PAPER *Special Issue on High-Performance and Low-Power Microprocessors*

A Controller LSI for Realizing V_{DD}-Hopping Scheme with Off-the-Shelf Processors and Its Application to MPEG4 System

Hiroshi KAWAGUCHI[†a)], **Gang ZHANG**[†], **Seongsoo LEE**[††], **Youngsoo SHIN**[†], ***Nonmembers,***
and **Takayasu SAKURAI**[†], ***Regular Member***

SUMMARY An LSI has been fabricated and measured to demonstrate feasibility of V_{DD}-hopping scheme in an embedded system level by executing MPEG4 CODEC. In the V_{DD}-hopping, supply voltage of a processor is dynamically controlled by a hardware-software cooperative mechanism depending on workload of the processor. When the workload is about a half, the V_{DD}-hopping is shown to reduce power to less than a quarter compared to the conventional fixed-V_{DD} scheme. The power saving is achieved without degrading real-time features of MPEG4 CODEC.

key words: *low power, real-time embedded system, dynamic voltage scaling, application slicing, MPEG4*

1. Introduction

These days, high-performance and low-power processors are demanded extensively to meet the increasing needs for portable systems like a palmtop PDA and an intelligent cellular phone. Therefore, saving further power is becoming one of the most important issues. To solve this issue, there have been several proposals to reduce the system power by dynamically providing optimum fine-grained supply voltage (V_{DD}) and clock frequency (f) to a processor [1]–[7]. Redesign of a processor, however, is required to implement these proposals because V_{DD} and f are controlled by a model of a critical path in the processor using hardware feedback. Then, it is difficult to make use of off-the-shelf processors sold on the market, which produces a big barrier to use the concept of the dynamic voltage scaling (DVS).

On the other hand, there are processors already sold on the market, which implement a DVS scheme on a chip [8], [9]. The system, however, performs the voltage scaling in coarse time resolution and cannot reduce power by making use of the data-dependent nature of the multimedia applications or guaranteeing the real-time feature, which is different from the scheme proposed in this paper. Thus, the system works fine in PC environments, but is not suitable for embedded processor environments.

This paper presents a novel LSI that externally provides V_{DD} and f, and a system to realize the DVS that can utilize off-the-shelf processors with the concept of the run-time voltage hopping [10], [11]. This novel DVS system is called V_{DD}-hopping, and Fig. 1 shows the conceptual diagram of the V_{DD}-hopping. The application program calculates workload of a task and then, sends speed information to the external V_{DD}-hopping hardware via processor, or the processor gets into a sleep mode if there is no task to execute. By using the speed information, the V_{DD}-hopping hardware provides V_{DD} and f to the processor. Thus, the V_{DD}-hopping utilizes dynamic adjustment of V_{DD} and f depending on the workload of the processor. When the workload is decreased, the power would be drastically reduced by decreasing f and V_{DD} because the power is proportional to the square of V_{DD}. By limiting the number of discrete voltage levels to two, and providing V_{DD} and f externally, the V_{DD}-hopping make it possible to use off-the-shelf processors. This means, in the V_{DD}-hopping, V_{DD} hops between only the two levels using software feedback. The reduction of number of levels is crucial in a product because many test sequences should be run, if the number is large. This V_{DD}-hopping is applied for the first time to an MPEG4 CODEC system without degrading real-time feature of the system.

2. V_{DD}-Hopping

Figure 2 shows three approaches to reduce power when

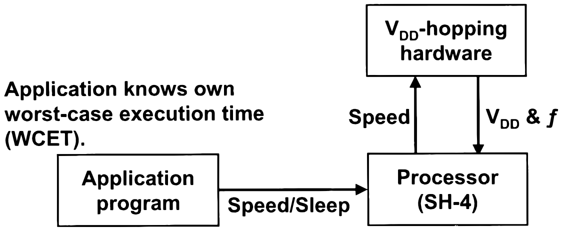

V_{DD}-hopping hardware
Application knows own worst-case execution time (WCET).
Speed
V_{DD} & f
Application program
Speed/Sleep
Processor (SH-4)

Fig. 1 Conceptual diagram of V_{DD}-hopping.

Manuscript received June 21, 2001.
Manuscript revised October 1, 2001.

[†] The authors are with the Center for Collaborative Research, the University of Tokyo, Tokyo, 153-8505 Japan.

[††] The author is with the Department of Information Electronics, Ewha University, 11-1 Daehyun-dong, Seodaemun-gu, Seoul, Korea.

a) E-mail: kawapy@iis.u-tokyo.ac.jp

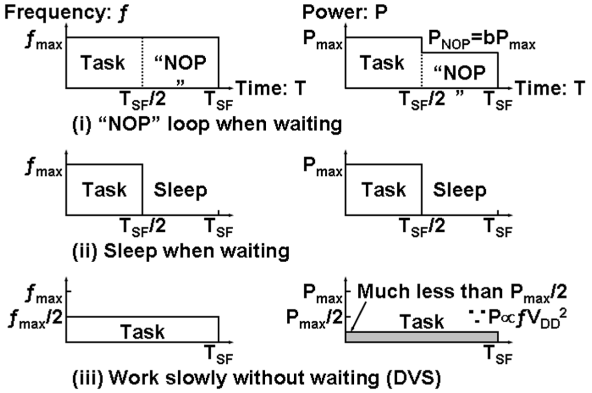


Fig. 2 Three approaches to reduce power when the workload is 50%. f_{max} is the maximum f when V_{DD} is at maximum, V_{DDmax} and then, the maximum power (P_{max}) is consumed. V_{DD} is fixed to V_{DDmax} in (i) and (ii), and only the task period is controlled. On the other hand, in (iii), f and V_{DD} are controlled dynamically. It is assumed that no power is consumed while sleep for simplicity.

the workload is 50%. The approach (i) and (ii) are the conventional approach while (iii) is the DVS, which shows the highest power saving. The point is to execute a task as slowly as possible.

(i) "NOP" loop when waiting: Even if there is no task to be done, application programs usually execute "NOP" loop to wait for either a next task or an interrupt. Then, clock generators with PLL/DLL, memories including caches and address calculations are executed which consume certain level of power, bP_{max}, where b is less than one. The normalized power, NP, is expressed as follows when the normalized workload is NW.

$$NP(NW) = (1-b)NW + b.$$

(ii) Sleep when waiting: If a sleep mode is available on a target processor, an application program can use the sleep mode after a task is completed until the next task starts or the interrupt acknowledges. In this case, since usually almost no power is consumed in the sleep mode, NP is given as follows.

$$NP(NW) = NW.$$

(iii) Work slowly without waiting (DVS): This corresponds to the DVS case. NW and NP are given by parametric functions of V_{DD} with α-power law MOSFET model as follows [12].

$$NW(V_{DD}) = \frac{V_{DDmax}}{V_{DD}}\left(\frac{V_{DD}-V_{TH}}{V_{DDmax}-V_{TH}}\right)^{\alpha},$$

$$NP(V_{DD}) = \left(\frac{V_{DD}}{V_{DDmax}}\right)^{2} NW(V_{DD}).$$

$$\therefore\ NP(NW) = NW^{\frac{\alpha+1}{\alpha-1}} \quad \text{if} \quad V_{TH}=0.$$

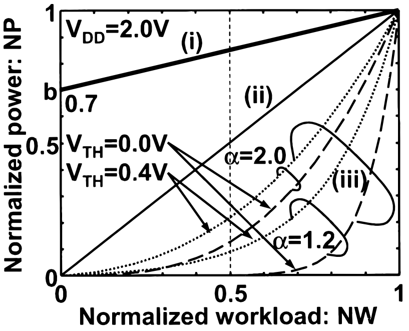


Fig. 3 NP dependence on NW. (i) "NOP" loop when waiting. b is assumed 0.7. (ii) Sleep when waiting. (iii) DVS.

V_{TH} denotes the threshold voltage of MOSFET. α represents a velocity saturation index, and is about 1.2 in a recent short-channel MOSFET while 2.0 in a long-channel one (classic Shockley model). NP dependence on NW for the three cases is shown in Fig. 3. In the DVS, V_{DD} is decreased to the level where the speed is just satisfied when NW is less than one. It is clear that the total power is effectively decreased by the DVS. Furthermore, it is seen from the figure, as MOSFET shrinks, and α decreases, the effectiveness of the DVS increases. This is because, if α is small, since the speed dependence on V_{DD} is small, V_{DD} can be decreased more. This would become advantage in the DVS.

V_{DD} and f are also controlled in the V_{DD}-hopping so that each task finishes its execution within its worst-case execution time (WCET) by software. The algorithm to adaptively use discrete levels of V_{DD} depending on the workload is of importance. Since the workload depends strongly on data, the control should be dynamic in run-time, and should not be static in a compile-time. It is too late to notice that the past task was an easy task which can be done much less than WCET, because, once the task is completed, there is no way to change V_{DD} to lower the power. On the other hand, it is impossible to predict the workload of the task to be done in the future without error. To solve this problem, the algorithm introduces an application slicing and a software feedback loop. By chopping an application into slices, executing the first slices at f_{max}, and checking the current time and the time margin to execute the next slice, the optimum f is adaptively selected by a software feedback loop. The details of the method used in this paper are summarized as follows with the help of Fig. 4 [10], [11].

(A) A task is sliced into N timeslots. Following parameters are obtained through static analysis of an application program or direct measurement [13].
- T_{SF}: Time constraint of sync frame where sync frame is the maximum time allowed for the task.
- T_{Wi}: WCET of i-th timeslot.
- T_{Ri}: WCET from $(i+1)$-th to N-th timeslot.

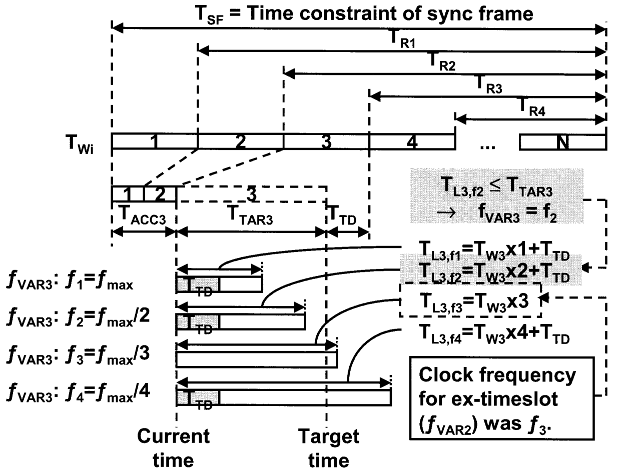
(B) For each timeslot, the target execution time,

T_{TARi}, is calculated as $T_{TARi} = T_{SF} - T_{ACCi} - T_{TD} - T_{Ri}$ where T_{ACCi} is execution time accumulated from 1st to $(i-1)$-th timeslot, and T_{TD} is a transition delay to change f and V_{DD}.

(C) For each candidate f_j, $f_j = f_{max}/j$ $(j = 1, 2, 3 \ldots)$, estimated maximum execution time, $T_{Li,fj}$, is calculated as $T_{Li,fj} = T_{wi} \times j + T_{TD}$. If f_j is equal to one of $(i-1)$-th timeslot, $T_{Li,fj} = T_{wi} \times j$. In general, candidate frequencies can have arbitrary values. The arbitrary frequency, however, causes a serious problem at interfaces with other peripheral devices [10], [11]. This is the reason why the candidate frequencies are limited to f_{max}/j where j is an integer. Then, it is possible to supply many levels of the frequencies. The power reduction is, however, minor as described later on, and test cost increases rapidly because testing at the multiple frequencies is needed to fully guarantee the normal operation at those frequencies.

(D) f of i-th timeslot, f_{VARi}, is determined as minimum f_j whose $T_{Li,fj}$ does not exceed T_{TARi}.

Thus, f and V_{DD} are dynamically controlled on a timeslot-by-timeslot basis inside each task by software. If the application finishes before WCET, the processor gets into a sleep mode. It should be noted that the proposed algorithm guarantees the real-time feature of the application. The relationship between f and V_{DD} is obtained by measuring physical characteristic, for instance, by Shmoo plot.



T_{SF} = Time constraint of sync frame
T_{R1}
T_{R2}
T_{R3}
T_{R4}
T_{Wi}
1
2
3
4
...
N
1
2
3
T_{ACC3}
T_{TAR3}
T_{TD}
$T_{L3,f2} \leq T_{TAR3}$
$\rightarrow f_{VAR3} = f_2$
$T_{L3,f1}=T_{W3}x1+T_{TD}$
$T_{L3,f2}=T_{W3}x2+T_{TD}$
$T_{L3,f3}=T_{W3}x3$
$T_{L3,f4}=T_{W3}x4+T_{TD}$
f_{VAR3}: f_1=f_{max}
f_{VAR3}: f_2=f_{max}/2
f_{VAR3}: f_3=f_{max}/3
f_{VAR3}: f_4=f_{max}/4
T_{TD}
T_{TD}
T_{TD}
Clock frequency for ex-timeslot (f_{VAR2}) was f_3.
Current time
Target time

Fig. 4 Method to determine f and V_{DD} in V_{DD}-hopping.

Figure 5 shows transient curves of power, f and V_{DD} with the V_{DD}-hopping for one sync frame obtained by a simulation for an MPEG4 SP@L1 CODEC, which is a typical real-time application for portable computing. The workload is 42% of the worst case. If infinite f levels, hence, infinite V_{DD} levels are provided, maximum power reduction is possible, but the power improvement is just 8% compared to two-level V_{DD}-hopping. That is the reason why the levels of f and V_{DD} are limited to two in the V_{DD}-hopping. Eventually, in this paper, f_{max} and $f_{max}/2$ are chosen as the two-level frequencies. This choice is reasonable as described in the Appendix.

In case of the two-level V_{DD}-hopping in Fig. 5, f_{max} is used only 6% of the time while the processor run at $f_{max}/2$ for 70% of the time. For the rest of the time, the processor is in the sleep mode. f_{max} is still needed because the processor will run at f_{max} for 100% of the time, when worst-case data comes which is very unlikely, and for most of the time, the workload is about a half on average. This tendency holds for other multimedia applications such as MPEG2 decoding and VSELP (voice CODEC) that are also simulated, and about an order of magnitude improvement in the power are assured. The V_{DD}-hopping can be applied to such applications which synchronize with regular period, and whose WCET is known.

3. Breadboard Design

An MPEG4 CODEC system is built to demonstrate the feasibility of the V_{DD}-hopping as shown in Fig. 6. The system makes use of off-the-shelf processor, Hitachi's SH-4, and its embedded system board [14], [15]. A block diagram of the V_{DD}-hopping system is shown in Fig. 7. H.263 standard image sequence "carphone" is used as input data. The image has 80×64 pixels (5×4 macroblocks), and is stored in the flash ROM as raw

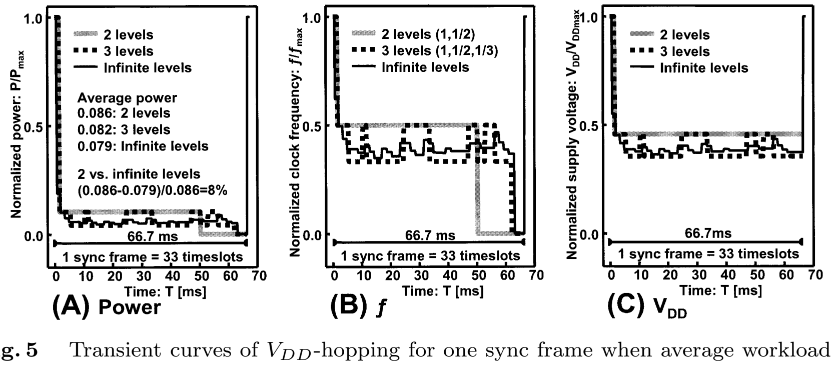


Fig. 5 Transient curves of V_{DD}-hopping for one sync frame when average workload is 42%. (A) Power. (B) f. (C) V_{DD}.

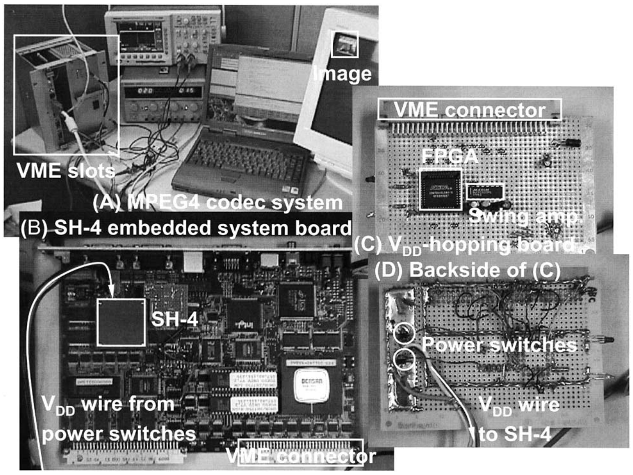


Fig. 6 (A) MPEG4 CODEC system with V_{DD}-hopping. (B) SH-4 embedded system board made by Densan. (C) V_{DD}-hopping board inserted in VME slot. (D) Backside of (C).

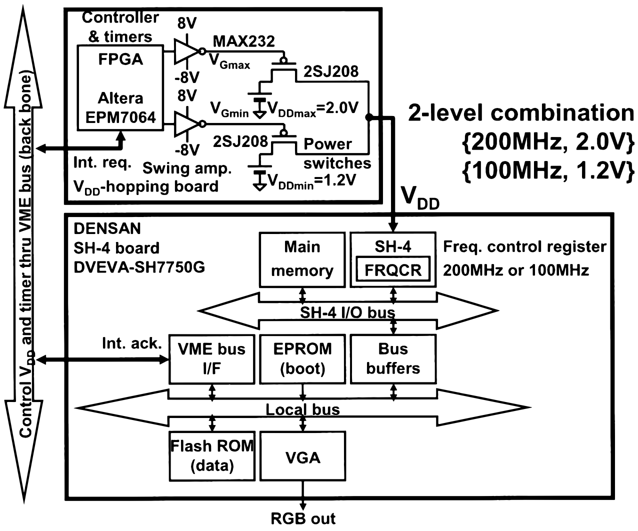


Fig. 7 Block diagram of V_{DD}-hopping system.

data. One macroblock corresponds to one timeslot discussed in the previous section. In addition, other two timeslots are assigned to initial and display routine. That is, the MPEG4 CODEC has 22 timeslots. In order to obtain WCET of the sync frames, the frame rate is varied to check that the system works in time without sync frame dropping and then, 200 ms is obtained as WCET, which means that the sync frame rate of the system is five per second. It should be noted that the image size and the sync frame rate are different from the standard. Nevertheless, feasibility of the V_{DD}-hopping can be verified in respect of both hardware and software.

In Fig. 7, the optimum f and V_{DD} are calculated by the SH-4 with the V_{DD}-hopping algorithm. Then, the speed information is sent through I/O bus of the processor and to VME bus, which controls V_{DD}-hopping board implemented by an FPGA (Altera EPM7064). Because only I/O instructions are required to implement the V_{DD}-hopping, no new instruction set is needed. This makes it possible to implement the V_{DD}-hopping system without redesigning the processor itself. The FPGA has timers in it. One of the timers watches the current time, and another timer is used to keep the processor in the sleep mode during the V_{DD} transition to avoid malfunction. In order to handle the external interrupt, the FPGA requests an interrupt (Int. req.) and the processor acknowledges the interrupt (Int. ack.) through the VME bus and a VME bus I/F chip.

There are a couple of points that should be handled with care in implementing the board level V_{DD}-hopping. The following subsections will describe these points.

3.1 Clock Frequency

The processor has a clock frequency control register (FRQCR) as shown in Fig. 7. The FRQCR can change the internal clock frequency instantaneously. The internal clock frequency is synchronized with external clock of 33 MHz. Since only 200 MHz and 100 MHz whose ratio is an integer are used as the clock frequencies, there is eventually no synchronization problem at the interface of the processor with the external systems. In a general processor, the clock frequency control register such as the FRQCR might not be implemented, and at that time, two kinds of frequencies should be applied externally. An LSI implementation described afterwards output the frequencies by itself.

Incidentally, according to the specification of the processor, 2.0 V is used as V_{DDmax} at 200 MHz. As V_{DDmin} at 100 MHz, 1.2 V is obtained by measurement.

3.2 Power Switch

On the V_{DD}-hopping board, V_{DD} hops between V_{DDmax} and V_{DDmin} using power switch MOSFETs (2SJ208 × 2), which has one of the lowest threshold voltages on the market. The threshold voltage is, however, 2.8 V that is higher than V_{DDmax}. Then, the MOSFET never turns on as it is. In Fig. 7, RS-232C driver (MAX232) is used as a voltage swing amplifier that amplifies the gate voltage to ±8 V.

Figures 8 and 9 are measured V_{DD} waveforms. The measured fall and rise time of V_{DD} are less than 200 μs and 100 μs respectively with decoupling capacitance $C_D + C_S$ of 30 μF at the node V_{DD}.

A care should be taken for the overlap of the V_{DDmax} enable signal (V_{Gmax}) and of the V_{DDmin} enable signal (V_{Gmin}). In the V_{DD} switching between V_{DDmax} and V_{DDmin}, there are two cases: One is that there is overlap between the signals, and the other is that there is no overlap between the signals. It is even-

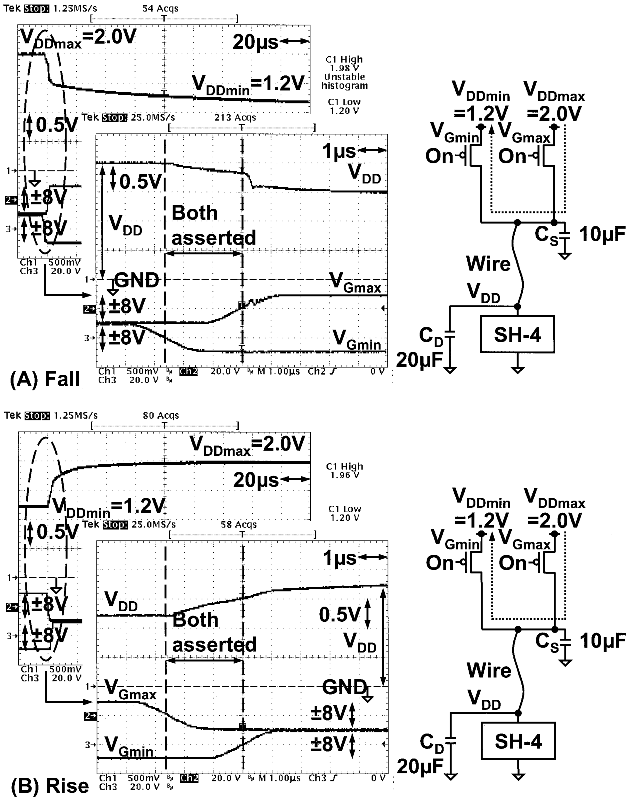


Fig. 8 V_{DD} waveforms when there is a period while both V_{Gmax} and V_{Gmin} are asserted. (A) Falling V_{DD} from V_{DDmax} to V_{DDmin}. (B) Rising V_{DD} from V_{DDmin} to V_{DDmax}.

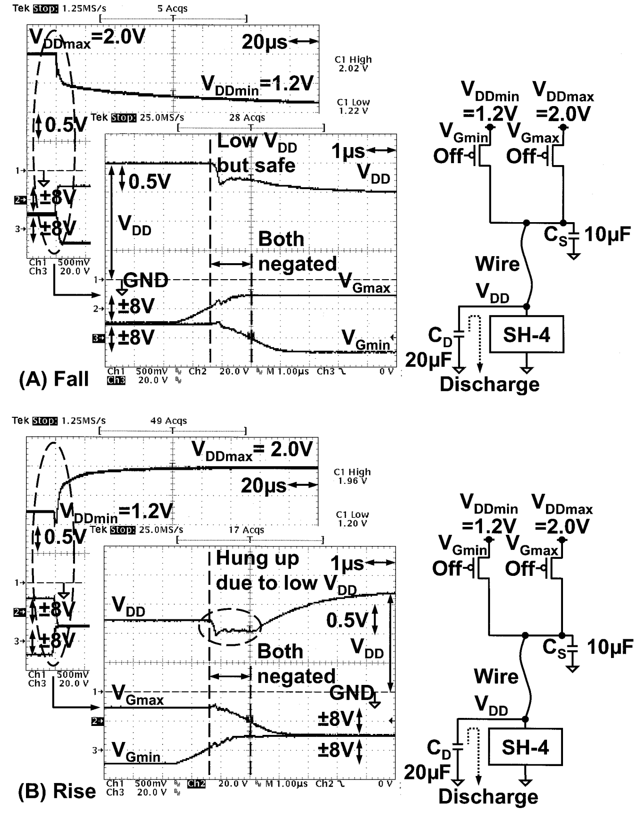


Fig. 9 V_{DD} waveforms when there is a period while both V_{Gmax} and V_{Gmin} are negated. (A) Falling V_{DD} from V_{DDmax} to V_{DDmin}. (B) Rising V_{DD} from V_{DDmin} to V_{DDmax}.

tually impossible to turn on one MOSFET and turn off the other MOSFET at the same time. If there is an overlap whose situation is depicted in Fig. 8, large current might flow from V_{DDmax} to V_{DDmin}, which might cause a problem. However, thanks to the decoupling capacitors, spike noise or voltage drop is not observed. The overlap between the signals is set to 2 μs.

If there is no overlap, there is a period while V_{DD} line is completely cut off from both V_{DDmax} and V_{DDmin}, which causes a serious problem as seen in Fig. 9. In rising V_{DD} case, V_{DD} sags below V_{DDmin} due to discharge from the decoupling capacitor, which might put the system in hung-up status. In conclusion, the switching between V_{DDmax} and V_{DDmin} should be carried out with a period while both V_{DDmax} and V_{DDmin} are connected to the V_{DD} line for a short time.

One more care other than the timing overlap is for a power-on sequence. V_{Gmax} should be asserted to connect V_{DDmax} to V_{DD} line at boot-up process to stably initiate the system.

3.3 Power

Figure 10(A) shows the measured power characteristics of the V_{DD}-hopping system. The power of the processor at 200 MHz is 0.8 W while the power at 100 MHz is 0.16 W. The power in the sleep mode is 0.07 W. Since the average time for V_{DDmax} is 8%, that for V_{DDmin} is 86%, and that for the sleep mode is 6%, the average power is 0.21 W. In the processor, I/O buffers are not optimized for the low-voltage operation. If the I/O buffers were designed carefully, V_{DDmin} could be below 0.9 V instead of 1.2 V. In that case, the power at 100 MHz could be reduced to about a half.

Based on Fig. 10(A), power dependence on workload can be obtained as shown in Fig. 10(B). 0.8 W at 200 MHz corresponds to full workload while 0.16 W at 100 MHz corresponds to a half workload. The processor consumes 0.07 W in the sleep mode when the workload is zero. Compared to (i) "NOP" loop when waiting case in Fig. 10(C), the V_{DD}-hopping works more effectively in low-workload region. On the other hand, against (ii) sleep when waiting case, the V_{DD}-hopping is the most effective when workload is a half because the second frequency is set to $f_{max}/2$.

4. LSI Design

After evaluating the V_{DD}-hopping breadboard, a V_{DD}-hopping LSI has been designed and fabricated, which has the same function as the breadboard. Fundamentally, the FPGA portion on the breadboard is imple-

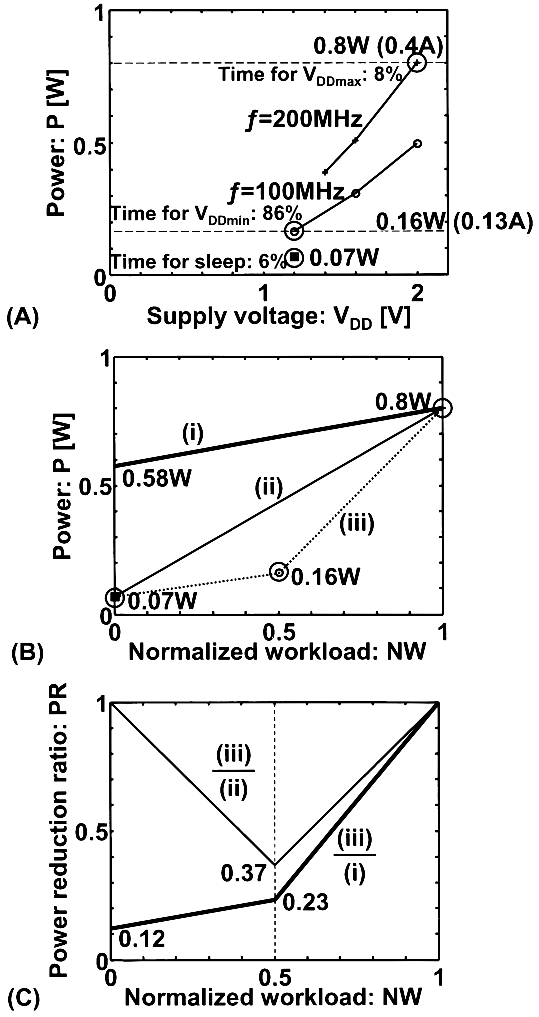


Fig. 10 (A) Measured power characteristics of V_{DD}-hopping system. (B) Power dependence on workload based on (A). (i) "NOP" loop when waiting. (ii) Sleep when waiting. (iii) V_{DD}-hopping. Processor consumes 0.58 W when executing "NOP." (C) Power reduction ratio of V_{DD}-hopping.

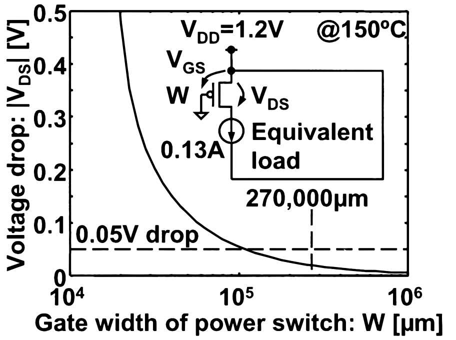


Fig. 11 Voltage drop dependence on gate width of power switch. This shows the worst case because of the minimum gate bias ($V_{GS} = -V_{DDmin} = -1.2$ V).

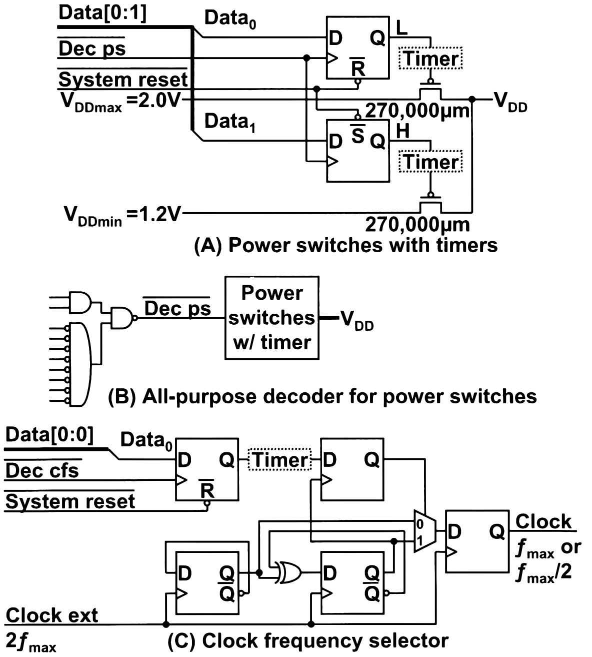


Fig. 12 (A) Power switches. Programmable timers adjust the timing overlap of V_{DDmax} and V_{DDmin}. (B) All-purpose decoder. (C) Clock frequency selector. Programmable timer avoids f changing during program execution.

mented by the LSI based on a standard cell design style.

In the V_{DD}-hopping LSI, the gate width of the power switch is also critical. The simulated V_{DS} curve is shown in Fig. 11. In the process technology used for the LSI design, the threshold voltage is 0.6 V that is smaller than V_{DDmin} (1.2 V) so that the signal swing amplifier is not needed which was required for the breadboard design. When the gate bias is 1.2 V, and load current is 0.13 A, the maximum gate width is needed. The gate width of 270,000 μm is found to be appropriate if the voltage drop by the switch is set to less than 0.05 V. This gate width can also draw large current of 0.4 A through it if V_{DD} is V_{DDmax} (2.0 V).

Figure 12 shows a schematic diagram of the V_{DD}-hopping LSI. Such as the breadboard design described in the previous section, the timing overlap between V_{Gmax} and V_{Gmin} is critical. In order to adjust the period of the overlap, programmable timers are put at the gates of the power switches. The LSI also has an all-purpose decoder for the power switches. One more care other than the timing overlap is for the boot-up process. V_{DDmax} should be connected to the V_{DD} line by $\overline{\text{System\_reset}}$ signal to stably initiate the system.

For processors that do not have a frequency control register, the V_{DD}-hopping LSI has a clock frequency selector to output either f_{max} or $f_{max}/2$. In general, a processor must be halted while f and V_{DD} is being changed to eliminate malfunctions due to the transition.

In addition, two other timers are available to watch the current time, and to wake up out of the sleep mode

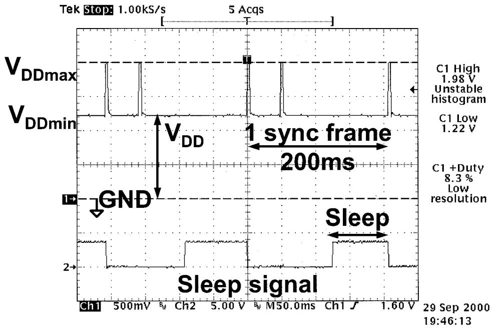


Fig. 13 Measured waveforms of V_{DD} and sleep signal of processor.

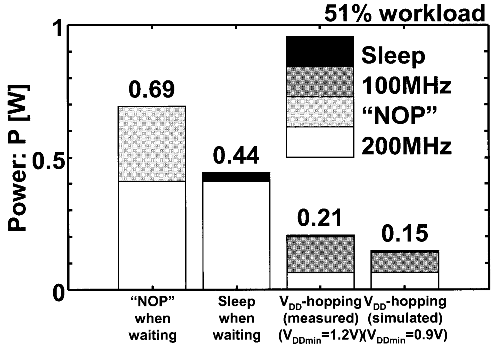


Fig. 14 Power comparison between V_{DD}-hopping and fixed-V_{DD} schemes. b is 0.72 at SH-4.

using the interrupt signal after the f and V_{DD} transition.

Figure 13 shows the measured waveforms of V_{DD} and the sleep signal of the processor. The input image data is the same as that on the breadboard. It should be noted that just two sync frames are shown in the figure. V_{DDmax} is used only 8% on average while the sleep period is 6% on average. This means that 86% left is used for V_{DDmin}. Therefore, the average workload becomes 51% (8% × 1 + 86% × 0.5 + 6% × 0).

Figure 14 shows a power comparison between the V_{DD}-hopping and other fixed-V_{DD} schemes for MPEG4 CODEC. The V_{DD}-hopping is measured to consume 0.21 W. If the I/O buffers of the processor were designed carefully, the power would become 0.15 W. In that case, the V_{DD}-hopping can reduce the power to less than a quarter of the fixed-V_{DD} scheme.

The LSI has been fabricated with Rohm 0.6 μm triple metal CMOS technology as shown in Fig. 15. The LSI consumes 0.01 W when the external clock is 33 MHz. The size is about 4.6 mm × 2.3 mm including two power switches of 270,000 μm width. The switches are implemented with comb-shaped pMOSs because of their huge width.

5. Conclusion

Feasibility of the V_{DD}-hopping has been verified based on the breadboard-level prototype, and it has been extended toward the design of the ASIC. From the software point of view, the V_{DD}-hopping exploits the application slicing while all in the ASIC are the power switches, plain login, and timers as the hardware. By applying the V_{DD}-hopping to MPEG4 CODEC, 75% power saving of the processor can be achieved without redesign of the processor. The power saving is achieved without degrading real-time features of MPEG4 CODEC.

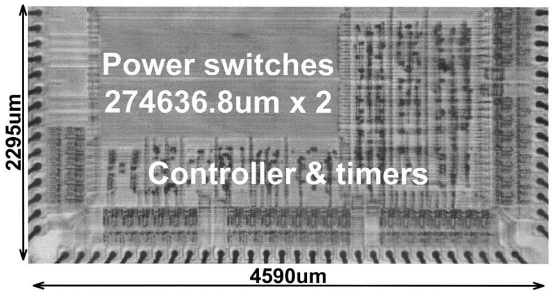


Fig. 15 LSI for V_{DD}-hopping.

Acknowledgment

The authors would like to thank Dr. K. Sasaki, Dr. K. Ishibashi of Hitachi for fruitful discussion and Mr. T. Iizuka of Densan for helpful advice. This work was supported by Grants from Hitachi and the Japan Society for the Promotion of Science. The LSI in this work was fabricated by the chip fabrication program of VLSI Design and Education Center (VDEC), the University of Tokyo with the collaboration by Rohm and Toppan Printing.

References

[1] A. Chandrakasan, V. Gutnik, and T. Xanthopoulos, "Data driven signal processing: An approach for energy efficient computing," Proc. International Symposium on Low Power Electronics and Design, pp.347–352, 1996.

[2] T. Kuroda, K. Suzuki, S. Mita, T. Fujita, F. Yamane, F. Sano, A. Chiba, Y. Watanabe, K. Matsuda, T. Maeda, T. Sakurai, and T. Furuyama, "Variable supply-voltage scheme for low-power high-speed CMOS digital design," IEEE J. Solid-State Circuits, vol.33, no.3, pp.454–462, 1998.

[3] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, and M.B. Srivastava, "Power optimization of variable voltage core-based systems," Proc. Design Automation Conf., pp.176–181, 1998.

[4] T. Pering, T. Burd, and R. Brodersen, "The simulation and evaluation of dynamic voltage scaling algorithms," Proc. International Symposium on Low Power Electronics and Design, pp.76–81, 1998.

[5] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," Proc. International Symposium on Low Power Electronics and Design, pp.197–202, 1998.

[6] T. Burd, T. Pering, A. Stratakos, and R. Brodersen, "A

dynamic voltage scaled microprocessor system," IEEE International Solid-State Circuits Conf. Digest of Technical Papers, pp.294–295, 2000.

[7] T. Okuma, H. Yasuura, and T. Ishihara, "Software energy reduction techniques for variable-voltage processors," IEEE Design & Test of Computers, March–April 2001.

[8] Transmeta Web Site, http://www.transmeta.com/.

[9] D.R. Ditzel, "Transmeta's crusoe: A low-power x86-compatible microprocessor built with software," Proc. International Symposium on Low-Power and High-Speed Chips (Cool Chips), pp.1–30, 2000.

[10] S. Lee and T. Sakurai, "Run-time power control scheme using software feedback loop for low-power real-time applications," Proc. Asia and South Pacific Design Automation Conf., pp.381–386, 2000.

[11] S. Lee and T. Sakurai, "Run-time voltage hopping for low-power real-time systems," Proc. Design Automation Conf., pp.806–809, June 2000.

[12] T. Sakurai and A.R. Newton, "Alpha-power law MOSFET model and its applications to CMOS inverter delay and other formulas," IEEE J. Solid-State Circuits, vol.25, no.2, pp.584–594, 1990.

[13] S. Lim, Y. Bae, G. Jang, B. Rhee, S. Min, C. Park, H. Shin, K. Park, and C. Kim, "An accurate worst case timing analysis for RISC processors," Proc. IEEE Real-Time Systems Symposium, pp.97–108, 1994.

[14] The Hitachi SuperH Web Site, http://www.superh.com/.

[15] Densan Web Site, http://www.densan.com/.

Appendix

The reason why two frequencies and not more than two frequencies are the best choice has been already explained in the text. Now, in this Appendix, the question is why $f_{max}/2$ and not f_{max}/j $(j > 2)$ is preferable as the second frequency. One of the reasons for multimedia applications treated in this work is that the average workload is about a half. If average workload of an application is about 1/3, and the processor is only used for the application, the best choice of the second frequency would be $f_{max}/3$. In general, however, a processor in recent systems is used for various applications. In such environment, average workload is unknown because the workload of applications running on a processor varies randomly from zero to one. Then, it is shown in this Appendix that the average power for that kind of system is minimized when $f_{max}/2$ is used as the second frequency.

Figure A· 1 shows the power dependence on workload. Segments OBC corresponds to the power dependence when $f_{max}/2$ is used as the second frequency while OAC corresponds to the case of f_{max}/j $(j > 2)$. The area under the segments is proportional to the average power when the workload of applications running on a processor varies randomly from zero to one. To show that the average power is minimized when $f_{max}/2$ is chosen, what should be demonstrated is that the area of quadrilateral OBCD (S_{OBCD}) is smaller than that of quadrilateral OACD (S_{OACD}). S_{OACD} and S_{OBCD} are given as follows,

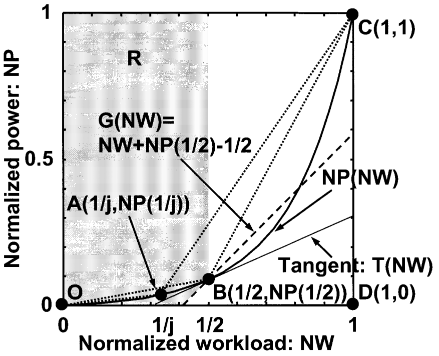


Fig. A· 1 Power comparison when f_{max}/j $(j > 2$, point A) and $f_{max}/2$ (point B) are used as second frequency.

$$S_{OACD} = NP\left(\frac{1}{j}\right) \times \frac{1}{j}/2 + \left[NP\left(\frac{1}{j}\right) + 1\right] \times \left(1 - \frac{1}{j}\right)/2, \quad \text{where} \quad j > 2,$$

$$S_{OBCD} = NP\left(\frac{1}{2}\right) \times \frac{1}{2}/2 + \left[NP\left(\frac{1}{2}\right) + 1\right] \times \left(1 - \frac{1}{2}\right)/2. \qquad (A\cdot 1)$$

$NP(1/j)$ signifies the normalized power when the workload is $1/j$. In order to show $S_{OACD} > S_{OBCD}$, the following inequality (A· 2) holds. The relationship is derived by using (A· 1).

$$NP\left(\frac{1}{j}\right) > \frac{1}{j} + NP\left(\frac{1}{2}\right) - \frac{1}{2}, \quad \text{where } j > 2. \qquad (A\cdot 2)$$

Now $1/j$ is substituted by NW, where NW is normalized workload. NW corresponds to the horizontal axis variable of Fig. A· 1. The inequality (A· 2) becomes

$$NP(NW) > NW + NP\left(\frac{1}{2}\right) - \frac{1}{2}, \quad \text{where} \quad NW < \frac{1}{2}. \qquad (A\cdot 3)$$

In the figure, a dashed line shows the function $G(NW) = NW + NP(1/2) - 1/2$, and the shaded region R corresponds to $R > NW + NP(1/2) - 1/2$. Therefore, if the curve $NP(NW)$ passes through the region R for $NW < 1/2$, the inequality (A· 3) holds and then, in turn $S_{OACD} > S_{OBCD}$ can be demonstrated.

Suppose that $T(NW)$ is the tangent line that touches $NP(NW)$ at the point B. Since $NP(NW)$ is a concave function, $(NP(NW) - T(NW))'' > 0$ for $0 \leq NW < 1/2$, and $(NP(NW) - T(NW))' = 0$ at $NW = 1/2$. Then $(NP(NW) - T(NW))' < 0$ for $0 \leq NW < 1/2$. Therefore, $NP(NW) - T(NW)$ is a decreasing function for $0 \leq NW < 1/2$, and is zero at $NW = 1/2$. This means that $NP(NW) > T(NW)$ for $0 \leq NW < 1/2$. If the slope of $T(NW)$ is less than 1, $T(NW)$ passes through the region R. In that case,

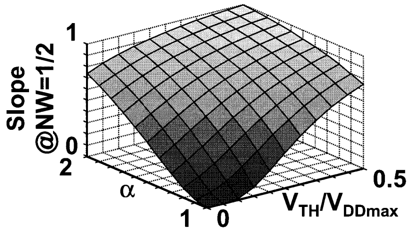


Fig. A· 2 Numerical solution of slope of $NP(NW)$ at $NW = 1/2$.

since $NP(NW) > T(NW)$, $NP(NW)$ passes through the region R, and $S_{OACD} > S_{OBCD}$ can be demonstrated. Thus, the condition for $S_{OACD} > S_{OBCD}$ now becomes the following.

$$\frac{dNP\left(\frac{1}{2}\right)}{dNW} < 1.$$

As described in Sect. 2, NP dependence on NW is shown as follows by parametric functions. It is difficult to write $NP(NW)$ in an explicit way.

$$NW(v_{dd}) = \frac{1}{v_{dd}}\left(\frac{v_{dd} - \dfrac{V_{TH}}{V_{DDmax}}}{1 - \dfrac{V_{TH}}{V_{DDmax}}}\right)^{\alpha},$$

$$NP(v_{dd}) = v_{dd}^2 NW(v_{dd}),$$

where

$$v_{dd} = \frac{V_{DD}}{V_{DDmax}}.$$

The slope at $NW = 1/2$ can be numerically calculated as follows, and the result is shown in Fig. A· 2.

$$Slope = \frac{dNP\left(\frac{1}{2}\right)}{dNW} = \left.\frac{\dfrac{dNP(v_{dd})}{dv_{dd}}}{\dfrac{dNW(v_{dd})}{dv_{dd}}}\right|_{NW(v_{dd})=\frac{1}{2}}.$$

In the region where $0 \leq V_{TH}/V_{DDmax} \leq 1$ and $1 \leq \alpha \leq 2$, which hold in normal VLSI processors, it is seen from the figure that the slope does not exceed 1. Therefore, $S_{OACD} > S_{OBCD}$ is now demonstrated and then, it is established that the average power is minimized when $f_{max}/2$ is chosen as the second frequency for the system where the average workload of applications running on a processor varies randomly from zero to one.

Hiroshi Kawaguchi received the B.S. and M.S. degrees in E.E. from Chiba University, Japan, in 1991 and 1993, respectively. He joined the Technological Research Institute, Konami Corporation in 1993, where he developed arcade entertainment systems. He moved to the University of Tokyo in 1996 as a technical associate and he is now engaged in research of high-performance and low-power system VLSI designs.

Gang Zhang received the B.S. degree in E.E. from Tokyo Institute of Technology and the M.S. degree in E.E. from the University of Tokyo, Japan, in 1998 and 2001 respectively. In 2001, he joined Tokyo Research Laboratory, IBM Japan Ltd., where he has been engaged in the research and development of cryptographic hardware and embedded systems.

Seongsoo Lee received the B.S., M.S. and Ph.D. degrees in E.E. from Seoul National University, Korea, in 1991, 1993, and 1998, respectively. From 1998 to 2000, he was a research associate in the Institute of Industrial Science, the University of Tokyo, Japan. In 2000, he joined Ewha Womans Unversity, where he is currently a research professor. His research interests include low-power VLSI systems, dynamic voltage scaling, dynamic power management, low-power multimedia signal processing, high-performance MPEG processors, motion estimation and rate control.

Youngsoo Shin received the B.S., M.S., and Ph.D. degrees in E.E. from Seoul National University, Korea, in 1994, 1996, and 2000, respectively. He subsequently joined the Center for Collaborative Research, the University of Tokyo, Japan, working as a research associate. He is currently a research staff member at IBM Thomas J. Watson Research Center. His research interests include system-level and low-power design.

Takayasu Sakurai received the B.S., M.S. and Ph.D. degrees in E.E. from the University Tokyo, Japan, in 1976, 1978 and 1981, respectively. In 1981, he joined Toshiba Corporation, where he designed CMOS DRAM, SRAM and BiCMOS ASICs. From 1988 to 1990, he was a visiting researcher at the University of California at Berkeley, doing research in the field of VLSI CAD. From 1990, back in Toshiba. From 1996, he has been a professor at the University of Tokyo, working on low-power and high-performance system LSI designs. He served as a program committee member for CICC, DAC, ICCAD, ICVC, ISPLED, ASP-DAC, TAU, CSW, VLSI and FPGA Workshop.