An Implementation of MPEG-2 Encoder to Multiprocessor System using Multiple MVPs (TMS320C80)

HyungSun Kim, Kenny Boo, SeokWoo Chung, GeonYoung Choi*
YongJin Lee, JaeHo Jeon, and HyunWook Park

Department of Information & Communication Engineering,
Korea Advanced Institute of Science and Technology (KAIST),
207-43 Cheongryangri, Dongdaemun, Seoul, Korea
* Also at Samsung Electronics Co., Ltd., Korea

ABSTRACT

This paper presents the efficient algorithm mapping for the real-time MPEG-2 encoding on the KAIST Image Computing System (KICS), which has a parallel architecture using five multimedia video processors (MVP's). The MVP is a general purpose digital signal processor (DSP) of Texas Instrument. It combines one floating-point processor and four fixed-point DSP's on a single chip. The KICS uses the MVP as a primary processing element (PE). Two PE's form a cluster, and there are two processing clusters in the KICS. Real-time MPEG-2 encoder is implemented through the spatial and the functional partitioning strategies. Encoding process of spatially partitioned half of the video input frame is assigned to one processing cluster. Two PE's perform the functionally partitioned MPEG-2 encoding tasks in the pipelined operation mode. One PE of a cluster carries out the transform coding part and the other performs the predictive coding part of the MPEG-2 encoding algorithm. One MVP among five MVP's is used for system control and interface with host computer. This paper introduces an implementation of the MPEG-2 algorithm with a parallel processing architecture.

Keyword *: MPEG-2, Multimedia Video Processor (MVP), Discrete Cosine Transform (DCT), Spatial partitioning, Functional partitioning, Digital signal processor (DSP)*

1. INTRODUCTION

In recent years, there are a lot of demands for the multimedia systems manipulating various kinds of data such as image, video, text and audio. The multimedia data processing systems need high-performance processing power because image computing requires a large number of computations and data transfers. Furthermore, multimedia data processing usually requires real-time operations. However, the media storage, the data transmission capacity and the data computing processors barely keep up with the demands for the multimedia applications in spite of recent significant development of the VLSI technology. Therefore, a great deal of efforts has been concentrated on reducing the vast amount of data of the multimedia information, thereby the various video coding standards such as H.261, JPEG, MPEG-1, and MPEG-2 have been established [1][2][3][4].

The MPEG is the video coding standard established by the Motion Pictures Expert Group of the International Organization for Standardization (ISO). The MPEG-1 standard specifies the video coding algorithm for digital storage media up to 1.5 Mbits/s with the recommended picture size of 360 x 240 pixels [3]. MPEG-2 is a generic standard that supports wide class of applications requiring bit-rates of 2 Mbits/s or above with an image quality ranging from a studio quality NTSC broadcasting up to the high-definition television (HDTV) [4]. The MPEG video compression algorithm employs two basic techniques; block-based motion compensation for the reduction of the temporal redundancy, and transform domain (DCT) coding for the reduction of spatial redundancy. The motion compensation technique is applied both in the forward and the backward directions. The motion-compensated prediction errors are coded using the transform-based

SPIE Vol. 3031 • 0277-786X/97/$10.00

technique. The motion vectors are transmitted together with the spatial information.

In the multimedia processing systems, the programmability and the flexibility are required to support various image processing algorithms. Therefore, the system using fully programmable processor such as general-purpose DSP is very useful for multimedia processing system, which can support a general image processing algorithm as well as an application specific algorithm such as MPEG-2 standard [5]. In addition, the real-time processing capability should be included.

In this paper, the KAIST Image Computing System (KICS) is introduced as a platform for MPEG-2 real-time encoding, which is a high-performance image processing system [6]. Also additional hardware for an implementation of a real-time motion estimation with a half-pixel accuracy and the software architecture for real-time MPEG-2 main profile at main level (MP@ML) encoding on the KICS are described. The hardware architecture of the KICS is based on the message passing model with hierarchically segmented buses which can fully support the generic model for MPEG-2 encoding. The system consists of two clusters. Inside each cluster, two processing elements (PE) are pipelined with each other. The MVP (Multimedia Video Processor; TMS320C80), which is a general-purpose digital signal processor (DSP) especially designed for image processing and multimedia applications, is used as a PE in the KICS. The master PE of the cluster efficiently accesses the image data in the common global memory through the status arbiter. The spatially partitioned image data are fetched by master PE's of the clusters, and are processed in a pipelined fashion. The KICS is a fully programmable system, which can perform a general image processing as well as a specific application such as real-time MPEG-2 encoding/decoding by software programming.

The software architecture of the KICS for the MPEG-2 encoding is presented. The encoding algorithm is partitioned according to the functional module such as motion estimation, motion compensation, DCT, quantization, inverse quantization, IDCT, and variable length coding (VLC). By the spatial partitioning of the input image, processing of a half-picture is assigned to each cluster. This paper shows the real-time MPEG-2 MP@ML encoding can be implemented successfully on the KICS, which has five MVP's and a specific hardware for motion estimation.

2. HARDWARE PLATFORM FOR MPEG2 ENCODER : KICS [6]

The overall hardware block diagram of the KICS is shown in Fig.1. The key features of the architecture are simple topology for message passing model with hierarchically segmented buses, powerful PE with programmable features, high-speed memory access with enhanced memory performance, and flexible structure of the inter-PE pipelining. Figure 2 shows the internal architecture of the MVP, which has a master processor (MP), four advanced DSP's (ADSP), on-chip memory, crossbar network, and transfer controller for internal and external data transfer [7][8][9].

The KICS consists of four kinds of functional modules, such as IPU (Image Processing Unit), IOU (Image I/O Unit), IMU (Image Memory Unit), and MESU (Motion Estimation Sub-Unit). The MVP (TMS320C80), which is a general-purpose DSP of Texas Instrument, is used as a PE in the KICS. Two PE's are grouped together to form a cluster. Two clusters are linked through global memory and data queue. Each PE inside the cluster has the local bus and its own local memory. The master MVP which is located in the IOU interprets the command from the host system, and manages the operation of PE's in the clusters. All tasks which are performed in the PE's are passed through the data and command queues between the master MVP and PE's. The global memory modules of the IMU are located between clusters and IOU. The real-time video module, which stores the digitized video data in the global memory modules or fetches the real-time image from the global memory modules, is interfaced to the global bus. Master PE's of two clusters and real-time video module can access the global memory modules. Master PEs and video module compete for occupying the global memory modules to access the image data through the status-arbiter. Here, the video module has higher priority than master PE's because the video module should provide real-time image acquisition and display. The status-arbiter is located at each memory module, and it reports the status of the occupation of the memory modules when a processor tries to occupy a global memory module. If the memory module is already occupied by other processor, it prohibits the processor to occupy the memory module until the other processor completes the memory access. Because the image data can be transferred in block data, it is reasonable to fetch a series of data without contention from any other processor by using the status arbiter.

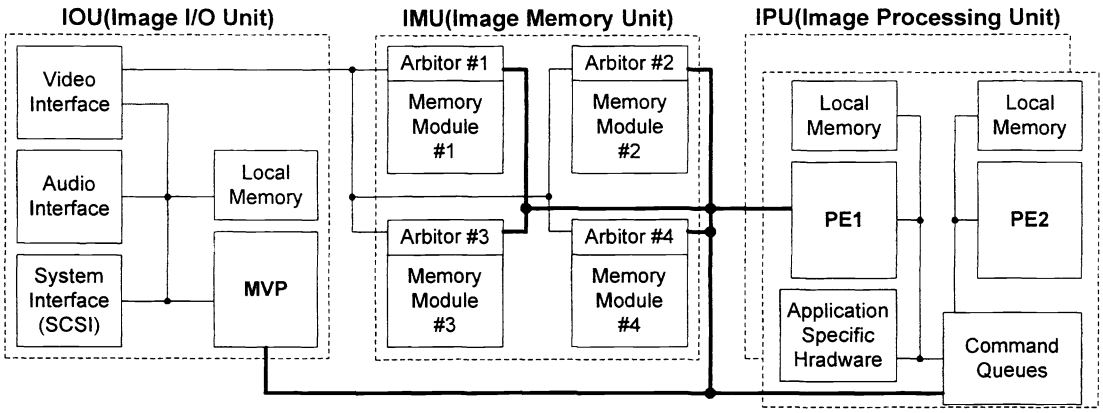

Fig. 1. The overall block diagram of the KAIST image computing system (KICS)

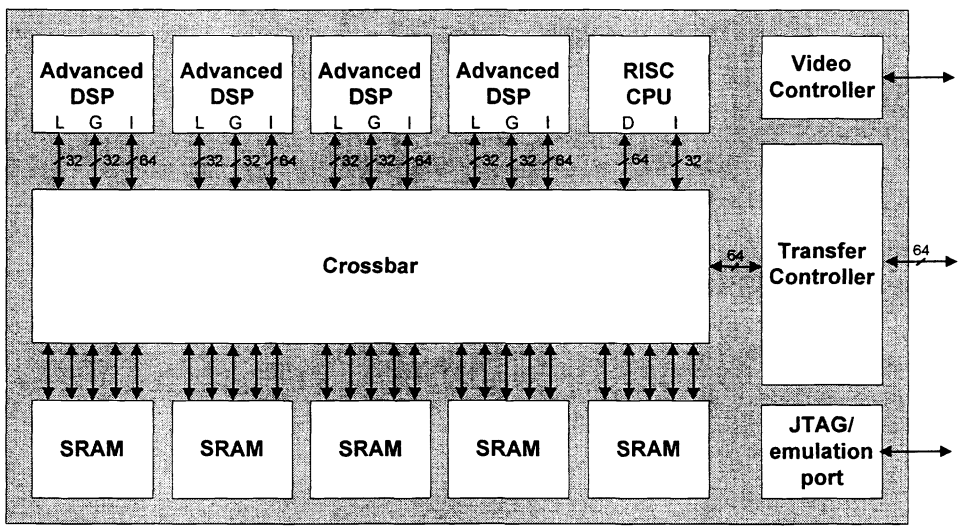

Fig. 2. The internal architecture of MVP

3. SOFTWARE STRUCTURE FOR MPEG-2 ENCODING ON KICS

3.1. Algorithm mapping of MPEG-2 encoding functions

For an implementation of the real-time MPEG-2 encoding on the KICS, the encoding algorithm is partitioned according to the functional entities, such as format conversion, motion estimation, predictive coding (motion compensation, DCT, and quantization), reconstruction, and entropy coding. By the spatial partitioning of the input data, processing of a half-picture (720 x 240 pixels) is assigned to each cluster. In order to implement real-time MPEG-2 encoding, a cluster has to complete encoding process of a half-picture within 1/30 sec. Figure 3 shows the algorithm mapping by functional partitioning on the KICS. The predictive coding (motion estimation, generation of the motion-compensated prediction errors, and reconstruction) is assigned to MVP1 and MVP3, whereas the transform coding (DCT, Quantization, Inverse Quantization and IDCT) is assigned to MVP2 and MVP4.

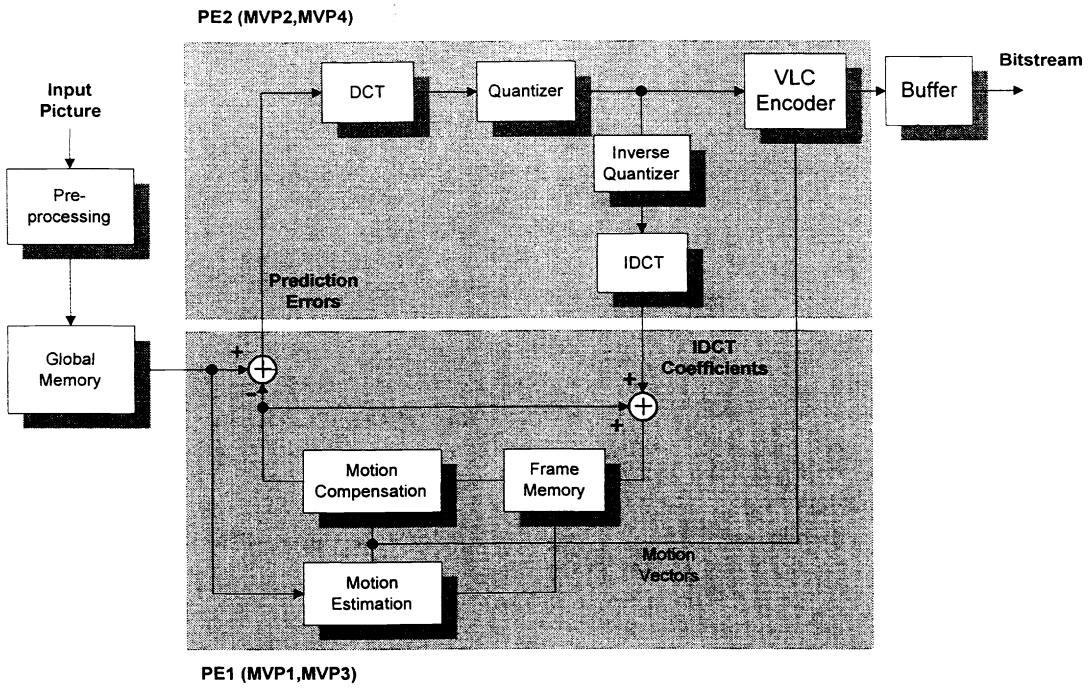

Fig. 3. Algorithm mapping of the MPEG-2 encoding on the KICS

3.2. Motion Estimation and Motion Compensation Assigned to MVP1

The MVP1 and MVP3 in the KICS perform the MPEG-2 encoding loop except the transform coding part (DCT, IDCT, Quantization, Inverse Quantization) with the image data of half-picture size (720 x 240 pixels), respectively. In this section, the computational complexity for implementing the various motion estimation on the ADSP of MVP is compared.

Let us assume block size is M x N, search region is $\pm p$, and K is the number of operations per pixel for a given criterion. For mean absolute difference (MAD) criterion, it is possible to compute eight sums of absolute difference in four instructions using the multiple-byte arithmetic operation in ADSP assembler of MVP. Therefore, we can calculate the MAD in 0.5 cycles per pixel, that is, $K = 0.5$. Table 1 shows the estimated number of cycles per MB for various block-matching algorithms [1][8].

Search Method	*Operations per Macroblock*	*Number of cycles per MB (p=15, M=N=16, K=0.5)*
Full-search	$(2p+1)^2MNK$	123,008 (cycles/MB)
2-D Logarithmic search	$(8[\log_2 p]+1)MNK$	4,224 (cycles/MB)
3-Level hierarchical search	$[(2[p/4] + 1)^2 + 180]MNK/16$	2,088 (cycles/MB)
2-Level hierarchical search	$[(2[p/2] + 1)^2 + 36]MNK/4$	10,400 (cycles/MB)

Table 1. Computation requirements for implementing various motion estimation algorithms using ADSP assembler

In order to implement any motion estimation algorithm on KICS, an MVP in a cluster should complete the MAD computations of a half-picture (675 MBs) within 1/30 sec (33.3 msec). Since one ADSP in MVP1 performs tasks of generating the prediction error and the reconstruction, other three ADSPs should perform the motion estimation processing in parallel. The total operation cycles available for computing MAD of one MB are estimated as follows :

- MVP clock speed : 40 MHz (25 nsec/cycle)
- Total number of MB in a half-picture (720x240 pixels) : 675 MBs
- Available number of operation cycles per MB for one ADSP:
  1/30 (sec/half-picture) / 675 (MBs/half-picture) / 25 (ns/cycle) = 1975 cycles/MB

For three ADSPs, total available number of operation cycles is 5925 cycles/MB. Therefore, we can implement the 2-D logarithmic search or 3-layer hierarchical search in real-time because their computational requirements are less than 5925 cycles/MB as shown in Table 1. However, it is impossible to implement the bi-directional search and half-pel accuracy motion estimation with above computing power on one MVP. Therefore, even if we use the fast search algorithms, such as 2-D logarithmic search or 3-layer hierarchical search, an extra hardware is required to relieve the MVP from the large amount of computations, in order to implement the bi-directional motion estimation as well as half-pixel motion estimation in real-time with one MVP on the KICS. The extra hardware, MESU, acts as a server of the MVP for performing the computing-intensive block matching algorithm. The MVP only transfers the data for block matching (the current MB and the corresponding search window) to the MESU, and receives the resulting motion vector from the MESU after it completes the block matching. The hardware module for finding the motion vector has the search range of [-8,+7] with MAD criterion, and the block size for block matching is 8 x 8 or 16 x 16 pixels. In the typical MPEG-2 MP@ML encoding systems, the motion estimation is performed using only luminance component with a block size of 16 x16. MPEG-2 encoder requires half-pixel accuracy motion vectors in the search range of -16.5 to +15.5, as well as bi-directional search in the B-picture. However, the MESU is designed to find the motion vector in the search range of only [-8, +7].

In order to implement the motion estimation in real-time on the KICS based on MPEG-2 standard including the bi-directional search and half-pixel motion estimation, we propose the three-level hierarchical search (TLHS) algorithm with half-pel precision. It performs a hierarchical coarse-fine search with a half-pixel precision in three levels as shown in Fig. 3.2, which is described as follows:

- First level : Decimate the current MB (16 x 16 pixels) and corresponding search window (47 x 47 pixels) with half sampling rate in the horizontal and the vertical directions. Motion estimation is performed with the decimated macroblock (8 x 8 pixels) on the hardware, MESU. Then, the motion vector with double pixel accuracy, MV_d, and MAD (Mean absolute difference) with double pixel accuracy can be obtained.
- Second level : Calculate the MAD at the eight adjacent points around MV_d which is obtained at the first level, with the macroblock size of 16 x 16 pixels. Then, the motion vector with full pixel accuracy, MV_f, and its MAD are obtained.
- Third level : Using linear approximation of MAD fields with four MADs at four adjacent points around MV_f, the motion vector with half-pixel accuracy, MV_h, is obtained [11].

Table 2 shows the execution time of the predictive coding part assigned to one MVP about I-picture, P-picture and B-picture, respectively. The execution time for the P-picture is longer than that of other two types of pictures, and it is much less than the allowed processing time, 1/30 sec/picture, for real-time encoding. Therefore, it guarantees the real-time processing of the MPEG-2 encoding sufficiently. As a result, the proposed three-level hierarchical search with ME hardware relieves the time consuming computation of block matching motion estimation, thus it makes the real-time MPEG-2 encoding possible on KICS successfully.

Function	I-picture	P-picture	B-picture
Retrieving frame sequence	1.62 ms	1.62 ms	1.62 ms
Motion estimation	–	7.29 ms	10.75 ms
Prediction	7.29 ms	7.29 ms	7.29 ms
Reconstruction	12.96 ms	12.96 ms	–
Total	**21.87 ms**	**29.16 ms**	**19.66 ms**

Table 2. The execution time for key functions in MPEG-2 encoding assigned to one PE of a cluster

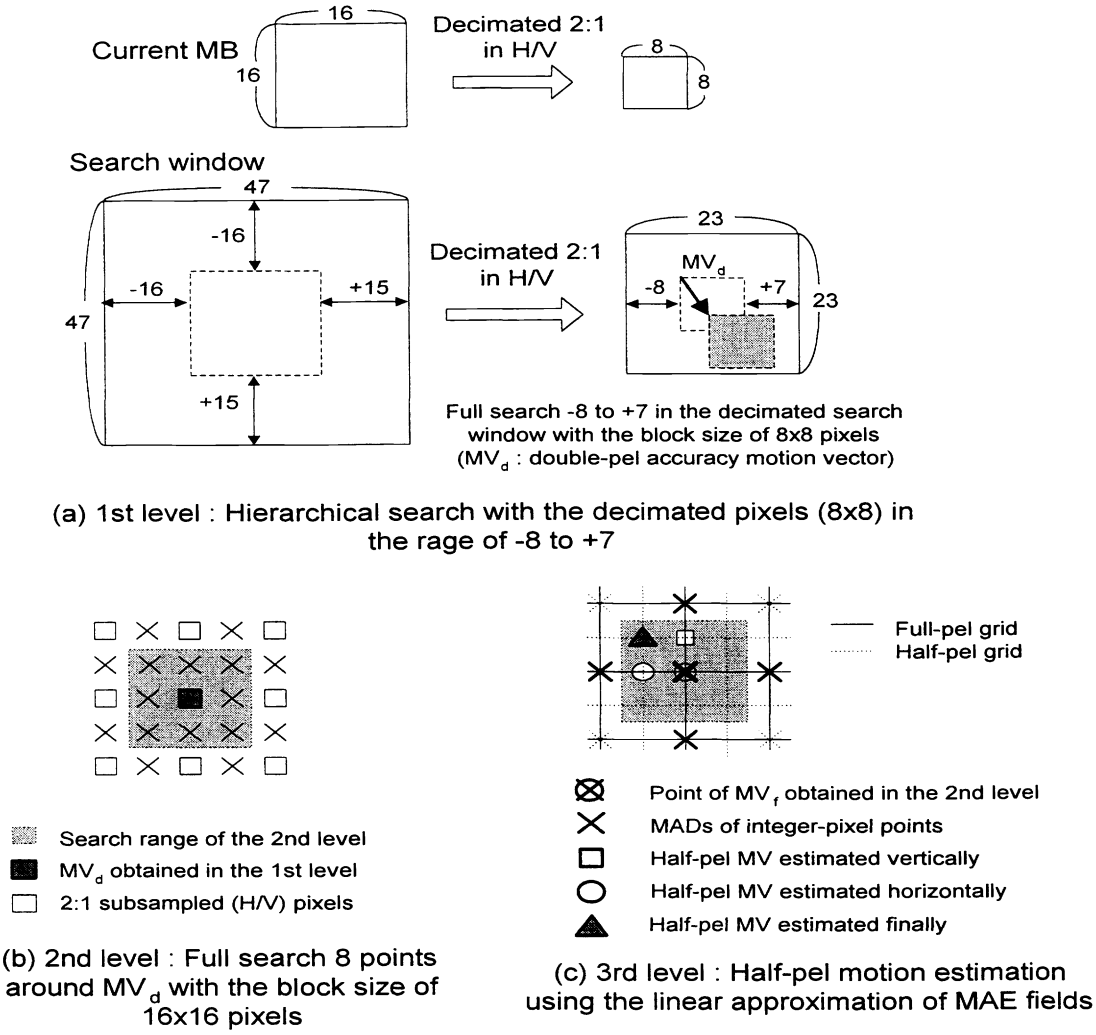

Fig. 4. The three-level hierarchical search with half-pixel accuracy

3.3. Transform coding assigned to MVP2

Two PEs of the MVP2 and MVP4 in KICS perform the MPEG-2 transform coding part (DCT, IDCT, Quantization, Inverse Quantization). Each PE can perform the transform coding with the image data of half-picture size (720 x 240 pixels). Two- dimensional DCT and IDCT are separable transformation. Therefore, it can be computed by applying 1-D DCT on every row and by applying 1-D DCT on every column to the result of 1-D DCT on every row. In order to achieve real-time implementation for a whole image, a fast computing algorithm is required. There are a lot of efforts to reduce the number of additions and multiplications and several fast DCT algorithms have been developed [12][13]. Some of the fast algorithms for the DCT exploit the properties of the DCT multiplicative constants. We modify the Chen DCT algorithm [13] by using scaling procedure so that all of eight outputs of one-dimensional DCT are associated with multiplicative constants as shown in Fig. 5. After performing one-dimensional DCT on every column, the 8 x 8 output matrix F for the 8 x 8 input matrix X can be decomposed as follows,

$$F=TX=\begin{bmatrix} C4v_{0,0} & C4v_{0,1} & C4v_{0,2} & C4v_{0,3} & C4v_{0,4} & C4v_{0,5} & C4v_{0,6} & C4v_{0,7} \\ C4v_{1,0} & C4v_{1,1} & C4v_{1,2} & C4v_{1,3} & C4v_{1,4} & C4v_{1,5} & C4v_{1,6} & C4v_{1,7} \\ C2v_{2,0} & C2v_{2,1} & C2v_{2,2} & C2v_{2,3} & C2v_{2,4} & C2v_{2,5} & C2v_{2,6} & C2v_{2,7} \\ C6v_{3,0} & C6v_{3,1} & C6v_{3,2} & C6v_{3,3} & C6v_{3,4} & C6v_{3,5} & C6v_{3,6} & C6v_{3,7} \\ C1v_{4,0} & C1v_{4,1} & C1v_{4,2} & C1v_{4,3} & C1v_{4,4} & C1v_{4,5} & C1v_{4,6} & C1v_{4,7} \\ C5v_{5,0} & C5v_{5,1} & C5v_{5,2} & C5v_{5,3} & C5v_{5,4} & C5v_{5,5} & C5v_{5,6} & C5v_{5,7} \\ C3v_{6,0} & C3v_{6,1} & C3v_{6,2} & C3v_{6,3} & C3v_{6,4} & C3v_{6,5} & C3v_{6,6} & C3v_{6,7} \\ C7v_{7,0} & C7v_{7,1} & C7v_{7,2} & C7v_{7,3} & C7v_{7,4} & C7v_{7,5} & C7v_{7,6} & C7v_{7,7} \end{bmatrix} = CV \tag{3.1}$$

where T is 1-D 8-point DCT matrix and $V=\{v_{i,j}\}$ is the intermediate result as shown in Fig. 5. In eq (3.1), C is diagonal matrix, whose diagonal components are $\{C4, C4, C2, C6, C1, C5, C3, C7\}$. Then, applying 1-D DCT on every row about the matrix F, the 8 x 8 final output matrix Y is also decomposed as follows,

$$Y=FT^T=\begin{bmatrix} C4C4h_{0,0} & C4C4h_{0,1} & C4C2h_{0,2} & C4C6h_{0,3} & C4C1h_{0,4} & C4C5h_{0,5} & C4C3h_{0,6} & C4C7h_{0,7} \\ C4C4h_{1,0} & C4C4h_{1,1} & C4C2h_{1,2} & C4C6h_{1,3} & C4C1h_{1,4} & C4C5h_{1,5} & C4C3h_{1,6} & C4C7h_{1,7} \\ C2C4h_{2,0} & C2C4h_{2,1} & C2C2h_{2,2} & C2C6h_{2,3} & C2C1h_{2,4} & C2C5h_{2,5} & C2C3h_{2,6} & C2C7h_{2,7} \\ C6C4h_{3,0} & C6C4h_{3,1} & C6C2h_{3,2} & C6C6h_{3,3} & C6C1h_{3,4} & C6C5h_{3,5} & C6C3h_{3,6} & C6C7h_{3,7} \\ C1C4h_{4,0} & C1C4h_{4,1} & C1C2h_{4,2} & C1C6h_{4,3} & C1C1h_{4,4} & C1C5h_{4,5} & C1C3h_{4,6} & C1C7h_{4,7} \\ C5C4h_{5,0} & C5C4h_{5,1} & C5C2h_{5,2} & C5C6h_{5,3} & C5C1h_{5,4} & C5C5h_{5,5} & C5C3h_{5,6} & C5C7h_{5,7} \\ C3C4h_{6,0} & C3C4h_{6,1} & C3C2h_{6,2} & C3C6h_{6,3} & C3C1h_{6,4} & C3C5h_{6,5} & C3C3h_{6,6} & C3C7h_{6,7} \\ C7C4h_{7,0} & C7C4h_{7,1} & C7C2h_{7,2} & C7C6h_{7,3} & C7C1h_{7,4} & C7C5h_{7,5} & C7C3h_{7,6} & C7C7h_{7,7} \end{bmatrix} = C^T HC \tag{3.2}$$

where $H=\{h_{i,j}\}$ is also the intermediate result of 1-D DCT of matrix V along every row. 2-D DCT of input matrix X is computed in three steps: First step computes V matrix from X matrix according to the flowgraph in Fig. 5. In second step, H matrix is computed from V matrix by the same way as first step. Finally, third step performs the multiplication of H matrix and pre-computed C parameters in eq (3.2). This three-step computation reduces total number of multiplications. The quantization matrix can be also absorbed in this look-up table of C parameter by scaling each component of these multiplicative constants. This modified Chen DCT requires only 388 cycles for 8 x 8 block including quantization because one multiplication and two additions are performed in one cycle on MVP ADSP. Similarly, for the IQ-IDCT, we use the following flowgraph as shown in Fig. 6, which needs 480 cycles for 8x8 IQ-IDCT. For MPEG-2 transform coding, 4 ADSPs are assigned to process DCT-Q and IQ-IDCT, respectively. The MPEG-2 transform coding is possible on our software architecture as shown in Table 3.

	F DCT-Q	IQ-IDCT
numbers of 8x8 block processing	388 cycles	480 cycles
numbers of 6 8x8 block processing(4:2:0)	2328 cycles	2880 cycles
required cycles for 720x480 frame	3142800 cycles	3888000 cycles
required time for 720x480 frame(40MHz)	78.57 ms	97.2ms
numbers of ADSPs for MPEG-2 real-time encoding	3	3

Table 3. The required cycles and numbers of ADSPs for MPEG-2 real-time encoding

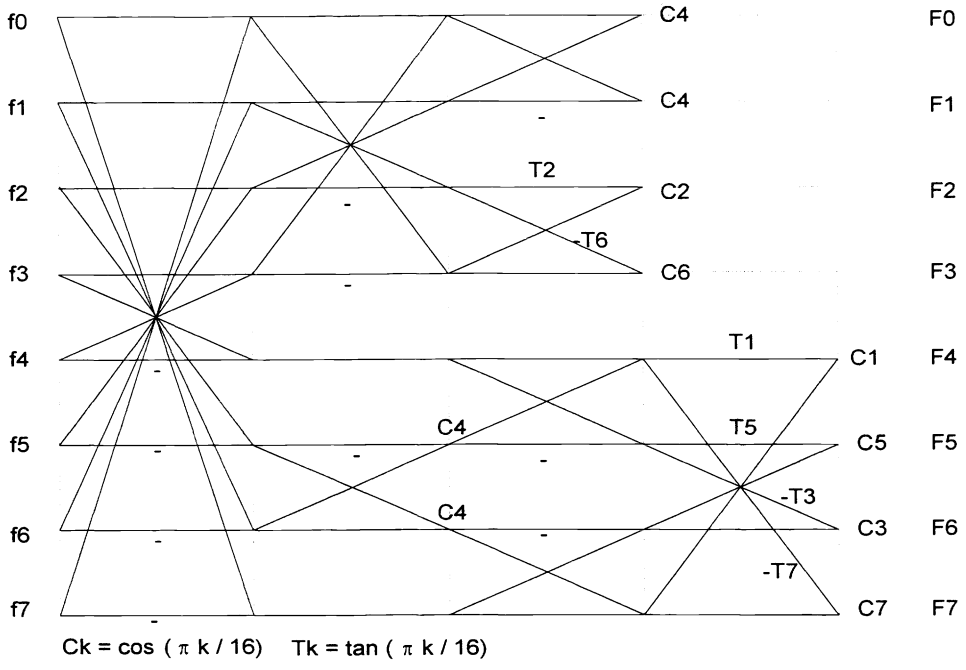

Fig. 5. The signal flowgraph for the scaled Chen 1-D DCT algorithm

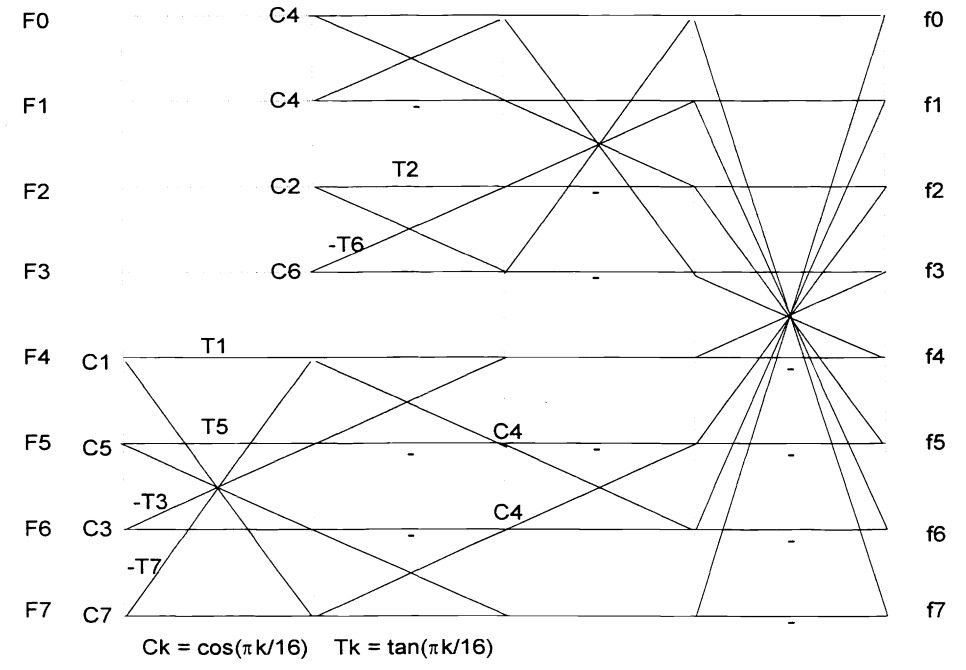

Fig. 6. The signal flowgraph for the scaled Chen 1-D IDCT algorithm

3.4. Variable length coding assigned to MVP2 ,4 MP

In the last stage, the host MVP generates the video bit-stream to be mixed up with fixed-length codes and variable-length codes that consist of mainly the run-level data of quantized DCT coefficients and other information associated with the macro block, such as the motion vectors, and prediction types. The quantized DCT coefficients are rearranged into one-dimensional array by scanning them as specified. This rearrangement puts the DC coefficient at the first location of the array and the remaining AC coefficients are arranged from low to high frequency, in both horizontal and vertical directions and higher frequencies are mostly zero. The rearranged array is coded into a sequence of the run-level data. The lossless encoding of the run-level data generated from the transform coding is assigned to MP of MVP2 and MVP4, whereas the host MVP encodes the macro block information generated from the motion estimation and motion compensation assigned to MVP1 and MVP3.

For the fast and simple search, the lookup-table, whose size is less than 2K bytes to assign to internal on-chip memory, is used to get a variable-length code. Since the longest VLC code is 17 bits including a sign bit, the table is 32-bit codes. The table is made for almost 1:1 mapping to the input data which is the output of the ADSPs of MVP2 and MVP4. Once two ADSPs finish DCT-Quantizer processing and the result is stored in DRAM0, MP processes VLC with DCT-Q result in DRAM0 and at the same time two ADSPs are processing the DCT-Quantizer with the input data in DRAM1. Figure 7 shows the memory map of on-chip memory. For computational efficiency, double buffering method is adopted for this memory map.

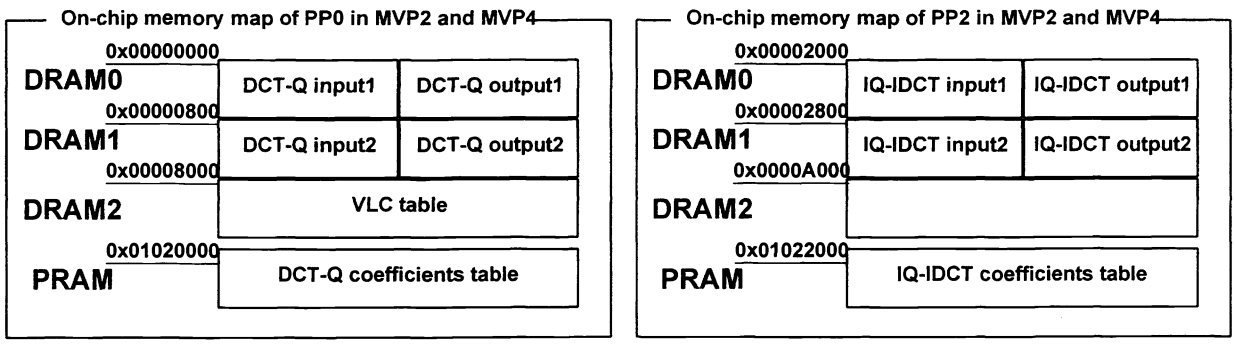

Fig. 7. On-chip memory of PP0 and PP2 for double buffering

4. CONCLUSIONS

This paper presents that the MPEG-2 encoding algorithm is mapped to multiple MVPs through the hybrid partitioning approach using both spatial and functional partitioning. The KICS is a multiprocessor system with five MVPs for MPEG-2 real-time encoder; one of them is used for system control and interface with host computer. There are two spatially partitioned processing clusters, each of which has two MVPs. One processing cluster processes the upper half of the video frame and the other processes the lower half of the video frame. Each MVP of the processing cluster performs different functions with balanced load, such as one MVP of a cluster carries out the forward and inverse discrete cosine transform (DCT) and the other performs the motion estimation through spatially partitioned ADSPs.

According to our simulation results, forward and inverse DCT's for 30 frames of 720x480 image need about 659 ms by using two MVPs and motion estimation processing by three-level hierarchical search with search range of ±16 requires two MVPs for MPEG-2 real time encoding. MPEG-2 audio compression is performed by the host MVP. Therefore, the KICS using five MVPs can perform the MPEG-2 real-time encoding.

5. REFERENCES

[1] V. Bhaskaran, K. Konstantinides, "Image and Video Compression Standards : Algorithms and Architectures," *Kluwer Academic Publishers*, Boston Dordrecht London, 1995.

[2] William B. Pennebaker, Joan L. Mitchell, "JPEG : Still Image Data Compression Standard,", Van Nostrand Reinhold, 1993.

[3] ISO/IEC JTC1, "Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to 1.5 Mbits/s," *MPEG-1 Committee Draft of Standard 11172, Int'l Organization for Standardization*, November 1991.

[4] ISO/IEC JTC1, "Generic Coding of Moving Pictures and Associated Audio," *MPEG-2 Committee Draft of Standard 13828, Int'l Organization for Standardization*, March 1995.

[5] Bouville, P. Houlier, J.L. Dubois, I. Marchal, B. Thebault, and M. Klefstad, "DVFLEX : A Flexible MPEG Real Time Video Codec," *ICIP96* vol.3 pp.829-832, 1996.

[6] J.H. Jeon, G.Y. Choi, H.Y. Kim, J.S. Kim, H.S. Kim, and H.W. Park, "KAIST Image Computing System (KICS): A Parallel Architecture for Real-Time Multimedia Platform using Multiple MVPs (TMS320C80)," *SPIE Medical Imaging 97*, 1997.

[7] Texas Instrument, Inc., "TMS320C80 (MVP) Master Processor : User's Guide," 1995.

[8] Texas Instrument, Inc., "TMS320C80 (MVP) Parallel Processor : User's Guide," 1995.

[9] Texas Instrument, Inc., "TMS320C80 (MVP) Transfer Controller : User's Guide," 1995.

[10] A.J. Jain, "Fundamentals of Digital Image Processing," *Prentice-Hall, Englewood Cliffs, N.J.*,1989.

[11] J. Jeong and W. Ahn, "Subpixel Accuracy motion estimation using a model for motion-compensated errors," *Picture Coding Symposium*. pp. 13.4, March 1993.

[12] B. G. Lee,"A new algorithm to compute the discrete cosine transform," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, no. 6, pp.1243-1245, Dec. 1984.

[13] W. H. Chen, C.H. Smith, and S. C. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-25, pp. 1004-1009, Sept. 1977.