CERN/EF 86-6
30 April 1986

A FAST AND VERSATILE WIRE CHAMBER SPACE POINT FINDING PROCESSOR

A. Casaca^(*) and A. Corre
CERN, Geneva, Switzerland

ABSTRACT

A four plane, four metre square, multiwire proportional counter at the CERN Omega spectrometer is used to digitize the spatial position of charged particles and generate a fast second level trigger. The coordinates of cluster centroids are fed into a modular space point finding processor which can analyse a ten particle event in 40 microseconds.

The detector and the algorithms are described in the first part of this paper, the hardware implementation and the performance are then discussed.

Submitted to Nuclear Instruments & Methods

(*) On leave of absence from Instituto Superior Técnico, Lisboa, Portugal

1. INTRODUCTION

This processor has been designed for use with a four plane, four metre square MWPC, one of the trigger detectors currently in operation at the CERN Omega Spectrometer. The short resolving time and the good multihit capabilities of MWPCs make them suitable for use in the trigger for high rate, fixed target experiments. Moreover, the overall cost of a MWPC system is significantly lower than that of a drift chamber.

This MWPC space point finding system differs from earlier designs [1-5] in that:

- i) most of the fast hit encoding logic is located on the detector frames;
- ii) the space point finding hardware is independent of the wire directions in each plane of the wire chamber module;

iii) the various processor blocks are implemented with standard MBNIM modules [6].

Ten particle events can be processed in 40 μ s, with a 10 μ s second level trigger dead time, thus allowing for:

- i) improved transverse momentum selection for the forward particles (e.g. experiment WA70 [7]);
- ii) on line identification of the particles by means of a ring finding logic associated with the RICH counter [8] which has been in operation at Omega since 1984. This ring finding logic will be either programmed into the MICE [9] event filtering processor in use at Omega or else hardwired in a FASTBUS system.

2. GEOMETRICAL AND COMBINATORIAL ASPECTS

The MWPC geometry is shown in Fig. 1. The planes are oriented in four distinct directions: Y, Z, U and V. Each plane has 1024 wires, whose numbering convention is shown in the figure. The angle α between planes (Y,V) and (Y,U) is 10.4° . An event typically contains n (~ 10) hits per plane. As a result of chamber inefficiency hits may be missing in any one of the four planes.

From the total number of coordinates given by the chamber readout system, the processor must find the n space points. The traditional algorithm to solve this problem analyses all the combinations of Y, Z, U, and V coordinates and selects those which correspond to a space point. The implementation of algorithm requires the execution of four loops, nested in three levels, which gives a computing time with an $\sim n^3$ dependence.

To minimize the computing time, the number of possible space points is reduced by two methods:

i) The small angle α in the chamber implies a significant reduction in the number of (Y_i, U_j) or (Y_i, V_k) pairs of coordinates which need to be considered in the search for space points.

ii) The coordinates are considered to be divided into sixteen different partitions, depending on the value of their four most significant bits. Flags in the processor indicate which partitions contain coordinates, and a pointer associated to each flag indicates the starting address of the partition in an input data RAM. When a certain coordinate value is processed during point finding, if the corresponding partition flag is "off" no search is made, and if it is "on" the partition is searched for the coordinate, all the coordinates in the other partitions are ignored. An example is given in Fig. 2.

In addition to optimizing the algorithm, high processing speed is achieved by using a fast circuit technology. A PROM based control unit and a multiple operation hardware structure have been specially adapted to exploit the parallelism of the algorithm.

3. THE POINT FINDER CONTROL ALGORITHM

A flow-chart of the point finder control algorithm is represented in Fig. 3. The coordinates from each plane are read out and stored in ascending order, in separate input buffer memories.

During execution of the algorithm, Y coordinates are sequentially read from the respective input buffer. A Y_i value is then combined with a valid U_j coordinate to form a (Y_i, U_j) pair. If no more valid U

coordinates exist for this Y_i , then the next Y coordinate is taken from the input buffer and the loop is repeated.

A U_j coordinate is considered to be valid if it satisfies the relation $U_{i \min} \leq U_j \leq U_{i \max}$ for a certain Y_i . The values $U_{i \min}$ and $U_{i \max}$ depend on the chamber geometry and for a small value of the angle α , the interval $[U_{i \min}, U_{i \max}]$ is narrow, which leads to a considerable reduction on the number of (Y_i, U_j) pairs to be considered.

For a given (Y_i, U_j) pair, there exists a range of possible hits in the Z and V planes which can be defined by:

$$P(Z_{ij}) = -a Y_i - b U_j + c \quad (1)$$

$$P(V_{ij}) = d Y_i + U_j - e \quad (2)$$

where a, b, c, d, e are parameters dependent on the geometry of the chamber, and are provided as input data to the processor.

The Z and V input buffers are then searched for values which satisfy the relations

$$|P(Z_{ij}) - Z_k| \leq \epsilon_1 \quad (3)$$

and

$$|P(V_{ij}) - V_l| \leq \epsilon_2, \quad (4)$$

respectively, where ϵ_1 and ϵ_2 are input constants. The relations are not exact due to the finite resolution of the detectors and to the fact that the particle does not move perpendicularly to the planes. If a Z_k and a V_l are found that satisfy both relations, then a four-plane solution has been obtained (Y_i, U_j, Z_k, V_l) . Three-plane solutions with either Z missing (Y_i, U_j, V_l) or V missing (Y_i, U_j, Z_k) can also be obtained by using this algorithm, if only one of the relations is true. If no relation is satisfied no solution exists for the pair (Y_i, U_j) . This loop is repeated for all the U coordinates valid to form a pair with Y_i . One should note that a missing coordinate may

either be created by chamber inefficiency or by the fact that a three-plane combination looks like a space point. The two cases cannot be distinguished from each other and they are referred to below as a Z or a V missing.

4. SIMULATION

A program to simulate the algorithm has been written in FORTRAN and installed in a VAX computer. This program was used to test the validity and the efficiency of the algorithm, before hardware implementation.

The program accepts commands that permit to specify the number of wires per plane, the value of the parameters ϵ_1 and ϵ_2 , the number of events to be simulated and the number of points to be randomly generated in each event. The spatial distribution of points generated was similar to that observed in the chamber during data taking at Omega. Figure 4 shows the results of a typical simulation run. The evaluated coordinates of the space points are shown and in a Z MISS the predicted value of Z is shown instead. The simulation run ends with a summary of the results.

5. THE PROCESSOR

5.1 The fast hit encoding logic

This fast readout system, FASTRO, differs from others in that wire signals are not brought to the counting room through many long, and expensive cables. Amplifiers, discriminators, digital delays and buffer registers are situated at one end of the chamber plane and they are connected to a fast encoder FASTEN, through a single 40 wire twisted pair cable.

A schematic of the FASTRO minicard is given in Fig. 5. It accommodates eight channels and plugs into a backplane carrying the power and control lines for the amplifiers, the discriminators, the digital delays and the slow serial readout described in Ref. [10].

Sixteen FASTRO minicards can be connected to a single bus segment and bus segments can be daisy chained through bus repeater minicards plugging into reserved sockets in the backplane. The first bus segment in a plane

is interfaced to the FASTEN module in order to allow for data transmission over long distances (typically 30 metres). A photograph of the on-chamber electronics together with one FASTEN module is given in Fig. 6.

To summarize, the on-chamber electronics involves three different minicards:

- i) the eight channel FASTRO card;
- ii) the bus repeater card (1 per 16 FASTRO);
- iii) the interface card (1 per plane).

After each trigger, the electronic delay section is scanned and hit channels are flagged into the buffer registers. Hit groups of eight channels are sequentially read by means of i) two fully handshaked control lines, and ii) a daisy chained OR of the set bits, inhibiting all but the last hit group of eight channels.

It can be seen that the readout speed is limited by the propagation time of the OR signal in the daisy chain. A bus segment OR has therefore been added thus allowing for a fast look-ahead mechanism.

The data format is straightforward and gives:

- i) The bit pattern in the least significant byte;
- ii) the minicard geographical position in the upper byte.

Hit bit patterns can be read every 300 ns for a 1 K channel system and cluster centroids are encoded by the FASTEN module without time overheads.

Coordinates are stored i) in a 40-word deep FIFO buffer for later readout via the CAMAC dataway and ii) into a fast memory driving an MBNIM port.

Front panel miniature coaxial sockets allow for the sequential scan of the fast memory. On board switches are used to select the scanning direction and to define two acceptance windows. Thus, the FASTENS associated with two or more trigger planes can be easily wired to allow for nested loops.

The system is mostly built with low power Shottky TTL integrated circuits and it is very reliable.

5.2 The MBNIM modules

Standard MBNIM [6] modules have been used to implement the various processor blocks. MBNIM is a modular trigger logic system mostly based on a flat cable ECL 10 K bus as a connecting medium. The bus may operate up to 100 MHz and it carries 16 data lines, a data ready line, an acknowledge line and a read/write line. The connectors are 40-pin stamp-on socket connectors, which allow fast and simple custom wiring.

MBNIM modules are a set of NIM and CAMAC modules performing the basic functions that are used in the trigger logic for most large high-energy physics experiments. The standard modules used in the processor are:

- i) ALU-16, a 16-bit arithmetic and logical unit in the CAMAC standard with two MBNIM bus inputs and one output;
- ii) RAHM, a 1 K x 16-bit random access high speed memory for look-up table applications. It is written and checked via CAMAC;
- iii) MISTER, a CAMAC pattern unit with two MBNIM bus inputs. Input data may be latched into a fast register, and transferred into a 40 event deep FIFO buffer. A 4-bit event flag may be associated to each dataword;

Two other modules were specially designed and manufactured to implement this processor:

- iv) COUNTER, a 16-bit up-down counter, which can be used either as a simple counter or as an address register and master device for a RAHM module.

- v) SPACER, a 16-bit register, which can be used as a buffer or to store the partition flags in the processor.

5.3 The structure of the point finder processor

The block structure of the point finder processor is represented in Fig. 7. In this figure the data flow and the control parts are clearly separated and each data block has its own independent control. All the data blocks are implemented by a set of interconnected MBNIM modules. Each control block is formed by a PROM-based control structure, implemented with ECL 10 K logic on a board compatible with the MBNIM. A photograph of the complete system is shown in Fig. 8.

5.3.1 The data blocks

There are eight data blocks: VDA, YDA, UDA, ZDA, PFI, PFIIV, PFIIZ, and PFIII. The first four are concerned with data acquisition for the processor and the last four implement the point finder algorithm. Data acquisition is done in parallel for the four planes and after its completion the point finder starts the data processing.

5.3.1.1 Data acquisition

The data acquisition blocks have all the same structure with the exception of the block YDA which has a simpler one. There are three functional groups of MBNIM units in the structure of each data acquisition block (UDA, VDA, ZDA): DATA RAM, POINTER RAM and FLAG REGISTER. Both RAMs have MBNIM counters as address registers.

Data comes directly from the respective FASTEN in ascending wire number order and it is stored in the DATA RAM. Each data word belongs to a certain partition and a FLAG REGISTER (MBNIM spacer) indicates which partitions contain data at the end of the data acquisition. When there is data in the partition, the associated POINTER RAM location will contain the DATA RAM address of the first word in the partition. Because of the structure of the algorithm which requires that all the values in DATA RAM Y have to be considered in sequence during point finding, the YDA block consists only of the DATA RAM and address register.

5.3.1.2 Point finding

The point finding section of the processor is pipelined into three stages to speed up the execution of the algorithm, whose flow-chart is shown in Fig. 3.

Stage I: The first stage, PFI, reads a Y_i coordinate from DATA RAM Y and it evaluates the corresponding $U_{i \max}$ and $U_{i \min}$. The values $U_{\max}$ and $U_{\min}$ for all possible Y values are stored as tables in MBNIM RAHMs. PFI then inspects the UFLAG REGISTER to determine whether there is at least one U coordinate in the partitions between UMAX and UMIN. If there is no data in these partitions it reads the next Y from DATA RAM Y and repeats all the process again. On the other hand, if there is data in these partitions, it will read the first word in DATA RAM U which belongs to these partitions and it will form a (Y_i, U_{ij1}) pair. $P(V_{ij1})$ and $P(Z_{ij1})$ are then evaluated using also MBNIM RAHM tables and the values Y_i , $P(V_{ij1})$ and $P(Z_{ij1})$ are then stored in interstage buffers as they are needed to be used in the next stages of the pipeline.

PFI proceeds to form the next (Y_i, U_{ij2}) pair, to predict new $P(V_{ij2})$ and $P(Z_{ij2})$ values and to store them and Y_i in the interstage buffers. This cycle continues until all the U's in the partitions between $U_{i \max}$ and $U_{i \min}$ have been taken. When there are no more U's in these partitions a new Y is read and the cycle is repeated. The different MBNIM tables used at this stage, are stored as files in a VAX computer. They are loaded into the RAHMs via CAMAC, at initialization.

Stage II: The second pipeline stage has two identical blocks, PFIIV and PFIIZ, which work in parallel, one searching a solution for plane V and the other for plane Z. Block PFIIZ is described here.

PFIIZ reads a $P(Z_{ij})$ value from the interstage buffer and searches DATA RAM Z for a value that satisfies relation (3). The search in DATA RAM Z starts at the lowest address in the partition to which $[P(Z_{ij}) - \epsilon_1]$ belongs. A sequential search is then made until a match or a data word greater than $[P(Z_{ij}) + \epsilon_1]$ is found. Whenever a valid Z_K is found, it is stored together with Y_i in two interstage buffers.

If no valid Z value is found, then $P(Z_{ij})$ and Y_i are stored. An indication on whether a valid value has been found is transmitted to the last stage by a control bus signal. PFIIZ proceeds to read the next predicted value for Z and it executes the loop again. The comparison between the predicted values and those read from DATA RAM Z is made by using a set of MBNIM ALUs.

Stage III: The last step in the pipeline consists basically of a FIFO (MBNIM MISTER) where the results are stored. When a four-plane solution or a three-plane solution with a V missing is found, the respective (Y_i, Z_K) pair is written into the FIFO. When a three-plane solution with a Z missing is found the respective $[Y_i, P(Z_{ij})]$ pair is written. A flag is written into the FIFO associated to each pair, to define the type of solution that corresponds to the pair of coordinates.

The structure described above is easily adapted to a three-plane chamber by simply disconnecting blocks ZDA and PFIIZ. Chambers with a different wire spacing or angle α will require different parameters in expressions (1) and (2). In these cases it is only necessary to change the look-up tables that are used to evaluate these expressions.

5.4 The control blocks

The basic hardware structure is similar for the eight control blocks. The main difference between them resides in the number and nature of the input/output signals. The diagram of one of these control blocks is shown in Fig. 9.

A control block typically consists of a PROM, input and output latches and multiplexers, forming a "link-path" addressable control structure [11].

The PROM contains the control program. The PROM outputs define the external output values, the next state of the control and the multiplexer address inputs. The PROM address inputs are defined in the input latches, which receive information from the input multiplexers and from PROM outputs (NEXT STATE). The multiplexers select the inputs which must be observed at a certain control state. The control block has its own (30 MHz) oscillator to generate the clock signal.

There are only four distinct control programs, as the data acquisition control is the same in the four planes and the control programs are also identical in blocks PFIIV and PFIIZ.

6. CONCLUSIONS

The space point finding processor described in this paper could be designed and implemented with MBNIM modules in less than eight months. Most of these modules had already been used in the trigger logic of former Omega experiments and the total cost of the special purpose units (COUNTER, SPACER and CONTROL modules) did not exceed 20 KSF. The actual cost of the electronics (chamber read-out, cables, processor, crates and power supplies) was less than 80 SF per wire despite the use of a highly modular and intrinsically expensive technique. The successful operation of this system shows that a faster, more densely packaged and much cheaper version, based on the same principles, might be designed in the FASTBUS standard.

Acknowledgements

The authors wish to acknowledge the help of the CERN OME-EF group in the manufacturing of the special purpose modules. They would also like to thank Drs. W. Beusch and P. Sharp for their confidence and support.

REFERENCES

- [1] M. Hansrool and C. Verkerk, Point and track finding processors for multiwire chambers, Bubble Chamber Seminar at the Rutherford Appleton Laboratory, UK (1973).
- [2] A. Fucci et al., Design of hardware processors for use in high-energy physics, IEEE Conf., London (1974).
- [3] C. Verkerk, Special purpose processors, CERN School of Computing (1974).
- [4] C. Verkerk, Hardware processors for pattern recognition tasks in experiments with wire chambers, Meeting on Proportional Chambers, Dubna, USSR (1975).
- [5] I. Gjerpe, A fast filter processor as a part of the trigger logic in an elastic scattering experiment, Computer Physics Communications 22 (1981).
- [6] A. Beer et al., MBNIM: a modular approach to fast trigger logic for high-energy physics experiments, Nucl. Instrum. Methods 160 (1979) 217-225.
- [7] L. Bachman et al., A fine-grained electromagnetic lead-liquid scintillator calorimeter, Nucl. Instrum. Methods 206 (1983) 85-90.
- [8] R. Apsimon et al., A ring image Cerenkov counter for the CERN Omega spectrometer, IEEE Trans. Nucl. Sci. NS-32, No. 1 (1985) 674.
- [9] J. Anthonioz-Blanc et al., User aspects of MICE, CERN 81-07 (1981) 266-275.
- [10] A. Beer et al., A pulse width modulation circuit for the remote read-out of very long shift registers, CERN 78-14 (1978).
- [11] C. Clare, Designing logic systems using state machines, (ed. McGraw-Hill).

Figure captions

- Fig. 1 : The MWPC geometry. The chamber has four planes and 1024 wires per plane. The angle α between (Y,V) and (Y,U) planes is such that $\cos \alpha = 63/64$.
- Fig. 2 : The coordinates of the hit wires are stored in the input data RAMs (Y,Z,U and VRAM) in ascending order. A coordinate assumes a value in the interval [0,1023]. Depending on the four most significant bits a coordinate belongs to one of 16 different partitions. A flag (Y,Z,U and VFLAG) at 0 indicates that there are no coordinates in the partition. A flag at 1 indicates that the partition contains one coordinate at least and the corresponding pointer (Y,Z,U and VPNT) defines the address of the first coordinate of the partition in the input RAM.
- Fig. 3 : Flow-chart of the point finder control algorithm.
- Fig. 4 : Results of a simulation run for a 10-hit event, 1024 wires, $\epsilon = \epsilon_1 = \epsilon_2 = 3$ and the input data from Fig. 2. It can be seen that for this particular event the ten points were detected and on one occasion a Z coordinate was not found in the input data to match a predicted Z value (ZMISS). Twenty-four (Y,U) pairs have been analysed, with a maximum of 35 searches for a Z and/or V matches, a number well below the required 10^3 combinations of coordinates which should be analysed in an exhaustive search. An auxiliary graphics shows the distribution of the space points in the chamber.
- Fig. 5 : Schematic of the eight channel FASTRO minicard. Wire signals are continually recorded in the n-bit electronic delay which is stopped whenever there is a loose trigger. The wanted data bits are then a fixed p location from the input. A shift by (n-p) locations brings the data towards the output and hence sets the three output registers. The serial out shift register is used for the slow readout of the data whereas the parallel out register displays the bit pattern on the FASTRO bus segment under control of the OR_n , ACKnowledge and DaTa Ready signals. The OR16 line is used to speed up the slow propagation of the OR_n/OR_{n-1} signals (daisy chained).

Fig. 6 : A photograph of the on-chamber electronics and a FASTEN. The fast read-out bus can be seen at the back of the 8-channel subcards (7 of them).

Fig. 7 : Block structure of the point finder processor. The DATA blocks, exclusively implemented with MBNIM modules are shown in the upper half of the figure. The control blocks, shown in the lower half, are based on PROMs and implemented in MBNIM compatible boards.

Fig. 8 : A photograph of the space point finding processor as currently laid in one of the Omega counting rooms. The look-up tables (MBNIM RAHMs) are grouped in a single CAMAC crate under control of the data acquisition computer (VAX 11/780).

Fig. 9 : The structure of a control block. This block is based on a 256×24 bit PROM (6×10149) and it has a "link-path" addressable structure. The latches synchronize the PROM inputs and outputs.

Fig. 1

INPUT DATA

YRAM	ZRAM	URAM	VRAM
170	102	215	253
263	186	267	328
288	339	291	360
328	411	503	409
561	443	523	621
570	470	536	651
609	502	689	695
833	522	800	900
865	564	898	987
915	924	925	1009

FLAGS AND POINTERS

YFLAGS	YPNT	ZFLAGS	ZPNT	UFLAGS	UPNT	VFLAGS	VPNT
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
1	1	1	2	0	0	0	0
0	0	0	0	1	1	1	1
1	2	0	0	1	2	0	0
1	4	1	3	0	0	1	2
0	0	1	4	0	0	1	4
0	0	1	6	1	4	0	0
1	5	1	8	1	5	0	0
1	7	0	0	0	0	1	5
0	0	0	0	1	7	1	6
0	0	0	0	0	0	0	0
0	0	0	0	1	8	0	0
1	8	0	0	0	0	0	0
1	10	1	10	1	9	1	8
0	0	0	0	0	0	1	9

Fig. 2

Fig. 3

Y	Z	U	V	
170	470	925	253	POINT
263	102	898	409	POINT
288	522	800	360	POINT
328	924	689	328	POINT
561	564	523	621	POINT
570	629 [P(Z)]	503	618	ZMISS
570	443	536	651	POINT
609	411	503	695	POINT
833	502	267	900	POINT
865	186	291	987	POINT
915	339	215	1009	POINT

SUMMARY OF RESULTS

NO. OF POINTS = 10

NO. OF VMISSES = 0

DETECTED POINTS = 10

NO. OF YUPAIRS = 24

NO. OF ZMISSSES = 1

NO. OF ZVLOOPS = 35

4a)

4b)

Fig. 4

PC BOARD BACKPLANE

Fig. 5

Fig. 6

Fig. 7

Fig. 8

Fig. 9

RECEIVED

RECEIVED

RECEIVED

RECEIVED

RECEIVED

RECEIVED

RECEIVED

RECEIVED