A HIGHLY-PARALLEL DSP ARCHITECTURE FOR IMAGE RECOGNITION

Hiroyuki Kawai, Yoshitugu Inoue, Robert Streitenberger
and Masahiko Yoshimoto

System LSI Labs., Mitsubishi Electric Corp.
4-1, Mizuhara, Itami, Hyogo, 664, Japan

ABSTRACT

This paper presents the architecture of a newly developed highly parallel DSP suited for realtime image recognition. The programmable DSP was designed for a variety of image recognition systems, such as computer vision systems, character recognition systems and others. The DSP consists of optimized functional units for image recognition: SIMD processing core, a hierarchical bus, Address Generation Unit, Data Memories, DMAC, Link Unit, and Control Unit. The DSP can process a 5x5 spatial filtering for 512x512 images within 13.1msec. Adopting the DSP to a Japanease character recognition system, the speed of 924characters/sec can be achieved for feature extractions and feature vectors matchings. The DSP can be integrated in a 14.5x14.5mm^2 single-chip, using 0.5um CMOS technology. In this paper, the key features of the architecture and the new techniques enabling efficient operation of the eight parallel processing units are described. Estimation of the performance of the DSP is also presented.

1. INTRODUCTION

Image recognition can be performed by three steps: pre-processing, feature extraction, and matching. The pre-processing is realized by convolution between a local image data and a coefficient matrix, which requires a very large amount of calculations, e.g. during a spatial filtering for noise reduction. In the second step, features (e.g. histogram, area, moment) of the pre-processed image are extracted. This operation is accelerated by using conditional branch instructions. In the third step, similarities between an input image and templates are measured and some of the templates are selected by sorting the respective similarities. The amount of the calculations in this matching operation depends on the number of the templates and the dimensions of the feature vector. A Japanease character recognition system causes approximately 0.8M distance calculations for recognizing one character.

Conventional image processing LSIs are classified into two approaches. One is a hardwired approach. The other is a programmable DSP approach, which includes three subgroups according to Flynn's criterion: i.e., SISD, SIMD, and MIMD. The major application of conventional hardwired LSIs[1][2] is restricted to the realtime pre-processing of images due to the lack of flexibility. Therefore, a different algorithm needs to introduce other appropriate LSIs onto their boards, and system designers suffer from long redesign time whenever one of the system functions must be changed.

A lot of conventional programmable DSPs feature the SISD-type [3][4]. They have high flexibility, but do not have enough processing power to carry out the realtime pre-processing, etc.. Parallel processing DSPs, which belong to either the SIMD-group or the MIMD-group, have been introduced in order to overcome this problem of the SISD-type DSPs. Unfortunately, their data handling functions are suited only for local image processing[5][6][7], so they can not execute efficiently matching operation. No DSP which had enough capabilities of both the processing power and the flexibility to achieve realtime image recognition had been presented.

In order to overcome the above problems, a new architecture of a programmable DSP has been investigated. It incorporates eight parallel/pipelined processing units operating in the SIMD manner and also supports a flexible data handling capability. Also, its performance and its die size has been estimated, under the assumption of using 0.5um CMOS process technology. Adopting this architecture, a DSP architecture, which is over 8 times more powerful than conventional image signal DSPs, was realized. A Jpanease character recognition system using this single DSP can achieve a speed of 924characters/sec. This paper describes the architectural features of the proposed DSP, the Image Recognition Engine (IRE). In the second section, the structure of the SIMD processing core and the techniques of the flexible data handling are explained. In the third section, the newly developed technique for an efficient conditional branch operation is addressed. Finally the benchmark results using typical image processing algorithms are presented.

2. ARCHITECTURE

The key feature of the DSP is the inclusion of the SIMD processing core with eight pipelined processing units. The block diagram of the DSP is shown in Fig.1. Internal clocks of the DSP, which are five times as fast as a system clock, are generated by the PLL, especially for executing a spatial filtering with horizontal size of five pixels. The main issue is how to conduct the execution of the eight processing units effectively for various algorithms, including image pre-processing, feature extraction, and matching operation.

0-7803-2431-5/95 $4.00

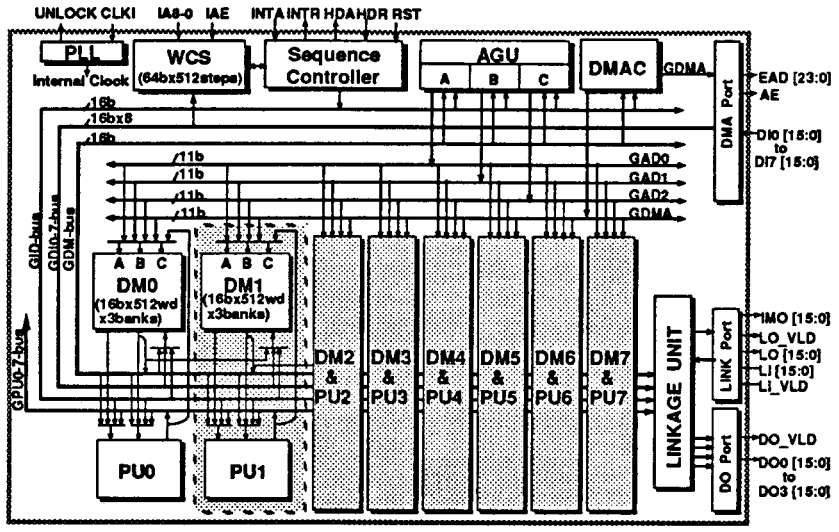

Fig. 1 Block diagram of DSP

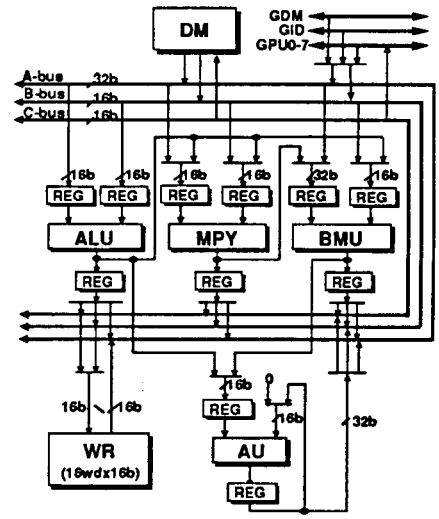

Fig. 2 Block diagram of PU

2.1 SIMD Processing Core

The IRE includes eight parallel processing units (PU0-7) organized to the SIMD processing core. Each processing unit consists of an Arithmetic Logical Unit (ALU,16bit), a Multiplier (MPY,16bitx16bit -> 31bit), a Bit Manipulation Unit (BMU,16bit), an Arithmetic Unit (AU,16bit), and Working Registers (WR,16bitx16wd) as shown in Fig.2. The number of the pipeline stages in PU is determined by instructions. These functional units are optimized for image processing, allowing that thirteen complex combinational operations are possible. The BMU has been introduced especially for binary image processing. The BMU supports bit-shift operations, logical operations and bit count operations. It can also execute either logical-operation/shift-operation or logical-operation/ set-bit-count in one cycle, in order to enhance the binary image processing capability. Hence, the SIMD processing core achieves the performance of 3.2GOPS at 100MHz operation, which is a 8 times higher than that of the previous parallel DSP[7] and is enough performance to process a heavy task in image processing.

2.2 Bus Structure

A hierarchical bus structure is introduced as illustrated in Fig.1. The buses in the IRE are classified into two groups: global buses and local buses.

The global buses are GPU0-7, GDM, GID and GDI0-7. The global buses can support three kinds of data transfer modes: a shifting, a broadcasting, and a DMA transfer. GPU0-7 are eight 16bit buses sustaining the data shift operations among the PUs. By issuing an instruction including the shift value (n), the i-th PU receives the output data of the {(i+n)MOD8}th PU via the GPU-buses. The GDM-bus(16bit) and the GID-bus(16bit) are implemented for supporting the broadcast operation. The GDM-bus is a single 16bit bus for broadcasting data from the selected data memory. The GID-bus is for transferring immediate data. GDI0-7 are eight 16bit DMA-buses, which deliver external data to the corresponding data memories. The GDI0-bus can also be utilized to transfer raster scanning image data to the DM0.

The local buses in each PU consist of an A-bus(32bit), a B-bus(16bit) and a C-bus(16bit). The A-bus and the B-bus are for the source data transfer. Both A-bus and B-bus can be connected to GPU, GID, GDM and the corresponding DM by the selector. The C-bus transfers the destination data. The C-bus is directly connected with the corresponding DM. This hierarchical bus structure provides the IRE with the elevation of both the flexibility and the efficiency.

2.3 Address Generation Unit

The Address generation unit (AGU) consists of three units (AGUA, AGUB, and AGUC) for generating two source addresses and one destination address. The AGUB also supports a special addressing mode for the preparation of local image data without irregular codes. In this mode, the AGUB generates addresses successively by increment operations until the number of addresses becomes equal to the horizontal size of the local image. The AGUB executes the above function synchronously with a DMA transfer in order to prohibit a on-chip DMA controller (DMAC) from overwriting to the DMs. A start address of the next local image is also generated automatically by the AGUB by incrementing the start address of the previous local image. This mode enables the IRE to execute a spatial filtering with the horizontal size of five pixels within the time interval (five cycles) of the DMA transfers, because any instructions are not needed for modifying address registers in the AGUB.

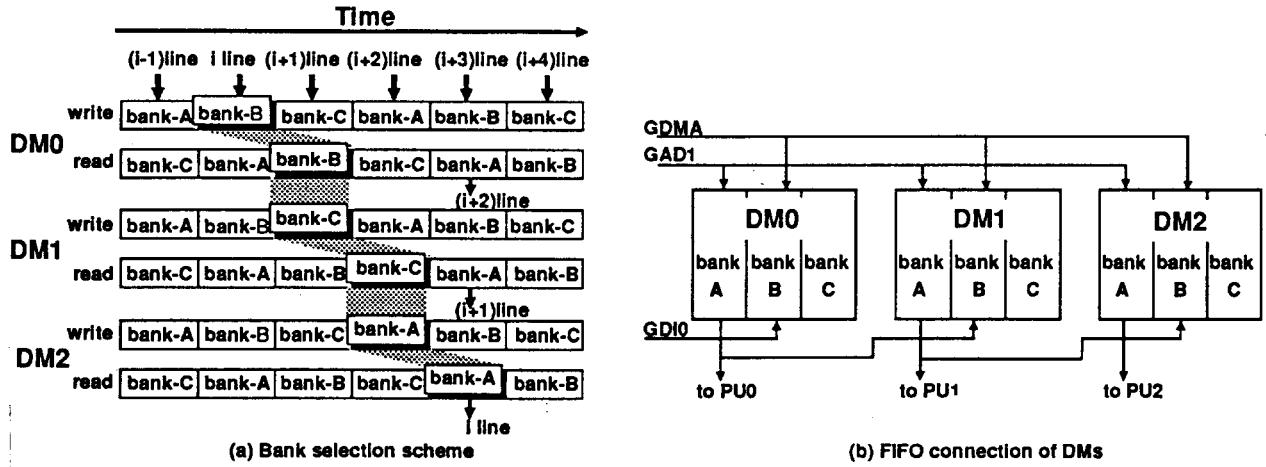

Fig. 3 FIFO operation in DMs

2.4 Data Memories

The Data Memories (DM0-7) serve as local memories combined with PUs. Each DM consists of three banks of a single port RAM (16bitx512wd). The external I/Os are designed so as to transfer any data every five machine cycles, because the external data transfer is not as fast as the internal operations. Therefore both the multiple-banks configuration of the DM and the concurrent DMAC are techniques indispensable for performing a seamless stream of the data transfer. The DM supplies 16-bit data to the corresponding PU via two direct pathes. Three common address buses (GAD0-2), a DMA address bus(GDMA), and the C-bus are connected to an address port of each DM. A table look up operation is achieved by selecting the C-bus as the address bus. The GID-bus, the one of the GDI-buses, and the output of the neighboring DM are connected to an input port of each DM. When a mode register is set to a FIFO mode, DMs behave as FIFO memories. The maximum number of horizontal pixels is 1536pixels (512x3). In the FIFO mode, the data read from a neighboring DM is selected except for the DM0 and written into the DM according to an address on the GDMA-bus. The DM0 can get the external input data via the GDI0-bus. The sequence of the bank access for each DM in the FIFO mode is illustrated in Fig.3. This FIFO operation enables the IRE to extract a local image out of raster-scanned input data in realtime.

2.5 Link Unit

Functions such as an accumulation and a sorting of PUs' outputs need inter-PU data transfers and cause degradation of the SIMD- core's performance. A linkage unit (LU) was introduced as a postprocessing unit handling the outputs of the PU0-7 without the inter-PU transfers. The LU supports four functions: accumulation, min., max., and sorting. The LU consists of a 16bit ALU, a 16bit counter, and two 16bitx128wd register files. The LU processes all or some of the nine data which consist of the eight output data of the PU0-7 and the link input data from an external data port. The LU is indespensable to execute a spatial filter in realtime and to realize high speed matching operations.

2.6 Control Unit

A controller (CU) consists of a sequence controller and an 512stepsx64bit writable control store (WCS). A new technique for controlling the activation of the PU0-7 has been developed. When a conditional branch instruction is fetched, the CU issues a test condition code to all PUs. Then the condition test is accomplished in each PU concurrently. The test result is referred to for making a judgment whether the following instructions have to be executed by the respective PU. PUs are given the permission to execute the following IF-process only if the respective result is high (true). After the IF-process, the permission for the ELSE-process is issued to the false PUs. Another feature of this technique is to skip over either the IF-process or the ELSE-process, if all PUs are in a same condition. Hence, our approach results in the reduction of 13.5% machine cycle overhead with a few additional circuitry. Thus, this technique is efficient to achieve a feature extraction, which requires many conditional branch operations.

3. INSTRUCTION

The 64bit horizontal micro-instructions stored into the WCS is categorized into four groups: (1) system operations, (2) load/move operations, (3) sequence control operations, and (4) arithmetic operations. The sequence control includes the conditional branch operation, the jump operations, and a new block repeat operation. The block repeat operation is a modified hardware loop function. The superiority of this scheme against the conventional hardware loops is the flexibility of the coding. The block repeat instruction has bit fields of both the block start address and the block end address. Hence, it is not necessary that this instruction is placed on one line above the first line of the repeated block.

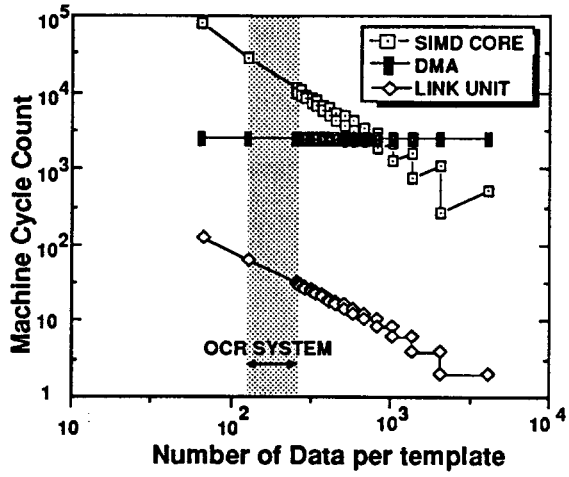

Fig.4 Relation between machine cycle count and size of a template

This block repeat proves itself to be competent whenever a time-restricted algorithm like the spatial filtering is coded with this set of the instructions. The arithmetic instructions include thirteen combinational instructions which have been chosen by the simulations with seventeen sample programs: spatial filters, feature extractions, and matchings with feature vectors.

4. PERFORMANCE

4.1 Performance Evaluation

The performance of the IRE has been evaluated by simulating the Verilog-HDL models with sample programs, such as pre-processing, feature extractions, and matchings.

The relation between the machine cycle count required and the size of a template for the SIMD-core, the DMAC, and the LU is shown in Fig.4. The machine cycle count has been evaluated on a condition that all data stored in one bank of each DM is processed. Similarity between an input data and a template is calculated by the eight pipelined PUs in parallel. Output data of the SIMD core is subsequently sorted in the LU. The DMAC transfers new template data from external data memory to each DM concurrently with the operations of both the PUs and the LU. The total performance is limited by the maximum among cycle counts required in these three units at each size of a template. This result proves that this architecture is free from external I/O bottleneck until the size of a template exceeds 681. This size of the template is far over the range of present OCR systems.

The pipelining in the IRE is designed to ensure the operation over 100MHz. Assuming that the cycle time is 10nsec, the results of the performance evaluations are listed in Table1. The IRE can process a 5x5 spatial filtering for 512x512 images within 13.1msec. The IRE can make a histogram of a 512x512pixels image within 1.7msec. This performance is about thirty-seven times higher than that of the conventional DSP[7]. Adopting the DSP to a Japanease character recognition system with 128 feature vectors per template and 3584 templates, the speed of 924characters/sec can be achieved, as the performance of feature extractions and feature vectors matchings.

Table 1 Performance

Application	Performance
■ Preprocessing	
- 5*5 spatial filtering	13.1 msec / (512x512 image)
- Edge detection	
3*3 Kirsch(8-templates)	7.9msec / (512x512 image)
3*3 Laplacian	7.9msec / (512x512 image)
■ Feature Extraction	
- PHL	0.3msec / (48x48image)
- LDC	0.1 msec / (48x48image)
- Area (without Labeling)	2.0msec / (512x512image)
- Histgram	1.7msec / (512x512image)
- Filet	2.6msec / (512x512image)
- Moment	5.6msec / (512x512image)
- DCT (8x8)	6.7msec / (512x512image)
■ Character Recognition	
Japanease Character Recognition with Feature vector matching	924characters/sec (128-data/template,3548templates)

4.2 VLSI Specification

The IRE integrates approximately 1.9M transistors, which includes 490K transistors for logic and 1.4M transistors for RAMs. Adopting 0.5um CMOS process technology, the IRE can be integrated in 14.5x14.5mm^2 die.

5. CONCLUSION

The architectural features and the control scheme of the IRE suitable to image recognition systems have been described in this paper. The proposed DSP incorporates the powerful and flexible highly parallel-pipelined processing core with the capability of the seamless stream of the data. The adaptiveactivation of the parallel processing units and the block repeat operation enhance the programmability of the DSP and thereby support different kinds of image processing. The performance of an image recognition system is drastically increased by applying this DSP.

REFERENCES

[1] P. Ruetz and R. Brodersen,"A realtime image processing chip set,"in IEEE ISSCC Dig. Tech. Papers, Feb.1986, pp148-149.

[2] J. Norsworthy et al., "A parallel image processor chip," in IEEE ISSCC Dig. Tech. Papers, Feb. 1988, pp158-159.

[3] K. Aono et al., "A realtime image signal processor (version II) with micro-programmable and expandable architecture," in Proc ESSCIRC'86, Sept. 1986, pp98-100.

[4] T. Oto et al., "A NEW DSP ARCHITECTURE SUITED FOR IMAGE ANALYSIS," Proc. of ICASSP'91, pp1181-1184.

[5] T. Fukushima et al., "An image signal processor," in IEEE ISSCC Dig. Tech. Papers, Feb. 1983, pp258-259.

[6] Y. Kobayashi et al., "A BiCMOS Image Signal Processor," in IEEE ISSCC Dig. Tech. Papers, Feb. 1987, pp182-183.

[7] M. Maruyama et al., "A 200MIPS Image Signal Multiprocessor on a Single Chip," in IEEE ISSCC Dig. Tech. Papers. Feb. 1990. pp122-123.