ANIZATION FOR NUCLEAR RESEARCH

CERN/ECP 93-3
6 June 1993

THE BEAUTY CONTIGUITY TRIGGER OF THE BEATRICE EXPERIMENT: DETECTOR, READOUT AND PROCESSOR OVERVIEW

A. Beer, A. Corre, Ph. Farthouat, L. Malferrari,
M. Passaseo, V. Ryzhov, G. Schuler and M. Weymann(*)

CERN, Geneva, Switzerland

C. Bruschini, V. Casanova, G. Darbo, P. Martinengo, L. Rossi and C. Salvo

Dipartimento di Fisica and INFN, Genova, Italy

A. Frenkel, M.Torelli and L. Zanello

Dipartimento di Fisica and INFN, Roma, Italy

Abstract

This report describes the impact-parameter trigger used by the WA92 BEATRICE fixed-target experiment at the CERN Omega spectrometer. A comprehensive description of the microstrip detector, readout electronics and trigger processor is given. Preliminary results on its operation in the 1992 data collection are also presented.

Submitted to Nuclear Instruments and Methods in Physics Research

(*) Now at Creative Electronic Systems SA (CES), Geneva, Switzerland.

Mac/TeXtures/5023F/PF/DTP

1 INTRODUCTION

Among the various strategies for studying heavy quark physics, the selection of events with evidence of at least one secondary vertex has proved to be very effective [1]. The BEATRICE experiment [2] has two innovative features that are important for the study of beauty in fixed-target interactions: the Decay Detector [3], which behaves like an electronic bubble chamber thus allowing the direct observation of the beauty decays, and the Beauty Contiguity Trigger (BCT) which selects events with secondary vertices and therefore enriches the beauty signal over the background.

The BCT, a pixel processor executing a contiguity mask algorithm, uses the information coming from a specially designed microstrip detector and readout system. High interconnectivity among processing elements and highly parallel algorithms allow for separate counting of tracks belonging to primary and secondary vertices in less than 30 μs for any track multiplicity. The whole system is based on three specially designed ASIC chips.

2 SILICON VERTEX DETECTOR

The WA92 vertex telescope consists of 15 silicon-microstrip planes. There are 6 Z and 6 Y planes, and 3 W and 2 U planes, tilted by 11° respectively to the Z and Y planes. The Z and Y planes have 2048 strips and 25 μm pitch, while the W and U planes have a pitch of 50 μm and 512 or 1024 channels. Figure 1 shows an event from the 1992 data-taking as seen in the decay and vertex detectors; the full lines in the vertex detector represent the Z planes.

Only the 6 Z planes are used by the Beauty Contiguity Processor (BCP) at the trigger level. Ten more planes (5 Z and 5 Y, 512 strips, 20 μm pitch), located upstream of the target (Cu or W, 2 mm thick) are used as beam monitor. The second and last Z planes are used at the trigger level to estimate the interaction point in the target. A photograph of the detector set-up is shown in fig. 2.

3 FRONT-END ARCHITECTURE

The whole vertex detector plus the beam hodoscope have a total of 25 microstrip planes, of which 2 Z of the beam and the 6 Z and 6 Y from the vertex part use a fast readout and encoding system. The need for such a fast system comes from the use of all the former Z planes by the BCP. Since the processing time of the trigger processor is of the order of a few tens of microseconds the constraint on the readout has been put to $\sim$ 10 μs. In addition, we want to use, at the trigger level, the hit multiplicity; such information is provided in $\sim$ 2 μs by the front-end electronics.

A second constraint to the readout system comes from the high number of channels (more than 24 000). To avoid huge amounts of cables from the experimental area to the counting room, it was decided to process the data as much as possible on the detector itself, and to transmit only the coordinate of the hit strips to both the trigger processor and the data acquisition system.

The mechanics of the detector demands that groups of 128 microstrip channels be processed independently. This resulted in the placement of 16 electronic boards, of 5 mm maximum thickness, around each detector plane. The limited available space strongly

influenced the choice of the technology: CMOS for low power consumption and SMD for small thickness.

A block scheme of the front-end electronics is shown in fig. 3(a). One detector plane is equipped with 16 front-end (FE) boards, each of them processing the signals coming from 128 strips (fig. 3(b)). Two Kapton buses connect these boards, to a power board (PB), which distributes the power supplies, and to an interface board (IB), which distributes the timing signals and interfaces to the Fastbus readout module.

A photograph of the front-end system is shown in fig. 4. Inside the Kapton ring the 16 FEs are visible, while the PB and the IB are, respectively, on the left and right side.

4 READOUT ELECTRONICS

4.1 Front-end board

Two CMOS chips were designed to process the data: one for the analog part Integrated Charge AmplifieR (ICAR) and one for the digital part Fast Encoding and ReadOut System (FEROS).

4.1.1 Front-end chip: ICAR

A 16-channel CMOS monolithic chip has been designed by the firm Smart Silicon System(*) to process the strip signals. This chip is named ICAR and is a modified version of the Amplex chip [4]. The ICAR block diagram is shown in fig. 5, and its main characteristics are listed in table 1.

The ICAR chip has, on each input, a low-noise preamplifier followed by a shaper. The peaking time of the shaper can be adjusted (between 150 ns and 300 ns) and we use it as a delay. When the signal track-and-hold (TH) occurs, the analog value of each input is latched. A 16:1 multiplexer and an output buffer are used to read out the 16 analog values. The chip select (CS) allows, if required, another level of multiplexing.

Eight of these chips are mounted on each FE board; the output of each ICAR feeds an external comparator with a programmable threshold, as shown in fig. 3(b).

Both threshold (Th 1 to Th 8) and peaking time adjustment (PT 1 to PT 8) are programmable through a serial shift register and DACs. The outputs of these comparators feed another ASIC chip: the FEROS.

4.1.2 Readout chip: FEROS

This chip encodes in 10 bits the coordinates of the hit strips. It calculates the multiplicity within the group of 128 strips and sends it to the Kapton bus. A distributed adder tree is used to calculate the overall multiplicity; each FEROS contains one adder. The total multiplicity is made available to the IB via the Kapton bus.

The coordinates of the hit strips are stored for future readout, controlled by the IB board. For test purposes, a pattern can be written at the input of the FEROS chip.

The FEROS is implemented in a 1.2 μm, double-level metal HCMOS gate array from SGS Thomson Microelectronics (ref. ISB12011) and uses 4500 equivalent gates.

(*) Smart Silicon System, 23 Av. de Chailly, CH-1012 Lausanne, Switzerland.

The design was done at CERN on a DAZIX(*) CAE workstation using the cell library from SGS Thomson. Pre-layout and post-layout simulation were performed, first with the default parameters given in the library, and then with the actual parameters of the routed chip. The measurements made on the real chip completely agreed with the simulation results and no further iteration was needed.

The main characteristics of FEROS are listed in table 2 and a photograph of a FE board is given in fig. 6.

4.2 Readout boards

Two modules are used for this purpose: one IB on the detector and one Fastbus module (COROM: for COordinate and ReadOut Module) (fig. 7).

The IB is connected to the FE boards through the Kapton bus and to the COROM module through a 30 m, 20 pair differential bus (FEROS bus). Both modules receive the strobe (STR) signal from the first-level trigger and the clear (CL) signal from the second-level trigger. One COROM module is able to handle two detector planes so that only seven Fastbus modules are used for the full readout of the detector.

The FEROS bus uses a simple protocol: first an instruction cycle defines the type of transaction to be executed, and then the data transaction takes place until the IB sends an end-of-transmission message.

The IB/COROM have four main tasks:

- Distribution of the timing signals. As soon as a STR signal occurs, the IB starts the analog readout sequence and the FEROS encoding. When a CL occurs, the system is reset and is able to accept a new event (after the 2 μs reset time of ICAR). Special instruction cycles allow the simulation of STR and CL for test and calibration purposes.

- Readout of multiplicity and of the coordinates of hit strips is done in two steps: first, a slave handshake phase which allows fast readout of the hit multiplicity (used by the COROM to give a signal to the trigger logic if the multiplicity is inside a preset window) and of the coordinates to feed the BCP at high rate (programmable from 2.5 MHz to 10 MHz); next, a full handshake phase is used by COROM to build the data buffer, including programmable clustering. Both phases can be interrupted by the CL signal without introduction of dead time.

- Programming and control of threshold, peaking time values and test data.

- Monitoring of power supplies.

The performances of the overall system in terms of speed are as follows: (i) multiplicity is available in COROM 2.5 μs after the interaction; (ii) a first coordinate is available to the BCP 2.6 μs after the interaction and then is followed by one coordinate each 200 ns. The average time to encode and transfer an event to the BCP is 10 μs.

(*) Daisy Cadnetix

The readout of COROM is done by a VME Sub-system Bus (VSB) to the Fastbus interface (FVSBI)(*) in block transfer mode. To avoid software overheads, the COROM Fastbus slave port implements a daisy chain capability so that all the COROMs in the crate are seen as a single module. The Fastbus readout speed is 10 Mbyte/s (400 ns per word).

The design of these boards was done at CERN, using a DAZIX CAE workstation. The first description and simulation of COROM was made using a hardware description language (DABL from DAZIX). Then, it was implemented in a finite state machine in PLD and fully simulated again. No iteration was needed between the prototype phase and the production.

4.3 Calibration procedure

The aim of the calibration is the determination of threshold values so as to guarantee good efficiency and little noise. To perform a calibration run a scan is done, changing the threshold values in steps of 20 mV (this corresponds to $\sim$ 1/8 of the signal due to a minimum-ionizing particle crossing a 300 μm thick silicon detector). For each threshold value the detector is triggered by a pulse generator and read out several times (typically 100). For each group of 16 channels, corresponding to one comparator and one ICAR, a distribution of the number of hits read out versus the threshold value is found (fig. 8). This distribution is the integral of a Gaussian where mean and σ represent respectively the pedestal and the r.m.s. of the electronics noise. The Gaussians of all the ICARs are fitted and the thresholds are then fixed at 4 σ above the pedestal for the two planes of the beam hodoscope, and at 3 σ above the pedestal for the planes of the vertex detector.

5 TRIGGER ALGORITHM

The trigger algorithm is designed to find tracks crossing the detector and flag them as primary or secondary tracks, according to their impact parameters (IP). It works in the xz projection along which tracks are not bent by the magnetic field. A total of eight microstrip detectors take part in the trigger algorithm: two in the beam hodoscope are used to define the primary interaction vertex (obtained extrapolating the beam to the target centre) and six in the vertex detector are used to reconstruct the tracks emerging from the interaction. The secondary vertices we are looking for should be between the target and the first downstream detector (at x_{v_1}). The algorithm consists of the following steps:

Step 1: the Z coordinate of the interaction point z_v is estimated from the position of the beam

$$z_v = \frac{z_{b_1} - z_{b_2}}{x_{b_1} - x_{b_2}}(x_v - x_{b_1}) + z_{b_1} \; , \tag{1}$$

(*) FVSBI, Creative Electronic System, 70 rte du Pont Butin, CH–1213 Petit Lancy 1, Geneva (Switzerland).

where (x_{b_1}, z_{b_1}) and (x_{b_2}, z_{b_2}) are the coordinate of the two points used to estimate the interaction point assumed to be at the centre of the target x_v along the x axis (fig 9(a)).

Step 2: let us consider the equation of a straight line in the x–z plane in the form

$$z = ax + z_v + h \ , \tag{2}$$

where $h = 0$ for primary tracks and IP $\simeq |h|$ for secondary tracks. The transformation

$$\begin{cases} x' = x \\ z' = \dfrac{x_{v_6}}{x}(z - z_v) \end{cases} \tag{3}$$

applied to eq. (2) gives

$$z' = z'_{v_5} + h\frac{x'_{v_6} - x'}{x'} \ , \tag{4}$$

where

$$z'_{v_6} = z_{v_6} - z_v = ax_{v_6} + h \ . \tag{5}$$

The transformation (eq. (3)) maps primary tracks into parallel tracks while secondary tracks are mapped into hyperbolae (fig 9(b)).

Step 3: clusters of hits belonging to parallel (i.e. primary) tracks are counted, the result being the number of primary tracks found. Such points are erased and any points left are used to look for secondary tracks (fig. 10(a)).

The track finding is performed using a contiguity mask algorithm like in the DELPHI contiguity processor [5]. Such an algorithm works on the "pixels" of the event and not on the hit coordinates, and is suitable for implementation on a two-dimensional mesh of simple processing elements (PE).

In fig. 10(b) white pixels represent PEs with an internal flag reset to zero, black pixels are PEs with the flag set to one, the flag being set according to the detector output after transformation eq. (3) (in figs. 10(b) and (c) the beam direction is from the bottom upward). Each PE is connected to its four neighbours by a set of links. If, using a programmed rule (contiguity mask), we create links around an active pixel, corresponding to a fired microstrip, then when a track crosses the detector a connective path (unbroken link) will exist from bottom to top. A voltage applied to the bottom row (*bottom register*) should then be detected at the top row (*top register*) (fig. 10(b)). The detected signals correspond to the end points of the primary tracks. Counting the number of ones in the top register gives the number of primary tracks found. Back propagation in the net of

links starting from the top register allows deletion from the image of active pixels (i.e. points belonging to primary tracks (fig. 10(c)).

Step 4: for hyperbolic (i.e. secondary) tracks, calculation of the impact parameter is based on the consideration that following the transformation eq. (3), the addition of eq. (4) of the quantity

$$s_{x'}(h)w = h\frac{x' - x'_{v6}}{x'} \ , \tag{6}$$

where w is the microstrip pitch and $s_{x'}(h)$ is the function of h that transforms a track having IP $= |h|$ to one parallel to the x axis. If a track of this kind is found we can calculate its IP inverting $s_{x'}(h)$. $s_{x'}(h)$ is called the *shift function*, the reason of which will become clear in the following, and has the property of transforming secondary tracks into a primary track so that the same track-finding algorithm can be used for secondary track search (fig. 11(a)).

Contiguity masks are applied to the remaining pixels to search for secondary tracks fig. 11(b). If connective links between bottom and top register are found, this means that tracks with IP around h_1 exist. The vertical masks are shifted then fig. 11(c), by values given by the shift function for $h = h_2$ and connectivity, is tested once more. The shift and connectivity search are repeated until all the classes of IP are exhausted. At each step, new end points may be added to the top register; at the end of the search, the number of ones in the top register gives the number of secondary tracks found.

We wish to make a few remarks on the contiguity mask algorithm we use:

- The contiguity mask is a map from the points of the event to the horizontal and vertical links between contiguous pixels. The size and the shape could be different from point to point, but in our implementation in the processor only differences from row to row are allowed.
- The contiguity masks for primary tracks have both horizontal and vertical links because we want to use back propagation to delete points from the event, while the masks for secondary tracks have only vertical links.
- Inefficiencies from the detector can be overcome either by using a mask higher than one row or by bypassing inefficient rows (i.e. making them transparent). In the implementation we used for the 1992 data taking we repeated each track search six times, each time making transparent in turn one of the six rows. The result is that we could find tracks having at least five points in the detector (algorithm 5/6).

6 PROCESSOR ARCHITECTURE

The whole BCP architecture is based on Fastbus, and is organized around three different kinds of modules, i.e. two FEROS to Beauty Contiguity Trigger (FBCT) interfaces, eight Beauty Contiguity Processor Slices (BCPS) and a Beauty Contiguity Processor

Controller (BCPC). All the previous modules, in addition to the seven COROMs and to the FVSBI module, sit in the same Fastbus crate (fig. 12).

Several buses link the system:

- **The FEROS bus:**
  these 14 buses connect the IB to the COROM. The FEROS buses are also connected to the FBCTs for the two beam planes (ZB1 and ZB2) and for the six vertex planes measuring the z-coordinate (ZD1, ..., ZD6).

- **The BCPS Broadcast Input (BBI) bus:**
  the hits from the six vertex detectors, after being transformed by the FBCT, are transmitted trough the six BBI buses running on the auxiliary Fastbus backplane.

- **The Trigger Data Lines (TDL):**
  the preprocessed data from the eight BCPSs are transmitted through the TDLs to the BCPC for the final track counting.

In addition, two signals coming from the experiment are needed: the first-level trigger (T1) which starts the BCP, and a clear (CL) to reset it. The BCP can be aborted at any time by a clear signal. The result coming from the BCP contributes to the second-level trigger, which eventually starts the data acquisition of the event.

6.1 The BCP interface: FBCT

Two FBCT interfaces are part of the system, each of them operating on the hits of three vertex detectors (ZV1, ZV2, ZV3 and ZV4, ZV5, ZV6 respectively), and using the two beam planes (ZB1 and ZB2) to extrapolate the beam z-coordinate into the target. They are the slave on the FEROS bus and the master on the BBI buses.

Steps 1 and 2 of the trigger algorithm are implemented in the FBCTs as shown in the block diagram of fig 13. First the vertex in target z_v is found using two adders and three look-up tables (LUT B1, LUT B2 and LUT ZV), then the transformation (eq. (3)) is executed by an adder and a look-up table (LUT V1, ..., LUT V6) for each of the six vertex planes used in the trigger.

The processing in the FBCT requires 100 ns/hit and is pipelined with the readout from the detector, thus adding no further dead time to the trigger (apart for the time needed to find the primary vertex position, which is $\sim$ 300 ns).

6.2 The BCP slice: BCPS

The eight BCPSs are the core of the BCP. Their architecture is based on a two-dimensional array of processing elements (PE), each one behaving as a very simple processor. All the PEs are driven by the same clock and execute the same instruction stream, making the BCP a single-instruction multiple-data (SIMD) machine.

The internal structure of a PE is shown in fig. 14; each bubble represents a 1-bit element, they are:

- **Image Memory (IM):**
  the detector data transformed by the FBCT are put on the BBI buses and transferred

to the IMs. Each IM spies its BBI bus and sets itself if the datum is equal to its logical address. Once the event is transmitted, the array of IMs is a bitmap of event as is shown in fig. 10, and the program in the BCPSs starts.

- **Working Memory (WM):**
  the WMs, in number of four, are general-purpose registers that can be used at any time in the programs.

- **Horizontal Connection Memory (HCM):**
  the HCM stores the status of the horizontal "switch" connecting the node with the one to its right-hand side. A "one" means that the switch is closed and a "zero" that it is open.

- **Vertical Connection Memory (VCM):**
  the VCM is the same as HCM, but it stores the status of the switches connecting the node itself with the bottom one. The pattern generated together by HCM and VCM is represented in fig. 10(a) and (b) by the links between nodes.

- **Contiguity Mask Network (CMN):**
  the CMN has no internal memory states and acts as a large combinatorial network of switches, where the signal loaded in the bottom (or top) registers propagates through the "closed switches" to the top (or bottom) register. The CMN is the basic PEs processing structure on which steps 3 and 4 of the algorithm are based.

- **Contiguity Mask Memory (CMM):**
  the CMM is again a memory structure able of reading the nodes of the CMN and is useful for tracing the track following in the six rows of the PEs.

The arrows in fig. 14 show the data flow originated by the instruction set. The HCM, VCM and CMN have, in addition, connections to the four neighbour PEs.

The PEs array have been built using a dedicated ASIC chip called CP232; 64 PEs are placed on one chip, 24 chips are in one BCPS, and 192 chips are needed for the whole trigger. The CP232 is implemented using the same 1.2 μm HCMOS technology from SGS Thomson Microelectronics used for the FEROS gate array, and the design has been done in Genoa on a DAZIX workstation. The main features of the CP232 are given in table 3. The large square chips in fig. 15 are the CP232.

The architecture of the BCPS [6] is shown in the block diagram of fig. 16. The main components are:

- **The PEs array:**
  there are 256×6 PEs in each BCPS and they receive their event data through six BBI buses. The first and last PEs column carry input and output lines to the BCPSs sitting respectively on the right-hand and left-hand sides; the PEs in the eight BCPSs can be considered as a unique array of 2048×6, reflecting the vertex detector layout. Two registers are connected to the upper (top register) and lower (bottom register) edge of the PEs array and can be used either to feed-in or read-out the data in the array. Finally, data in the top register are grouped by eight and are sent out to the

32 TDLs connecting each BCPS to the BCPC. These lines transport the head points of the tracks found at steps 3 and 4 of the algorithm.

- **The Program and Data Memory (PDM):**
  the memory stores the trigger program and the initialization parameters. It has 8 k words 64-bit long. Instructions are fetched from PDM in succession and no changes in program flow (jump and conditional branches) are allowed by the BCP architecture. All PEs execute the same instruction, two instruction lines per PE row are used as enable or modifier of the instruction itself. It has been chosen to have the PDM replicated on each BCPS being simpler to carefully control the precise timing of the single-clock line than the 64-bit wide instruction bus.

- **The Front-End Buffer (FEB):**
  data from PEs can be down-loaded in groups of 32 columns into the FEB for later readout by Fastbus. The FEB is 48-bit wide with 32 bits used for data and 16 bits for data identifier.

- **The Registers (IREG, DREG and OREG):**
  there are two pipeline registers that are used to equalize the delays of different components in the BCPS and to allow the clock to run at its maximum speed. These registers are the Data REGister (DREG) and the Instruction REGister (IREG) at the output of the PDM, and the Output REGister (OREG) between the Top REGister (TREG) output and the FEB input. These registers add only a delay to the execution of the current instructions, but are completely transparent to the programmer since no breaks of the program flow are permitted by the BCPS architecture.

Owing to the architecture used, the time needed to reconstruct the tracks and measure their IP depends neither on the number of tracks nor on the number of hits in the event, but only on the number of instructions executed by the processor. The number of hits in the event (but not the number of tracks) affects only the time needed for the detector readout.

6.3 The BCP controller (BCPC)

The BCPC is the last of the BCP modules and has two functions: it counts the tracks found by the BCPSs to take the trigger decision and generates the control signals for the whole BCP.

Tracks end points from the BCPSs are received through the TDLs as hit patterns in 256-bit long words. Each of these words could represent tracks belonging to a primary vertex or to different IP classes. The BCP trigger (T2) decision is finally taken as a function of the number of tracks in those words. Such a result is provided 300 ns after the BCPSs have finished processing.

The BCPC receives the two signals CL and T1 and generates all the timing and control signals needed to the FBCTs and to the BCPSs.

6.4 The BCP cross-assembler and simulator

A trigger system of the complexity of the BCP has required an important effort in designing a rather large set of software packages, the main ones being a cross-assembler and two simulators [6].

The programming-like language of the BCP is a macro-assembler; a cross-compiler (named Eve) has been written in VAX11 macro. The Eve compiler translates each instruction of the source code into the 64-bit BCPS machine instructions.

Of the two simulators, one named Helen, runs the direct BCP source code, while the other executes the trigger program at the algorithm level and is included in the simulation package of the experiment. Helen can run in interactive debugging mode, which allows the step-by-step tracing of the program and the inspection of all the internal BCPSs' memory structures (fig. 17). The "production mode" of Helen allows the processing of up to 10 events/s and is useful for statistical analysis of the trigger algorithms. Helen can run either on simulated or real events, the second option being very useful for hardware monitoring during data taking.

7 ISSUES FROM 1992 DATA TAKING

The BCT was successfully operated during the 1992 BEATRICE data taking, when some 90 million events were collected. While the data analysis is under way, some comments can be made about the performances of the BCT. The trigger program was originally designed to reconstruct tracks having six hits, one per plane. As a result of the experience gained in the 1991 test run, the program was modified to accept tracks with at least five hits, allowing for one plane being missed. This has raised the processing time from 10 μs to 30 μs, in addition to the readout time ($\sim$ 10 μs). With the new algorithm, the average number of primary tracks reconstructed per event increases from 5.3 to 7.7 (on a Cu target, with real events) and the beauty signal acceptance from 41% to 58% (on a Cu target with simulated data and a fully efficient detector).

As trigger condition we asked for at least three tracks (3P) pointing to the interaction vertex in the target (to constrain the primary vertex to be in the target itself and to deplete the contribution of interactions downstream to the target) and at least two tracks (2S) missing the primary vertex by more than 100 μm to select events with secondary vertices. In fig. 18 the distribution of the x-coordinate of the vertices were reconstructed according to different processor requirements. The 3P condition depletes downstream primary interactions with respect to minimum-bias interactions (INT), while the 2S condition selects events with downstream vertices.

The beauty acceptance, on simulated events, for this trigger condition is more than 55% while the trigger rate was 7% using a Cu target and 10% with a W target (both 2 mm thick), corresponding to a rejection factor, for minimum-bias interactions, of 14 and 10 for Cu and W, respectively. The higher rate we observed in tungsten is a consequence of the higher tracks multiplicity per event (we have, in average, two more tracks in W than in Cu). The BCP results have been cross-checked with an off-line analysis which has confirmed the tracks found by the BCP and their impact parameter measurements.

Acknowledgments

We are grateful to L. Cracco, G. Critin, C. Veccia, and F. Vernocchi for the excellent technical support they gave during the many phases of the development of the system. We want to thank, as well, G. Brouant, M. Dalbignat, J.P. Marcelin and C. Millerin of the B.W. Heck's ECP/PES group together with A. Montfort (SL/CO) for the support in the layout and production of the many printed circuit boards needed by the project.

We are indebted to E. Flaminio and C. Lazzeroni for the precious help they gave us in understanding and monitoring the quality of the data coming from the Vertex Detector and BCP. Finally, we want to thank F. Bourgeois and P. Innocenti for the strong support they gave to the project.

References

[1] M. Adamovich et al., IEEE Trans. Nucl. Sci. NS–37, No. 2 (Apr. 1990) 236–240;
G. Darbo and L. Rossi, Nucl. Instr. and Meth. **A289** (1990) 584–591.

[2] M. Adamovich et al., Nucl. Phys. B27 (1992) 251–256.

[3] M. Adinolfi et al., A Microstrip decay detector for beauty physics, CERN/PPE 92–181, to appear in Nucl. Instr. and Meth. in Phys. Res. A.

[4] E. Beuville et al., Nucl. Instr. and Meth. **A288** (1990) 157–167.

[5] G. Darbo and S. Vitale, Nucl. Instr. and Meth. **190** (1981) 81–88;
G. Darbo, B.W. Heck and J.M. Wildman, IEEE Trans. Nucl. Sci. NS–38, No. 2 (Apr. 1991) 861–865.

[6] C. Bruschini, Studio di un trigger rapido su vertici secondari per la ricerca di particelle dotate di beauty, Thesis, University of Genova.

TABLES

Table 1

Parameter	Value	
Number of inputs	16	
Power consumption	67	mW
Gain	40	mV/fC
Peaking time	200	ns
Eq. input noise ($C_{det} = 5$ pF)	1000	e^-
Reset time	2	μs
Readout frequency	> 6	MHz

Table 2

Parameter	Value	
Number of input channels	128	
Encoding time	25	ns
Multiplicity calculation	25	ns

Table 3

Parameter	Value
Device type	ISB 12054
Number of eq. gates	17312
Package	CPGA 145
Number of I/O	129
Clock (in system design)	20 MHz

FIGURE CAPTIONS

Fig. 1 The detector set-up for the BCT. An event from the 1992 data taking is shown inside the vertex and decay detectors.

Fig. 2 View of the vertex detector in the experimental area.

Fig. 3 (a) Block scheme of the electronics set-up around a microstrip detector plane; (b) Scheme of principle of one front-end board.

Fig. 4 View of the front-end system.

Fig. 5 The ICAR block diagram.

Fig. 6 View of the FE board. From top to bottom, one can see the connections to the detector, the eight ICAR chips, the FEROS chip and the three connectors to the Kapton bus.

Fig. 7 Side view of COROM.

Fig. 8 Number of hits read out changing the threshold of one ICAR chip fitted with the integral of a Gaussian. The relative Gaussian is shown below.

Fig. 9 (a) Step 1 of the algorithm; (b) Step 2 of the algorithm.

Fig. 10 Step 3 of the algorithm.

Fig. 11 Step 4 of the algorithm.

Fig. 12 Readout and Beauty Contiguity Trigger (BCT) architecture.

Fig. 13 FEROS to Beauty Contiguity Trigger (FBCT) interface.

Fig. 14 Processing Element (PE).

Fig. 15 BCPS: the 24 gate arrays housed in a BCPS board are visible at the top.

Fig. 16 Beauty Contiguity Processor Slice (BCPS).

Fig. 17 Helen: screen dump of the BCP interactive simulator.

Fig. 18 X position of reconstructed primary vertices. The profile of the density of the target and of the first 10 planes of the decay detector is clearly visible in all three distributions (INT).

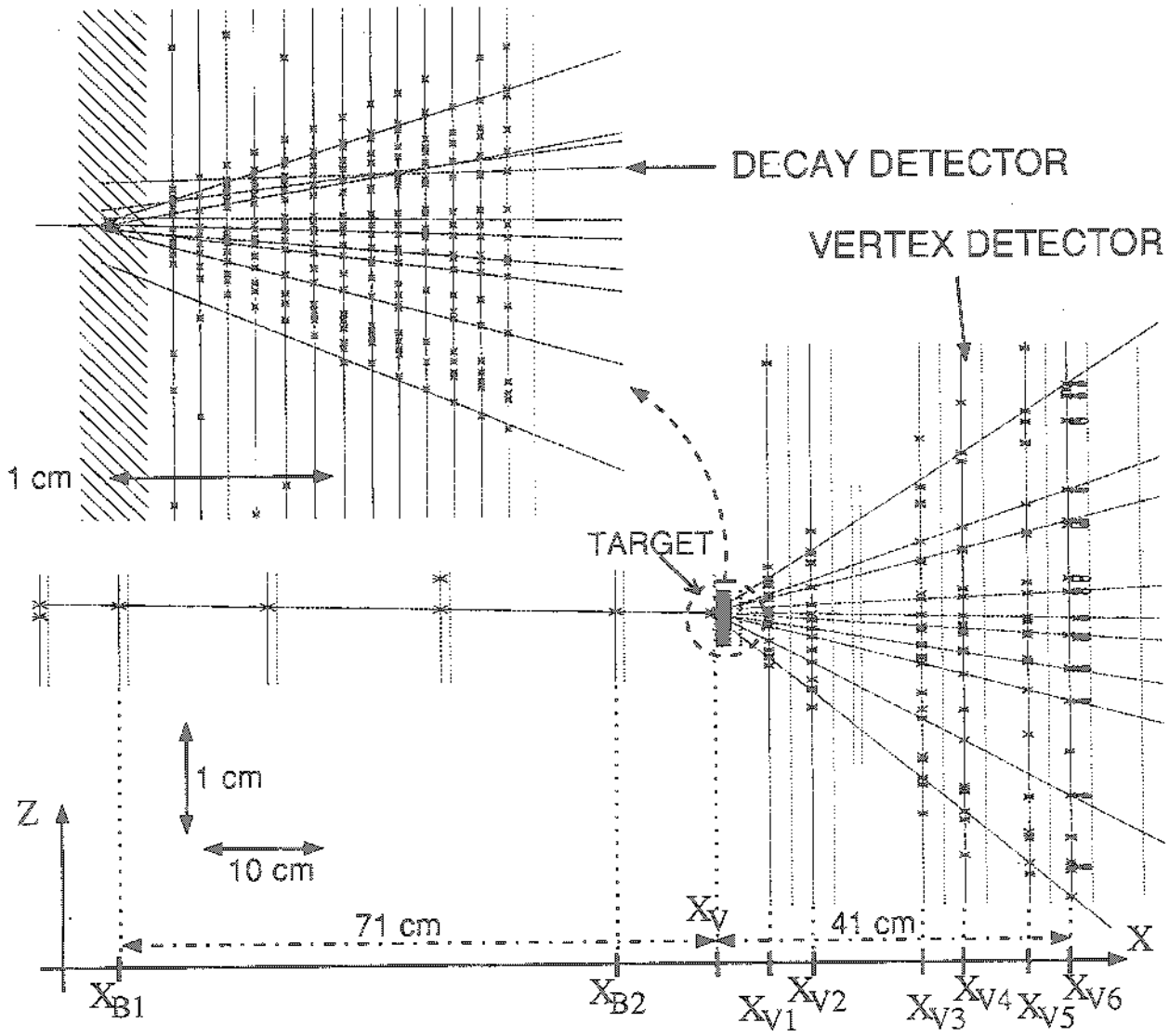

DECAY DETECTOR
VERTEX DETECTOR
1 cm
TARGET
1 cm
10 cm
Z
71 cm
41 cm
X
X_V
X_{B1}
X_{B2}
X_{V1}
X_{V2}
X_{V3}
X_{V4}
X_{V5}
X_{V6}

Fig. 1

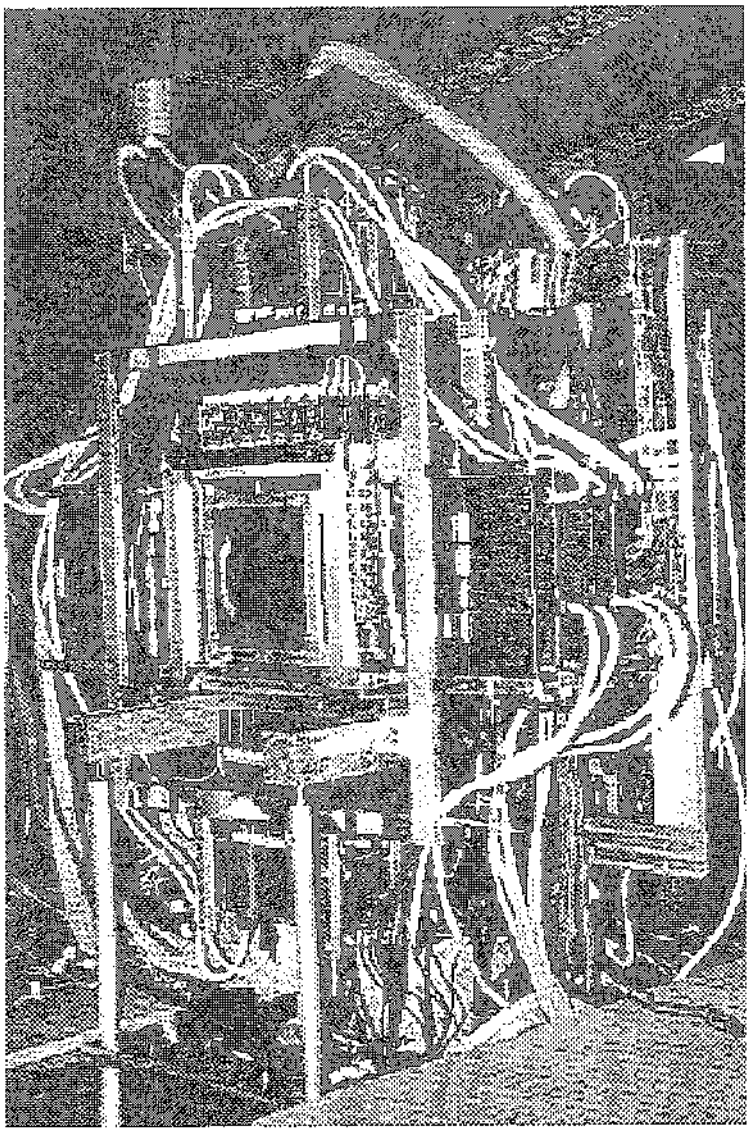


Fig. 2

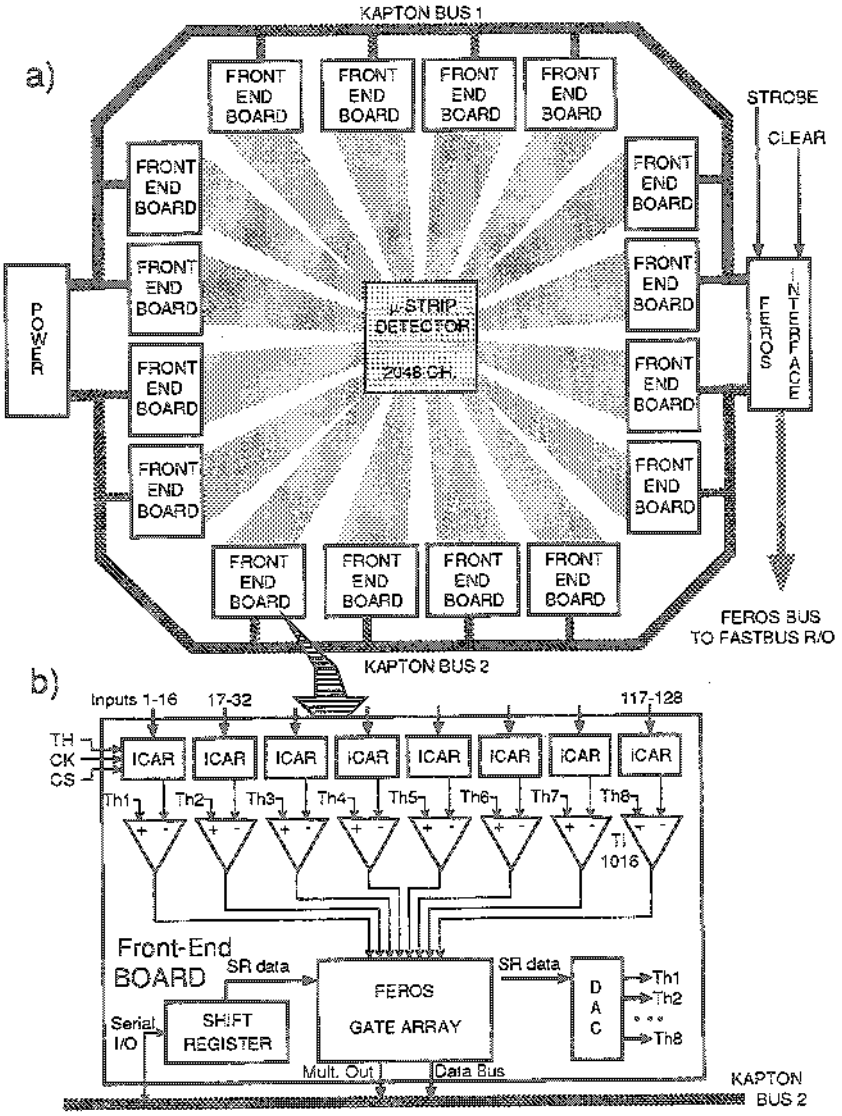

a)
KAPTON BUS 1
FRONT END BOARD
STROBE
CLEAR
POWER
μ-STRIP DETECTOR 2048 CH.
FEROS INTERFACE
FEROS BUS TO FASTBUS R/O
KAPTON BUS 2
b)
Inputs 1-16
17-32
117-128
TH
CK
CS
ICAR
Th1
Th2
Th3
Th4
Th5
Th6
Th7
Th8
TH 1016
Front-End BOARD
SR data
SHIFT REGISTER
Serial I/O
FEROS GATE ARRAY
SR data
D A C
Th1
Th2
Th8
Mult. Out
Data Bus
KAPTON BUS 2

Fig. 3

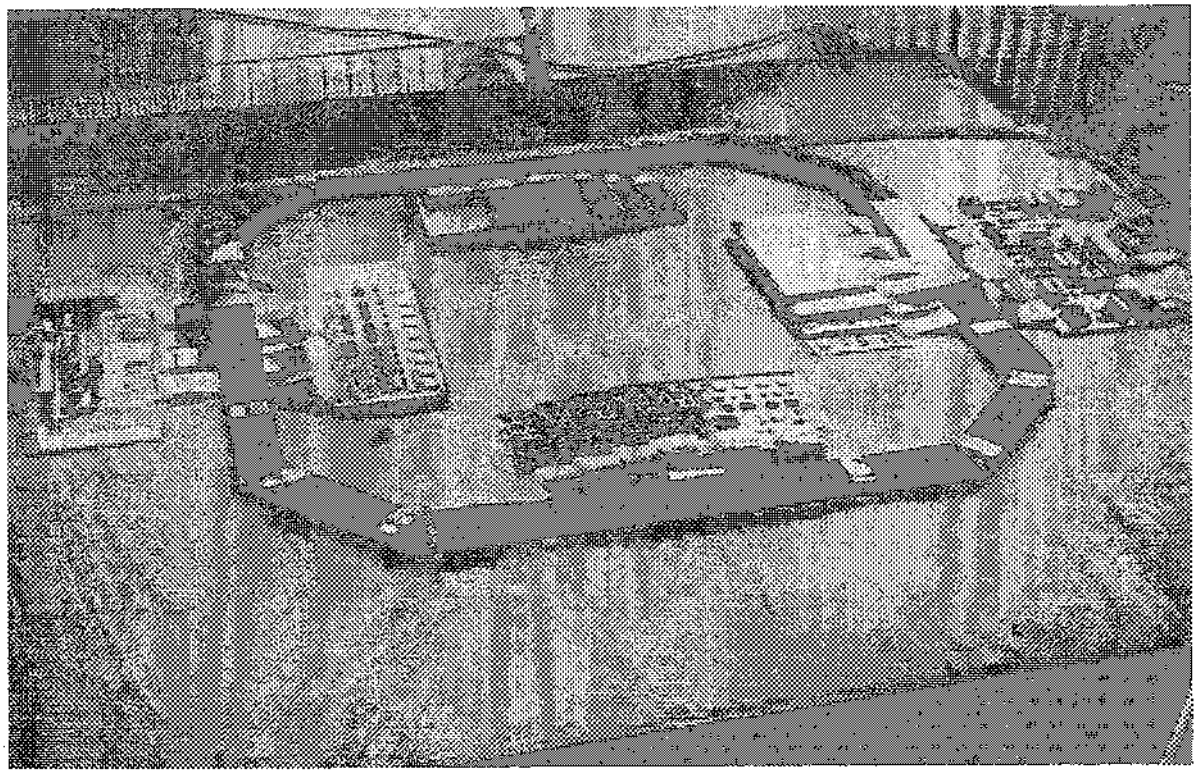


Fig. 4

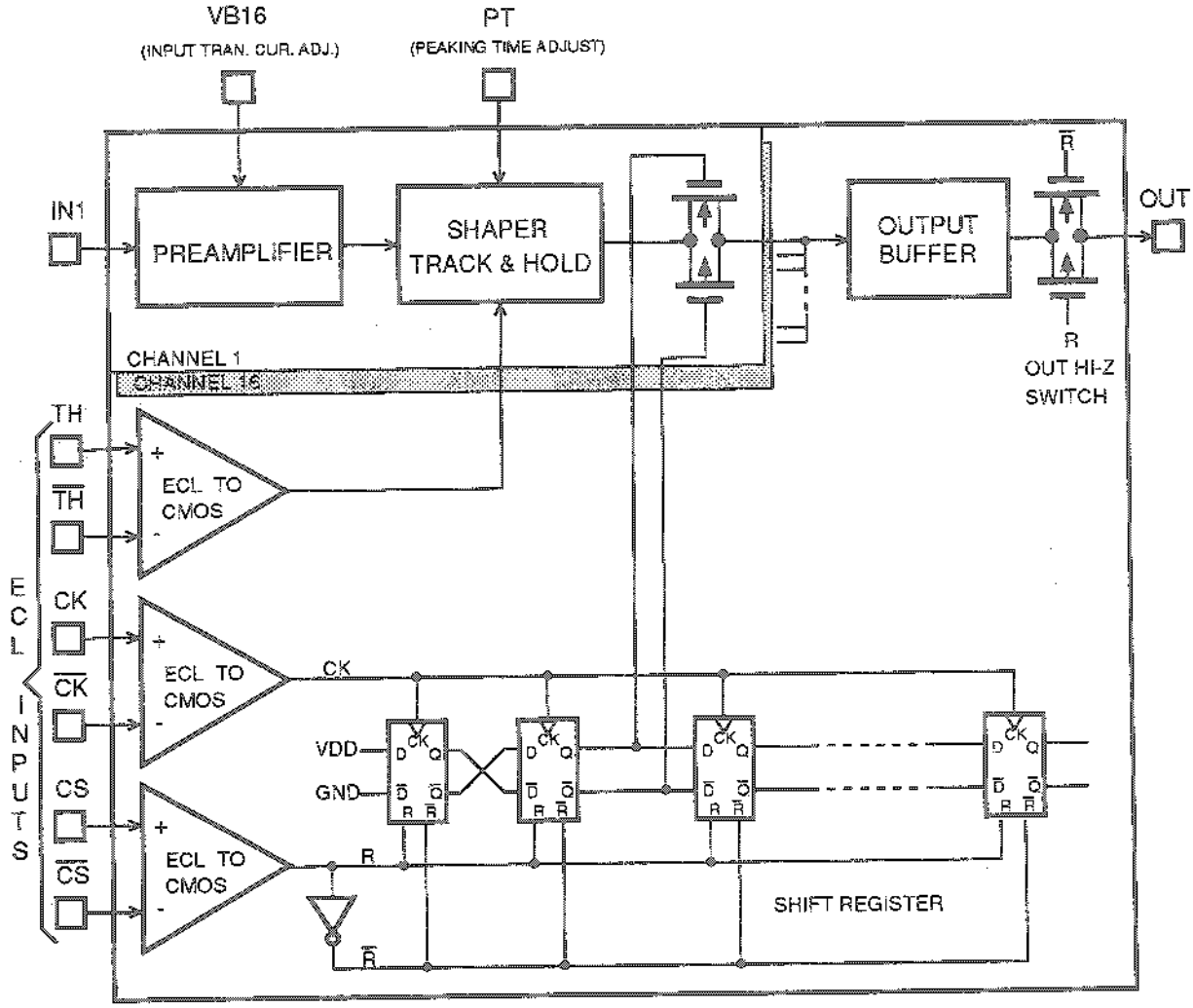

VB16
(INPUT TRAN. CUR. ADJ.)
PT
(PEAKING TIME ADJUST)
IN1
PREAMPLIFIER
SHAPER
TRACK & HOLD
OUTPUT
BUFFER
OUT
R
R
OUT HI-Z
SWITCH
CHANNEL 1
CHANNEL 16
TH
TH
CK
CK
CS
CS
ECL INPUTS
ECL TO
CMOS
ECL TO
CMOS
ECL TO
CMOS
CK
VDD
GND
R
R
SHIFT REGISTER

Fig. 5

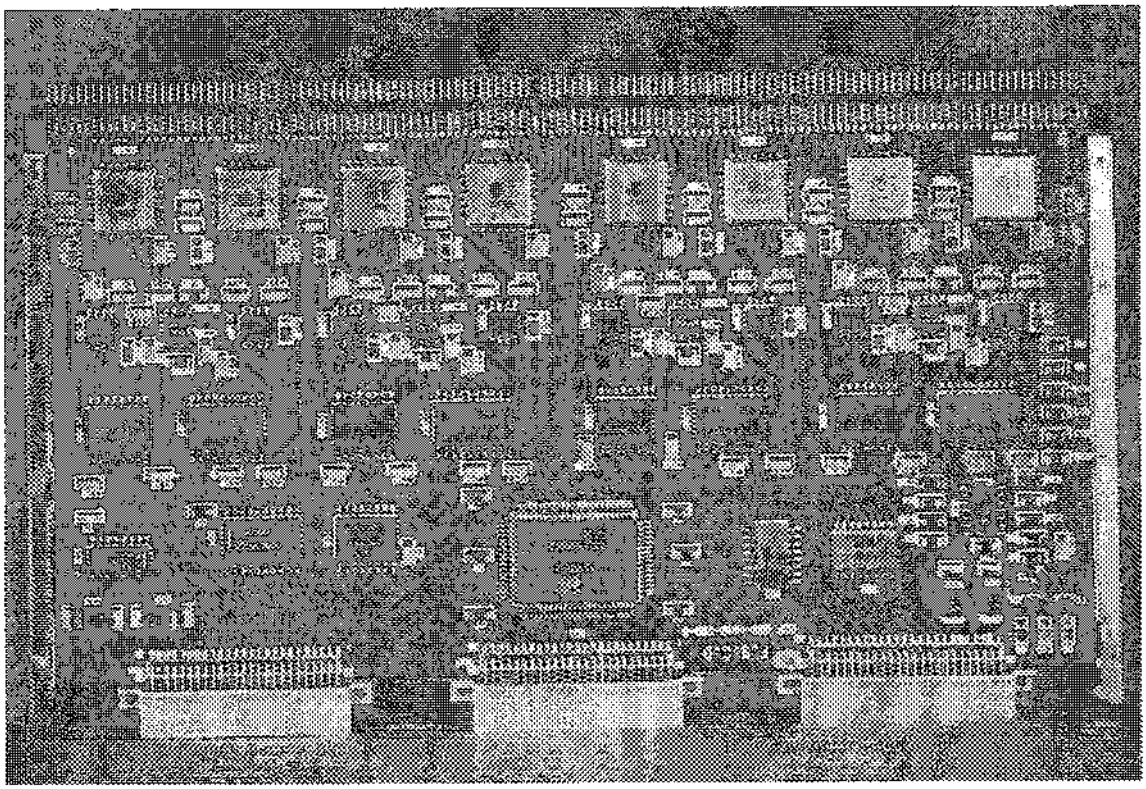


Fig. 6

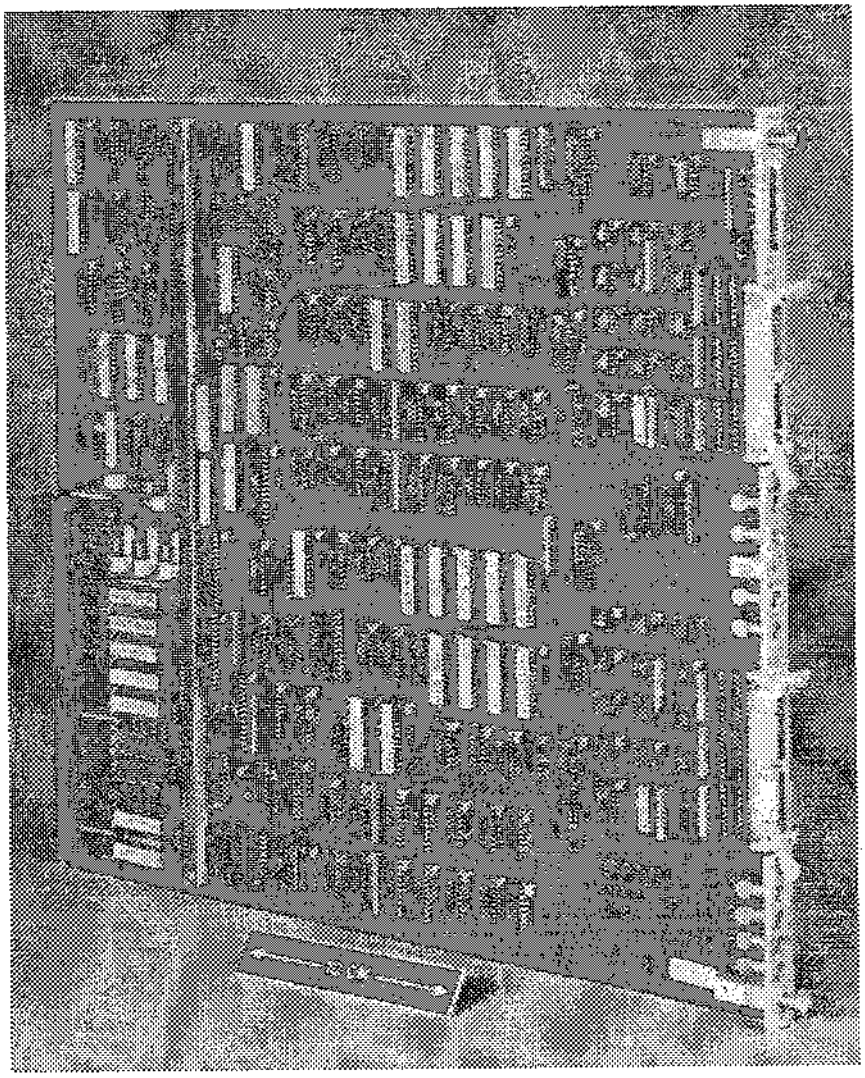


Fig. 7

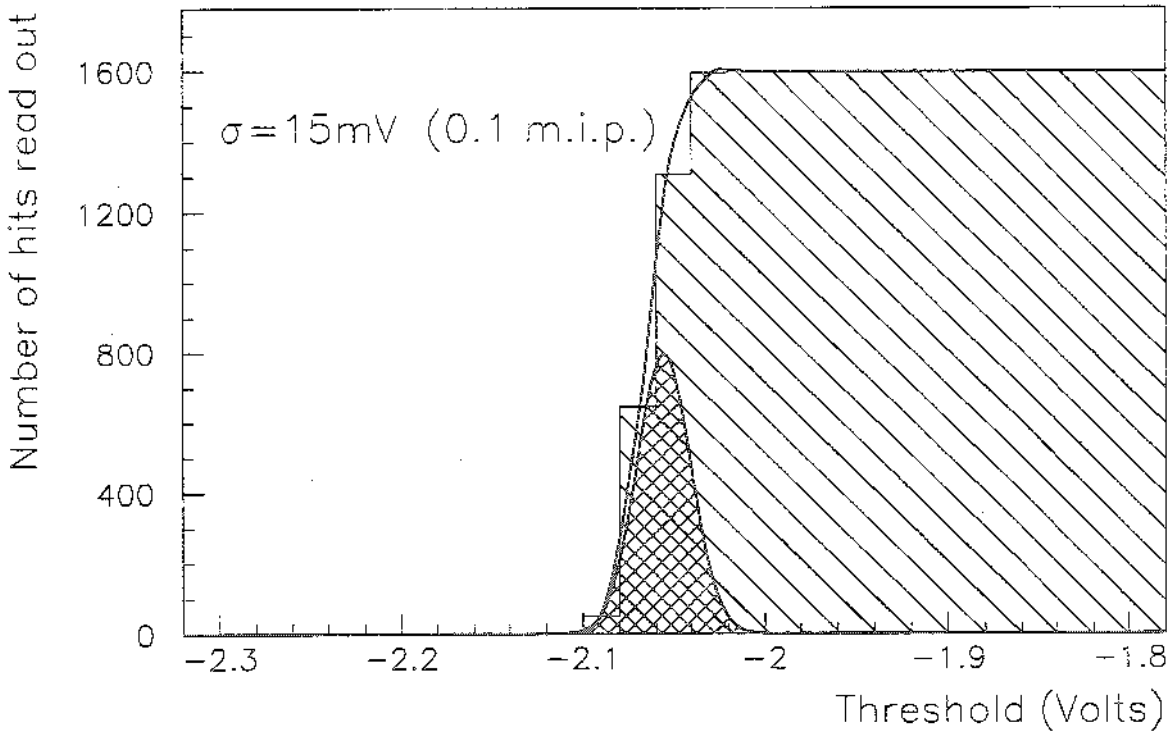

Number of hits read out
1600
1200
800
400
0
σ=15mV (0.1 m.i.p.)
−2.3
−2.2
−2.1
−2
−1.9
−1.8
Threshold (Volts)

Fig. 8

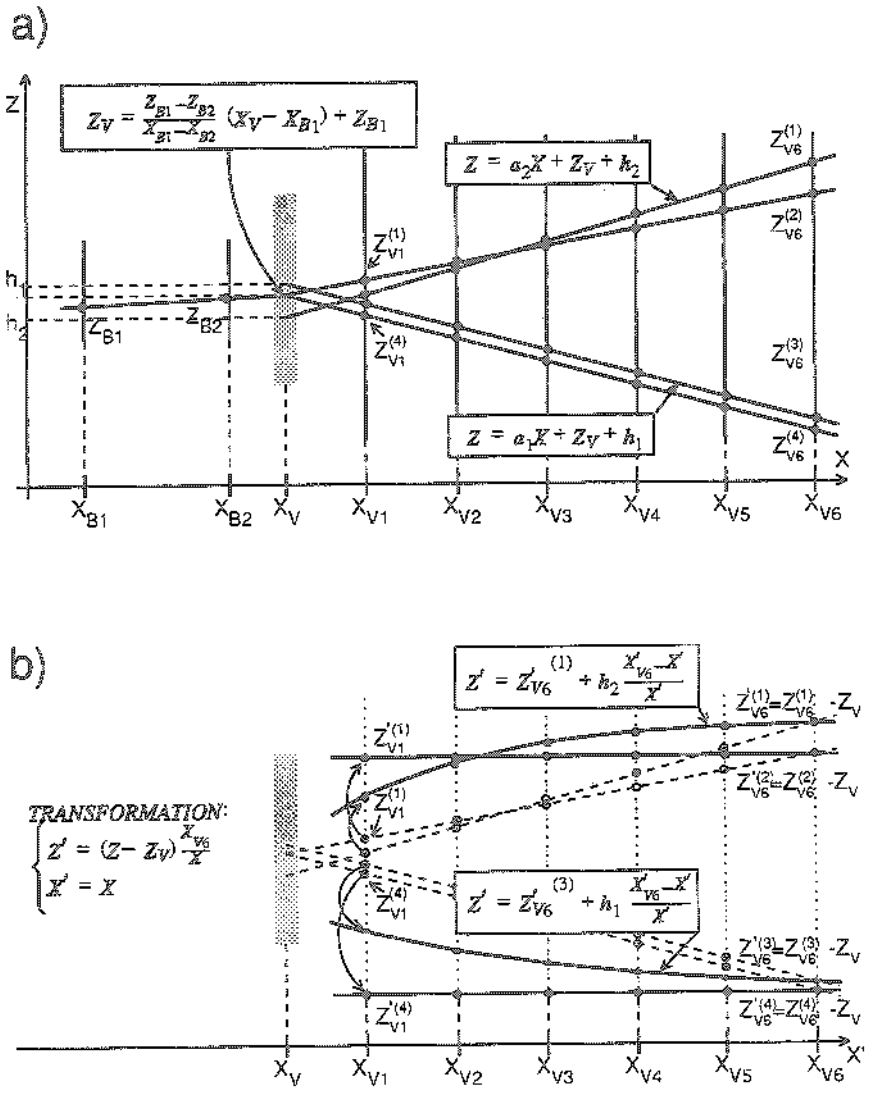

a)
Z
$Z_V = \frac{Z_{B1}-Z_{B2}}{X_{B1}-X_{B2}}(X_V - X_{B1}) + Z_{B1}$
$Z = a_2X + Z_V + h_2$
$Z = a_1X + Z_V + h_1$
$Z^{(1)}_{V6}$
$Z^{(2)}_{V6}$
$Z^{(3)}_{V6}$
$Z^{(4)}_{V6}$
$Z^{(1)}_{V1}$
$Z^{(4)}_{V1}$
h_1
h_2
Z_{B1}
Z_{B2}
X
X_{B1} X_{B2} X_V X_{V1} X_{V2} X_{V3} X_{V4} X_{V5} X_{V6}
b)
TRANSFORMATION:
$Z' = (Z - Z_V)\frac{X_{V6}}{X}$
$X' = X$
$Z' = Z'^{(1)}_{V6} + h_2\frac{X'_{V6}-X'}{X'}$
$Z' = Z'^{(3)}_{V6} + h_1\frac{X'_{V6}-X'}{X'}$
$Z'^{(1)}_{V6}=Z^{(1)}_{V6}-Z_V$
$Z'^{(2)}_{V6}=Z^{(2)}_{V6}-Z_V$
$Z'^{(3)}_{V6}=Z^{(3)}_{V6}-Z_V$
$Z'^{(4)}_{V6}=Z^{(4)}_{V6}-Z_V$
$Z'^{(1)}_{V1}$
$Z^{(1)}_{V1}$
$Z^{(4)}_{V1}$
$Z'^{(4)}_{V1}$
X'
X_V X_{V1} X_{V2} X_{V3} X_{V4} X_{V5} X_{V6}

Fig. 9

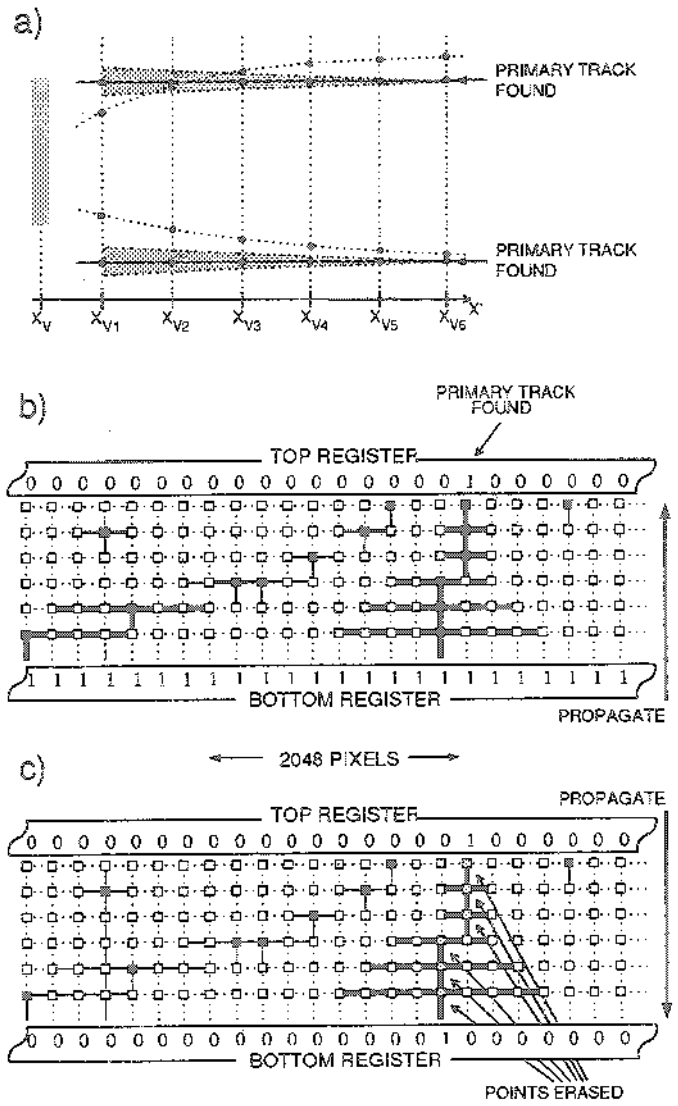

a)
PRIMARY TRACK FOUND
PRIMARY TRACK FOUND
X_V X_{V1} X_{V2} X_{V3} X_{V4} X_{V5} X_{V6} X
b)
PRIMARY TRACK FOUND
TOP REGISTER
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
BOTTOM REGISTER
PROPAGATE
2048 PIXELS
c)
PROPAGATE
TOP REGISTER
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
BOTTOM REGISTER
POINTS ERASED

Fig. 10

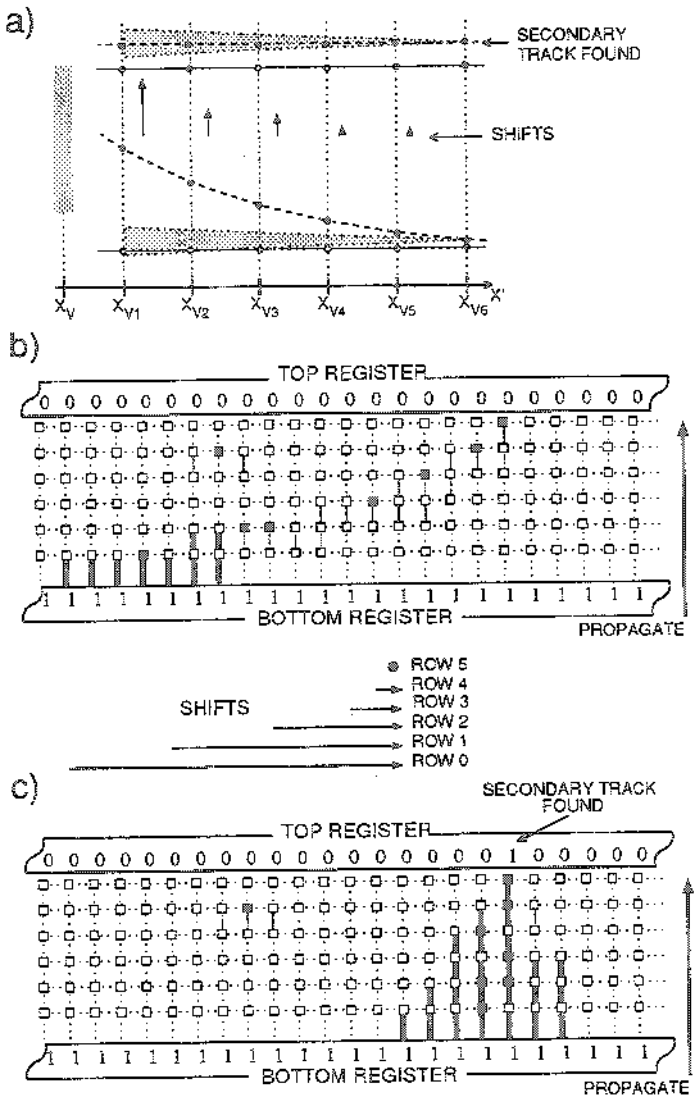

a)
SECONDARY TRACK FOUND
SHIFTS
X_V X_{V1} X_{V2} X_{V3} X_{V4} X_{V5} X_{V6} X'
b)
TOP REGISTER
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
BOTTOM REGISTER
PROPAGATE
ROW 5
ROW 4
ROW 3
ROW 2
ROW 1
ROW 0
SHIFTS
c)
SECONDARY TRACK FOUND
TOP REGISTER
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
BOTTOM REGISTER
PROPAGATE

Fig. 11

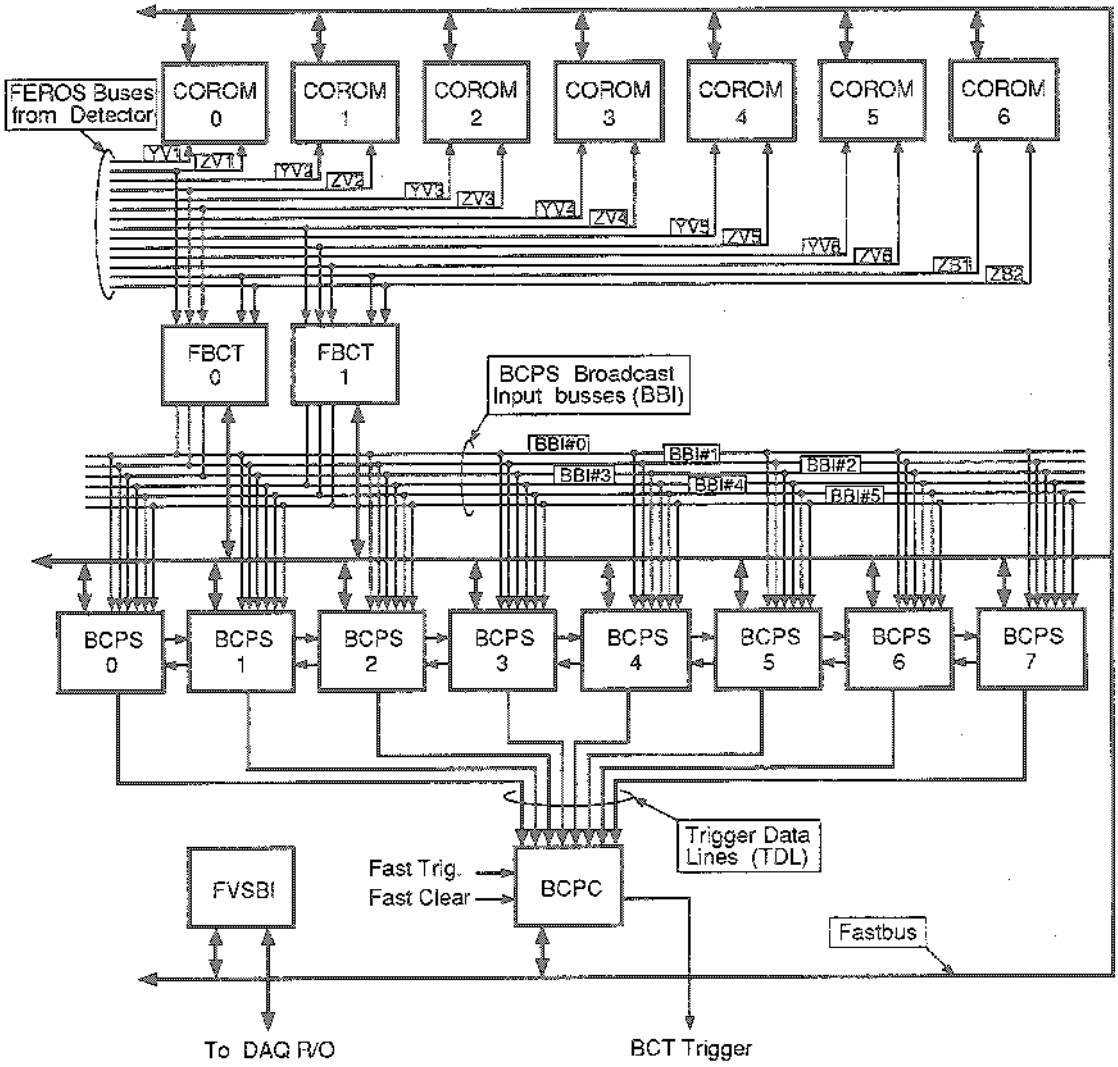
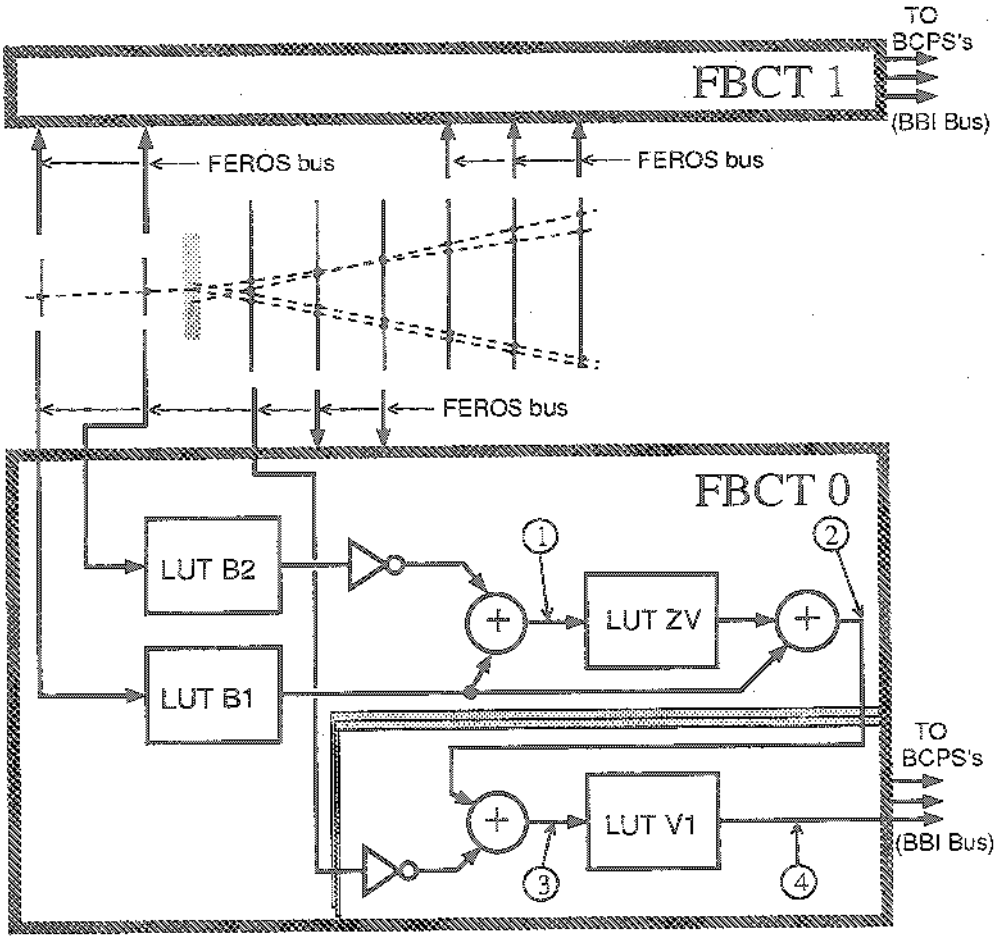

FEROS Buses from Detector
COROM 0
COROM 1
COROM 2
COROM 3
COROM 4
COROM 5
COROM 6
YV1
ZV1
YV2
ZV2
YV3
ZV3
YV4
ZV4
YV5
ZV5
YV6
ZV6
ZS1
ZS2
FBCT 0
FBCT 1
BCPS Broadcast Input busses (BBI)
BBI#0
BBI#1
BBI#2
BBI#3
BBI#4
BBI#5
BCPS 0
BCPS 1
BCPS 2
BCPS 3
BCPS 4
BCPS 5
BCPS 6
BCPS 7
Trigger Data Lines (TDL)
FVSBI
Fast Trig.
Fast Clear
BCPC
Fastbus
To DAQ R/O
BCT Trigger

Fig. 12



① $U = Z_{B1} - Z_{B2} - 1$

② $Z_V = \frac{Z_{B1} - Z_{B2}}{X_{B1} - X_{B2}} (X_V - X_{B1}) + Z_{B1}$

③ $V = Z_V - Z_{V1} - 1$

④ $Z' = (Z - Z_V) \frac{X_{V6}}{X}$

Fig. 13

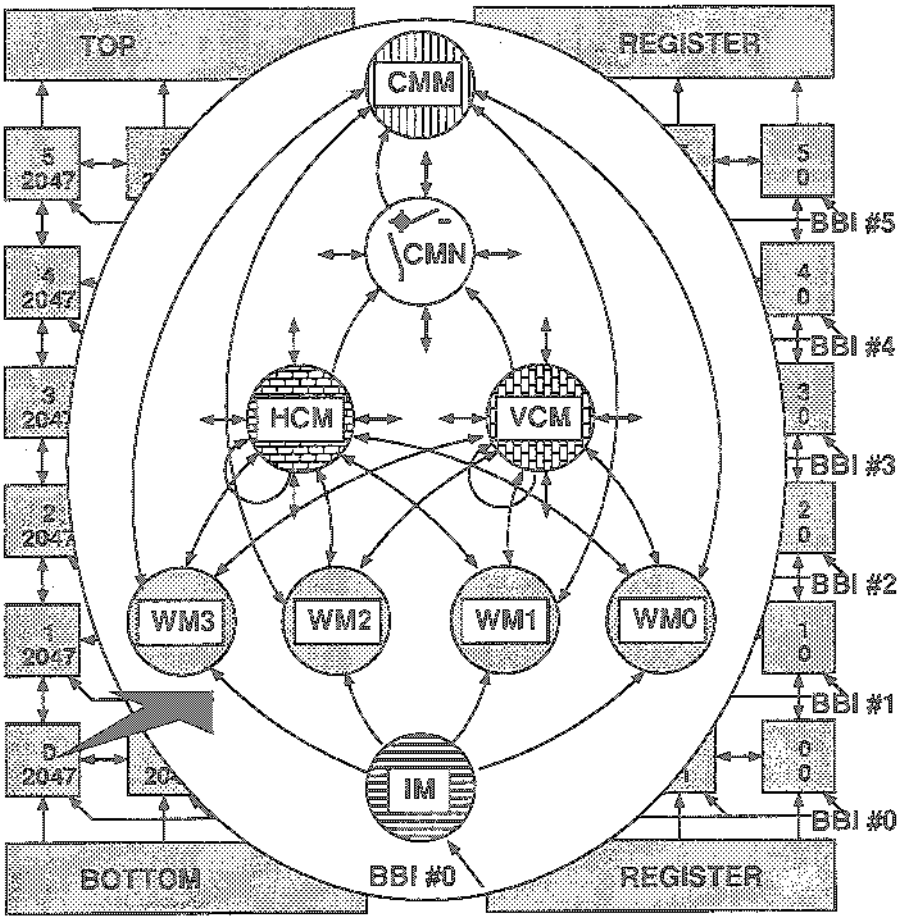

TOP
REGISTER
CMM
CMN
HCM
VCM
WM3
WM2
WM1
WM0
IM
BBI #5
BBI #4
BBI #3
BBI #2
BBI #1
BBI #0
BOTTOM
BBI #0
REGISTER
IM: Image Memory
WMx: Working Memory
HCM: Hor. Conn. Memory
VCM: Vert. Conn. Memory
CMN : Contiguity Mask Netw.
CMM: Contiguity Mask Mem.
BBI: BCPS Broadcast In Bus

Fig. 14

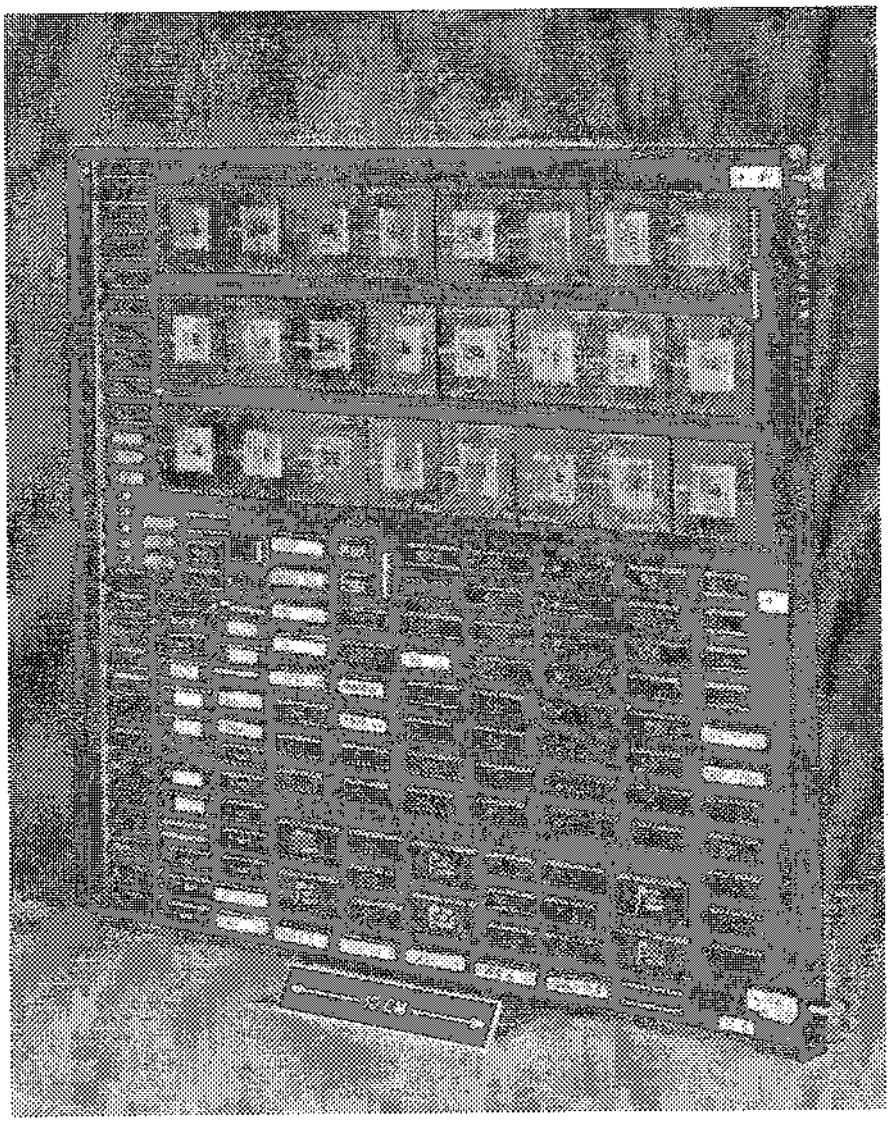


Fig. 15

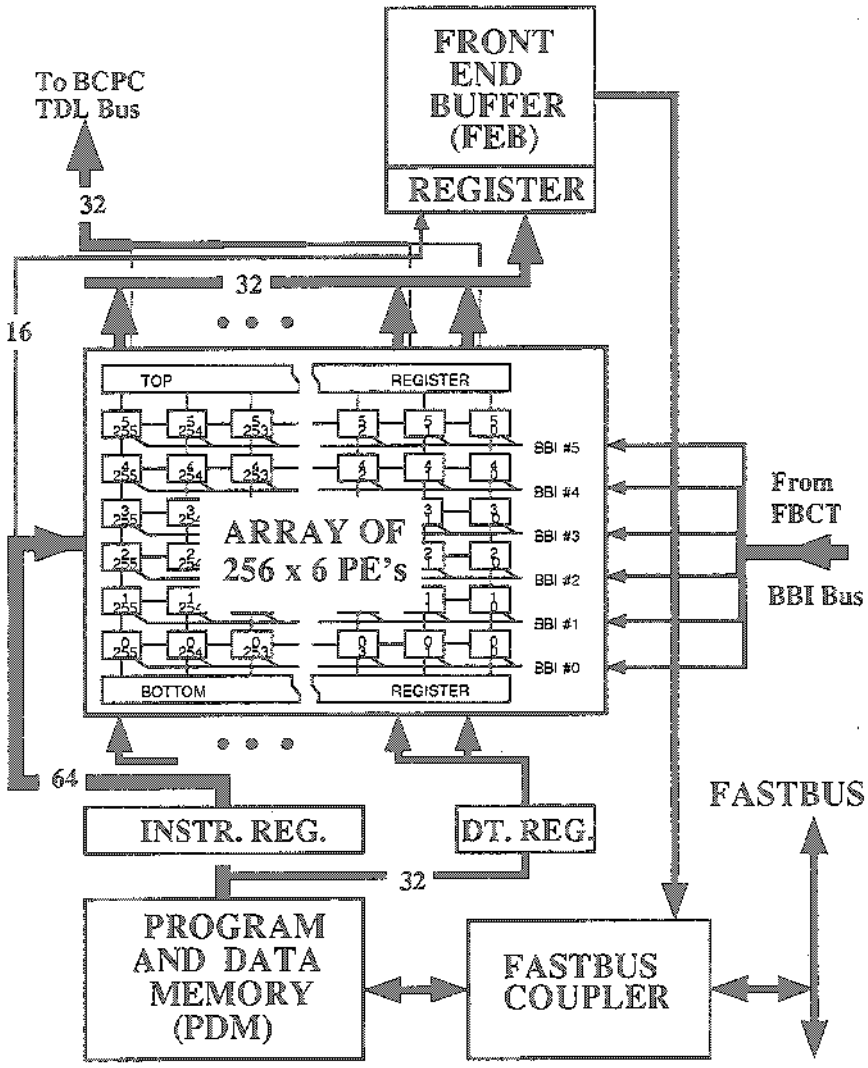

FRONT
END
BUFFER
(FEB)
REGISTER
To BCPC
TDL Bus
32
32
16
TOP
REGISTER
ARRAY OF
256 x 6 PE's
BBI #5
BBI #4
BBI #3
BBI #2
BBI #1
BBI #0
BOTTOM
REGISTER
From
FBCT
BBI Bus
64
INSTR. REG.
DT. REG.
32
FASTBUS
PROGRAM
AND DATA
MEMORY
(PDM)
FASTBUS
COUPLER

Fig. 16

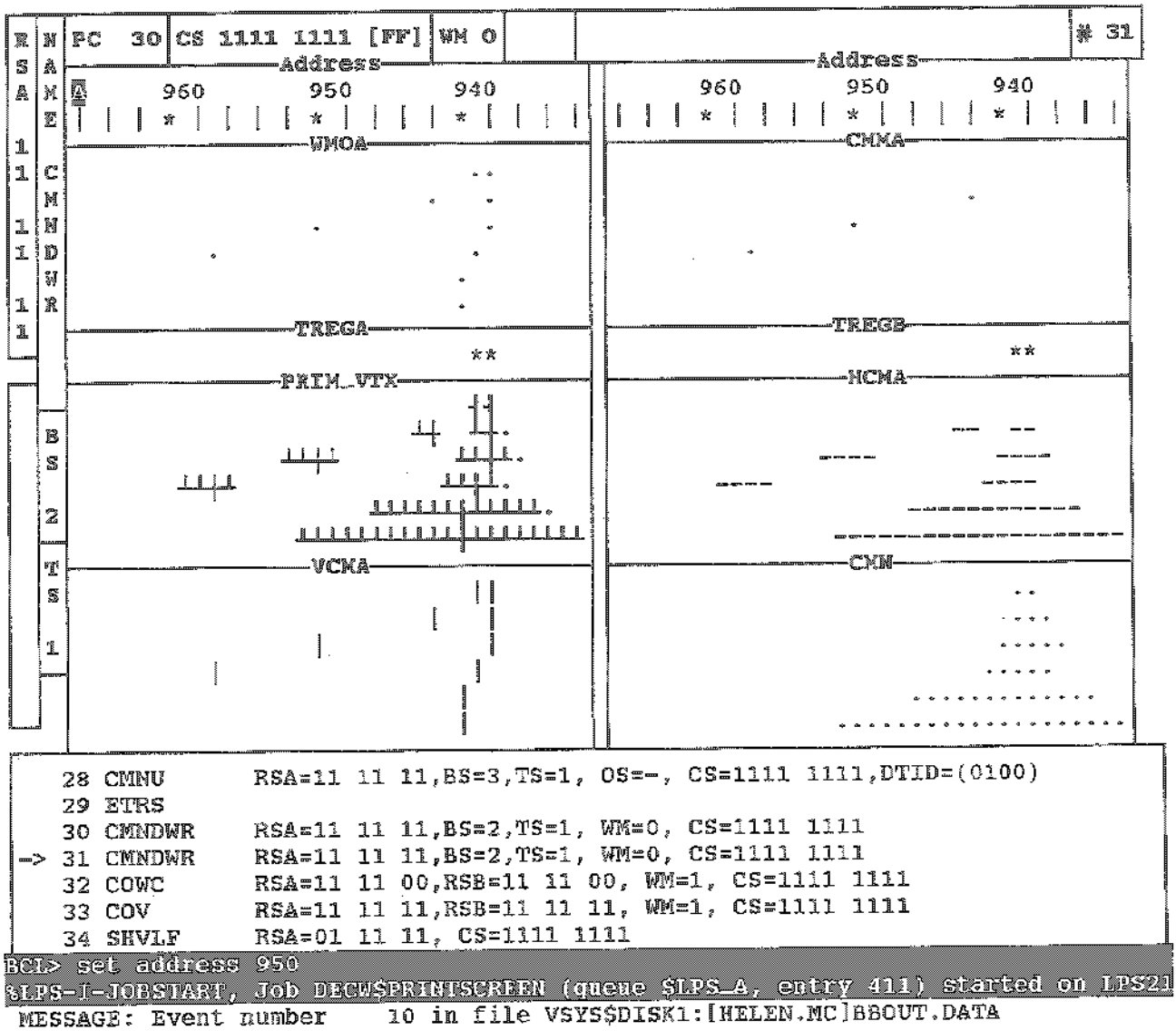

RSA 11 11 11
NAME
PC 30
CS 1111 1111 [FF]
WM 0
31
Address
960
950
940
WMOA
CMMA
CMNDWR
TREGA
TREGB
PRIM_VTX
NCMA
BS 2
TS 1
VCMA
CMM
28 CMNU RSA=11 11 11,BS=3,TS=1, OS=-, CS=1111 1111,DTID=(0100)
29 ETRS
30 CMNDWR RSA=11 11 11,BS=2,TS=1, WM=0, CS=1111 1111
-> 31 CMNDWR RSA=11 11 11,BS=2,TS=1, WM=0, CS=1111 1111
32 COWC RSA=11 11 00,RSB=11 11 00, WM=1, CS=1111 1111
33 COV RSA=11 11 11,RSB=11 11 11, WM=1, CS=1111 1111
34 SHVLF RSA=01 11 11, CS=1111 1111
BCL> set address 950
%LPS-I-JOBSTART, Job DECW$PRINTSCREEN (queue $LPS_A, entry 411) started on LPS21
MESSAGE: Event number 10 in file VSYS$DISK1:[HELEN.MC]BBOUT.DATA

Fig. 17

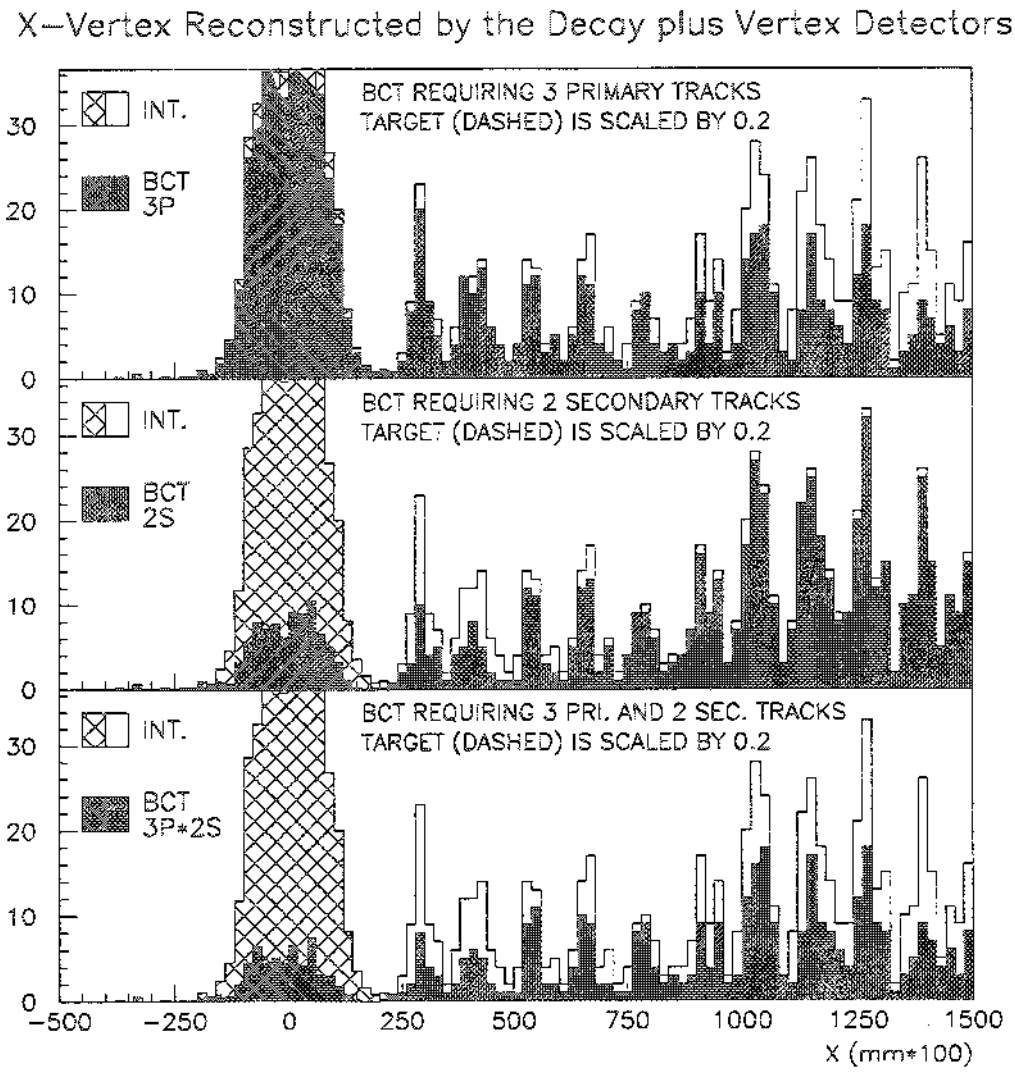

X–Vertex Reconstructed by the Decay plus Vertex Detectors
INT.
BCT 3P
BCT REQUIRING 3 PRIMARY TRACKS
TARGET (DASHED) IS SCALED BY 0.2
INT.
BCT 2S
BCT REQUIRING 2 SECONDARY TRACKS
TARGET (DASHED) IS SCALED BY 0.2
INT.
BCT 3P*2S
BCT REQUIRING 3 PRI. AND 2 SEC. TRACKS
TARGET (DASHED) IS SCALED BY 0.2
0
10
20
30
−500
−250
0
250
500
750
1000
1250
1500
X (mm*100)

Fig. 18