

# A PIPELINE MULTIPROCESSOR ARCHITECTURE FOR HIGH SPEED CELL IMAGE ANALYSIS

**Kenneth R. Castleman  
Kenneth H. Price  
Raymond Eskenazi  
Musa M. Ovadya  
Mois A. Navon**

**IEEE COMPUTER  
SOCIETY REPRINT**

Reprinted from 1983 IEEE COMPUTER SOCIETY WORKSHOP ON  
COMPUTER ARCHITECTURE FOR PATTERN ANALYSIS AND  
IMAGE DATABASE MANAGEMENT — CAPAIDM



**IEEE COMPUTER SOCIETY**  
1109 Spring Street, Suite 300  
Silver Spring, MD 20910

# A Pipeline Multiprocessor Architecture for High Speed Cell Image Analysis

Kenneth R. Castleman, Kenneth H. Price, Raymond Eskenazi  
Musa M. Ovadya and Mois A. Navon

Jet Propulsion Laboratory  
Pasadena, California 91109 USA

## ABSTRACT

A pipeline multiple microprocessor architecture for high speed digital image processing is being developed at JPL. The application is a compact, fast and low cost Pap smear analyzer for cervical cancer detection. Each processor communicates with one or two upstream processors and from one to thirteen downstream processors via shared memory. Each of the identical pipeline modules (PC boards) has a Motorola MC6809 microprocessor with a 2 Megabyte memory management unit, two 64K byte dual-port image memories (shared with upstream processors) and one 64K byte two-port program memory (shared with a host computer). Intermodule communication is achieved by ribbon cables connected to connectors at the top of the boards. This allows considerable flexibility in configuring the system. This architecture should facilitate efficient (fast, low-cost) implementations of complex single purpose image processing systems.

## INTRODUCTION

Cervical cell analysis is typical of a class of computationally intensive image processing applications that do not lend themselves well to either conventional, single CPU architecture or to real time raster scan image processing hardware. Array processors are likewise inefficient means for implementation.

The processing steps of image segmentation, cell measurement and cell classification require random access to the image and are considerably data dependent. Thus, the single CPU architecture is useful, but frequently too slow to meet performance requirements. Real time raster scan implementations perform some of the steps well, but are inappropriate for others. Since most of the computational load is 8-bit arithmetic, the typical floating point array processor is inefficient. The overhead of data transfer to and from the array processor is frequently as time consuming as simply performing the operations in the host.

One solution to this problem is to use a number of inexpensive but powerful microprocessors all executing simultaneously. The two major decisions that remain are (1) how to divide the images

and algorithms among the CPU's and (2) how to implement interprocessor communication.

## THE PIPE

For a single purpose system, such as a Pap smear analyzer, one can design hardware and software together to arrive at an optimal solution. We have chosen a pipeline multiprocessor architecture with data flow via shared dual-port memory. In this arrangement the processing algorithm is partitioned into sequential steps, and each step is implemented on a separate CPU. The images flow from one CPU to the next, assembly line fashion, and emerge from the last one fully processed. Since each CPU shares image memory with its upstream and downstream neighbors, there is no I/O per se. When one CPU writes its output into shared memory it automatically appears for use by the downstream processor.

Figure 1 shows the architecture of the Pap smear analysis system. The host (master) computer handles operator communication and keeps track of the number of normal and abnormal cells found. Each of the identical modules implements one step in the analysis of a cell.

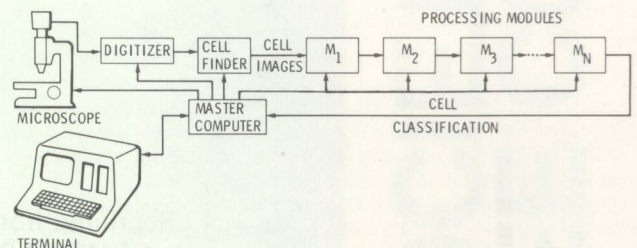


Figure 1

The Pap smear analysis system consists of a microscope with image digitizer, a cell finder circuit to detect the presence of cells in the images, a master or host computer with operator's console, and a series of identical processing modules. Each module performs one step in the analysis and passes the cell image on to the next module. The last module identifies the cell as normal or abnormal. The host tabulates the number of each cell type and reports a diagnosis. It also controls the microscope, digitizer and cell finder and downloads programs into the modules.

Each module operates independently and autonomously. As long as there is an input image from the upstream module available for processing and output memory space available in the downstream module, the program executes. Otherwise it waits for available data or memory. The algorithm partitioning must be done in such a way as to minimize "choking" (no available output memory) and "starving" (no input data) of the pipeline. Since execution times are image dependent, this must be done on a statistical basis. In the development stage, each module keeps track of its choking and starving time so that repartitioning of the algorithm can keep these to a minimum.

### THE MODULES

Figure 2 shows a block diagram of the pipeline processing module. Each board contains the processor, two input memories and a program memory that can be downloaded from the host. In the most recent design all three RAMs are 64K bytes. An MC6829 memory management chip handles the mapping of the 64K byte address space of the MC6809 into the 2M byte physical address space of the system.

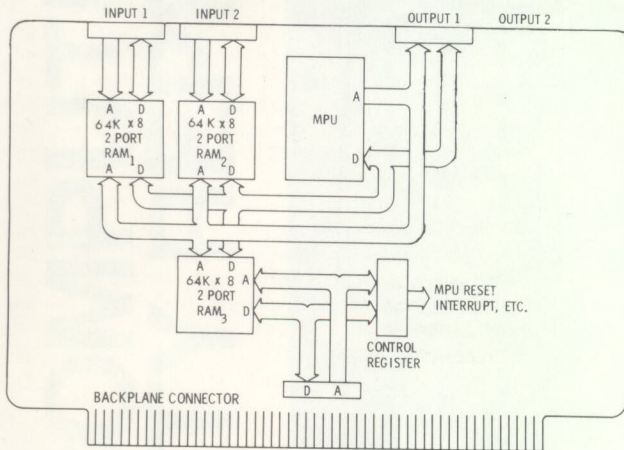


Figure 2

Each processing module is implemented on a single printed circuit board. Each contains the MC6809 microprocessor, and three two-port RAMs, two shared with upstream processors via input connectors, and one shared with the host via the backplane. A control register permits the host to control the MPU. The address bus is expanded to 21 bits by an MC6829 memory management chip. The address and data busses are brought out through ribbon cable connectors so that output data can be stored in downstream modules. For the Pap smear application, image size is fixed at 128 by 128 bytes.

Figure 3 shows how the modules are connected together. The ribbon cable technique allows considerable flexibility in configuring the pipeline, which can be reconfigured simply by changing the cables. The assignment of particular software processes to individual modules must, of course,

match the cabling configuration.

Figure 4 shows how the various memories can appear in MC6809 address space. The local program can assume input and output arrays are always at the same locations. Furthermore I/O operations as such are never required since the shared memory concept makes one module's output array coincide with another module's input array.

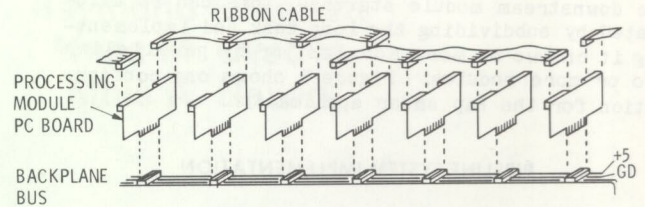


Figure 3

The processing modules communicate via ribbon cables connected to the top of the cards. Each module can receive input data from up to two modules and deliver data to as many as thirteen. The host can download program memory and control each module via the backplane bus.

### MEMORY MAP

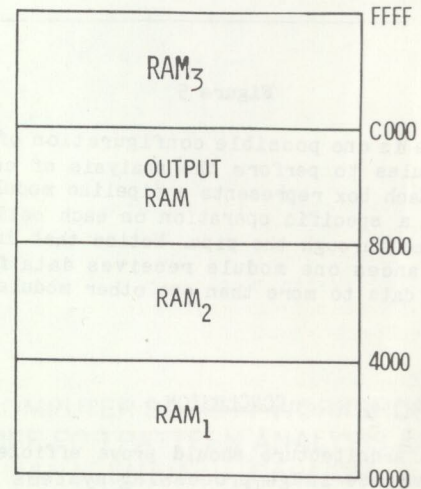


Figure 4

The memory map of the MPU contains the program memory (RAM<sub>3</sub>) the two input memories (RAM<sub>1</sub> and RAM<sub>2</sub>) and the output RAM which resides in a downstream module. Since input and output image are always in addressable memory, no I/O operations are required. The memory management chip takes care of mapping the 64K byte MPU address space into the 2M byte physical address space of the ribbon cable busses.

## CONFIGURATION

Each system design begins with the algorithm flow chart. The overall process is partitioned into tasks assigned one to a module, and the modules are connected by ribbon cables. Each task program is developed in the host (a Motorola Exorcisor development system) and downloaded to the appropriate module. During testing, records are kept of module choking and starving times.

If one module is overworked compared to its neighbors, the upstream module will choke while the downstream module starves. This can be alleviated by subdividing the long task and implementing it on two or more modules, or by paralleling two or more modules. Figure 5 shows one configuration for the Pap smear application.

### PIPELINE SYSTEM IMPLEMENTATION

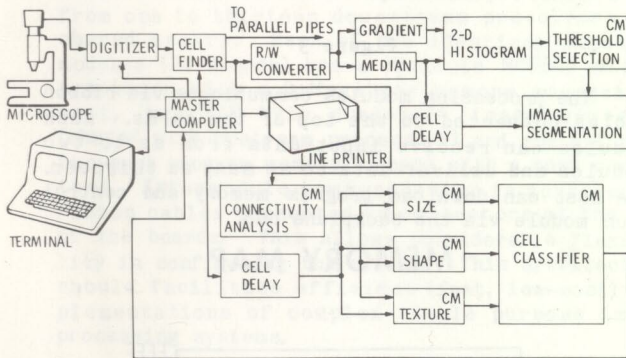


Figure 5

This is one possible configuration of pipeline modules to perform the analysis of cervical cells. Each box represents a pipeline module that performs a specific operation on each cell image that comes through the pipe. Notice that in several instances one module receives data from or delivers data to more than one other module.

## CONCLUSION

This architecture should prove efficient for single purpose image processing systems having performance requirements of high volume, high speed, low cost, low power, small size and light weight. It allows simultaneous optimization of hardware configuration and software. The basic processing module and the architecture are flexible enough for a wide variety of applications.