Low-Power Video Decoding on a Variable Voltage Processor for Mobile Multimedia Applications

Seongsoo Lee

This paper proposes a novel low-power video decoding scheme. In the encoded video bitstream, there is quite a large number of non-coded blocks. When the number of the non-coded blocks in a frame is known at the start of frame decoding, the workload of the video decoding can be estimated. Consequently, the supply voltage of very large-scale integration (VLSI) circuits can be lowered, and the power consumption can be reduced. In the proposed scheme, the encoder counts the number of non-coded blocks and stores this information in the frame header of the bitstream. Simulation results show that the proposed scheme reduces the power consumption to about 1/10 to 1/20.

Keywords: Low-power, SoC, video decoding, variable voltage, MPEG.

Manuscript received Jan. 11, 2005; revised Aug. 09, 2005.

This work was supported by the Soongsil University Research Fund. The material in this work was presented in part at IT-SoC 2004, Seoul, Korea, Oct. 2004.

Seongsoo Lee (phone: +82 2 820 0692, email: sslee@ssu.ac.kr) is with the School of Electronics Engineering, Soongsil University, Seoul, Korea..

I. Introduction

Recently, mobile multimedia applications such as mobile videophone and mobile multimedia broadcasting are becoming more and more important as information and communication technologies make great progress.

Power consumption of a mobile multimedia terminal is one of the primary design goals, since battery operating time is one of the important issues in commercial product design [1]. Conventional mobile terminals such as cellular phones operate more than several hours with a single battery, but last less than an hour in mobile multimedia terminals. Digital multimedia broadcasting (DMB) is one of the killer applications in mobile multimedia applications, but an hour is not enough to enjoy even a single movie.

Most mobile multimedia applications include video decoding, audio decoding, and other system functions. Figure 1(a) shows the power consumption of a typical mobile DMB application. As shown in Fig. 1(a), video decoding dominates total power consumption. Due to its huge computation and power consumption [2], [3], video decoding is usually implemented in the dedicated multimedia system-on-a-chip (SoC), but still the power consumption is too large for commercial use.

Figure 1(b) shows the power consumption of video decoding in DMB applications. In video decoding, macroblock-level decoding dominates the total power consumption. Therefore, it is essential to develop a low-power macroblock-level video decoding scheme to reduce the power consumption in DMB applications.

Dynamic voltage scaling [4] is one of the most efficient low-power technologies for SoC design. In most cases, dynamic power consumption due to the switching current of CMOS gates dominates total power consumption of VLSI circuits.

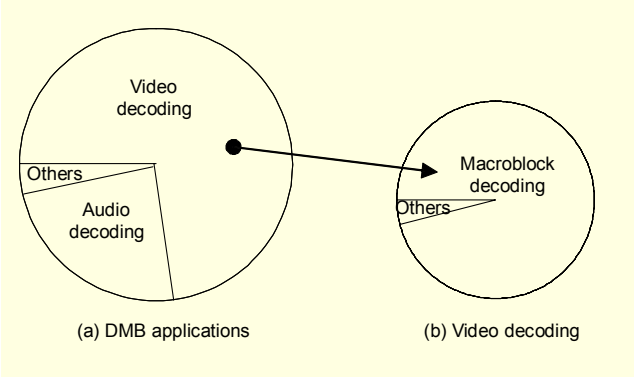

Fig. 1. Power consumption in DMB applications.

Dynamic power consumption P is given by $P \propto \frac{1}{2}C_L V^2 f$, where C_L is the load capacitance, V is the supply voltage, and f is the clock frequency [4]. Lowering supply voltage V is the most effective way to reduce power consumption, since power consumption is proportional to the square of the supply voltage. However, the circuit delay T is given by $T = 1/f \propto V/(V-V_{TH})^{\alpha}$, where V_{TH} is the threshold voltage of CMOS circuits and α is the velocity saturation index (approximately 1.3) [5]. Consequently, circuit delay increases and its corresponding system operating frequency decreases when the supply voltage is lowered. When the required workload of given applications is lower than the maximum system throughput, that application finishes its execution before its deadline. In this case, the execution of that application can slow down such that it finishes its execution just at its deadline, and the system operating frequency can be lowered. In this case, the supply voltage also can be lowered, and the power consumption can be reduced [4].

Many researches [6]-[10] have been presented in dynamic voltage scaling of multimedia signal processing. However, algorithmic support of multimedia signal processing is still needed for efficient power reduction, since there are many problems when dynamic voltage scaling is directly applied to multimedia signal processing.

This paper proposes a new low-power video decoding scheme called zero-block skipping macroblock decoding. It is effective for mobile multimedia SoCs based on dynamic voltage scaling. In the proposed scheme, information of uncoded blocks and macroblocks is stored in the user data fields of a video bitstream. This information is exploited by the video decoder SoC, and the supply voltage is lowered to the lowest possible extent for frame-level real-time video decoding.

II. Problems of Conventional Dynamic Voltage Scaling in Video Decoding

It has been reported that dynamic voltage scaling (DVS) shows significant power reduction on video encoding [6], [7] and decoding [8]-[10] since workload fluctuation is quite large due to the input data dependency of video sequences. However, in video decoding, conventional dynamic voltage scaling schemes [8]-[10] have the following implementation problems in mobile multimedia applications.

1. Hard Real-Time Constraint

Video decoding is a frame-based hard real-time application, that is, every frame should be decoded and displayed with an interval of 1/(frame rate). Therefore, dynamic voltage scaling should accurately estimate the actual execution of frame decoding so as not to miss the hard real-time deadline. Various estimation methods are proposed in conventional dynamic voltage scaling schemes on video decoding [8]-[10]. However, they cannot exactly estimate the frame decoding time, since the actual decoding time can be exactly known only after the frame is decoded [8]. Therefore, the violation of a hard real-time constraint often occurs in conventional dynamic voltage scaling schemes, and they are thus not suitable for video decoding.

In [8], a feedback controller is used to exploit a non-linear statistical model in order to estimate the decoding time of a frame. In the feedback controller, various encoded bitstreams of various frame types were decoded on the target variable voltage processor platform, and their frame decoding times were measured. There is a strong positive correlation between the encoded frame bitstream length and decoding time. Different estimation models are used for different frame types. This is quite effective in soft real-time applications. However, this approach often fails to exactly estimate the decoding time, so a missed deadline may occur as shown in Fig. 2. When the encoded frame bitstream length is 5 kbits for frame #1 and frame #2 as shown in Fig. 2(a), the feedback controller estimates the decoding time as 1/60 s at 266 MHz for both frames as shown in Fig. 2(b) since they have the same amount

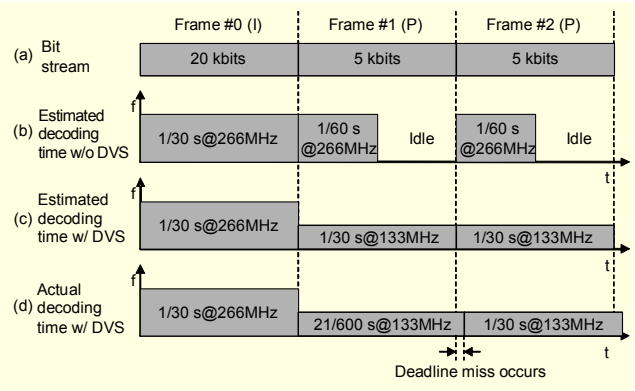

Fig. 2. Problems of the feedback controller.

of encoded bitstream. The feedback controller determines the frequency as 133 MHz for both frames as shown in Fig. 2(c). However, the actual decoding time can be different even if the encoded frame bitstream lengths are the same. Assume that the actual decoding times of frame #1 and frame #2 are 21/600 s and 1/30 s, respectively. As shown in Fig. 2(d), a missed deadline occurs although the difference between the estimated and actual decoding times is only 5%.

Frame-based dynamic voltage and frequency scaling [9] exploits a linear predictor with a moving average and weighted average to estimate the decoding time of a frame. It was reported that the prediction error of the frame decoding time is 5 to 25% [9], so it suffers from similar problems as the feedback controller. To overcome these problems, frame-based dynamic voltage and frequency scaling divides video decoding into frame-dependent decoding (FD) and frame-independent decoding (FID), and the estimation error of the frame-dependent decoding is compensated in the frame-independent decoding as shown in Figs. 3(a) to 3(c). Yet the frame-independent decoding cannot fully compensate the under-estimated frame-dependent decoding time in frame-based dynamic voltage and frequency scaling. There is an upper limit of the system operation frequency, and the required frequency

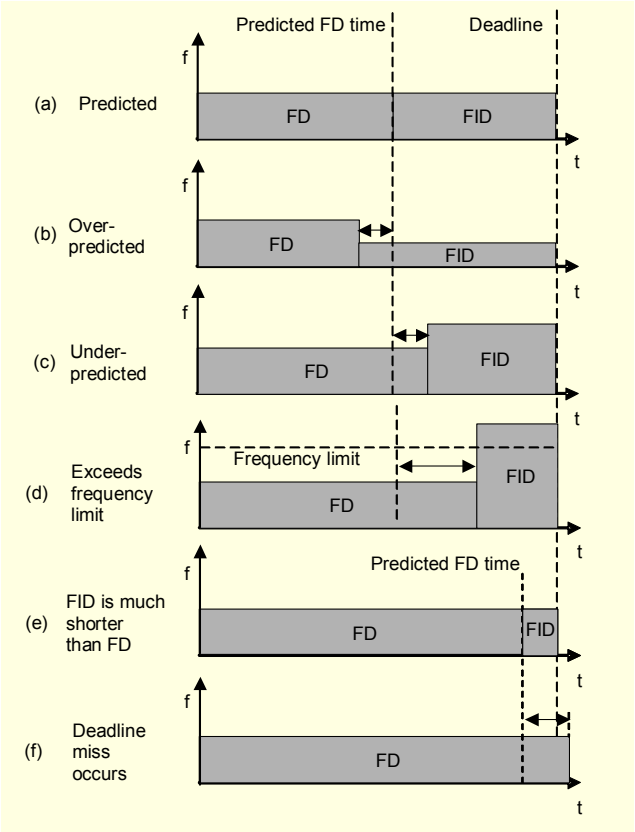

Fig. 3. Problems of the frame-based dynamic voltage and frequency scaling.

of frame-independent decoding often exceeds that limit when frame-dependent decoding is too greatly under-estimated, as shown in Fig. 3(d). Furthermore, in most video decoding, the execution time of frame-independent decoding is less than 10% that of frame-dependent decoding as shown in Fig. 3(e) since most of the execution time is occupied by macroblock decoding. Therefore, as shown in Fig. 3(f), the actual execution time of frame-dependent decoding exceeds the deadline even if it is under-estimated by more than 10%, no matter how fast the frequency is.

Power-aware video decoding [10] assumes that the video decoding runs at its worst-case execution time. The supply voltage is lowered to the extent that the video decoding can be finished within its deadline even if the worst-case execution occurs. Consequently, a missed deadline never occurs. However, power-aware video decoding cannot reduce the supply voltage enough, since the actual execution time is much shorter than the worst-case execution time in most cases [11], [12]. In [10], it is reported that the power consumption is reduced by about 10%, while the power consumption is reduced by about 40 and 30% in [8] and [9], respectively.

2. Transition Time to Change Supply Voltage

In MPEG-4 video coding [13], an 8 × 8 pixel block is transformed by discrete cosine transform (DCT), and the transformed DCT coefficients are quantized by quantization (Q). During Q, some DCT coefficients are truncated to zero. When all 64 DCT coefficients are truncated to zero, the 8 × 8 pixel block is non-coded, and the decoder does not need to perform either inverse quantization (IQ) or inverse DCT (IDCT) for that block. In low-bitrate mobile multimedia communication, the quantization parameter *mquant* is quite large, and many blocks are non-coded. Note that motion compensation (MC) is always performed even for non-coded blocks. Each macroblock header in the encoded bitstream stores non-coded block information in the *cbpy* and *mcbpc* fields. Consequently, the decoder knows whether to skip the blocks during decoding only when it starts to decode each macroblock.

In order to apply dynamic voltage scaling to macroblock decoding, the decoder determines the proper clock frequency and its corresponding supply voltage only after it decodes the *cbpy* and *mcbpc* fields in the current macroblock header since the decoder knows the actual workload to process the current macroblock only after it reads the non-coded block information. This means that the video decoder SoC determines and changes the supply voltage on a macroblock basis.

Figure 4 illustrates the problems when dynamic voltage

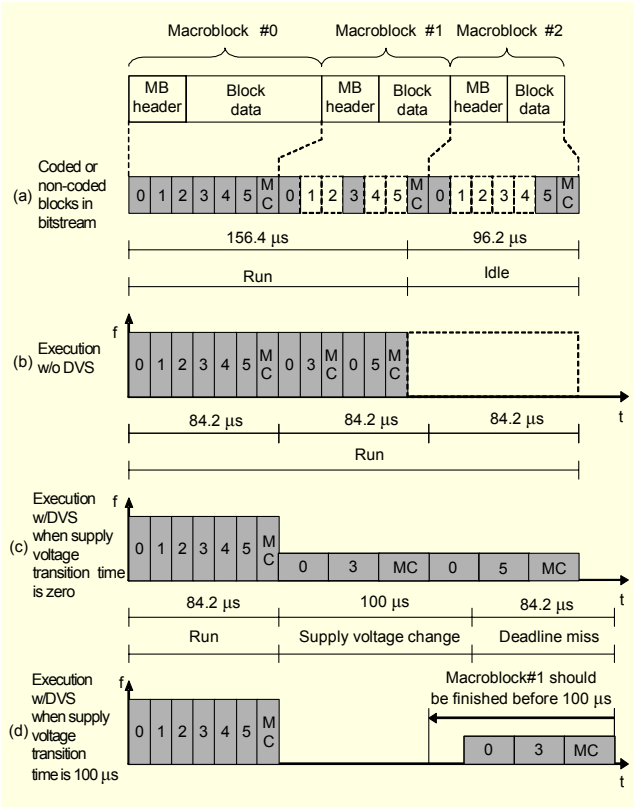

Fig. 4. Applying dynamic voltage scaling to macroblock decoding.

scaling is directly applied to macroblock decoding. Assume that the decoder processes CIF sequences (352 × 288 pixels, 30 frames/s). There are 396 macroblocks in a frame, and the time interval for each macroblock is 84.2 µs. Assume that the workloads for block decoding and motion compensation are equal. Also assume that all blocks are coded in macroblock #0, and only the 0th and 3rd blocks are coded in macroblock #1, and only the 0th and 5th blocks are coded in macroblock #2, as shown in Fig. 4(a). Note that motion compensation is performed even if some blocks are not coded. When the dynamic voltage scaling is not used, the video decoder SoC runs during 156.4 µs, and idles during 96.2 µs, as shown in Fig. 4(b).

If dynamic voltage scaling is applied and the transition time to change the supply voltage is zero, the video decoder SoC decodes only three blocks out of six blocks in macroblock #1 and macroblock #2. Consequently, the workload of the video decoder SoC is reduced to 3/7, and the clock frequency and supply voltage are reduced to about 1/2 as shown in Fig. 4(c) to finish decoding macroblock #1 and macroblock #2 in 84.2 µs, respectively. The power is proportional to the clock frequency and square of the supply voltage. The power is reduced to 1/8, since the supply voltage and clock frequency are each reduced to 1/2. Consequently, the power consumption is reduced to 1/4.

However, considering the transition time to change the supply voltage, dynamic voltage scaling has a critical problem. Usually, it takes quite a long time to change the supply voltage of VLSI circuits, since the settling time of a state-of-the-art DC-DC converter to change a supply voltage is about 100 µs. Consequently, as shown in Fig. 4(d), a missed deadline occurs in macroblock #1, and it is impossible to apply dynamic voltage on a macroblock-by-macroblock basis. Note that the decoding time for a macroblock header is very short compared to block data decoding and is ignored in Fig. 4.

III. Zero-Block Skipping Macroblock Decoding

To solve the above problem, we modified the video encoding and decoding scheme to store non-coded block information in the frame header. In MPEG-4 video coding, user data can be stored in a bitstream using the *pei* and *psupp* fields. These fields are usually ignored, and they have no effect in conventional decoders.

In this paper, we propose zero-block skipping macroblock decoding, a low-power video decoding scheme for dynamic voltage scaling. In the encoder, the total number of non-coded blocks in a frame is counted during encoding. This counting hardly requires extra computation in the encoder. The counted number of non-coded blocks is stored in the *pei* and *psupp* fields in the frame header, and the encoder bitstream is transmitted to the decoder.

In the decoder, the counted number of non-coded blocks is decoded at the start of frame decoding, and the decoder knows the total number of blocks to be decoded in a frame. This means that the decoder knows the actual workload to decode a frame, and the decoder determines the proper clock frequency and its corresponding supply voltage to finish frame decoding just at the frame deadline. In this case, the video decoder SoC changes the supply voltage only once at the start of frame decoding, and it doesn't need to change the supply voltage during frame decoding. The transition time to change the supply voltage (approximately 100 µs) is negligible compared to the frame processing time (33333.3 µs), and the video decoder can change the supply voltage without missing the deadline. This scheme can be easily adapted to H.264 [14] and other video coding standards.

The proposed scheme is illustrated in Fig. 5 in detail. In the encoder, the reference frame is encoded, and the number of non-coded blocks in the reference frame is counted, as shown in ① of Fig. 5(a). The generated bitstream is stored in the memory until the encoding of the reference frame is finished, ② of Fig. 5(a). After that, the number of non-coded blocks is inserted in the *pei* and *psupp* fields of the frame header, ③ of Fig. 5(a). The *pei* and *psupp* fields are 1 and 8 bits, respectively. The numbers of total blocks in a reference frame are 2376 and

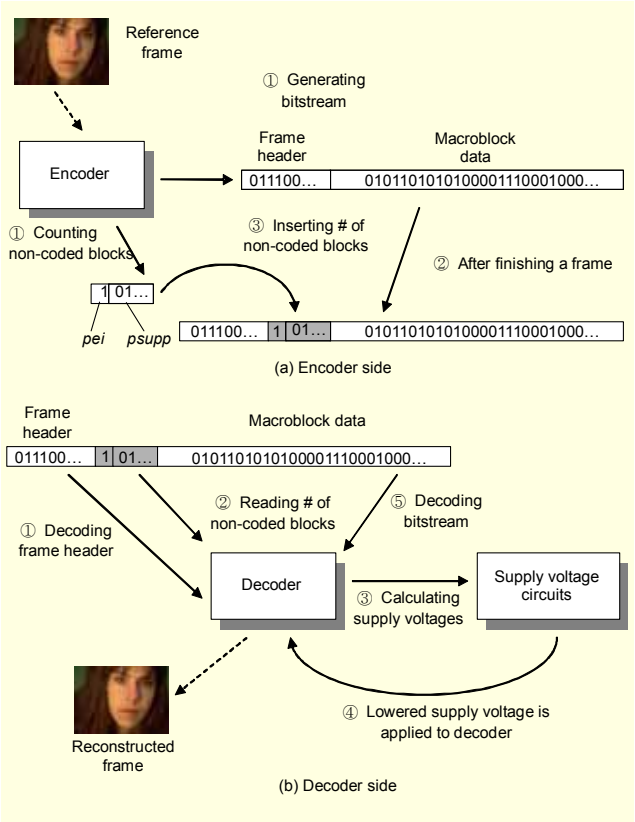

Fig. 5. The proposed zero-block skipping macroblock decoding scheme.

9504 in CIF and CCIR601 formats, respectively, and two pei and psupps are required to store the number of non-coded blocks. The additional data is 18 bits per frame, and it increases 0.28% of the encoded bits when the bitrate is 192 kb/s and the frame rate is 30 frames/s, which is negligible.

In the decoder, the number of non-coded blocks is read during frame header decoding, ① and ② of Fig. 5(b). The decoder calculates the proper supply voltage and its corresponding supply voltage to finish frame decoding just at its deadline, ③ of Fig. 5(b). The video decoder SoC lowers the supply voltage to reduce power consumption, ④ of Fig. 5(b), and it runs at the single lowered supply voltage during frame decoding, ⑤ of Fig. 5(b). When the video decoder SoC doesn't support dynamic voltage scaling, the number of non-coded blocks in the *pei* and *psupp* fields are simply ignored, and the video decoder SoC performs conventional video decoding.

IV. Determination of Clock Frequency and Supply Voltage for Dynamic Voltage Scaling

In the proposed scheme, the proper supply voltage for dynamic voltage scaling is calculated as follows: N is the total number of blocks in a frame, N_{NC} is the number of non-coded blocks in a frame, and f_{MAX} and V_{MAX} are the maximum system operating frequency and maximum supply voltage to decode a frame when there are no non-coded blocks, respectively. When the video decoder SoC runs at f_{MAX} and V_{MAX}, the required system operating times for block decoding and macroblock motion compensation are t_B and t_{MC}, respectively. The required system operating frequency and the required supply voltage to decode a frame when there are N_{NC} non-coded blocks are f_{SYS} and V_{SYS}, respectively.

Assuming that the decoding time for a frame header is negligible, f_{SYS} is calculated as (1). Note that f_{SYS} decreases as N_{NC} increases, since the required computation decreases as the number of non-coded blocks increases. When f_{SYS} decreases, the required supply voltage is lowered and the power consumption is reduced. Note that (1) is calculated only once per each frame, and the computational overhead is negligible.

$$f_{SYS} = f_{MAX} \times \frac{\left(N - N_{NC}\right) \times t_B + \frac{N}{6} \times t_{MC}}{N \times t_B + \frac{N}{6} \times t_{MC}}. \qquad (1)$$

After f_{SYS} is calculated, V_{SYS} is calculated as (2) since $T = 1/f \propto V/(V\text{-}V_{TH})^{\alpha}$, where V is the supply voltage, V_{TH} is the threshold voltage, and α is the velocity saturation index (approximately 1.3) [5]. Since f_{MAX}, V_{MAX}, V_{TH}, and α are constant for given VLSI circuits, V_{SYS} can be pre-calculated and tabulated as a lookup table of f_{SYS}. Note that the computational overhead for (2) is also negligible since it is performed by a table look-up.

$$f_{MAX} \times \frac{V_{SYS}}{\left(V_{SYS} - V_{TH}\right)^{\alpha}} = f_{SYS} \times \frac{V_{MAX}}{\left(V_{MAX} - V_{TH}\right)^{\alpha}}. \qquad (2)$$

Figure 6 illustrates the power reduction of the proposed scheme when $t_B = t_{MC}$ and $N_{NC} = (5/12)N$. When dynamic voltage scaling is not applied, the decoder runs and stops as in Fig. 6(a), and the supply voltage and its corresponding power consumption are determined as in Fig. 6(b) and 6(c). When dynamic voltage scaling is applied, $f_{SYS} = (1/2)f_{MAX}$ and V_{SYS} is approximately $(1/2)V_{MAX}$, and the supply voltage and its corresponding power consumption are determined as in Fig. 6(d) and 6(e). The power is proportional to the clock frequency and the square of the supply voltage. The power is reduced to 1/8, since the supply voltage and clock frequency are each reduced to 1/2. Consequently, the power consumption is reduced to 1/4.

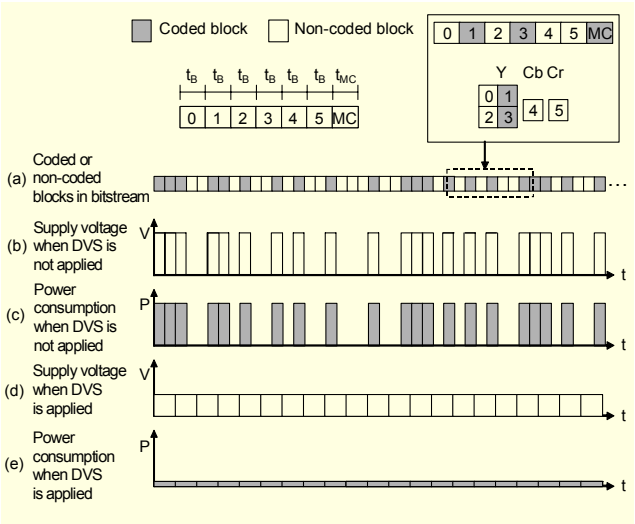

Fig. 6. Power reduction of the proposed zero-block skipping macroblock decoding scheme.

V. Simulation Results

In this paper, MPEG-4 SP@L2 video coding was tested with four CIF sequences (352 × 288 pixels, 30 frames/s), "Claire," "Miss America," "Foreman," and "Carphone." The encoded bitrate is 192 kb/s. The supply voltage V_{DD}, the threshold voltage V_{TH}, and the velocity saturation index α are 2.5 V, 0.5 V, and 1.3, respectively. Transition delay to change the supply voltage is 100 μs.

Since we do not yet have a variable voltage processor platform for dynamic voltage scaling, the simulation results were obtained as follows. First, the encoded bitstream with the number of macroblocks in the frame header was obtained by running an MPEG-4 video encoder program. Second, the MPEG-4 video decoder program was simulated in the fixed voltage processor platform. The execution time of each block decoding without dynamic voltage scaling was measured by accessing the processor's internal real-time clock. Third, we made a variable voltage processor emulator, and the execution time and power consumption with dynamic voltage scaling were calculated on the emulator.

Table 1 shows the normalized number of non-coded blocks N_{NC}/N, the normalized supply voltage V_{SYS}/V_{MAX}, and the normalized power consumption P_{SYS}/P_{MAX}. As shown in Table 1, non-coded blocks are about 50 to 65% of the total blocks in a frame. Note that the encoded bitrate is 192 kb/s, which is typical for low bitrate mobile multimedia applications. Since the bitrate is quite low, the quantization parameter is quite large; therefore, many DCT coefficients are truncated to zero, and thus the number of non-coded block N_{NC} is quit high.

From (1), the supply voltage is greatly lowered when N_{NC} is high, and the supply voltage and corresponding power consumption are also greatly lowered. In the simulation results, the supply voltage is lowered to about 25 to 30%, and the power consumption is reduced to about 5 to 10%.

Table 1. The normalized number of non-coded blocks, the normalized supply voltage, and the normalized power consumption.

Parameters	Claire	Miss America	Foreman	Carphone
Normalized number of non-coded blocks, N_{NC}/N	0.66	0.65	0.52	0.58
Normalized supply voltage, V_{SYS}/V_{MAX}	0.24	0.25	0.30	0.28
Normalized power consumption, P_{SYS}/P_{MAX}	0.054	0.057	0.102	0.072

VI. Conclusion

In this paper, a novel low-power video decoding scheme for mobile multimedia SoC is proposed based on dynamic voltage scaling. In video decoding, some DCT coefficients are truncated to zero during quantization, and a block with all zero DCT coefficients are not coded in the bitstream. This non-coded block needs no decoding, and the decoder can slow down the system operating frequency. By exploiting dynamic voltage scaling, the supply voltage can be lowered and the power consumption can be reduced. However, the decoder knows the existence of a non-coded block when it starts to decode the macroblock header, so the supply voltage should be adjusted on a macroblock basis. Usually, the transition time to change the supply voltage is longer than macroblock decoding, so a missed deadline often occurs and dynamic voltage scaling cannot be applied. In the proposed scheme, the encoder counts the number of non-coded blocks in a frame, and stores this information into the bitstream. Using this information, the decoder can calculate the proper single supply voltage for frame decoding, and the dynamic voltage can be applied to reduce power consumption. The proposed scheme is very flexible and can be applied to most existing video coding standards. Simulation results show that the proposed scheme reduces the power consumption to about 1/10 to 1/20 in MPEG-4 video decoding.

References

[1] J. Rabaey, "Low-Power Silicon Architectures for Wireless Communication," *Proc. of Asia and South Pacific Design Automation Conf.*, 2000, pp. 379-380.

[2] S. Lee, "Pipelined Macroblock Processing to Reduce Internal Buffer Size of Motion Estimation in Multimedia SoCs," *ETRI J.*,

vol. 25, no. 5, Oct. 2003, pp. 297-304.
[3] S. Kim, J. Park, S. Park, B. Koo, K. Shin, K. Suh, I. Kim, N. Eum, and K. Kim, "Hardware-Software Implementation of MPEG-4 Video Codec," *ETRI J.*, vol. 25, no. 6, Dec. 2003, pp. 489-502.
[4] A. Chandrakasan and R. Brodersen, *Low Power Digital CMOS Design*, Kluwer Academic Publishers, 1995.
[5] T. Sakurai and A. Newton, "Alpha-Power Law MOSFET Model and Its Application to CMOS Inverter Delay and Other Formulas," *IEEE J. of Solid State Circuits*, vol. 25, no. 2, Feb. 1990, pp. 584-594.
[6] S. Lee, S. Lee, and T. Sakurai, "Energy-Constrained VDD/VTH Hopping Scheme with Run-Time Power Estimation for Low-Power Real-Time VLSI Systems," *J. Circuits, Systems, and Computers*, vol. 11, no. 6, Dec. 2002, pp. 601-620.
[7] C. Im and S. Ha, "Energy-Optimization for Latency-and Quality-Constrained Video Applications," *IEEE Design and Test of Computers*, vol. 33, no. 5, Oct. 2004, pp. 358-366.
[8] J. Pouwelse, K. Langendoen, R. Lagendijk, and H. Sips, "Power Aware Video Decoding," *Proc. Picture Coding Symposium*, 2001, pp. 303-306.
[9] K. Choi, K. Dantu, W. Cheng, and M. Pedram, "Frame-Based Dynamic Voltage and Frequency Scaling for a MPEG Decoder," *Proc. Int'l Conf. Computer-Aided Design*, 2002, pp. 732-737.
[10] C. Poellabauer and K. Schwan, "Power-Aware Video Decoding Using Real-Time Event Handlers," *Proc. Int'l Workshop on Wireless Mobile Multimedia*, 2002, pp. 72-79.
[11] D. Shin, J. Kim, and S. Lee, "Low-Energy Intra-Task Voltage Scheduling Using Static Timing Analysis," *Proc. Design Automation Conf.*, 2001, pp. 438-443.
[12] S. Lee and T. Sakurai, "Run-Time Voltage Hopping for Low-Power Real-Time Systems," *Proc. Design Automation Conf.*, 2000, pp. 806-809.
[13] ISO/IEC JTC1/SC29/WG11 14496-2, *Coding of Audiovisual Object: Visual*, 1998.
[14] ITU-T Rec. H.264, *Advanced Video Coding*, 2002.

Seongsoo Lee received the BS, MS, and PhD degrees in electrical engineering 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, University of Tokyo, Japan. From 1998 to 2002, he was a research professor in the Department of Information Electronics Engineering, Ewha Womans University, Korea. Since 2002, he has been an assistant professor in the School of Electronics Engineering, Soongsil University, Korea. His research interests include multimedia SoC, low-power SoC, low-power multimedia signal processing, and system-in-package.