Development of Low-power and Real-time VC-1/H.264/MPEG-4 Video Processing Hardware

Masaru Hase, Kazushi Akie, Masaki Nobori, Keisuke Matsumoto

System Design Dept. 3
System Design Div.
System Solution Business Group
Renesas Technology Corp.
Tokyo, Japan
Tel : +81-42-312-6592
Fax : +81-42-327-8623

Abstract - This paper covers a multi-functional hardware intellectual property (IP) for the encoding and decoding of digital moving pictures with low power consumption. The IP is mainly intended for mobile products such as cellular phones, digital still cameras (DSCs), and digital video cameras (DVCs). It includes VC-1 functionality for Internet content plus AVC (H.264) functionality for digital television broadcasting and MPEG-4 functionality for TV telephony, and is capable of processing D1-sized moving pictures (720 pixels by 480 lines) in real time at an operating frequency of 54 MHz. In addition, original algorithms employed in the IP reduce power consumption by up to 22%.

I Introduction

Fine hardware processes and enhanced processing capabilities have led to the broad adoption of mobile phones, DSCs, and DVCs. At the same time, the video functionality of mobile digital products for consumers has come to require more complex algorithms and processing capabilities for larger screens.

Accordingly, we have been developing a hardware IP that has multiple functionality in the encoding and decoding of digital moving pictures with low power consumption. The main target for the IP is the mobile-application field.

The features of the hardware IP are described below.

II. Real-Time Multi-Codec Processing

1. Video Codec 1 (VC-1): The IP supports the decoding requirements of this standard, which has been adopted by the Society of Motion Picture and Television Engineers, Inc. (SMPTE).

2. H.264 and MPEG-4: The IP supports the encoding and decoding requirements of these international standards in the field of digital moving-picture compression. In addition, it is capable of performing real-time processing equivalent to images at D1 resolution (720 pixels by 480 lines) and 30 fps.

The above three standards respectively correspond to fields that are indispensable to mobile phones, i.e. TV-phone communications (MPEG4), terrestrial digital broadcasting (H.264) based on the Association of Radio Industries and Businesses (ARIB), and the delivery of moving pictures over the Internet (VC-1); the hardware IP achieves real-time processing of all three.

III. Low Power Consumption

Power consumption is kept down by achieving real-time processing at the low clock frequency of 54 MHz. An original architecture and method of control have also been developed for a further reduction of power consumption during operation by up to 22%.

IV. Top-down Modeling

Top-down modeling improved accuracy in verifying the operation of the complex and large-scale algorithms.

II. Real-Time Multi-Codec Processing

A. Multi-Codec Hardware Architecture

To decide the architecture for implementation of the VC-1, H.264, and MPEG-4 standards, both all-hardware and software-type implementations were considered.

For example, software decoding of H.264-encoded QVGA sized moving pictures at the rate of 15 fps on a typical RISC engine or digital signal processor (DSP) requires an operating frequency above 200 MHz. Required operating frequencies for encoding at a given frame rate and resolution are two to four times as high. Handling of TV signals typically requires an operating frequency of 1.6 GHz. Power consumption makes this approach impracticable for mobile products such as mobile phones, DSCs, and DVCs.

Therefore, we employed an architecture that includes dedicated hardware to achieve the above functions for the IP.

Fig.1 shows the standards and video encoding and decoding techniques.

Different entropy coding techniques are used in VC-1, H.264, and MPEG-4. The frequency transforms and quantization techniques are also different. However, intra-frame prediction and inter-frame prediction in the three standards are almost the same. Therefore, attempting to implement all of the video codec techniques in a single hardware core leads to a very large number of gates. We thus implemented the gray-shaded techniques in the figure as

1-4244-0630-7/07/$20.00 ©2007 IEEE.

hardware shared with all three standards. These include intra-prediction, inter-prediction, and the deblocking filter, since this hardware is used with all three standards. This reduces the number of gates relative to a fully hardware implementation.

We prepared dedicated logic for the elements of fig.1. This approach (that is, sharing hardware and preparing dedicated logic) reduced operating frequencies by between one tenth and one twentieth (1/10 to 1/20) relative to processing by a software program on a general-purpose processor. However, this method requires a large number of supporting tool sets, which led to a large circuit scale. By sharing functions common to the AVC, MPEG-4, and so on in unified processing blocks, we can reduce the circuit scale. Specifically, we applied the several measures listed below to reduce the scale of circuits.

- Sharing processing required for both encoding and decoding

The VC-1, H.264, and MPEG-4 standards require that decoding needs to be performed at the same time as encoding. For this reason, redundant portions are shared between encoding and decoding processing in this hardware IP.

Encoding and decoding are generally considered to be inverse processes. However, some units execute essentially the same processing for encoding and decoding. This is the case for the unit performing image input/output, that is, the direct memory access controller (DMAC). Furthermore, the processing unit performing motion estimation (ME) at the time of encoding can also be exploited as the processing unit that performs motion compensation (MC) at the time of decoding. We have consolidated these units (those which perform the same processing in encoding and decoding) in this hardware IP.

- Effectively exploiting functions unique to each standard

Typically, the deblocking filter is not used in MPEG-4. We thus employed a structure in which the deblocking filter is selectable for post-processing (additional processing) of the moving-picture data. This allows the user to select or deselect the filter for use with MPEG-4.

- Reducing the amount of memory and the number of access operations by optimizing data paths

A detailed description is given in part C of this section (II).

Fig.2 is a block diagram of the implemented VC-1, H.264, and MPEG-4 hardware cores. The hardware core consists of sub-modules for rate control, entropy coding/decoding, frequency transformation, quantization, intra-prediction, motion estimation (ME), filtering, motion compensation (MC), and direct memory access (DMA).

B. Real-time processing

This hardware IP includes the following techniques for achieving low-frequency yet real-time processing.

1. Constraints imposed by the minimum unit of processing

For the individual modules in Fig. 2, we must consider the minimum unit of processing. The unit of processing is the macro-block, which is 16 pixels by 16 pixels. This reduces the capacities of the buffers required to transfer processing between the respective functional blocks relative to the case for processing in frame units.

The processing time required to ensure real-time processing is derived below.

2. Real-time processing cycle

The target image size is D1, that is, 720 by 480, and the frame rate is 30 fps. The target operating frequency is 54 MHz. In this condition, the number of macro-blocks to be processed per second is:

720 * 480 * 30/256 = 40500

The number of cycles available for processing one macro-block is thus:

54 MHz/40500 macro-blocks = 1333 cycles/macro-block

The upper bound of clock cycles in each MB processing is determined to be 1280 cycles. Thus, allowing a margin for such practical considerations as the processing latency of the processing pipeline and control from outside the IP, we selected 1280 as the maximum number of clock cycles for the processing of each macro-block.

That is, the hardware IP is basically designed so that all the processing sub-modules are capable of processing one macro-block of the image in 1280 or fewer cycles.

3. Dynamic Time Slot (DTS) System

Our goal was to achieve real-time processing at low operating frequencies. The focus is design of the pipeline processing where inputs and outputs are sequentially transferred through the functional blocks (e.g. decoding, frequency transform, and overlap).

Fig.3 is a timing chart for the various modules. The horizontal axis corresponds to time in units of 1280 clock cycles, i.e. pipeline slots. Once a module finishes processing a macro-block, it sends the data to the next module. The columns of fig.3 represent pipeline slots. The respective processing blocks, other than those paired in the figure, handle data for different macro-blocks, i.e. n-1, n, and n+1.

We call this pipeline mechanism the Dynamic Time Slot (DTS) system.

However, an external factor might cause the pipeline to stall in some cases. For example, the processing time can vary and delays may be introduced by the state of the external bus at the VLC-decoder, which reads the stream from an external buffer memory.

Entropy coding can also cause the pipeline to stall.

To handle these problems on the pipeline slot where they occur, the whole pipeline can be stalled for one slot (macro-block) and then restarted at the beginning of the next pipeline slot on completion of unfinished macro-block processing. This stall mechanism allows stable control of the

pipeline.

Fig.4 shows an example of pipeline stalling. In fig.4, the pipeline is stalled at pipeline slot “n” and restarted at slot “n+1”.

4. Advantages of the Dynamic Time Slot (DTS) System

When the DTS is introduced, time slots for all functional blocks are synchronized, facilitating the switching of parameter signals and the control of operations. This produces control circuits that are both simpler and smaller scale.

Buffers for pooling units of images to be processed between the respective functional blocks would typically be required. The DTS system facilitates buffer control by equalizing the rate of processing with respect to each functional block.

C. *Reducing the amount of memory and the number of access operations by optimizing data paths*

As stated above, real-time processing is achieved by having that the respective processing blocks of the hardware IP handle the data for each macro-block (16 by 16 pixel) unit in 1280 cycles. However, some types of processing cannot be completed in macro-block units. In other words, certain types of processing may be termed “inter-macro-block.” Such types of processing extend throughout the overall process of moving-picture encoding and decoding. Fig.5 consists of two flowcharts illustrating the decoding of VC-1 data. A detailed description follows.

For certain functions of the VC-1 codec, such as the overlap transform and deblocking filter, completing the processing of a macro-block requires the surrounding macro-blocks (left, right, above, and below). The typical flow is shown as case 1. Here, the input is the entire picture after the completion of all previous processing steps.

Specifically, the overlap transform and deblocking filter are used to filter the pixels adjacent to the boundaries of macro-blocks to smooth the borders between the macro-blocks. In other words, pixels adjacent to the boundaries of each macro-block that is processed are adjusted to smooth the moving picture as a whole.

The redundant portions of the processing in its functional level are found and such portions are removed. In case 2 (lower half of the figure), transposition memory is shared by the overlap transform and deblocking filter. This eliminates the need for frame memory. TABLE I below shows memory requirements and numbers of access operations for the two methods. The result demonstrates case 2’s superiority.

III. Low Power Consumption

Our measure for reduced power consumption and its results are briefly described below.

A. *Dynamic Clock Supply stop to Functional Blocks*

One measure for reduced power consumption is to monitor processing by functional blocks and dynamically stop the supply of clock-signals when operation is not required.

As shown in fig.4, the number of clock cycles required to process a macro-block of data depends on the input streams or input references.

For example, VLC decoding takes less time per macro-block for a still image than for moving pictures (i.e. 1280 clock cycles). In this case, after processing of the current macro-block, supply of the clock signal to that sub-module is stopped (see fig.7). The supply of clock signals to the internal SRAM can also be stopped, further improving efficiency by lowering power consumption.

Moreover, hardware specifically for encoding is not required while moving pictures are being decoded. Here, too, the supply of clock signals can be cut off.

We call the scheme as a whole Dynamic Clock Supply Stop (DCSS).

B. *Measurement of Power Consumption*

The table II, III shows the reduction in power consumption. These are current values as measured in simulated operation of the layout net. Details on the conditions are given below.

The DCSS scheme reduces power consumption in decoding and encoding by up to 22% and 17%, respectively .

IV. Top-down Modeling

This hardware IP is complex and includes embodiments of large algorithms. Any error found in its functions in the initial phases of development leads to backtracking and lengthens the development period. Avoiding this requires verification of the respective functions in the early stages of development. Highly accurate verification in the late stages is also indispensable. We thus divided the process of developing this hardware IP into the following phases.

A. *Phases of Development*

Phase1

Functional level: Create C language models to achieve the functionality of each unit. Comprehend the details of specifications.

- Verify the functional models and determine the architecture

Phase2

Architecture level: Create Register Transfer Level (RTL) descriptions and compare them with the C language models, bit by bit.

- Verify the architectural models to check real-time operation.
- Use the FPGA to improve the pace of verification.

The flow chart of development is shown in fig.8.

B. *Functional-model Replacement Method (FRM)*

The FRM offers an efficient debugging environment. In the architecture-level verification environment, an RTL sub-module with a problem or incomplete logic will be unable to execute top-level RTL simulation.

The FRM involves the execution of functional models (C language) and the output of processed macro-block data. This

output data is stored in a file, which is provided to the architecture-level verification environment.

The FRM eliminates the need to gain skills on newly introduced EDA tools, saving the corresponding amount of time.

Fig.9 shows the FRM in overview.

C. Other Approaches Used to Raise Accuracy in Verification

To ensure the accuracy of RTL simulation, the following methods have been applied in the hardware IP.

- Random pattern generation
- Static coverage
- Assertion based simulation:

Assertion scripts based on past problem cases were inserted into the RTL source code.

For example, these test the accuracy of bit widths and the protocols between interfaces.

V. Summary and Conclusions

We have succeeded in developing a hardware IP capable of operating as a real-time multi-codec handling the VC-1, H.264, and MPEG formats at D-1 resolution (720*480) and 30 fps with the low operating frequency of 54 MHz. This hardware IP is also suitable for use in mobile products that strongly require low power consumption. An original architecture and control method reduce power consumption in operation by up to 22 percent.

Acknowledgements

The authors thank Hiromi Watanabe, Hiroshi Hatae (Renesas Technology Corp.) and Tetsuya Yamada (Hitachi Ltd.) for their continuing support.

References

[1] International Standard, Information technology – Coding of audio-visual objects- Part2: Visual, ISO/IEC 14496-2 Third Edition, 2004/06/01

[2] ISO/IEC 14496-10, ”Advanced Video Coding” & ITU-T Rec. H.264, 2003

[3] SMPTE STANDARD VC-1 Compressed Video Bitstream Format and Decoding Process, Approved 24-February-2006

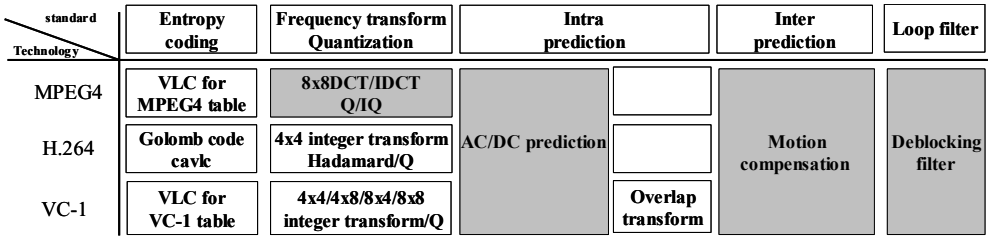


Fig. 1. Standards of video encoding and decoding technologies

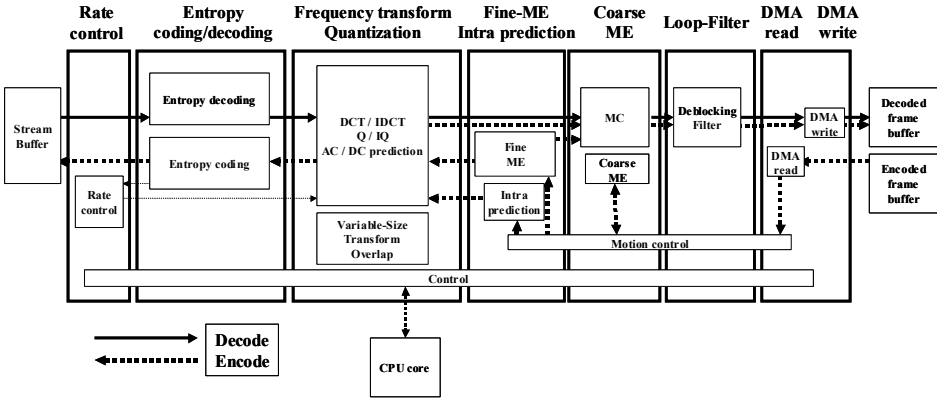
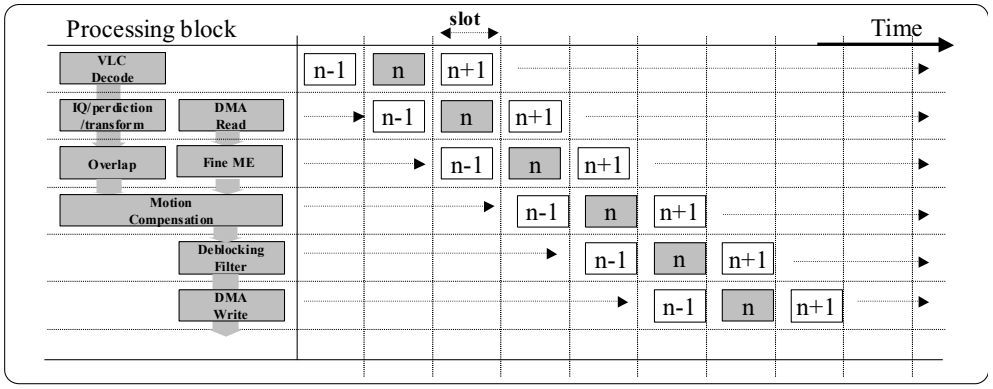


Fig.2. Block diagram



8 parallel processing ability with 6 pipeline stages

Fig.3. The timing chart for the various modules in VC-1 decoding

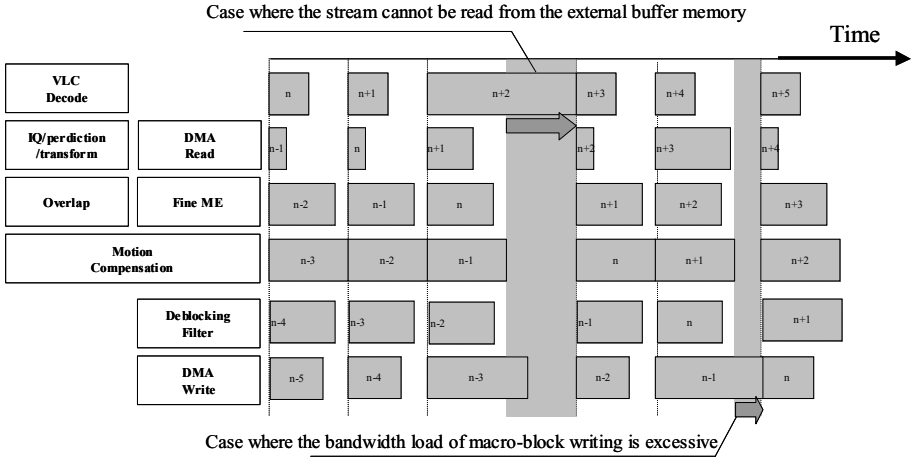


Fig.4. Examples of pipeline stalling

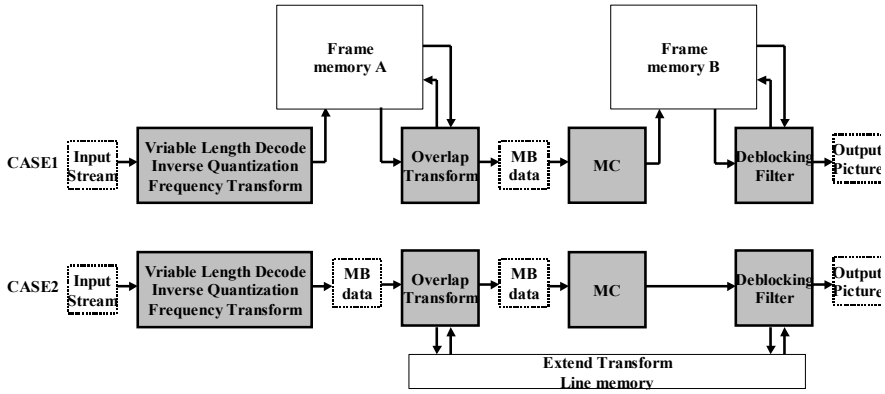


Fig.5. Flows of VC-1 processing (MB = macro-block)

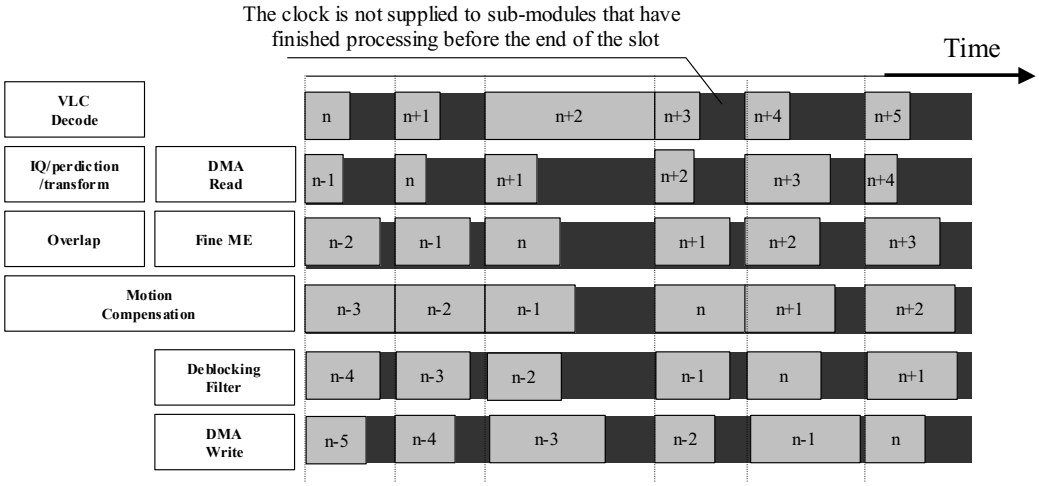


Fig.7. An example of DCSS

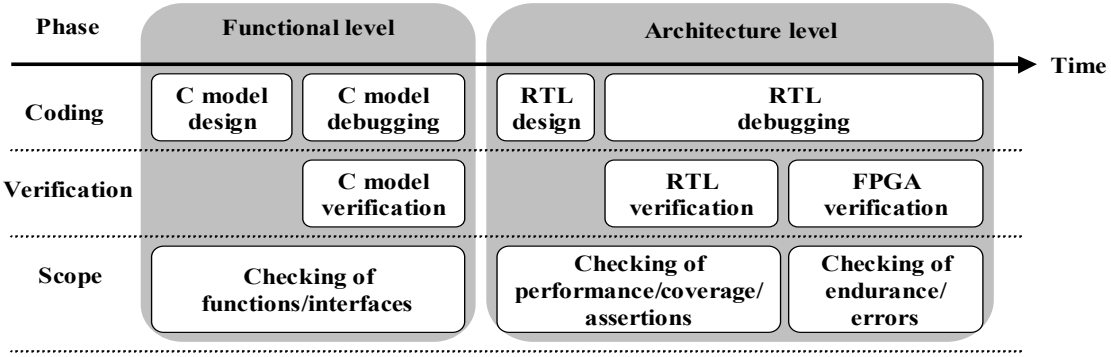


Fig.8. Flow of the development process

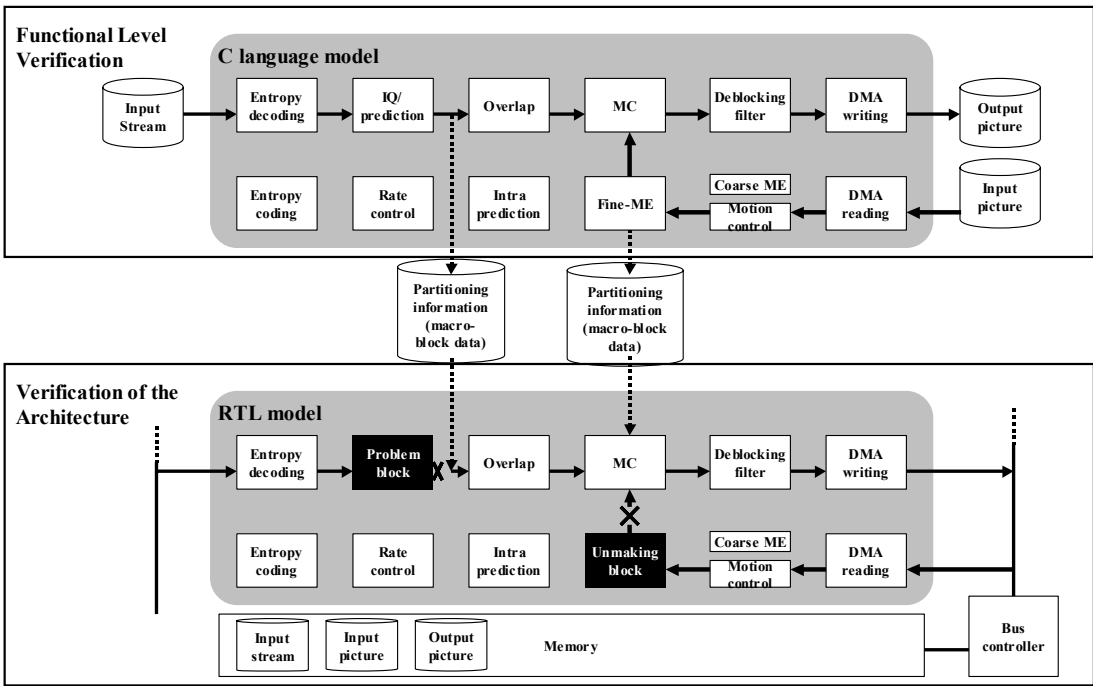


Fig.9. FRM in overview

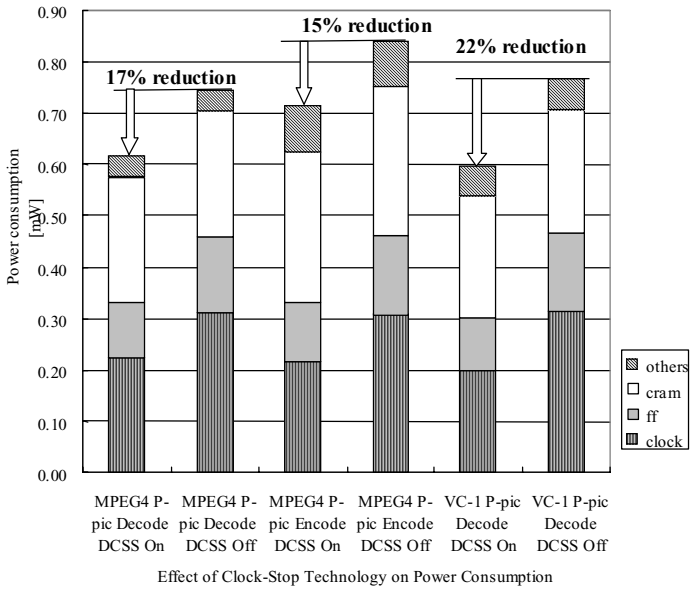
TABLE I

Comparison of cases 1 and 2 of VC-1 processing

	Memory[Byte]			Access cycles (per decoded macro-block)		
	Overlap	Deblock	Total	Overlap	Deblock	Total
CASE1	864000	345600	1209600	760	480	1240
CASE2	20608		20608	634	288	922

TABLE II

Power consumption and clock-stop technology



Effect of Clock-Stop Technology on Power Consumption

TABLE III

Conditions of power consumption

Measurement condition	
V=1.2[V]	MPEG4 Decoding of QCIF @15fps (DCSS On/Off)
	MPEG4 Encoding of QCIF @15fps (DCSS On/Off)
	VC-1 Decoding of QCIF @15fps (DCSS On/Off)

Effectiveness of DCSS control		
logic	Logic gates (excluding FFs)	◯
ff_data ff_clk	Logic gates (FFs)	◯
clock_drv	Clock tree	◎
cram	RAM	◯