A Feed-Foreward Dynamic Voltage Frequency Management by Workload Prediction for a Low Power Motion Video Compression

Masahiko Yoshimoto

Department of Computer and Systems Engineering
Kobe University
1-1 Rokkodai, Nada-ku, Kobe,
657-8501, Hyogo, Japan
Tel : +81-78-803-6214
Fax : +81-78-803-6390
e-mail : yosimoto@cs.kobe-u.ac.jp

Kentaro Kawakami

Department of Electrical and Electronic Engineering
Kanazawa University
2-40-20, Kodatsuno, Kanazawa,
920-8667, Ishikawa, Japan
Tel : +81-76-234-4862
Fax : +81-76-234-4873
e-mail : kawakami@mics.ee.t.kanazawa-u.ac.jp

Abstract - This paper proposes a feed-forward dynamic voltage and frequency management (FFDM) method to minimize the total power of software based video compression processing. This method cooperatively controls operating voltage/frequency and body bias voltage according to the workload predicted by a forward analysis to reduce both of dynamic power and leakage power. Simulation results indicate that the FFDM method can reduce power dissipation of MPEG4 encoding by 65% to 80%, depending on sequence activities.

1. INTRODUCTION

The 3rd generation wireless communication services have been started, and rich media services, mainly audio/visual communication or streaming services, have been expected to be a key application through mobile phone terminals. The visual communication processing requires high processing performance around several hundred mega operation per second (MOPS), therefore dedicated hardware approach has been a major approach in terms of low power advantages on mobile terminals [1] [2]. However, required specifications for the video compression LSI, which is a key device in the visual communication system, are not only low power characteristics, but also flexibility for the future system. In the coming ubiquitous era, video compression LSI has to realize flexibility to various video compression standards, extensibility for expansion of resolution and frame rate, and re-usability as a intellectual property (IP) core. To satisfy these three requirements, software based processing is the best approach. Recent embedded RISC processor fabricated by sub-decimicron technology achieves several hundred MOPS, and it can handle a real time video compression software. However, the power consumption of software based processing is not adequately low.

Moreover, with the progress of the technology scaling beyond 90[nm], the threshold voltage is lowered, and it causes growth of leakage power. Figure 1 shows simulated power consumption of 256KB SRAM in 90[nm] technology. It indicates that a suppress of both the dynamic power and the leakage power is indispensable in the coming sub-decimicron technology era. This paper proposes a feed-forward dynamic voltage and frequency management method to minimize the total power of software based video compression processing. This method cooperatively controls operating voltage/frequency and body bias voltage to reduce both of dynamic power and leakage power.

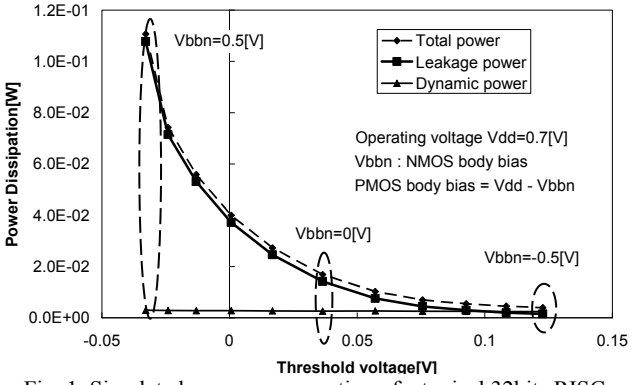


Fig. 1. Simulated power consumption of a typical 32bits RISC processor in 90[nm] technology.

2. POWER CONSUMPTION OF 32BITS RISC PROCESSOR IN 90NM TECHNOLOGY

In sub-decimicron technology, both of subthreshold leakage power and dynamic power should be taken into account. A V_{dd}-hopping scheme [3] and V_{bb}-hopping scheme [4] was proposed to reduce the dynamic power and the leakage power, respectively. These schemes dynamically controls operating voltage or body bias voltage in association with operating frequency. Considering the reduction of total power consumption including both of the dynamic and the leakage power, excessive hopping down of V_{dd} to reduce the dynamic power increases the leakage power. The scaling down of V_{dd} degrade the operating frequency, therefore the body bias voltage has to be controlled toward lowering the threshold voltage to compensate the operating frequency. It results in the increase of the leakage power. For this reason, balancing the supply voltage and the body bias voltage is desirable to minimize the total power consumption [5].

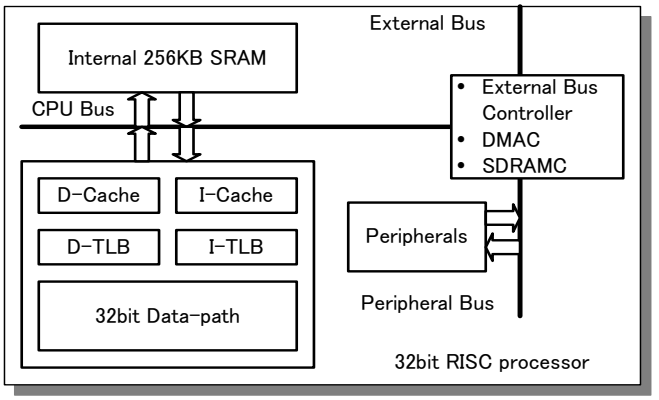


Fig. 2. Block diagram of a conceptual 32bits RISC processor.

Figure 2 shows the block diagram of the example of 32bits RISC processor to implement software video compression, which has a 32bits data-path, 16KB data-cache (D-cache), 16KB instruction-cache (I-cache), 256KB internal SRAM, DMA controllers, peripherals and so on. A SPICE simulation was executed to estimate the RISC processor performance using a model file of "Common Design Rules for 0.1 micron" recommended by the Semiconductor Technology Academic Research Center (STARC). The toggle rate of logic portion is set to 15[%]. The accessing rate of the D-cache, the I-cache and the internal SRAM are set to 50[%], 90[%] and 13[%], respectively. These values are estimated by the HDL (hardware description language) level simulation with MPEG4 visual compression software.

Simulation results are shown in Fig. 3, Fig. 4 and Fig. 5. Figure 3 shows the maximum operating frequency versus the operating voltage V_{dd} and the NMOS body bias voltage V_{bn}. The PMOS body bias voltage V_{bp} is set symmetrically to V_{dd} - V_{bn}. Applying the higher NMOS body bias voltage lowers the threshold voltage, which thereby allows a higher operating frequency. Figure 4 shows the minimum NMOS body bias voltages required for 50, 100, 150, 200, and 250 [MHz] operation; Fig. 5 shows power dissipation under those body bias conditions. The five broken lines in Fig. 4 correspond to those in Fig. 5. Power dissipation of the V_{dd}-control processor (constant V_{bn}) and the V_{bb}–control processor (constant V_{dd}) are also shown in Fig. 5. Simulation results indicate the existence of minimum-power V_{dd} and V_{bn} combinations because of the difference of the dynamic and the leakage power characteristics. Therefore, we infer that the choice of proper V_{dd} and V_{bn} combinations for each frequency, namely the V_{dd}-V_{bb}-control, can minimize the total power dissipation.

3. PROCESSING PERFORMANCE FOR MPEG4-VISUAL ENCODING

MPEG4 QCIF 15[frame/s] video compression requires approximately 200-300 [MOPS]. However, these values are the average values and high motion sequence requires more performance, and low motion sequence requires less performance. Required performance depends totally on the video sequence activity. Figure 6 shows a block diagram of MPEG4 processing. Shaded blocks in Fig. 6 are the processing function whose performance is affected by video sequence activity. Required performance of motion compensation (MC), Inverse DCT (IDCT), Inverse

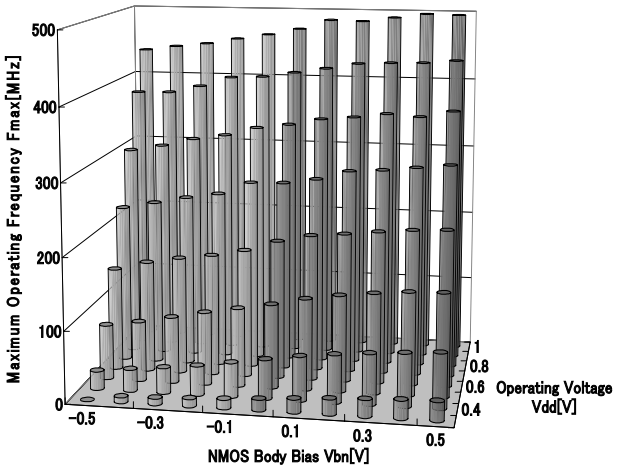


Fig. 3. Maximum operating frequency of a 32bits RISC processor (simulated).

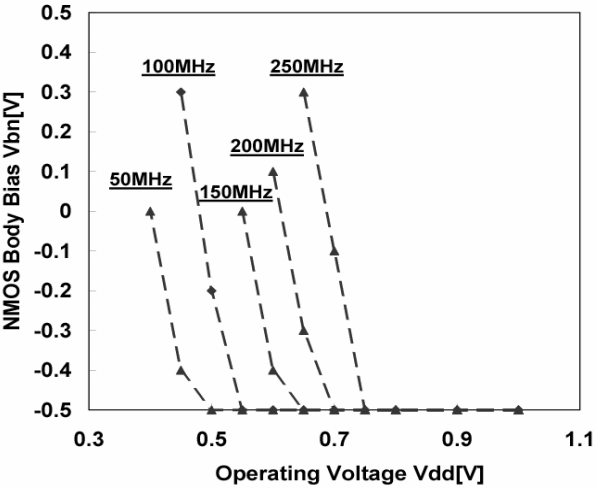


Fig. 4. Vdd – Vbb combination to achieve certain operating frequency of a conceptual 32bits RISC processor in 90[nm] technology (simulated)..

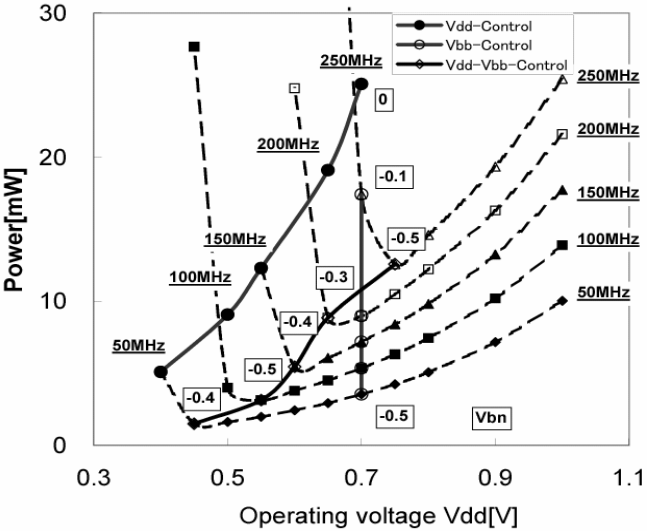


Fig. 5. Simulated power consumption of a conceptual 32bits RISC processor in 90[nm] technology (simulated).

Quantization (IQ), and variable length coding (VLC) has been influenced according to the characteristics of the video sequence. Each processing is computationally intensive function, and approximately eighty percent of total MPEG4 performance is occupied by MC, IDCT, IQ, and VLC. Consequently, total required MPEG4 processing performance completely varies according to the sequence activity.

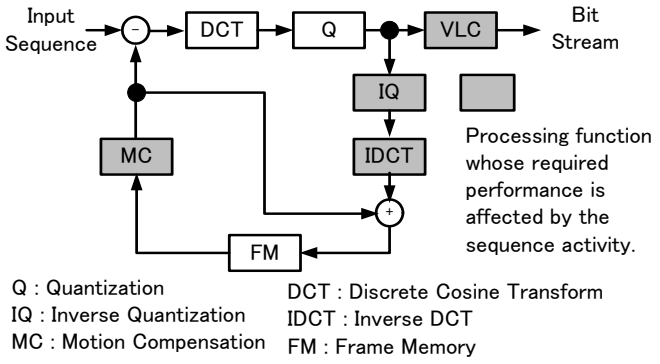


Fig. 6. Block Diagram of MPEG4 processing.

4. DYNAMIC VOLTAGE/FREQUENCY MANAGEMENT METHOD BY FORWARD ANALYSIS

As mentioned in Section 3, the required performance of MPEG4 visual compression dynamically changes according to the sequence activity. Also as simulated in Section 2, the operating voltage/frequency management drastically reduces the power of RISC processors. There exist the power minimum V_{dd}-V_{bb} combinations. Combining these characteristics, the proposed feed-forward voltage/frequency management using our unique forward analysis algorithm can minimize the power of the software based MPEG4 visual compression.

4.1 DETAILS OF OUR PROPOSED FEED-FORWARD DYNAMIC VOLTAGE/FREQUENCY MANAGEMENT METHOD

A control of feed-forward dynamic management method is show in Fig. 7, and the timing sequence of MPEG4 process adopting the feed-forward dynamic voltage/frequency management method are shown in Fig. 8. MPEG4 visual compression processing using the our proposed method is as follows;

1) Prediction of the workload for MPEG4 visual compression per frame using the parameters of motion activity or other parameters.
2) Calculation of the required operating frequency (F_p) based on the predicted workload.
3) Controlling the frequency at predicted value and setting V_{dd} and V_{bb} which minimize the power at that frequency.
4) Compression of the new frame at the modified voltage and frequency.

The above 1) - 3) processes correspond to the feed-forward dynamic voltage/frequency management method, and it requires only less than 1[MHz] cycle, which is negligible compared to MPEG4 compression. The duration time for voltage and frequency stability after controlling voltage value ("B" in Fig. 8) is micro seconds order which is also negligible comparing to the allocated time for frame (ex. allocated time for a frame in case of 15[frame/s] is 66.7[ms]).

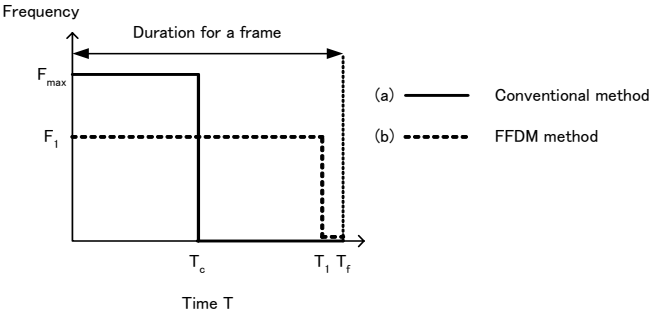


Fig. 7. Frequency/voltage control method.

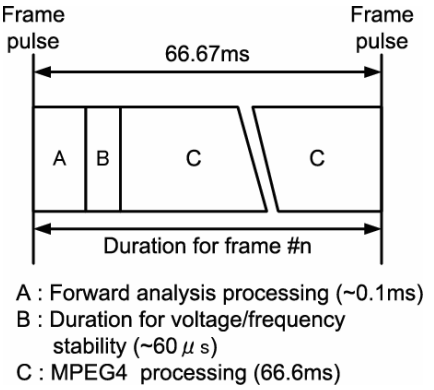


Fig. 8. Timing sequence for the FFDM method.

Table 1. Processing function.

Function	Parameters affecting for the workload	
MC	Numbers of MB matching	*N*
IQ	Numbers of valid DCT coefficients	*VC*
IDCT	Numbers of valid blocks	*VB*
VLC	Numbers of valid blocks	*VB*

4.2 PREDICTION OF REQUIRED PERFORMANCDE BY FORWARD ANALYSIS

The forward analysis algorithm predicts future frame processing performance. Table 1 describes the parameters that affect to these processing functions. The forward analysis algorithm predicts the required performance from the following parameters;

1) Number of MB block matching : *N*
2) Number of valid DCT coefficients: *VC*
3) Number of valid blocks: *VB*

The parameters of *N*, *VB*, *VC* are assumed to be predicted from the following equations respectively;

$$N = a \times N' + b \times ABS_f + c \times \Delta Q \quad (1)$$

$$VB = d \times VB' + e \times ABS_f + f \times \Delta Q \quad (2)$$

$$VC = g \times VC' + h \times ABS_f + i \times \Delta Q \quad (3)$$

where ABS_f is sum of absolute difference of luminance between a current frame and the previous frame, *N'*, *VB'* and *VC'* are a actual number of MB matching processing times, an actual number of valid blocks, and an actual number of valid DCT coefficients at the previous frame, respectively. To predict *N* value, three parameters (*N'*, ABS_f and *VC*) are chosen as affecting parameters.

N': Video sequences have good correlation between frames. When number of MB matching is large in a frame, it is tended to be large in a next frame.

ABS_f: ABS_f indicates the differential between frames, and when ABS_f is large, then *N* will be large.

ΔQ: The increase of ΔQ results in the prediction error. The prediction error increases the *N*.

VB and *VC* are also assumed to be predicted from three parameters with the same assumption. Furthermore in this paper, required processing performance for motion compensation processing (F_{me}), IQ processing (F_{iq}), IDCT (F_{idct}), VLC (F_{vlc}) are assumed to be predicted from the following equations respectively;

$$F_{me} = j + A \times N \quad (4)$$

$$F_{iq} = k + B \times VC \quad (5)$$

$$F_{idct} = l + C \times VB \quad (6)$$

$$F_{vlc} = m + D \times VC \quad (7)$$

where *A* is processing performance for a MB matching, *B* is processing performance for a IQ processing, *C* is processing performance for a IDCT processing, *D* is processing performance for a VLC processing, and *j*, *k*, *l*, *m* are constant parameters. Total required performance F_p is ;

$$F_p = F_{me} + F_{iq} + F_{idct} + F_{vlc} + F_{others} \quad (8)$$

where F_{others} is rest of MPEG4 processing. Substituting Eq. (1) - (7) to Eq. (8), F_p is predicted from the parameters of *N'*, *VB'*, *VC'*, ABS_f and ΔQ defined as Eq. (9).

$$F_p = n + \alpha \times N' + \beta \times VB' + \gamma \times VC' + \delta \times ABS_f + \varepsilon \times \Delta Q \quad (9)$$

where $\alpha, \beta, \gamma, \delta, \varepsilon$ are coefficients, and *n* is constant value.

5. SIMULATION RESULTS

5.1 CLOCK FREQUENCY PREDICTION

In order to decide constant parameters at Eq. (9), simulation has been executed on the reference kit of a commercial 32bit RISC processor [6]. The simulation has been led with constant *Q* by 17 sequences each of which has originally 150 frames or 5 seconds. Figure 9 shows examples of the simulated sequences. The simulation condition is described in Table 2. The simulation flow is as follows;

1) Monitoring actually required performance (F_a), *N'*, *VB'*, and *VC'* form MPEG4 software running on the 32bits RISC processor.
2) Determination of the values of coefficient of *n*, $\alpha, \beta, \gamma,$ and δ in Eq. (9) by the regression analysis method.

Also from the simulation results and Eq. (9), predicted frequency F_p[MHz] is defined as;

$$F_p = (89.94 + 0.0114 \times N' + 0.0666 \times VB' + 0.0031 \times VC' + 0.4150 \times ABS_f \times 10^5) / 10^6 \quad (10)$$

Eq. (10) is obtained by the regression analysis method from 1018 points in 17 sequences. Figure 10 shows the correlation between predicted frequency (F_p) from Eq. (10) and actually required frequency (F_a). The measured F_a lies between 92[MHz] and 188[MHz] for high quality implementation, depending on characteristics of video sequences. These values are reasonable for the single RISC architecture without additional DSP core. Form the Fig. 10, Eq. (10) well predicts the actually required frequency. The case that the predicted frequency is less than the actually required frequency results in a failure situation. Therefore, Eq. (10) should be modified in order to avoid the frequent error situation. Prediction mismatch does not occur in the area of $F_p > F_a$ in the Fig. 10. By the following Eq. (11) modified from Eq.(10), 99.9% of points satisfy the condition of $F_p > F_a$.

$$F_p = (89.94 + 0.0114 \times N' + 0.0666 \times VB' + 0.0031 \times VC' + 0.4150 \times ABS_f \times 10^5) \times 1.1 / 10^6 \quad (11)$$

(a) ``Akiyo'' (b) ``Boat''

(c) ``Bus'' (d) ``Susie''

Fig. 9. Examples of simulated sequences.

Table 2. Simulation condition.

Frame Size	QCIF (176 × 144 pixels)
Frame rate	15 frame/s
Target bitrate	128 kbps
ME algorithm	Three Step Search
ME search range	±11 × ±11

5.2 POWER CONSUMPTION REDUCTION

The required maximum frequency is roughly 230[MHz] assuming the maximum number of the block matching N, the valid block VB and the valid DCT coefficient VC. The power consumption reduction ratio r is defined as follows;

$$r = \frac{1}{M} \sum_{i=1}^{M} \frac{p_a}{p_h} \qquad (12)$$

where M is the number of frame in a sequence, p_a is power consumption per frame controlled by our method, and p_h is power consumption per frame at conventional method. Here we assume a 32bit processor with five operating modes, each of which has operating frequency of 250MHz, 200MHz, 150MHz, 100MHz and 50MHz. In the case that the predicted frequency is greater than 100[MHz] and less than equals to 150[MHz], the predicted frequency range becomes 150[MHz]. In this range, for example, p_a of the V_{dd}-hopping scheme equals to 12.3[mW] from Fig. 5. Figure 11 shows the power consumption reduction ratio r with four sequences that includes the best case (low-motion sequence "Akiyo") and the worst case (high-motion sequence "Bus") among the 17 sequences. The power dissipation is estimated to be reduced by 65[%] to 82[%] with the V_{dd}-V_{bb}-hopping scheme. The reduction rate is enhanced in case of a larger capacity of embedded SRAM.

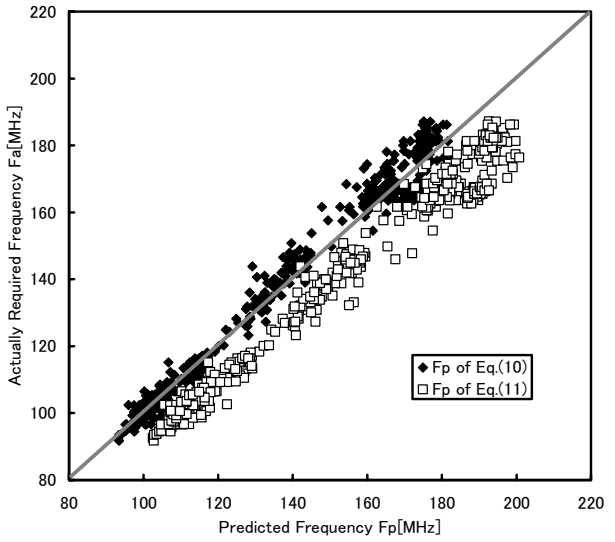
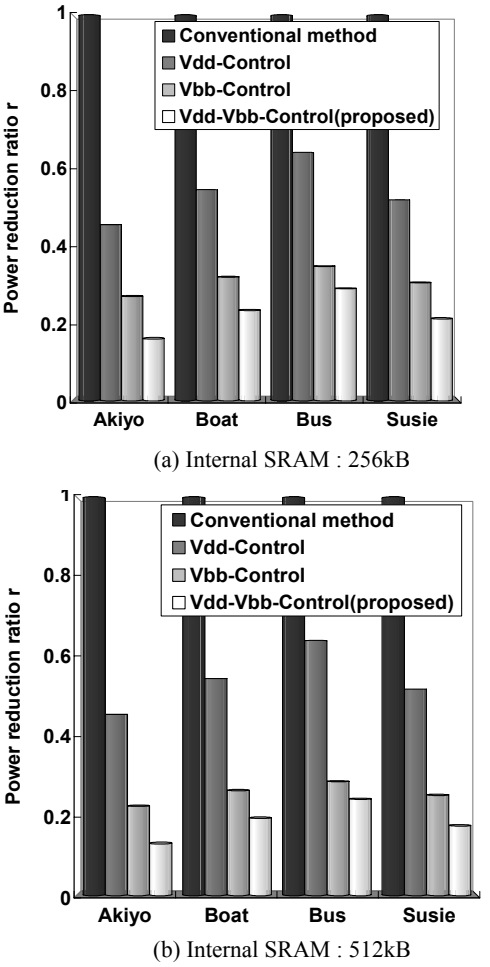


Fig. 10. Predicted frequency (F_p) v.s. actual frequency (F_a).

5.3 Error Recovery Methods from Prediction Mismatch

If forward analysis predicts less workload than the actual workload, the encoding process cannot be finished within the allocated time and the system fails. For this case, we introduce two types of error recovery methods. One of them is the following:

> The number of completed MBs is checked at every third frame. If that number is less than the expected number, the operating frequency is adjusted higher to keep pace with the process.



(a) Internal SRAM : 256kB

(b) Internal SRAM : 512kB

Fig. 11. Power consumption reduction ratio.

Figure 12(a) shows an example of frequency adjustment. The vertical axis represents the number of uncompleted MBs. Figure 12(a) shows that the frequency is adjusted to F_{adj} at time $2T_f$ /3; thereby, the remaining performance is enlarged. Applying this error recovery method, almost all frames are encoded within the allocated time. However, another recovery method is prepared for very few failure frames. The flow of this recovery method is the following:

> In cases where the encoding process cannot be finished within the allocated time, uncompleted MBs are forcedly processed as not-coded MBs, as shown in Fig. 12(b). The forcedly not-coded MB process requires only two simple processes. One is to output a predefined not-coded bit string to the bit stream. The other is to copy pixels corresponding to the not-coded MBs from a reference frame memory to a reconstructed frame memory. These two processes require only a few thousand cycles per MB. Therefore, the switchover from the normal encoding process to the forcedly not-coded MB process reduces the workload remarkably. Consequently, the encoding process can be finished within the allocated time using this recovery method.

However, these not-coded MBs are recognized as error data; thereby, the picture quality of the error frame is subjectively damaged. At the next frame, efficiency of motion compensation becomes worse around the forcedly not-coded MBs, which may cause some degradation to the next continuous frames. We have simulated the case in which successive 1, 3 and 5 MBs are forcedly not-coded at the end of frame #0 to evaluate both PSNR and subjective quality. Simulation results show that the PSNR of the error frame is degraded 0.75 [dB] at most, and that the error frame is subjectively damaged. However in the following frames, the error is not subjectively recognized as shown in Fig.13 and the degradation of PSNR shown in Fig. 14 is less than 0.03 [dB].

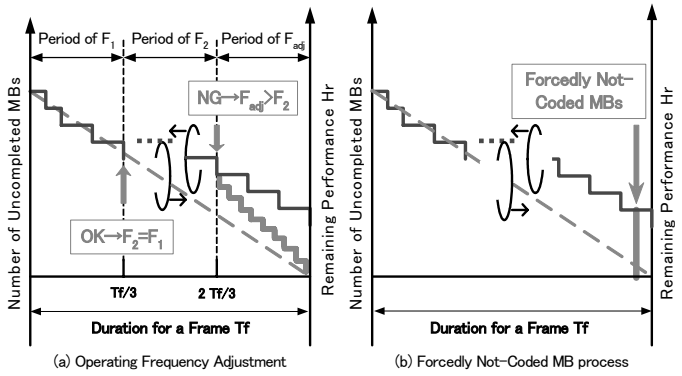


Fig. 12. Error Recovery Methods from Prediction Mismatch.

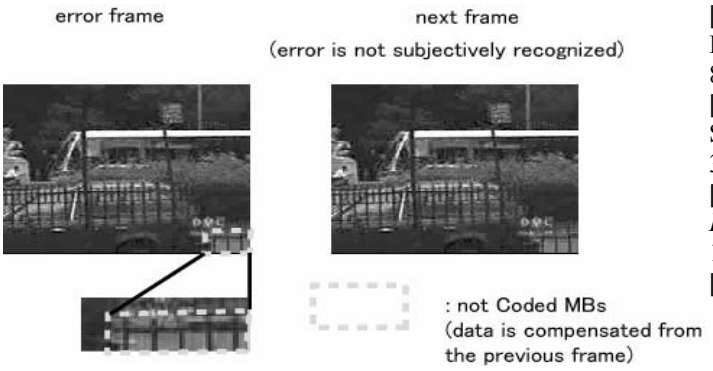


Fig. 13. Error frame caused by prediction mismatch.

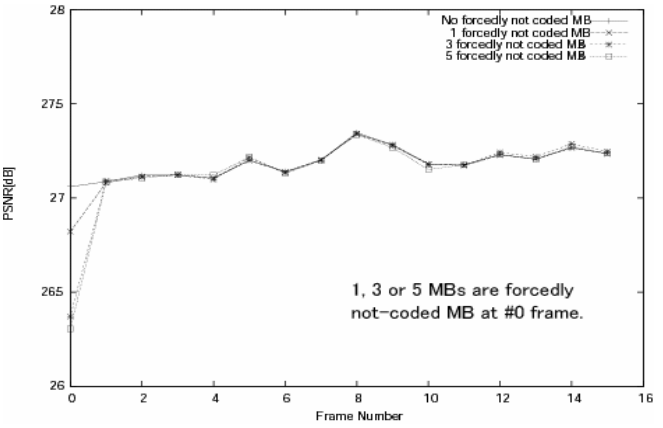


Fig. 14. Impact of the forcedly not-coded MB process.

6. SUMMARY

Low power approach for MPEG4 visual compression processing applying the feed-forward dynamic voltage frequency management has been presented. Combining the feed-forward dynamic management according to the predicted processing performance and the characteristics of video compression processing, the proposed method achieves the minimum power consumption. The simulation results indicate that the forward analysis algorithm well predicts the actual processing performance. By controlling voltage and frequency of a RISC processor dynamically by every frame, the power dissipation is reduced by 65[%] to 82[%]. In on and after 90[nm] technology era, the feed-forward dynamic voltage frequency management adopting the forward analysis algorithm effectively reduces the total power consumption of visual compression processing.

7. REFERENCES

[1] M. Takahashi, M. Hamada, T. Nishikawa, H. Arakida, Y. Tsuboi, T. Fujita, F. Hatori, S. Mita, K. Suzuki, A. Chiba, T. Terazawa, F. Sano, Y. Watanabe, H. Momose, K. Usami, M. Igarashi, T. Ishikawa, M. Kanazawa, T. Kuroda, and T. Furuyama. “A 60mW MPEG4 video codec using clustered voltage scaling with variable supply-voltage scheme.” *Proceedings of IEEE International Solid-State Circuits Conference*, 1998, pp.36–37.

[2] H. Ohira, T. Kamemaru, H. Suzuki, K. Asano, and M. Yoshimoto. “A low power media processor core performable CIF30fr/s.” *IEICE Transactions on Electronics*, Feb. 2001, vol.E84-C, no.2, pp.157–165.

[3] S. Lee, et al., “Run-time voltage hopping…,” in IEEE/ACM Proc. Design Automation Conf., 2000, pp. 806-809.

[4] K. Nose, et al., “Vth-Hopping Scheme to Reduce Subthreshold Leakage…,” *IEEE J. of Solid-State Circuits*, Vol. 37, No. 3, 2002, pp. 413-419.

[5] J. Kao, et al., “A 175-mV Multiply-Accumulate Unit…,” *IEEE J. of Solid-State Circuits*, Vol. 37, No.11, 2002, pp. 1545-1554.

[6] M32700μT-Engine, http://www.renesas.com/