Investigation of a VLSI Neural Network Chip as Part of a Secondary Vertex Trigger

B. Denby
Fermi National Accelerator Laboratory, Batavia, Illinois

Th. Lindblad, C.S. Lindsey
Manne Siegbahn Institute, Stockholm, Sweden

Geza Szekely, J. Molnar
Institute of Nuclear Research, Debrecen, Hungary

Åge Eide
Ostfold College, Halden, Norway

S.R. Amendolia, A. Spaziani†
University of Sassari and INFN Pisa, Italy

May 25, 1993

Abstract

An analog VLSI neural network chip (ETANN) has been trained to detect secondary vertices in simulated data for a fixed target heavy flavour production experiment. The detector response and associative memory track finding were modelled by a simulation, but the vertex detection was performed in hardware by the neural network chip and requires only a few microseconds per event. The chip correctly tags 30% of the heavy flavour events while rejecting 99% of the background, and is thus well adapted for secondary vertex triggering applications. A general purpose VME module for interfacing the ETANN to experiments, equipped with ADC/DAC circuits and a 68070 CPU, is also presented.

---

†currently at ENEA, Roma.

1 Introduction

The study of particles containing heavy flavour quarks continues to be a major topic in High Energy Physics (HEP), both at fixed target and collider experiments. The high resolution silicon microvertex detectors with which many of today's experiments are equipped are powerful tools for offline validation and reconstruction of heavy flavour events. Heavy flavour events however are very rare compared to background processes; the signal to background ratio ranges from 1:1000 (hadron colliders) to 1:50000 (fixed target experiments).

The signal to background ratio for the events written to tape can be significantly improved by implementing triggers which select certain subsets of all heavy quark decays, for example by looking for the high transverse momentum lepton from a semileptonic decay or the J/ψ emitted in some of the B meson decays. These triggers have low inherent efficiency since they are sensitive only to a fraction of all heavy flavour decays; furthermore, high transverse momentum leptons and J/ψ's can emerge from other sources as well. The ideal trigger for heavy flavour decays would be sensitive to the presence in the event of secondary decay vertices as detected in the silicon microstrip detector, since it would then have high efficiency for *all* types of heavy flavour decays and excellent rejection of background.

Reconstruction of secondary vertices is a challenging problem even in the offline analysis; to attempt it in the trigger is even more difficult because the very short time available and the large number of channels (several thousand), which preclude preprocessing and necessitate special trigger hardware. Nevertheless a number of projects are underway to install secondary vertex triggers in working experiments. In [1], involving the WA92 experiment at CERN, signals from the microvertex detector which have been preprocessed by a contiguity processor [2] will be examined, along with other event information, by two types of artificial neural network chips operating in parallel (the 80170NX or ETANN by Intel [3]) and the MA16 of Siemens [4]). In [5], which describes a trigger being prepared for the CDF proton antiproton collider experiment at the Fermilab Tevatron, hit information from the silicon detetector is passed to the so-called *associative memory* chips [6] which generate a track list. Digital signal processors (DSP's) will then be used to detect secondary vertices. Associative memory chips will also be used in a secondary vertex trigger for the upgrade of the E771 experiment at Fermilab [7].

The present work reports on a hardware study of the Intel ETANN chip

for trigger level secondary vertex detection in a fixed target experiment. In an earlier work [8], neural network algorithms were investigated for finding secondary vertices in such an experiment [9], using as input the track list found by a set of associative memory chips. As in that work, the present results are based upon events generated by Monte Carlo [10], passed through a detailed detector simulation [11], and finally through a simulation of the associative memory chips [12]. In [8] preprocessing was performed on the associative memory track list before it was passed to the (software) neural network. In the present work an improved associative memory simulation is used, and the track list is converted to analog information and passed directly to the ETANN for vertex detection, without preprocessing.

2 Experiment Simulation

We simulated here a fixed-target experiment (NAxx at CERN) which was proposed to investigate triggering on B decays. It is representative of many of the experiments using silicon vertex detectors to search for decays of long-lived heavy flavour particles. A proton beam of energy 450 Gev is directed onto a 1 mm copper target (see figure 1). Particles emerging from the interaction are detected in a set of silicon microstrip detectors downstream of the target, which measure the intercepts of the tracks at each plane in each of two perpendicular coordinates, x and y. Signals on the microstrips are amplified and passed via cables to the data acquisition system, where digitization takes place, and where the hit positions are calculated. The hit positions for each plane are then passed serially to the associative memory chips, which can be thought of simply as look-up tables of all possible tracks in the system (see section 3). As the hit information passes through the associative memories, the track list is generated; when all hit information has passed through, the track list is complete (figure 2).

The tracks are parametrized by the impact parameter D which measures the distance of closest approach of the track to the origin, and Φ, which is the angle of the track with respect to the beam direction. Note that the vertical displacement of the beam is not constant. In the general case, tracks emanating from a common vertex should lie upon sinusoids in this space; however, since the decay angles are small, tracks from a common vertex will in fact lie upon a line whose slope is proportional to the distance of the vertex from the primary vertex. In the ideal case, then, the primary vertex will appear as a set of points on a horizontal line, and secondary vertices as a

set of points on sloped lines intersecting the primary line. In practice, finite acceptance, noise, interactions within the silicon, spurious tracks, etc., can significantly alter the picture. A few typical signal and background events are shown in figure 3.

3 The Associative Memory Readout

The Associative Memory is functionally similar to a Content Addressable Memory which stores all possible hit configurations for legal tracks, e.g., strip 105 in layer one, strip 115 in layer two, strip 125 in layer three, etc. Presentation of a valid hit configuration outputs an address in RAM which contains the slope and intercept of the track. The pattern presented need not match exactly the template stored in the memory; a programmable number of missing hits can be tolerated. Each chip stores patterns corresponding to 128 different tracks; many chips are ganged together to store the many thousands of patterns corresponding to all possible tracks.

The actual operation of the Associative Memory is in fact quite different from that of a CAM, in that the patterns are presented 'on the fly', as shown in figure 2. The columns in the figure correspond to silicon layers and the rows to stored hit patterns. After an event is registered, hit addresses from each layer pass sequentially down the columns, and if the address corresponds to one contained in a particular track, a latch is set at that memory location. When all of the latches in a given row (or an acceptable subset in the case of missing hits) are set, then the corresponding track is 'found' and the RAM address for that track is sent out on the bus. Thus, after all the data from the layers has passed through the memory, all of the addresses of found tracks will have been sent out on the bus.

The Associative Memories (in the present version) can be clocked at about 20 MHz, so that an event with 20 tracks (i.e. approximately 20 hits per detector plane) can be handled in a little more than a microsecond.

The operation of the AM chips was simulated with a FORTRAN routine which modelled varous instrumental effects such as inefficiencies in track finding (about 16 %) and generation of spurious tracks (of the order of 2%) [12].

4 Generation of Data

Data were generated with the PYTHIA Monte Carlo Program [10] assuming a proton on proton collision. The interaction with the nucleus and consequent particle production was treated in a special generation routine, added to the main $p - p$ interaction routine [12]. Events were generated without heavy flavour production (background sample, only QCD minimum bias events), and with heavy flavours (signal sample, only $b\bar{b}$ events). The incoming beam energy was 450 GeV and a 1 mm copper target was assumed (figure 1).

The detector consisted of 10 planes of silicon microstrip telescope, each plane 300 μm thick and 5 by 5 cm square, the first plane at 4 cm from the target, the last plane at 13 cm. There were 4 planes with X-oriented strips, 4 with Y-oriented strips, 1 U and 1 V stereo planes. In the present work, the X and Y patterns were treated separately, and the stereo planes were not used.

Tracks were straight in the detector region (no magnetic field). The particles were traced in the detector using the GEANT code [11]: all the possible secondary decays, interaction with the detector, gamma conversions, delta rays etc. were "switched on" in the simulation. The operation of the silicon detectors allowed for charge spread onto strips neighboring the hit strip. To reduce the number of hits entering the Associative Memories, a barycenter algorithm was applied. Such an algorithm, to achieve a higher accuracy, can output strip coordinates which correspond to non-integral strip numbers. This effectively doubled the number of possible strip addresses to consider per plane, thus quadrupling the number of patterns which must be considered for the track finding. The algorithm is applied to purely digital information (no pulse height was used). In the present work it was part of the FORTRAN Associative Memory simulation, but it can be implemented in real time in a real experiment using only simple electronics.

5 Training the Network

The neural network was implemented in the ETANN VLSI chip[3]. The use of this chip for detector pattern recognition tasks has been described previously [13] [14]. Since the methods of emulating, training and testing the chip for this test are similar, we will only briefly review the steps here and refer the reader to those works for more details.

A single ETANN can implement a network with up to 64 inputs, 64 hidden units and 64 outputs. Here the inputs to the network are the D and Φ values for each track in both planes. The 64 inputs were divided into 4 sections of 16 inputs each. The first 16 inputs were used for the D values of up to 16 tracks in the xz plane. The second 16 inputs were for the Φ values for those tracks. Similarly, the third and fourth sections were used for the D and Φ values of up to 16 tracks in the yz plane. (The tracks in the two planes are not correlated by the associative memory so the number of tracks found in the two layers can be different due to different solid angles, efficiencies, etc.) The D and Φ values were normalized to a -1 to 1 scale for the software emulation, corresponding to 0.0v to 3.0v for the actual chip inputs.

The network had 64 hidden units and 8 output units. The first four outputs should each be equal to 1 for background events, -1 for signal events. Similarly, outputs 5-8 should be complementary to the first four outputs: -1 for background and 1 for signal. In an experimental setup, each of the four outputs could be fed into a summing junction to obtain a single output value. Although each output in a set of four is trained to the same target value, there can be slight variations in the outputs due to variations in the performance of the analog components. As described in reference [14], summing of multiple outputs can average out some of these variations.

With the event generation and detector simulation described in section 4, there were 2500 background (BG) events produced and 500 signal (SIG) events. To increase the statistics, both samples were doubled taking these events and reversing the D and Φ values in the xz plane. This is allowed by the symmetry of the problem. The data was divided into 4000 BG events + 800 SIG events for training, and 1000 BG events + 200 SIG events for testing. The signal events were distributed uniformly throughout the samples.

The Intel iNNTS PC-based system [15] controls the basic communication functions between the PC and the chip via an extender board connected by a cable to an external trainer box. In addition, an ETANN simulation program [16] was used for both the software training and, in combination with the iNNTS software, the chip-in-the-loop training. With this sytem a table of input values *versus* neuron output was obtained directly from one of the chip neurons and used in the software for the neuron transfer function. The ETANN transfer functions are not exact sigmoid functions as normally used in neural networks simulations so it is important to use the hardware function in the simulation. (see reference [14] for a discussion of the ETANN transfer function.) The ETANN weights are limited to ± 2.5; this restriction

was also held in software.

The training in software was done in two steps. First, the net was trained on only events having between 3 and 8 tracks in each plane. Second, after the network performance for this subset was deemed satisfactory, it was trained on the full sample. There were about 1 million training back-propagation iterations for each step.

After the software training, the weights from the simulated network were downloaded into the chip. Because of imperfections in such an analog system, the chip will not immediately perform exactly as the simulation. To reach suitable performance levels, it is necessary to do *chip-in-the-loop* training as well. Patterns are presented to the chip and outputs compared to the target values. In software, the necessary weight adjustments are calculated and then downloaded to the chip. Starting from a net downloaded from a simulation greatly reduces the time needed for the chip to train as compared to starting from random weights. For this case the chip approached simulation performance levels after about 12000 iterations.

6 Results

After training, both software and hardware networks were tested using a data set independent of the training set. The efficiency for correct identification of the target patterns is measured with a *strict* and a *limited* criteria. The strict criteria demands that both the BG outputs and the SIG outputs agree with the target. That is, the event is declared background if the BG outputs are > 0.0 and the SIG outputs are < 0.0, vice versa for signal events. If both output sections are above or below zero, then the result is labeled ambiguous. The limited criteria only looks at the SIG outputs. If the summed output is above some cut, then the event is labeled a signal event. If the output is below the cut then the event is labeled a background event.

Table 1 shows the software and chip network performance on the test data using the strict criteria. The simulation and the chip perform fairly closely, although the simulation does somewhat better than the chip for signal identification.

Figure 4 shows result for the chip and the simulation according to the limited criteria for correct identification as a function of the cut on the output. For a tight cut on the output, the efficiency is about 99% for background identification and about 30% for signal. A loose cut on the output

Strict Cut Results

Simulation	Good	Bad	Ambiguous
Signal	57.0%	37.0%	6.0%
Background	91.8%	6.4%	1.8%
ETANN	Good	Bad	Ambiguous
Signal	47.5%	47.5%	5.0%
Background	91.7%	6.4%	1.9%

Table 1: Percentages of signal and background events which are identified correctly (Good), incorrectly (Bad) or which are ambiguous, for neural net executed in software (Simulation) and hardware (ETANN).

gives about 90% background and 50% signal efficiency. The simulation and the chip are in rough agreement for the background cuts but the software does better on the signal acceptance by about 10%.

7 VME Interface Board

With its large number of neurons and synapses and its fast processing, the ETANN was judged the only suitable chip now available for this application; however, there is the complication of having digital signals from the associative memory as inputs to the analog chip. A VME board has been built which can be used to interface the ETANN to a variety of environments. The board can accept either analog signals, received directly through front panel inputs, or digital signals, which are used to set DAC's (one DAC per input signal) connected to the ETANN inputs. The analog outputs of the ETANN are also available on the front panel, either directly or after summing and discrimination, or they can be digitized by the fast ADC's provided for all 64 outputs. The DAC's have a 4 μs settling time and the ADC's a 5 μs digitization time. These overheads increase the total processing time.

The board was built with an ELTEC SAC-711 VME card [17], which employes a 68070 CPU, 2 MB of dual-ported memory, I/O-channels, etc., as well as a prototyping section connected to a proprietary bus. This card was previously used for a purely analog input implementation [14]. A block diagram of the present solution is shown in fig. 5 and a photograph of the VME module is shown in fig.6. A detailed description of this VME-board,

as well as its performace will be given in ref. [18].

Software was developed in assembler and downloaded to the 68070/80170 VME card in S-format. This code is used to set the ETANN output gain voltage and the reference voltages to the values used during the CIL training. It will then wait for a flag in the common SRAM, set a busy flag and read 64 bytes from the SRAM into the ETANN inputs. Controlling the ETANN time-sequences, the code will then read the ETANN outputs and copy them, following the digitization by the AD7228 ADC chips [19], to the SRAM.

To test the functionality of the 68070/80170 VME-module, C code was developed in a VME-based PC/AT residing in slot 1 of the crate. This code can present patterns to the chip and display the desired and actual outputs on the screen.

A chip is first trained and tested with the iNNTS sytem, then installed in the VME board (which has no weight setting capabilities) and used in the feed-forward mode. With the above C code, test patterns can be presented to the chip to measure its performance. It is important that the control voltages be the same within a few tens of mV to the values in the trainer. Following this procedure we obtained performance for the chip in the VME board similar to those in the trainer.

8 Conclusions and Discussion

We have shown that an analog VLSI neural network can be trained to separate events with secondary vertices from background events with good efficiency, using as input a list of the impact parameters and angles of the tracks as determined by an associative memory technique. The best background rejection was obtained by using the *limited* cut procedure, i.e., simply requiring the 'signal' output units to be above a threshold. The performance obtained, 99% background rejection, 30% signal efficiency, is comparable to that obtained by real heavy flavour experiments in their offline analyses, and would provide a high enrichment of the data with heavy quark events if such a system were implemented in the trigger.

Although the associative memory processing was done in a simulation, the total processing time through the entire system, including associative memories, settling of input DAC's, and processing by the ETANN should be no more than several microseconds. This is adequate for many triggering applications (faster analog processing times with other chips may be possible [20]). Although the chip is analog and the inputs from the associative

memory are digital, we have shown that a compact VME board with the required D-A and A-D conversions can be built that performs to the necessary processing accuracy and speed.

References

[1] R. Odorico,*Heavy Flavor Discrimination by Neural Networks*, invited talk at IEEE 1992 Nuc. Sci. Sym., Orlando. Submitted to *Trans. Nucl. Sci.*

[2] G. Darbo and L. Rossi, *Nucl. Inst. and Meth.* **A289**(1990)584-591.

[3] 80170NX Electrically Trainable Analog Neural Network, Data Booklet. Intel Corp. 2250 Mission College Boulevard, Santa Clara, Ca. 95052-8125.

[4] U. Ramacher, J. Beichter, W. Raab, J. Anlauf, N. Bruls, U. Hachmann and M. Wesseling, *Design of a 1st Generation Neurocomputer* in "VLSI Design of Neural Networks", U. Ramacher and U. Ruckert (eds), Kluwer Acad. Pub. 1991, pp 271-310

[5] G. Punzi and L. Ristori, Results of SVT simulation with real SVX data from the current run, CDF/DOC/TRIGGER/PUBLIC/2037, CDF Collaboration internal note, unpublished, April 7, 1993.

[6] S.R. Amendolia et al., The AMchip: a VLSI associative memory for track finding, *Nucl. Inst. Meth.* **A315** 446-448.

[7] Fermilab Proposal P867, Appendix D, unpublished.

[8] L. Gupta et al., Neural network trigger algorithms for heavy quark event selection in a fixed target high energy physics experiment, *Pattern Recognition* **25** (1992) 385-400.

[9] S.R. Amendolia et al., Research and development on fast trigger systems for heavy flavour triggering in fixed target and collider experiments, Proposal to SPSC, CERN/SPSC 89-5, 13 January 1989.

[10] T. Sjöstrand and M. van Wijl, The PYTHIA Monte Carlo. *Phys. Rev.* **D**36(1987)2019.

[11] GEANT3 User's Guide, CERN DD/EE/84-1(1985).

[12] A. Spaziani, private communication.

[13] C.S. Lindsey et al., *Nucl. Instr. and Meth.* **A317**(1992)346-356.

[14] Tommy Akkila, Tom Francke, Thomas Lindblad and Åge Eide, *Nucl. Instr. and Meth.* **A327**(1993)556-572.

[15] *iNNTS Neural Network Training System User's Guide*, Intel Corp. 1992.

[16] NeuroDynamX, Inc. 319 B Raymondale, South Pasadena, Ca. 91030.

[17] ELTEC Elektronik, GMbH, Mainz, Germany, SAC-700/800 User's Manual

[18] J. Molnar, G. Szekely, Th. Lindblad, C. Lindsey, B. Denby, S.R. Amendolia, and Åge Eide, to be published.

[19] Analog Devices, Norwood, MA.

[20] J.R. Hansen, *Proceedings of the Second International Conference on Software Engineering, Artificial Intelligence, and Expert Systems for Nuclear and High Energy Physics*, La Londe-les-Maures, France, January 1992, World Scientific, to appear.

Figure 1. NAxx experiment layout. See section 2 for a description.

Figure 2. Operation of the Associative Memories. See section 3.

Figure 3. Two signal and two background events in D-Phi space, and net the histogram representation of an event as input to the ETANN.

Figure 4. Identification efficiency versus cut on the output (fraction of output full scale.)

Figure 5. Block diagram of the hardware implementation on a VME prototyping card. The main structure is built around the local bus with the dual-ported SRAM as an essential part. A,D and C stands for address, data and control, respectively and DEC refers to decoding. Signals corresponding to 'signal' and 'background' are available following the discriminators (DISCR).

Figure 6. Photograph of the VME module sketched in fig. 5. Eight 8-fold DAC's are mounted below the ETANN piggy-back card.

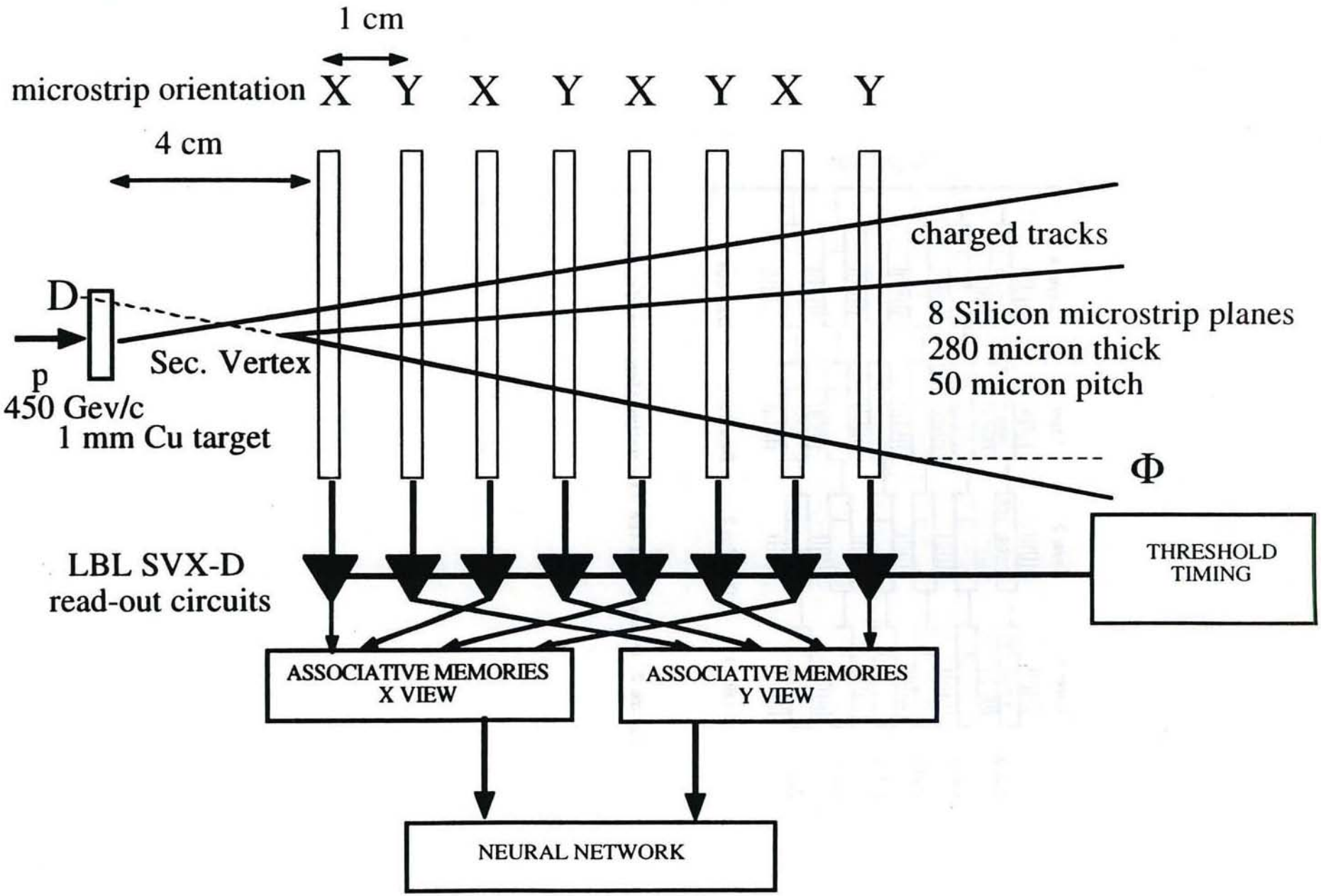


Figure 1. NAxx experiment layout. See section 2 for a description.

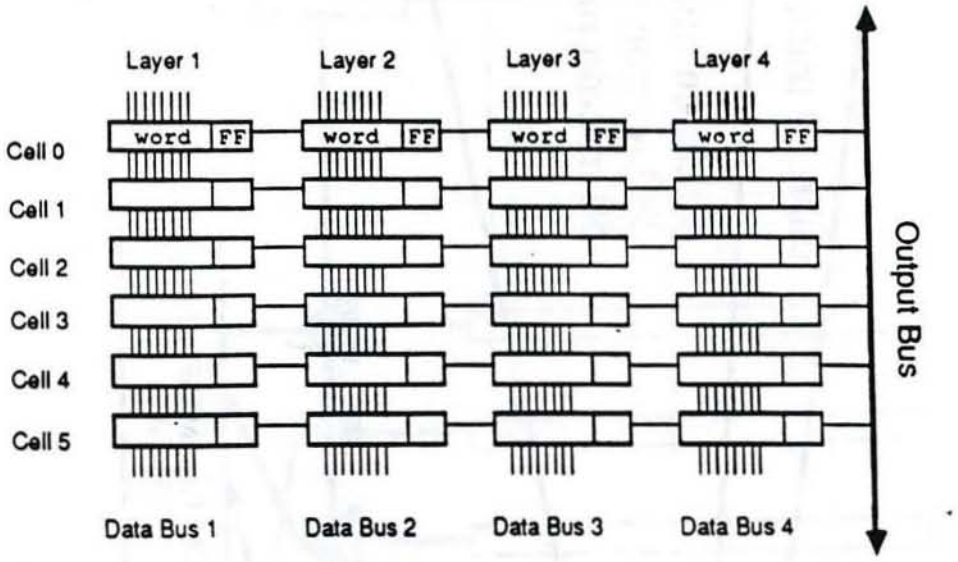


Figure 2. Operation of the Associative Memories. See section 3.

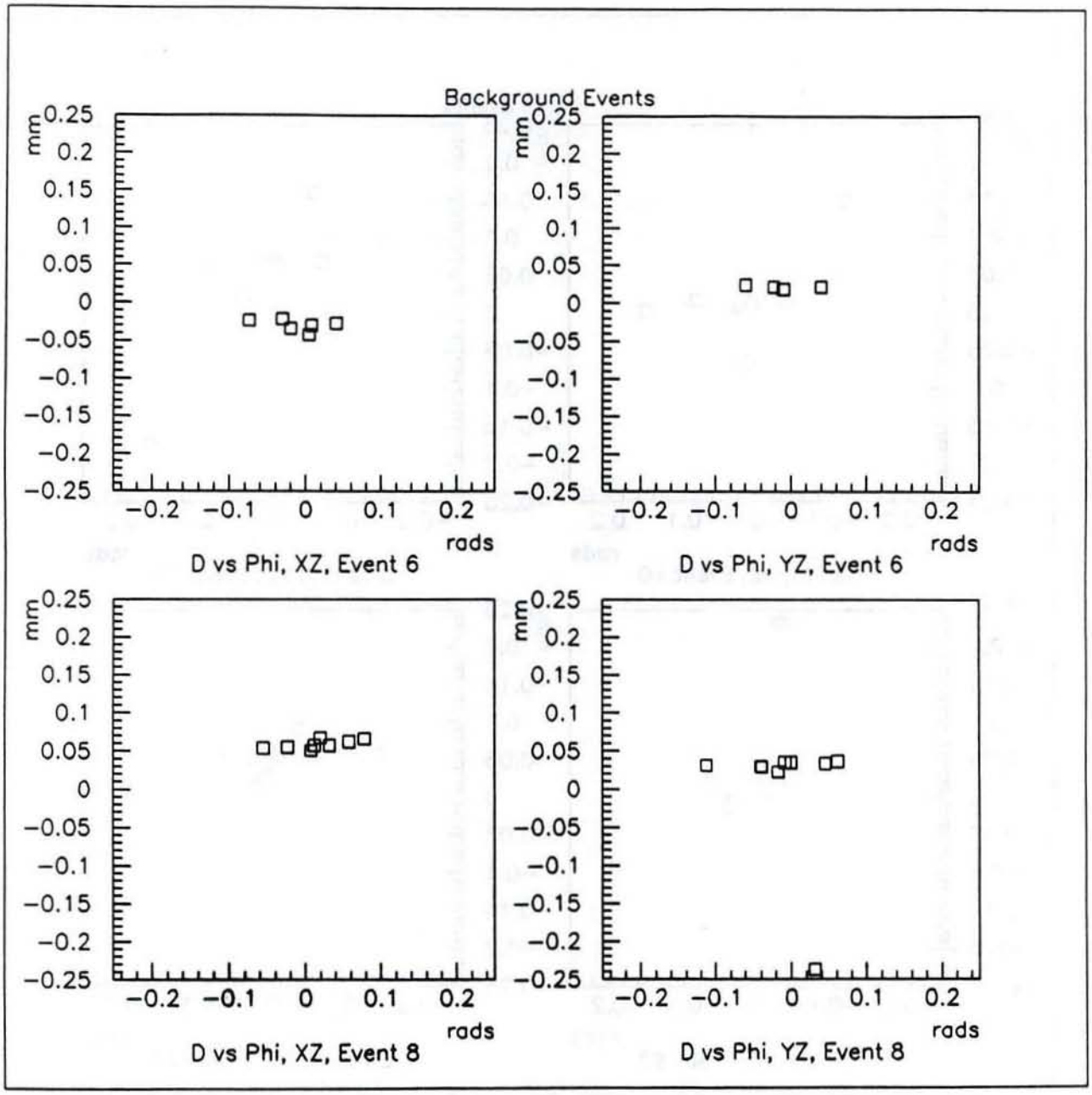
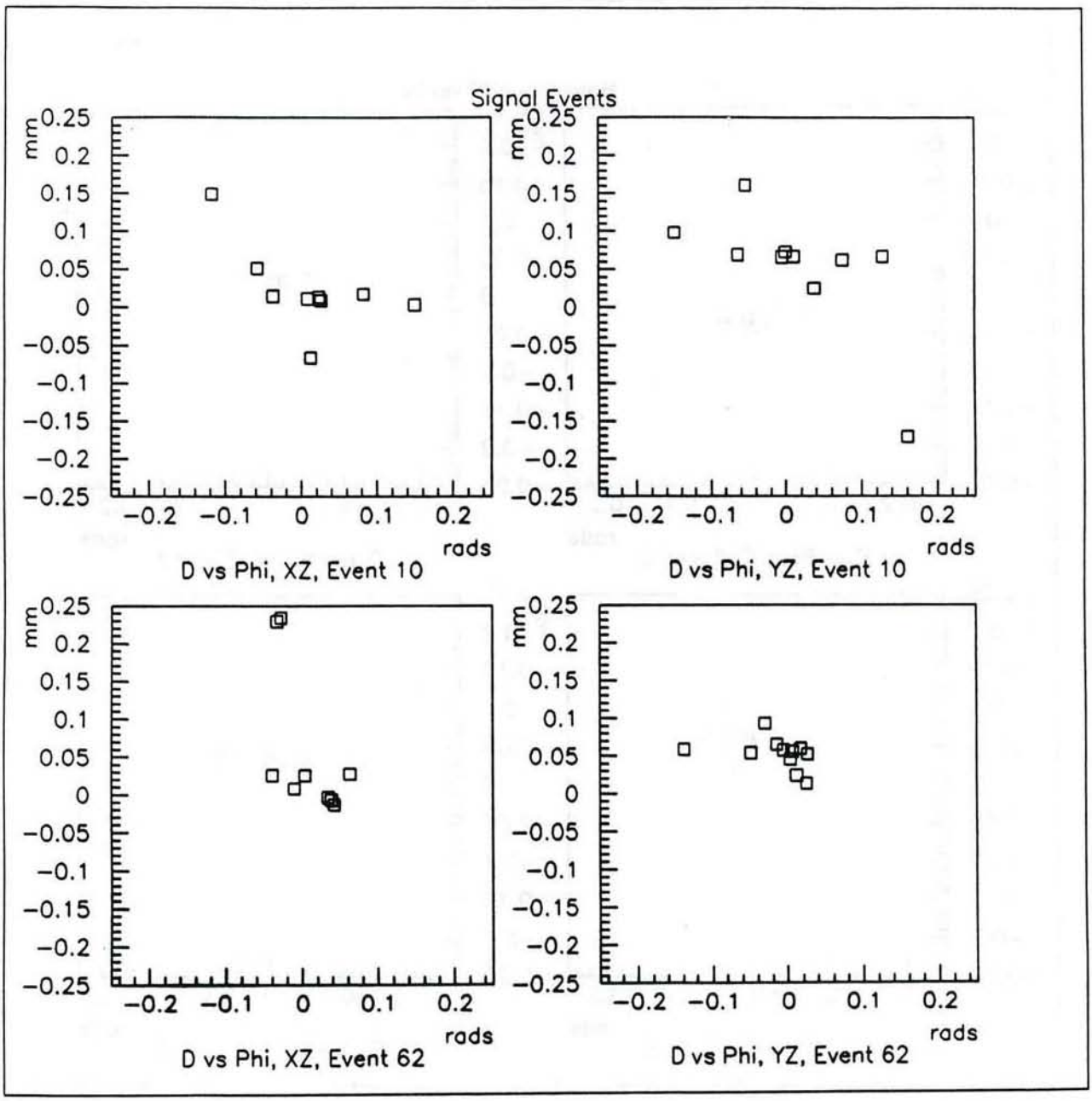


Figure 3. Two signal and two background events in D-Phi space, and net the histogram representation of an event as input to the ETANN.


Signal Events
mm
0.25
0.2
0.15
0.1
0.05
0
−0.05
−0.1
−0.15
−0.2
−0.25
−0.2 −0.1 0 0.1 0.2
rads
D vs Phi, XZ, Event 10
D vs Phi, YZ, Event 10
D vs Phi, XZ, Event 62
D vs Phi, YZ, Event 62

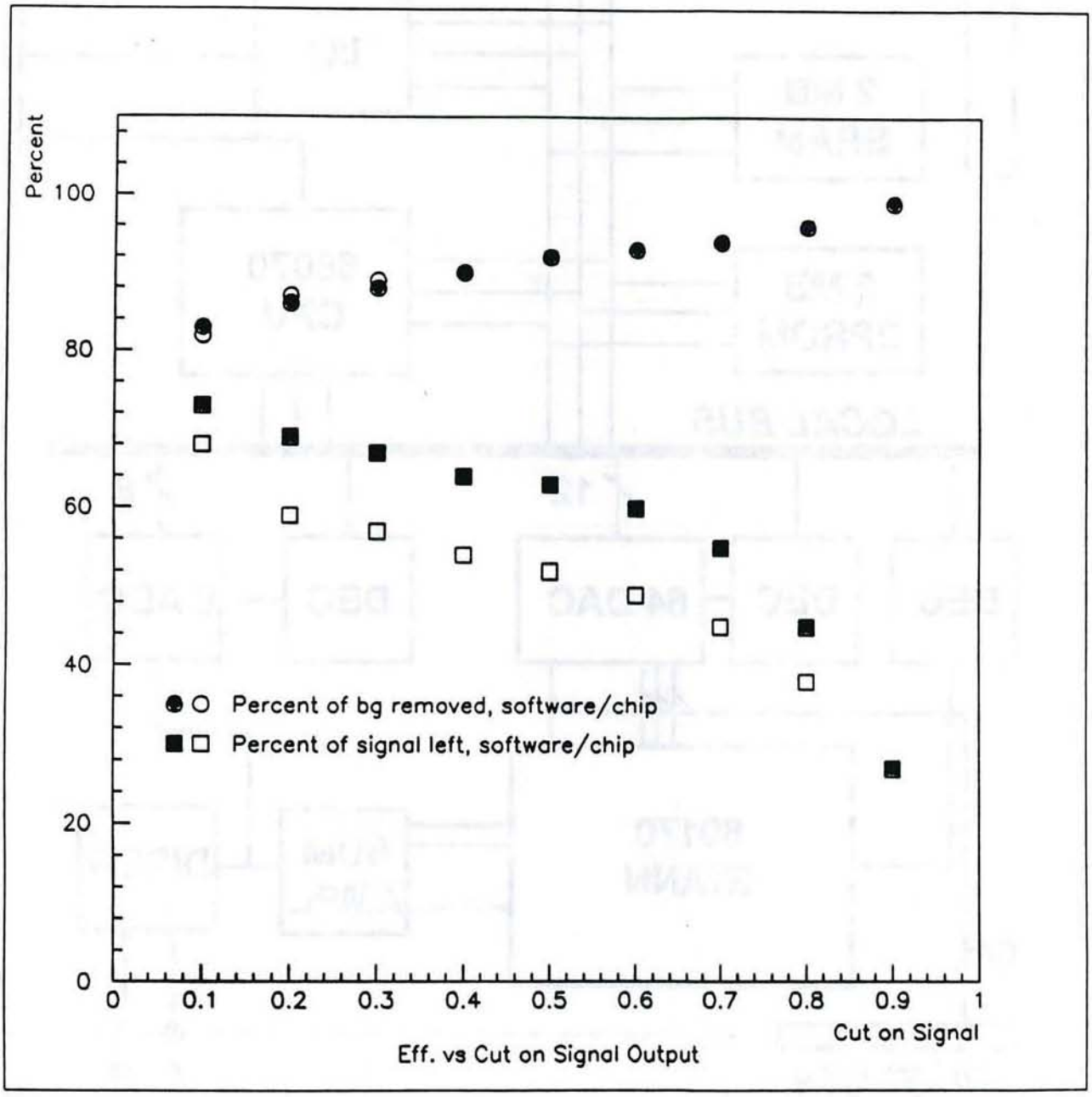


Figure 4. Identification efficiency versus cut on the output (fraction of output full scale.)

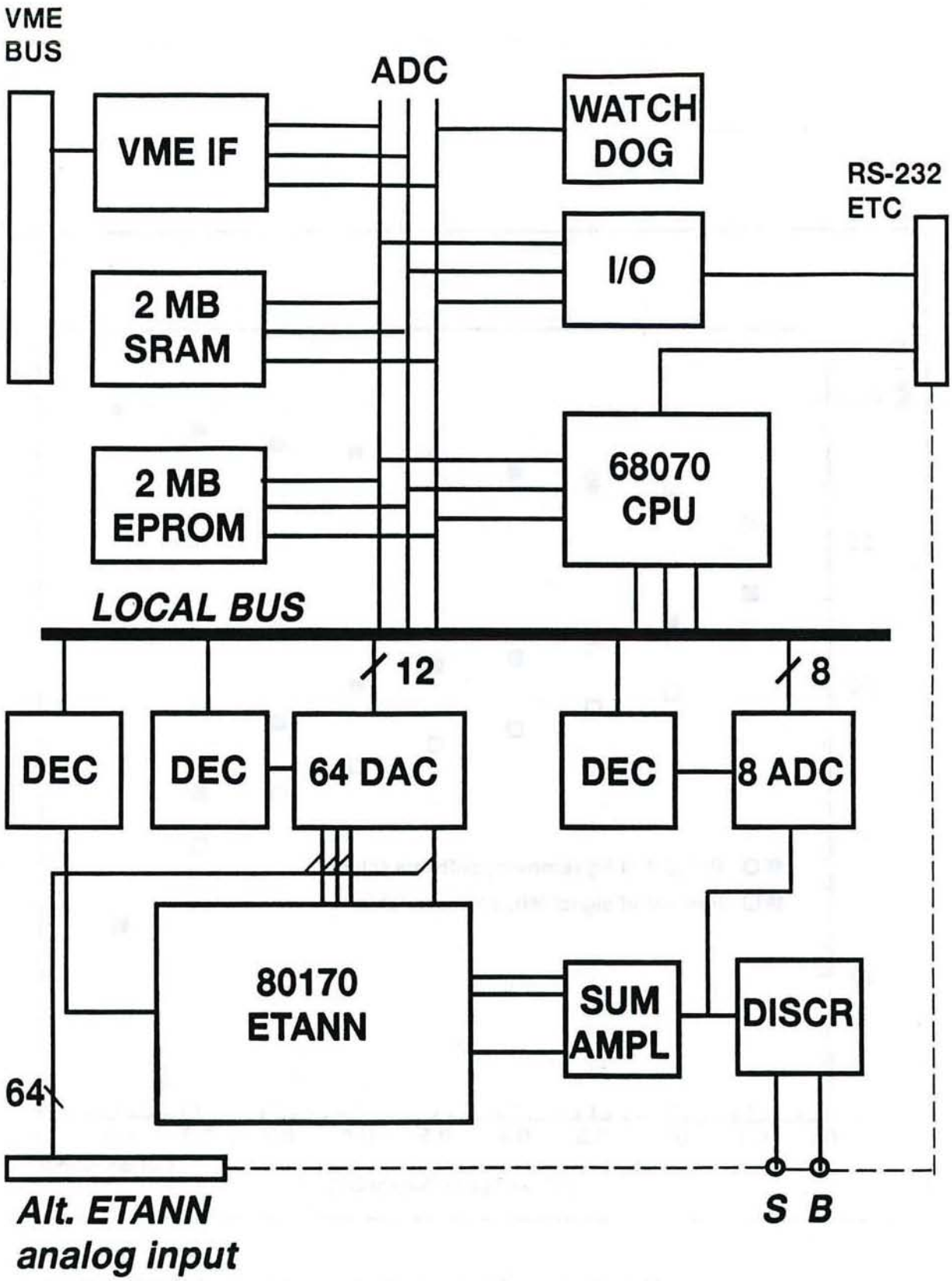


Figure 5. Block diagram of the hardware implementation on a VME prototyping card. The main structure is built around the local bus with the dual-ported SRAM as an essential part. A,D and C stands for address, data and control, respectively and DEC refers to decoding. Signals corresponding to 'signal' and 'background' are available following the discriminators (DISCR).

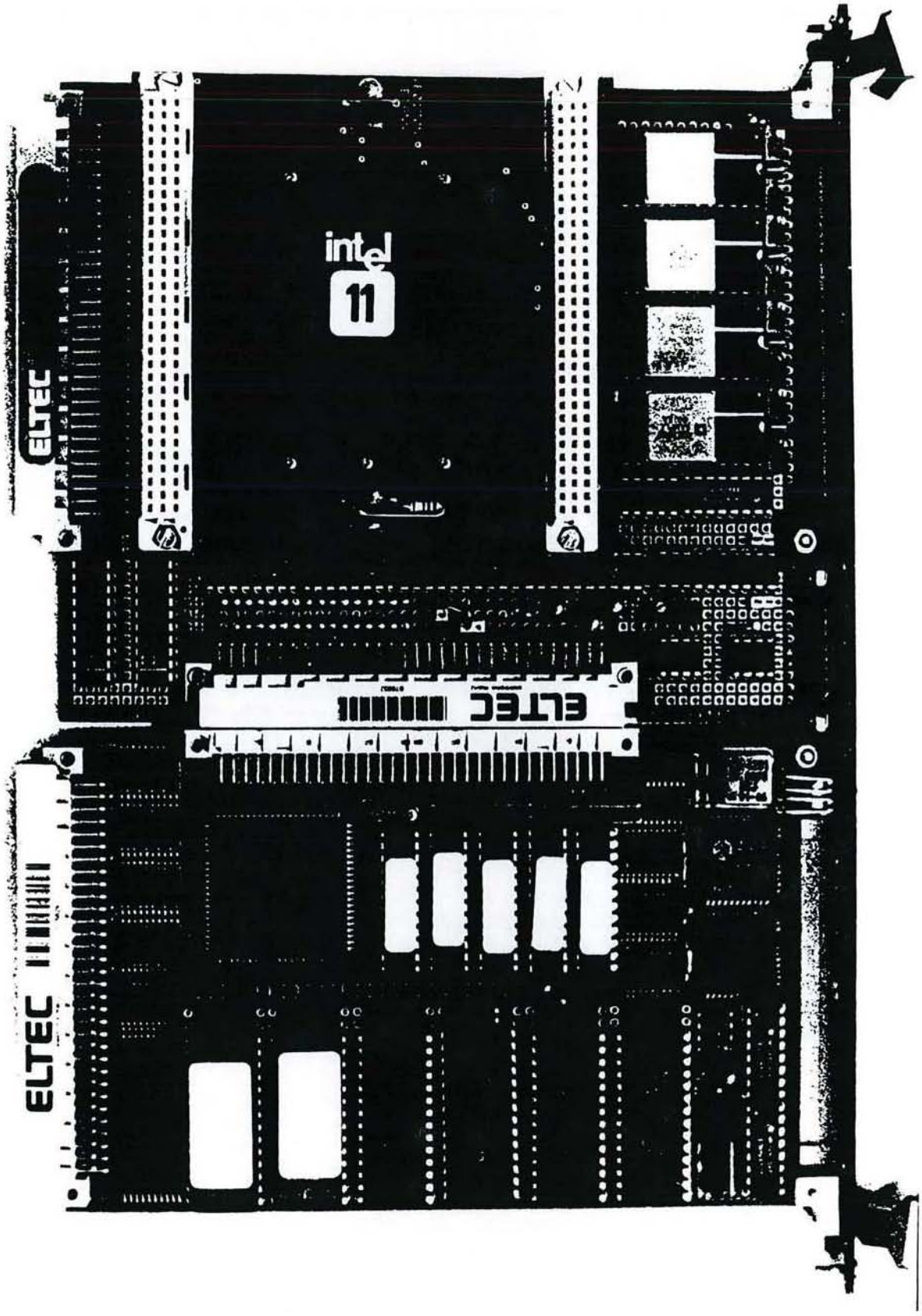


Figure 6. Photograph of the VME module sketched in fig. 5. Eight 8-fold DAC's are mounted below the ETANN piggy-back card.