The Contiguity Trigger for the Delphi TPC

G. Darbo ¹⁾ and B.W. Heck ¹⁾

Abstract

We describe here the Contiguity Processor which will be used as main 2nd level track trigger device for the Delphi experiment. It searches for tracks of events in 3 dimensions in less than 10 μ s using "Contiguity Mask" algorithms. A description of the principles of the method, of the processor architecture and simulation results are given.

CERN LIBRARIES, GENEVA

CM-P00061170

Presented at the 2nd Topical Seminar on Perspectives for
Experimental Apparatus at Future High-Energy Machines
5-9 May 1986, San Miniato, Italy

¹⁾ EF Division, CERN, Geneva

1. Introduction

The Delphi experiment has a big cylindrical Time Projection Chamber (TPC) of 2.8 m diameter and 2.8 m length. It is made of two cylinders and each half has 6 sectors of 60° to cover the 360° ϕ angle. The electron clouds generated by charged tracks drift to two opposite directions in the two detector halves and are detected by 16 circular pad rows and 192 sense wires sitting on the chamber end-plates. The use of a Contiguity Processor (CP) [1] [2] as main 2nd level track trigger for the Delphi experiment has been proposed in '83 [3] [4] [5]. The CP uses data coming from the pads of the TPC and employs a powerful pattern recognition algorithm to reconstruct tracks in 3 dimensions. With a drift time of $25\ \mu\text{s}$ and a CP processing time in the range of $5 \div 8\ \mu\text{s}$ the trigger fits well in the scheduled $35 \div 40\ \mu\text{s}$ for the Delphi 2nd level trigger.

This processor implements "Contiguity Mask" algorithms [1] [2] in hardware, uses an array of semicustom integrated circuits, specifically developed for this application, and covers the TPC barrel region. The forward TPC region is covered by a first level trigger based on signals from TPC wires. It reconstructs short pieces of tracks (10 cm) during a drift time of $1.6\ \mu\text{s}$.

2. A quasi-3D Contiguity Processor

2.1 Signals from TPC

The 1680 pads on each of the 12 sectors of the TPC produce analog signals, which will be amplified, shaped, sampled in time and finally digitized in 8 bit ADCs before being used for the complete reconstruction of the event in the TPC. For the trigger system however a reduced number of 384 signals per sector will be used. They will be generated by grouping pad signals together, in groups of different sizes. On pad rows with a small radius only a few pads are used together, while there will be up to 6 in one group of a row with a large radius (see fig 1c). This selection is done in order to generate groups of constant ϕ angle of 2.5° , which is independent of the pad row radius. In this way each pad row will be divided into 144 groups.

The second dimension for the signals of the trigger system is given by the radius r of the pad row. There are 16 rows on each of the two endplates of the TPC.

As the third dimension for the 3D trigger system we have chosen the angle θ , because all tracks with reasonably high transverse momentum are in first approximation straight in the rz projection, i.e. parallel to the θ direction. In other words, if we divide the trigger region of $35^\circ \leq \theta \leq 145^\circ$ into 10 similar bins of "thickness" $\Delta\theta = 11^\circ$ (see figure 1b), then every track will fall completely into one of these bins, unless it is close to the border. In this latter case a small overlap in θ of the neighbouring bins will guarantee that we do not lose the track even then. In order to sort the data into these bins, we will do a time-sampling on the signals from the pad groups, which is a simple function of the drift time and pad row radius. This function can be stored in PROM and the sampling will be done by the CP.

The described procedure can be summarized as a mapping of the cylindrical TPC volume onto a three-dimensional rectangular matrix with 144 bins in ϕ , 16 bins in r and 10 bins in θ . The CP then stores all the information of an event in its "Image Memories" (IM), one for each θ slice with 144×16 cells. Processing itself takes place after the drift time, based on data from one IM at a time. A search will be done for a row of contiguous hits deposited in the IM which corresponds to a track coming from the vertex. The CP looks for such patterns of tracks, locates them precisely and counts the number of them. The information from this process will be used for the final trigger decision later on.

2.2 Contiguity Processor architecture

In figure 2 we have a block diagram of the Contiguity Processor for one half of the TPC. A complete CP is built up by using 2 different modules, which will be described now:

1. Contiguity Processor Slice (CPS):

In each CPS we have the following blocks:

- *Contiguity Array (CA)*: It is the CP central processing unit, a bidimensional geometric processing unit, having 16 rows of 24 pixels each. It consists of a matrix of semi-custom integrated circuits (Gate Arrays) specifically developed for the aim. A complete description of the CA is given in the next chapter.
- *Switch Yard Logic (SYL)*: It is divided in two parts: one inside the CA and one outside. The outside part is basically a memory of 256 32-bit words and a 10 MHz counter. Inside each CA row has its SYL connected with two bits of the outside memory. When the counter is clocked we get out of the memory 32 trains of pulses that are used inside the CA in the following way. Let us consider only one CA row and two bits of the memory (bit 1 and bit 2). Starting from reset condition a pulse coming from bit 1 enables the first IM for the signals coming from the TPC. If a second pulse arrives on bit 1 the second IM is enabled too, and so on up to the 5 IMs. If we get however a pulse on bit 2, the first IM is disabled. A second pulse disables the second IM, and so on, as for bit 1. With this mechanism we can get θ -slices of the event as discussed in paragraph 2.1 and also the necessary overlap of θ -slices.
- *Program and Data Memory Slice (PDMS)*: This memory is loaded via Fastbus. It provides instructions and data for the trigger processing and finally also for the readout of the trigger data.
- *Trigger and Fastbus Readout Interface (TFRI)*: It interfaces the CA output with Fastbus and the central 2nd level trigger logic. It has also functions of event front-end buffer and implements simple logic decisions like zero skipping.

2. Contiguity Processor Controller (CPC):

In the CPC we have the following blocks.

- *Program Sequencer and Controller (PSC)*: it is a 12-bit address counter and program controller. It generates addresses for the Controller Program Memory (CPM) and the 6 PDMSs.
- *Controller Program Memory (CPM)*: This memory is addressed in parallel with the PDMSs and keeps in it the part of instruction word used for jump addresses, feedback control bits for the CPC and general flags for CP working states. Its 32-bit words together with 6 times 32-bit words from the PDMSs form the complete CP instruction word of 224 bits.

The following busses and external connections link the CP's modules together:

- *Fastbus*: Both CPS and CPC modules have a slave Fastbus port. It is used to load program and data into the PDMSs and CPM, to read the processed event from the TFRI and to read/write CSR registers.
- *Contiguity Processor Bus (CPB)*: It connects the CPC with the 6 CPSs and carries the 12 address lines coming from the PSC in CPC plus some other control lines.
- *TPC Signal Input (TSI)*: The pad discriminator outputs, after being grouped, are brought into the CPSs. A bundle of 384 lines (a complete TPC sector) goes to one CPS module through three auxiliary Fastbus connectors.

- *Internal and External Trigger Data lines (ITD and ETD)*: Each CPS provides a 4-bit pattern in which each bit indicates whether or not a track was found in a 15° region in ϕ . 5 of such patterns are sequentially generated by the CPS and transmitted over ITD lines, one for each of the 5 θ -regions. The CPC has, instead, some control and timing lines (ETD); these, using a "fast protocol" notify the CP of Beam Cross Over (BCO), first and second level trigger decision, readout phase and so on. All ETD lines are connected with a "Trigger Supervisor" which takes the 2nd level trigger decision, after having correlated TPC information with the other detectors.
- *Contiguity Array Carry lines (CAC)*: These are optional connections to connect adjacent CPS modules of the CA matrix. If these connections are done the CP works on a big 16×144 matrix, instead of 6 individual 16×24 matrices. Each module has 32 lines to connect to the left hand side module and another 32 for the right hand side. If wanted the first CPS module can be connected to the last one to process, without boundary effects, cylindrical images.

The processing of an event by the CP is divided in 4 different phases. An event can either pass the 4 steps or the system can be reinitialized at any point in between, and up to 500 μ s can be taken before a new event could come. Let us see the details of the 4 phases and the timings related to BCO. Using Delphi terminology:

- *Warning (WNG)*: starts 1 μ s before BCO
All memories inside the CA are reset and the SYL is initialized with this signal.
- *1st level trigger (LEV1)*: starts at BCO
SYL counter is clocked and event is acquired. If 1st level is negative (after 3 μ s) a new event starts with a WNG.
- *2nd level trigger (LEV2)*: starts at BCO as well
SYL counter keeps being clocked up to end of the drift time (25 μ s). SYL then is disabled and 2nd level trigger processing starts running. It lasts $5 \div 8 \mu$ s and at BCO + 35 μ s a 2nd level decision is taken. If it is negative we start again with WNG.
- *Front End Freeing (FEF)*: starts at BCO + 35 μ s
A longer program that will last $200 \div 300 \mu$ s runs in the CP. These few hundreds microseconds are envisaged to load the front-end buffer in the TFRI with further processed data that will help forthcoming trigger levels and the offline analysis.

2.3 The central processing unit: the Contiguity Array.

The Contiguity Array (CA) shown in figure 3 is organized in a 3-level structure. The 3 levels are:

- *Image Memory (IM)*:
- *Horizontal and Vertical Connection Memory (HCM and VCM)*:
- *Contiguity Mask Network (CMN)*:

All the blocks except the CMN are basically bidimensional memory structures organized around $m \times n$ flip-flop matrices. The CMN has no memory inside and, as we will see later on, is a combinatorial logic of $m \times n$ "nodes". These matrices are composed of 16 rows and 24 columns at Fastbus board level, but of 16×144 at CP level. Due to this regular and repetitive structure we implement a submatrix, of 2×4 nodes, in semicustom integrated circuits using CMOS Gate Array technology. 576 chips are required for the two CPs.

Before going into detailed explanations of the CA, helped by an analog equivalent of the CMN, we explain how the CA works in finding tracks.

Let us consider the net of switches in figure 4. The "big dots" represent a track as seen in IM. Each node of the net is connected to the four contiguous ones by a switch. To choose a Contiguity Mask means to close, for each big dot, some surrounding switches. For example, in figure 4 we have decided to close two vertical switches for each big dot (above and below) and two horizontal ones (to the left and the right, but one row further down). This "upside-down T" mask, applied to all the points of the event, generates a connective path from the first to the last row of the IM. If we apply now a signal to all of the bottom nodes and probe at the top ones we can find and count all such connective pathes. They give an indication about the existence and number of tracks passing through the detector. Since the Contiguity Masks can be programmed, one will later choose those ones, which optimize track acceptance and reduce background sensitivity best.

There are 5 IMs in the CA, each bit of an IM is directly connected to a TPC pad group. If the SYL enables a given row, of one or more IMs, the pulse coming from the detector sets the corresponding flip-flop(s) in the IM(s). The IMs are also organized as columns of shift registers in such a way that data coming into the bottom row can be shifted up; and by 16 clock cycles a complete IM can be filled with a given pattern or trigger data can be read out from the CA top row. More than that, each IM can be separately addressed and the output is bit-by-bit linked to the inputs of the HCM and VCM.

HCM and VCM store the on/off information of the switch net of figure 4.
 If flip-flop (i,j) of HCM is set the switch just on the right of node (i,j) is closed.
 If flip-flop (i,j) of VCM is set the switch just underneath node (i,j) is closed.
 Starting from the event pattern stored in one of the 5 IMs we close switches according to the chosen Contiguity Mask using the following instructions.

- *COH(x), COV(x)*: The addressed IM (x) is copied either to HCM (*COH(x)*) or to VCM (*COV(x)*).
- *SHHD, SHVD*: HCM (*SHHD*) or VCM (*SHVD*) is shifted down by one row.
- *SHHL, SHVL*: HCM (*SHHL*) or VCM (*SHVL*) is shifted left by one column.
- *SHHR, SHVR*: HCM (*SHHR*) or VCM (*SHVR*) is shifted right by one column.
- *SHHU, SHVU*: HCM (*SHHU*) or VCM (*SHVU*) is shifted up by one row.

Every shift operation can be a "shift only" or "shift with feedback". In the former case the contents of the HCM (VCM) is shifted out and not automatically restored in the source, while in the latter case a set bit in a node is ORed with the bit which is being shifted in.

In figure 5 we explain how we load the switch pattern for the "upside-down T" mask into the two connection memories with 5 instructions.

CMN is used after HCM and VCM have been worked out. At this time, CMN is equivalent to the switch network with closed switches around the track points.

If an input pattern is presented to row i of CMN, reading row j ($j > i$) we get the heads of the clusters connecting the two rows. The foreseen trigger algorithm search for connections between row 1 and row 16, but in the case of a whole pad row missing or a very high inefficiency, the corresponding row in CMN can be bypassed by software programming.

Since the program can define the CMN input pattern, the starting row and the readout end row, the processor is well suited for cluster finding and prefiltering of track pieces to be used in the forthcoming trigger levels and in the offline analysis.

Each instruction involving IMs, HCM and VCM is done in a single clock cycle (100 ns). Instructions using CMN need $400 \div 500 \mu s$ to get the signal rippling through the big combinatorial network. Due to the high parallelism of the CP, an IM can be treated in $1 \div 1.5 \mu s$ and the whole event in $5 \div 8 \mu s$.

3. Simulation of Contiguity Trigger efficiency.

We have done a very accurate simulation of the Contiguity Trigger, from event generation to trigger decision:

- *Event generation:* Either internal events with tracks having uniform outgoing angle and momentum distributions or more refined events like $e^+e^- \rightarrow qq$ events can be used as input sample.
- *TPC response:* Simulated signals on pads are in a very good agreement with those got from TPC prototype.
- *CT simulation:* Different pad grouping and Contiguity Masks have been tried.

We report here some of the results got out of this Montecarlo. For this simulation the 10° – regions cover from 35° to 145° and the geometric acceptance for straight tracks in Z has been settled at ± 2.5 cm.

In figure 6 we have the Z acceptance for single track events. No inefficiencies and TPC dead zones have been considered.

In figure 4 we had seen an event with an “upside-down T” mask, now in figure 7a we have used a “double link” mask whose result is the rejecting of low momentum tracks. The effect on momentum cut for three different masks is in figure 7b.

In the case we need a more selective trigger we can either ask for more than one track to trigger as in figure 8a or use a momentum-cut mask as in figure 8b.

As last result, in figure 9 the track counting provided by the trigger is shown. Tracks’ number is got counting the θ, ϕ bins (10×24) that are fired by a connective path crossing an IM. If the events are generated with vertex at $Z = 0$, even with a jet like structure, we have good correlation between number of generated and found tracks in the TPC barrel region (figure 9a). If the vertex is 50 cm upstream in the far most of the cases no-tracks are found and in only a very few cases more then 1 track is seen. (figure 9b).

4. Conclusions

In this paper we have described the hardware, which we are currently building, and some of the performances in track recognition and triggering. We now want to summarize the major features and advantages of this unusual algorithm and hardware.

The CT reconstructs tracks in 3 dimensions. Due to its high granularity it really sees tracks and not random combinations of dots giving a trigger. In addition the algorithm has an intrinsic high parallelism and can be implemented in an very regular and repetitive hardware structure. Finally beeing the system done with a relatively few components we expect to achieve high reliability.

The complete logic for 1 CA node has been realised as part of a Multi Project Chip (MPC). This MPC uses $3\ \mu\text{m}$ CMOS standard cells technology; the layout of our design, using $4.5\ \text{mm}^2$ of silicium, is shown in figure 10. This chip is now under test. Soon we will start the implementation of the final Gate Array.

Acknowledgments

We acknowledge gratefully the help of Prof. S. Vitale and Dr. L. Rossi during the early phases of this project. We are especially indebted to Dr. H. Wenninger for his interest in the development of the Delphi trigger system and in general for the support of our work at Cern. Drs. J.V. Allaby and H.-J. Hilke helped us in the planning and also by useful discussions. The help by the INVOMECE division of the Interuniversity Micro-Electronic Center at Leuven and especially from ir. H. Vanden Wijngaert, who supervised the design of the MPC, are acknowledged.

References

- [1] G. Darbo, maschere di contiguità e loro applicazione nella ricostruzione di tracce di particelle in tempo reale, thesis Università di Genova (1980).
- [2] G. Darbo, S. Vitale, A second stage trigger for track reconstruction based on contiguity ideas, Nucl. Instr. and Methods 190 (1981) 81–88
- [3] G. Darbo, B. Heck and L. Rossi, A three-dimensional ripple trigger for the DELPHI Time Projection Chamber, Paper presented at the Time Projection Chamber Workshop, Vancouver, B.C., Canada, June 1983 and AIP Conference Proceedings No. 108.
- [4] G. Darbo, B. Heck and L. Rossi, The Contiguity Processor for DELPHI, DELPHI–83–29 Elec – April 1983.
- [5] DELPHI Technical Report, DELPHI 83–66/1.

Figure Captions

- Fig. 1 In a) we have a $r\phi$ TPC view of an $e^+e^- \rightarrow q\bar{q}$ event, while in b) we have the rz projection of the same event in TPC sector 6. The dashed lines in b) show the 10 θ -regions in which the drift space is divided; high momentum tracks from the beam interaction point fall completely into one of them. Five internal pad rows with the chosen pad grouping are shown in c).
- Fig. 2 Block diagram of Contiguity Processor architecture. Only one out of the 6 CPS is shown together with the connections to the CPC.
- Fig. 3 Block diagram of the Contiguity Array.
- Fig. 4 Contiguity Trigger algorithm. The big dots represent the event of fig. 1, as seen by pad groups, in the 2nd θ -region of sector number 6. The signal applied to the bottom row of the mesh of switches propagates only to the top row for tracks crossing the TPC. The two tracks are detected by the trigger: two ORs produce "1" at their output.
- Fig. 5 Example of processing an Image Memory. The "upside-down T" mask has been chosen. The first 2 instructions close the vertical switches in VCM, the next 3 close the two horizontal switches, the last one finds the tracks. $1\mu s$ is needed for the full processing.
- Fig. 6 Vertex cut with single track events.
- Fig. 7 P_t cut with "double link" masks. In a) the 2 GeV/c track (on the right) is found using the "double link" mask, while the 1 GeV/c track does not bring the signal to the TPC top row. In b) the trigger acceptance versus track transverse momentum is shown for 3 different kinds of masks.
- Fig. 8 Vertex acceptance along the beam direction of high multiplicity events with a mask giving no transverse momentum cut (a) and for a mask giving transverse momentum cut (b). The three superimposed curves are for a trigger requirement of at least 1, 2 or 3 tracks.
- Fig. 9 Track counting in high multiplicity events. Above, events are generated coming from the interaction point ($V_z = 0$) and below from upstream ($V_z = 50$ cm).
- Fig.10 Layout of a Standard Cell design of one Contiguity Processor node. The chip has 20 pins and an area of 4.5 mm^2 .

Fig. 1

Fig. 2

Fig. 3

Fig. 4

Fig. 5

Fig. 6

Contiguity
Mask

$P_t = 1.0 \text{ GeV}/c$

$P_t = 2.0 \text{ GeV}/c$

Fig.7(a)

Fig. 7(b)

Fig. 8

Fig. 9

Fig. 10