IEICE TRANS. FUNDAMENTALS, VOL.E89–A, NO.12 DECEMBER 2006

PAPER Special Section on VLSI Design and CAD Algorithms

A 50% Power Reduction in H.264/AVC HDTV Video Decoder LSI by Dynamic Voltage Scaling in Elastic Pipeline

Kentaro KAWAKAMI[†a)], **Jun TAKEMURA**[††], ***Student Members***, **Mitsuhiko KURODA**[†††], **Hiroshi KAWAGUCHI**[†††], ***Nonmembers***, ***and*** **Masahiko YOSHIMOTO**[†††], ***Member***

SUMMARY We propose an elastic pipeline that can apply dynamic voltage scaling (DVS) to hardwired logic circuits. In order to demonstrate its feasibility, a hardwired H.264/AVC HDTV decoder is designed as a real-time application. An entropy decoding process is divided into context-based adaptive binary arithmetic coding (CABAC) and syntax element decoding (SED), which has advantages of smoothing workload for CABAC and keeping efficiency of the elastic pipeline. An operating frequency and supply voltage are dynamically modulated every slot depending on workload of H.264 decoding to minimize power. We optimize the number of slots per frame to enhance power reduction. The proposed decoder achieves a power reduction of 50% in a 90-nm process technology, compared to the conventional clock-gating scheme.

key words: *H.264/AVC, DVS (dynamic voltage scaling), elastic pipeline, low power, divided entropy decoder*

1. Introduction

Dynamic voltage scaling (DVS) is an effective technique for general-purpose processors to achieve both a high peak performance and low average power [1], [2]. Because an operating frequency in a CMOS digital circuit, f, is formulated as follows [3], an LSI performs faster as a supply voltage, V_{dd}, becomes higher.

$$f \propto \frac{(V_{dd} - V_{th})^{\alpha}}{V_{dd}}, \tag{1}$$

where α represents a velocity saturation index in a short-channel MOSFET, and is about 1.6 in a 90-nm process technology used in this study. V_{th} is a threshold voltage of the MOSFET. In addition, since a power in an LSI, P, is expressed as follows, the supply voltage quadratically increases the power.

$$P \propto f \cdot V_{dd}^2. \tag{2}$$

Figure 1 illustrates relations between an operating frequency and power. In DVS, since a supply voltage can be optimized if an operating frequency does not need to be the maximum, the power can be dramatically decreased by the low operating frequency and low supply voltage. If an operating frequency is set low, in other words, if a required performance is low, DVS adaptively lowers both the operating frequency and supply voltage in order to reduce the power. If the maximum performance is instantaneously needed, the highest supply voltage and highest operating frequency are utilized so that DVS can accommodate the peak performance.

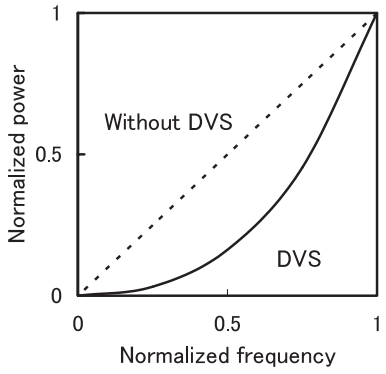


Fig. 1 Relations between power and frequency in DVS.

Because (1) and (2) are approved for every CMOS digital circuit, applications of DVS is not limited to general-purpose processors, and thus it is theoretically applicable to hardwired logic circuits. However, there is few example of applying DVS to hardwired logic circuits for real-time processing. This reason is explained in this way. A dedicated hardware is often built with a pipeline architecture for high throughput, which requires a certain number of clock cycles to process the worst-case workload. Supposing the worst-case workload, each pipeline stage is configured, and every pipeline operation simultaneously starts at the beginning of an allocated period. Consequently, a required operating frequency is uniquely fixed. There is no room to change the operating frequency, and DVS is not applicable.

In order to apply DVS to hardwired logic circuits, we propose an elastic pipeline architecture. Since this architecture can save the number of cycles in a pipeline operation depending on characteristics of input data, it enables hardwired logic circuits to employ DVS, and thus achieves lower power. As a design example, we propose a novel H.264/AVC high-definition television (hereafter, H.264 HDTV) decoder. The power can lower to 50% of

Manuscript received March 17, 2006.
Manuscript revised June 13, 2006.
Final manuscript received August 1, 2006.

[†]The author is with the Graduate School of Science and Technology, Kobe University, Kobe-shi, 657-8501 Japan.
[††]The author is with the Graduate School of Natural Science and Technology, Kanazawa University, Kanazawa-shi, 920-1192 Japan.
[†††]The authors are with the Department of Computer and Systems Engineering, Kobe University, Kobe-shi, 657-8501 Japan.
a) E-mail: kawakami@cs28.cs.kobe-u.ac.jp

DOI: 10.1093/ietfec/e89–a.12.3642

the conventional pipeline architecture with gated clock.

The following sections are configured as follows. The elastic pipeline architecture is proposed in Sect. 2. The reason why the number of execution cycles in H.264 decoding is varies in pipelined stages, is mentioned in Sect. 3. In Sect. 4, the novel H.264 HDTV decoder architecture is described. The effectiveness of the proposed architecture is verified by designing the H.264 HDTV hardwired decoder in Sect. 5. Finally, the findings of this paper are summarized in Sect. 6.

2. Elastic Pipeline Architecture

2.1 Conventional Pipeline Architecture

Figure 2 shows a timing diagram of the conventional pipeline architecture. The worst-case execution cycles (WCEC) signify the maximum number of execution cycles required for one pipeline process. A grayed area in the figure represents processing cycles in which a stage is running with a datum. Since all pipelined processes in the conventional architecture start at the beginning of the allocated WCEC supposing the worst-case workload, all the stages have to idle until the next WCEC even if they finish earlier. The clock-gating technique may be utilized to cut off the unnecessary power caused by the idle cycles, but the power save is limited.

With the help of Fig. 2, let us consider a case that M pipeline stages process N data. In the conventional pipeline architecture, the number of execution cycles to complete the N-th datum in the M-th pipeline stage, EC_{conv}, is expressed as follows.

$$EC_{\text{conv}} = (M + N - 2) \times WCEC + W_{M,N}, \tag{3}$$

where $W_{M,N}$ ($\leq WCEC$) indicates the processing cycle for the N-th datum in the M-th stage. As N is larger, EC_{conv} becomes closer to $N \times WCEC$, which means that there is not much room to apply DVS. Namely in the conventional pipeline architecture, only the execution cycle variation of $W_{M,N}$ can be exploited.

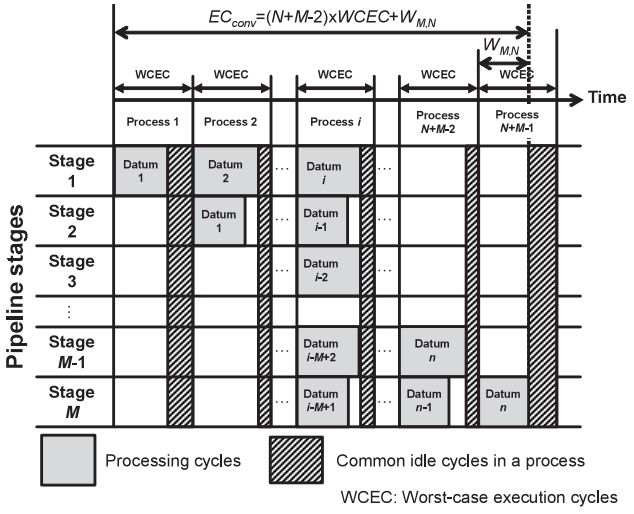


Fig. 2 Timing diagram of the conventional pipeline.

2.2 Elastic Pipeline Architecture

Figures 3(a) and (b) illustrate a concept and timing diagrams of the proposed elastic pipeline architecture, respectively. A stage in the elastic pipeline sends a process completion signal to a pipeline controller once its process is completed. If all the completion signals arrive at the pipeline controller, it sends back start signals to the pipeline stages. After receiving it, each stage turns off the process completion signal, and then starts the next pipeline process again. In this manner, the elastic pipeline architecture shortens the number of execution cycles as many as possible, and save its execution cycles. As illustrated in Fig. 3(b), the elastic pipeline architecture requires less number of cycles than the conventional one since the common idle cycles are eliminated (compare with Fig. 2).

Eventually, N data processed by the M-stage elastic pipeline require the following execution cycles, EC_{prop}.

$$EC_{\text{prop}} = \sum_{i=1}^{M+N-1} \max(W_{1,i}, W_{2,i-1}, \cdots, W_{M,i-M+1}), \tag{4}$$

where $W_{p,q}$ represents the number of execution cycles for

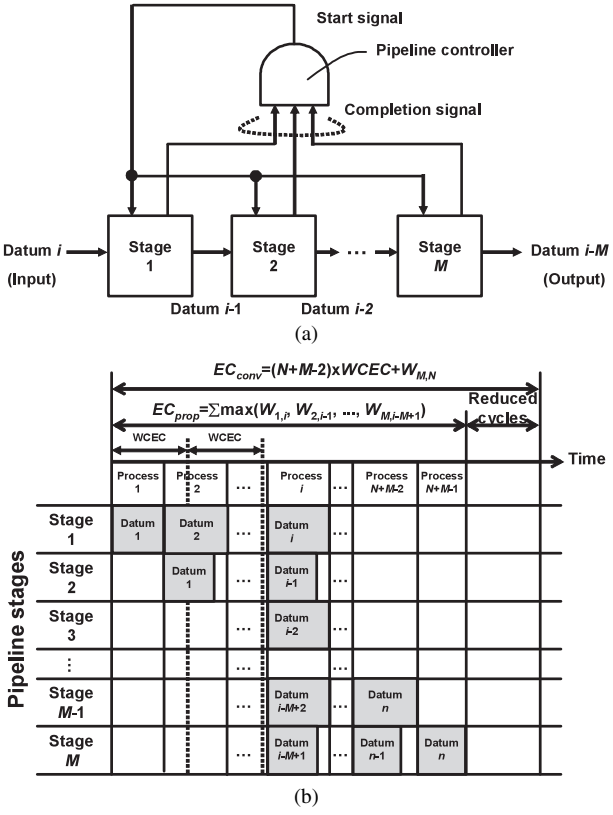


Fig. 3 The proposed elastic pipeline architecture, (a) concept, and (b) timing diagrams.

the q-th datum processed by the p-th pipeline stage. If either $q < 1$ or $q > N$, then $W_{p,q}$=0.

2.3 Design Conditions in H.264 HDTV Decoder

We briefly estimate the numbers of execution cycles in the conventional and proposed elastic pipelines architectures. In this paper, an H.264 [4] HDTV decoder LSI is implemented as a design example. We assume conditions of the design as follows.

- The decoder is built as a macroblock (MB) pipeline since image data are processed on an MB-by-MB basis in H.264. An MB is comprised of 16×16 pixels.
- The MB pipeline has six pipelined stages.
- The decoder operates at 108 MHz, which is reasonable compared to the previous work [5].
- 244,800 MBs are processed per second when the decoder handles an image sequence at 30 frame/s. As a result, 440 cycles are allowed in an MB pipeline process, in which the worst-case data can be processed.
- So that, each stage is designed to process the worst-case MB in almost 440 cycles.
- The number of processing cycles is a half of the worst case (220 cycles).

Since the H.264 HDTV has 8,160 MBs, the conventional MB pipeline needs 3.6×10^6 (= $(6 + 8{,}160 - 2) \times 440 + 220$) cycles per frame. On the other hand, the number is to 1.8×10^6 (= $(6 + 8{,}160 - 1) \times 220$) in the elastic pipeline thanks to the half average workload. This indicates that the elastic pipeline can reduce the number of execution cycles to a half of the conventional MB pipeline, and lower an operating frequency under these conditions.

3. Execution Cycle Variations in H.264 Decoding

Figure 4 shows a block diagram of a typical H.264 decoder. In the following six functional blocks, the numbers of execution cycles are varied, which can be exploited with the proposed elastic pipeline architecture.

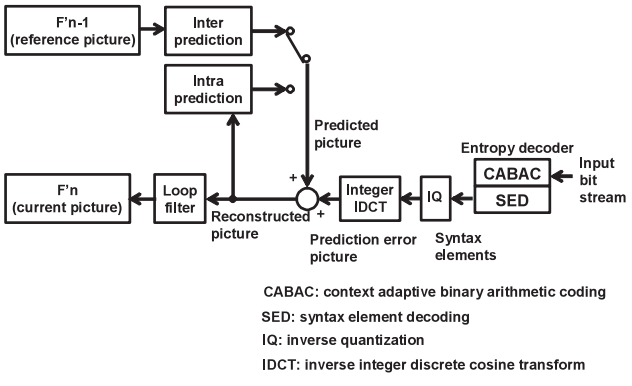


Fig. 4 Block diagram of typical H.264 decoder.

3.1 Entropy Decoder

An entropy decoder transforms an input bit stream to syntax elements, which include information on MB types, intra prediction modes, motion vectors, coded block patterns, coefficients, and so on. If context-based adaptive binary arithmetic coding (CABAC) is selected by a encoder, bit streams are expanded by a CABAC decoder at first, and then syntax elements are decoded by a syntax element decoder (SED). Some MBs have more than a hundred of syntax elements, but other MBs have only one syntax element if they were encoded as skip MBs. Consequently, the number of execution cycles required for the entropy decoders depends on how many syntax elements an MB has.

3.2 IQ and IDCT

According to coefficients decoded by the SED, an inverse quantization (IQ) and inverse integer discrete cosine transform (IDCT) generate a prediction error picture. The coefficients are structured in a 4×4 matrix form, and thus the IQ and IDCT basically carry out 4×4 matrix operations. An MB has 24 matrixes comprised of 4×4 coefficients. A 4×4 matrix is sometimes a zero matrix and in this case, the matrix operations in the IQ and IDCT can be cancelled. Hence, the number of execution cycles for the IQ and IDCT is varied depending on how many non-zero matrices an MB includes.

3.3 Intra Prediction

An intra prediction generates a predicted picture with neighbor MBs that were beforehand processed, according to intra prediction modes. H.264 provides 13 types of intra prediction modes, which require different calculations. For example, the Intra_4 $\times$ 4_Horizontal and Intra_4 $\times$ 4_Vertical prediction modes just copy pixel data without calculation. These prediction modes do not make any calculation. On the other hand, the Intra_4 $\times$ 4_DC prediction mode requires a 8-tap filter operation. Therefore, the number of execution cycles for the intra prediction depends on which intra prediction modes are chosen in an MB.

3.4 Inter Prediction

An inter prediction generates a predicted picture from motion vectors and a reference picture which was previously constructed as a current picture. H.264 provides an integer-pixel, half-pixel, and quarter-pixel precision motion vectors. If a motion vector has an integer-pixel precision, a portion of the reference picture that is pointed by the integer-pixel precision motion vector is simply utilized for the predicted picture. Else if a motion vector has a half-pixel precision, a 6-tap filter is required in order to generate a half-pixel precision picture from the integer-pixel precision reference picture. In the other case that a motion vector is a quarter-pixel

precision, another 2-tap filter is needed in order to generate a quarter-pixel precision picture from the half-pixel precision picture. In this way, the number of execution cycles for the inter prediction depends on a pixel precision of motion vectors.

3.5 Loop Filter

A reconstructed picture is generated by adding a prediction error picture to a predicted picture. The reconstructed picture needs to pass through a loop filter before it is output as a current picture. The loop filter smoothes pixels on sub-block boundaries. Although an MB has 48 sub-block edges to be filtered, all the edges do not need to be filtered. Each edge is adaptively filtered depending on an MB type and pixel values, which influences the number of execution cycles required for the loop filter.

4. An Architecture Design of H.264 Decoder LSI

This section proposes a novel H.264 decoder LSI on which the elastic pipeline architecture is implemented. A target specification is H.264 Main Profile Level 4 with 108-MHz operating frequency. DVS properly works since the proposed architecture saves the number of execution cycles thanks to the elastic pipeline.

Figure 5 illustrates the block diagram of the propose H.264 HDTV decoder, which supports Main Profile Level 4. Since this specification requires a 96-Mb memory as a buffer for decoded pictures, the proposed architecture uses an external DRAM.

4.1 Double Buffer Scheme

Some buffers are located between functional blocks. All the buffers are two-bank SRAMs for double buffering, except a RAM for intra prediction that is not frequently accessed. For instance, while the IQ/IDCT block is writing a prediction error picture into Bank 0, the prediction error adder is reading the previous prediction error picture which was written by the IQ/IDCT block. In the next pipeline process, Bank 0 and Bank 1 change places. Since each MB is encoded as an inter MB or intra MB, either the inter prediction or intra prediction operates in every pipeline process.

The IQ/IDCT, inter prediction and intra prediction have no data dependency each other, therefore they can be parallelized. A predicted and prediction error pictures can be added at the prediction error adder. Finally, the loop filter smoothes the reconstructed picture.

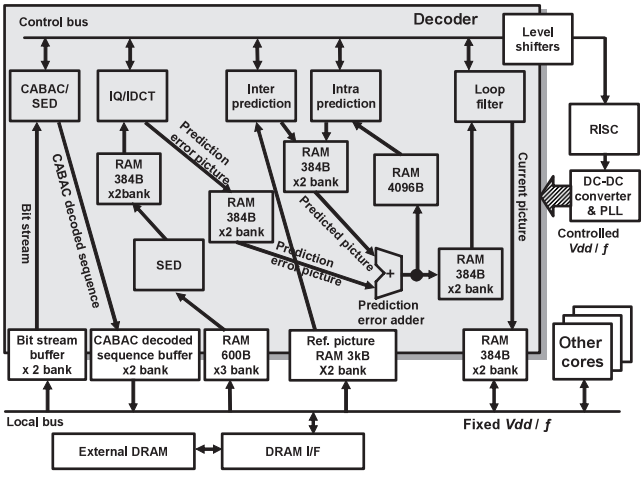


Fig. 5 Block diagram of the proposed H.264 decoder.

4.2 Divided Entropy Decoder

As entropy decoding, H.264 Main Profile provides two methods, CABAC and context-based adaptive variable length coding (CAVLC). Because CABAC achieves approximately 15% higher coding efficiency than CAVLC [6], this paper considers only CABAC as entropy decoding.

In case of CABAC, each syntax element is generated from bit streams through the following three steps.

1. A context is selected by an SED according to a kind of objective syntax element.
2. By utilizing the selected context, a CABAC decoder generates CABAC decoded binary sequence from received bit streams. Since this sequence is compressed by CABAC, the maximum volumes of CABAC decoded binary sequences may turns out to be 1.33 times larger than those of bit streams [4].
3. The objective syntax element is generated by the SED from the binary sequence.

The H.264 standard limits the maximum number of bits contained in each data unit as shown in Table 1. An MB can include 3,200 bits, however, the total number of bits in a frame should be less then 6.5 Mbits ($< 3,200\,\text{bits/MB} \times 8160\,\text{MBs/frame}$). This means that an MB has 797 bits on average. In the same way, the total number of bits per second is limited to 20 Mbits in the bit stream level. In this case, an MB contains only 82 bits on average, however, cycle reduction is difficult because a CABAC decode has strong sequential process dependence.

The table also shows the required operating frequency. It takes three cycles to process one bit in the CABAC decoder [7] if a dedicated hardware is designed. In a case of building a CABAC decoder onto an MB pipeline, 2.35 GHz is required, while in a case of synchronizing a CABAC de-

Table 1 Maximum number of bits in each data unit.

Maximum number of bits per unit	Unit	Required operating frequency	Number of bits per MB
3,200 bits	MB	2.35GHz (=3,200 bits/MB ×3 cycles/bit×8160 MBs/frame ×30 frames/s)	3,200 bits
6.5 Mbits *	Frame	585MHz (=6.5 Mbits/frame ×3 cycles/bit×30 frames/s)	797 bits
20 Mbits **	Second	60MHz (=20 Mbits/s×3 cycles/bit)	82 bits

* In case of 30 frames/s@Level 4
**In case of Level 4

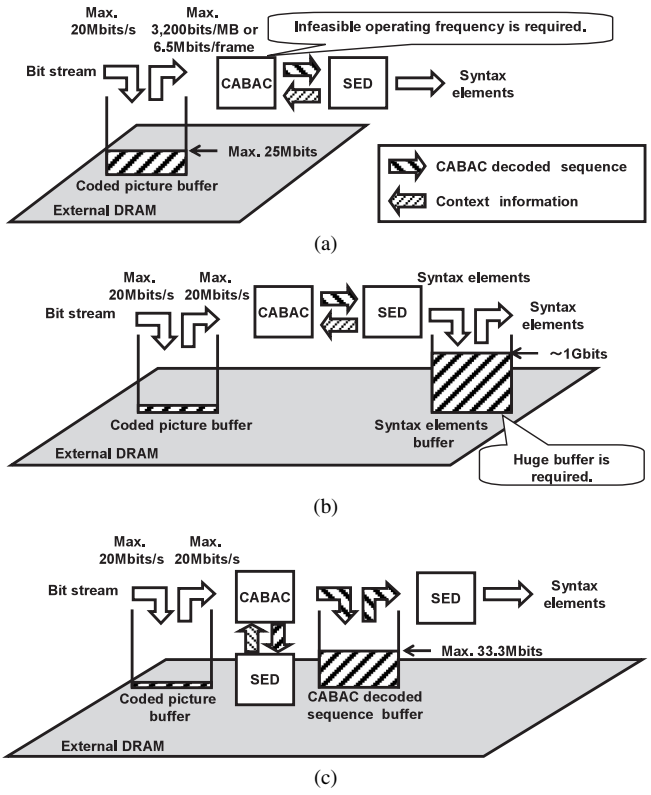


Fig. 6 Entropy decoder configuration, (a) the straightforward configuration, (b) the buffering configuration, and (c) the proposed configuration.

coder with a frame, 585 MHz is needed but this frequency is still infeasible (Fig. 6(a)). Thus for a lower frequency, the workload of the CABAC decoder must be smoothed over a time scale of the bit stream (on the second time scale), in which case the operating frequency of the CABAC decoder turns out to 60 MHz and can be reduced to less than 108 MHz.

Figure 6(b) illustrates a straightforward buffering scheme based on the second time scale. This scheme demands a huge size of memory. H.264 Level 4 requests decoders to prepare 25-Mbits buffer in which received bit stream is stored. In case that a bit rate is 20 Mbps, 25-Mbits bit stream contains coefficients data of 1.25-seconds. Since a MB required 384 bytes of coefficients data, the required buffer size for these coefficients reaches 936 Mbits (= 384 bytes/MB × 8160 MB/frame × 30 frame/second × 1.25 seconds). In reality, an MB requires some header information such as motion vectors, MB partitions, intra prediction modes and so on, hence the total buffer size reaches about 1 Gbits.

Figure 6(c) shows the proposed divided entropy decoder scheme to achieve both of a lower operating frequency and smaller buffer size. The proposed scheme shown in Fig. 6(c) divides the entropy decoding into CABAC decoding and SED.

A bit stream is soon decoded by the CABAC decoder and SED once it is received, and then CABAC decoded sequences are stored in a sequence buffer prepared in an external DRAM. Since the CABAC decoder processes bit streams at an average rate of 82 bits per 440 cycles, the infeasible high operating frequencies or huge buffer can be avoided.

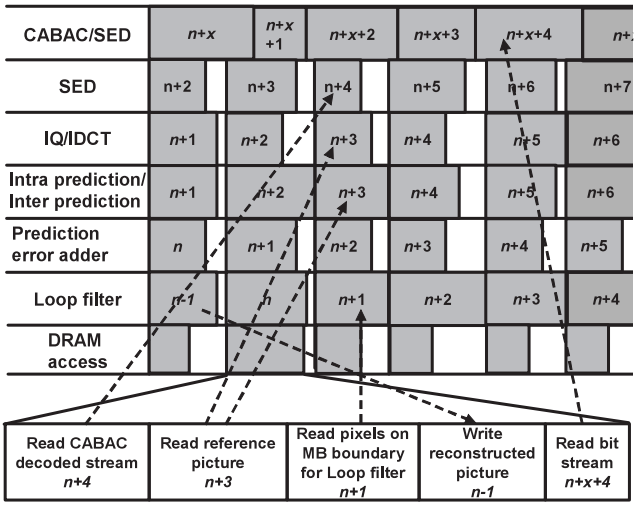


Fig. 7 Timing diagram of the proposed elastically-pipelined decoder.

The proposed scheme requires a smaller buffer memory for storing CABAC decoded sequences, which are generated from bit streams by the CABAC decoder. The size of this buffer can be reduced to 33.3 Mbits (= 25 Mbits × 1.33), since the compression rate of CABAC is limited to 1.33.

The proposed scheme brings a side effect. Since CABAC decoder is required to decode 82 bits (=20 Mbps/30 frame/s/8160 MB/frame), CABAC can be embedded into the pipeline even if 1-bit CABAC decoding consumes three cycles.

4.3 Elastic Pipeline

Figure 7 illustrates the timing diagram of the proposed elastically-pipelined decoder. The DRAM bandwidth needs to be shared by functional blocks, and thus it is important to consider data amount from/to the external DRAM. If the data amount is exceeded, the pipeline has to stall. An external DRAM is accessed by the CABAC decoder, SED, IQ/IDCT, the intra/inter prediction, the prediction error adder and the loop filter.

The CABAC decoder reads a bit stream from the external DRAM and writes back a binary sequence to the DRAM. SED reads a CABAC decoded sequence from the DRAM to decode syntax elements. The inter prediction reads reference pictures pointed by motion vectors from the DRAM. The loop filter writes back a filtered picture to the DRAM. The required traffic to decode one MB is 41 kbits in the worst case, and thereby the bandwidth between the decoder and DRAM is estimated at 10.0 Gbps. In actual SoC designs, a wider bandwidth is necessary since other cores such as audio codec and post video processing request certain traffic.

If all the pipeline stages finish within the WCEC, the

elastic pipeline reduces processing cycles. As mentioned in Sect. 2.3, 440 cycles are available for an MB process at an operating frequency of 108 MHz. Since the function of the prediction error adder is 384-time additions of simple 8-bit data and clipping, it would always need 384 cycles and destruct elastic pipelining. However, this value is reduced to a quarter (96 cycles) by a four-degree parallelism. The area overhead is small because the 8-bit addition and clipping operation are fortunately not much.

4.4 Interface Problem

DVS is implemented to the proposed elastically-pipelined decoder. However, a supply voltage and operating frequency outside the decoder should not be controlled since it is preferable that the DRAM interface operates at the fixed supply voltage and operating frequency for compatibility among other hardware cores. In order to solve the interface problem between the DVS domain and outside, we locate two-bank and three-bank SRAMs there. Figure 8 demonstrates how to control the mutli-bank SRAMs. It is possible to change a supply voltage in a 1024-bits SRAM within on clock cycle [8]. Therefore, even in the 3 kB, the transition time overhead to change the supply voltage is supposed to be a several cycles, which is quite less than those allocated to one pipeline process.

Now, the supply voltage and operating frequency in Bank 0 are controlled in synchronization with those of the decoder for instance. While the decoder reads/writes data from/to Bank 0, the supply voltage and operating frequency in Bank 1 are as same as those of the local bus and the external DRAM. The external DRAM can read/write data from/to Bank 1. In the next pipeline process, the roles of these are alternated. Thereby, the dynamic controls of the supply voltage and operating frequency in the decoder do not give any influence to the outside.

The SRAM in which the binary sequences are stored is also organized as a three-bank SRAM. Since it is difficult to know how many binary sequences are required to decode one MB, the binary sequences for one MB often straddle two banks. In this case, a third bank is demanded to store the CABAC decoded stream from the CABAC decoded buffer for the next MB. Since each bank in the three-bank SRAMs also cyclically alternate its role in every pipeline process, the proposed architecture never disturbs the outside of the decoder core.

5. Implementation & Performance Estimation

5.1 Cycle Reduction by Elastic Pipeline

In order to verify the execution cycle reduction by the elastic pipeline, we designed every functional block with the H.264 reference software JM 9.6 [9], and carried out cycle simulation with Celoxica Handel-C [10]. The decoding conditions are listed in Table 2.

Figure 9(a) shows the fluctuation of the execution cycles in the pipeline process. The normalized local bus traffic is also considered as well, when a clock frequency of the external DRAM is varied. The local bus width is assumed to be 16 bits. The horizontal axis shows the pipeline process number. The vertical one is an execution cycle ratio normalized by the WCEC, and normalized traffic on a local bus. This figure indicates the following points.

- The local bus traffic is varies in synchronization with the execution cycle of the decoder.
- If a low-frequency DRAM is used, the workload of the local bus becomes larger than that of the decoder. This hinders an execution cycle reduction of the decoder since the DRAM interface occupies multi-bank SRAMs in the decoder for a longer time than the decoder's execution time.
- If the bus frequency is 400 MHz, the workload of the local bus becomes as same as that of the decoder.
- The average execution cycles in the decoder are approximately 50% of the WCEC.
- Thus, if a high frequency DRAM is utilized, an operating frequency of the decoder may be halved, and power reduction by DVS can be expected.

Figure 9(b) shows the cases of the normalized execution cycles per frame in the decoder, and traffic of the local bus per frame. Since a B picture requires a larger volume of data for inter prediction, the bus traffic becomes busier

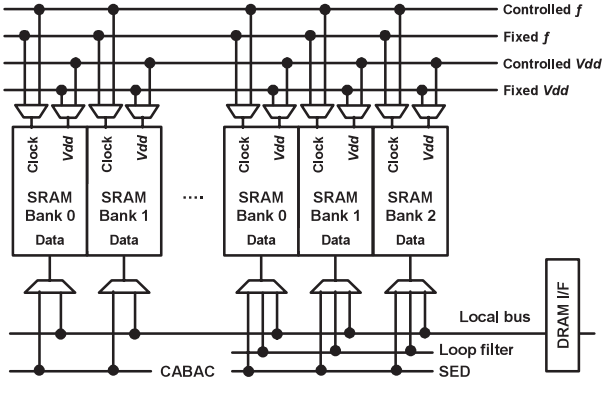


Fig. 8 Multi-bank SRAM for interface with fixed-voltage local bus.

Table 2 Simulation condition.

Profile / Level	Main profile / 4
Frame rate	30frame/s
Bit rate	10Mbps
Number of reference frame	2
Motion estimation algorithm	UMHexagon search
Search range	±64 x ±64
Entropy coding method	CABAC
JM version	9.6

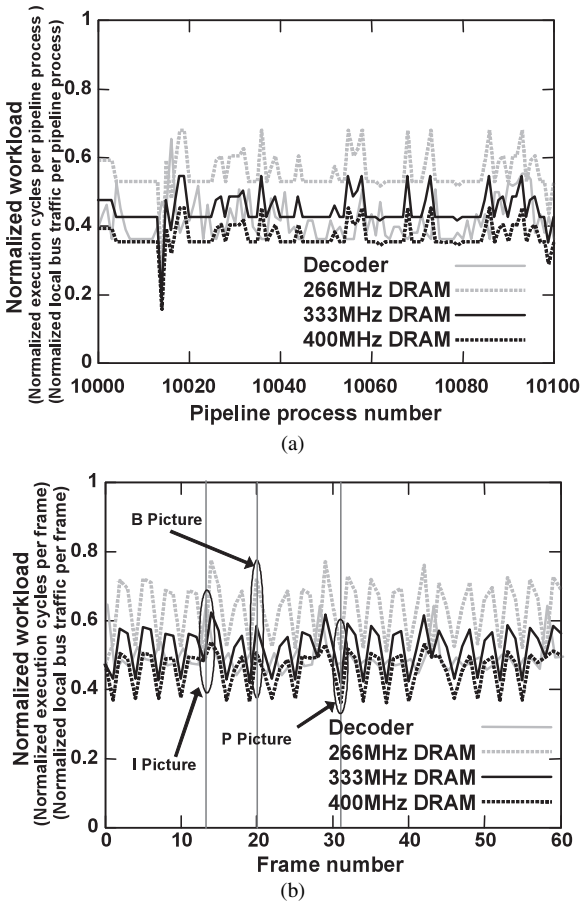


Fig. 9 Cycle reduction of proposed H.264/AVC decoder architecture, (a) cycle reduction in each pipeline process, and (b) cycle reduction in each frame.

when the 266-MHz DRAM is used. In this case, the elastic pipeline wastes common idle cycles until the high-volume data arrival from the DRAM. It degrades the performance of the elastic pipeline.

5.2 Power Estimation

For DVS, a supply voltage and operating frequency are changed by a feedback mechanism as shown in Fig. 10 [11]. A frame is divided into slots that have same amount of data to be processed. A set of MBs are assigned to a slot.

If there is a margin to make an operating frequency lower, the feedback mechanism selects the lower frequency on a slot-by-slot basis.

The feedback mechanism is explained with the help of Fig. 10. First slot in a frame is always processed at the maximum frequency, since there is no margin to change operating frequency at the begging of a frame. The elastic pipeline reduces the number of execution cycles in pipeline processes. Therefore, the first slot is potentially completed earlier. Then, assume that the second slot also reduces its execution cycles. Now, the third slot acquires a time margin, ΔH. Even considering a voltage/frequency transition time, the third slot has a time twice as long as T_{slot} (processing time for a slot), which allows the third slot to be processed at a half of $f_{\max}$. Note that a real-time operation is guaranteed in this feedback mechanism.

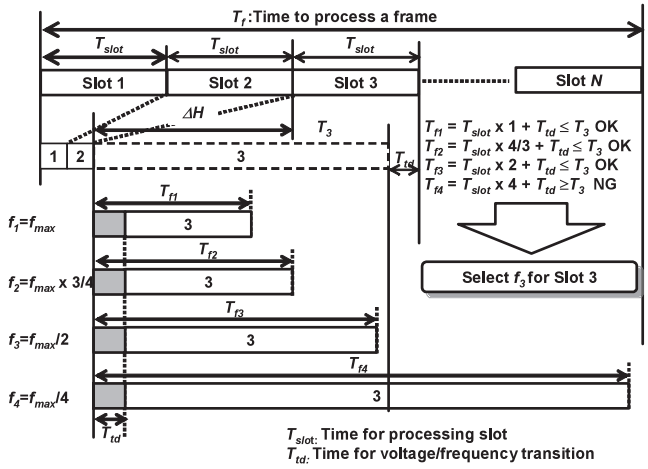


Fig. 10 Feedback control mechanism.

Table 3 Operating frequency-supply voltage dependency and assumed sets.

Mode	Operating frequency (MHz)	Supply voltage (V)	4 sets	2 sets
4	f_4=108	1.00	×	×
3	f_3=81	0.85	×	
2	f_2=54	0.70	×	×
1	f_1=27	0.55	×	

A power reduction factor depends on the number of slots prepared. If there are few slots, there are few chances to lower an operating frequency and supply voltage, which in turn increases power. Alternatively, if there are many slots, there are many chances to lower them. However, it consumes much transition times to change the operating frequency and supply voltage, which makes substantial processing time shorter, and increases the power. Namely, there is the optimum number of slots per frame.

Every functional block is designed to verify the power reduction of the proposed elastic pipeline architecture. The total transistor count and capacity of SRAMs are obtained to be 2,401,316 transistors and 117 kbits, respectively. Table 3 shows the relations between the operating frequencies and supply voltages of our design, which are obtained by SPICE simulation.

The power reduction ratio is calculated by (5).

$$r = P_{\text{prop}}/P_{\text{conv}}, \tag{5}$$

where

$$P_{\text{prop}}, P_{\text{conv}} = \sum_{i=1}^{Num} \sum_{j} \sum_{k} (P_{j,k} \cdot T_{i,j,k} + P'_{j,k} \cdot T'_{i,j,k}).$$

The suffix i, j, k represents frame number, operating modes,

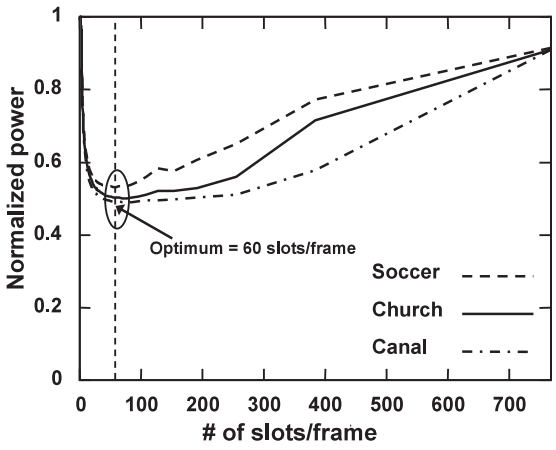


Fig. 11 The number of slots vs. power.

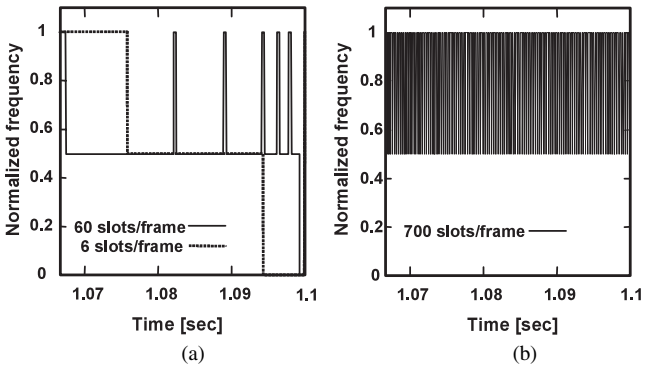


Fig. 12 Operating frequency transitions, (a) the adequate number of slots achieves low power, but (b) the excessive number of slots degrades power reduction.

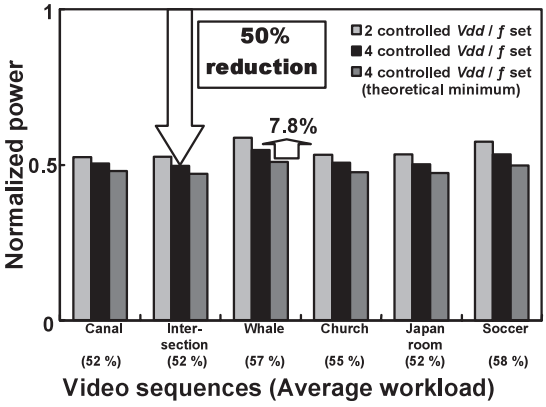


Fig. 13 Power reduction ratio.

and stages in the pipeline, respectively. Since the proposed architecture has six functional stages, k is 1, 2, ..., 6. $P_{j,k}$ represents total power consumption of k-th stage in j-th operating mode. $P'_{j,k}$, represents leakage power consumption of k-th stage in j-th operating mode, which means the power of gated-clock periods. $T_{i,j,k}$, stands for how long k-th stage operates at j-th operating mode in i-th frame. $T'_{i,j,k}$ denotes how long the clock signal of k-th stage is gated at j-th operating mode in i-th frame. Since the conventional architecture has only one mode, mode 4 in Table 3, j takes only four in P_{conv}.

In this study, the transition time is assumed to be 50 μs. Since logic parts occupies a larger area than the interface SRAMs, the logic parts requires longer time for voltage transitions. It is also assumed that the numbers of prepared frequency/voltage sets are two ($f_{\max}$ and $f_{\max}/2$) and four ($f_{\max}$, $3f_{\max}/4$, $f_{\max}/2$ and $f_{\max}/4$).

Figure 11 shows the power reduction dependency on the number of slots per frame. The optimum number of slots per frame is obtained as 60 from the result of seven video sequences. If the number of slots becomes smaller than the optimum number, the power reduction becomes drastically worsen. Alternatively, if the number of slots becomes larger than the optimum number, the power reduction becomes gradually worsen.

Figure 12 concretely shows behaviors of frequency transition when the number of slots per frame is varied. The number of frequency transitions increase according to the number of slots per frame. In Fig. 12(b), a terrible situation happens. Because many frequency transitions are occurred, the maximum frequency periods occupies a half of the total processing time, in which case power is not reduced very much.

Figure 13 shows the power reductions in video sequences, when the number of slots per frame is set to 60. It can be seen that the maximum power reduction of 50% is achieved in the sequence "Intersection." 45% and 49% power reductions are obtained on average when the two and four frequency/voltage sets are available, respectively.

A theorem for the power-minimum frequency/voltage scheduling algorithm under DVS environment is proposed in [12]. If the number of required cycles is preliminary known, this theorem uniquely tells which operating frequency should be used and how long time the decoder should operate at each frequency. The minimum power derived by this theorem is also shown in Fig. 13. The differences of power reductions are less than 7.8% between the feedback mechanism and the power-minimum scheduling. We conclude that the feedback mechanism achieves enough low power without knowledge of required cycles.

6. Conclusion

We proposed the elastic pipeline architecture with which DVS can be applied to hardwired logic circuits, and designed a hardwired H.264/AVC HDTV decoder to verify its feasibility. The proposed decoder has the divided entropy decoder scheme and multi-bank SRAMs between the functional blocks. This structure solves the interface problem between the decoder and outside where a supply voltage and frequency are different. The proposed decoder architecture is familiar with other hardware cores such as audio codec, post video processing, and thus can coexist with them on a chip. Compared with the conventional pipeline architecture with the clock gating scheme, a power reduction of 50% is

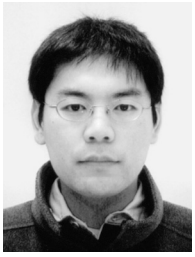
possible.

Acknowledgements

This work is supported by VLSI Design and Education Center (VDEC), the University of Tokyo with the collaboration with Celoxica Ltd.

References

[1] K.J. Nowka, G.D. Carpenter, E.W. MacDonald, H.C. Ngo, B.C. Brock, K.I. Ishii, T.Y. Nguyen, and J.L. Burns, "A 32-bit PowerPC system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling," IEEE J. Solid-State Circuits, vol.37, no.11, pp.1441–1447, Nov. 2002.
[2] H. Ohira, K. Kawakami, M. Kanamori, Y. Morita, M. Miyama, and M. Yoshimoto, "A feed-forward dynamic voltage control algorithm for low power mpeg4 on multi-regulated voltage CPU," IEICE Trans. Electron., vol.E87-C, no.4, pp.3290–3297, April 2004.
[3] 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, April 1990.
[4] Joint Video Team (JVT) of ISO/IEC MPEG&ITU-T VCEG, ISO/IEC 14496-10, May 2003.
[5] H. Iwasaki, J. Naganuma, Y. Nakajima, Y. Tashiro, K. Nakamura, T. Yoshitome, T. Onishi, M. Ikeda, T. Izuoka, and M. Endo, "A 1.1 W single-chip MPEG-2 HDTV codec LSI for embedding in consumer-oriented mobile codec systems," Proc. IEEE Custom Integrated Circuits Conference, pp.177–180, Sept. 2003.
[6] S. Saponara, C. Blanch, K. Denolf, and J. Bormans, "The JVT advanced video coding standard: Complexity and performance analysis on a tool by tool basis," IEEE Packet Video 2003.
[7] J. Chen, C. Chang, and Y. Lin, "A hardware accelerator for context-based adaptive binary arithmetic decoding in H.264/AVC," IEEE International Symposium on Circuits and Systems, vol.5, pp.4525–4528, May 2005.
[8] Y. Morita, H. Fujiwara, H. Noguchi, K. Kawakami, J. Miyakoshi, S. Mikami, K. Nii, H. Kawaguchi, and M. Yoshimoto, "A Vth-variation-tolerant SRAM with 0.3-V minimum operation voltage for memory-rich SoC under DVS environment," Digest of Technical Papers of 2006 Symposium on VLSI Circuits, pp.16–17, June 2006.
[9] H.264/AVC reference software, http://iphome.hhi.de/suehring/tml/
[10] http://www.celoxica.com/
[11] H. Kawaguchi, Y. Shin, and T. Sakurai, "μITRON-LP: Power-conscious real-time OS based on cooperative voltage scaling for multimedia applications," IEEE Trans. Multimed., vol.7, no.1, pp.67–74, Feb. 2005.
[12] K. Kawakami, M. Kanamori, Y. Morita, J. Takemura, M. Miyama, and M. Yoshimoto, "Power-minimum frequency/voltage cooperative management method for VLSI processor in leakage-dominant technology era," IEICE Trans. Fundamentals, vol.E88-A, no.12, pp.3290–3297, Dec. 2005.



Kentaro Kawakami received the B.E. degree in electrical and information engineering and the M.E. degree in electronic and information system from Kanazawa University, Ishikawa, Japan, in 2002 and 2004, respectively. He transferred his school from Kanazawa University to Kobe University in 2005. He is currently a Ph.D. candidate at Kobe University, Kobe, Japan. His research interests include low power circuits, motion video codec and LSI design methodology.

Jun Takemura entered Information and Systems Engineering at Kanazawa University, Ishikawa, Japan in 2000. He received the B.E. degree in information and systems engineering and the M.E. degree in electronic and information system from Kanazawa University, Ishikawa, Japan, in 2004 and 2006, respectively. He joined Renesas Technology Corporation, Tokyo, Japan, in 2006. His research interests include low-power multimedia VLSI designs.

Mitsuhiko Kuroda received the B.E. degree in computer and systems engineering from Kobe University, Kobe, Japan, in 2006. He is currently working in the M.E. course at the same university. His research interests include low power circuits, low power motion picture codec LSI.

Hiroshi Kawaguchi received the B.E. and M.E. degrees in electronic engineering from Chiba University, Chiba, Japan, in 1991 and 1993, respectively. He received the Ph.D. degree in engineering from the University of Tokyo, Tokyo, Japan, in 2006. He joined Konami Corporation, Kobe, Japan, in 1993, where he developed arcade entertainment systems. He moved to the Institute of Industrial Science, the University of Tokyo, as a Technical Associate in 1996, and was appointed to be a Research Associate in 2003. Since 2005, he has been a Research Associate in the Department of Computer and Systems Engineering, Kobe University, Kobe, Japan. He is also a Collaborative Researcher in the Institute of Industrial Science, the University of Tokyo. He is a recipient of the IEEE ISSCC 2004 Takuo Sugano Award for Outstanding Far-East Paper. He has served as a program committee member for IEEE Symposium on Low-Power and High-Speed Chips (COOL Chips). He is a guest associate editor of IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences. His current research interests include low-power VLSI design, hardware design for wireless sensor network, and recognition processor. Dr. Kawaguchi is a member of the IEEE and ACM.

Masahiko Yoshimoto received the B.S. degree in electronic engineering from Nagoya Institute of Technology, Nagoya, Japan, in 1975, and the M.S. degree in electronic engineering from Nagoya University, Nagoya, Japan, in 1977. He received the Ph.D. degree in Electrical Engineering from Nagoya University, Nagoya, Japan in 1998. He joined the LSI Laboratory, Mitsubishi Electric Corp., Itami, Japan, in April 1977. From 1978 to 1983 he was engaged in the design of NMOS and CMOS static RAM including a 64 K full CMOS RAM with the world's first divided-word-line structure. From 1984, he was involved in research and development of multimedia ULSI systems for digital broadcasting and digital communication systems based on MPEG2 and MPEG4 Codec LSI core technology. Since 2000, he has been a Professor of the Dept. of Electrical and Electronic Systems Engineering at Kanazawa University, Japan. Since 2004, he has been a Professor of the Dept. of Computer and Systems Engineering at Kobe University, Japan. His current activity is focused on research and development of multimedia and ubiquitous media VLSI systems including an ultra-low-power image compression processor and a low power wireless interface circuit. He holds 70 registered patents. He served on the Program Committee of the IEEE International Solid State Circuit Conference from 1991 to 1993. In addition, he has served as a Guest Editor for special issues on Low-Power System LSI, IP, and Related Technologies of IEICE Transactions in 2004. He received the R&D100 awards from R&D Magazine for development of the DISP and development of a realtime MPEG2 video encoder chipset in 1990 and 1996, respectively.