Real–Time Pattern Recognition by Massively Parallel Systolic Processors*

Reinhard Männer
Lehrstuhl für Informatik V, Universität Mannheim, Mannheim, Germany, and
Interdisziplinäres Zentrum für Wissenschaftliches Rechnen, Univ. Heidelberg, Heidelberg, G
maenner@mp-sun1.informatik.uni-mannheim.de

Joachim Gläß, Frank Klefenz, Andreas Kugel, Klaus–Henning Noffz, Ralf Zoz
Lehrstuhl für Informatik V, Universität Mannheim, Mannheim, Germany

Roland Baur
Physikalisches Institut, Universität Heidelberg, Heidelberg, Germany

November 23, 1995

Abstract

Three VMEbus controlled systolic array processors for high–speed pattern recognition are described. Within 3 μs to 20 μs ill–defined straight and curved lines, and circles are identified. The first processor uses direct matching of image pixels with a set of programmable masks. It is fully pipelined and scalable in the image size and the number of masks. Testing of a prototype begun in autumn 1993. The second processor is based on a parallel Hough transform that histograms all possible coordinate pairs (radius of curvature, starting angle). The number of well defined patterns is counted using the Euler relation. A prototype of the processor with 1,120 processing elements (PEs) implemented by programmable gate arrays is under test. The full scale system will consist of 26,880 PEs. The third processor consists of 28,160 PEs assembled on 1 m^2 PC boards filled with 2 μ CMOS custom VLSI chips containing 8×8 PEs each. The fully programmable processor is in operation since spring 1991.

INTRODUCTION

*Work partly supported by the German Bundesministerium für Forschung und Technologie (BMFT) under grant PH/RS6-93/09; the Gesellschaft für Schwerionenforschung (GSI) Darmstadt, Germany, under grant HD Spe C; and the RD–11 and RD–6 collaborations at CERN, Geneva, Switzerland.

15 years ago many experiments in nuclear and high energy physics used the old CAMAC standard for front–end data processing. However this bus system was too slow and too inflexible for new experiments, and due to inherent limitations not suited for larger set–ups. One alternative system, FASTBUS, had been designed mostly within the high energy physics community. However it aimed at solving all problems encountered with CAMAC at once, and was too expensive for most applications. Moreover industry did not accept FASTBUS, particularly since it was not well adapted to the 16– and 32–bit microprocessors that appeared on the market. Thus the VMEbus was highly welcome and it was applied to front–end data processing short time after its specification.

Today it is the most widely used bus system for such applications. Clearly the requirements for data processing in high energy physics have raised considerably in the last ten years. Existing accelerators like the Large Electron Positron Collider (LEP) [1] at CERN, Geneva, Switzerland, produce interactions between bunches of particles at a rate of, e.g., $5 \cdot 10^4$/s. The next generation of accelerators like the Large Hadron Collider (LHC) [2], planned also for CERN, will have bunch crossing rates of about $5 \cdot 10^7$/s. At comparable speed *events* are produced, data sets that describe reactions of particles with each other. They consist of a very large number of coordinates of reaction products (other particles), their energies, energy losses, etc. Large experiments have $10^5 - 10^6$ data channels that may be activated in one event. Thus this enormous data stream has to be reduced in real–time before it can be stored on tape, i.e., interesting events have to filtered out.

This is done in several steps. Usually a *first level trigger* is used that works on the same time scale as the bunch crossing rate of the accelerator. It identifies uninteresting events using very simple combinations of parameters. This allows to reduce the data stream by one or two orders of magnitude. A *second level trigger*, which has therefore accordingly more time for processing, usually applies pattern recognition techniques to reduce the remaining data stream also by such a factor. Events that also pass the second filter step are usually forwarded to a *third level trigger* that consists of a "farm" of microprocessors. This is a multiprocessor system in which every processor does a rather deep analysis of a complete event, and the events are distributed over the processors as they become available. Here often VMEbus based commercial microprocessors are applied. However the two earlier data processing stages also use often VMEbus, at least to control special purpose processors for first and second level triggering.

THE PATTERN RECOGNITION PROBLEMS

At the second trigger level simple patterns have to be recognized within microseconds. The patterns are recorded by detectors in response to particles crossing them. Depending on the detector type the patterns may be straight lines of different slope, lines of different curvature and different starting point and angle, circles of different radius, etc. In the following, three examples are given.

The first one is the simplest in processing complexity. The task is the identification of "electron compatible" tracks in the TRD detector [3] of the LHC collider within 10 μs. This problem is solved by a pattern recognition processor, the Enable Machine [4], as described below. This system has been designed within the EAST/RD–11 collaboration at CERN and conforms exactly to the EAST benchmark specifications [5]. It has been implemented in programmable gate arrays. Prototype testing started in autumn 1993.

The Enable Machine is designed to find tracks in the outer parts of the TRD detector. As a second level trigger system it receives information about regions of interest from a first level trigger, in this case calorimeter modules positioned around the TRD surface. In the considered region the TRD is assembled of equidistant slices along the beam axis z. Each one consists of radially oriented "straws", little drift chambers of 4 mm diameter. If hit by a particle a straw marks a pixel in the (ϕ, z) plane where ϕ is given by the straw number and z by the slice number. Since the detector operates in a homogeneous magnetic field particles have trajectories of nearly constant radius of curvature, i.e., constant $d\phi/dz$. In the (ϕ, z) plane such tracks correspond to straight lines of different slope. To identify electron compatible tracks, one has to take into account the amplitude of the signal generated in a straw. For a rough discrimination two thresholds are applied to these signals, generating a "total" data set containing all signals above a low threshold, and a "high" data set containing only signals above a high threshold (Fig. 1). To solve the pattern recognition task, the best defined track in the "total" image has to be identified, and the ratio of the number of hits along this track in both images has to be computed.

The second example relates to a situation in which the pattern recognition task is more complicated. It is solved by mapping the input image onto a different space in which the essential features can more easily be extracted. The coordinate transform is executed by a parallel Hough transform processor. A prototype of such a processor has been developed and is under test since autumn 1992.

The OPAL jet chamber [6] records particle tracks in three dimensions. As with the TRD, tracks have essentially a constant curvature. Interesting events contain a certain number of tracks of a maximal curvature. Moreover most interesting tracks come from the interaction zone in the center of the beam pipe. Unfortunately a complete 3D track recognition is computationally too demanding for a trigger system. Yet 2D track recognition in the (r, ϕ) plane is sufficient for triggering. In this plane the interesting tracks are given as circle segments of an arbitrary starting angle and a minimal radius of curvature. Fig. 2 shows two sectors with a simulated event. The trigger task is to find and to count in the (r, ϕ) projection of the detector image all circle segments with a common vertex point at the origin. A new event is produced every 22 μs and the complete operation has to be finished within this time.

In the third example an even more complicated task has to be solved. Here the images to be processed are read out from a ring image Cherenkov counter

Figure 1: Typical images in which straight lines have to be identified (top: "total" data, bottom: "high" data).

[7]. It records individual photons in a cone around the direction of electrons. The position of the circle's center is determined by the particle direction. The radius of the circle depends only on the particle type, i.e., the electrons searched for create circles of a fixed diameter. Unfortunately, all circles are determined only by a statistically fluctuating number of photons corresponding to 10 to 20 individual pixels in the image plane. Moreover, these pixels are embedded in noise (Fig. 3). Even worse, there are different physical processes that create electron pairs. Mostly uninteresting "close" electron pairs are detected that can be identified due to their short distance in the image plane. Interesting events contain at least one "open" electron pair, i.e., two circles of a minimum distance. If such an event is detected, a trigger signal is required to start read–out of the detector. Since events are recorded at a rate of 50 kHz, the pattern recognition process has to be finished within 20 μs.

STRAIGHTFORWARD SOLUTION OF THE PATTERN RECOGNITION PROBLEMS

The three pattern recognition problems described can be solved by histogramming. Using this method it is simply counted how many pixels of the input image are compatible with all possible patterns searched for.

Figure 2: Typical track distribution in two OPAL sectors.

To find the best electron compatible track in the TRD detector, two histograms have to be generated, one for the total data set and one for the high data set. In both histograms one counter is assigned to every possible track. The straight lines searched for are given by a combination of a certain slope and offset. The counter contents then represents the probability with which the respective track is present in the input image.

To identify circular tracks by applying a Hough transform—as appropriate for the second example problem—requires to use a suitable coordinate mapping. The OPAL detector consists of 24 sectors which operate independently. Each track i is defined by a set of track points $(r, \phi)_i$. Because interesting tracks come from the origin, each one is determined by its starting angle ϕ_s and its radius of curvature r_c. To find well defined tracks a Hough transform is used. Such a transform maps the detector coordinates in the (r, ϕ) plane onto new coordinates in a $(1/r_c, \phi_s)$ plane where r_c is the radius of curvature and ϕ_s is the starting angle of a track. Of course a single pixel in the (r, ϕ) may belong to many tracks, already if only tracks are considered that come from the origin. Only a set of tracks is fixed whose $(1/r_c, \phi_s)$ coordinates have a simple geometric relation to the coordinates of that pixel in the detector image. This relation can be read from Fig. 4 as $\frac{r}{2}/r_c = \sin(\phi - \phi_s)$. Given the (r, ϕ) coordinates of a pixel, the Hough transform maps it onto a sine curve of amplitude $2/r$ and a phase shift of $\phi - \phi_s$ in the Hough space. If a second input pixel is transformed the corresponding sine curves intersect in the Hough plane. The intersection point marks the $(1/r_c, \phi_s)$ coordinates which are compatible with a track that comes from the origin and goes through both pixels.

Accordingly the Hough plane can be realized as an array of counters that all are initialized to zero. If a pixel is transformed the corresponding sine curve is generated and all counters below it are incremented. The transform result for a

Figure 3: Typical image generated by a RICH detector. The circles searched for in the first processing step are indicated.

complete image then represents directly the probability for the presence of tracks of parameters $(1/r_c, \phi_s)$. Clearly it is not possible to handle all input pixels concurrently. This would require to handle sine curves of all possible amplitudes and all possible phase shifts at the same time. It is required to do something sequentially in order to limit the implementation effort. The corresponding parameter is suggested by the structure of the detector. The OPAL jet chamber is a gas detector. Particles passing through it set free electrons along their trajectories. These electrons drift to detecting wires and are recorded as soon as they arrive there. Every detector sector has 160 wires that are read out in parallel. It is therefore natural to handle the ϕ coordinate sequentially and the r coordinate in parallel, i.e., all pixels of a certain ϕ but all possible r at the same time t. This simplifies the implementation of the Hough transform considerably. Now every time step sine curves of all possible amplitudes but of a single phase shift have to be handled. In addition the sine curves can be approximated by straight lines since one half–sector of the detector covers only an angle of 7.5 degree.

The Hough transform result consists of a 2D histogram which contains high peaks at those positions in the $(1/r_c, \phi_s)$ plane that correspond to well defined tracks. These peaks have to be identified and counted for the trigger decision. To do so a global threshold is set first in the 2D histogram. Its value corresponds to the minimum number of hits that are required for a well defined track. This operation creates a binary image that contains clusters of adjacent pixels at these positions. These clusters are counted by applying the Euler relation. This operation is performed by scanning the binary image with two 2×2 masks and counting the frequency H of their occurrences. The difference directly gives the

Figure 4: Geometric relation between the coordinates of a pixel in the (r, ϕ) detector plane and the $(1/r_c, \phi_s)$ Hough plane.

number of objects in the image minus the number of included holes. Here the pixel clusters are convex. Thus the relation holds

$$\text{Number of well defined tracks} = H\left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}\right) - H\left(\begin{bmatrix} \mathrm{x} & 1 \\ 1 & 0 \end{bmatrix}\right).$$

In the most complicated third example, the pattern recognition process—to find the positions of all relatively well defined circles of a fixed diameter and a minimum distance in the input picture—is executed in several steps. The first one is to determine the probability to find the center of a circle at all possible positions (x, y) in the input image. This information is obtained by a two–dimensional correlation of the input picture with a mask picture which consists of a single circle of the required diameter. The result of this correlation is a two–dimensional picture whose analog values $f(x, y)$ represent the probability distribution required. In this distribution, each peak corresponds to one circle as long as circles do not overlap considerably.

The second step consists in finding the position of peaks having sufficient height, i.e., the position of sufficiently well defined circles. This is done by proper thresholding followed by a thinning operation that yields a single pixel per peak. The new intermediate picture generated in this way consists of individual pixels representing the remaining centers of well defined circles.

In a third step all circle centers of a distance smaller than the required minimum have to be deleted. To check all mutual distances in parallel, this intermediate picture is correlated with a circular disk of a radius equal to the minimum distance. In the two–dimensional correlation result, all pixels (x, y) with a function value $f(x, y) = 1$ correspond to "isolated" circle centers which do not have neighbor pixels within the minimum distance. By computing the logical AND of all pixels with an associated function value of 1 and the intermediate

picture (= the centers of all well defined circles), all circles are deleted which are too close together.

IMPLEMENTATION BY SYSTOLIC ARRAYS

The operations described above have to be applied to all pixels of the input images. Thus massive parallelism should be exploitable to reach the required speed. This is most easily achieved if the pattern recognition processors are implemented as systolic arrays. In such an architecture a large number of very simple identical processing elements (PEs) operate in lock–step on spatially distributed data. In order to avoid data routing problems the images to be processed are pipelined through the processing array so that only nearest–neighbor interconnects are required. Both features of a PE, simple structure and local interconnects, ease greatly an VLSI implementation of systolic processors. The three pattern recognition problems described above can easily be implemented by such processors as shown now.

The Enable Machine for the recognition of electron–compatible tracks is composed of three building blocks, one histogram generation unit for "total" and "high" data each, and a trigger decision unit (Fig. 5). The histogram generation units are used to count the number of pixels that may belong to all physically reasonable particle tracks. The trigger decision unit analyzes the contents of the histogram channels and does the electron/pion classification in programmable hardware. The Enable Machine has a pipelined architecture. One column of the two input images each are entered every step and are forwarded through a number of pipeline stages. Only a small number ($\leq$10) of different slopes are of interest. In addition about 20 ϕ values must be handled. Thus every column of the Enable Machine handles all tracks of a single slope, i.e., parallel tracks of all possible offsets. Only one pattern has to be stored in a local memory and can therefore be chosen arbitrarily. The input image is compared with this track pattern after the corresponding offset has been added. Where matches are found counters are incremented, one for every offset. After the last column of both images have passed the first column of the two histogramming units, the first two histograms are ready.

The histograms are read out to the trigger decision unit that first has to find the maximum of the weighted sum of the two corresponding histogram channels within this column. This is done by shifting the counter contents sequentially into a lookup table where the weighted sum is computed. It is compared to the so–far found maximum which is updated accordingly. This lookup/compare operation is done in parallel to the ongoing histogram generation for further columns which produces now two more histograms every step. As soon as the local maximum in the first processor column has been found, the corresponding value is forwarded to the following column. Here it can be compared with the local maximum that has been computed exactly when the forwarded value arrives. Eventually the global maximum has been found and the two counter

contents corresponding to this track are asserted onto a look–up table. The ratio of (number of "high" pixels)/(number of "total" pixels), used for electron identification/pion rejection, is read out and compared to a programmed threshold.

A prototype of the Enable Machine has been implemented in field programmable gate arrays. This implementation has been chosen since it provides high flexibility and reduces substantially the development time. The prototype serves as a test and evaluation board. One column of the histogramming unit has been mapped into a single Xilinx 3190 chip operating at a clock cycle frequency of 40 MHz. The architecture of the Enable Machine allows easily to adjust several critical parameters. It is possible to change the trigger algorithm by reprogramming the Xilinx chips. In addition the tracks searched for as well as the threshold of the electron/pion ratio are specified by freely programmable look–up tables. Exploiting these features it is also possible to search for specific patterns along tracks or for nonlinear tracks if this should be required. The contents of the look–up tables as well as the Xilinx contents, i.e., the architecture of the Enable Machine, can be programmed from a host workstation via a VME interface. The configuration program for the Xilinx chips, e.g., can be downloaded in a few milliseconds.

The performance of the Enable Machine is essentially limited by the speed of the input data flow. With a worst case clock frequency of 40 MHz the trigger decision time is 6.5 μs and the latency 7.0 μs.

The Enable machine uses direct template matching. This approach is suggested by the simple patterns searched for—straight lines of different slope and offset. The same technique cannot be applied easily in a situation where a pattern may consist of pixels in the whole image plane, as it is the case with the circular tracks of our second example. Thus a different implementation has been chosen here.

This technique requires three steps, preprocessing of the detector data for adaption to the systolic architecture, the Parallel Hough Transform itself, and postprocessing to count the number of well defined peaks in the transform result. The block diagram of the corresponding processor is shown in Fig. 6. In the first stage a preprocessing unit is assigned to every detector wire. The signals are digitized and analyzed by a hit detector that determines the drift times t_i from the signal distributions. They are converted by table lookup to the corresponding values ϕ_i. These values ϕ_i are converted into a time–ordered bit stream which is entered into the Hough Transform processor.

The Hough transform corresponds to the mapping described above. For every input pixel it is required to increment all counters in the Hough plane which are covered by a certain straight line. The essential operation is to generate for each r_i the corresponding mask in the $(1/r_c, \phi_s)$ plane shifted by a phase angle that increases linearly with time. Now instead of shifting the mask synchronously with ϕ over the counter plane it is more appropriate to shift the contents of the counters synchronously with ϕ column by column in the oppo-

site direction. In this case the mask can be held fixed. After all ϕ are handled sequentially the Hough transform has been computed.

All lines are generated by incremental moves of tokens in upward and sideward direction. Fig. 7 demonstrates the line generation process for one line with a slope of 45 degrees. Sideward moves are restricted to one step because the contents of the counter array are shifted systolically only one column at a time. Therefore all slopes are realized by the appropriate choice of upward moves. In order to generate slopes greater than 45 degrees several counters are activated simultaneously. For slopes less than 45 degrees delays are introduced in the vertical shift registers.

The Hough processor is therefore divided into two building blocks. The first one is an array of counter elements, the second part an activation matrix for the counters. The parallel count–enable lines of each column are connected to an activation unit for each column. Each activation unit creates the right activation pattern for one line. The serial bit stream is transformed into the successive activation of counters with a fixed arrangement of delay elements and a fixed mapping of activation lines in every stage. Since the Euler operation is fully local it can again easily be computed in parallel by a systolic processor. The binary image that was created by thresholding is shifted into the Euler processor one column every time step in parallel. The processor consists of pattern matching units for both 2×2 masks. Subsequently an accumulator sums up the number of their occurrences and the difference is computed. It represents the number of tracks which is taken as the trigger criterion.

As with the Enable machine all processing stages (activation units, adders, comparators, Euler processor, integrators, subtracters) have been realized with Xilinx field programmable gate arrays [8]. The prototype (for one detector sector) consists of 21 devices in total. It operates at a clock rate of 40 MHz. The flow patterns of the distributed serial shift registers are the program of the systolic array. Since all other parts are uniform and static components which never have to be redefined this flow programming has been automated. The control flow pattern extracted from the simulation is directly converted to logic equations, the Xilinx programming cycle is initiated, and the resulting bit stream down–loaded to the Xilinx chips. The threshold used by the comparators is reprogrammable in a similar manner.

The circular tracks of our second example could be identified easily using a Hough transform since they all come from the same origin. In our last example the situation is more difficult since the rings created by Cherenkov light can be located anywhere in the image. Also additional processing steps like checking of mutual distances are required here. Therefore a correlation technique has been used to slove these tasks.

The RICH trigger processor that finds open circle pairs consists essentially of two parts (Fig. 8), a systolic processor array that operates on image pixels, and a sequencer that stores the microprogram and controls the processor array [9]. Additionally there are two memories to store input and output data. They are

necessary for testing the processor array independently of the read–out system. The systolic processor array consists of 176×160, i.e., 28,160 PEs, one per pixel. As the PEs operate in parallel, identical control signals are asserted onto each of them. This processor array is assembled of 22×20 VLSI chips where each one contains 8×8 PEs. These application specific integrated circuits (ASICs) have been designed in 2 μ CMOS standard cell technology. They are assembled on four boards of 11×10 chips each. In contrast to the Enable machine and the Hough processor which have a one–dimensional interconnection structure, the PEs are here arranged as a grid with connections from/to the next four neighbors. Topologically the PEs are interconnected as a cylinder to allow shifting of the stored image along one axis without boundary problems. Thus it is possible to read out the processor array without loosing data by shifting the image once around the cylinder. Each PE (Fig. 9) contains a pixel flip–flop that stores one of the pixels of the image to be processed. These flip–flops are first loaded with the detector image. During processing, the contents of the pixel flip–flops are modified to represent intermediate images or the result image. In addition each PE contains a mask flip–flop to store a mask image and a 6–bit counter to determine the correlation results. The operation of the counter is controlled by four "counter enable select" signals and by the contents of the pixel flip–flop.

At each clock cycle, the current state of each data input or internal information can be selected via a data multiplexer and loaded into the pixel flip–flop or the mask flip–flop. If the output of the pixel flip–flop is selected as the internal input, the current image is held without processing. By selecting the logical AND or OR of the output of the pixel flip–flop and the counter comparison output, simple image procession operations can be executed. The 8 possible inputs are selected by a 3–bit function code that switches the input data multiplexer. The mask flip–flop can be loaded in parallel to the pixel flip–flop if the "mask load" control signal is active, else the mask is held without modification. A "mask enable" control signal is used to clear all pixel flip–flops where a mask bit is set. This is used to mask out unwanted areas of the detector image. The discriminator receives data from the read–out system of the detector. These data arrive at a frequency of 40 MHz. They determine the x, y–position of a pixel in the detector. The x, y–data of pixels which fulfill certain threshold conditions are transferred to the trigger processor. They are used to address one specific PE of the array. As the x, y–data come sequentially from the read-out system, pixels are set individually in the array by asserting this pair of set signals. Alternatively data can be loaded from the input memory also at a frequency of 40 MHz. It can be accessed from the VME bus.

During the processing phase the trigger processor is controlled by a sequencer that asserts 11 identical control signals onto all VLSI chips. It consists of a memory that contains the microprogram and an address counter which is incremented by the system clock. The microprogram memory is also connected to the VME bus and can be down–loaded from there. The start and the end

address of the microprogram is selected by programmable registers, so that it is possible to store several programs. Eventually it has to be checked whether the trigger condition (two isolated rings) is fulfilled. Therefore the remaining centers found in the processor array are counted by a pseudo adder. This is done by reading out the trigger processor column by column. At each step the pseudo adder sums up one complete column. It is implemented as two levels of look–up tables that compress one column of the 176×160 array into two bits indicating whether the column contains no pixel, one pixel, or more than one. If one pixel has been found a 1 bit counter is incremented, whose carry activates the trigger. If more than one pixel has been found the trigger signal is output immediately. For the whole trigger process 543 sequencer steps, i.e. 543 clock cycles, are necessary. With a frequency of 30 MHz the trigger decision is computed in 18.1 μs.

Because of the high number of PEs required a chip has been designed in 2 μ CMOS standard cell technology. A single PE occupies about 0.6 mm^2 silicon area including the routing space. In addition each chip has 61 I/O pads and 27 power and ground pads. Thus the total silicon area required for an array of 8×8 PEs is about 50 mm^2. This fitted well to the largest chip available to us that provided 77 mm^2. The number of pins required for data and control signals and for power and ground fitted also well the packages available which had 88 pins. The chip dimension is 3.3×3.3 cm^2, so that four printed circuit boards of size 40×40 cm^2 were be needed to set up the whole processor. Distributing the processor array over multiple boards was no problem because only point–to–point links and common signals were required. The systolic processor described above is a very powerful device. In our application 28,160 PEs are used in parallel. At its operation frequency of 30 MHz this setup represents a computing power of 844,800 million operations/s. One operation is typically a parallel shift–and–increment or a low–level image operation (AND, OR, etc.).

All processing steps can be programmed freely. Therefore it is possible to search, e.g., for rings with other diameters. Even the exact form of the object searched for can be chosen arbitrarily. Also the neighborhood for the peak detection can be programmed as needed as well as the minimum distance between valid objects and the minimum and maximum peak heights. The size of the processor array can be adjusted by adding or removing individual boards or chips. The trigger processor for the CERES experiment is in operation at CERN since May 1991.

CONCLUSIONS

The three systolic processors presented solve different pattern recognition tasks in binary images at a very high speed. It should first be noted that the development of large–scale systems of this type is less demanding than expected. The reason is the regular structure of these processors. The PEs are simple and can relatively easily be developed using CAD tools. The nearest–neighbor interconnection structure can also easily and efficiently be implemented in programmable gate arrays or in application specific integrated circuits. As an example, the third system described—the RICH trigger processor with 28,160 PEs—has completely been designed, built, and tested by to diploma students without any previous experience in computer design.

The pattern recognition tasks presented range from the recognition of straight lines to relatively complicated thresholding operations that have to be freely user programmable. However they could all be implemented on systolic architectures that use only very simple operations on nearest–neighbor connected PEs. These operations are increment or decrement of counters, and shifting of data in the spatial or time domain (shifting in the time domain corresponds to the delay of data or the forwarding over a number of data). So the question arises whether a unified architecture exists on which all three processors could be implemented. Such an architecture is under consideration and we believe that it could be used to implement the processors described above efficiently. A step towards such a general purpose processor of programmable *architecture* is the DecPeRLe machine [10], essentially a 2D field of programmable gate arrays surrounded by memories.

References

[1] Plass G., Picasso E.: The LEP Machine Design; Europhys. News, Vol. 20, No. 6 (1989) 80–91.

[2] Jarlskog G., Rein D. (Eds.): Proc. Large Hadron Collider Workshop; Rep. CERN 90–10/ECFA (1990).

[3] ATLAS collaboration: ATLAS Letter of Intent for a General–Purpose pp Experiment at the Large Hadron Collider at CERN; Rep. CERN/LHCC/92–4 LHCC/I 2 (1992).

[4] Klefenz F., Zoz R., Noffz K.–H., Männer R.: The ENABLE Machine—A Systolic Second Level Trigger Processor for Track Finding; Proc. Comp. in High Energy Physics, Annecy, France; CERN Rep. 92–07 (1992) 799–802.

[5] Badier J., Bock R.K., Busson Ph., Centro S., Charlot C., Davis E.W., Denes E., Gheorghe A., Klefenz F., Krischer W., Legrand I., Lourens W.,

Malecki P., Männer R., Natkaniec Z., Ni P., Noffz K.–H., Odor G., Pascoli D., Zoz R., Sobala A., Taal A., Tchamov N., Thielmann A., Vermeulen J., Vesztergombi G.: Evaluating Parallel Architectures for two Real–Time Applications with 100KHz Repetition Rate; IEEE Tr. Nucl. Sci., Vol. 40, No. 1 (1993) 45–55.

[6] Heuer R.D., Wagner A.: The OPAL Jet Chamber; Nucl. Instr. Meth., Vol. A265 (1988) 11–19.

[7] Faschingbauer U., Trauth M.G., Wurm J.P., Drees A., Fischer P., Glässel P., Guckes M., Irmscher D., Olsen L.H., Pfeiffer A., Ries H., Schön A., Sickmüller H., Specht H.J., Ullrich T.S., Barasch E.F., Breskin A., Chechik R., Fraenkel Z., Sauvage D., Steiner V., Tserruya I.: Proposal SPSC/P237, CERN/SPS 88–25 (1988).

[8] Klefenz F., Noffz K.–H., Zoz R., Conen W., Männer R.: Track Recognition in 4 μs by a Systolic Trigger Processor Using a Parallel Hough Transform; IEEE Tr. Nucl. Sci., Vol. 40, No. 4 (1993) 688–691.

[9] Baur R., Gläß J., Männer R.: First Experiences with a Systolic Trigger Processor for RICH Detectors; in: Watanabe Y., Abe F. (Eds.): Computing in High Energy Physics '91, Frontier Science Series; Vol. 3, Tokyo (1991) 211–218.

[10] Bertin P., Roncin D., Vuillemin J.: Programmable Active Memories: A Performance Assessment; Digital Equipment Paris Research Lab., Rep. Nr.$\tilde{2}$4 (1993).

Figure 5: Structure of the Enable Machine.

Figure 6: Block diagram of the Parallel Hough Transform Processor.

Figure 7: Line generation by appropriate control token flow.

Figure 8: Block diagram of the RICH trigger processor.

Figure 9: Structure of a processing element.