ISTITUTO NAZIONALE DI FISICA NUCLEARE

Sezione di Genova

INFN/AE-89/9
25 Settembre 1989

G. Dardo, L. Rossi:

FAST TRIGGER FOR BEAUTY HADROPRODUCTION STUDY

Presented to the 4th Pisa Meeting on
Advanced Detectors
Isola D'Elba - La Bidola (May 21-25, 1989)

and

To be published on NIM

Servizio Documentazione
dei Laboratori Nazionali di Frascati

INFN - Istituto Nazionale di Fisica Nucleare
Sezione di Genova

INFN/AE-89/9
25 Settembre 1989

Fast Trigger for Beauty Hadroproduction Study

G. Darbo and L. Rossi, INFN, Genoa, Italy

Abstract

A bidimensional pixel processor based on contiguity ideas has been designed. This processor, using data coming from a telescope of microstrip detectors, should allow the selection of events with evidence of secondary vertices in about $10\mu s$.

The algorithm, tested on experimental data, has shown a rejection factor of 200 still keeping a large fraction ($> 50\%$) of heavy quark events.

1. Introduction

Hadronic interactions are strong sources of heavy quarks$^{(*)}$. Charm and beauty factories can therefore be conceived in fixed target accelerators once high interaction rates can be tolerated and selective trigger algorithms for heavy quarks can be worked out.

Amongst the various trigger strategies employed to attack the heavy quark problem the most promising seems to be the selection of events with evidence of (at least) one secondary vertex.

The weak decays of charm and beauty happen with lifetimes of about $1ps$. The production and decay vertices are therefore distant enough to be separated using high accuracy track detectors like silicon microstrips [1]. This procedure, often used in data analysis, has been already successfully employed at the trigger level [2]. In this case only events with at least one track missing the primary vertex by more than $100\mu m$ were selected.

Aim of this paper is to show that more stringent requirements on secondary vertex existence can be made in a very short time ($\sim 10\mu s$) if a parallel processor architecture with distributed intelligence is used [3].

The paper is organized as follows: in section 2 we describe the algorithm, in section 3 we discuss its implementation and in section 4 we draw the conclusions.

2. The Trigger Algorithm

The trigger algorithm we propose is an evolution of the impact parameter trigger [2]. It may, therefore, be useful to briefly recall the principle of the impact parameter trigger. The layout, shown in figure 1, is made of a focusing telescope of microstrip detectors pointing to the center of a thin target. The tracks are reconstructed in the xz plane; those coming from primary interactions in the target

---

$^{(*)}$ One single burst of the CERN SPS accelerator can, for instance, produce 10^9 charm and 10^6 beauty pairs; i.e. more than what has been produced by existing e^+e^- machines in their lifetime.

give, in the appropriate reference system, equal hit addresses in all microstrip planes, while those coming from secondary vertices do not. It is then enough to erase all the hit addresses that are equal in at least two of the three microstrip detectors. The surviving hits, if any, are used to find potential straight tracks, defined by three points, and their impact parameter with respect to the vertex.

The event is accepted if any of the absolute values of the impact parameters (IP) lies within a preset interval of 0.1 to 1.0 $mm^{(*)}$.

This algorithm, executed in $\sim 350\mu s$ by a bit-slice programmable processor [4], enriches the charm sample by a factor of 15, as shown in figure 2 (from ref. [5]), still keeping about 50% of the charm signal. This proves that the detection of one track missing the primary vertex is a good criterion for the selection of heavy quarks.

It is interesting to observe that more stringent requirements on secondary vertices can be applied without appreciably losing charm signal. Demanding, for instance, a downstream crossing of two tracks with IP$> 75\mu m$ in the xz plane the sample shown in figure 2a is reduced only by a factor 0.9, while 1 interaction out of 200 satisfies the trigger. This translates into an enrichment factor of about 100 for charm and slightly more for beauty.

A downstream crossing trigger looks therefore quite appealing especially if one can execute the algorithm in a short time ($\sim 10\mu s$) in order to minimize the dead time losses.

The setup that produces the data on which the trigger processor should act is shown in figure 3. A set of six $25\mu m$ pitch microstrip detectors, placed behind a thin target, measures the coordinates of the tracks emerging from the interaction in the xz plane. Each detector has 2048 channels read out individually through an integrated digital chain. The vertex detector is not focusing by mechanical construction, but it will became focusing through appropriate coordinate transformation performed in the trigger processor, as we shall see below.

Let now describe how the trigger algorithm works. We call (z_v, x_v) the vertex coordinate and $(z_i^{(j)}, x_i)$ the hit coordinates in the vertex detector, where x_i ($i = 1, \cdots, 6$) is the position of the i^{th} downstream microstrip detector and $z_i^{(j)}$ the j^{th} point in this detector. A track in the xz plane is a straight line of equation

$$z = bx + z_v + h \tag{2.1}$$

where the z_v is the z-coordinate of the vertex and h is zero for primary vertex tracks, while for secondary tracks $IP \simeq |h|$. Let's now see the algorithm that in 4 steps will find 2 downstream crossing tracks.

step 1: coordinate transformation.

If we apply the following transformation to all the points $(z_i^{(j)}, x_i)$

$$z_i'^{(j)} = \frac{x_6}{x_i}(z_i^{(j)} - z_v) \tag{2.2}$$

(*) The impact parameter of a decay track from a particle with lifetime τ has an average value $\simeq c\tau$; i.e. $\sim 200\mu m$ for charmed particles and $\sim 350\mu m$ for beauty particles.

the (2.1) is transformed to

$$z = z_6^{(j)} - z_v + h\frac{x_6 - x}{x} = z_6'^{(j)} + h\frac{x_6 - x}{x} \tag{2.3}$$

The transformation (2.2) has the feature of transforming primary tracks (having $h = 0$) into constants and secondary tracks into hyperbolas (see example in figure 4).

step 2: erase primary vertex tracks.

From (2.3) a primary vertex track has 6 points having the same abscissae (we here neglect the inefficiency and the quantization noise due to the finite resolution of the detector), if we erase all of them the few points left can be used to look for secondary tracks (see figure 5).

step 3: shift and search for impact parameter tracks.

If we add to all the $z_i^{(j)}$ the amount

$$n_i(h)w \simeq h\frac{x_i - x_6}{x_i} \tag{2.4}$$

where w is the microstrip pitch and n_i is an integer, function of h, that better approximate the (2.4), the equation of a track having an IP= $|h|$ will became a constant function of x after this transformation. From the (2.4) results that each h defines a $n_i(h)$ (shift amount) in each of the 6 detector layers. If such a constant track has been found we can obtain its IP from the number of shift in the first detector (i.e. n_1).

step 4: downstream crossing.

The condition for downstream crossing of two IP tracks is defined by the following condition

$$(z_6^{(m)} > z_6^{(n)}) \wedge (h_m < h_n) \tag{2.5}$$

where $z_6^{(m)}$ and $z_6^{(n)}$ are the end points of the two downstream crossing tracks and h_m and h_n are their intercept in the target.

3. Algorithm Implementation Using a Contiguity Processor

Before going into the details of the algorithm implementation, we want to recall here what a contiguity processor (CP) is, and describe its most outstanding features. We therefore recall the main characteristics of the Delphi Contiguity Trigger [3].

The basic principle of the CP rely on the concept of Contiguity Mask Network (CMN). A rectangular mesh of switches (see figure 6) better explains the functional principle of a CMN than a full set of electronic diagrams. The small dots represent latches reset to zero, whereas the large dots are latches set to one. These latches are connected to the sensitive cells of a 2D detector, and the pulses from this detector will control the state of the latches (in the CP terminology this matrix of latches is called Image Memory – IM). In the example of figure 6 there are two tracks going through the detector from the bottom to the top, leaving behind some ionization, which finally set the corresponding latches. In the figure is shown a simulated event in a 60° sector of the Delphi Time Projection Chamber (TPC) as seen by the contiguity trigger in one of its IM's. The task of the trigger processor is to find out whether or not there were tracks in the detector. This is achieved with the help of vertical and horizontal switches between neighbouring cells. If, for each large dot, we close a few switches around it, according to a programmed rule (Contiguity Mask – CM), in case of tracks crossing the whole detector a connective path from bottom to top is created. If a voltage is applied to the bottom row of vertical switches, the same voltage is detected on the top row of nodes if there was any full track in the detector. This is mainly what the Delphi contiguity trigger does but, once we have a fast clustering network as the CMN, other functions can be considered as, for instance, the clearing of the image points (this will be used by the "Beauty Contiguity Trigger" for erasing primary vertex tracks).

Let us see some other features of a CP. Inside a CP there are two more memories, the Vertical (VCM) and the Horizontal Connection Memory (HCM). These two memories store the status of all the switches (either open or closed); if a bit is set in $VCM_{i,j}$ then the vertical switch connecting nodes $(i-1, j)$ and (i, j) of two adjacent rows is closed, if instead, $HCM_{i,j}$ is set the the horizontal switch connecting nodes (i, j) and $(i, j+1)$ is closed (i runs from bottom to top and and j from right to left). IM, HCM and VCM have all the same size and together with the "switches" in the CMN they are organized in a bidimensional array of Processing Elements (PE)

The CP we are going to design for the secondary vertex trigger it will look very like the Delphi one, but the PE organization will reflect the microstrip detector and to each strip will correspond a PE for a total number of 6×2048 PE's (see figure 7). More generally each PE is a very simple processor, which can communicate with each other and with the external world by 3 kind of communication busses and an instruction bus:

- *Broadcast Bus (B–Bus)*

  B–Bus is a bus that ties together all the PE's on a row (2048) and is used mainly for processor initialization and for event loading from the detector. It has relatively low bandwidth because one PE at the time can be addressed, but it is a multibit bus.

- *Local Synchronous Bus (LS–Bus)*

  Each PE is interconnected with each of its 4–neighbours by a few bit bus and information is shipped by all enabled PE's to one or more of its neighbours. If data must be transmitted from two not directly interconnected PE's this is done in few steps (the number of steps is proportional to the PE's distance). Data are transmitted synchronously with the system clock and 50 ns are needed to move informations from two contiguous PE's. This bus is mainly used for generating the contiguity masks or for global event transformation.

- *Local Asynchronous Bus (LA–Bus)*

  LA–Bus is as well as LS–Bus a 4–neighbours interconnection limited to a very few bits. LA–Bus is the skeleton of the CMN, and each PE can deal with it by opening and closing its own two switches (the horizontal and the vertical one). Signals on this bus ripple through the PE's not driven by the clock and with a delay of about 1 ns/PE. This bus structure is used to clusterize interconnected patterns (tracks).

- *Instruction Bus (I–Bus)*

All PE's are driven by a single clock and they execute the same instruction stream, making the CP a Single Instruction Multiple Data machine (SIMD).

Up to now we have described the algorithm that in 4 steps looks for secondary vertex. Now, we have to see how to implement it into a CP, making the best use of its architecture and of the 3 levels of data busses.

- *step 1:* Event acquisition.

  Each time an interaction inside the target is detected by a first level trigger, the trigger microstrips start up the readout. First the coded address (z_v) of the fired beam microstrip is transferred to the trigger and stored into a register. The 6 downstream detectors are read in parallel, and their hit addresses are sent sequentially to the CP. For each point in a detector layer the expression (2.2) is computed and z' is put to the corresponding B–Bus, PE's recognizing their logical address will set the to "1" its IM bit. Acquisition phase could take $4\mu s$ if we consider $10 \div 15$ hits/plane.

- *step 2:* Erase primary vertex tracks.

  After the transformation (2.2), tracks from primary vertex have almost the same address in the 6 rows of PE's (quantization error makes them a little staggered). Appropriate contiguity masks are therefore generated and all the points inside clusters making connective paths from row 1 to 6 are erased (primary tracks). Mask are generated all in parallel in a few clock cycles using the HCM, VCM and the LS–Bus. Eventually CMN and LA–Bus are used once to erase in parallel all the points inside the IM which have generated connections between the two extreme rows. Time for this step is well under $1\mu s$ being about $50ns$ times the number of links of the programmed mask plus $100ns$ for signals rippling through the CMN and erasing the points from the IM. The time needed by this step does not depend on the number of tracks or points, but only slightly on the mask size.

- *step 3:* Search for Impact Parameter Tracks.

  Once points belonging to primary tracks are erased, we want to reconstruct IP tracks. If we shift along a row the IM by a different amount n_j ($j = 1,..,6$) as given in (2.4) then a track with IP$= |h_n|$ will be aligned. Again a suitable contiguity mask will find it. All the tracks having the same h_n are found in parallel, whereas different h_n ranges must be tryed sequentially. If we want to search tracks having $100\mu m \leq$IP$\leq 1mm$ we must perform a maximum of 200 shifts in both directions. Duplicating HCM and VCM (for shifting in both directions at the same time, and thus searching positive and negative h_n in parallel), shifting contiguity masks instead of IM (we have to generate the mask only once and not each time we want to test an IP) and finally reducing the minimum shift step to two PE's, the total number of clock cycles is of the order 100 (i.e. $5\mu s$) The rippling through CMN can be made in pipeline with the shift operation and this will slightly increase the processing time ($\sim 1\mu s$ more).

- *step 4:* Find downstream crossings.

  For this last step we use two more lines in the LA–Bus of the upper row of PE's (row mapped to the most downstream detector in the trigger) that we can rename Daisy Chain Left (DCL) in which signals propagate from right-to-left and Daisy Chain Right (DCR) from left-to-right (see figure 8). Initially all PE's gate on the two daisy chains. When a PE's in the former step (step 3) detects an IP track it gates off the two daisy chains; DCL takes value

0 for all the PE's on its left, whereas DCR takes value 0 on the right hand side. Moreover a positive IP track is accepted if DCL=1 and negative IP's if DCR=1. Because IP tracks are found in the step 3 ordered by increasing IP and step 4 is in pipeline with the former, we have a very fast selection of downstream crossing with virtually no more processing time adding up.

Table 1 gives a comparison between the Delphi and the "Beauty" CP's. For the Beauty CP we foresee an higher integration and this will make it smaller in size (a few fastbus boards) and then faster. We are investigating possible layout of processing elements inside semicustom chips (gate array) and a good tradeoff between chip complexity and system performances seems to occur for 64 PE's in each integrated circuit. This would bring the chip to a complexity of about 15.000 equivalent gates, 192 of such circuits are needed for the whole trigger.

4. Conclusions

The design of a bidimensional pixel processor for triggering on events with secondary vertices has been described.

The selectivity ($5 \cdot 10^{-3}$ and the enrichment factor (10^2) of the trigger has been measured on real data. The speed of the processor ($10\mu s$/event) has been estimated extrapolating data from an existing contiguity processor architecture.

The characteristics of the processor should allow the design of a fixed target experiment aiming at the study of beauty hadroproduction.

5. REFERENCES

[1] E.H.M. Heijne et al., Nucl. Instr. and Meth. 178(1980)331.
B.Hyams et al., Nucl. Instr. and Meth. 205(1983)99.

[2] J.F. Baland et al., Nucl. Phys. B (proc. suppl.) 1B(1988)303.

[3] G. Darbo and B.W. Heck, IEEE Trans. Nucl. Sci. 34,1 (Feb 1987) 227-231.
G. Darbo and B.W. Heck, The Contiguity Processor. A SIMD architecture for a 2^{nd} level track trigger, Proceedings "International Conference on the Impact of Digital Microelectronics and Microprocessors on Particle Phisics" held in Trieste 18-30 March 1988, World Scientific Singapore

[4] J. Anthonioz-Blanc et al., MICE, a fast user programmable emulator of the PDP-11, CERN-DD/80-14

[5] L. Rossi, paper presented to the IX Int. Conf. on Physics in Collision, Jerusalem, Israel, June 19-21 1989 (to appear in the proceedings).

Table 1

Comparison between the Delphi and Beauty CP's

	Delphi	Beauty
Trigger decision (μs)	28	10
Clock frequency (MHz)	(*)10	20
Processing elements	4.608	12.288
Number of custom IC's	288	192
Number of PE's/chip	16 (4 × 4)	64 (2 × 32)
Number of gates/chip	3.600	15.000

(*) The real frequency has been found to be in excess of 16 MHz.

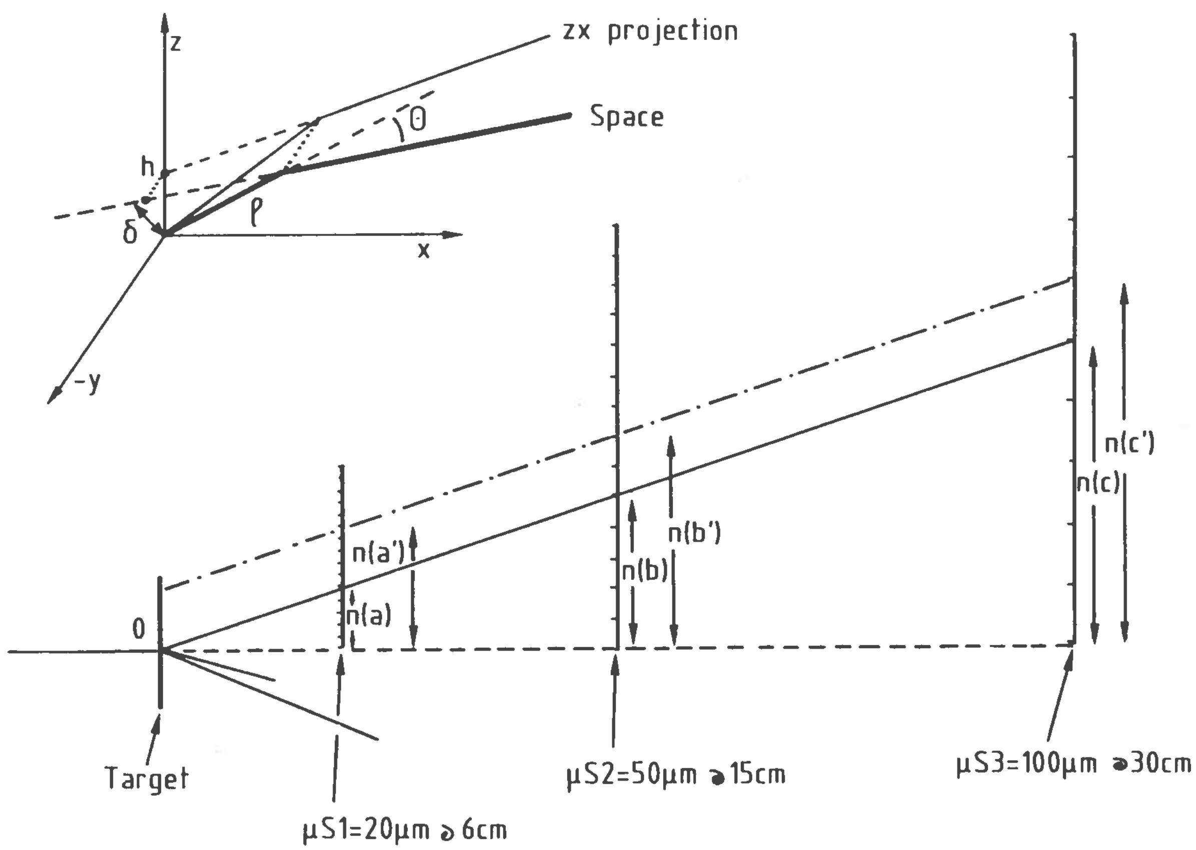


Fig. 1: The focusing telescope used for the WA82 Impact Parameter trigger.

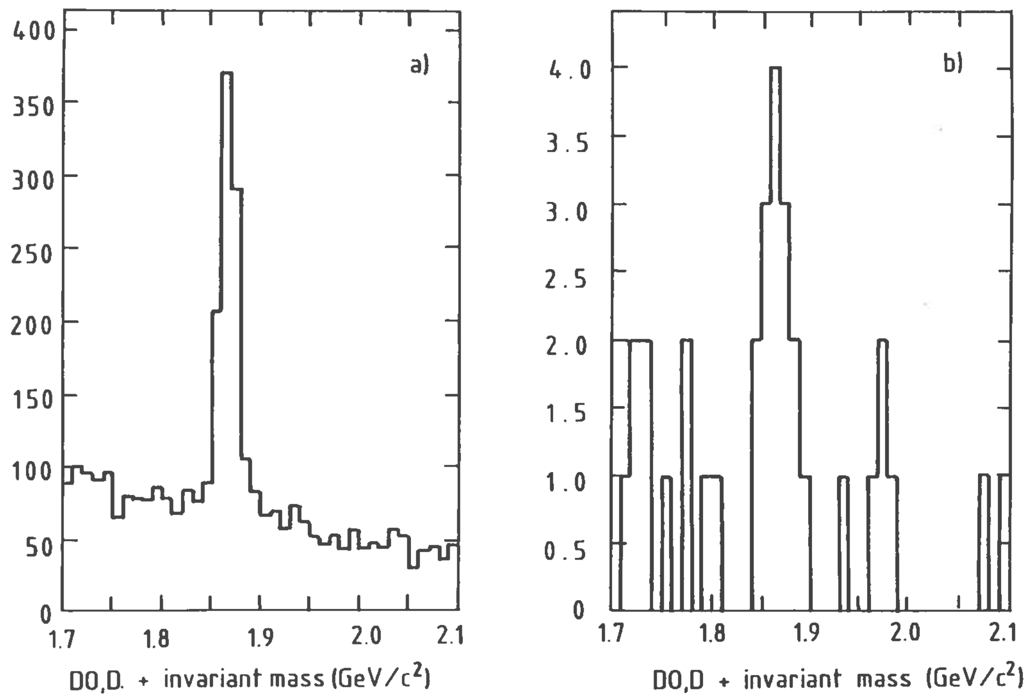


Fig. 2: The D-mesons invariant mass peak obtained from:
a) $7.5 \cdot 10^6$ events selected through the impact parameter trigger,
b) $1.5 \cdot 10^6$ interaction events.
Both samples have been reconstructed by the same program.

DETECTORS: 25μm pitch, 2000 pists, squared.

Telescope characteristics:
- High redundancy (B-events multiplicity)
- High resolution ($\sigma_{IP} \leq 10\mu m$)
- Good 2-track resolution ($\sim 10^{-4}$ rad)
- High trigger granularity (1.2×10^4 strips)

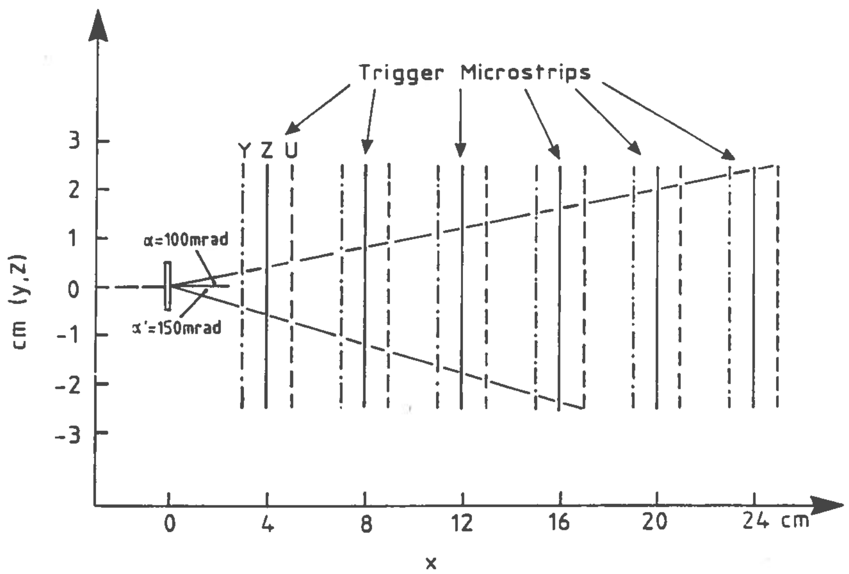


Fig. 3: Setup of the microstrip detectors to be used by the downstream crossing trigger. Six layers of z-measuring detectors spaced by 4 cm are placed starting from a distance of 4 cm from a thin target.

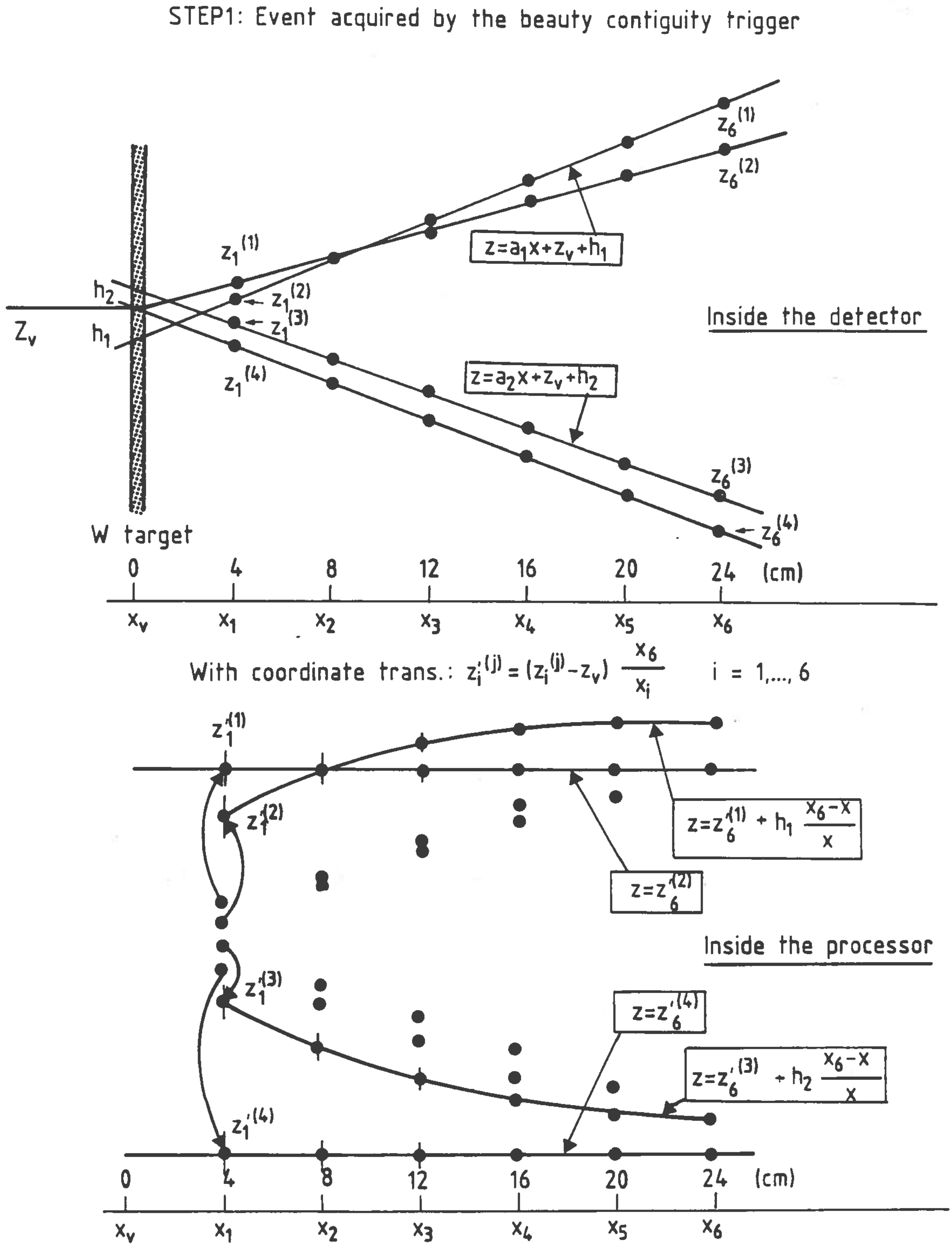


Fig. 4: (a) Example of an event with two primary tracks and two secondary vertex tracks. (b) The same event after the transformation (2.2) (see text). The transformation (2.2) (see text) has the feature of transforming the two primary tracks into constants and secondary tracks into hyperbolas (b).

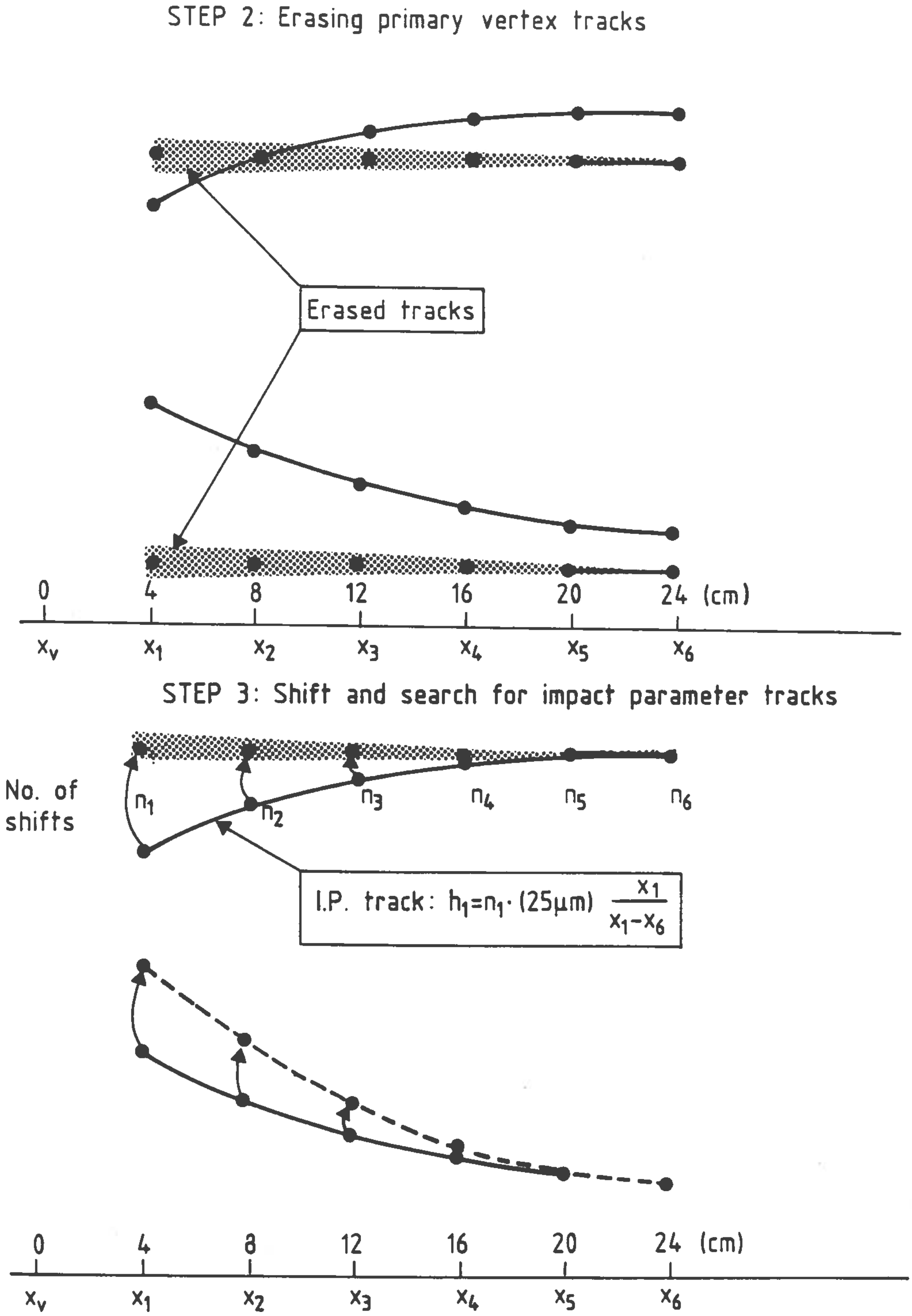


Fig. 5: The two primary tracks of figure 4 are erased because they are constants (a). Then, the tracks having impact parameter are found shifting the z as defined by the (2.4) (see text) (b).

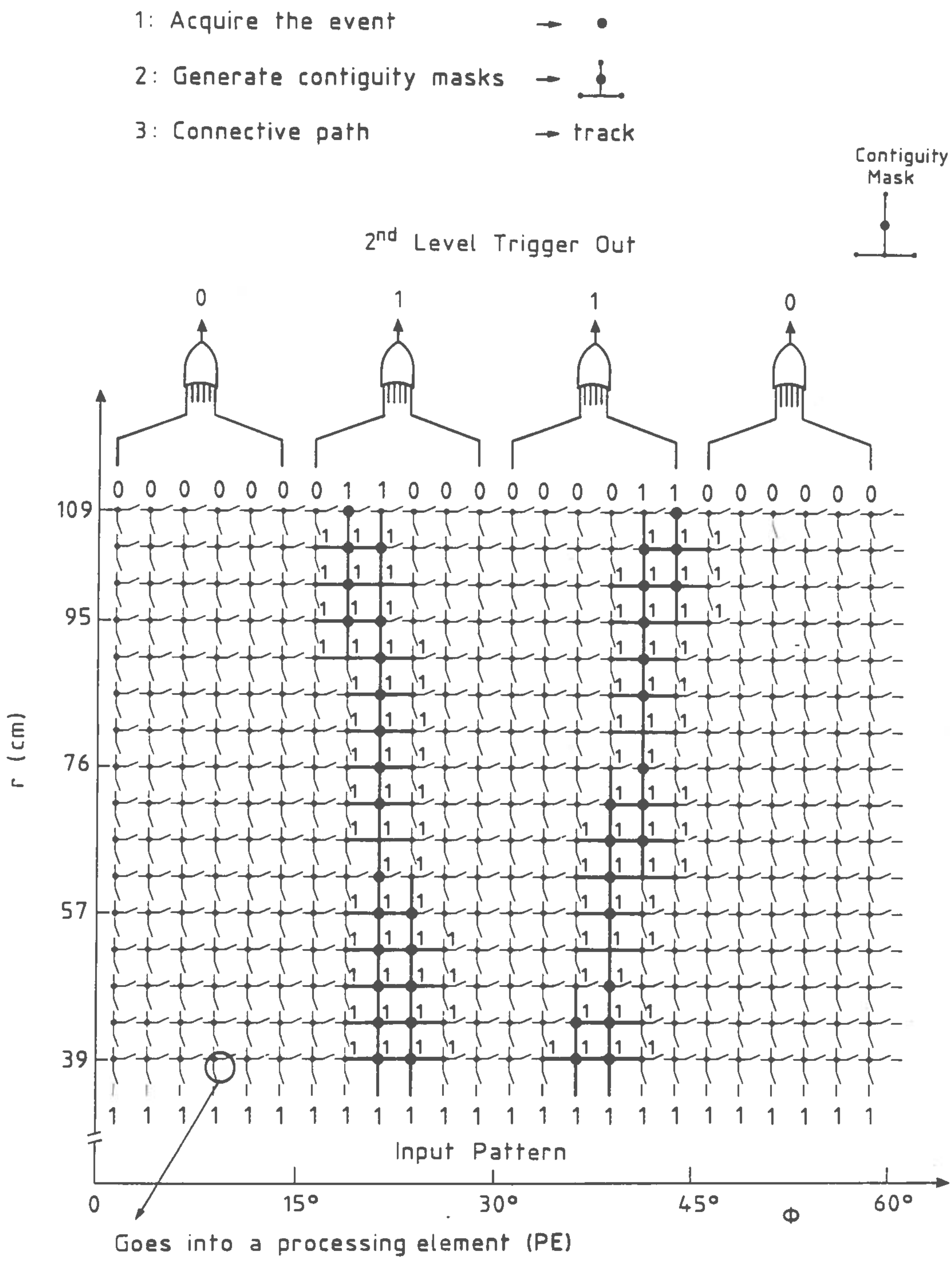


Fig. 6: The contiguity mask network represented by a mesh of switches. Two tracks with their connective path as can be seen in the Delphi contiguity trigger.

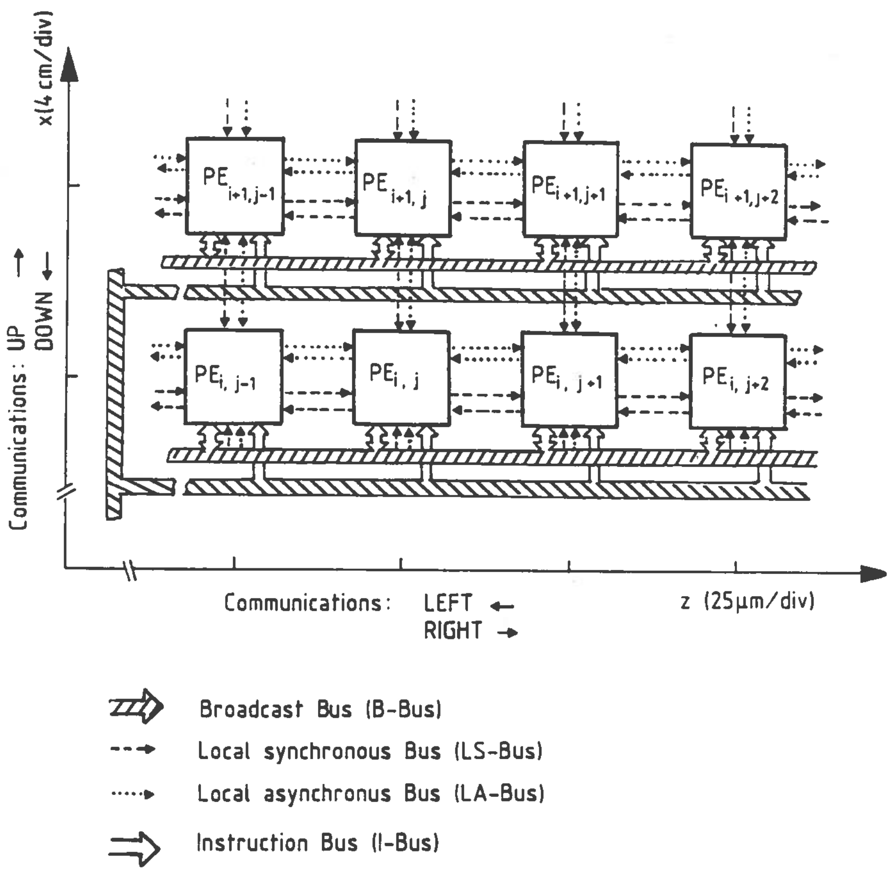


Fig. 7: The architecture of the proposed "beauty" contiguity processor. Each row has 2048 PE's and there are 6 rows in total. The topmost row of PE's corresponds to the last downstream microstrip detector in the trigger.

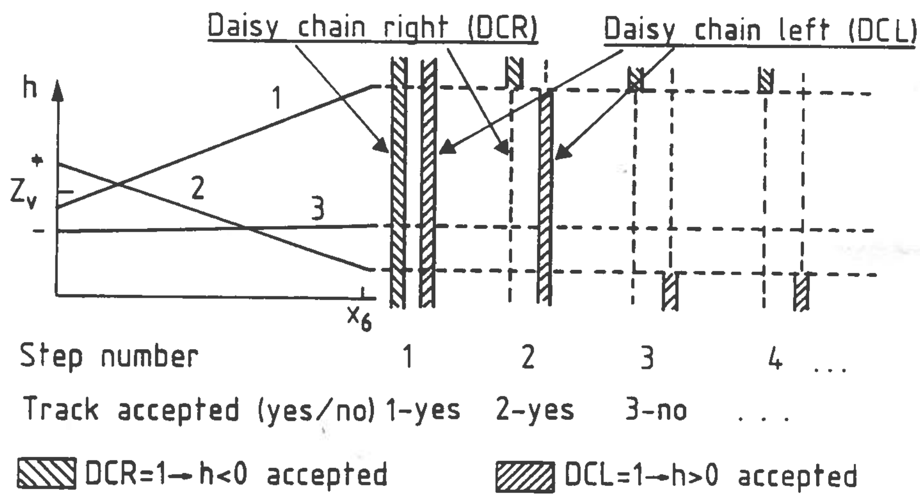
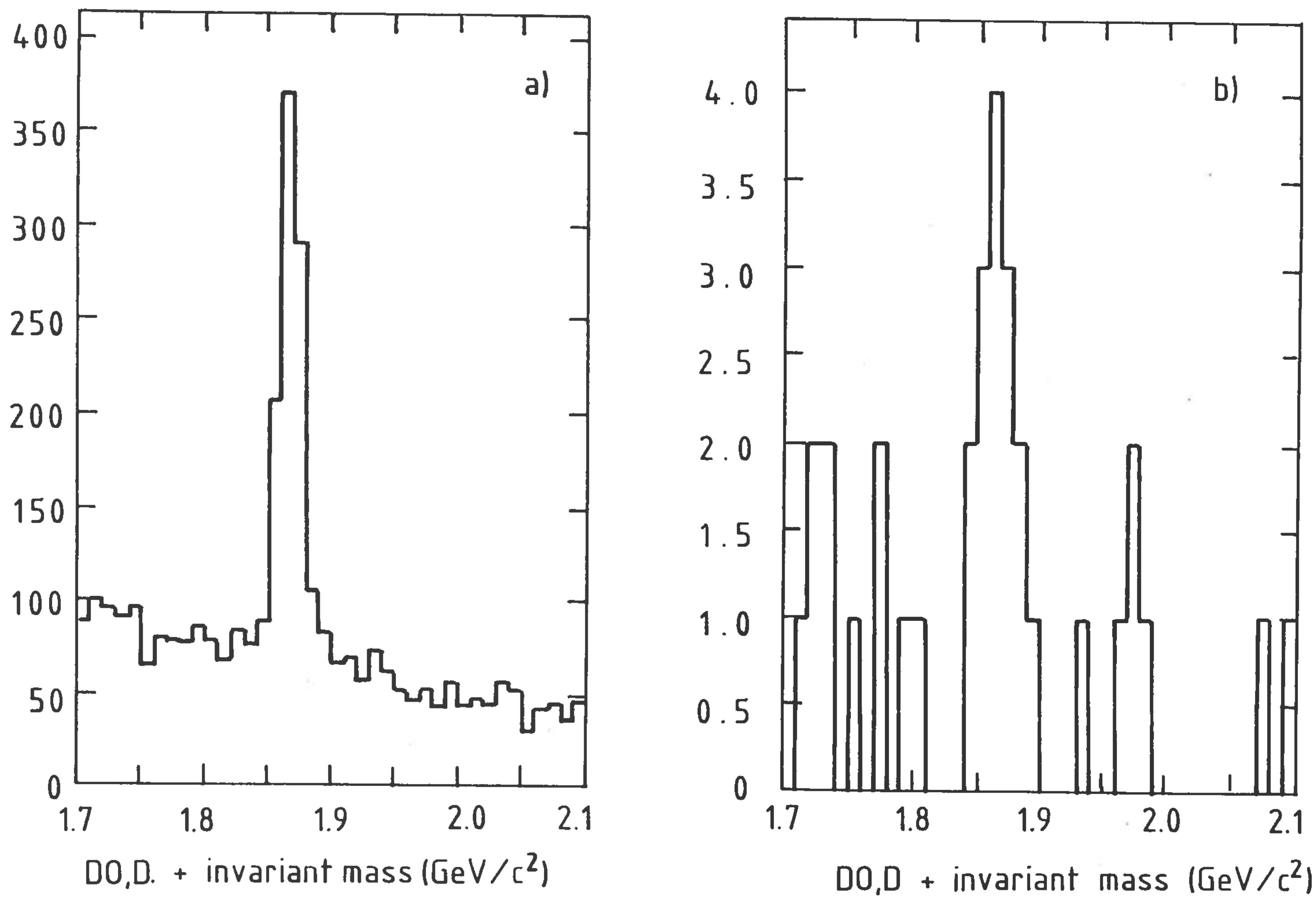


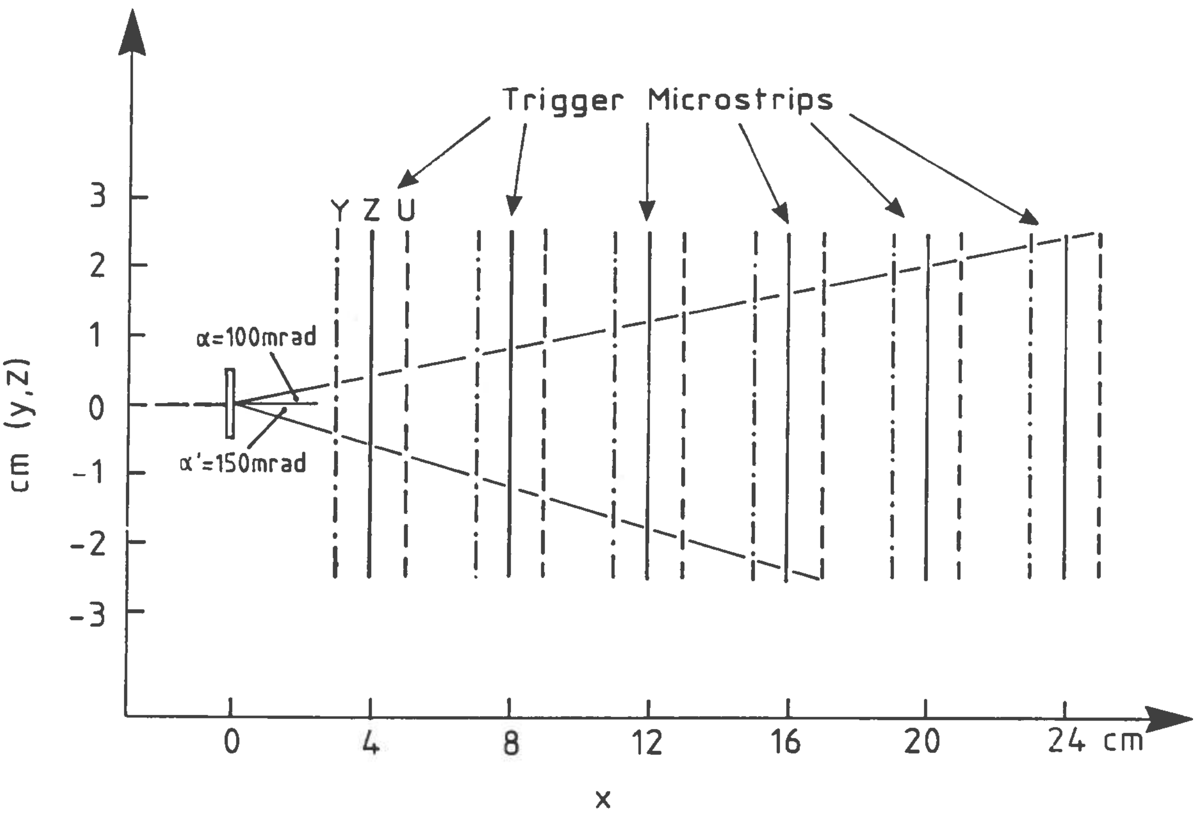
Fig. 8: Finding of downstream crossing tracks: secondary tracks are sequentially found by increasing IP. (see text for details).


a)
400
350
300
250
200
150
100
50
0
1.7
1.8
1.9
2.0
2.1
DO,D. + invariant mass (GeV/c2)
b)
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
1.7
1.8
1.9
2.0
2.1
DO,D + invariant mass (GeV/c2)

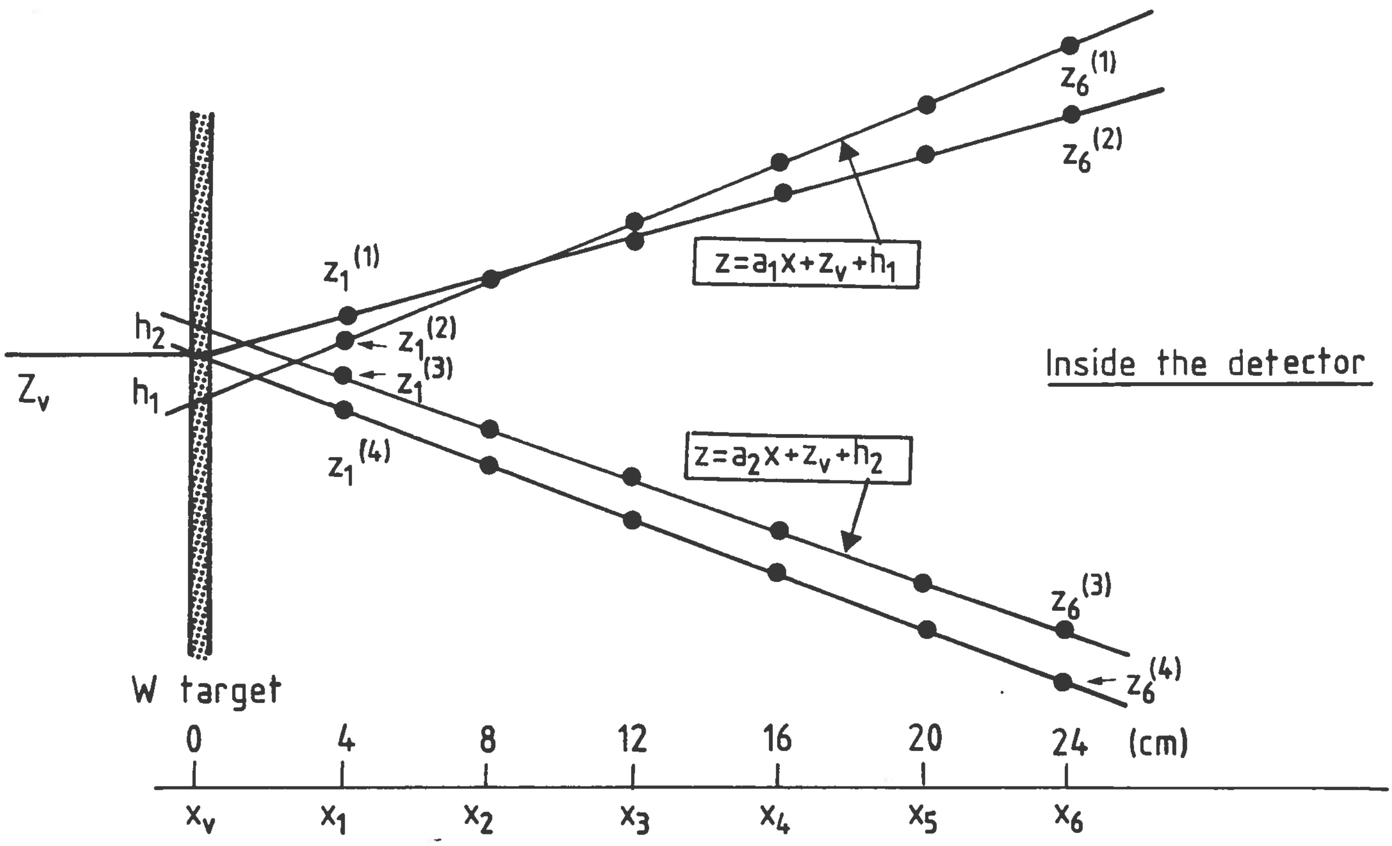
DETECTORS: 25μm pitch, 2000 pists, squared.

Telescope characteristics:

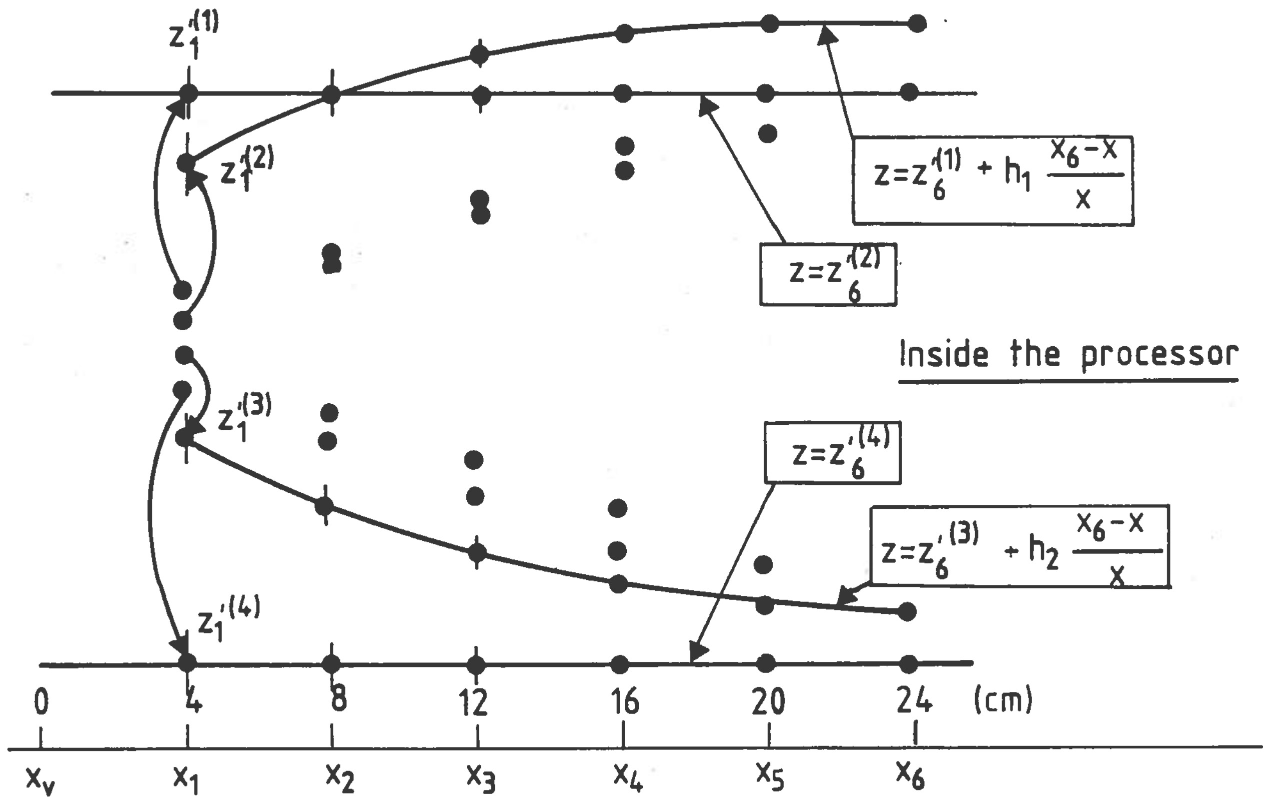
- High redundancy (B-events multiplicity)
- High resolution ($\sigma_{IP} \lesssim 10 \mu m$)
- Good 2-track resolution ($\sim 10^{-4}$ rad)
- High trigger granularity (1.2×10^{4} strips)



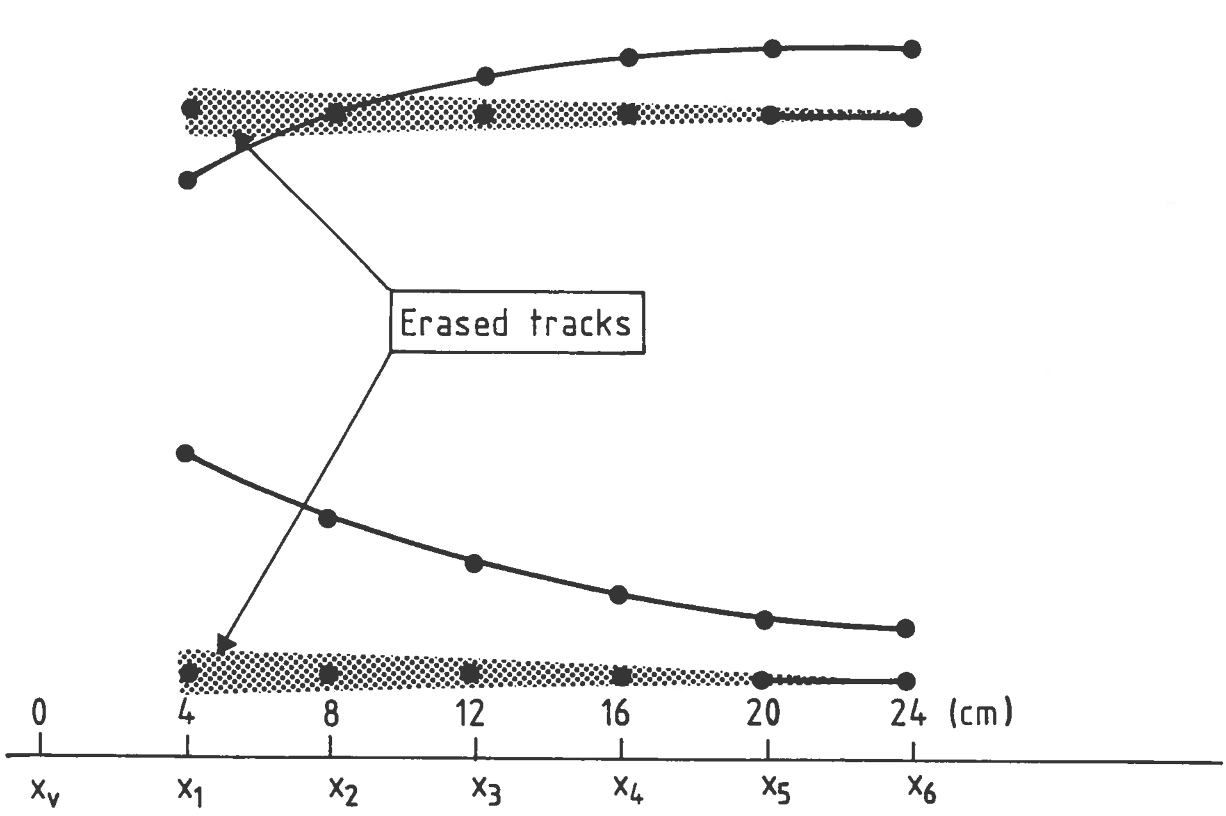
STEP1: Event acquired by the beauty contiguity trigger



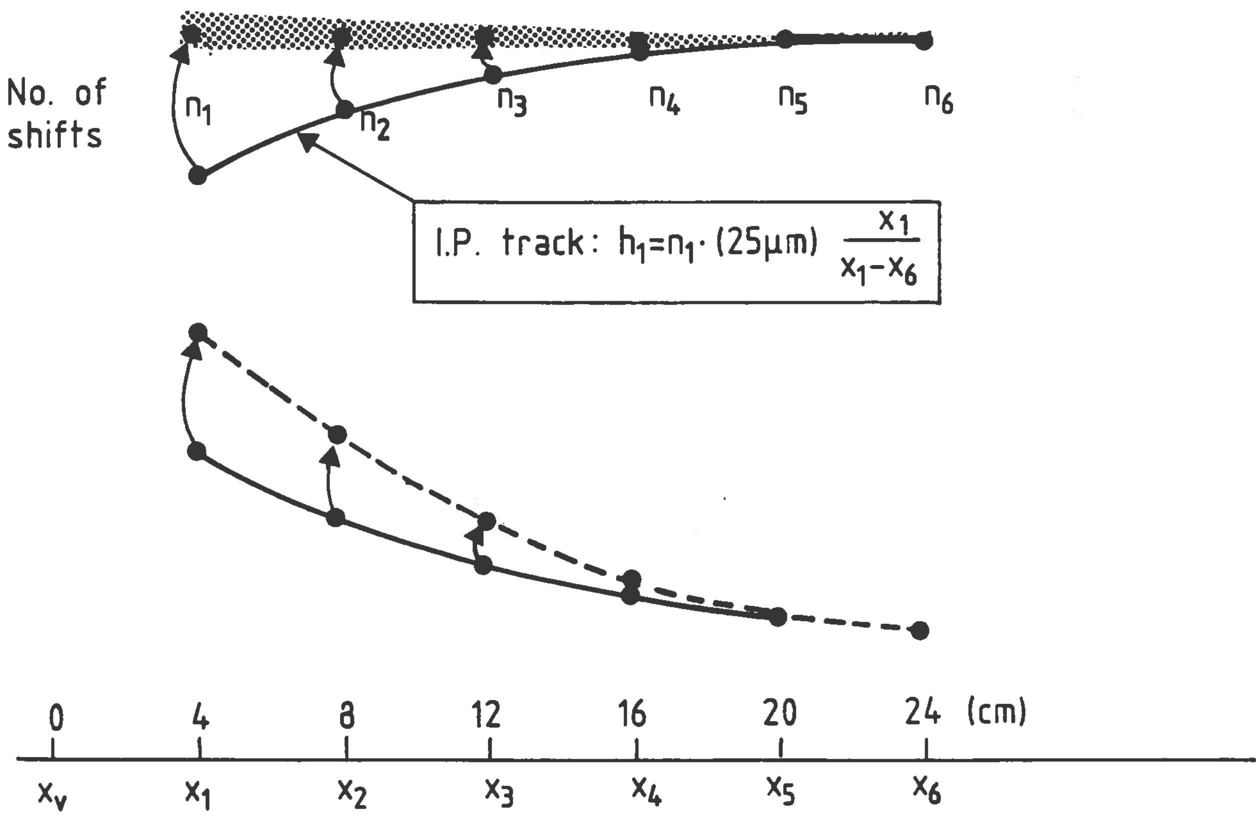
With coordinate trans.: $z_i'^{(j)} = (z_i^{(j)} - z_v)\ \dfrac{x_6}{x_i}$ $\quad i = 1, \ldots, 6$



STEP 2: Erasing primary vertex tracks



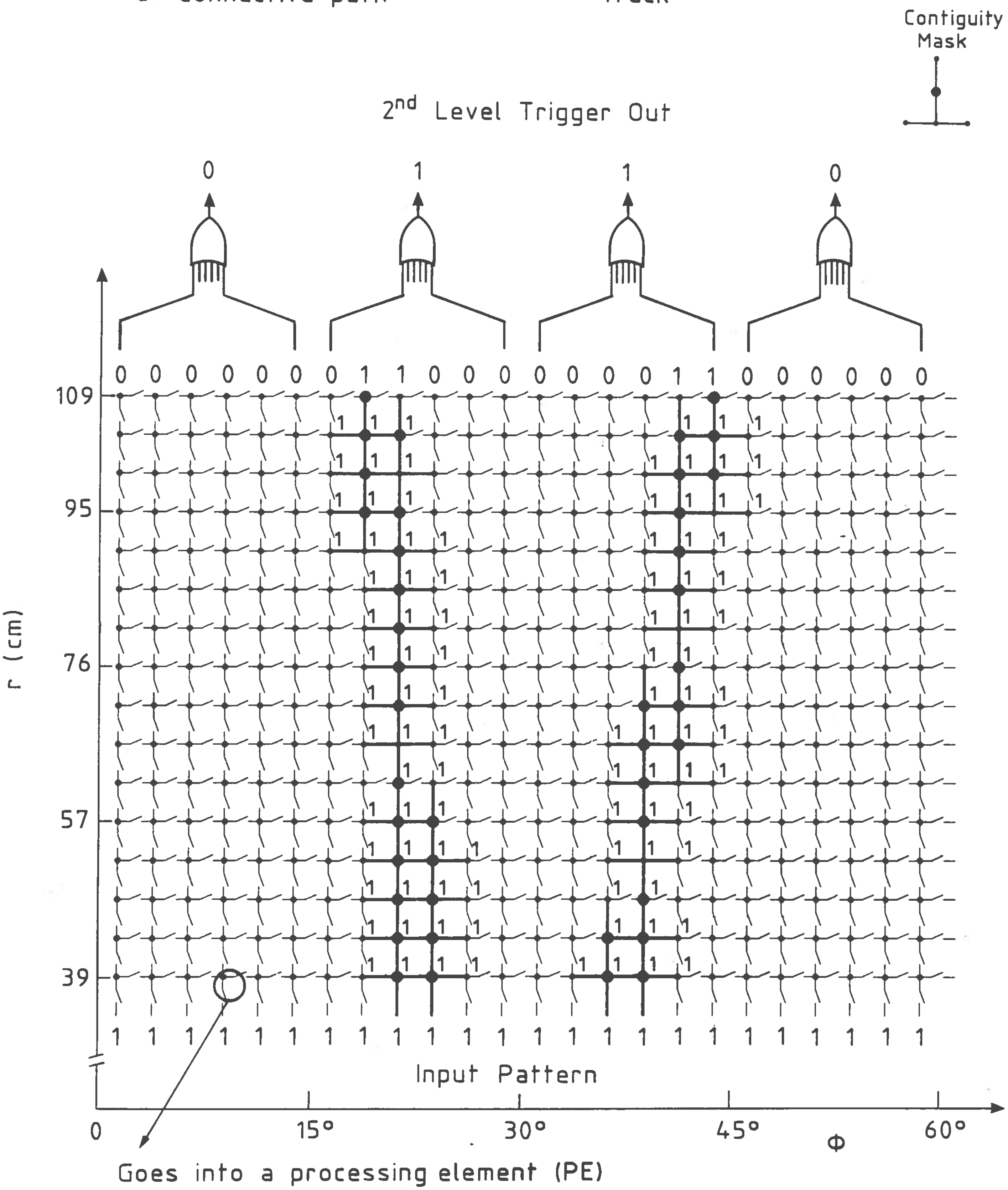
STEP 3: Shift and search for impact parameter tracks

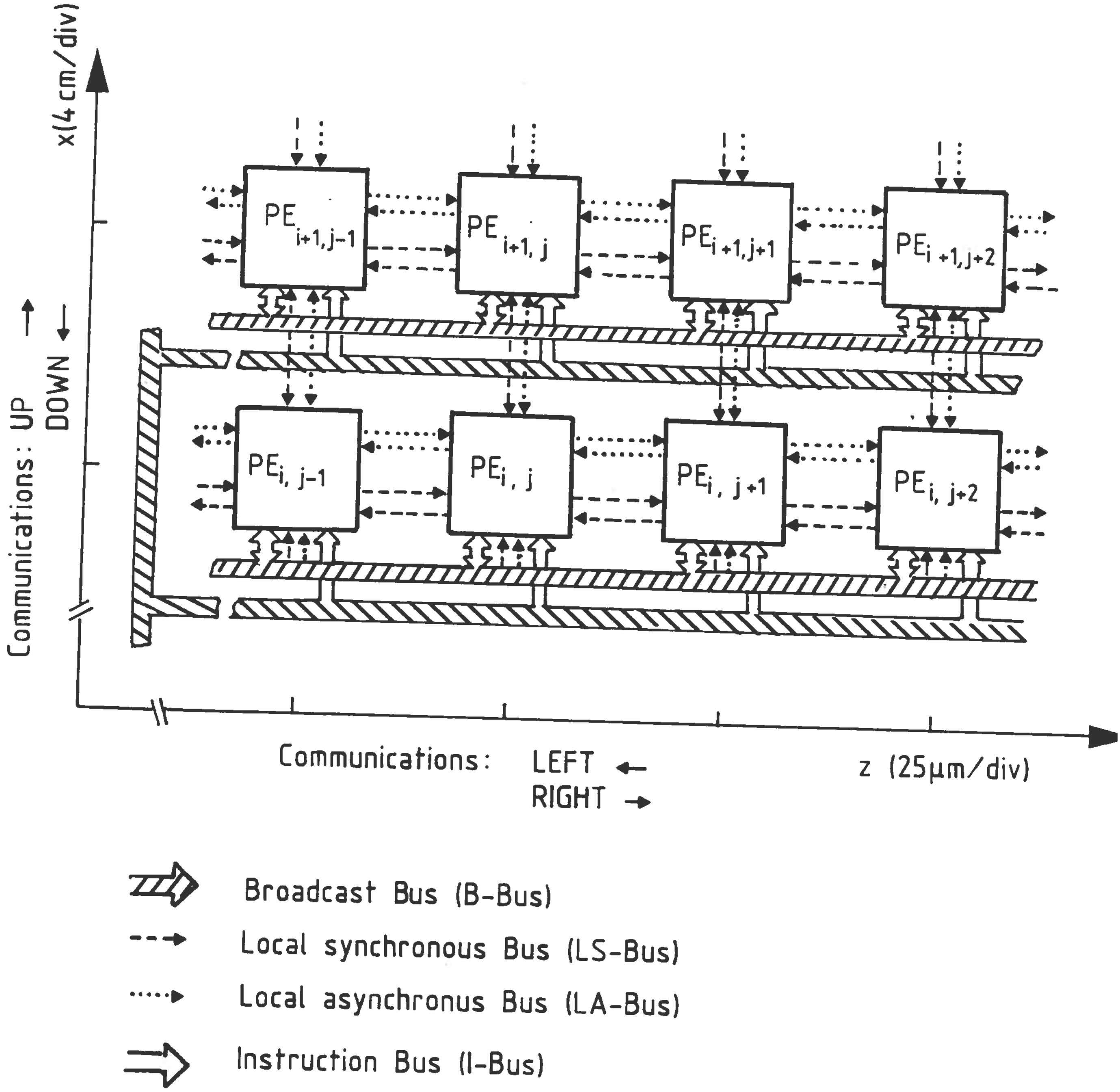


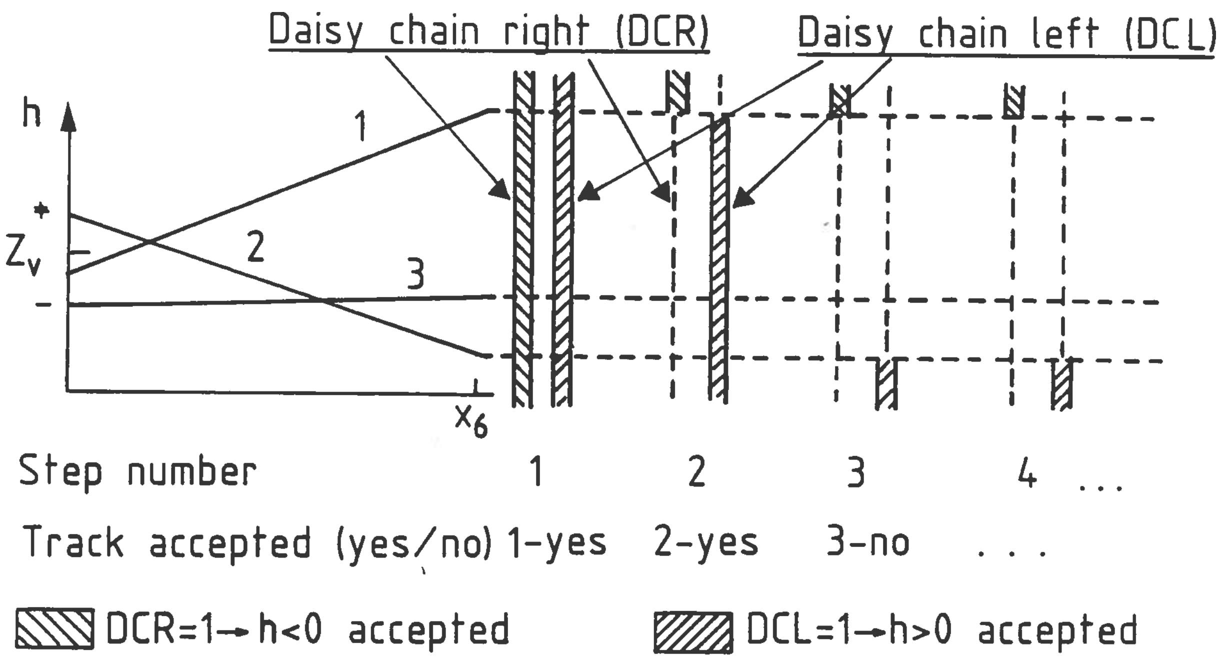
1: Acquire the event → •

2: Generate contiguity masks →

3: Connective path → track




x(4 cm/div)
PE i+1, j-1
PE i+1, j
PE i+1, j+1
PE i+1, j+2
PE i, j-1
PE i, j
PE i, j+1
PE i, j+2
Communications: UP ↑
DOWN ↓
Communications: LEFT ←
RIGHT →
z (25µm/div)
Broadcast Bus (B-Bus)
Local synchronous Bus (LS-Bus)
Local asynchronous Bus (LA-Bus)
Instruction Bus (I-Bus)


Daisy chain right (DCR)
Daisy chain left (DCL)
h
1
2
3
Z_v
+
−
x_6
Step number 1 2 3 4 ...
Track accepted (yes/no) 1-yes 2-yes 3-no ...
DCR=1→ h<0 accepted
DCL=1→h>0 accepted