# Systolic Processors as Second Level Triggers for High Energy Physics Experiments*

R. Baur[1], J. Gläß[2], F. Klefenz[2], R. Männer[2,3], R. Zoz[2]

[1]Physikalisches Institut, Universität Heidelberg, Heidelberg, Germany
[2]Lehrstuhl für Informatik V, Universität Mannheim, Mannheim, Germany
[3]Interdisziplinäres Zentrum für Wissenschaftliches Rechnen, Universität Heidelberg, Heidelberg, Germany

**Abstract**

This paper describes two fully programmable systolic processors for second level trigger processing in high energy physics experiments at CERN, Geneva, Switzerland. One of them has been used successfully since spring 1991 at the CERES/NA45 experiment for the recognition of Cherenkov rings in a ring image Cherenkov (RICH) detector. The processor consists of a $1m^2$ systolic array of $176\times160$, i.e. over 28,000 processing elements which are packed into $22\times20$ VLSI chips that have been designed in $2\mu$ CMOS standard cell technology.

The second processor, the ENABLE machine, is under design within the EAST/RD–11 collaboration at CERN that studies the data processing problems foreseen for the next generation of colliders like LHC, SSC, etc. It has been designed for the TRD detector of the LHC as a test application where a complex trigger decision has to be taken in less than 10 $\mu$s. The ENABLE machine identifies arbitrary patterns in binary images by direct template matching. For each pattern the coincidences of the mask template and the picture under study are histogrammed. The machine is scalable in picture size as well as in the number of patterns processed concurrently. It exploits programmable gate array technology so that its architecture can be programmed into standard hardware.

Both architectures and the consequences of their different implementation are discussed.

## 1. Introduction

High energy physics is among the few fields that have extremely high computing requirements on all levels of complexity of the data processing required. Data are delivered by large detectors in such a high volume and at such a high rate that they have to be reduced in real–time in several steps before there is any chance to analyze them in some detail or to store them for off–line analysis. Traditionally these steps are divided into first, second, and third level triggers, and on–line and off–line data processing. First level triggers operate on a time scale comparable to gate delays, usually of ECL logic, i.e., in the nsec range. Typically simple logical functions of input parameters are computed here. Second level triggers work in the $\mu$sec range

*Part of this work has been performed within the EAST/RD–11 collaboration of CERN, Geneva, Switzerland, and has been supported by the Gesellschaft für Schwerionenforschung Darmstadt (GSI), Darmstadt, Germany.

and perform considerable more complicated operations. Most of them require the recognition of relatively simple patterns in pixel images but at the extremely high rate of, e.g., $10^5$ to $10^6$ images/s. This task can be solved neither by special–purpose hardware nor by supercomputers, not even by traditional multiprocessor systems. Multiprocessors which exploit parallelism on the task level operate on the msec range and are used as third level triggers.

In this paper we concentrate on systolic processors which seem to be particularly well suited as second level triggers. Systolic processors belong to the class of SIMD (single instruction multiple data stream) machines. These consist of a high number of identical processing elements (PEs) which all are controlled by a unit that issues the same command to them. Parallel processing therefore takes place on the instruction level, i.e., the same instruction is executed by many PEs concurrently on data that are assigned individually to them. In systolic SIMD architectures data are passed from one PE to the next neighbor synchronously. Since the problem of partitioning an application into sufficiently many independent tasks, as it is usually encountered in multiprocessors, does not exist here, SIMD machines are well suited for exploiting massive parallelism. In addition since all PEs are identical and relatively simple, it is straightforward to set up arrays of such PEs by CAD for an implementation in VLSI or programmable logic.

Our group has been involved in the development of systolic processors at the Physikalisches Institut, Universität Heidelberg, and at the Lehrstuhl für Informatik V, Universität Mannheim [11]. Below we describe two of them built for second level trigger processing. The first one has been used successfully since spring 1991 for the recognition of Cherenkov rings in a ring image Cherenkov (RICH) detector. The second one is under design within the EAST/RD–11 collaboration at CERN that studies the data processing problems foreseen for the next generation of colliders like LHC, SSC, etc. The paper discusses both architectures and the consequences of their different implementation in semi–custom VLSI and programmable gate arrays.

## 2. The 2nd Level Trigger Processor for CERES/NA45

The first systolic processor described here is used as a programmable processor for the recognition of Cherenkov rings in a RICH detector in the CERES/NA45 experiment [3] at CERN. The detector outputs continously images of $160\times160$ pixels which have to be analyzed in respect to the conditions described below. Every image has to be processed within 20 $\mu$s. Due to the short time available massive parallelism is exploited by assigning one PE to every pixel. The CERES trigger processor therefore consists of a systolic array of $176\times160$ PEs (the additional 16 PEs are required to avoid boundary problems). It is controlled by a sequencer that can be programmed freely via VME bus and operates at 40 MHz during data read–in and at 30 MHz during data processing.

The CERES/NA45 experiment investigates the production of $e^+e^-$ pairs. These and an enormous amount of secondary particles are detected via ill–defined circles that are generated by the small number of photons in the Cherenkov cone around the direction of the particles. The radius of these Cherenkov rings depends on the particle

type. Most of the ring pairs are uninterestingly "close" and have to be suppressed. Interesting images contain at least one "open" electron pair, i.e., two rings with a certain minimum distance. For each image it is therefore required 1) to identify all relatively well defined rings of a given diameter, 2) to drop all ring pairs which do not have a certain minimal distance, and 3) to count all remaining rings.

The identification of rings is achieved by a two–dimensional correlation of the input image with a mask image, a ring of the nominal diameter. The correlation result is computed in parallel by shifting the input image over a counter plane along such a ring and by incrementing in each step all counters that are covered by pixels in the input image. The result of this correlation is a two–dimensional distribution whose analog values $f(x, y)$ represent the probability to find the ring center at the position $(x, y)$ of the input image. In the correlation result high peaks correspond to well defined rings. The next step consists of finding sufficiently high peaks, i.e., the position of sufficiently well defined rings. Due to binning effects, proper thresholding followed by a thinning operation has to be used to locate their positions, i.e., to compute a single pixel per peak. The new intermediate image consists of individual pixels representing the remaining centers of well defined rings. To drop all rings that are too close a second correlation step is done. To check all mutual distances in parallel the intermediate image is correlated with a circular disk of a radius equal to the minimum distance. In this two–dimensional correlation result all pixels $(x, y)$ with a function value $f(x, y) = 1$ correspond to "isolated" ring centers which do not have neighbor pixels within the minimum distance. By deleting all pixels with a function value $> 1$, a result image is formed that contains pixels only at positions where an isolated ring with the radius searched for was found in the detector image. The generation of the trigger signal requires to determine whether at least two ring centers remained. This is done by shifting the result image out of the systolic array into an adder. The whole operation requires 543 sequencer steps, i.e. 543 clock cycles. With a frequency of 30 MHz the trigger decision is computed in 18.1 $\mu$s. A detailed description of the pattern recognition process is given elsewhere [4, 2].

The complete CERES trigger processor consists of four parts, a discriminator that preprocesses the detector data, a systolic processor array that filters out rings with a minimum distance, a sequencer that stores the microprogram and controls the processor array, and a pseudo adder that determines whether there were at least two rings with a minimum distance (see Fig. 1). In the following we restrict on the systolic processor array. The $176 \times 160$, i.e., of 28,160 PEs are assembled of $22 \times 20$ custom VLSI chips where each one contains $8 \times 8$ PEs. These semi–custom chips have been designed on a VENUS system [8] in 2 $\mu$m CMOS standard cell technology. The PEs are arranged as a grid with connections from/to the next four neighbors. The chips are placed onto four boards of $11 \times 10$ chips each. 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. The $160 \times 176$ identical PEs are controlled by a sequencer that asserts identical control signals onto all VLSI chips.

The CERES trigger processor is a very powerful device. At its operation fre-

Figure 1: The CERES/NA45 trigger processor.

quency of 30 MHz the concurrent operation of 28,160 PEs represents a computing power of 844,800 million operations/s where one operation is a parallel shift–and–increment or a low–level image operation (AND, OR, etc.). Our experience has shown the importance of the programmability of all processing steps. In the CERES application the trigger algorithm has been improved several times, e.g., by adding steps to suppress detector noise due to bias problems in the front–end electronics.

## 3. The ENABLE Machine for the TRD Detector at the LHC

The second system is a systolic processor for the identification of particle tracks in the TRD detector of the LHC collider. According to the constraints of the EAST benchmark specifications a flexible special–purpose processor, the ENABLE machine, has been designed to solve the demanding pattern recognition problems encountered in LHC. It is a modular, scalable, and upgradeable system which can be customized to specific application needs. The implementation in field–programmable gate array devices (FPGA) has been chosen because it provides flexibility in custom designed solutions, and reduces substantially the development time as our experience has shown in other related high energy physics experiments [11].

In order to identify electron and pion tracks in the binary detector image the ENABLE machine is composed of two main functional building blocks; a histogram generation unit, and a programmable table look–up unit for maxfinding and other related functions. The architecture of the ENABLE machine obeys the parallel com-

puting paradigms proposed by W. Händler [6]. All guidelines applicable to SIMD computers are reflected in the architecture of the ENABLE machine as outlined below.

Figure 2: Schematic layout of the ENABLE Machine.

Evolution. The ENABLE machine evolved from its ancestor, a parallel Hough transform machine [10, 9, 11] which has been devised as a second level trigger for drift chambers like OPAL [7] of LEP. Most functional parts of this ancestor machine could be incorporated in the new machine design. The main building blocks are XILINX field programmable gate arrays which perform specific subtasks. We have elaborated a library pool of standard application modules like adders, comparators, counters etc., comparable to the devices in the 74xxx series of TTL parts, which can easily be customized to new architectures.

Universality. In Fig. 2 a schematic overview of the architecture of the ENABLE machine is given. This machine identifies arbitrary patterns in binary images by direct template matching. For a number of predefined patterns, which are stored in successive RAM–blocks, the coincidences of the mask template and the picture under study are histogrammed in parallel. This is done by shifting the pixel image one column at a time into the mesh connected array of processsing elements of the ENABLE machine. In doing so the parallel compare–and–increment operation is performed on different columns for different templates concurrently according to the

underlying space–time correlation and routing procedure. Since the image columns are pipelined through the processor array one histogram result is computed every step and can be analyzed in the trigger decision unit to derive the classification result. It should be noted that a big class of binary templates can be specified which makes the ENABLE machine applicaple to many different pattern matching problems.

Homogeneity/Simplicity. The ENABLE machine consists basically of a correlation layer, an image layer, and a histogramm layer. Each one is composed of identical processor elements of very modest functionality. The data flow in each layer is unidirectional, i.e., a simple feed–forward architecture has been adopted. The image layer consists of a rectangular grid of flip–flops which store the binary image. Each PE in the image layer is only connected to its nearest neighbour to the left. In the correlation layer the image is correlated with the mask pattern by AND–ing the image and mask pixels together. The histogramming units accumulate the results and store the final channel content for each pattern after the correlation process has been completed. The homogeneity of the approach is obtained by the SIMD paradigm applied, i.e., the same operation is applied to multiple data streams by a mesh of PEs. The systolic data flow is maintained thoroughly from data input up to result storage in registers.

Scalability. According to the TRD requirements the prototype board can handle pictures of a height of 60 pixels. Any width can be handled as long as there is no result overflow since the columns of the image are handled sequentially. The prototype board is able to histogram up to $k \times 60$ patterns in parallel. Depending on the application the system throughput can be increased by scaling the processor in three different dimensions. First the image height can be multiplied by adding boards vertically so that each column of boards operates on a different column of the image. Second the number of patterns can be increased by adding more columns of boards horizontally. Third the images can be split in the dataflow direction into subimages which can be processed concurrently by separate matrixes of boards. Since all boards are fully utilized after the pipeline set–up delay the total trigger frequency scales nearly with (clock frequency)$/n$ where $n$ is the number of subimages. For this operation mode the partial histogram results have to be summed up.

Nearest neighbour. All processing elements operate synchronously in a lock–step fashion. They are connected only to their nearest neighbours so that a uniform floorplan can be achieved on the chip level. This results in a highly regular interconnection and routing scheme in each layer which is the base of the inherent scalability. The processing time is given by the number of clock cycles, i.e., the time to pipe the data through the systolic array.

Space sharing. For read–out of the front–end electronics of the TRD the HIPPI bus has been chosen. The ENABLE machine will therefore be equipped with a HIPPI interface [1]. The classification results will be transferred to a host machine for further processing. The systolic array is composed of XILINX chips and can be programmed on a SUN workstation in standard high–level languages under UNIX and X–Windows. The prototype board will serve as a test and evaluation board for the TRD emulator. In conjunction with the detector emulator [5] model assumptions can be tested and evaluated.

The speed of the ENABLE machine can be given in terms of frequency and latency. Frequency is defined here as the trigger decision rate and latency as the time to fill the pipeline before the first result is available. The operation is deadtime free because resetting of histogram channels is done in a pipelined fashion. New data can be shifted in without wait times. The architectural studies have been completed and the final hardware scheme has been devised. Several possibilities for design optimization have been considered and integrated into the final design. The layout of every unit is in progress. Worst–case simulation studies indicate a clock cycle frequency of 40 MHz. The simulation has shown the functionality of the machine in terms of electron/pion discrimination, latency, and speed. A prototype which will serve as a test and evaluation board is under development. The prototype will be realized with modest cost and will be functionally testable in 1992.

## 4. Implementation Differences and Consequences

The two systolic arrays described here are special–purpose processors for low–level pattern recognition tasks. Due to the extreme speed requirements given, massive parallelism has been exploited by assigning one PE to every pixel of the images to be processed. Due to the high regularity of systolic arrays and the relative simple structure of the PEs a VLSI implementation could be done easily.

In the first case we developed a chip in standard cell technology. From the detailed circuit design up to the test of the first samples only one man–year had to be invested. Most of the time has been spent for optimizing the PE for high–speed operation and dense layout. The first samples fulfilled their specifications completely. It took several months more to set up the whole processor and to adjust all timing requirements on the 1 $m^2$ area which is completely filled with pin grid arrays operating synchronously at 40 MHz. This task would have been considerably easier with a full–custom chip in $\leq 1$ $\mu$m technology. In this case it would probably have been possible to pack $16\times16$ PEs into one chip, reducing the processor size to about $50\times50$ $cm^2$, and allowing an even higher operating frequency.

In the second case we avoided the chip design process by applying programmable gate arrays. From the installation of the design software to the simulation of the first chips only two man–months had to be invested. Again most of the time has been spent for optimizing the PEs for high–speed operation and dense layout. Since no chips have been programmed yet we are not sure about the final operation frequency. It should be comparable, however, to the custom design chips of our first processor. However the much smaller layout density makes this technology less attractive for massive parallel processors in case enough time and funding is available for a full custom design.

# References

[1] T. Anguelov, R. MacLaren: The HIPPI Daughter Board of SLATE; EAST note 91–12 (1991).

[2] R. Baur, J. Gläß, R. Männer: First Experiences with a Systolic Trigger Processor for RICH Detectors; Proc. Comp. in High Energy Physics '91, Tsukuba City, Japan (1991) 211–218.

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

[4] J. Gläß, R. Baur, R. Männer: Programmable Trigger for Electron Pairs in Ring Image Cherenkov Counters; IEEE Trans. Nucl. Sci., NS–37 (1990) 241–247.

[5] B. Green, N. Mihai, J. Strong: SLATE Hardware; EAST note 91–13 (1991).

[6] W. Händler: Nature Needed Billions of Years...; Proc. Parallel Computing Technology '91, Novosibirsk, USSR (1991) 1–16.

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

[8] E. Hörbst, M. Nett, H. Schwärtzel: VENUS—Entwurf von VLSI–Schaltungen; Springer, Berlin, Germany (1986).

[9] F. Klefenz, R. Männer: A Systolic Track Finding Trigger Processor; Proc. Comp. in High Energy Physics '91, Tsukuba City, Japan (1991) 219–222.

[10] F. Klefenz, R. Männer: Echtzeit–Spurerkennung mit einem systolischen Arrayprozessor; in: Burkhardt H., Höhne K.H., Neumann B. (Eds.): Mustererkennung 1989; Informatikfachberichte 219, Springer, Berlin (1989) 401–405.

[11] R. Männer, J. Gläß, F. Klefenz: Massively Parallel Systolic Processors for High–Speed Recognition of Simple Patterns; Proc. Parallel Computing Technology '91, Novosibirsk, USSR (1991) 98–108.