

A Software Implementation of Minimum Energy Point Tracking Algorithm for Microprocessors

SHENGYU LIU^{1,a)} JUN SHIOMI¹ TOHRU ISHIHARA¹ HIDETOSHI ONODERA¹

Abstract: A minimum energy point (MEP) is defined as a pair of supply voltage (V_{DD}) and threshold voltage (V_{TH}) of a circuit, which minimizes the energy consumption of the circuit under a specific performance constraint. In this paper, a software implementation of an existing MEP tracking algorithm which minimizes the energy consumption of target devices at runtime under a wide process, voltage and temperature (PVT) condition is proposed. By exploiting monitor circuits integrated into a target processor, the proposed power management software autonomously optimizes V_{DD} and V_{TH} at runtime so that the processor can operate at MEPs even if MEPs dynamically shift due to a PVT fluctuation and change in the performance constraint. A 32-bit RISC processor chip fabricated with a 65-nm process technology demonstrates that the proposed MEP tracking system consisting of interface circuits mapped on an FPGA and the power management software running on a host computer can accurately track the MEP of the processor chip at runtime even if PVT conditions and performance constraint widely change.

1. Introduction

Electronic device has already been spread in every corner of world. With the increasing number of these electronic devices such as computing servers in data centers, the consumption of the electric power has been increasing rapidly. At the same time, the improvement of performance on a single device also causes a rapid increase of the power consumption in electronic systems. Due to the increasing trend of Internet of Things (IoT), computer systems are embedded in different kinds of items and they are connected to Internet, which causes an increase of entire power consumption in the highly information-oriented society. Typically, services provided by computer systems are implemented by software programs due to their flexibility and portability. Those programs are monitored and controlled by an operating system (OS), which coordinates the various requirements from applications, such as low power consumption, dependability and real-time responsiveness. Therefore, the OS plays an important role in achieving energy efficiency without sacrificing reliability and responsiveness of the system. In this paper, we present a software implementation of a minimum energy point (MEP) tracking algorithm proposed in [1]. The MEP tracking algorithm finds a pair of supply voltage (V_{DD}) and body bias (V_{BB}), which minimizes the power consumption of a processor under a certain demand of performance only by obtaining current status of the target device. The goal of this work is to implement the MEP tracking algorithm as a sub-function of OS, which keeps underlying microprocessors always running at MEP with coordinately consid-

ering other requirements such as system reliability and real-time responsiveness. Since the algorithm is implemented by a software sub-function, it is flexible and portable. For example, aggressive voltage scaling by the MEP tracking algorithm sometimes increases sensitivity to noises and degrades the reliability of the system, which can be a fatal disadvantage of a specific type of applications such as a mission critical system. If the algorithm is implemented by a software program, it is possible to flexibly coordinate the requirements of energy efficiency and system reliability appropriately. Moreover, the software implementation makes it very simple for porting the algorithm into not only power managers for IoT devices but also OSs running on general purpose processors used in data centers and cloud servers.

The rest of the paper is organized as follows. In section 2, the history of low-power methods and related research are shortly reviewed. An architecture of MEP tracking software and hardware supports required for the software implementation of the MEP tracking algorithm are presented in section 3. Section 4 shows test and evaluation results of the MEP tracking system. Section 5 concludes the paper.

2. Related Work

2.1 Dynamic Power Management

The MEP tracking system is based on a system design policy called dynamic power management (DPM). The concept of DPM is to provide a required performance with a minimum number of active components or a minimum load on such components [2]. DPM achieves energy efficient computation by selectively turning off (or reducing the performance of) system components when they are idle. Therefore, to achieve the energy savings by DPM, it is necessary to predict the future idle periods of the sys-

¹ Department of Communications and Computer Engineering, Graduate School of Informatics, Kyoto University

^{a)} liusy@vlsi.kuee.kyoto-u.ac.jp

tem components in a certain level of confidence. This is because the energy savings can be achieved only if the energy overhead for turning off and waking up the system components is smaller than the energy consumed in the idle period of the system components. If the idle period is too short, the energy cannot be saved by DPM. At the same time, the trade-off between power consumption and performance also needs to be considered carefully [3]. In several DPM implementations, OSs play important roles in predicting the future idle periods of system components and switching the states of the components among running, idle and sleeping so that the total energy consumption of the components is minimized under a specific performance constraint.

To obtain further energy savings on a processor or any other LSI circuits by DPM, a control method called dynamic voltage and frequency scaling (DVFS in short) has been introduced in [4]. DVFS tunes the operating voltage and thus corresponding operating frequency dynamically so that the energy consumption of the processor is minimized under a frequency constraint. This method is based on a characteristic of CMOS circuits where the energy consumption is quadratically proportional to the operating voltage while the delay is approximately proportional to the operating voltage if threshold voltage is fixed. Once the frequency constraint of a processor is set, the energy consumption of the processor is minimized when the maximum possible frequency of the processor is just met the frequency constraint. At a system level, it typically exploits task schedulers within OSs that assign operating voltage and frequency to each task in addition to CPU time which has been handled by conventional real-time OSs. In this case, the scheduler implements a policy that sets the operating voltage and the clock frequency of the processor so that the energy consumption of the processor is minimized under the real-time constraints given to all tasks.

2.2 Minimum Energy Point Tracking

A method for tracking the minimum energy point (MEP) is presented in [5]. It tunes not only operating voltage and frequency but also threshold voltage of transistors for minimizing the energy consumption of CMOS circuits using in-situ power monitor. The threshold voltage is controlled by tuning body bias of transistors. This method is based on relative power values between two measurement iterations. The in-situ power monitor computes difference between power consumed in current and previous steps. Based on the sign of the difference (i.e., negative or positive), power monitor decides whether the operating voltage and threshold voltage should be increased or decreased. A simpler method than the method presented in [5] is proposed in [6] for quickly tracking the MEP. With a single sampling of temperature, dynamic and static power consumption values separately, the algorithm identifies whether the current operating point (i.e. the pair of operating voltage and threshold voltage) is the MEP or not. The methods largely reduce the energy consumption of the processor compared to the DVFS methods which only handle the operating voltage or frequency as a tuning knob. However, to the best of our knowledge, there is no existing work that exploits

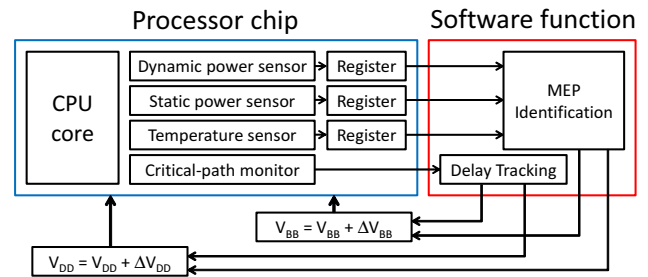


Fig. 1 Overview of MEP Tracking System

task schedulers within OSs for tracking the MEP. This paper for the first time presents a software implementation of the method as a first step towards OS-based MEP tracking. This provides flexibility and portability of the MEP tracking functionality.

3. Minimum Energy Tracking System

3.1 System Outline

In this section, we present a MEP tracking system which minimizes the power consumption of the target processor dynamically under a frequency constraint using the MEP tracking algorithm as a core logic. The MEP tracking algorithm is based on the algorithm proposed in [1]. The core logic is implemented as a software function. It is composed of delay tracking and MEP identification which finds the optimal pair of supply voltage and body bias at runtime. The overview of the system is depicted in Fig. 1. To verify if the processor is running at a specified frequency, a critical path monitor which is typically based on a critical path replica (CPR) is introduced. Since the CPR is designed to represent the current critical-path delay of the processor, we can see the critical-path delay of the processor by monitoring the CPR delay. At the same time, in order to judge if the processor is running at MEP, it is necessary to estimate temperature, dynamic power and static power consumption values of the processor separately at runtime. The details of the delay tracking function and the MEP identification function are presented in subsection 3.3. Typically the voltage setting of the processor is updated periodically towards MEP. Since the processor and the sensors are running concurrently, the sensor values required for the MEP identification are stored in corresponding registers and ready to be collected at the end of every period. Once the sensor values are obtained, the MEP identification function makes the decision of voltage tuning (i.e. stepping up, down or staying at the current voltage).

3.2 Required Hardware Supports

In addition to the hardware supports required for the DVFS systems, the MEPT systems require temperature sensor, dynamic and static power sensors. Notice that the MEPT requires sensing the dynamic and static power values separately. This function can be incorporated by dedicated online sensors for measuring dynamic and static power consumption separately. This also can be implemented by a total power sensor and a static power sensor. The dynamic power value can be obtained by subtracting the static power value from the total power consumption. A power efficient temperature sensor can be implemented based on [7]. The

Algorithm 1 MEP Tracking

```

while (Current Operating Point != MEP) do
    Delay Tracking
    MEP Identification
end while

```

sensor values should be easily accessible from the software program running on the processor. Therefore, it is preferable that the sensor values are stored in dedicated registers as shown in Fig. 1.

Another important component for the MEP tracking system is a body bias generator. Although DC-DC converters and PLLs for dynamically changing the operating voltage and frequency for DVFS systems are intensively studied over the last several decades, body bias generators for dynamically tuning the body bias of transistors are not sufficiently studied. An energy and area efficient body bias generator is proposed in [8]. In principle, our MEPT system does not limit the type of body bias generator. One could select any type of body bias generator providing wide range of body bias from reverse bias to forward bias in an energy and area efficient way that covers MEPs under wide range of performance constraint.

3.3 Software Implementation

We assume that the MEP Tracking function shown in **Algorithm 1** is invoked periodically by timer interruption or external interruption for example. Once the MEP tracking function is invoked, the first task is to find an operating point (i.e. a pair of operating voltage (V_{DD} in the following) and body bias (V_{BB} in the following)) which satisfies a given frequency constraint. This is done by the Delay Tracking function shown in **Algorithm 2**. Although the frequency constraint is satisfied by using the operating point found by the Delay Tracking function, the energy consumption of the processor is not necessarily minimized under the frequency constraint. Therefore, the next task of the MEP Tracking function is to update the operating point towards the MEP under the frequency constraint. This is done by the MEP Identification function shown in **Algorithm 3**. The mode is a variable to specify a direction of stepping voltage in the Delay Tracking function. When V_{DD} had been changed in

Algorithm 2 Delay Tracking

```

Require: Current Delay ( $D$ ), Delay Constraint ( $D_s$ ), Supply Voltage ( $V_{DD}$ ),
Body Bias ( $V_{BB}$ ), Margin ( $\Delta D$ ), mode
Ensure: Supply Voltage ( $V_{DD}$ ), Body Bias ( $V_{BB}$ )
Measuring Current Delay
while ( $|D - D_s| > \Delta D$ ) do
    if ( $D < D_s$ ) then
        if (mode = 1) then
             $V_{BB}$  go one step down
        else
             $V_{DD}$  go one step down
        end if
    else
         $V_{DD}$  go one step up
    end if
end while
Go to MEP Identification

```

Algorithm 3 MEP Identification

```

Require: Dynamic Power ( $P_d$ ), Static Power( $P_s$ ), Supply Voltage ( $V_{DD}$ ),
Body Bias ( $V_{BB}$ ), margin of slope ( $\Delta S$ )
Ensure: Supply Voltage ( $V_{DD}$ ), Body Bias ( $V_{BB}$ ), mode
Calculate slope of Energy Contour ( $S_e$ ) and Frequency Contour ( $S_f$ )
if ( $|S_e - S_f| > \Delta S$ ) then
    if ( $S_e > S_f$ ) then
         $V_{DD}$  go one step up
        mode = 1
        Go to Delay Tracking
    else
         $V_{BB}$  go one step up
        mode = 0
        Go to Delay Tracking
    end if
else
    END
end if

```

the MEP Identification function, we change V_{BB} prior to V_{DD} in the Delay Tracking function. On the other hand, we change V_{DD} prior to V_{BB} in the Delay Tracking function when V_{BB} had been changed in the MEP Identification function. If we do not have the variable mode, V_{DD} may oscillate in such a way that V_{DD} steps up in the MEP Identification function and then steps down in the Delay Tracking function. These two tasks are repeatedly invoked until the MEP is found under the frequency constraint.

4. Case Study Using a 32-bit RISC Processor with On-Chip Sensors

As a case study, the proposed MEP tracking system consisting of an FPGA and a host PC is verified through silicon measurement of a 32-bit RISC processor with on-chip sensors.

4.1 Implementation Example of MEP Tracking System

4.1.1 Target Circuit

A target circuit is a 32-bit, 5-stage pipelined RISC processor fabricated in a 65-nm SOTB process technology. Its photograph is shown in Fig. 2. The processor employs 4-kB I-Cache, 8-kB I-SPM (Scratch Pad Memory), and 16-kB D-SPM. A Voltage-Controlled Oscillator (VCO) implemented to the chip is utilized to generate a clock signal. As described in section 3, on-chip sensors of dynamic/static power dissipation, chip temperature, and critical path delay play an important role in the proposed MEP tracking system. The target circuit employs fully digital on-chip

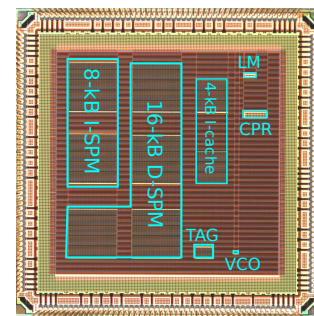


Fig. 2 Photograph of the target processor. LM: Leakage Monitor. CPR: Critical Path Replica.

sensors proposed in [9], [7], [10]. An overview of the on-chip sensors is briefly presented in section 4.1.2.

4.1.2 On-Chip Sensors

Dynamic power sensor: Reference [9] proposed a modeling method of dynamic power dissipated in embedded processors. It pointed out that dynamic power dissipated in embedded processors does not considerably fluctuate even if they execute different instructions. It also pointed out that access frequency of on-chip memories is a key parameter to accurately model the dynamic power consumption of the processor. The facts imply that we can accurately estimate the dynamic power consumption by just monitoring the number of executed instructions and access frequency of on-chip memories. Based on the fact, [6] presented the dynamic-power estimation method by counting the number of the following hardware events in the embedded processor: (i) instruction execution, (ii) cache access, and (iii) SPM access. Its approach is summarized as follows:

$$P_d = (A \times [\text{The number of instructions executed per second}] + B \times [\text{The number of cache accesses per second}] + C \times [\text{The number of I-SPM accesses per second}] + D \times [\text{The number of D-SPM accesses per second}] + E \times f)V_{DD}^2, \quad (1)$$

where V_{DD} and f are a supply voltage and a clock frequency of the target processor. The parameters A , B , C , D and E are fitting parameters determined by process technologies and processor architectures. In this paper, the fitting parameters are derived through several training programs. Exploiting the dedicated performance counters of the four hardware events, this paper estimates the dynamic power consumption of the target processor.

Static power and temperature sensor: Leakage-driven inverter cells proposed in [7] are utilized to estimate the static power consumption of the target processor. The overview of the leakage-driven ring oscillator is shown in Fig. 3. According to [7], the oscillation frequency of the ring oscillator (f_{leak}) is proportional to the subthreshold leakage current of the transistor “C0” if the transistors “C0” and “C1” in the cells are OFF and ON, respectively. Based on the fact, [6] estimated the static power of the embedded processor (P_s) by monitoring f_{leak} . The key equation is summarized as follows:

$$I_{leak} = k_s f_{leak} V_{DD}, \quad (2)$$

$$P_s = I_{leak} V_{DD} = k_s f_{leak} V_{DD}^2, \quad (3)$$

where I_{leak} and k_s are the leakage current introduced by the target processor and a fitting parameter, respectively. (2) and (3) indicate that we can estimate the static power consumption by just measuring f_{leak} if we derive the exact value of k_s in advance. A dedicated frequency counter for the leakage-driven ring oscillator is implemented into the proposed MEP tracking system so that the MEP tracking software can obtain the oscillation frequency value.

Reference [7] also pointed out that chip temperature can be estimated by monitoring the oscillation frequency of the leakage-driven ring oscillator. Since I_{leak} in (2) is exponentially proportional to chip temperature, (2) can be converted into the following simple equation [6]:

$$\ln(f_{leak}) = a_T \cdot \frac{1}{T} + b_T, \quad (4)$$

where a_T and b_T are fitting parameters. (4) indicate that chip temperature can be estimated by measuring f_{leak} if we know the exact values of a_T and b_T in advance.

Critical Path Replica: A critical path replica proposed in [10] is implemented into the target circuit. Reference [10] proposed a simple method to synthesize a path so that its propagation delay is close to the critical path delay of the target processor. The technique proposed in [Park2011] enables to design a path whose propagation delay is always slower than the critical path delay. The target processor employs a ring oscillator where the input/output signals of the critical path replica are mutually connected. Since its oscillation frequency is slower than the maximum operating speed of the target processor, we can guarantee that the processor operates without timing violations (i) by monitoring the its oscillation frequency, and (ii) by checking whether or not the frequency is slower than the target frequency. In this paper, the critical path monitor circuit and a frequency counter for it are implemented into the proposed MEP tracking system.

4.1.3 Implementation Example of the Entire System

The proposed MEP tracking system is implemented utilizing a commercial Field-Programmable Gate Array (FPGA), a host computer, and voltage regulators. The overview of the implementation example is shown in Fig. 4. The on-chip sensors described in section 4.1.2 are implemented into the target processor. “HW event detectors” in Fig. 4 flip their output signals when their corresponding hardware events are observed. Since the target processor and an FPGA are mutually interconnected, the FPGA can measure the number of hardware events per second, and oscillation frequencies of the leakage-driven ring oscillator and the critical path replica. The host computer can obtain these counters’ output signals through an Universal Asynchronous Receiver/Transmitter (UART) module. The MEP tracking algorithm proposed in section 3 is performed by the host PC. Based on the MEP tracking algorithm, the host PC tunes V_{DD} and V_{BB} of the processor through commercial voltage regulators.

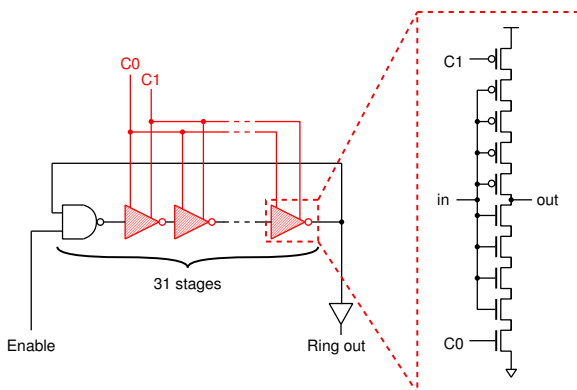


Fig. 3 Ring oscillator consisting of leakage-driven inverter cells [7].

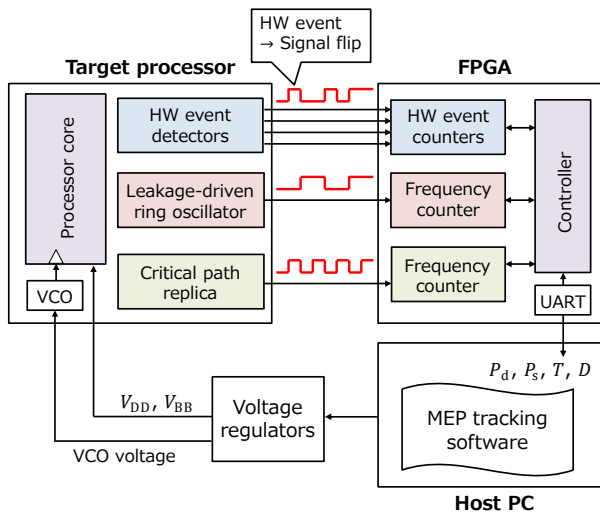


Fig. 4 An implementation example of the proposed MEP tracking system.

Table 1 Average hardware events performed per second for a 63 MHz clock frequency. V_{DD} and V_{BB} are 0.75 V and -0.5 V. Pr.: Program.

Pr.	P_d	Inst. execution	Cache	I-SPM	D-SPM
1	3.99 mW	12.8 MHz	10.0 MHz	1.68 MHz	1.59 MHz
2	1.89 mW	3.02 MHz	0	0	0
3	1.92 mW	3.11 MHz	0	0.125 MHz	0.113 MHz
4	2.90 mW	7.14 MHz	4.56 MHz	6.97 MHz	0
5	3.61 mW	12.7 MHz	476 Hz	87.2 MHz	3.12 MHz

Table 2 Average hardware events performed per second for a 3 MHz clock frequency. V_{DD} and V_{BB} are 0.75 V and -0.5 V. Pr.: Program.

Pr.	P_d	Inst. execution	Cache	I-SPM	D-SPM
1	0.47 mW	574 kHz	467 kHz	9.78 kHz	85.3 kHz
2	0.13 mW	34.8 kHz	0	0	0
3	0.21 mW	40.5 kHz	0	735 Hz	5.95 kHz
4	0.24 mW	263 kHz	214 kHz	4.56 kHz	0
5	0.40 mW	573 kHz	22.0 Hz	473 kHz	85.1 kHz

4.2 Experimental Results

4.2.1 Measurement Setup for Dynamic/Static Power Sensors

Dynamic power estimation: In order to estimate dynamic power dissipation of the processor, the fitting parameters A , B , C , D and E in (1) need to be derived through training programs in advance. In this paper, five Discrete Cosine Transform (DCT) loop programs are utilized as training programs for the dynamic power estimation. Although the five programs perform the same program, memory configurations for the programs are different from each other. For example, the first program (Program 1 in the following) utilizes all the on-chip memories (i.e., I-cache, I-SPM and D-SPM) to execute the DCT loop program while the second program (Program 2 in the following) program utilizes no on-chip memories. The measurement results of the hardware events for a 0.75 V supply voltage and a 0.5 V reverse body bias are summarized in Tabs. 1 and 2. Different clock frequencies (63 MHz and 3 MHz) are utilized in the evaluations. Based on the measurement results, the parameter fitting for A , B , C , D and E in (1) is performed through the least-square method. The fitting result is summarized in Table 3.

Static power estimation: In the similar way to the dynamic power estimation, the parameter k_s in (3) need to be determined in advance for estimating the static power dissipated by the target

Table 3 Fitting result of the A , B , C , D and E in (1) [$\text{nW}/(\text{Hz}\cdot\text{V}^2)$].

A	B	C	D	E
-0.77	0.763	0.966	0.229	0.272

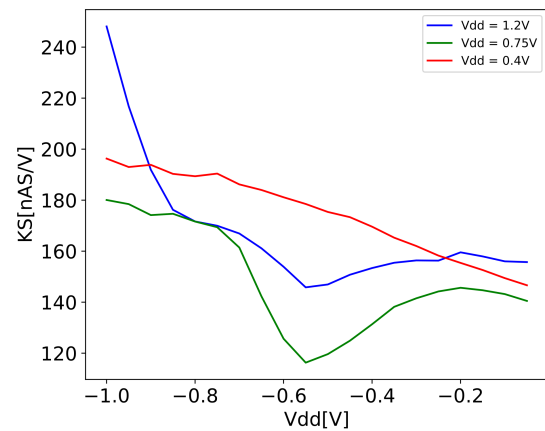


Fig. 5 Evaluation result of k_s under various voltage conditions.

processor. The parameter fitting for k_s is thus performed. Figure 5 shows the evaluation results of k_s for various voltage conditions. The k_s values are derived (i) by directly measuring P_s and f_{leak} for each voltage condition, and (ii) by calculating the k_s value based on (3). As can be seen from Fig. 5, the k_s value does not considerably vary over a wide range of voltage conditions. Therefore, this paper utilizes the k_s value for a 0.75 V supply voltage and a 0.5 reverse body bias as a representative of k_s .

4.2.2 MEP Tracking Results

In order to verify the proposed MEP tracking system, the following three performance constraints are given: (i) a 63 MHz clock frequency, (ii) a 63 MHz clock frequency, and (iii) a 80 MHz clock frequency. The target program is Program 1. In the experiments, this paper assumes that the target processor operates at room temperature and that no temperature monitoring is required for simplicity. 10 mV voltage steps are used as ΔV_{DD} and ΔV_{BB} . This paper regards the propagation delay of the critical path replica as the critical path delay of the target processor.

Figure 6 shows the MEP tracking results for a 63 MHz performance constraint. Vertical axis and horizontal axis are the supply voltage and the body bias of the target processor. Note that the threshold voltage of the target processor increases as we move leftward in Fig. 6. A solid zigzag line is the locus of the proposed MEP tracking system. “True MEP” in Fig. 6 is the actual MEP obtained through exhaustive search. Note that the propagation delay of the critical path replica is regarded as the critical path delay of the target processor in the exhaustive search. The result shows that the energy consumption of the processor can be reduced to 59.28 fJ/cycle by the proposed MEP tracking while the actual minimum energy consumption is 59.21 fJ/cycle. Therefore, the proposed MEP tracking system can minimize the energy consumption with an 0.11% estimation error. The time consumption that is required to find the MEP is 2.86 s.

Figures 7 and 8 are the MEP tracking results when the performance constraint is set as 80 MHz and 30 MHz, respectively. In

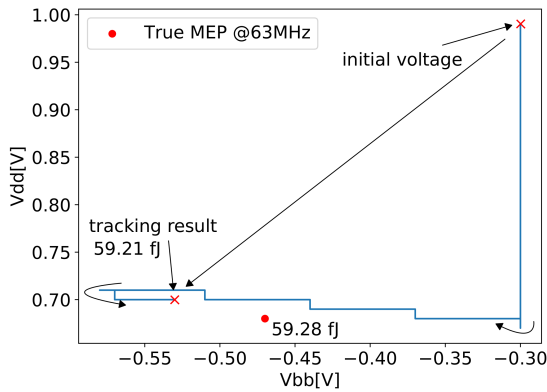


Fig. 6 MEP tracking result for a 63 MHz performance constraint.

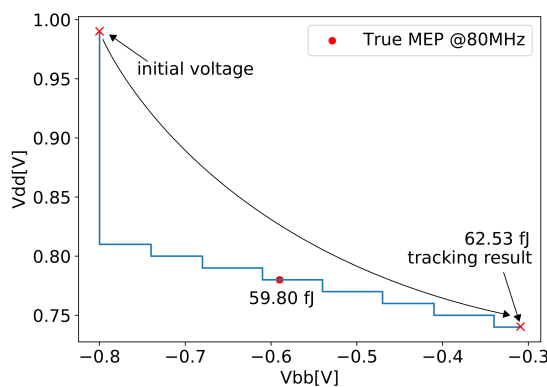


Fig. 7 MEP tracking result for a 80 MHz performance constraint.

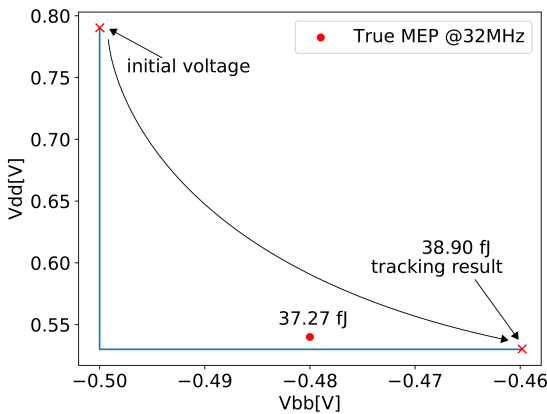


Fig. 8 MEP tracking result for a 32 MHz performance constraint.

the two scenarios, the maximum energy overhead introduced by the proposed system is 4.4% at the worst case, which implies that the proposed MEP tracking system can track the MEPs with acceptable energy overheads. The time consumptions required to find the MEPs in Figs. 7 and 8 are 1.35 s and 1.23 s, respectively. Therefore the proposed MEP tracking system can find the MEPs in the order of seconds.

5. Conclusion and Future work

In this paper, a minimum energy point (MEP) tracking system

based on a minimum energy tracking algorithm presented in [1] has been designed. The core part of the system is implemented by software programs. It can minimize the power consumption of target processor while satisfying a given performance demand at runtime. The software implementation of the tracking system has good portability and flexibility which make it simple for porting the tracking system into existing operating systems. Through a case study using a RISC processor chip, we confirmed that the MEP tracking system has a sufficient accuracy where the energy loss introduced by our MEP tracking system is 4.4% at the worst case compared with the energy consumption at the actual MEP. Our future work will be first focused on extending the current tracking system to take the impact of temperature on MEP into account. Our future work also includes porting the MEP tracking system into existing operating systems.

Acknowledgement

This work is partly supported by Grant-in-Aid for Scientific Research 17H01712. This work is supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Cadence Design Systems, Inc., Synopsys, Inc. and Mentor Graphics, Inc.

References

- [1] S. Hokimoto, T. Ishihara and H. Onodera, "Minimum Energy Point Tracking Using Combined Dynamic Voltage Scaling and Adaptive Body Biasing," 29th IEEE International System-on-Chip Conference (SOCC), Seattle, WA, 2016, pp. 1-6.
- [2] L. Benini, A. Bogliolo and G. De Micheli, "A Survey of Design Techniques for System-Level Dynamic Power Management," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 8, no. 3, pp. 299-316, June 2000.
- [3] L. Benini and G. de Micheli, "System-Level Power Optimization: Techniques and Tools," in ACM Transactions on Design Automation of Electronic Systems (TODAES), vol. 5, no. 2, pp. 115-192, April 2000.
- [4] M. Weiser, B. Welch, A. Demers and S. Shenker, "Scheduling for Reduced CPU Energy," in Proceedings of the 1st USENIX Conference on Operating Systems Design and Implementation, pp. 13-23, November 1994.
- [5] N. Mehta and B. Amrutur, "Dynamic Supply and Threshold Voltage Scaling for CMOS Digital Circuits Using In-Situ Power Monitor," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 5, pp. 892-901, May 2012.
- [6] S. Hokimoto, J. Shiomi, T. Ishihara and H. Onodera, "All-Digital On-Chip Heterogeneous Sensors for Tracking the Minimum Energy Point of Processors," IEEE International Conference on Microelectronic Test Structures (ICMTS), Austin, TX, 2018, pp. 128-133.
- [7] A. K. M. M. Islam, J. Shiomi, T. Ishihara and H. Onodera, "Wide-Supply-Range All-Digital Leakage Variation Sensor for On-Chip Process and Temperature Monitoring," in IEEE Journal of Solid-State Circuits, vol. 50, no. 11, pp. 2475-2490, Nov. 2015.
- [8] N. Kamae, A. Tsuchiya and H. Onodera, "A Forward/Reverse Body Bias Generator with Wide Supply-Range down to Threshold Voltage," in IEICE, vol. E98-C, no. 6, pp. 504-511, June 2015.
- [9] A. Sinha, N. Ickes and A. P. Chandrakasan, "Instruction Level and Operating System Profiling for Energy Exposed Software," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 11, no. 6, pp. 1044-1057, Dec. 2003.
- [10] J. Park and J. A. Abraham, "A Fast, Accurate and Simple Critical Path Monitor for Improving Energy-Delay Product in DVS Systems," IEEE/ACM International Symposium on Low Power Electronics and Design, Fukuoka, 2011, pp. 391-396.