

AN MPEG-2 ENCODER ARCHITECTURE BASED ON A SINGLE-CHIP DEDICATED LSI WITH A CONTROL MPU

Yasushi OOI, Osamu OHNISHI, Yutaka YOKOYAMA, Yoichi KATAYAMA, Masayuki MIZUNO[†]
Masakazu YAMASHINA[‡], Hideto TAKANO[‡], Naoya HAYASHI[‡], and Ichiro TAMITANI[‡]

Information Technology Res. Labs, NEC Corporation,
1-1, Miyazaki 4-chome, Miyamae-ku, Kawasaki, Kanagawa 216, Japan

[†] Microelectronics Res. Labs, NEC Corporation,
1120, Shimokuzawa, Sagami-hara, Kanagawa 229, Japan

[‡] ULSI Systems Dev. Labs, NEC Corporation,
1753, Shimonumabe, Nakahara-ku, Kawasaki, Kanagawa 211, Japan

ABSTRACT

This paper describes an MPEG-2 encoder architecture based on a hard-wired LSI with a control MPU. All basic functions of MPEG-2 MP@ML video compression are integrated in the dedicated LSI. For the motion estimation, a horizontally subsampled, diamond search was employed as a simplified first search step. It can reduce operations to 20% of the full-search, with an estimated SNR degradation of only -0.1dB. To help achieve a single-memory interface, a pair of 81MHz, 16Mb SDRAMs are used as a frame buffer and a code buffer. Data bandwidth between the SDRAMs and the LSI is kept to less than 94% of the maximum data rate. Jobs assigned to the control MPU need be executed less frequently than those of the macroblock coding, which helps reduce the requirements for MPU performance to about 7MIPS.

1. INTRODUCTION

The MPEG-2 algorithm [1] is generic in the sense that multimedia applications in several different fields take advantage of this technology. Since 1994, the year that the Draft was approved as the International Standard (IS), a number of MPEG encoding LSIs have been reported [2, 3, 4, 5]. Unlike MPEG-2 decoders, conventional encoding systems include multiple VLSI chips and several kinds of memory LSIs. The die size of the chips is not suitable for low-cost mass production, however, a critical limitation for future personal encoding applications. In addition, their power dissipation is not low enough for portable use.

This paper describes a single-chip video encoder LSI that performs all basic functions of MPEG-2 video compression. After describing the specifications of the LSI, we review three critical topics related to the single-chip integration: a simplified motion estimation algorithm, reduction of the memory bandwidth in a single memory architecture, and reduction of the coding control performance required of the associated MPU.

2. OVERVIEW OF THE VIDEO ENCODER LSI

A block diagram of the video encoder LSI is shown in Figure 1. The LSI consists of a system control (SYS) unit; a motion estimation (ME) unit; a block processing (BP) unit for

DCT, quantization, and other numerical operations; a variable length coding (VLC) unit; video input/output (VI/VO) units; a host interface (HIF) unit connected to the control MPU; a synchronous DRAM (SDRAM) interface (SIF) unit; and a packet generation (PG) unit.

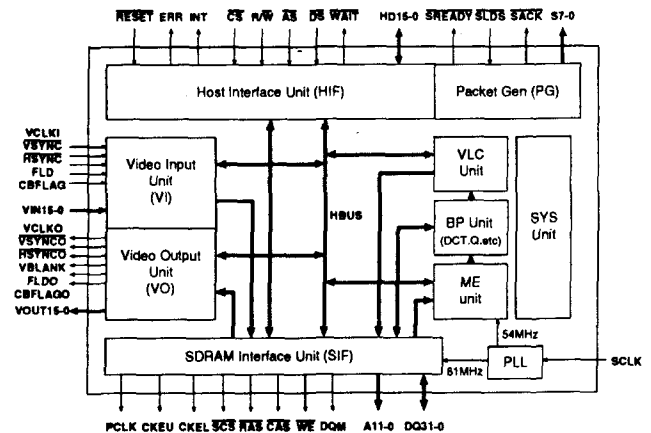


Figure 1: Block diagram.

The SYS unit is the main control unit for video coding operations. Start signals for each unit are generated to manage a macroblock-pipeline timing scheme. The unit also handles commands issued by the control MPU. Typical commands are: encode, stop (for encoding), capture, display (for video input/output), and SDRAM power-down.

In the ME unit, a simplified, local-decode-based two-step search algorithm is adopted to help reduce required operations and memory bandwidth, as will be described later. The BP unit incorporates a mixed-pixel-pipeline scheme in the macroblock execution period, providing 2-pel/cycle DCT/IDCT operations and 1-pel/cycle Q/IQ operations. The VLC unit generates either an MPEG-1 or MPEG-2 bitstream.

The VI unit has a 6-tap 4:2:2 to 4:2:0 chrominance down-sampler, an ITU-R 601 to SIF converter, and a temporal noise reducer. The VO unit generates monitor output to evaluate the coded picture quality. It also has up-sampling converters and filters.

The PG unit outputs a system bitstream. It attaches

stream headers, which had been stored in SDRAMs and updated by the MPU, to the streams of audio and video. It also inserts Clock Reference fields into a merged audio/video stream.

While most of the units operate at 54MHz, the VI/VO units operate at 13.5MHz. To maintain high-speed transfer of video data and codes, SDRAMs are controlled by the SIF unit to operate at 81MHz (12.3ns) cycle, and this unit issues continual, 3-cycle latency SDRAM cycles.

3. SIMPLIFIED MOTION ESTIMATION

One of the key issues in designing an MPEG-2 video encoder LSI is to reduce the complexity of the motion estimation. A full-search block matching algorithm in a large search area involves considerable computation as well as high memory bandwidth. While several simplified motion vector search algorithms have been proposed [6, 7, 8, 9], none of these proposals discusses its effectiveness in MPEG-2 MP@ML video encoding, the area most in need of discussion.

In our approach, we first listed a set of algorithm primitives that can reduce the number of operations. We classified these into three categories:

1. 2-to-1 subsampled searches after interpolation (LPF): full-search(F)/horizontal(H)/+vertical(V)
2. search point reduction: full-search(F)/diamond(D)/checkered(C)
3. narrowing down the reference fields of the 2nd search: {top,bottom}{for,back}(4)/(2)/(1),

in which the diamond search uses a rhombic search window whose area is half of the original rectangle, and the checkered search uses 2-to-1 subsampled, skew grids for the reference picture.

The algorithm primitives are then combined to form a complex algorithm, abbreviated as 'HC2' in reference to the horizontal/checkered primitives above. Search points for 'HF_n' and 'FC_n' are shown in the example, illustrated in Figure 2. Note that reducing search points of the first search step (white circle) leads to an increase of the second half-pel search points (black circle).

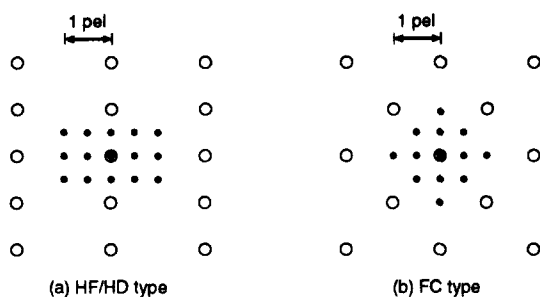


Figure 2: Examples of simplified ME algorithms.

We have also examined how the reduction of motion estimation complexity affects the quality of the compressed video, considering the trade-off of ratio of operations (normalized as 'FF4'=1) versus SNR degradation from 'FF4' (Figure 3). Four sequences (Cheer, Mobile, Bicycle, and

Flower) were simulated under the following conditions: 4Mbps, bidirectional predictions ($m=3$, and $n=15$), searches within ± 16 pixels per frame, and sequences of 60 NTSC frames each.

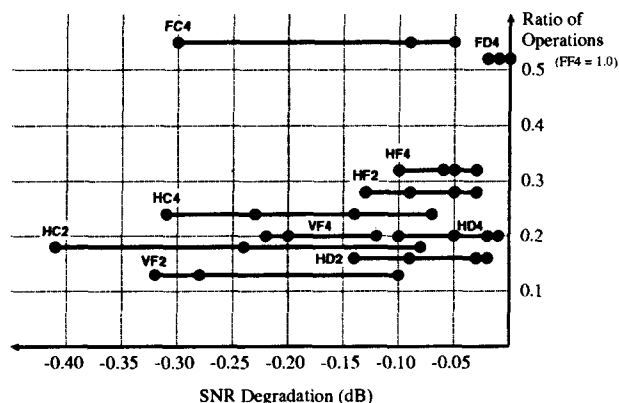


Figure 3: Ratio of operations v.s. SNR with simplified MEs (bidirectional predictions, $m=3$).

Figure 3 shows degradation when the V algorithm primitive in the first category is used. The C primitive in the second category fails to find the best match in some sequences. While there are no such clear-cut differences in the third category, in terms of SNR degradation, they are not, nevertheless, as to be negligible. We performed similar tests for the dual-prime prediction (Figure 4).

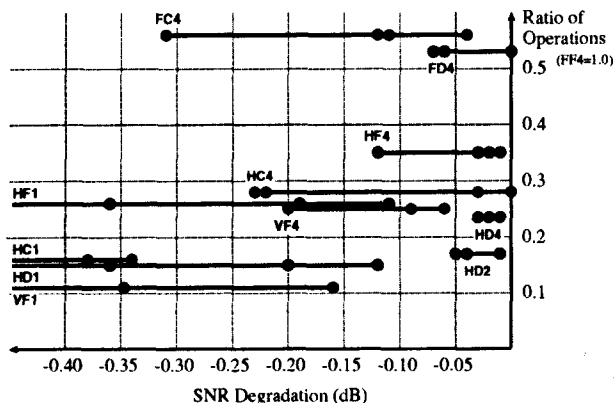


Figure 4: Ratio of operations v.s. SNR with simplified MEs (dual-prime predictions, $m=1$).

We concluded from these results that (1) only subsampling in the horizontal direction is permissible, (2) reducing search points for oblique vectors works effectively, and (3) as many the field-based second searches should be performed as possible. We employed 'HD4' for bidirectional prediction and 'HD2' for dual-prime prediction, each of which provide 20% of the computations of the full-search and produces at most -0.1dB SNR degradation. These results are superior to [9], in which 'VF4' is employed and the SNR degradation is

estimated to be -0.2dB. (Our results for 'VF4', obtained for nearly the same number of operations as for 'HD4', confirm this figure.)

Our main concern here is circuit design, especially the effectiveness of diamond search design. To overcome hardware restrictions, we divided the search window into multiple rectangular segments over the rhombus (Figure 5). We have designed an array of ME processing elements so that that efficiency of operations approaches 100% [10]. In addition, the height and the vertical position of segments in our design can, if necessary, be changed. A optional window shift control is also added to follow out-bounded motion vectors.

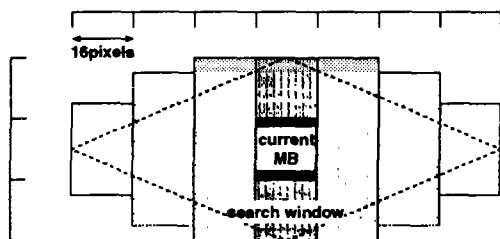


Figure 5: Segmented search window (P-picture, m=3).

4. SINGLE MEMORY ARCHITECTURE

An example of the MPEG-2 encoder system configuration using the video encoder LSI is shown in Figure 6.

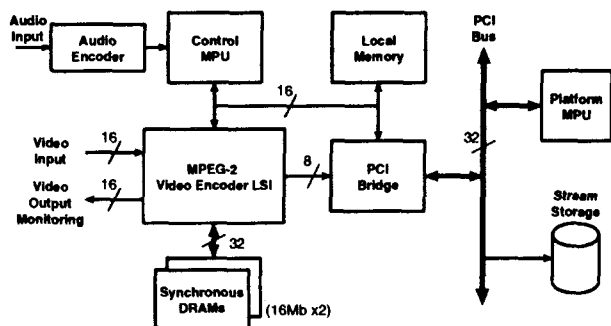


Figure 6: System configuration.

To achieve a single memory architecture, we used two 16Mb SDRAMs as a frame buffer and a code buffer. The memory bandwidth is modeled as $ClockRate \times 4(byte/word) \times \gamma$, where γ is the efficiency of burst SDRAM accesses and is here assumed to be 0.8. Since 81MHz SDRAMs (3 times the 27MHz system clock rate) are used, the limit of the memory bandwidth is 259MB/s.

To estimate memory bandwidth, we roughly classified the memory accesses into ME reference accesses and "other", which consists of an external video input, a read for video encoding, a write for decoded picture for prediction, an external video output, and a video read for noise reduction.

Note that these five all have the same rate and their total may be given as $(720 \times 480 \times 30 \times 1.5) \times 5 = 77.76MB/s$.

ME reference accesses consist of the first search reference read, the second search reference read, and a prediction read. We need six times the luminance video data rate to fetch a ± 16 vertical area in the bidirectional first search, which involves $720 \times 480 \times 30 \times 6 = 62.21MB/s$. We also need more than six times the luminance video data rate to fetch two bidirectional frame candidates and four bidirectional field candidates in the second search. The grand total, then, is over 200MB/s.

In addition, memory alignment restrictions and unavoidable margins push up this estimate, which makes it hard to conduct all transactions within 259MB/s. For this reason, we decided that the first search should be made on the local decode picture to reduce the number of memory transactions, and in additional simulations we were able to confirm that the SNR degradation produced by this restriction is negligible.

In SDRAM arbitration, we used a simple circulating algorithm for SDRAM access scheduling (Figure 7). Further, in actual implementation, we were able to keep the bandwidth between the SDRAMs and the video encoder LSI to less than 94% of the maximum data rate (Figure 8).

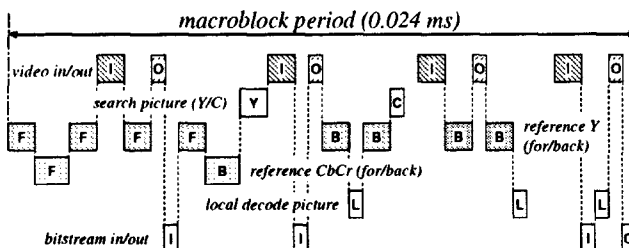


Figure 7: SDRAM access scheduling.

	Others	F	B
Field B (94%)	62%	16%	16%
Frame B (86%)	62%	12%	12%
Field P (78%)	62%	16%	
Frame P (74%)	62%	12%	

Figure 8: Ratio of SDRAM bandwidth.

5. CODING CONTROL BY THE MPU

The computation requirements for video encoding are enormous; even after our simplification of the ME algorithm, we still need a performance of 30GOPS, of which 28GOPS are for motion estimation. Since VLC operations alone consume 100MOPS in the worst case, the control MPU can hardly be expected to play a main part in MPEG-2 video encoding algorithm execution.

In our design, jobs assigned to the control MPU need be executed less frequently than those of the macroblock coding, which helps reduce the requirements for MPU performance. It calculates parameters for buffer control, activity control, search control, and bitstream headers. Audio code transfer is also a part of MPU processing.

MPU performance in these real-time controls is estimated at about 7MIPS (Table 1), which means that a mid-class RISC will have sufficient performance for our purposes. Any extra computation ability can also be used for video quality improvement and power reduction of the ME unit.

TABLE I: MPU PERFORMANCE ESTIMATION

Processing	Frequency (times/s)	Step Count of the MPU	Performance (MIPS)
Audio Code Transfer	< 50,000	112	< 5.6
Rate Control per Slice	900	< 1,100	< 0.99
Others per Picture	30	< 2,900	< 0.09
Total			< 6.68

Figure 9 is a timing diagram of picture tasks. In our estimates, we assumed a 33MIPS RISC. The MPU is triggered by video interrupts to accomplish concurrent processing with the video encoder LSI.

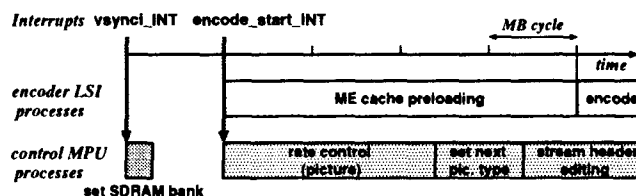


Figure 9: Timing diagram of the picture tasks.

6. CONCLUSION

We have presented here an MPEG-2 encoder architecture based on a dedicated LSI with an MPU. The video encoder LSI integrates 3.1M transistors on a $12.45\text{mm} \times 12.45\text{mm}$ die in $0.35\mu\text{m}$ three-metal-layer CMOS (Figure 10). The chip consumes 1.5W at 3.3V (2.5V for the ME unit) and at a 54MHz internal clock [10]. With reduced chip count and the use of a mid-class MPU, we are able to achieve both system cost-down and system power reduction.

ACKNOWLEDGMENTS

The authors would like to thank Takao Nishitani, Masao Fukuma, Hideo Kurokawa, and Shigeo Niitsu for their encouragement and continual support. We are also very grateful to Noriyuki Miki of NEC Microcomputer Technology, Ltd. for his work on designing and checking programs for the control MPU. Additionally, the help of all project members that contributed to the design of the experimental chip has been very greatly appreciated.

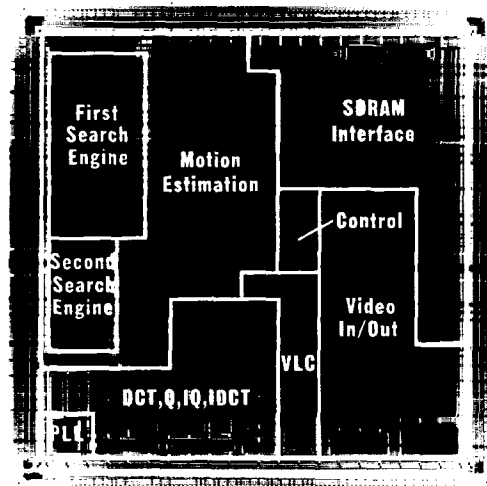


Figure 10: Photomicrograph of the video encoder LSI.

REFERENCES

- [1] "Information Technology - Generic Coding of Moving Pictures and Associated Audio," ISO/IEC 13818-2 International Standard (Video), Nov. 11, 1994.
- [2] T. Matsumura, et al., "A Chip Set Architecture for Programmable Real-Time MPEG2 Video Encoder," Proc. of CICC, 17.1, pp. 393-396, May 1995.
- [3] A. Ohtani, et al., "A Motion Estimation Processor for MPEG2 Video Real Time Encoding at Wide Search Range," Proc. of CICC, 17.4, pp. 405-408, May 1995.
- [4] J. Murdock, et al., "VLSI Architecture of the I-Frame Encoder for the MPEG-2 Video Compression," Digest of Hot Chips VII, 4.2, pp. 103-110, Aug. 1995.
- [5] T. Kondo, et al., "A Two-Chip Real-Time MPEG2 Video Encoder with Wide Range Motion Estimation," Digest of Hot Chips VII, 4.1, pp. 95-101, Aug. 1995.
- [6] N. Hayashi, et al., "A Compact Motion Estimator with a Simplified Vector Search Strategy Maintaining Encoded Picture Quality," Proc. of CICC, 17.5, pp. 409-412, May 1995.
- [7] P. Pirsch, et al., "VLSI Architecture for Video Compression - A Survey," Proc. of the IEEE, vol. 83, no. 2, pp. 220-246, Feb. 1995.
- [8] H. D. Lin, et al., "A Programmable Motion Estimator for a Class of Hierarchical Algorithms," VLSI Signal Processing VIII, IEEE Press, pp. 411-420, 1995.
- [9] K. Suguri, et al., "A Real-Time Motion Estimation and Compensation LSI with Wide-Search Range for MPEG2 Video Encoding," Digest of Technical Papers, ISSCC'96, pp. 242-243, Feb. 1996.
- [10] M. Mizuno, et al., "A 1.5W Single-chip MPEG2 MP@ML Encoder with Low Power Motion Estimation and Clocking," Digest of Technical Papers, ISSCC'97, Feb. 1997.