

Load Balancing Method for Multiprocessor Image DSPs Using a Multistage Switching Network

YASUYUKI OKUMURA*, KAZUNARI IRIE* and RYOZO KISHIMOTO*

One of the greatest problems in realtime image processing with multiprocessor Digital Signal Processors (DSPs) is the dynamic balancing of an unevenly distributed load on certain processors at a certain time. This paper proposes a dynamic load balancing method for multiprocessor DSPs that can achieve Vector Quantization (VQ) with Motion Compensation (MC) in a multistage switching network. First, it is shown that at most 30% of all picture elements require VQ processes that consume large amounts of processing power. A multiprocessor scheduling method is then introduced that uses a multistage switching network to achieve load balancing. The proposed multiprocessor DSP can reduce the required processing power by using the load balancing method. Finally, packet switching and circuit switching are compared as operating modes for a multistage switching network, and it is shown that the processing power required for the former can be reduced to 20% less than that required for the latter.

1. Introduction

Digital video transmission system development has been triggered by the needs of video communication in areas such as TV conferences, CATV, and high-definition television (HDTV) along with the advances in high-speed digital transmission and broadband integrated services digital networks (ISDN) technologies using optical fibers [1]. Video coding technology plays a key role in the construction of broadband communication facilities. Various video coding algorithms have been developed, such as motion compensation (MC) [2], differential PCM (DPCM) [3], discrete cosine transformation (DCT) [4], and vector quantization (VQ) [5]. Video coding equipment should be flexible and programmable, because during its development, several video coding algorithms must be studied from the viewpoints of picture quality, coding efficiency, and cost. In addition, programmable video coding equipment must be economically adaptable to various service grades. For these reasons, much research has been done on digital signal processor (DSP) applications to various video processing methods including video coding [6-8].

Video coding necessitates the design of a multiprocessor architecture because of the immense processing load

involved. For example, processing power of between 1 GOPS (giga operations per second) and 10 GOPS is required for real-time HDTV video coding. One of the greatest difficulties in video coding multiprocessor systems is the dynamic balancing of an unevenly distributed load on certain processors at a certain time. A multistage switching network that provides a dynamic load balancing function would be very valuable.

This paper describes the results of simulation analysis of a multiprocessor DSP video coding system using a multistage switching network. First, taking a TV conference as an example, we show that a time sequential and spatial distribution of the processing power required for the video coding clarifies the problems in multiprocessor DSP video coding. Next, we introduce a dynamic load balancing method using a multistage switching network in the multiprocessor DSP and demonstrate its load balancing performance. Finally, we compare packet switching and circuit switching as operating modes of a multistage switching network from the viewpoint of the amount of buffer and processing power required.

2. Technical Problems of Multiprocessor DSPs in Video Coding

One of the greatest problems in video coding multiprocessor DSP systems is the load imbalance between DSPs. In the video signal coding process example using VQ in Fig. 1, pre-processing is normally used to

This is a translation of the paper that appeared originally in Japanese in Transaction of IPSJ, Vol. 31, No. 10 (1990), pp. 1513-1520.

*Transport Processing Laboratory, NTT Transmission Systems Laboratories, Room 1004C, 1-2356, Take, Yokosuka-shi, Kanagawa 238-03, Japan.

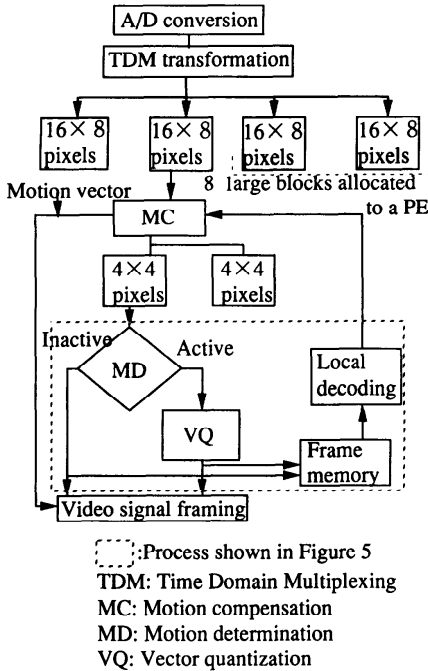


Fig. 1 Example of VQ video signal coding process.

reduce both the quantity of information and the number of calculation steps. This pre-processing involves determining whether or not to continue the current VQ by correlating a video frame and the previous frame. This is called motion determination (MD). In the first step of this example, the MC process is carried out as follows: First, the video frame is divided into small blocks, and then a motion vector is obtained for each block, to minimize its prediction error. In this example, the block size is 16 × 8 pixels, which is suitable

for the MC field. In the second step, based on the MD result, if the correlation between a video block and the previous block is small, the block is called “active,” and the coding proceeds to the VQ process. On the other hand, if the correlation is large, the block is called “inactive,” and the coding does not proceed to the VQ process. Before MD, the video block is again divided into 4 × 4-pixel blocks to fit into the VQ process. For a TV conference scene containing few moving objects, the processing load between multiprocessor DSPs must be balanced because the “active” blocks are unevenly distributed in the video frames and vary with time.

To clarify the above phenomena quantitatively, the multiprocessor DSPs’ video coding load density for a TV conference scene is analyzed from the viewpoint of the scene’s variation with space and time. In this analysis, the scene includes two persons who are standing, and the video frame (= 704 × 480 pixels) is divided into 8 × 8-pixel blocks. The block distance (BD) between each block in the current frame and the corresponding block of the previous frame is calculated as follows:

$$BD = \sqrt{\sum_{n=1}^N (X_n - Y_n)^2 / N}, \quad (1)$$

where

X_n : n -th pixel’s value in a block of a frame,

Y_n : n -th pixel’s value in the corresponding block of the previous frame,

N : number of pixels in a block (= 64 pixels).

If BD is greater than the preset threshold, the block is interpreted as “active” and forwarded to the next process. In the analysis, the threshold is set at 3, which is fairly small in comparison with each of the pixel values X_n and Y_n (from 0 to 255). The percentage of active blocks is then measured in the large block, which is formed by 9 × 10 blocks.

(1) Spatial distribution of video coding load

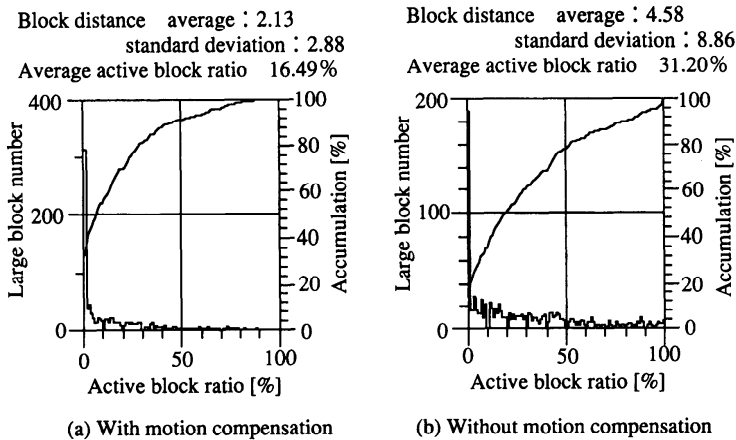


Fig. 2 Spatial distribution of video signal coding load (threshold: 3).

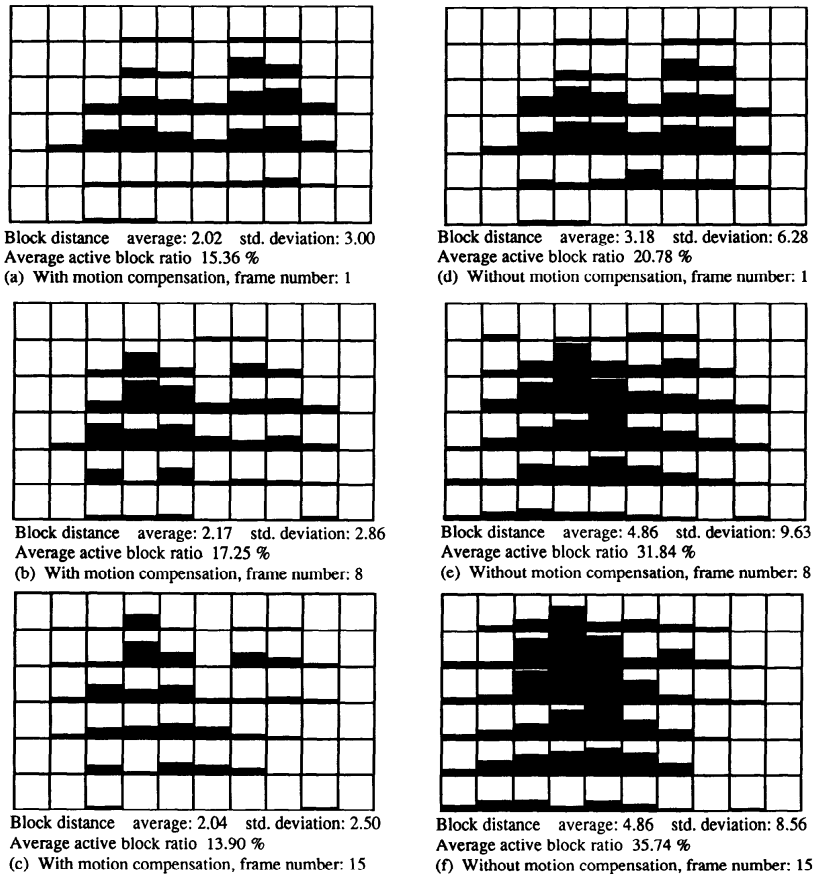


Fig. 3 Time sequential transition of video coding load (threshold: 3).

The spatial variation of the active block percentage in the video frame is shown in Figs. 2(a) and (b) with and without MC. The horizontal axis shows the percentage of active blocks, and the vertical axis shows the corresponding number of large blocks and the number of large blocks accumulated. The percentage of active blocks contained in almost all large blocks is less than 10%, although the active block percentage in the large blocks varies from 0 to 100%. Additionally, on average, the percentage is 16% in the example with MC before pre-processing, and 31% without MC. Therefore, the multiprocessor DSPs, each of which is assigned a fixed area of large blocks, require several times as much processing capability as those that are assigned equal loads.

(2) Time sequential transition of video coding load

The variation of the active block density with time is shown in Figs. 3(a) to (f) for each large block. In these figures, the large rectangle represents a video frame, and the small squares making up the video frame correspond to large blocks. For each large block, the black area shows the active block percentage; if a small

square is completely blackened, the percentage is 100%. The examples with MC before pre-processing are shown in Fig. 3(a) to (c), and those without MC are shown in (d) to (f). The number of frames increases with time, and therefore, time passes from (a) to (c). These figures show that the high active-block-density area moves from bottom to top in the video frame. Therefore, the large blocks must be assigned to each DSP dynamically for video coding.

3. Video Coding Scheduling Using Multiprocessor DSP

3.1 Multiprocessor Architecture Using Multistage Switching Network

We propose employing a multistage switching network for the distribution of an unevenly distributed load in the video frame to processor elements (PEs). The proposed multiprocessor architecture is shown in Fig. 4(a). The number of stages in the switching network depends on the number of PEs, in this example, it

is $\log_2 N$ when the number of PEs is N . In this paper, the minimum load distribution algorithm [10], which employs load information transfer links among the switching elements (SEs) and the PEs, is used to assign the load to the PE with the smallest load. The algorithm does not necessarily perform best when the data transfer overhead among PEs is large [11]. However, the video coding load, which contains a huge amount of pixels, can be broken down into large grain loads that require almost the same coding processes. Moreover, as can be in Section 2, at most 30% of the coding load must be transferred among PEs. Therefore, the data transfer overhead is expected to be small in comparison with the PEs' coding load, and the minimum load distribution algorithm is applied because of its simplicity and good performance.

A block diagram of an i -th stage SE than can execute the above-mentioned algorithm is shown in Fig. 4(b). This SE consists of a main data stream circuit and a control circuit. The main data stream circuit handles video signal data and consists of two buffer memories, two output circuits, a 2×2 switch, and an output control circuit. The control circuit determines the output port by comparing the loads on the SEs in the next stage. It is divided into two groups of circuits: one consists of the comparator and two $(i+1)$ -th stage load value memories, and the other of an i -th stage load value memory and two latch circuits. The first group compares the loads of the $(i+1)$ -th stage SEs, and the other group renews its own load value. When a SE receives a differential load value from the previous stage SE, it augments its own load value memory. When a SE sends data through one of its output ports to the next stage's SE, it also sends a differential load value to the destination SE. The minimum load value algorithm [9] is executed by these functions.

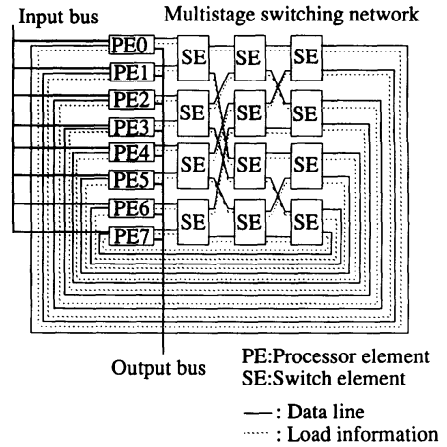
3.2 Multiprocessor DSP Scheduling

The multiprocessor DSP scheduling time chart for VQ is shown in Fig. 5. The operation time chart in this figure corresponds exactly to the VQ coding process shown in Fig. 1. This process can be described as follows:

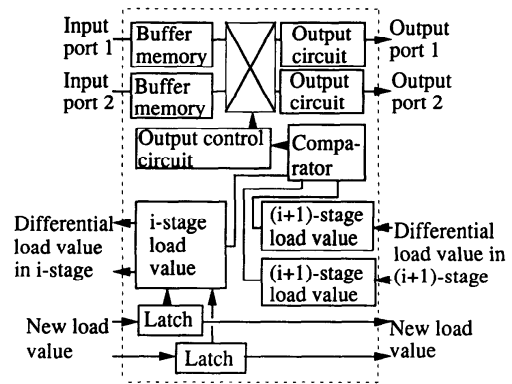
Step 1: The input video data is separated into large blocks that are allocated to various PEs through the input bus. Each PE executes the pre-processing (active/inactive evaluation).

Step 2: During the pre-processing, the PEs select large blocks containing active blocks from those allocated and then transfer the active large blocks to the multistage switching network. The transferred large blocks are given the address numbers of the PEs to which they were initially allocated. The switching network conveys the large blocks to PEs with smaller loads.

Step 3: A PE encodes the received large blocks, using the VQ algorithm, and then transfers them into the switching network again. The switching network feeds



(a) Multiprocessor configuration



(b) Switch element configuration for minimum load distribution

Fig. 4 Multiprocessor and switch element configuration.

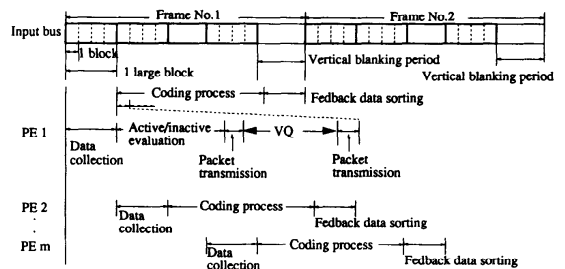


Fig. 5 Multiprocessor scheduling for video signal coding.

the encoded large blocks back to their originally allocated PEs according to the address numbers.

Step 4: The original PEs finally form the video signal frame from the fed back blocks and the inactive blocks.

Then, they start to transmit the video signal into the output bus from PE#1 in the same order as the data blocks were received from the input bus.

3.3 Multistage Switching Network Mode

There are two operating modes for the switching network:

Packet Switching Mode: The SE contains buffer memories in the same configuration as in Fig. 4(b). It stores the received data into the memories and forwards them asynchronously to the SEs in the next stage. This switching mode is shown in Fig. 6(a). If the data packets received from both input ports are routed to the same output port, they are stored in the buffer memories to avoid any contention.

Circuit Switching Mode: The PEs contain buffer memories at the input and output ports. They begin transmitting the data after the switching network has established the route. If the data packets received from both input ports are routed to the same output port in an SE, one of them is routed to the output port. The originally transmitting PE is informed of any rejected data, as shown in Fig. 6(b). Each PE stores the data until data transmission is successfully completed. If the PE is informed that data have been rejected, it re-attempts to transmit them. The SE configuration is the same as in Fig. 4(b), except that it contains a circuit for informing PEs of data rejection instead of buffer memories.

In the circuit-switching mode, the amount of hardware making up the multiprocessor system can be reduced, because no SEs contain buffer memories. However, the fact that data transmission is continuously retried until it is successfully completed may adversely affect the throughput.

4. Required Processor Power and Buffer Memory Capacity

4.1 Simulation Methods

The required processor power was evaluated for both switching network modes, using computer simulation. The parameters used in the simulation are summarized in Table 1. The constant initial load and time-variant initial load were allocated to PEs as shown in Figs. 7(a) and (b). For both initial conditions, some of the PEs were allocated only active blocks, and the others were allocated only inactive blocks, on the basis of the results presented in Section 2.

The simulation was performed according to the following procedures. First, the number of PEs, N , with only active blocks was calculated from the active block ratio, which is defined as the ratio to all PEs of those initially allocated active blocks. In the constant initial load model, the PEs in the center of the 32-processor line were allocated the active blocks. In the variant initial load model, the active blocks were initially

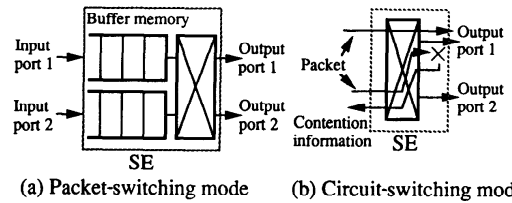


Fig. 6 Data transmission methods.

Table 1 Simulation parameters.

Item	Value	Item	Value
PE number	32	Frame cycle	30 frames/sec
PE processing power	33 ~ 100 MOPS	Pixel number	1.651×10^6 /frame
Switch element transmission speed	33 ~ 100 Mbps	Required operations for A/I*	80 operations
1 block	4*4 pixels	Required operations for VQ	1270 operations
Large block	31 blocks	Simulation clock	100 kHz (= large block period)
Area allocated to a PE	104 large blocks	1 frame length	3328 (= 104×32) large blocks

*A/I: Active/inactive evaluation

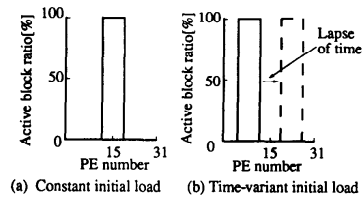


Fig. 7 Input load model for simulation.

allocated from PE#1 to PE# N , where N was calculated from the active block ratio. Those PEs then moved to adjacent PEs after each frame period. Each PE started the coding processes at a different time, as shown in Fig. 5.

After the pre-processing (active/inactive evaluation) steps shown in Table 1, the PEs transferred only the active large blocks to the switching network. All the PEs stored the received video data in their buffer memories and transferred them to the switching network again after the VQ process steps shown in Table 1. Then, the number of steps from the time of the PEs' data acquisition to the time of their receiving coded data was counted. This step number is the required processing time. However, the time required to form a video signal frame from the fed back data blocks was ignored because it was extremely short in comparison with the other processes.

4.2 Multiprocessor DSP Performance

The maximum processing delay of the multiprocessor DSP relative to the active block ratio when the initial processing load is constant is shown in Fig. 8(a), and the maximum processing delay for the time-variant initial load is shown in Fig. 8(b). The input load models are assumed to be as in Figs. 7(a) and (b). The maximum processing delay increases along with the active block ratio in both constant initial load and time-variant load models and in both packet- and circuit-switching networks. It increases abruptly when it exceeds 3328 clocks, which is equal to one frame period of the video signal, as shown in Table 1. This abrupt increase occurs because the amount of incoming data surpasses the processing ability of the multiprocessor DSP.

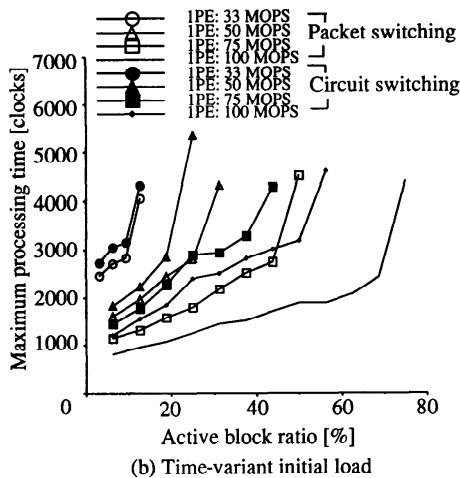
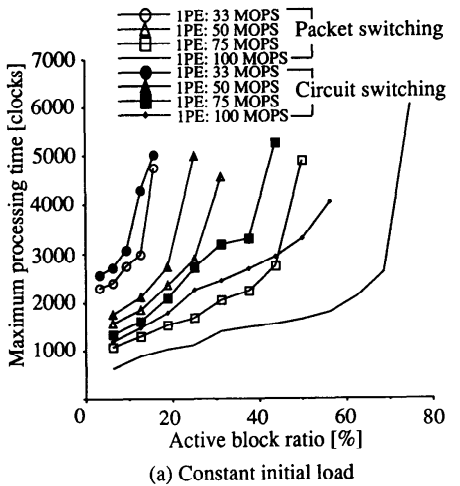


Fig. 8 Maximum processing time characteristics.

4.3 Required Multiprocessor DSP Power

The multiprocessor DSP must complete the processing of a frame within one frame period, that is to say, before the next frame of the video signal reaches it. This condition corresponds to the 3328 clock limit on the maximum processing delay, as explained in Section 4.2. This requirement and Fig. 8 suggest that the multiprocessor DSP power required corresponds to the active block ratio shown in Figs. 9(a) and (b). In both cases, the proposed multiprocessor DSP can reduce the required processing power by using the load-balancing method. Furthermore, the packet-switching mode is superior to the circuit-switching mode in reducing the required processing power, achieving a 20% reduction when the active block ratio is around 20% to 30% as in TV conferences. The extra processing power required in the circuit-switching mode must be due to the drop in transmission efficiency that occurs when the PEs continue to re-attempt to transmit data until they complete the task.

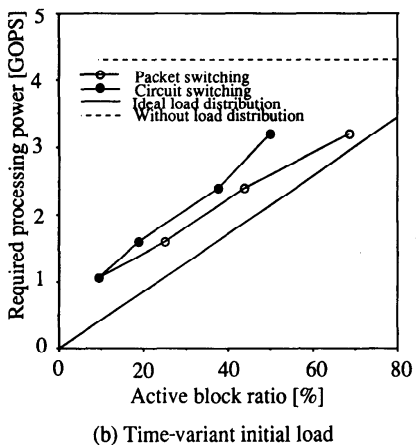
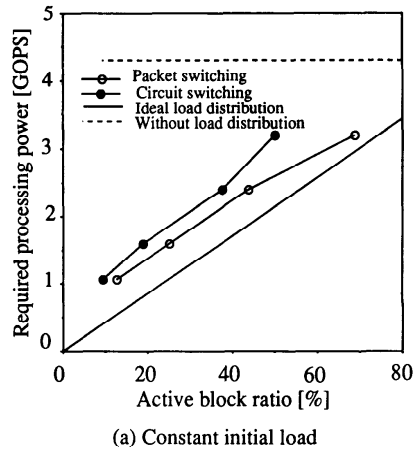


Fig. 9 Required processing power characteristics.

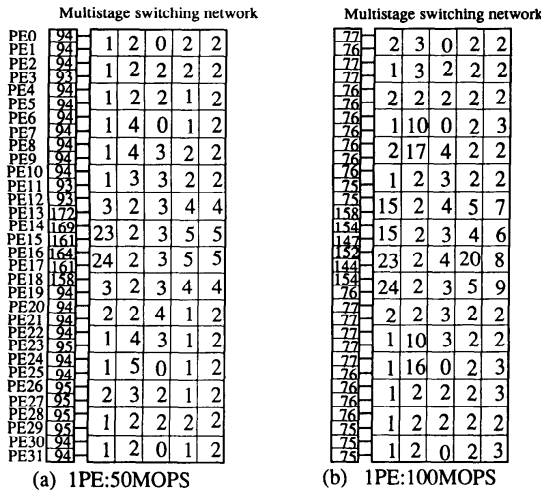


Fig. 10 Buffer memory consumption in the packet-switching network.

4.4 Required Buffer Memory Capacity

The maximum number of data packets residing in the SEs and PEs over a period of time was calculated in order to evaluate the required buffer memory capacity in both switching modes. The maximum number in each SE and PE of the packet-switching mode is shown in Fig. 10, and the number in each PE of the circuit-switching mode is shown in Fig. 11. In both figures, the input load model was the same as in Fig. 7(a). The PE processing power is 50 MOPS in Fig. 10(a), and 100 MOPS in Fig. 10(b). In these figures, the maximum number of data packets is large in the PEs in the middle of the row because the active blocks are assigned to those PEs initially. It is also large in the SEs in the middle of the first-stage row, but the data packets are distributed almost equally among the SEs in the other stages. In Fig. 11, the number is large in the middle PEs, for the same reason as above.

The required amount of buffer memory capacity for both switching modes is shown in Fig. 12, based on the results in Figs. 10 and 11. It was calculated on the assumption that all SEs have the same configuration and contain buffer memory whose capacity is equal to the maximum number of data packets residing in all SEs. The same assumption applies to the PEs. As seen in Fig. 12, the amount of buffer memory capacity required for the circuit-switching mode falls below the amount required for the packet-switching mode. The difference in the capacity is 15% if each PE's processing power is 100 MOPS and the active block ratio is around 20% to 30%. The increased amount of buffer memory in the packet-switching mode must be due to it having SEs with distributed buffer memories whose capacity is the maximum number of data packets residing in all SEs.

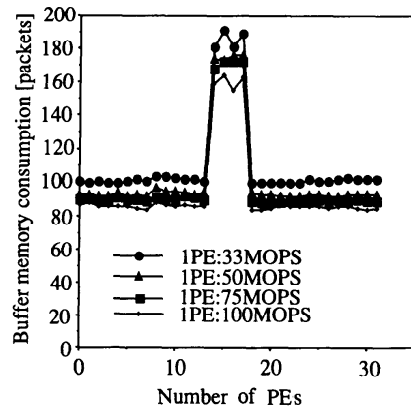


Fig. 11 Buffer memory consumption in circuit-switching mode.

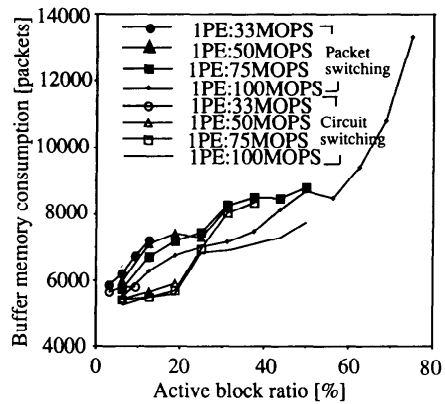


Fig. 12 Required buffer memory amount.

If the input load model in Fig. 7(b) is used, the range of PEs and SEs with a large number of data packets broadens to include those initially assigned the active blocks. However, the difference in the amount of buffer memory capacity required for packet-switching and circuit-switching modes is almost the same as that in the input load model of Fig. 7(a).

4.5 Discussion

As stated above, the amount of buffer memory required for the packet-switching mode is 15% larger than that required for the circuit-switching mode, but the former requires 20% less processing power than the latter. To encode HDTV signals, which currently contain the largest number of pixels, the buffer memory of the packet-switching mode must be increased by 100 Mbits. This can be accomplished by adding twenty-five 4-Mbit RAMs. This overhead is negligible, considering the total amount of hardware involved. The circuit-switching mode also requires additional overhead in the form of a circuit for informing the packet data rejection.

tion. In the design of multiprocessor DSPs, these two overheads tend to cancel each other out. Therefore, the packet-switching mode is superior to the circuit-switching mode because packet switching offers higher performance.

5. Conclusion

A new load-balancing method has been proposed for multiprocessor DSPs in order to solve one of the greatest difficulties in video coding: load concentration on certain processor elements. Using computer simulation, the processing power required for the new multiprocessor DSP is shown to be reduced to about half that of conventional multiprocessor DSPs when the active block ratio is around 30%. This paper has also clarified the relation between the multistage switching network mode and the load-balancing efficiency. As a result, the amount of buffer memory required for the packet-switching mode is larger than that required for the circuit-switching mode by a negligible 15%. However, the processing power required for the former can be reduced to 20% less than for the latter. The hardware implementation of the proposed multiprocessor DSP will be undertaken in the future.

Acknowledgement

The authors would like to thank Dr. Masaki Koyama and Dr. Haruo Yamaguchi of NTT Transmission Systems Laboratories for their participation in several fruitful discussions on broadband transmission, and for their overall guidance.

References

1. ARMBRUSTER, H. Broadband ISDN Satisfies Increasing Telecommunications Needs, *Telecom Report*, 9 (1986), 238-246.
2. NETRAVALI, A. N. et al. Motion-compensated Television Coding: Part I, *Bell Syst. Tech. J.*, 58 (1979), 631-670.
3. KOGA, T. et al. A Study on Inter/Intra-frame Adaptive Prediction Coding of TV Signals (in Japanese), *Technical Report of IEICE*, CS 80-79 (1980), 63-70.
4. CHEN, W. H. et al. Scene Adaptive Coder, *IEEE Trans. Comm.*, COM-32 (1984), 225-232.
5. GOLDBERG, M. et al. Image Sequence Coding Using Vector Quantization, *IEEE Trans. Comm.*, COM-34 (1986), 703-710.
6. NISHITANI, T. et al. Video Signal Processor Configuration by Multiprocessor Architecture, *Proc. ICASSP* (1986), 797-800.
7. ITOH, T. et al. A 64 kbps Motion Picture Coder, *Proc. GLOBECOM* (1986), 63-67.
8. LIU, W. et al. A Class of Parallel/Pipeline Architectures for Realtime Image Application, *Proc. SPIE Visual Comm. and Image Processing II* (1987), 329-336.
9. KOGA, T. et al. Motion-compensated Interframe Coding of Conference TV Signals (in Japanese), *Technical Report of IEICE*, IE 81-54 (1981), 85-90.
10. KISHIMOTO, R. et al. Self-routing Benes Network Configuration Using Dynamic Load Distribution (in Japanese), *Trans. IEICE Japan*, J72-B-1 (1989), 420-428.
11. SUGIE, M. et al. Load-dispatching Strategy on Parallel Inference Machines, *Proc. International Conference on Fifth Generation Computer Systems* (1988), 987-993.