power management for battery-operated embedded systems," in *Proc. ACM/IEEE Des. Autom. Conf.*, 2005, pp. 895–900.

[2] V. Kursun, S. G. Narendra, V. K. De, and E. G. Friedman, "Monolithic DC-DC converter analysis and MOSFET gate voltage optimization," in *Proc. IEEE Int. Symp. Quality Electron. Des.*, 2003, pp. 279–284.

[3] T. Ninomiya, N. Matsumoto, M. Nakahara, and K. Harada, "Static and dynamic analysis of zero-voltage-switched half-bridge converter with PWM control," in *Proc. IEEE Power Electron. Spec. Conf.*, 1991, pp. 230–237.

[4] M. M. Jovanovic, M. T. Zhang, and F. C. Lee, "Evaluation of synchronous-rectification efficiency improvement limits in forward converters," *IEEE Trans. Ind. Electron.*, vol. 42, no. 4, pp. 387–395, Aug. 1995.

[5] W. Chen, G. Hua, D. Sable, and F. Lee, "Design of high efficiency, low profile, low voltage converter with integrated magnetics," in *Proc. Appl. Power Electron. Conf. and Expo.*, 1997, pp. 911–917.

[6] V. Kurson, S. G. Narendra, V. K. De, and E. G. Friedman, "Efficiency analysis of a high frequency buck converter for on-chip integration with a dual V_{DD} microprocessor," in *Proc. Eur. Solid-State Circuits Conf.*, 2002, pp. 743–746.

[7] A. Stratakos, "High-efficiency low-voltage DC-DC conversion for portable applications," Ph.D. dissertation, Univ. California, Berkeley, CA, 1999.

[8] T. Simunic, L. Benini, and G. De Micheli, "Cycle-accurate simulation of energy consumption in embedded systems," in *Proc. ACM/IEEE Des. Autom. Conf.*, 1999, pp. 867–872.

[9] M. Pedram and Q. Wu, "Design considerations for battery-powered electronics," in *Proc. ACM/IEEE Des. Autom. Conf.*, 1999, pp. 861–866.

[10] D. Rakhmatov, S. Vrudhula, and C. Chakrabarti, "Battery-conscious task sequencing for portable devices including voltage/clock scaling," in *Proc. ACM/IEEE Des. Autom. Conf.*, 2002, pp. 189–194.

[11] F. Yao, A. Demers, and A. Shenker, "A scheduling model for reduced CPU energy," in *Proc. IEEE Found. Comput. Sci.*, 1995, pp. 374–482.

[12] G. Quan and X. S. Hu, "Minimum energy fixed-priority scheduling for variable voltage processors," in *Proc. IEEE Des. Autom. Test Eur.*, 2002, pp. 782–787.

[13] W. Kim, D. Shin, H. S. Yun, J. Kim, and S. L. Min, "Performance comparison of dynamic voltage scaling algorithms for hard real-time systems," in *Proc. IEEE Real-Time and Embedded Technol. and Appl. Symp.*, 2002, pp. 219–228.

[14] Y. Shin, K. Choi, and T. Sakurai, "Power optimization of real-time embedded systems on variable speed processors," in *Proc. IEEE Int. Conf. Comput.-Aided Des.*, 2000, pp. 365–368.

[15] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," in *Proc. ACM Symp. Operating Syst. Principles*, 2001, pp. 89–102.

[16] H. Aydin, R. Melhem, D. Mosse, and P. M. Alvarez, "Dynamic and aggressive scheduling techniques for power-aware real-time systems," in *Proc. IEEE Real-Time Syst. Symp.*, 2001, pp. 89–102.

[17] W. Kim, J. Kim, and S. L. Min, "A dynamic voltage scaling algorithm for dynamic-priority hard real-time systems using slack time analysis," in *Proc. IEEE Des. Autom. Test Eur.*, 2002, pp. 788–794.

[18] D. Shin, J. Kim, and S. Lee, "Intra-task voltage scheduling for low-energy hard real-time applications," *IEEE Des. Test Comput.*, vol. 18, no. 2, pp. 20–30, Mar. 2001.

[19] F. Gruian, "Hard real-time scheduling using stochastic data and DVS processors," in *Proc. IEEE Int. Symp. Low Power Electron. and Des.*, 2001, pp. 46–51.

[20] J. Seo, T. Kim, and K. Chung, "Profile-based optimal intra-task voltage scheduling for hard real-time applications," in *Proc. ACM/IEEE Des. Autom. Conf.*, 2004, pp. 87–92.

[21] W. Kwon and T. Kim, "Optimal voltage allocation techniques for dynamically variable voltage processors," in *Proc. ACM/IEEE Des. Autom. Conf.*, 2003, pp. 125–130.

[22] L. H. Chandrasena, P. Chandrasena, and M. J. Liebelt, "An energy efficient rate selection algorithm for voltage quantized dynamic voltage scaling," in *Proc. Int. Symp. Syst. Synth.*, 2001, pp. 124–129.

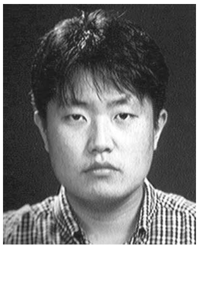
[23] C. Locke, D. Vogel, and T. Mesler, "Building a predictable avionics platform in Ada: A case study," in *Proc. IEEE Real-Time Syst. Symp.*, 1991, pp. 181–189.

[24] N. Kim, M. Ryu, S. Hong, M. Saksena, C. Choi, and H. Shin, "Visual assessment of a real-time system design: A case study on a CNC controller," in *Proc. IEEE Real-Time Syst. Symp.*, 1996, pp. 300–310.

[25] Linear Technology, *LTC3445—I2C Controllable Buck Regulator with Two LDOs in a 4 mm × 4 mm QFN*. [Online]. Available: http://www.linear.com/pc/productDetail.jsp?navId=H0,C1,C1003,C1042,C1032,C1064,P8250

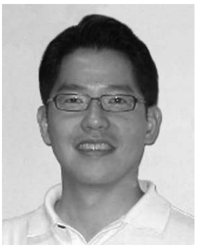
[26] Texas Instruments, *TPS40009—Low-Input High-Efficiency Synchronous Buck Controller*. [Online]. Available: http://focus.ti.com/docs/prod/folders/print/tps40009.html

[27] Texas Instruments, *TPS62100—Multimode Low-Power Buck Converter*. [Online]. Available: http://focus.ti.com/docs/prod/folders/print/tps62100.html



Yongseok Choi (S'01) received the B.S. and M.S. degrees in computer science and engineering from Seoul National University, Seoul, Korea, in 2000 and 2002, respectively. He is currently working toward the Ph.D. degree at the School of Computer Science and Engineering, Seoul National University.

His research interests include embedded systems design and system-level low-power design.



Naehyuck Chang (M'97–SM'05) received the B.S., M.S., and Ph.D. degrees from Seoul National University, Seoul, Korea, in 1989, 1992, and 1996, respectively.

He has been with the School of Computer Science and Engineering, Seoul National University, since 1997, where he is currently an Associate Professor. His research interest includes embedded systems and low-power systems. He has published more than 70 technical papers in these areas.

Dr. Chang is a member of the Association for Computing Machinery/Special Interest Group on Design Automation (ACM SIGDA). He currently serves on the Technical Program Committee of ACM SIGDA and the IEEE Circuits and Systems Society conferences and symposiums such as DAC, ICCAD, ISLPED, and ISQED. He is currently an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the *Journal of Low-Power Electronics*, and the *Journal of Embedded Computing*.

Taewhan Kim (M'93) received the B.S. degree in computer science and statistics and the M.S. degree in computer science from Seoul National University, Seoul, Korea, in 1985 and 1987, respectively, and the Ph.D. degree in computer science from the University of Illinois, Urbana–Champaign, in 1993.

From 1993 to 1998, he was with Lattice Semiconductor Corporation and Synopsys Inc., where he was involved in logic and high-level synthesis, and from August 1998 to 2003, he was with the Department of Electrical Engineering and Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, Korea. Currently, he is a Professor with the School of Electrical Engineering and Computer Science, Seoul National University. His research interests are in the area of computer-aided design of integrated circuits and combinatorial optimizations.