CERN/LHCC/98-36
30 October 1998

INFN, Section of Rome

University of Rome "La Sapienza"

Proceedings of the Fourth Workshop on Electronics for LHC Experiments

Rome, September 21-25, 1998

Organised by INFN, Section of Rome
and the University of Rome "La Sapienza"
on behalf of the CERN LHCC Electronics Board

FAST PARTICLES IDENTIFICATION IN PROGRAMMABLE FORM AT LEVEL-0 TRIGGER BY MEANS OF THE 3D-FLOW SYSTEM.

D. Crosetto, 3D-Computing, Inc., 900 Hideaway Pl., DeSoto, TX
(email: crosetto@bonner-ibm2.rice.edu)

Abstract

The 3D-Flow Processor system is a new, technology-independent concept in very fast, real-time system architectures. Based on either an FPGA or an ASIC implementation, it can address, in a fully programmable manner, applications where commercially available processors would fail because of throughput requirements. Possible applications include filtering-algorithms (pattern recognition) from the input of multiple sensors, as well as moving any input validated by these filtering-algorithms to a single output channel. Both operations can easily be implemented on a 3D-Flow system to achieve a real-time processing system with a very short lag time. This system can be built either with off-the-shelf FPGAs or, for higher data rates, with CMOS chips containing 4 to 16 processors each. The basic building block of the system, a 3D-Flow processor, has been successfully designedin VHDL code written in "Generic HDL" (mostly made of reusable blocks that are synthesizable in different technologies, or FPGAs), to produce a netlist for a four-processor ASIC featuring 0.35 micron CBA (Cell Base Array) technology at 3.3 Volts, 884 mW power dissipation at 60 MHz and 63.75 mm sq. die size. The same VHDL code has been targeted to three FPGA manufacturers (Altera EPF10K250A, ORCA-Lucent Technologies OR3T165 and Xilinx XCV1000). A complete set of software tools, the 3D-Flow System Manager, equally applicable to ASIC or FPGA implementations, has been produced to provide full system simulation, application development, real-time monitoring, and run-time fault recovery. Today's technology can accommodate 16 processors per chip in a medium size die, at a cost per processor of less than \$5 based on the current silicon die/size technology cost.

1. TECHNICAL OBJECTIVES AND APPROACH FOR A TECHNOLOGY-INDEPENDENT IMPLEMENTATION

1.1 Area of application

The purpose of this project is to design a programmable, scalable, and modular solution for high-speed, front-end applications. Not long ago, front-end electronics were built with analog techniques using discrete components. Later, with the rapid advances in digital technology, Digital Signal Processors (DSPs) replaced analog circuitry up to certain speeds. However, in many applications the user still had to design a specific hardware to implement an algorithm on the front-end signal from a detector (or sensors) because the DSPs were not fast enough or flexible enough.

The throughput of this system can fetch as many input data (16-bit for a 16-bit-wide bus structure of the processor) per second from a device (detector) as clock cycles of the technology implementation, yet unlike currently available systems of comparable speed, it is fully programmable and extremely flexible.[i-ii-iii-iv-v]

1.2 How to design a technology-independent application

The quality and the level of a technology-independent solution are determined by several factors, the most important of which is avoiding the selection of an architecture that leaves insufficient margins in speed or performance beyond the present requirements.

This decision comes at a very early stage, when a solution to a problem is conceived. For example, in the case of the problem of processing thousands of input data at a continuously sustained input data rate of 40 MHz, use of the approach of the 3D-Flow system and selection of a processor speed of at least 80 MHz places the user in a safe position because:

a) the architecture approach allows in principle the acquisition of input data at the same processor speed, and the factor two between input data rate and processor speed is a safe margin;
b) the architecture allows one to run more complex algorithms simply by adding 3D-Flow layers (see Section 2.4) to the system, without having to redesign the entire system. This provides cost optimization for each application without the need to replace the original processors with faster ones;
c) besides de-coupling the parallel processing system from the sensor device, the FIFOs memory at each input port enables the system to sustain input data at a higher peak rate for short periods of time with respect to the nominal 40 MHz.

Other factors that will determine the quality and level of technology independence of a system include:

1. The code should be written in "Generic HDL."
2. The coding "style" should be targeted to ASIC without having an architectural knowledge of the basic elements and routing characteristics of one FPGA versus another (e.g., CLBs from Xilinx versus LEs from Altera versus PFUs from Lucent Technologies).

3. When implementing a design usingFPGAs, it is best not to make use of macros (besides the memory blocks), or to manually floor plan the design. The use of macros makes the design not ready for simulation; on the contrary, an HDL behavioral model equivalent to the functionality of the macro must be written. One should submit the same HDL code to the powerful tools provided by the different vendors (Altera, Lucent Technologies and Xilinx) and let the tools optimize the floor plan and routing. In the event one would like to modify the circuit, the additional time needed to go through some manual phases will cost more than purchasing the new component with increased performance.
4. In the case of an implementation in ASIC, it is worthwhile to synthesize the design using different vendor libraries (Cell Base Array, Gate Array, etc.). Some companies are providing tools that help to submit a design using different vendor libraries, keeping track of file handling and results handling for an easy comparison of different technology performances on a specific design.

1.3 Advantages of a technology-independent application versus "ad hoc" FPGAs or ASICs solutions

The cost and time required to design a circuit is minimal with respect to the time required to develop all the tools for testability, simulating, troubleshooting, writing documentation and maintenance of the circuit, and for making provisions to easily implement modifications that one might think beneficial in the future.

The 3D-Flow System Manager (see Section 3) provides the same information and features to the user regardless of the technology chosen to implement the circuit (FPGA: Altera, Xilinx, Lucent Technology, orASICs).

The same VHDL code is used in all cases: the same tools to initialize the circuits, to download the algorithm through RS232, and to monitor the system in real time through the scratch pad register file in the silicon that memorizes consecutive cycles (see Section 7).

The tools to create a new application with a different system size, different algorithm, different speed, and different input data rate, and to display all information of the system remain the same.

The only variable is the filtering algorithm (consisting of 10 to 20 lines of code that the tools of the 3D-Flow System Manager help the user to generate) that will be downloaded into the system, while an "ad hoc" circuit (FPGA or ASIC) implementing a specific algorithm will require one to develop a set of different tools for testability, accessibility, and maintenance for each application in addition to the need to change the circuit.

2. A SINGLE COMPONENT FOR SEVERAL APPLICATIONS

The main characteristics of the 3D-Flow system architectures based on a single 3D-Flow component are the following:

2.1 System level

Objective:
Oriented toward data acquisition, data movement, pattern recognition, data coding and reduction.
Design considerations:
- Quick and flexible acquisition and exchange of data, but not necessarily in fully bi-directional manner.
- Possibility of dedicating small area to program memory in favor of multiple processors per chip and multiple execution units per processor, data-driven components (FIFOs, buffers), and internal data memory. (Most algorithms that this system aims to solve are short and highly repetitive, thus requiring little program memory.)
- Balance of data processing and data movement with very few external components.
- Programmability and flexibility provided by enabling downloading of different algorithms into a program RAM memory.
- High priority of modularity and scalability, permitting solutions for many different types and sizes of applications using regular connections and repeated components.

2.2 System architecture

The goal of this parallel-processing architecture is to acquire multiple data in parallel (up to the clock speed of the technology implementation) and to process them rapidly, accomplishing digital filtering on the input data, pattern recognition, data moving, and data formatting.

The system is suitable for "particle identification" applications in HEP (calorimeter data filtering, processing and data reduction, track finding and rejection).

The compactness of the 3D-Flow parallel-processing system in concert with the processor architecture allows processor interconnections to be mapped into the geometry of sensors (such as detectors in HEP) without large interconnection signal delay, enabling real-time pattern recognition. This work originated by understanding the requirements of Level-1 (and Level-0) triggers for different experiments, past and present, as well as future trigger designs. Each one has been studied in some detail, with visits to the site of the experiment when possible and attempts to define a system architecture, processor architecture, and assembly architecture that had the commonality features to implement all of them. To maintain scalability with regular connections in *real time*, a three-dimensional model was chosen, with one dimension essentially reserved for the unidirectional time axis and the other two as bi-directional spatial axes (Fig. 1).

The system architecture consists of several processors arranged in two-orthogonal axes (called layers; see Fig. 2), assembled one adjacent to another to make a system (called a stack; see Fig. 3). The first layer is connected to the input sensors, while the last layer provides the results processed by all layers in the stack.

Data and results flow through the stack from the sensors to the last layer. This model implies that applications are mapped onto conceptual two-dimensional grids normal to the time axis. The extensions of these grids depend upon the amount of flow and processing at each point in the acquisition and reduction procedure.

Four counters at each processor arbitrate the position of the bypass/in-out switches in order to achieve the proper routing of data. An image-processing application fits this model quite closely. When new data arrive or when the reduction possible with the program executing in one plane is finished, the intermediate data are transferred to the next plane, which has a number of processing elements compatible with the new data extension. Higher-dimensional models were considered too costly and complex for practical scalable systems, mainly due to interconnection difficulties.

2.3 *Processor architecture*

The 3D-Flow processor is a programmable, data stream pipelined device that allows fast data movements in six directions with digital signal-processing capability. Its cell input/output is shown in Figure 1.

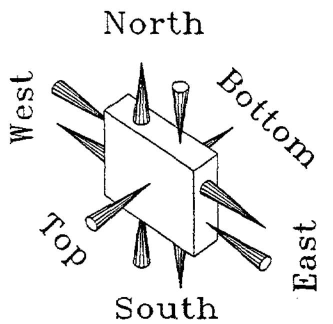

Figure 1. 3D-Flow input/output.

The 3D-Flow operates on a data-driven principle. Program execution is controlled by the presence of the data at five ports (North, East, West, South, and Top) according to the instructions being executed. A clock synchronizes the operation of the cells. With the same hardware one can build low-cost, programmable Level-1 triggers for a small and low-event-rate calorimeter, or high-performance, programmable Level-1 triggers for a large calorimeter capable of sustaining up to one event per clock.

The 3-D Flow processor is essentially a Very Long Word Instruction (VLIW) processor. Its 128-bits-wide instruction word allows concurrent operation of the processor's internal units: Arithmetic Logic Units (ALUs), Look Up Table memories, I/O busses, Multiply Accumulate and Divide unit (MAC/DIV), comparator units, a register file, an interface to the Universal Asynchronous Receiver and Transmitter (UART) used to preload programs and to debug and monitor during their execution, and a program storage memory.

The high-performance I/O capability is built around four bi-directional ports (North, East, South and West) and two mono-directional ports (Top and Bottom). All of the ports can be accessed simultaneously within the same clock cycle. N, E, S, and W ports are used to exchange data between processors associated with neighbouring detector elements within the same layer. The Top port receives input data and the Bottom port transmits results of calculations along successive layers.

A built-in pipelining capability (which extends the pipeline capability to the system) is realized using a "bypass mode." In bypass mode, a processor will ignore data at its Top port and automatically transmit it to the Top port of the processor in the next layer. This feature therefore provides an automatic procedure to route the incoming events to the correct layer. Several 3D-Flow processing elements, shown in Figure 1, can be assembled to build a parallel processing system, as shown in Figure 2.

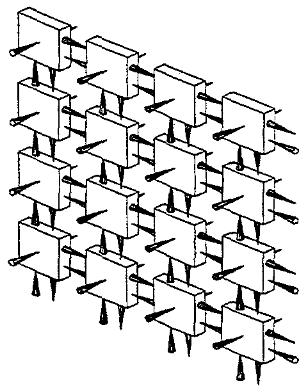

Figure 2. One layer (or stage) of 3D-Flow parallel processing.

2.4 *Introducing the third dimension in the system*

In applications where the processor algorithm execution time is greater than the time interval between two data inputs, one layer of 3D-Flow processor is not sufficient.

The problem can be solved by introducing the third dimension in the 3D-Flow parallel-processing system, as shown in Figure 3.

In the pipelined 3D-Flow parallel-processing architecture, each processor executes an algorithm on a set of data from beginning to end (e.g., the event in HEP experiments, or the picture in graphics applications).

Data distribution of the information sent by the calorimeter as well as the flow of results to the output are controlled by a sequence of instructions residing in the program memory of each processor.

Each 3D-Flow processor in the parallel-processing system can analyze its own set of data (a portion of an event or a portion of a picture), or it can forward its input to the next layer of processors without disturbing the internal execution of the algorithm on its set of data (and on its neighboring data set at North, East, West, and South that belongs to the same event or picture).

The programming of each 3D-Flow processor determines how processor resources (data moving and computing) are divided between the two tasks or how they are executed concurrently.

A schematic view of the system is presented in Figure 3, where the input data from the external sensing device are connected to the first layer (or stage in Fig. 3) of the 3D-Flow processor array.

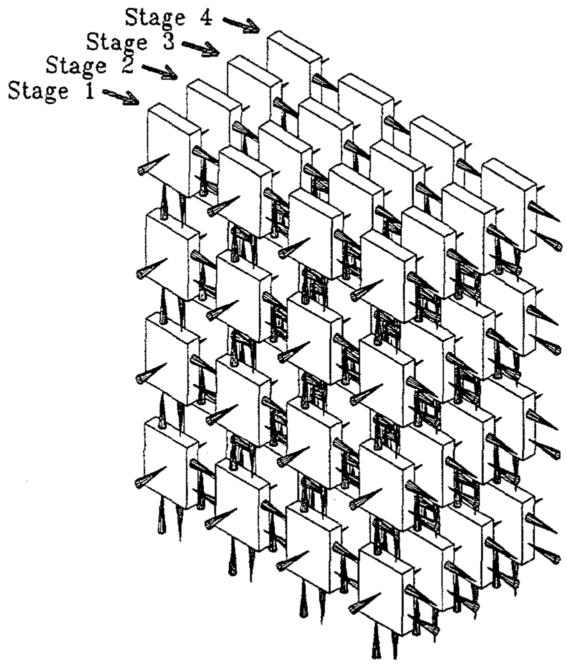

Figure 3. General scheme of the 3D-Flow pipeline parallel-processing architecture.

The main functions that can be accomplished by the 3D-Flow parallel-processing system are:

- Operation of digital filtering on the incoming data related to a single channel;
- Operation of pattern recognition to identify particles; and
- Operations of data tagging, counting, adding, and moving data between processor cells to gather information from an area of processors into a single cell, thereby reducing the number of output lines to the next electronic stage.

In calorimeter trigger applications, the 3D-Flow parallel-processing system can identify particles on the basis of a more or less complex pattern recognition algorithm and can reduce the input data rate and the number of input data channels.

In real-time tracking applications, the system calculates track slopes, momentum, pt, and the extrapolated co-ordinates of a hit in the next plane.

Figure 4 shows the timing (at the bunch crossing rate) of the input data to each layer (or stage) and the algorithm execution time (latency) in the 3D-Flow pipelined architecture.

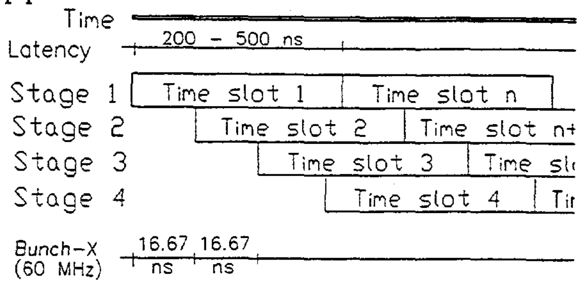

Figure 4. Timing diagram of four 3D Flow pipelined layers (or stages).

3. 3D-FLOW SYSTEM MANAGER FOR A TECHNOLOGY-INDEPENDENT IMPLEMENTATION

The 3D-Flow System Manager is a set of tools that allows the user to:

1. create a new 3D-Flow application (called project) by varying size, throughput, filtering algorithm, and routing algorithm, and by selecting the processor speed, lookup tables, number of input bits and output results for each set of data received for each algorithm execution;
2. simulate a specified parallel-processing system for a given algorithm on different sets of data. The flow of the data can be easily monitored and traced in any single processor of the system and in any stage of the process and system; and
3. monitor in real-time a 3D-Flow system via the RS232 interface, whether the system at the other end of the RS232 cable is real or virtual.

A flow guide helps the user through the above three phases.

A system summary displays for a 3D-Flow system created by the 3D-Flow System Manager the following information:

1. characteristics such as size, maximum input data rate, processor speed, maximum number of bits fetched at each algorithm execution, number of input channels, number of output channels, number of layers filtering the input data, number of layers routing the results from multiple channels to fewer output channels;

2. time required to execute the filtering algorithm and to route the results from multiple channels to fewer output channels.

A log file retains the information of the activity of the system when:

1. loading all modules in all processors;
2. initializing the system;
3. recording all faulty transactions detected in the system (e.g., data lost because the input data rate exceeded the limit of the system or because the occupancy was too high and the funnelling of the results through fewer output channels exceeded the bandwidth of the system);
4. recording any malfunction of the system for a broken cable or for a faulty component.

A result window can be open at any time to visualize the results of the filtering or pattern recognition algorithm applied to the input data as they come out at any layer of the system.

The generation of test vectors for any processor of the system can be selected by the user at any time to create the binary files of all I/O's corresponding to the pins of a specific FPGA or ASIC chip. These vectors can then be compared with those generated by the chip itself or by the VHDL simulation.

4. IMPLEMENTATION USING OFF-THE-SHELF COMPONENTS

Figure 5 illustrates an implementation of the 3D-Flow system using off-the-shelf components.

For a lower input data rate, the entire 3D-Flow system can be built with off-the-shelf components by using:

a) any one of the following FPGAs: Altera EPF 10K250A, Lucent Technology OR3T165, or Xilinx XCV300 to implement the 3D-Flow with an 8-bit-wide bus structure and 16-bit precision; the Xilinx XCV1000 can implement the 3D-Flow with a 16-bit-wide bus structure and 32-bit precision. The speed performance (without any macro instantiation, besides memory blocks, and without floor planning manual optimization) ranges from 12 to 16 MHz.

b) chips from National Semiconductor 16-40 MHz 10-Bit Bus LVDS (Low Voltage Differential Signaling) Serializer and Deserializer DS92LV1021 and DSLV1210 chip set for board-to-board communication. It features 400 Mbps serial bus LVDS bandwidth (at 40 MHz clock). It may transmit data over heavily loaded back-planes, or 10-meter cable, or unshielded twisted pair cable. The LVDS low-differential voltage is 350 mV at 3.5mA.

The same VHDL code for one 3D-Flow processor with 8-bit bus width has been targeted to any of the three FPGAs mentioned above from Altera, Lucent Technologies or Xilinx.

5. 4- VERSUS 16-PROCESSOR ASIC

For applications with higher input data rate, from four to sixteen 3D-Flow processors can be implemented into a single ASIC, and the LVDS Serializer, Deserializer function can be incorporated into the ASIC's I/O pin drivers/receivers.

The VHDL code for the chip is written in "Generic HDL" using a "style" targeted to ASIC. The netlist for a 3D-Flow ASIC of four 16-bit processors exists for a 0.35 micron CBA technology at 3.3 Volt. The simulation shows dissipation of 884 mW at 60 MHz and a die size of 63.75 mm sq.

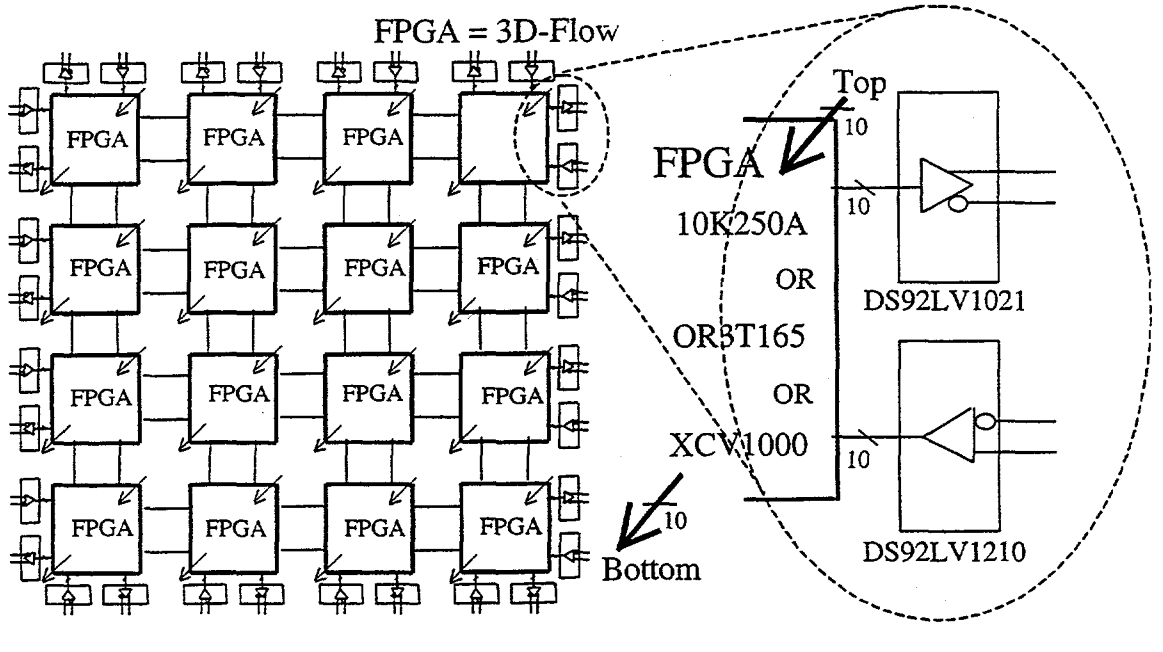

Figure 5. One board with 16 3D-Flow processors implemented using off-the-shelf components.

With the use of the LVDS interface technology now available from a few silicon vendors, the number of I/O pins required for the inter-chip and inter-board communication between 3D-Flow processors is a simplified problem, resulting in lower cost in connectors and ASIC packaging.

LVDS can transmit data quicly and at low power. For example, the commercially available component DS90LV031A LVDS quad CMOS differential line driver has an idle power state of 13 mW, and the 3.5 mA loop current from the driver will develop a differential voltage of 350 mV across the 100 ohm termination resistor, which the receiver detects with a 250 mV minimum differential noise margin.

With the availability of the LVDS driver, the 3D-Flow chip is not an I/O bound chip; thus the die can easily accommodate 16 identical processors (the circuit shown in Figure 5 accommodating 16 FPGAs --3D-Flow processors— on a printed circuit board, could be implemented into a single ASIC), lowering the cost per processor to less than $5 based on the current silicon die-size/technology cost.

6. TIMING AND SYNCHRONIZATION

The 3D-Flow system is synchronous. This makes it easier to debug and to build.

The most important task is to carry the clock, reset and trigger signals to each 3D-Flow component pin within the minimum clock skew. (The overall task is easier if each component accommodates 16 processors.)

This task can be accomplished without using special expensive connectors, delay lines, or sophisticated expensive technology since the processor speed required to satisfy the design is running at only 80 MHz. The expected worst clock skew for the distribution of one signal to up to 2,916 chips (equivalent to a maximum of 46,656 processors), using components PECL 100E111L or DS92LV010A Bus LVDS Transreceiver, is less than 1 ns according to the worst skew between different components.

This task is not difficult to achieve with the aid of today's powerful printed circuit board layout tools that can make required traces of equal length.

The other consideration in building the 3D-Flow system is that all input data should be valid at the input of the first layer of the 3D-Flow system at the same time.

All other signals in the 3D-Flow system are much easier to control than for any other system (given the modularity of the 3D-Flow approach) because they are of short distance, reaching only the neighboring components.

7. HOST COMMUNICATION AND MULFUNCTION MONITORING

An essential part of the 3D-Flow design is that every single processor is individually accessible by a supervising host, via an RS-232 line. In addition to providing the ability to download and initialize the system, this feature also provides the capability to periodically test the processor's performance by down-loading test patterns and/or test programs. A continuous monitoring can be performed by reading through RS232 the status of eight consecutive cycles of all processors and comparing them with the expected ones. These status bits are saved into a silicon scratch pad register at the same time in all processors at a pre-recorded trigger time corresponding at a selected line of the program executing the filtering algorithm in a selected layer.

In the case of suspected or detected malfunction, the processor performance could be tested remotely and its performance diagnosed. In the event of catastrophic malfunction (e.g. a given processor completely failing to respond, or a broken cable), normal operation, excluding the sick processor (or connection), can still be maintained by downloading into all the neighbors a modified version of the standard algorithm, instructing them to ignore the offending processor.

Obviously physics considerations would dictate whether such a temporary fix is acceptable, but it is a fact that the system itself does contain the intrinsic capabability of fault recovery, via purely remote intervention.

8. ACKNOWLEDGMENTS

This work was supported by DoE contract DE-FG03-95ER81905. The author would like to acknowledge also Sergio Conetti, Brad Cox, Albert Werbrouck, Billy Bonner, Abdul Akbari, Duccio Lugari, Shi Bing, Gloria Corti, Jeannine Robertazzi, and Walter Selove.

9. REFERENCES

[i] Crosetto, D., Ninth Conference on Real-Time Computer Applications in Nuclear, Particle, and Plasma Physics. MSU, East Lansing, MI, May 23-26, 1995. IEEE Transactions in Nuclear Science, Feb. 1996.

[ii] Conetti, S. and Crosetto, D., 'Implementing level-1 trigger," IEEE Transaction on Nuclear Science, Feb 1996.

[iii] Alderighi, M, Crosetto, D., et al., (ICA3PP-95). IEEE 0-7803-2018-2195. Vol. 2, pp. 761-763.

[iv] Crosetto, D., "Massively Parallel-Processing System with 3D-Flow Processors." Published by IEEE Computer Society. 0-81816-6322-7194, pp. 355-369.

[v] Crosetto, D., "Programmable Level-1 Trigger with 3D-Flow Array," Computing in HEP, San Francisco, CA, 21-27 April 1994. Editor: S.C. Loren, pp. 57-61.