1

THE DØ LEVEL 1.5 CALORIMETER TRIGGER

D. Edmunds, S. Gross, and P. Laurens*,
for the DØ Collaboration,
Department of Physics and Astronomy,
Michigan State University, E. Lansing, MI

Abstract

We describe the design, implementation, and performance of the Level 1.5 Calorimeter Trigger used by DØ, an elementary particle physics experiment at the Fermi National Accelerator Laboratory. This system is based on 12 Texas Instruments TMS320C40 processors. It uses 11 Local DSP's, operating in parallel, to search for localized objects (*e.g.* electrons) within DØ's 2560 channels of digitized calorimetry triggering data. A single Global DSP collects the lists of objects found by the Local DSP's, and forms an overall result (*e.g.* a cut on the number of electrons found). The system has an average latency of 136 μs.

Processor–based triggering in particle physics has been desirable, because it allows the triggering algorithm to be separated from the architecture of the platform. However, the restricted input data bandwidth of conventional processors makes them unsuitable for this application. We use custom hardware interfaced to the high–speed Comm Ports of the 'C40 chips to achieve an input data bandwidth of 400 MBytes/sec.

I. The DØ Trigger

The DØ Experiment [1, 2] is a high-energy physics experiment at Fermi National Accelerator Laboratory (Fermilab), in Batavia, IL. Fermilab is the world's most powerful particle collider, producing, every 3.5 μs, proton–antiproton collisions ("beam crossings") with a 1.8 TeV center–of–mass energy. The DØ Detector collects signals from these collisions. A subset of the signals collected by the DØ Detector is sent to the DØ Trigger [3], shown in Figure 1. The DØ Trigger, composed of a hardware phase [4] (Levels 1 and 1.5) and a software phase [5] (Level 2), must select "interesting" events for offline analysis. A high-precision data readout, separate from the rapid triggering data readout, is performed on the Detector for these interesting events.

*This work was supported by the National Science Foundation with grant PHY-9319216, and with construction funds from the U.S. Department of Energy and Fermi National Accelerator Laboratory.

Figure 1: The DØ Trigger

A. *The Hardware Trigger*

1) *Level 1:* The Level 1 Trigger consists of Level 1 Trigger Subsystems and the Level 1 Trigger Framework. Each Level 1 Trigger Subsystem produces, for every beam crossing, an indication of the activity seen by its associated Detector component. The major Level 1 Trigger Subsystems are the Level 1 Muon Trigger and the Level 1 Calorimeter Trigger. Their indications are transmitted to the Level 1 Trigger Framework, which determines whether the event is to be digitized for further analysis, or rejected as "uninteresting." This decision must be made between beam crossings (3.5 μs), before the information disappears. The Level 1 Trigger performs, in hardware, fixed algorithms with programmable parameters. It accepts about 1 event out of every 300 beam crossings. The Level 1 Trigger Framework and Level 1 Calorimeter Trigger were designed and built at Michigan State University.

2) *Level 1.5:* The Level 1.5 Trigger produces additional rejection by allowing more complex data processing, at the expense of experiment deadtime. It also consists of both Muon and Calorimeter Trigger Subsystems (again associated with Detector components) and a Trigger Framework. Level 1.5 Trigger Subsystems may take up to 250 μs to produce their triggering information. The Level 1.5 Trigger is used to further qualify a subset of the

events which have been accepted by Level 1. The Level 1.5 Trigger accepts about 1 event out of every 5-10 which it examines. The Level 1.5 Trigger Framework and Level 1.5 Calorimeter Trigger were also designed and built at MSU.

B. *The Software Trigger*

The Level 2 Trigger is a "farm" of 48 MicroVAX processors, each performing detailed analyses, using the high-precision Detector data, of entire events. The Level 2 Trigger makes the final decision to record an event on tape for offline analysis. The Level 2 input rate is about 150 Hz, and its output rate is approximately 3 Hz.

II. Motivation for the Level 1.5 Calorimeter Trigger

The DØ Calorimeter is a 5 m long cylindrical liquid argon and uranium calorimeter centered at one of Fermilab's 6 interaction regions. The trigger signal from the Detector is composed of 1280 projective Trigger Towers, arranged 40 in the polar (η) direction by 32 in the azimuthal (ϕ) direction. Note that the ϕ coordinate "wraps" around the Calorimeter, such that ϕ=32 is adjacent to ϕ=1. Figure 2 shows the DØ Calorimeter "unrolled." Each Trigger Tower is each divided into electromagnetic and total transverse energy (EM E_t and Total E_t) components.

The energy deposited by a single particle travelling through the Calorimeter is frequently shared between neighboring Trigger Towers—the signals from these neighboring Towers must be taken into account to determine the particle's true energy deposit. The Level 1 Calorimeter Trigger cannot do this, thus compromising triggering based on the energy of an individual particle. The Level 1.5 Calorimeter Trigger [6, 7] described here addresses this limitation.

III. Design Goals

Several designs for a Level 1.5 Calorimeter Trigger were considered. All prospective designs were required to meet a set of physics goals. Each design was also measured against a set of implementation goals.

A. *Physics Goals*

1) *Rejection and Efficiency:* Preliminary studies showed that relatively simple algorithms could address the energy sharing problem described in Section II, and provide rejection factors of 3 to 10 with greater than 90% efficiency.

2) *Neighbor Range:* The class of algorithms envisioned all required energy signals from Trigger Towers within plus or minus 2 Trigger Tower indices of the candidate Tower. This is the minimum neighbor range that must be made available to any processing element.

Figure 2: Calorimeter Trigger Coordinate System

3) *Rates and Dead Time:* The input rate to a Level 1.5 Calorimeter Trigger was envisioned to be up to 500 Hz. At this rate, no more than 5% deadtime was desired, implying a processing time goal of 100 μs per event.

B. *Implementation Goals*

1) *Take Advantage of Existing "Hooks":* The Level 1 Calorimeter Trigger was designed with a "hook" to feed the 2500 Trigger Tower energy signals to a more sophisticated Trigger Subsystem, such as the Level 1.5 Calorimeter Trigger described here. All designs were required to be compatible with this "hook."

2) *Implement on a Short Time Scale:* Several hardware-only designs met the above requirements, but would have required a complex and expensive new system that would have taken many years to build. The processor-based solution described here was a faster path to a working system.

3) *Separate Architecture from Algorithm:* In addition, a processor-based design separates the algorithm from the architecture. The resultant system is very flexible, allowing both incremental "tuning" changes to the algorithm, or radically new approaches to triggering. This platform is also open to more advanced filtering which might not be reasonable to implement in hardware.

IV. Reasons for Choosing the TMS320C40 and Hydra-II

A. *Major Challenge of the System*

The major technical challenge in designing a Level 1.5 Calorimeter Trigger is *not* in the application of a particular algorithm. It is rather in making the data from the 2500 Trigger Towers rapidly available to the processing elements. DSP's, much more than conventional CPU's, are optimized for data manipulation.

B. *Features of the 'C40*

Many DSP chips were considered during design studies. The Texas Instruments TMS320C40 [8, 9] proved to be the most suitable processor for the Level 1.5 Calorimeter Trigger. The reasons for this were many-fold.

1) *High I/O Bandwidth:* The TMS320C40 provides high I/O bandwidth via its 6 high-speed Comm Ports and associated DMA Channels. This provides a path for injection of the Trigger Tower data into the Level 1.5 Calorimeter Trigger.

2) *On-chip Boot Loader:* In addition, the 'C40 provides a powerful On-Chip Boot Loader. This greatly simplifies the task of booting the system.

3) *Development and Debugging Environment:* The 'C40 also boasts a very powerful code development and debugging environment. In particular, the JTAG interface allows real-time parallel debugging.

C. *Features of the Ariel Hydra-II (Quad 'C40)*

Several 'C40 boards were also considered. The Ariel Hydra-II [10, 11], a quad 'C40 VME card, emerged as the logical choice.

1) *Access to Comm Ports:* The Hydra-II provides direct external access to 4 of the 6 Comm Ports on each 'C40, and uses the other 2 Comm Ports for a ring connection between the 'C40 chips.

2) *External Control of 'C40:* The Hydra-II also provides external control of each 'C40 via its VME slave interface. This simplifies the connection to the experiment's control system.

3) *Shared Dual Port Memory with VME Access:* The Hydra-II provides 32KB of Dual Port Memory, which is visible from all 4 'C40s as well as via VME. This is crucial for downloading code and parameters to the Level 1.5 Calorimeter Trigger.

4) *VSB Master:* Finally, the Hydra-II provides a VSB master interface. The Global DSP uses this to save its results to a buffer memory for readout.

V. Architecture and Implementation

The Level 1.5 Calorimeter Trigger processing steps are:

1. Collect all Trigger Tower energy data
2. Scan all 1280 Trigger Towers looking for candidates
3. Use an algorithm to confirm or reject these candidates
4. Combine the results and form a decision based on a global analysis

Fundamental to the architecture of the Level 1.5 Calorimeter Trigger is the realization that the Trigger Tower data need not be examined sequentially. The neighbor energy summing process is a *local* process, requiring data from a limited set of Trigger Towers. The Calorimeter is therefore partitioned geographically and serviced by parallel Local DSP's. Specifically, these Local DSP's are responsible for the collection, scanning, and candidate confirmation phases of the Level 1.5 Calorimeter Trigger architecture. They then pass the confirmed objects to a Global DSP, which is responsible for the global analysis processing. A Service CPU oversees the operations of the DSP chips, communicates with the rest of the experiment, and provides error recovery. The architecture of the Level 1.5 Calorimeter Trigger is shown in Figure 3.

A. *Parallel Local DSP's*

There are 11 Local DSP's, each of which receives data from all 256 Trigger Towers in a patch of 32 ϕ by 8 η Towers. Each Local DSP is responsible for finding candidates from the central region of 4 η Towers. The 2 η–Tower–wide patches flanking the central region are provided to meet the neighbor range goal described in Section III. These 8 η–Tower–wide patches overlap, such that each Trigger Tower datum is delivered to 2 Local DSP's—it is in the central candidate region in one Local DSP, and the flanking neighbor region in another. Special cases at the ends of the Calorimeter are eliminated by providing zero–energy data for the nonexistent neighbors.

The Trigger Tower data arrive via 4 Comm Ports on each Local DSP (except for the Local DSP's at the ends of the Calorimeter, which only require 2 Comm Ports for Trigger Tower data input).

Each Local DSP passes its confirmed candidates to the Global DSP via a Comm Port. However, only four of the Local DSP's have direct connections to the Global DSP. The other seven Local DSP's pass their confirmed candidates to one of the directly connected Local DSP's via the ring-connected Comm Ports.

Figure 4 illustrates the arrangement of the DSP's.

Figure 4: Arrangement of DSP's

Figure 3: The Level 1.5 Calorimeter Trigger

B. *Global DSP*

The Global DSP receives the confirmed candidates from all 11 Local DSP's via 4 Comm Ports. The Global DSP can perform a global analysis on the confirmed candidates (*e.g.* determine if one or more electrons have been found), and return that information to the Level 1.5 Trigger Framework via the Service CPU.

C. *Energy Input Data Path*

Each Trigger Tower provides two 8-bit energy signals, EM E_t and Total E_t, to the Level 1.5 Calorimeter Trigger. This data comes from the Level 1 Calorimeter Trigger via the "hook" described in Section III. Eighty new readout cards, called Energy Readout Paddle Boards [12] (ERPB's), based on Xilinx 4002A FPGA's [13], were added to the Level 1 Calorimeter Trigger. Each ERPB is responsible for a 4 ϕ by 4 η segment of the Calorimeter.

Each group of 8 ERPB's, servicing a 32 ϕ by 4 η segment of the Calorimeter, is connected to a Distributor Cap [12] (DC) which sequences ERPB readout. The DC converts the CMOS signals used by the ERPB's into differential ECL, which is used for the long (10 m) cable runs to the Level 1.5 Calorimeter Trigger. There, each group of 5 DC cables is serviced by a Cable Receiver Card (CRC), which converts the ECL signals to the TTL levels used by the 'C40 Comm Ports, and manages the Comm Port communication protocol. The CRC also performs the data duplication referred to in Section I, as well as the separation of EM E_t and Total E_t for delivery to separate Comm Ports.

D. *Control Data Path*

Additional signals, used to initiate Level 1.5 Calorimeter Trigger processing and to transmit the results, arrive and depart via parallel I/O cards, which are controlled by the Service CPU, described below.

E. *Readout Data Path*

The Level 1.5 Calorimeter Trigger records its results in a Data Block. The Data Block is typically on the order of 300 32-bit words, but can be expanded approximately 10-fold for monitoring and debugging purposes. This Data Block is built partially by the Global DSP, and partially by the Service CPU. It is written to one of two Motorola MVME-214 [14] multiport (VME/VSB) memory cards. At all times, one of the two multiport memory cards is available for receiving the current event's Data Block via VSB, while the other has been handed over for readout of the previous event's Data Block via VME. The Data Block is read out by a VME Buffer/Driver [15] (VBD), which sends it to Level 2. Both the VBD and multiport memory cards operate under the control of the Service CPU.

F. *Service CPU*

The Service CPU is responsible for initiating Level 1.5 Calorimeter Trigger processing when required, transmitting the Global DSP's results to the Level 1.5 Trigger Framework at the completion of processing, managing the Data Block readout, and system error recovery. It is a Motorola MVME-135 [14] board, containing a 68020 processor, local memory, and both VME and VSB interfaces.

Note that the Service CPU's functions could all be performed by the Global DSP, if necessary. The inclusion of the Service CPU was prompted by a desire to simplify the 'C40 software, as well as the recognition that much of this software (specifically, the Data Block readout management) could simply be ported from a similar system which had already been developed for the Level 1 Framework and Level 1 Calorimeter Trigger. All of the Service CPU software was written in 68K assembly language.

G. *Split VME Backplane*

The Level 1.5 Calorimeter Trigger is installed in a single, 21-slot VME/VSB crate, shown in Figure 5. The VME backplane is split into two halves, designated Processing Bus and Readout Bus. The Processing Bus links the DSP's, the Service CPU, and the parallel I/O cards together, while the Readout Bus links the multiport memory cards to the VBD. This split backplane allows these two distinct functions (Processing and Readout) to operate without interference. The VSB bus straddles the two backplane halves, and is used to transport the Data Block data from the Processing Bus to the Readout Bus.

H. *Trigger Control Computer*

The Level 1.5 Calorimeter Trigger's executable code and parameters are loaded via the same Trigger Control Computer (TCC) used to manage the Level 1 and Level 1.5 Trigger Frameworks and the Level 1 Calorimeter Trigger. The TCC is a DEC VAXstation 4000 Model 60 [16] with a pVBA [15] adapter. The pVBA allows the TCC to appear as a VME master on the Processing Bus.

VI. 'C40 SOFTWARE

A major component of the Level 1.5 Calorimeter Trigger is the software which runs on the 'C40s. Although some of this software has been described above, it is useful to separately describe the software as a whole.

The 'C40 software is divided into two classes—Frame and Tool—which are described below. All 'C40 software is written in assembly language.

A. *DSP Tool*

The Tool code implements the chosen algorithm, and confirms or rejects candidates submitted by the Frame. Although the Tool provides the physics functionality required from the Level 1.5 Calorimeter Trigger, it accounts for a small fraction of the Level 1.5 Calorimeter Trigger software. The Tool is kept distinct from the Frame code, so that it can be easily exchanged with Tool code that implements a different algorithm. The distinction is also made between Local Tool code and Global Tool Code. Identical Local Tool code executes on the 11 Local DSP's.

The current Tool is designed to find isolated electrons, and was selected by performing Monte Carlo analyses of various Tool options.

The current Local Tool algorithm:

- Receives a candidate from the Frame
- Calculates the 1x2 EM E_t Sum of this candidate Tower plus the most energetic of its 4 neighbors
- Calculates the 3x3 Total E_t Sum centered around this candidate Tower
- Confirms this candidate as an electron if both
  1. 1x2 EM E_t Sum $\geq$ 1x2 EM E_t Threshold
  2. 1x2 EM E_t Sum / 3x3 Total E_t Sum $\geq$ Ratio Threshold

The current Global Tool algorithm:

- Confirms the event if the number of confirmed electrons is $\geq$ Electron Count Threshold

This processing is not traditional digital signal processing, but the powerful data access features of the 'C40 are still crucial for the algorithm. As an example, the circular addressing mode of the 'C40 is used to mirror the cylindrical, ϕ-wrap nature of the Calorimeter.

Figure 5: Level 1.5 Calorimeter Trigger VME Crate

B. *DSP Frame*

The Frame includes everything that is not the algorithm. Again the distinction is made between Local Frame and Global Frame.

The Local Frame:

- Receives energy data from the Level 1 Calorimeter Trigger
- Scans energy data to find candidates
- Calls the Local Tool for each candidate
- Sends the confirmed candidates to the Global DSP
- Handshakes with the Service CPU
- Performs initialization tasks including receiving and checking operating parameters

The Global Frame:

- Receives confirmed candidates from the Local DSP's
- Calls the Global Tool for each confirmed candidate
- Handshakes with the Service CPU (including returning the results of the Tool)
- Builds the DSP portion of the Data Block
- Performs initialization tasks including receiving and checking operating parameters

The Frame is by far the majority of the DSP software. Again, it does not perform traditional DSP operations, but uses the 'C40 architecture to efficiently perform its required tasks. For example, all inter-DSP communication, as well as the energy data reception, occurs via DMA over Comm Ports. These DMA's are dependent on the 'C40's linked-list DMA features.

Each Local Frame is slightly different from the other 10 Local Frames. For example, each Local node uses different Comm Ports for energy data input, services different geographical regions of the Calorimeter, and appears as a different link in the Local-to-Global transfer chain described above. Minimizing these differences, or, maximizing module re-usability, was a major development challenge. The chosen solution was to have all node-dependent information stored in node-specific initialization modules (with one such module for each Local DSP). All other modules were written in a node-independent fashion. Each Local executable, then, is built primarily of common source modules, with one node-specific module linked into each executable. This approach both reduced the complexity of the system, and reduced the amount of redundant code to be developed.

The node-specific initialization modules make extensive use of both macros and include files, again minimizing the possibility of simple "typo" bugs, and simplifying both maintenance and functional modifications.

VII. Timing Performance

The most critical aspect of the Level 1.5 Calorimeter Trigger's timing performance is its response time for sending an answer back to the Level 1.5 Trigger Framework. It is this time that contributes to the experiment deadtime described in Section I. Less critical is the time required to prepare the Data Block for events which are to be read out. The Level 1.5 Calorimeter Trigger therefore returns its answer to the Level 1.5 Trigger Framework as quickly as possible, even if further processing is needed before the system can be ready for the next event.

Recognizing that the goal of the Level 1.5 Calorimeter Trigger is event rejection, the processing has been optimized for maximum speed at this task. The time needed for additional processing (after the answer has been delivered) will be longer for events which are accepted than events which were rejected. But, with an input rate below 500 Hz, on average 2 ms is available between sending the answer to the Level 1.5 Trigger Framework and the next processing request.

The Service CPU also is designed to make best use of its time. It typically separates the initiation of an action from checking for its successful completion. Between events, the Service CPU prepares for the upcoming event by completing as many "housekeeping" tasks as time permits (*e.g.* preparing a readout buffer for the upcoming event).

The flow of a typical event through the system is detailed in Table 1.

VIII. Conclusions

The Level 1.5 Calorimeter Trigger has been used at DØ since August, 1994. The system's operation during this time has been compared to the original design goals.

A. *Physics*

1) *Meets Rejection Goals:* The Level 1.5 Calorimeter has met the rejection goals of 3-10 with the required 90% efficiency.

2) *Meets Rate Goals:* Although the Level 1.5 Calorimeter Trigger's processing time of 136 μs is longer than the desired 100 μs, the input rate is such that the total deadtime remains within acceptable levels. In Section C, several methods of reducing the latency are discussed.

3) *Interest from Other Experiments:* A trigger very similar to the Level 1.5 Calorimeter Trigger has been independently proposed for the ATLAS experiment at CERN, in Geneva, Switzerland. Preliminary results from the Level 1.5 Calorimeter Trigger were presented to ATLAS in May 1994.

B. *Technical*

1) *No Down Time:* It is crucial that the hardware trigger at DØ operate with high reliability. The hardware trigger is at the heart of the experimental data taking process—hardware trigger failures completely halt the operation of DØ. The Level 1.5 Calorimeter Trigger has caused no downtime due to hardware or software failures during the 2.5 months that it has been operating at DØ.

2) *Implementation Time Scale:* Approximately 1 calendar year, and 3 man-years, were required for the design and implementation of the Level 1.5 Calorimeter Trigger. This is considerably shorter than a hardware-only trigger, composed of custom circuit boards, would have required. Only 5 types of printed circuit boards were designed specifically for this project (3 of which had no active components, and 1 of which has become a commercial product offered by Ariel).

3) *Architecture and Algorithm Separated:* The successful separation of architecture and algorithm has been achieved. During initial testing of the Level 1.5 Calorimeter Trigger, 3 generations of algorithms were used. In each case, replacing the algorithm was been simple and straightforward. This clearly could not have been done with a hardware-only trigger, in which the algorithm is embedded in the architecture.

C. *Upgrade Paths*

Two classes of upgrades have been considered: speed and functionality increases. Upgrade paths exist for both classes.

1) *Increasing the Speed:* An examination of Table 1 shows that the single largest consumer of processing time is the Local-to-Global data transfer. This is due to two factors:

- The relatively large amount of data (25 longwords) sent by each Local DSP
- The limited bandwidth into the Global DSP (the Global DSP only uses 4 Comm Ports to service 11 Local DSP's)

Reducing the amount of data sent by each Local DSP would be straightforward. To simplify the transfer, each Local DSP transfers a fixed amount of data to the Global DSP. This list is long enough to describe 8 confirmed candidates. Either a variable-length list (exactly long enough for the number of confirmed candidates), or a shorter fixed-length list could be sent instead. Increasing the Global DSP's input bandwidth could be accomplished by building a new interface between the Local DSP's and the Global DSP, bypassing the Global DSP's Comm Ports entirely.

Level 1.5 Calorimeter Trigger Activity	Step Time (μs)	Elapsed Time with One Tool Call (μs)
Processing request for event n	0	0
All energy data arrives at Local DSP's	18.8	18.8
Local DSP candidate scan (with no Tool calls)	30.7	49.5
Each Local Tool call	6.6	56.1
Local DSP's transfer confirmed candidates to Global DSP	56	112.1
Global DSP confirmed candidate scan	5.5	117.6
Each Global Tool call	0.5	118.1
Answers presented to Level 1.5 Trigger Framework	9.4	127.5
Experiment deadtime ends	8.5	136
Data Block for event n built	252	388

Table 1: Level 1.5 Calorimeter Trigger Timing

Also, since the Level 1.5 Calorimeter Trigger is based on commercial hardware, it can benefit from industry trends. DSP chips are clearly a very active sector of the electronics and computer industry. For example, 60 and 80 MHz versions of the 'C40 have already been announced, and allow very simple upgrade paths. The TMS320C80 [17], with 4 DSP's and a general-purpose processor embedded on the same chip, also provides a potential upgrade path.

2) *Increasing the Functionality:* The Level 1.5 Calorimeter Trigger is now used as an electron filter, but offers many other possibilities. The separation of architecture and algorithm allows both improvements in the field of electron filtering and completely new and innovative algorithms.

More requirements could be added to the Global Tool, which currently only counts electrons. In general, the Global Tool could look for correlations between candidates confirmed by the Local Tool. This type of triggering may offer attractive benefits for DØ.

The Level 1.5 Calorimeter Trigger could also be duplicated to run several crates in parallel. Multiple crates could operate in parallel but independently, on either the same or different events, while applying different algorithms.

More ambitious upgrades could involve importing and correlating information from other DØ Detector components.

References

[1] M. Abolins, et. al, *Design Report: The DØ Experiment at the Fermilab Antiproton–Proton Collider*, Fermilab Publication, 1984.

[2] S. Abachi, et. al., "The DØ Detector," *Nucl. Instrum. and Meth.*, Vol A338, pp. 185-253, 1994.

[3] M. Abolins, D. Edmunds, P. Laurens, J. Linnemann, and B. Pi, "A High Luminosity Trigger Design for the Tevatron Collider Experiment in DØ," *IEEE Trans. on Nucl. Sci.*, Vol. 36, pp. 384-389.

[4] M. Abolins, D. Edmunds, P. Laurens, and B. Pi, "The Fast Trigger for the DØ Experiment," *Nucl. Instrum. and Meth.*, Vol. A289, pp. 543-560, 1990.

[5] J. Linnemann, "The DØ Software Trigger," in *Proc. of the Int. Conf. on Computing in High Energy Physics*, Annency, France, Sept. 1992, pp. 199-201.

[6] J. Linnemann, et. al, "The DØ Level 1.5 Calorimeter Trigger," in *Proc. of the Int. Conf. on Computing in High Energy Physics*, San Francisco, CA, May 1994.

[7] D. Edmunds, S. Gross, P. Laurens, "A DSP Based Calorimeter Trigger for the DØ Experiment," in *IEEE Nucl. Sci. Symp.*, Norfolk, VA, Nov. 1994.

[8] Texas Instruments Inc., Dallas TX.

[9] Texas Instruments Inc., *TMS320C4x User's Guide*, Texas Instruments Inc., 1993.

[10] Ariel Corp., Highland Park NJ.

[11] Ariel Corp., *Preliminary User's Manual for the Hydra-II*, Ariel Corp., 1994.

[12] University of California at Irvine, Irvine, CA.

[13] XILINX Inc., San Jose CA.

[14] Motorola Inc., Phoenix AZ.

[15] Zeller Research Labs Ltd., Bristol RI.

[16] Digital Equipment Corp., Maynard MA.

[17] Texas Instruments Inc., *TMS320C80 MVP Technical Brief*, Texas Instruments Inc., 1994.