THE VERSATILE IMAGE PROCESSOR V.I.P. (HARDWARE DESIGN)

G. Gugliotta, A. Machi'
I.F.C.A.I. Italian National Research Council
Via M. Stabile, 172 90139 Palermo, ITALY

ABSTRACT

This paper presents the architecture of a medium-grain parallel processor well suited for image analysis. The processor, named V.I.P. is composed by clusters of 4 Intel i860 RISC processors connected among themselves and to I/O units through a parallel bus in the industrial standard VME, a parallel custom Video Bus and a serial network. The processors operate concurrently on cluster and system shared memories through the parallel busses and exchange messages among crates and with a host system through the serial network. Modularity and alternative path availability allow to tailor the system to the problem; video bus high bandwidth and computational modules processing power allow very low processing time; usage of general purpose processors and of simple contention control routines make easy programming the machine. Details of the architecture and of the prototype performance are given.

INTRODUCTION

Real time Image analysis in machine vision often require very high computational power and different data manipulation strategies are applied in the various steps of the analysis process. At low level, homogeneous and input-output intensive algorithms process pixel neighbourhood, while, at high level data dependent algorithms randomly access complex data structures representing object features; a mixed usage of both kind of algorithms is done while extracting features from iconic data .

Parallelism is supposed to help in speeding-up computation towards real-time performance, but, up to now, no limited cost parallel machine has demonstrated to be sufficiently flexible and efficient to cope with the different requirements of both low and high-level image processing. For instance, bit-serial parallel architectures appear adequate for just low-level local operations, while VLSI processors execute in real-time pre-defined algorithms, limited numbers of tight-connected shared-memory processors well behave in accessing sparse data structures and distributed memory processors, instead, perform better on well-partitionable low-communication tasks.

In practical applications, an ultimate limit to speed-up is given by the time required to scatter the image to processor array and to gather back the analysed one in a global store (shared memory or control processor) before the next processing step. Moving one quarter Mby TV image back and forth takes from half to several frame time intervals using DMA techniques (ex. 25 ms on a 20 Mby/s VME bus, or 125 ms exploiting the full 4 Mby/s pass-through bandwidth of a Transputer serial network) [1].

The integration of various flavours of parallelism into a single powerful and flexible architecture, able to support optimization techniques in the various steps of the analysis is the main goal of the Versatile ImageProcessor (V.I.P.) project, developed in Italy in a cooperation between IFCAI of Italian C.N.R. and TECNINT ltd.

Project sponsored by Progetto Finalizzato Sistemi Informatici e Calcolo Parallelo of Italian National Research Council under Grant N° 212363/69/9107184

THE V.I.P. ARCHITECTURE

In the machine architecture, multiple clusters of processors (referred in the following as Computational Units or CU) are connected together and to Input Output Units (IOU) by means of several parallel and serial busses (fig. 1).

Processors are organised into three hierarchical levels: the cluster, the crate and the system level .

At the cluster level, 4 i860 super computing microprocessors share on a single board a double bank of Video RAM. The processors access in concurrence the memory through a 64 bit wide local bus, using the RAM parallel port.

At the crate level, up to 16 CUs and IOUs communicate each other through tight inter-crate connections. Two parallel busses are available for that: a 32 bit wide bus using the VME standard and a 64 bit wide custom Video bus (TCViBus). The first will allow to integrate into the system third party units as an host platform or process controllers, while the second is devoted to very high speed data block transfers. The VME bus connects memories on CUs and IOUs from the parallel side only, while the TCViBus connects them from both the parallel and serial ones.[2]

Finally, at the system level, a loose connection is available for mid-distance communications. Serial links are available on one IOU to implement a point-to-point network using INMOS technology and protocols. The network allows control of multi-crate configurations in a distributed environment. An INMOS T800 transputer on each crate manages message dispatch on the serial links and may operate as a crate controller distributing tasks to the processors into the PUs. If required, it may communicate with an external host for debugging or file service purposes.

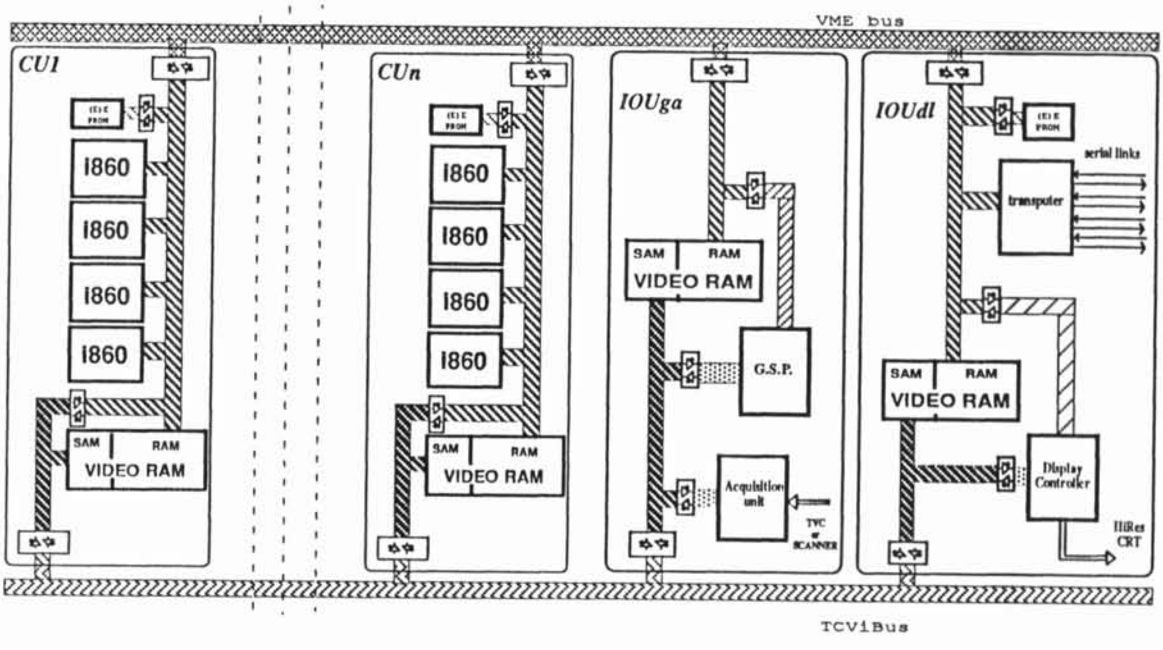

Fig.1:The Versatile Image Processor (VIP) architecture

THE V.I.P. PROTOTYPE

A system prototype, composed by two clusters is presently under test at IFCAI.

Each CU (fig. 2) consists in a single board equipped with 4 Intel i860, 4 Mby of dynamic VRAM, VME and TCViBus interface logic and a local bus arbiter.

The processors operate at 32 MHz clock cycle and hold both an instruction and a data cache on chip.

Memory is arranged in two 64 bit banks interleaved to allow to use cheap 100 ns VRAMs; during processor cache fill and write back cycles a sustained throughput of 80 Mby/s is achieved through the memory parallel port. The full implementation of the two stage pipelined access mode for the local bus will increase the bus throughput up to 130 Mby/s.

Hardware mailboxes are mapped into CU memory and allow to independently interrupt each processor on the cluster from local, VME and TCViBus.

A local arbiter manages concurrence among the processors, the refresh circuitry and the VME and TCViBus interface; it uses an algorithm which assigns higher priority to TCViBus requests and serves the others requesters on a round-robin base.

The VME interface implements a pure slave protocol and allows a moderate sustained throughput of 15 Mby/s managed from external bus masters.

The TCViBus offers three access modalities: single, pipelined and fixed-size block transfer. The single and pipelined access modes allow to randomly access memory in CUs and IOUs; during data transfer both local busses in the communicating units are busy, as well as the TCViBus. A throughput of 32 Mby/s is achievable in the prototype during concurrent random access, and a 80 Mby/s peak throughput is expected during exclusive privileged transfers of aligned data in the pipeline mode (not yet implemented). In the third mode, block transfer, 8Kby data blocks may be moved at 260 Mby/s. between cluster video memories in the same crate accessing VRAMs from the serial ports. TCViBus is an active bus and its circuitry is responsible for managing the transfer; the local busses of the two Units involved in the transfer are taken busy for just a small fraction of the transfer time, just a few clock cycles while data are moved from (to) the memory arrays to (from) their serial ports. Just 0.8 msec. are required to move a standard 512x512x8 bit CCD frame through the TCViBus. The maximum expected throughput is 320 Mby/s obtainable driving the transfer with a 40 MHz bus clock.

At power on, a start-up code allows each processor to self assign a different identity code and to implement a separate stack. The contents of volatile variables in the cache, and in the stack, distinguish processor operations, the remaining of memory (code and other data) being shared.

A hardware mechanism based on explicit addressing preserves part of the memory from being cached so that data integrity is maintained during concurrent accesses. Exclusion is implemented through the i860 LOCK instruction and hardware signal. After having asserted the LOCK signal, for a few tens of instructions the processor is granted to exclusively access the local bus to perform read-modify-write cycles .

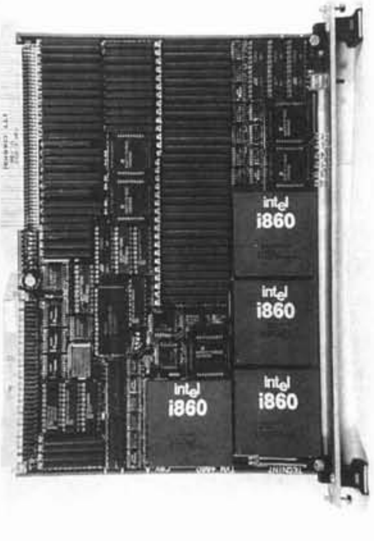

Fig. 2: One V.I.P Computational Unit.

IMAGE PROCESSING PERFORMANCE EVALUATION

Parallel implementation and optimization of image processing algorithms on the prototype are presently being performed on the machine prototype and detailed data are not yet available for communication but few preliminary results allow to estimate with sufficient confidence the performance figures for low level processing algorithms.

Let's then follow the processing of a 512 by 512 by 8 bit TV image using a homogeneous algorithm of low-level.

The image is scattered into CU local memories and each processor analyses a different block. During the scan loop it accesses each pixel and its neighbourhood and stores the result locally Because the data cache can contain more than one ten of rows, each pixel is read just once in cache and then accessed there. Output data are written into local memory by cache write back mechanism one cache row (32 bytes) a time; they are then copied back to a common storage using block transfer on the TCViBus. In the system prototype, for each byte pixel analysed, the overhead for block transfers amounts to 2/8 of a clock cycle, and the overhead for the reading in cache amounts to 13/32 of a clock cycle. Write cycles to local memory are overlayed to processing, provided that the local bus is not saturated. This last condition is not granted when the processing time is less than the effective write cycle length times the number of processors (13/32*4 cycles due to cache row update).

Histogramming and thresholding require about 10 cycles per pixel, sobel filtering about 50, and a standard 3x3 convolution one hundred. Image processing time on a single processor then ranges from tens to hundred of milliseconds and decreases linearly with the number of processors because the local bus saturation condition is not met and the TCViBus occupancy ratio is limited to a few percent of the time required for the execution of the algorithm by just one processor.

CONCLUSIONS AND FUTURE WORK

The prototype of a multi-cluster shared memory machine has been built. Preliminary results obtained verifying Read/Write time cycles on the various busses and processor code execution are encouraging. In the execution of low level image processing algorithms the block transfer mechanism on the TCViBus greatly reduces the overhead due to image movements; on chip cache greatly reduces bus load on the cluster bus. Absolute execution times very near to real time are expected after system and code optimization.

More work has to be done to evaluate, in the execution of mid-level algorithms, the overhead due to the LOCK mechanism and to the combined usage of block and random single accesses through the TCViBus.

ACKNOWLEDGEMENTS

The AA. are gratefully indebted to Eng. F.Piccirelli and to Mr C. Granuzzo, Project Managers of TECNINT for the continuous and fruitful support of ideas and technical solutions to the development of VIP prototype.

REFERENCES

[1] Anzalone A. , Gerardi G., Lenzitti B., Machì A.: "Parallel implementation of low level image processing algorithms on a linear array of transputers and on a shared memory based multiprocessor" in: Supercomputing Tools for Science and Engineering ; F. Angeli Milano 1989.

[2] Nicoud J.D.:"Video RAMs Structure and Applications" IEEE Micro Feb88 pp. 8-27.