SDC-92-251

SDC
SOLENOIDAL DETECTOR NOTES

A SECOND LEVEL CALORIMETER CLUSTER PROCESSOR FOR SDC

May 1, 1992

B. Bylsma, L.S. Durkin, T.Y. Ling, T.A. Romanowski and C.J. Rush
Ohio State University

SDC-92-251 (5/1/92)

A Second Level Calorimeter Cluster Processor for SDC

B. Bylsma, L.S. Durkin, T.Y. Ling, T.A. Romanowski, C.J. Rush
The Ohio State University

1. Introduction

We describe a preliminary design of a fast digital cluster finder for the second level trigger for SDC. The average processing speed is expected to be less than $10\mu s$ per event. This processing speed is fast but leisurely when compared to the pipe-lined first level trigger clocked at 16 ns steps. Therefore this processor can take a more detailed look at the first level calorimeter trigger data thereby enabling us to better find and identify physics objects. All parameters relating to the physics will be programmable. Extensive diagnostics will be incorporated as part of the design.

The proposed cluster processor receives the raw first level trigger data - digitized EMC and HAC energies - for each calorimeter tower. It finds and identifies hadronic and electromagnetic clusters (groups of contiguous calorimeter towers having certain minimum energy deposition). The information associated with each cluster found will be organized into packets and sent to the rest of the level 2 trigger processor. In addition, global event information will also be calculated and sent as a summary packet.

To achieve the processing speed of 10 μs, parallel hardware will be used to simultaneously process different sections of the calorimeter. The processing algorithms will be partitioned so that hardware can be analysing different phases of the same event or successive events simultaneously. The goal will be to accomplish the maximum possible processing with each clock cycle. Extensive use will be made of pipeline registers, microprogrammed control, fast ECL logic, programmable gate arrays, PLD's and custom chips where required.

This system provides flexibility for using other similar cluster finding algorithms, and additional hardware could be incorporated to match detected clusters with particle tracks. A similar cluster finding system was designed, built, and is presently operating at the ZEUS detector to study ep collisions from HERA.

2. A Specific Design

For the purpose of starting a discussion of the merit and function of a calorimeter cluster finder within the context of the overall SDC trigger architecture, we present a specific design in this note. We view this as a starting point to be modified to the specific needs of both the first level trigger and the rest of the second level trigger.

2.1 Calorimeter Geometry

The SDC calorimeter is shown in figure 1. Imagine it is cut along a line of constant ϕ and rolled flat. Figure 2A shows the relationship of the cells in a ϕ - η coordinate system. One possible data organization for level one consists of .2 by .2 cells (we assume level 1 sums four .1 x .1 cells where necessary). Figure 2B shows a memory map of this organization where memory address corresponds to η and bit position to ϕ. A bit value of '1' implies a signal over threshold for the corresponding calorimeter cell. This memory configuration we call a hit map. The cluster finding algorithm requires two hit map memories representing hits at two different user thresholds which can be down-loaded by the user. A cluster is defined as a collection of adjacent hits at least one of which is 'big'. Figure 2B shows both maps on a single diagram. Blank spaces and 'x' are logical zero, '1' and solid locations are logical one. Solid locations are the contents of the hit memory at the highest threshold. They are of course a subset of the hits at the low threshold. The 'x' locations represent what we call peripheral members. The cluster finding algorithm will find both members and peripherals and tag them as related but separate entities.

2.2 The Hardware

Input Card - The input cards (Figure 3) communicate with and receive their data from the first level trigger. To speed the cluster search process the calorimeter could be partitioned into three (for example) sections. Each section would be served by a crate of input cards. Each input card would transfer and process a relatively small (about 30) number of cells. Thus the entire transfer of data from level 1 is accomplished in 480ns (30 $\times$ 16ns). While this data is being transferred EMC and HAC values are converted to a linear scale and added together to form a total energy signal for each cell. This signal is compared to two user loaded thresholds and the results stored in shift registers on the input card. The hit data in the shift registers is then transferred to the cluster card in each crate.

Cluster Card - The cluster card (Figure 4) searches the hit data memories for clustered data. This is done in two phases, a seed search and a member search. The seed search mode examines the data in both the seed memory (high threshold hits) and the candidate memory (low threshold) a column per clock pulse till it finds a seed hit. The controller then shifts to a member search mode starting at the seed column and finds cluster members one column per clock pulse. There are eight clock cycles of overhead for the cluster search process. The search continues in this manner until all clusters are found. For a cluster spread over four columns the time to find all members is 12 clock cycles. Existing cluster finding logic in ZEUS now runs at a 50ns clock period. Figure 4 attempts to show the basics of the process. One output of the cluster finding algorithm is a sequence of column words with bits set for cluster members. These columns are sent to a FIFO which is read by the control card. Also generated each clock cycle is the mask word for the next column and a replacement word for the candidate memory. As cluster members are found their bits are erased from the candidate memory.

Control Card - The control card (figure 5) reads the output FIFO of the cluster card. Each bit corresponds to a physical tower whose value is in an input card memory. The control card calculates the location of this data and fetches it. After a few cycles of overhead EMC, HAC, SUM AND TOWER ADDRESS values for cluster members and peripheral cells are transferred each clock cycle to the processor card. A 4 bit tag word is associated with each data transfer to identify member words, peripheral words, end of cluster and end of event. While these transfers are proceeding the cluster card continues to search for additional members, new clusters and could even be working on the next event.

Processor Card - The processor card (figure 6) examines the data sent from the control card, finds and saves the maximum value and its address, feeds the data to a bank of multipliers and accumulators and to a local memory. At the end of each cluster transfer various physical parameters for that cluster such as it's total energy, total EMC and HAC energies, it's transverse energy and their running total values will exist in the multiplier/accumulator bank.

This data is then transferred to a holding register so that another cluster can be processed immediately. Since the detector is effectively partitioned into three sections with each section serviced by independent hardware it is possible for clusters to be split. The processor card organizes clusters which are totally contained in a single section into a packet of information describing the cluster. It also looks at the processed data, compares the data to user loaded parameters and generates an ID word which identifies the physics object associated with the cluster.

Clusters which are on a boundary are assumed to be fragments and are sent to a final card called the event card. Here they are compared with cluster fragments from adjacent sections and if they share an adjacent cell they are joined to form a complete cluster. The data from the joined cluster is formated in packets as above. As soon as a data packet is complete it is available to the rest of the level 2 trigger.

Event Card - While an event is being processed the Event card (figure 7) is continually trying to join cluster fragments. At the end of the event it can finish this process and unambiguously identify all clusters and make their data packets available to the rest of level 2. Simultaneously it can read the running totals of total energy, E_x and E_y from each of the processor cards and calculate missing transverse energy and total energy for the event. This information along with counts of each type of physics object found is organized into a summary packet and made available to the rest of the level 2 trigger.

2.3 Summary

For each cluster this instrument calculates the following:

total energy
EMC energy
HAC energy
peripheral energy
transverse energy

location of the tower with maximum energy
value of the maximum energy
number of cluster members
the location and value of each member
the location and value of each peripheral member.

At the end of the event the following calculations are complete:

number of clusters found
number of each kind of cluster found
horizontal component of transverse energy (E_x)
vertical component of transverse energy (E_y)
total energy sum
missing transverse energy

All or any subset of this data could be organized as output to the rest of the level 2 trigger.

Except for the dual port memory on the input card which would be an ASIC, this instrument could be built with existing technology and achieve a clock speed of 50ns per cycle. We are investigating methods to solve the split cluster problem without waiting till end of event time. We plan to model this system and study its behavior against monte carlo events. This will allow us to identify choke points in the data flow and better estimate the processing time for an average event and understand the processing time distributions which might be encountered. Figure 8 shows a time line for the processing of a typical event.

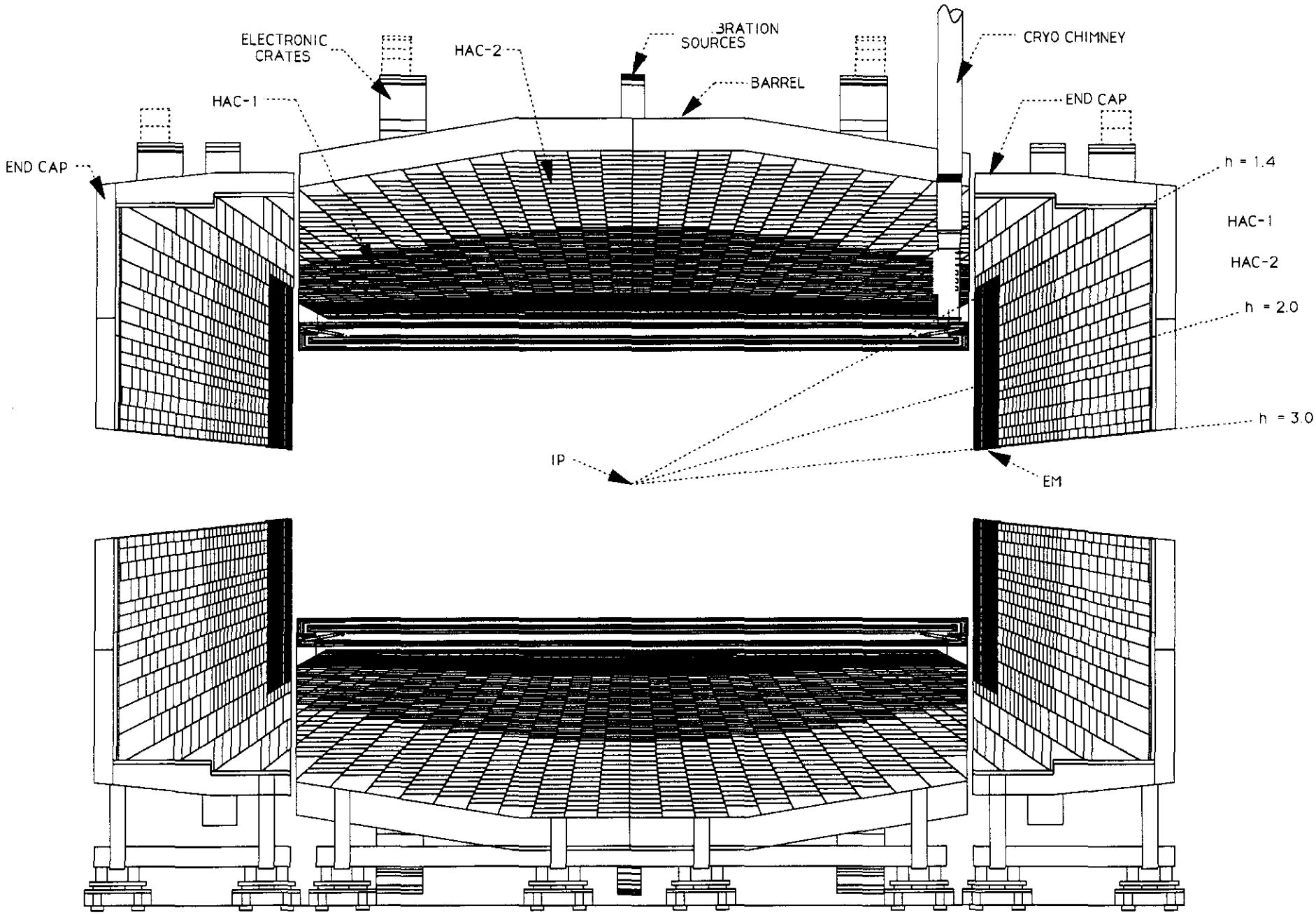

FIG. 1 SDC CALORIMETER

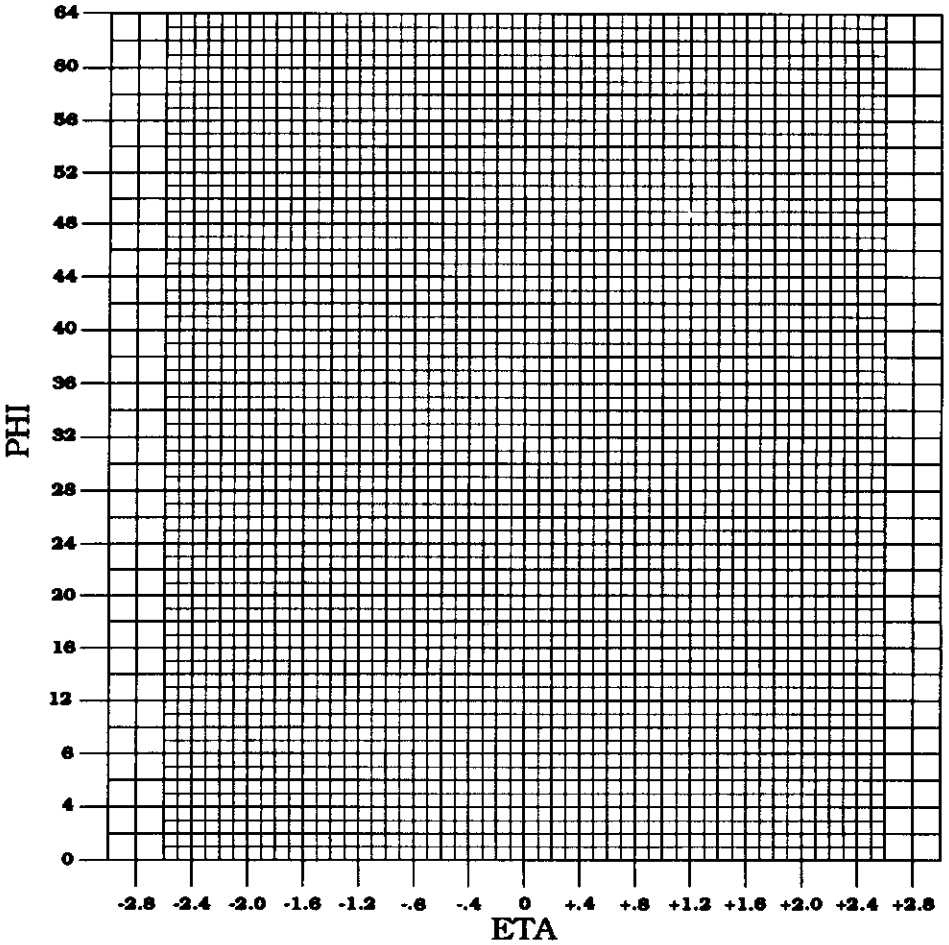

FIG. 2A FLATTENED CALORIMETER

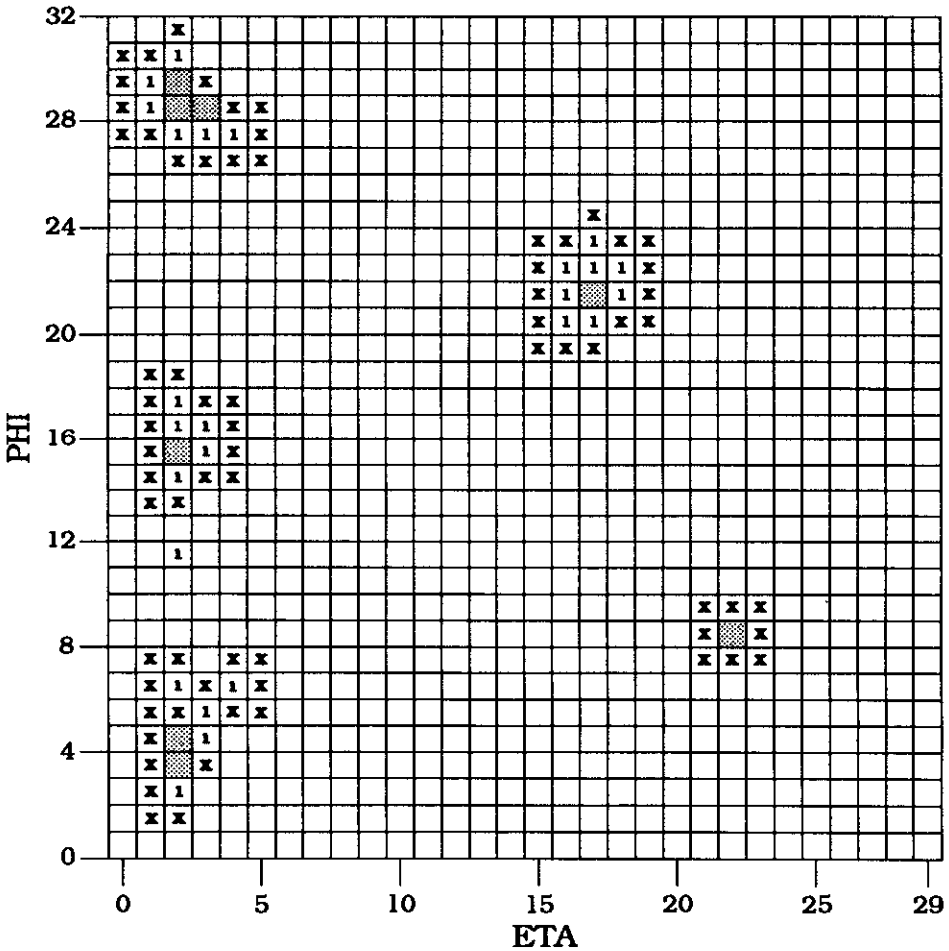

FIG. 2B MEMORY MAP OF CALORIMETER

D603A_05

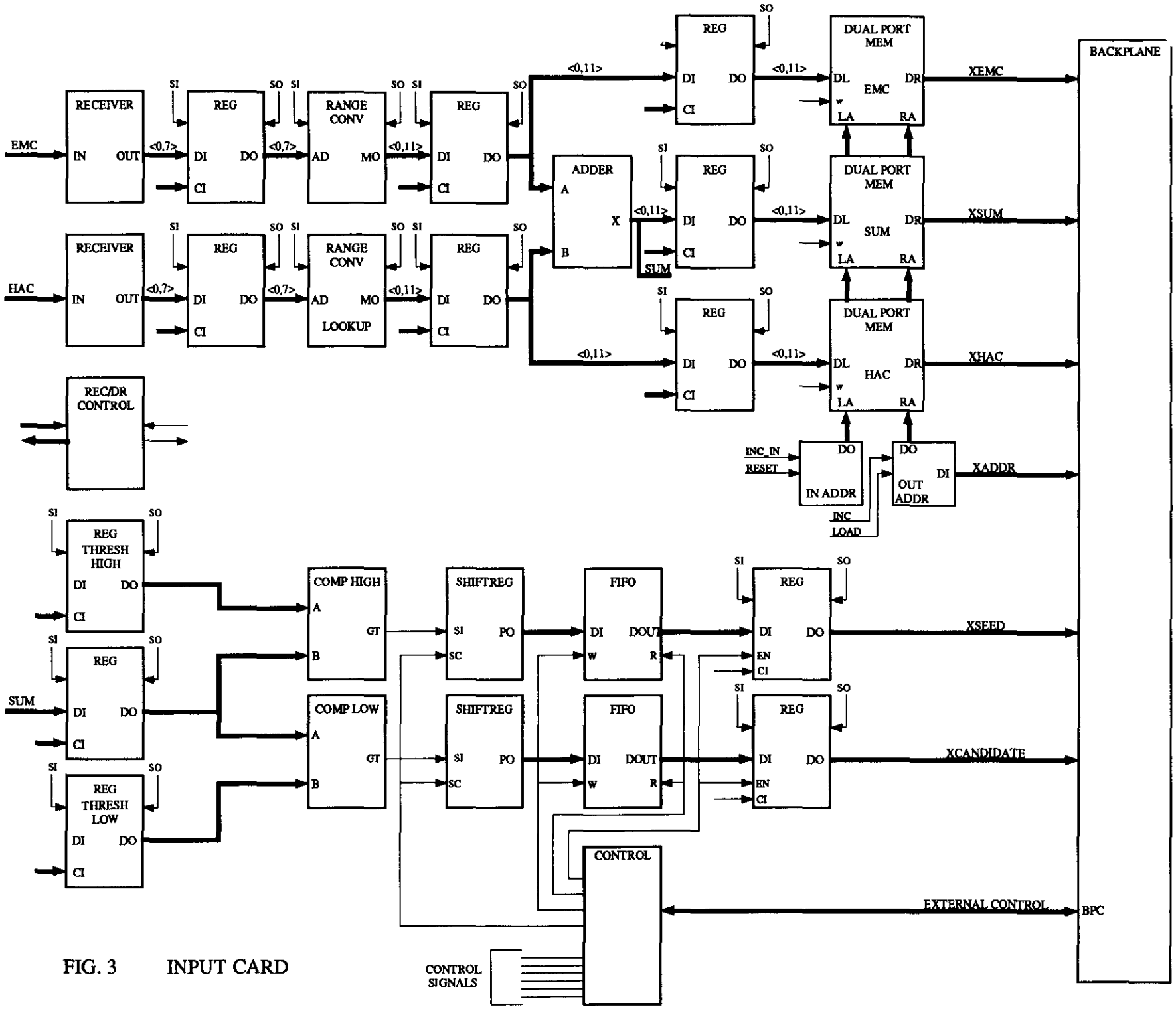

FIG. 3 INPUT CARD

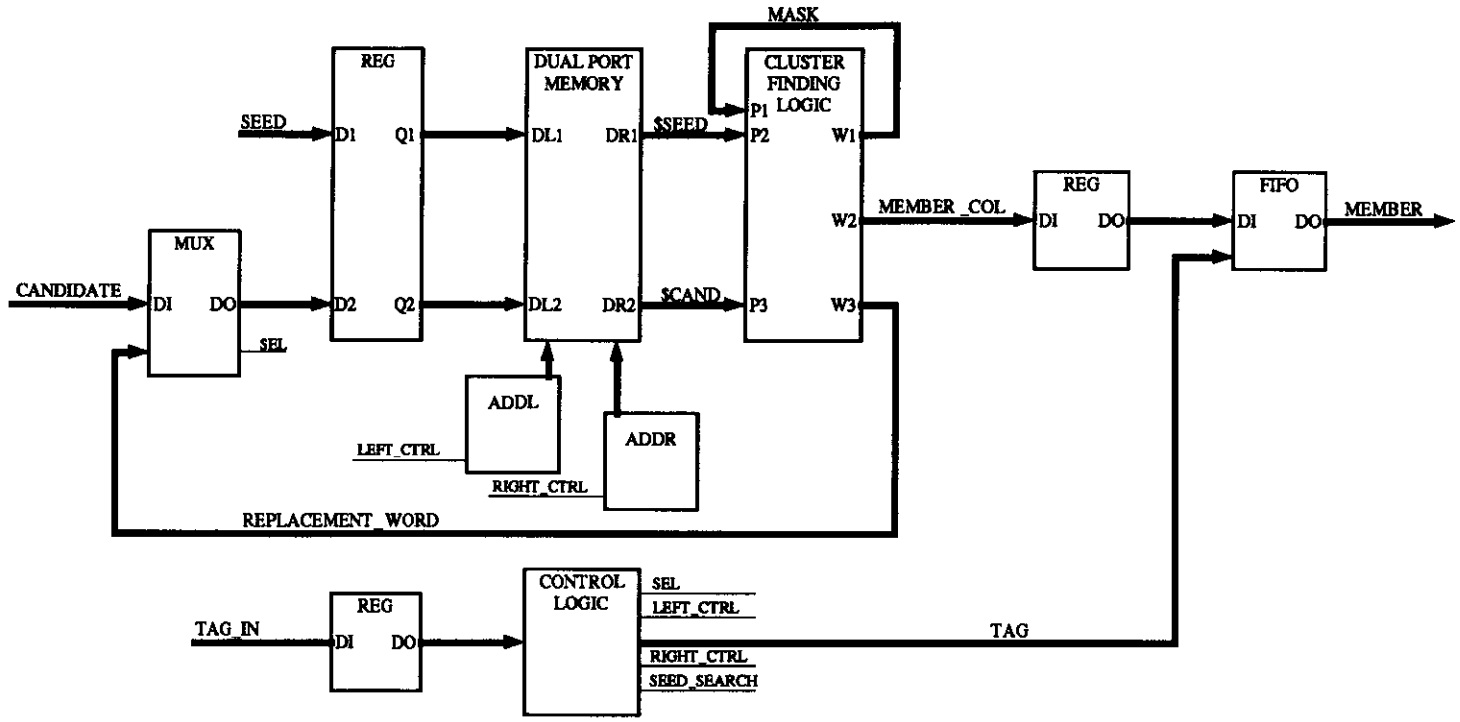

FIG. 4A CLUSTER CARD

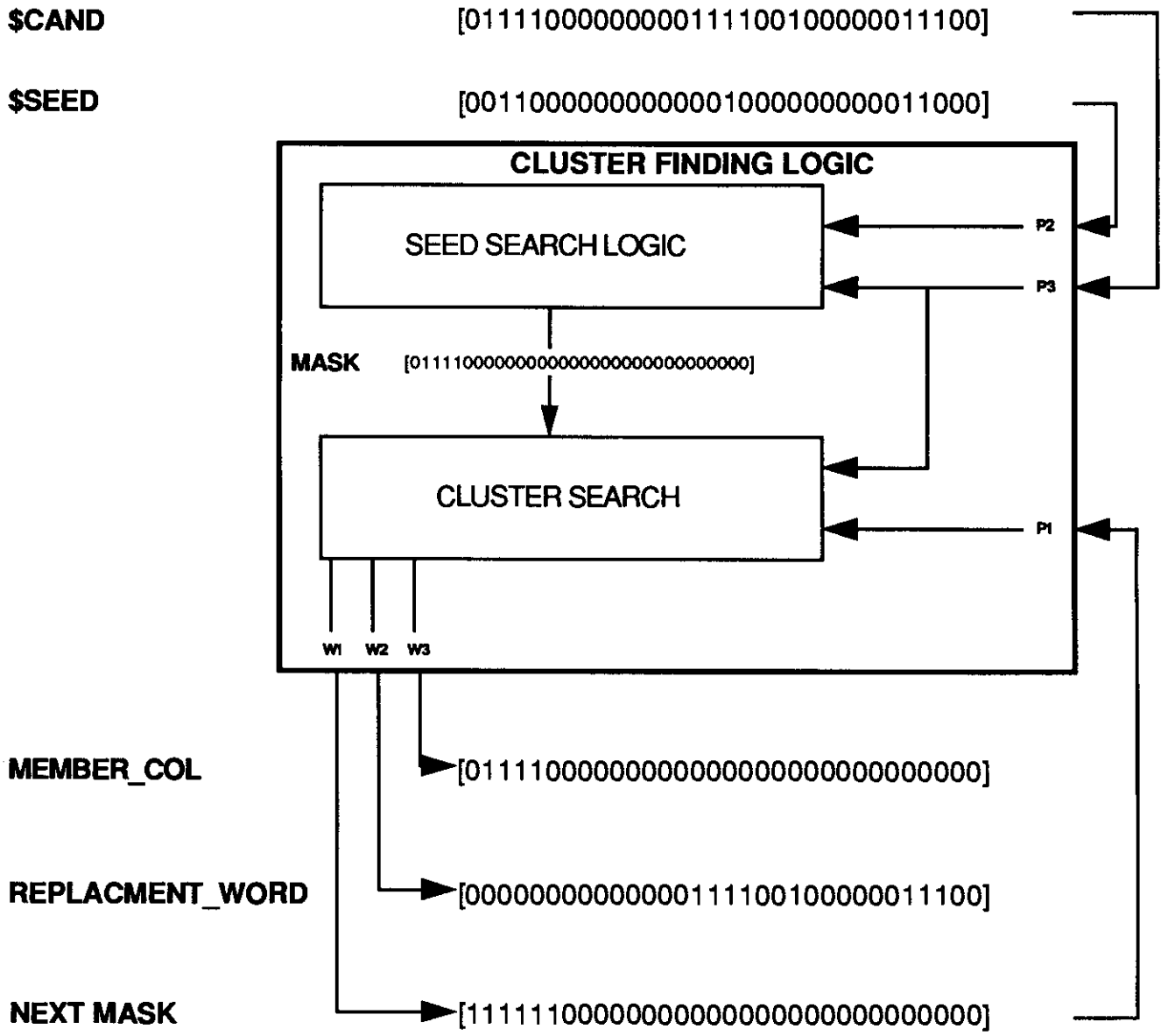

FIG. 4B CLUSTER SEARCH LOGIC

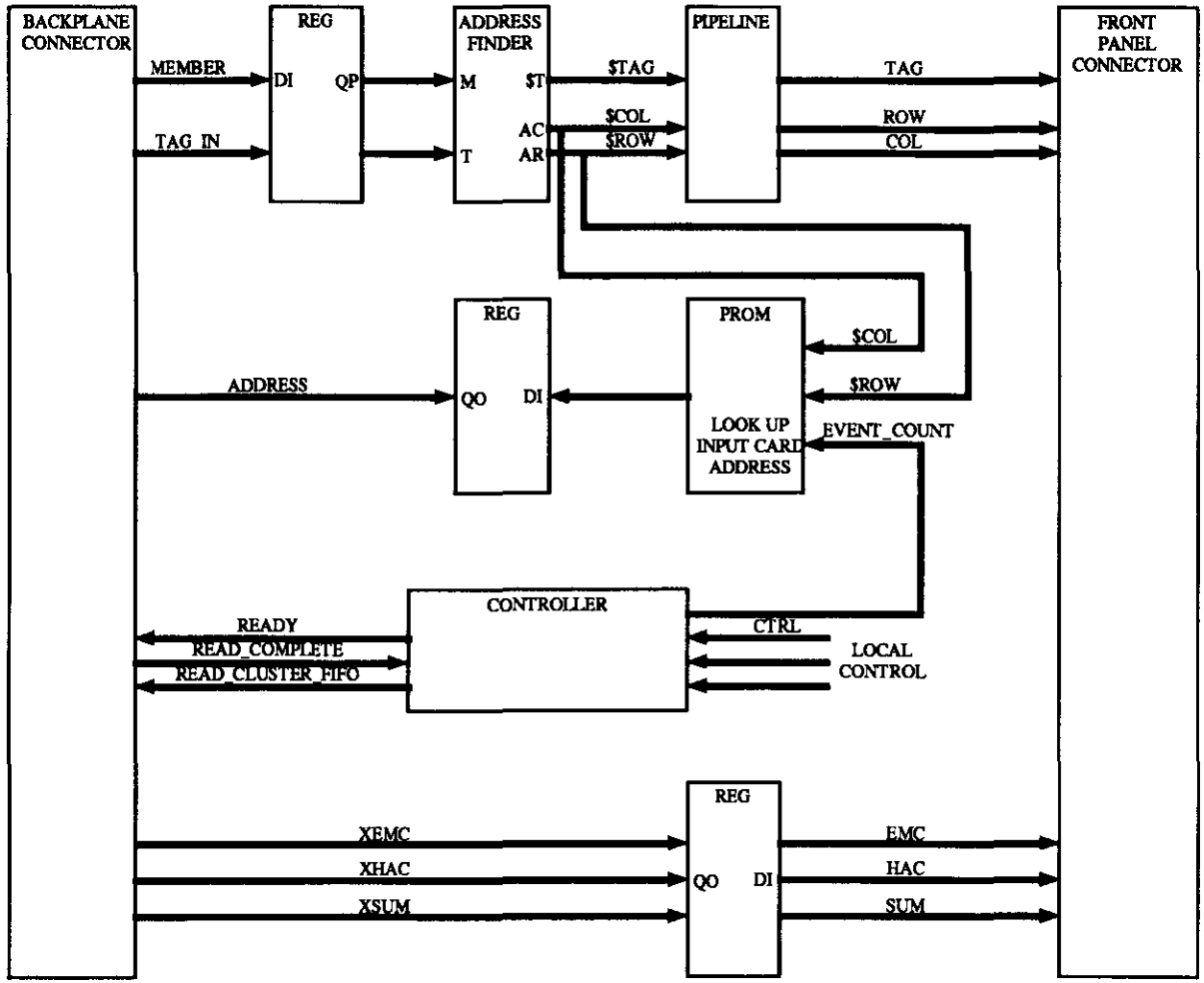

FIG. 5 CONTROL CARD

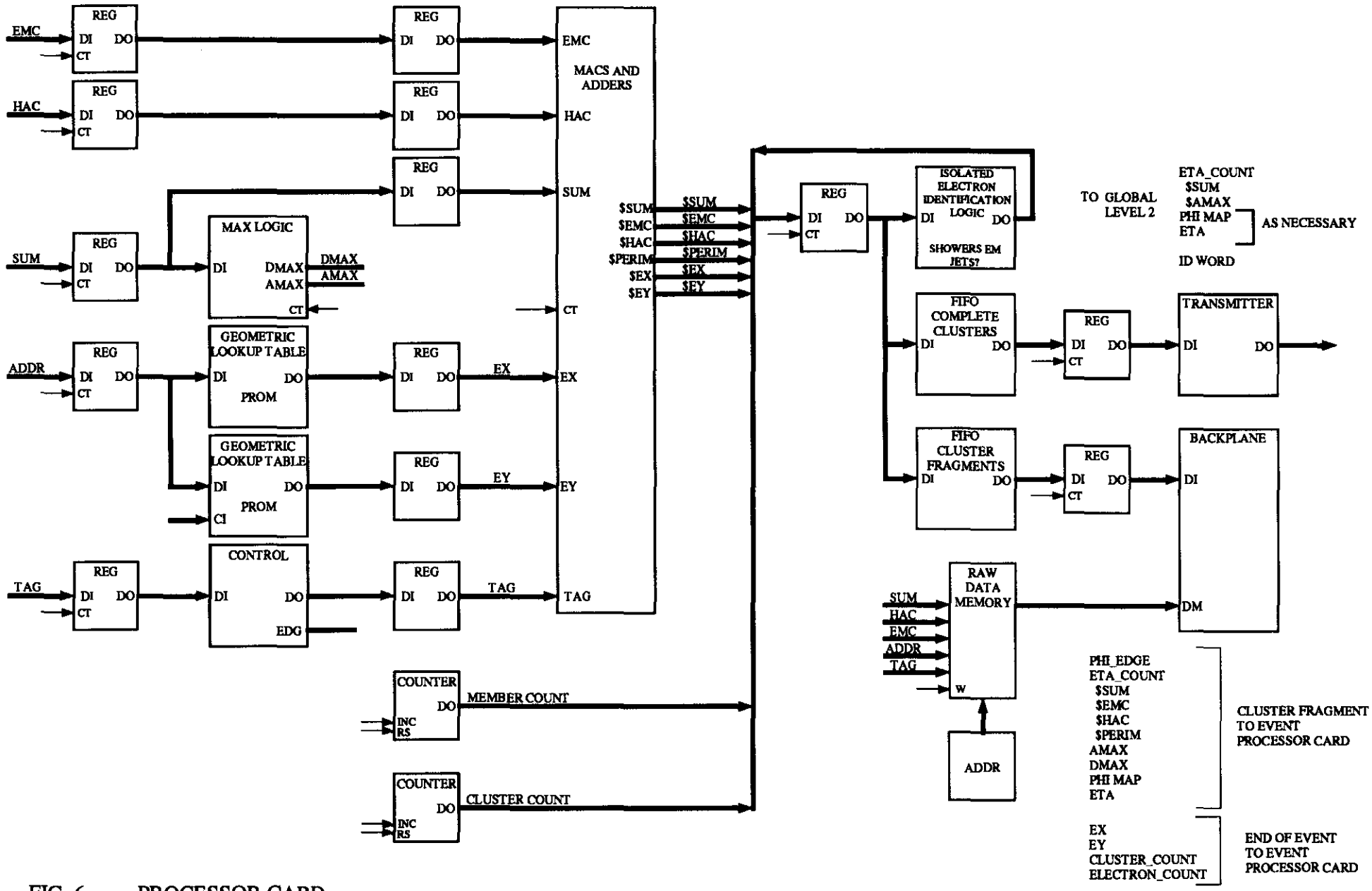

FIG. 6 PROCESSOR CARD

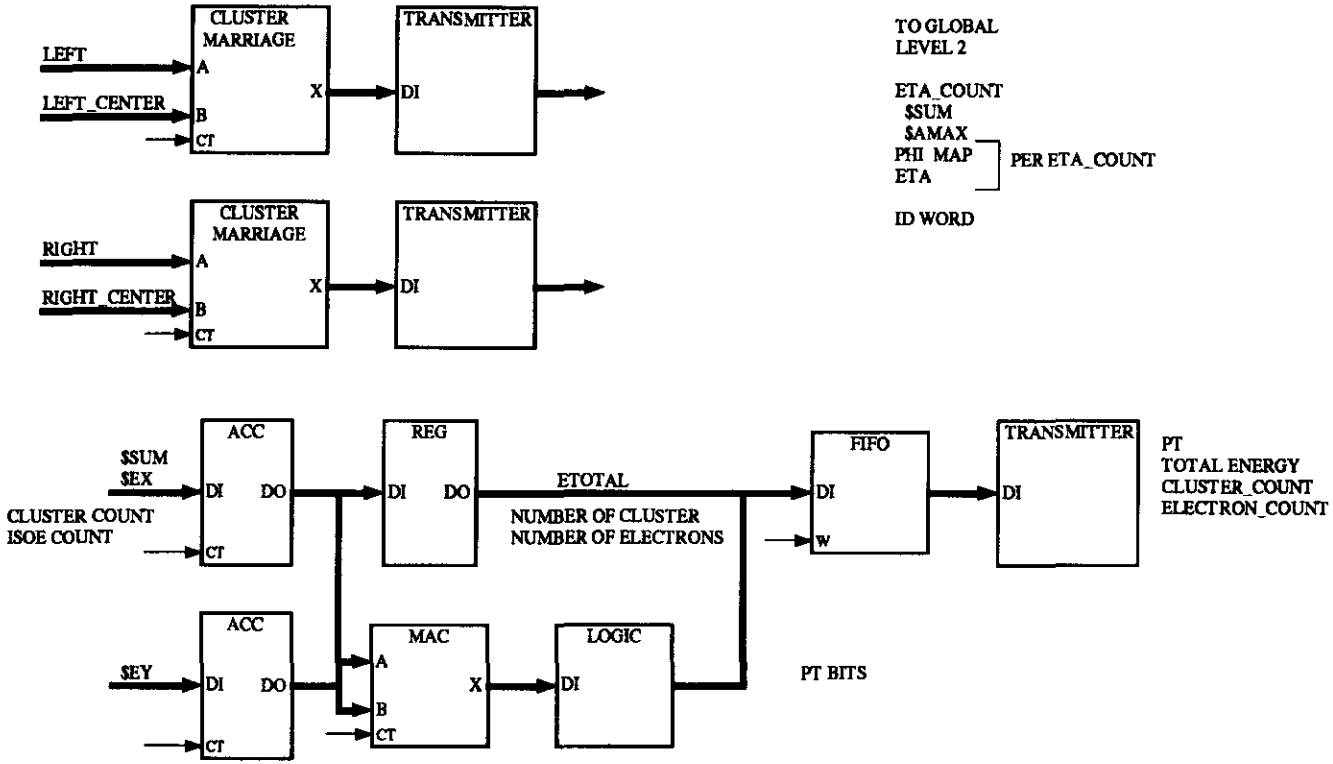

FIG. 7 EVENT CARD

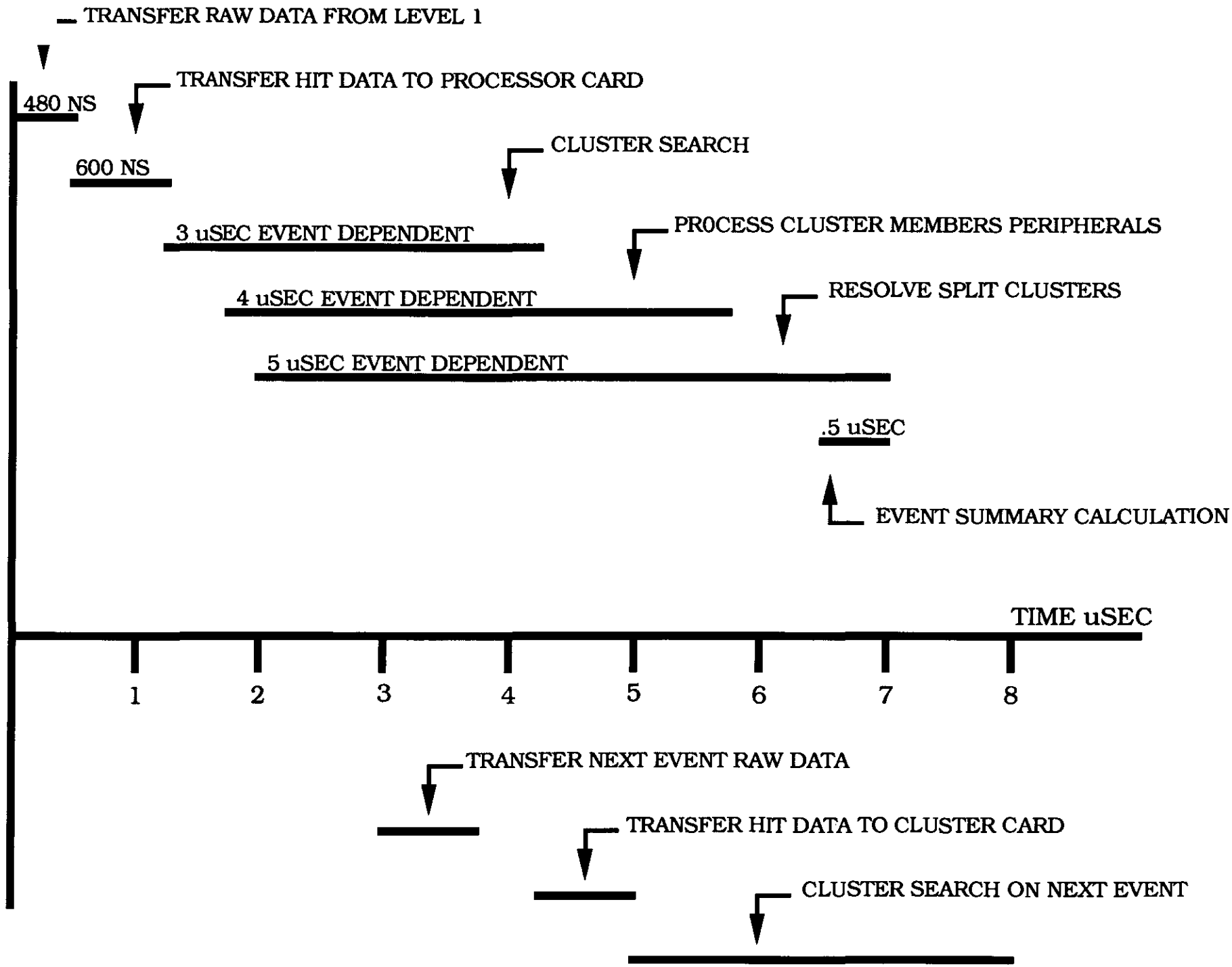

FIG.8 TIME LINE FOR EVENT PROCESSING