MVA2002 IAPR Workshop on Machine Vision Applications, Dec. 11 - 13, 2002, Nara - ken New Public Hall, Nara, Japan

15—5

A 51.2GOPS Programmable Video Recognition Processor for Vision based Intelligent Cruise Control Applications

Shorin KYO*
Multimedia Research Laboratories
NEC Corporation

Abstract

This paper describes a 51.2 GOPS video recognition processor that provides a cost effective device solution for vision based ICC (Intelligent Cruise Control) applications. By integrating 128 4-way VLIW processing elements into a single chip based on a SIMD linear array architecture, and operating in 100 MHz, the processor achieves to provide a computation power enough for a weather robust lane mark and vehicle detection function written in a high level programming language, to run in video rate.

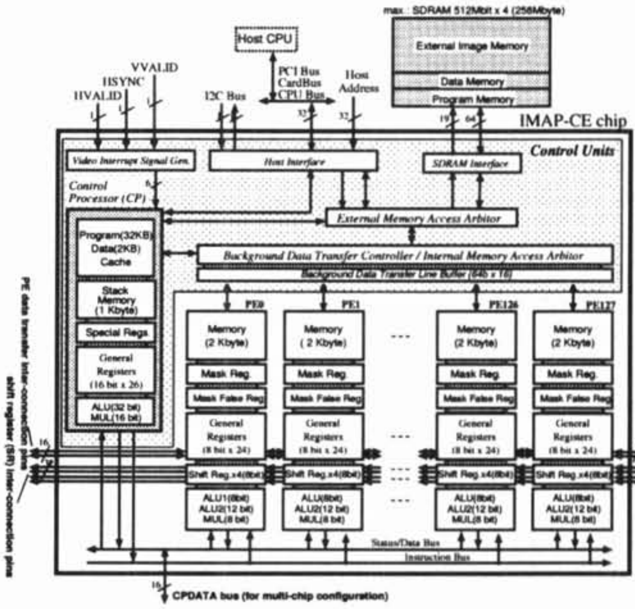
1 Introduction

Video recognition for timing crucial ICC applications of the ITS (Intelligent Transport System) fields requires not only high computation performance but also high programmability due to the necessity of using various recognition algorithms for coping with change of situations and change of sizes and appearances of target objects such as vehicles, lane marks and obstacles. Existing microprocessors and DSPs provide high programmability but fail to offer enough performance due to the lack of sufficient architectural support for the required pixel parallelism and non-continuous memory access, while dedicated hardware such as ASICs and FPGAs, although can provide sufficient computation performance, have to devote die spaces to each particular recognition algorithm prepared for coping with exclusively existing situations such as driving at highway versus local streets, or stop-and-go versus normal cruise. An integrated memory array processor for car electronics (IMAP-CE) characterized by the following 1) to 4) provides both high performance and high programmability for ICC applications. 1) parallel execution by a multiple of 128 SIMD 4-way VLIW processing elements (PEs), 2) parallel algorithm design based on a linear and loop back PE connection and the ability of simultaneous access of data located in different memory address by each PE (indexed addressing), 3) automatic background video data mapping to each PE by an asynchronous shift register (SR) structure, 4) an optimizing extended C compiler that generates fully efficient codes, together with a user friendly visual programming tool for partitioning computation between host PC and IMAP-CE.

*Address: 4-1-1 Miyazaki, Miyamae-Ku, Kawasaki, 216-8555, Japan. E-mail: s-kyo@cq.jp.nec.com

2 Hardware Design Features

IMAP-CE is based on a SIMD processor array architecture, which is supposed to be one of the most powerful parallel architecture for parallel image processing[1][2], and which enables the design of a compact and high performance system. Figure 1 shows both the chip specifications and the block diagram. The chip is a fourth generation of a series of SIMD linear processor array designed by the author's group[3]-[5]. Beside integrating a SIMD processor array of 128 8bit PEs in a single chip, standard external bus interfaces such as PCI and I2C, a CPU bus interface for NEC V series microprocessors, and a SDRAM controller are also unified. These external interfaces facilitate IMAP-CE to be used as a compact video recognition engine for embedded systems.


max : SDRAM 512Mbit x 4 (256Mbyte)
Host CPU
External Image Memory
Data Memory
Program Memory
VVALID
HSYNC
HVALID
I2C Bus
PCI Bus
CardBus
CPU Bus
Host Address
IMAP-CE chip
Control Units
Video Interrupt Signal Gen.
Host Interface
SDRAM Interface
Control Processor (CP)
External Memory Access Arbitor
Program(32KB) Data(2KB) Cache
Background Data Transfer Controller / Internal Memory Access Arbitor
Background Data Transfer Line Buffer (64b x 16)
Stack Memory (1 Kbyte)
Special Regs
General Registers (16 bit x 26)
ALU(32 bit) MUL(16 bit)
PE0
PE1
PE126
PE127
Memory (2 Kbyte)
Mask Reg.
Mask False Reg
General Registers (8 bit x 24)
Shift Reg.x4(8bit)
ALU1(8bit) ALU2(12 bit) MUL(8 bit)
Status/Data Bus
Instruction Bus
PE data transfer inter-connection pins
shift register (SR) inter-connection pins
CPDATA bus (for multi-chip configuration)

Number of PE	128
Frequency	100MHz
Peak Performance	51.2GOPS
Process	NEC CB-11 0.18 μ m process
Power Dissipation	2.5-4.0W @ 1.8V
Package	500 pin TBGA

Figure 1: Chip Specification and Block Diagram

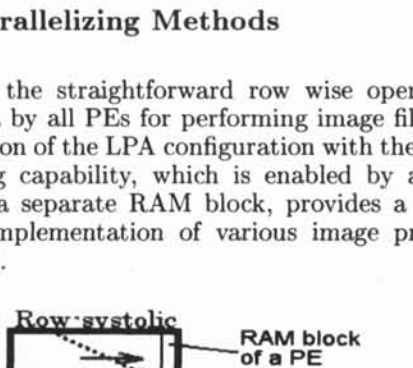
2.1 Parallel Execution by 128 VLIW PEs

128 RISC type 4-way VLIW PEs are connected in series to form a one-dimensional SIMD linear processor array (LPA), where leftmost PE is connected to the rightmost PE. Each PE is consisted of a 2K byte RAM block as local memory, 24 8-bit general purpose registers, two 1-bit mask registers for PE activity control, four 8-bit shift-registers (SRs), two 8-bit ALUs and a multiplier. Each PE executes up to 4 instructions per clock cycle to provide a peak performance of 51.2 GOPS in a single chip, or up to 819.2 GOPS in a cascade connected 16 chip configuration at 100MHz. The Control Processor (CP) within the Control Unit is a 16-bit RISC processor which provides instructions to the LPA. The CP issues up to 4 instructions per clock cycle, within which up to one instruction is decoded for the CP, and up to 4 are decoded and broadcasted to the LPA.

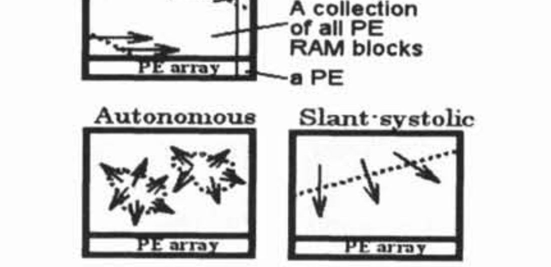
In order to support efficient operation of the PE array, an user programmable background data transfer controller (figure 1) is included for hiding data transfer overheads between collection of all PE memories (RAM blocks) and the external SDRAM. Under multi chip configuration, the PE data transfer and SR interconnection pins are connected in order to form a larger LPA in which all PEs operates in a synchronous way. The CPDATA bus is used for gathering status information of all PEs, or data broadcast from a specific PE to all other PEs.

2.2 Parallelizing Methods

Beside the straightforward row wise operation of pixel data by all PEs for performing image filters, the combination of the LPA configuration with the indexed addressing capability, which is enabled by assigning each PE a separate RAM block, provides a base for parallel implementation of various image processing tasks[6][7].



(a) Parallelizing methods based on index addressing.



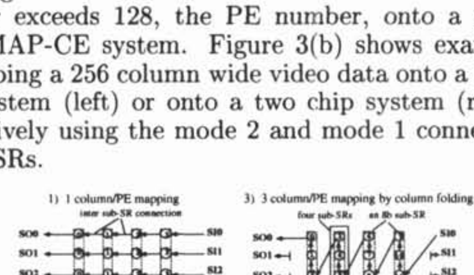
(b) An example of applying region growning on road area detection based on the autonomous method

Figure 2: Indexed Addressing based Parallel Execution

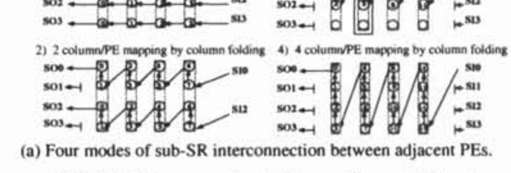
Figure 2(a) shows three indexed addressing based basic parallelizing methods: 1)row-systolic method for parallelizing collective operations such as grey level histogram calculation and area/variance measurement of each connected components. 2) slant-systolic method for parallelizing raster scan type operation such as distance transform and dither transform, and 3) autonomous method for parallelizing propagation based algorithms such as connected component labeling, region growing, and active contour based segmentations. An example of detecting road area by an autonomous method based region growing is shown in figure 2(b).

2.3 Automatic Video Data Mapping

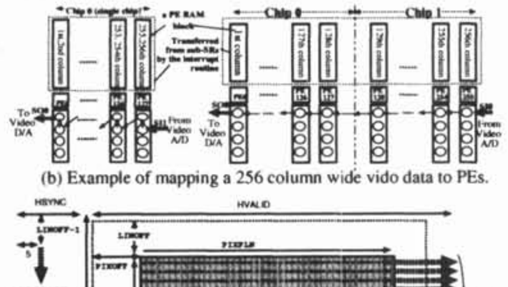
Automatic background video data mapping to each PE is achieved by using an asynchronous SR structure(Figure 3(a)(b)), An SR is further consisted of 4 sub-SRs and operate in video clock. Video data are shifted into the 4 sub-SRs in parallel with the PE execution. Whenever a video data line is completely shifted in, an interrupt signal is generated by the CP's Video Interrupt Signal Generator. The interrupt signal in turn invokes a software interrupt routine to transfer one video data line from the sub-SRs to the PE RAM block or to the external SDRAM, and in exchange store one video data line into the same sub-SRs for video data output. The inter-connection between the 4x8b sub-SRs is configurable into any of the 4 modes shown in Figure 3(a), to enable mapping of up to 4 columns of video data to each PE. This functionality enables mapping of source video data whose horizontal pixel number exceeds 128, the PE number, onto a single chip IMAP-CE system. Figure 3(b) shows examples of mapping a 256 column wide video data onto a single chip system (left) or onto a two chip system (right), respectively using the mode 2 and mode 1 connection of sub-SRs.


1) 1 column/PE mapping
inter sub-SR connection
3) 3 column/PE mapping by column folding
four sub-SRs
an 8b sub-SR
2) 2 column/PE mapping by column folding
4) 4 column/PE mapping by column folding
SO0
SO1
SO2
SO3
SI0
SI1
SI2
SI3

(a) Four modes of sub-SR interconnection between adjacent PEs.



(b) Example of mapping a 256 column wide vido data to PEs.



(c) The interrupt signal issue timing in conjunction with video signals.

Figure 3: Video data I/O functionalities

Six interrupt signals, ODDSTART to EVENEND, as shown in figure 3(c), are generated by the Video Interrupt Signal Generator (figure 1) according to the HSYNC, HVALID, and VVALID video signals that designate the valid period of respectively horizontal and vertical scan of video data. LINOFF, PIXOFF, PIXPLN, and VLINES are special registers whose value can be changed to control the timing of issuing the above six interrupt signals, and also to adjust the target video data region cutting out from the NTSC or PAL formatted video data as input video data.

3 Software Environment

Programmability is another important feature of IMAP-CE. The software environment, running on the host PC, is consisted of an optimizing VLIW compiler for 1DC (One Dimensional C)[7], a source-level debugger with graphical user interface, over a hundred of image processing library functions, and a visual programming tool calls VP1DC, for automatic communication between the C program running on the host PC and the 1DC program running on IMAP-CE.

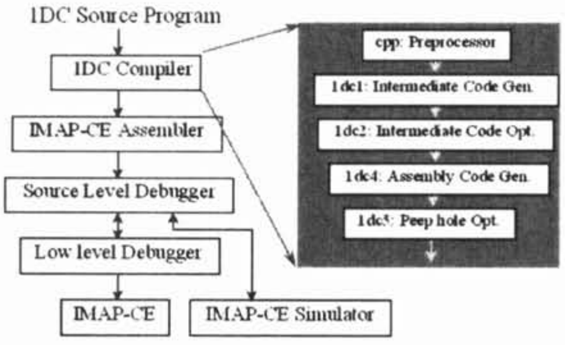


Figure 4: The 1DC Optimizing Compiler

3.1 The Extended C Language

1DC is designed as a data parallel C language, with enhancements limited only to essential necessities for the sake of clarity, and also for the support of a virtual LPA. The enhancement of 1DC from C is straightforward: (a) extended declaration of entities which associated to the PE array using the *sep* (or *separate*) keyword, (b) extended constructs for selecting active processor groups such as *mif...[melse...]*, *mwhile...*, and *mfor(...;...;...)*, and assuming E_{sep} and E_{sca} respectively as a *sep* and a *scalar* expression, (c) extended operators for manipulating data on the PE array such as "E_{sep}:[E_{sca} :]" for extracting the scalar element of E_{sep} on the E_{sca}*th* PE, ":>E_{sep}" and ":<E_{sep}" respectively for referring to the scalar element of E_{sep} located at its left and right adjacent PEs, and ":&&E_{sep}" and ":|| E_{sep}" respectively for producing a *scalar* entity whose value is the logical AND and OR of every scalar element of E_{sep}. Following is the 1DC description of a simple thresholding operation, in which the *mif...[melse...]* construct is used, and a 3×3 average filtering operation, in which the sum of each local 3×3 pixels are obtained by, combining the sum of three 1×3 pixels produced by each PE, with that produced by its two adjacent PEs using the ":>E_{sep}" and ":<E_{sep}" operators.

```
sep unsigned char src[NROW],dst[NROW];
void binarize(int thres)
{
 int   i;
 for(i=1; i<NROW-1; i++)
    mif (src[i]>thres) dst[i]=0xff; melse dst[i]=0;
}

void average_filter()
{
 sep unsigned int   acc;
 int   i;
 for(i=1; i<NROW-1; i++){
    acc = src[i-1] + src[i] + src[i+1];
    acc += (:<acc + :>acc);
    dst[i] = (acc / 9);
 }
}
```

A significant feature of the 1DC compiler for IMAP-CE is the virtualization of number of PE and video data mapping, by which modification of source codes is not required even if the number of chips or the video data mapping of the running system changes.

3.2 A Visual Programming Tool

VP1DC provides a graphical user interface which helps users to build applications containing a mixture of library function blocks written in C and 1DC. By drawing a connection between a C function block and an 1DC function block, an executable file can be generated, which automatically maps respectively the execution of C functions to the host PC processor and the execution of 1DC functions to IMAP-CE, while establishing communications between them. Figure 5 shows a canvas window snap shot of VP1DC, in which a 1DC function and a C function is connected, with additional scale and radiobutton tools for runtime adjustment of arguments and control flows, and thumbnail windows, which can be magnified for detail inspection under request, in order to observe resulting image data output of any function block.

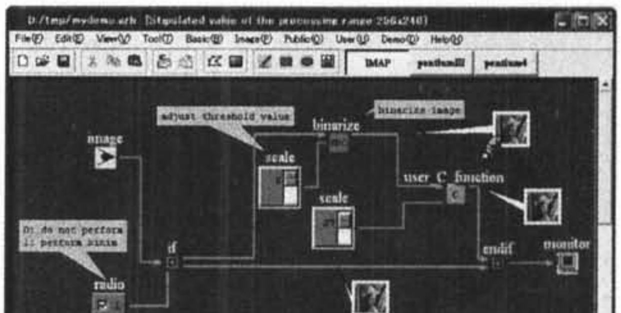


Figure 5: Visual Programming Tool

4 Performance Estimation

Figure 6 compared the estimated processing time for some image filter functions using 250x240 gray scale images, between 1) running 1DC codes on a single chip IMAP-CE, 2) running hand-written assembly functions using MMX/SSE instructions, and 3) running conventional C codes on a desktop PC.

Figure 6 shows that 1DC codes running on a single chip 100MHz IMAP-CE are about, 3 to 30 times

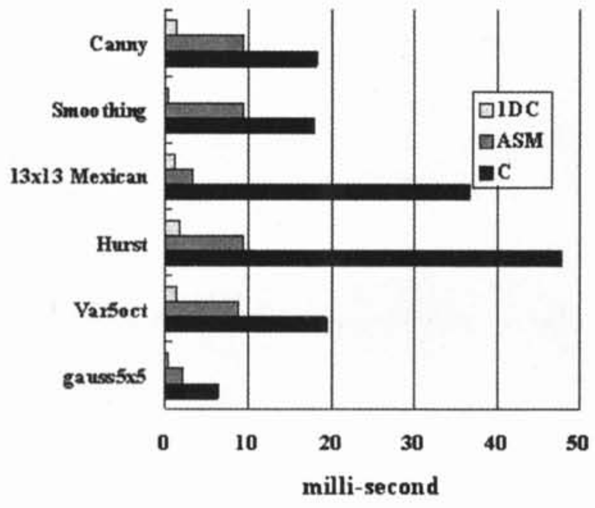


Figure 6: Performance evaluation using Image filters

faster than corresponding hand-written assembly functions, and 10 to 50 times faster than corresponding C functions, running on an 2.8GHz desktop PC.

Figure 7 shows a sample result of a lane-mark and vehicle detection/tracking function developed in 1DC for ICC applications[8][9]. In order to improve the performance under reduced visibility conditions, various image processing techniques are required, thus consumes additional computation power. The processing time of IMAP-CE compared with that of IMAP-VISION[5] is also shown in figure 7. IMAP-VISION is a full size PCI card, a predecessor system operating at 40MHz, and consisting of 256 PEs by an eight chip configuration using IMAP-V, a predecessor chip of IMAP-CE. Figure 7 shows that a single chip IMAP-CE, which is about three times faster than an IMAP-VISION system, can achieve a robust lane-mark and vehicle detection/tracking function in video rate. The result demonstrates its high potential as a compact engine for ICC applications that require crucial real-time responses.

5 Conclusion

Hardware features of IMAP-CE are its concurrent video data I/O capability based on the shift-registers, high memory-processor bandwidth and performance produced by a 128 4-way VLIW processing element and memory integration, and the scalability and programmability originated from a SIMD linear array PE configuration. Performance of codes written in 1DC, an extended parallel C language, and runs on a 100MHz single chip IMAP-CE are estimated to be about 3 to 50 times better than corresponding assembly codes or C codes running on an 2.8GHz general purpose processor. A robust lane-mark and preceding vehicle detection/tracking function for ICC applications are estimated to achieve a real-time performance using a single chip IMAP-CE system. We believe that, IMAP-CE can fulfill the size, power and performance requirements, for embedded systems that require high performance to fulfill real-time restrictions.

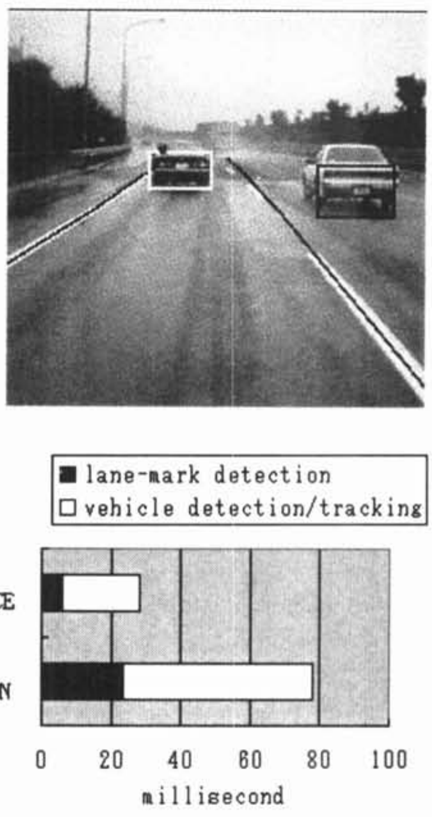


Figure 7: Performance evaluation of ICC applications

References

[1] T.J.Fountain, "The CLIP7A Image Processor," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, Vol.10, No.3, pp.310–319 (1988).

[2] L.A. Schmitt et al.,"The AIS-5000 Parallel Processor," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, Vol.10,No.3,pp.320–330 (1988).

[3] Y.Fujita et al.,"IMAP: Integrated Memory Array Processor",*Journal of Circuits, Systems and Computers*, Vol.2, No.3, pp.227–245 (1992).

[4] N.Yamashita et al.,"A 3.84GIPS Integrated Memory Array Processor LSI with 64 Processing Elements and 2Mb SRAM", *ISSCC94*, FA15.2, pp.260–261 (1994).

[5] Y. Fujita et al.,"A 10 GIPS SIMD Processor for PC-based Real-Time Vision Applications", Proc. of IEEE Workshop on Computer Architecture for Machine Perception (CAMP'97), pp.22–26, (1997).

[6] S.Kyo et al., "A Parallelizing Method for Implementing Image Processing Tasks on SIMD Linear Processor Arrays", *Proc. of IEEE Workshop on Computer Architecture for Machine Perception (CAMP'97)*, pp.180–184, (1997).

[7] S. Kyo et al., "Efficient Implementation of Image Processing Algorithms on Linear Processor Arrays using the Data Parallel Language 1DC", Proc. of IAPR Workshop on Machine Vision Applications (MVA'96), pp.160–165 (1996).

[8] S.Sakurai et al., "A Lane Recognition Algorithm Based on White Line Detection and Road Area Detection," Proceedings of the 1998 Intelligent Vehicles Conference, pp.58–62 (1998).

[9] S. Kyo et al.,"A Robust Vehicle Detecting and Tracking System for Wet Weather Conditions using the IMAP-VISION Image Processing Board", IEEE/IEEJ/JSAI Proc. of International Conference on Intelligent Transportation Systems (ITSC'99), pp.423–428, (1999).