DD

EFI 95-25

The KTeV Hardware Cluster Counter

C. Bown, E. Cheu, J. Dusatko, H. Sanders, M. Zeleznik

Enrico Fermi Institute

University of Chicago, 5640 S. Ellis Ave, Chicago, Illinois 60637.

May 26, 1995

SW9525

ABSTRACT

We have designed and built a trigger processor that determines the number of isolated showers deposited in the electromagnetic calorimeter of the KTeV detector. Our algorithm takes advantage of pattern recognition and parallel processing techniques to increase its speed relative to more conventional cluster finding algorithms. With a relatively modest clock speed of 20 MHz, this processor reaches a decision in under $2\mu\text{s}$.

1 Introduction

The KTeV detector [1] is currently under construction at Fermilab. Its main goal is to determine the parameter $\text{Re}(\epsilon'/\epsilon)$, a measure of direct CP violation. In addition, the KTeV experiment will search for and study many rare neutral kaon decays. Of particular interest in the study of CP violation is the decay $K \rightarrow \pi^0\pi^0$, where the final state consists of four photons. In order to measure these decays, the KTeV detector includes a CsI calorimeter [2] which will be one of the most precise high energy electromagnetic calorimeters. A drawing of this detector is shown in Figure 1. In this figure each of the squares corresponds to a single CsI crystal with transverse dimensions of either $(2.5\text{ cm})^2$ or $(5.0\text{ cm})^2$ and a length of 50 cm. In total there are 3100 CsI crystals in the calorimeter with two holes near the center of the array through which two neutral kaon beams pass.

Events which are read out from the KTeV detector are required to pass a two level trigger system. The first level trigger for those events which contain only neutral particles requires a minimum amount of energy deposited in the whole CsI calorimeter. For such events the first level trigger rate will be approximately 20 kHz. At the second level the trigger is designed to distinguish $K \rightarrow \pi^0\pi^0$ decays from $K \rightarrow \pi^0\pi^0\pi^0$ and other decays by counting the number of isolated clusters of energy. An isolated cluster is a group of contiguous CsI blocks all containing energy above a predetermined threshold, separated by some distance from other such clusters. For example, $K \rightarrow \pi^0\pi^0\pi^0$ events typically will have six isolated clusters in the calorimeter while $K \rightarrow \pi^0\pi^0$

Figure 1: A drawing of the KTeV calorimeter. A kaon which decayed into four photons is shown overlaid on the calorimeter. Each of the shaded squares indicate crystals which contain a significant amount of energy with the black squares indicating crystals which are used by the cluster counter. The dashed region indicates the “ghost” row of crystals required by the cluster counting algorithm. At the bottom the division of the calorimeter into individual cards and crates for the CAB system is shown.

events will produce four clusters. Other non $2\pi^\circ$ decays only contain either one or two isolated clusters.

2 The Cluster Counting Algorithm

Typical cluster finding algorithms, such as the one used by the E731 cluster counter built at the University of Chicago, [3] are based upon locating seed blocks which are used to form the beginning of a cluster. In such algorithms a list of all hit crystals (crystals containing energy above a preset threshold) is first generated. From this list one of the blocks is chosen as a seed block. All hit crystals adjacent to the seed block are determined. Hit neighbors next to the new blocks are then located and this process repeats itself until none of the neighbors is determined to be a hit crystal. In this manner a cluster grows outward from the initial seed block. All of these hit crystals are tagged as part of a single cluster and are removed from the hit crystal list. From the remaining crystals in the hit crystal list a new seed block is chosen with the process continuing until no entries remain in the hit crystal list. Each time a new seed block is located the count of the number of clusters is incremented.

The preceding algorithm has two major drawbacks. The first is that the cluster finding process cannot begin until the whole list of hit crystals has been loaded into the processor. So, this algorithm requires a fixed amount of time before even the first cluster can be found. The second drawback is that the algorithm does not have a fixed processing time. The processing time grows with the number of clusters and hit crystals since the cluster finder processes each cluster serially. An event containing 20 clusters will require significantly more processing time than an event containing only one cluster and large clusters will require more time than small clusters.

For our cluster finder we have implemented an algorithm which can process clusters in parallel since it does not require simultaneous information about all hit crystals.[4] We start with the idea that an isolated cluster can be enclosed by a continuous perimeter. If one were to travel in a given direction around the perimeter of a cluster, one would complete a 360° turn upon returning to the starting position. In the KTeV calorimeter which consists of rectangular blocks, a circuit around any cluster corresponds to four 90° turns. While traveling clockwise around the perimeter of a cluster, if one were to assign a +1 for every right turn and -1 for every left turn, then, after completing the circuit around the cluster, the sum of "right turns" minus "left turns" would be four. To determine the number of clusters in the whole calorimeter, one can simply count of the number of "right turns" and subtract the number of "left turns." The resulting number is four times the **total** number of clusters.

Right and left turns can be determined by examining the pattern of hits in a 2×2 array of square blocks. Figure 2 shows all possible configurations of hits in such a group. As can be seen, if no blocks are hit in the 2×2 array, then the 2×2 array is not part of the perimeter of a cluster. When all four blocks are hit or any two adjacent blocks are hit, the 2×2 array is part of a cluster, but does not constitute either a "right" or "left" turn. Any single hit crystal in the 2×2 array corresponds to a "right turn" around a cluster, while three hits correspond to a "left turn." Two hit crystals which touch at their corners correspond to two "right turns" since this configuration is the case where two clusters touch at their corners. Each of the patterns shown in Figure 2 is assigned a value which varies between -1 and +2. We determine the pattern value for each combination of 2×2 blocks. This means that each block is used in to determine four different pattern values. Summing up all

Hit Block Patterns	Pattern Value
0 - No Turns
+1 - One Right Turn
-1 - One Left Turn
+2 - Two Right Turns
0 - No Turns

Figure 2: All possible hit patterns for a 2x2 array of blocks. The value assigned to each group of patterns is shown to the right of each group.

of the 2x2 patterns results in a number four times the number of isolated clusters in the array. One drawback of the algorithm described above is that a cluster of hits which surrounds a non-hit block (i.e. a doughnut) is counted as zero. This is because it contains four “right” turns and four “left” turns. We have studied this effect in simulations of $K \rightarrow \pi^0 \pi^0$ decays and found that it almost never occurs.

The KTeV calorimeter contains a mixture of large and small blocks. This geometry has regions which do not fit well into 2x2 groups of blocks. We handle this problem by treating any column (row) of large blocks which shares a column (row) with small blocks as two columns (rows). For example, a large block which is adjacent to two small blocks is considered to be a 2x2 group of blocks where we treat the large block as two blocks. The outer edge of the calorimeter is another region which we have to treat in a special manner. One cannot decide if crystals in this region constitute part of a cluster perimeter unless there is another ring of crystals surrounding the outer edge. In our algorithm we need to include a perimeter of “ghost” crystals which surround the outer edge of the CsI calorimeter. These “crystals” are just unused inputs which surround the outer edge of the calorimeter. As shown in Figure 1 the KTeV calorimeter contains 14 columns with only large crystals and 48 columns of small crystals. By adding an additional column of “ghost” crystals on either edge we have an effective 64x64 array of elements.

One of the major advantages of our clustering algorithm is the ability to reduce cluster finding to many similar discrete operations. The algorithm only depends upon determining the pattern value for each 2x2 group of crystals. So, we can divide the array into a few (four) distinct regions and process each of these regions simultaneously. This division reduces the total processing time significantly although it requires us to correctly handle the overlaps between neighboring regions. Another advantage of this algorithm is that the cluster counting procedure can begin once the first 2x2 array of blocks has been loaded which also increases its speed. In practice we load a full 64 element column into the processor and begin processing once two such columns have been loaded. Because we only have to determine the values of each 2x2 array of blocks, the total number of clock cycles is fixed. The cluster counting processor is easily incorporated into the rest of the KTeV trigger system since it is a synchronous processor.

3 System Description

The KTeV cluster counter consists of four major components:

- Front end electronics
- Column Alignment Buffer Boards
- Cluster Counting Unit
- Cluster Counter Controller

These systems are described below. Figure 3 shows an overview of these components and their interconnections.

Figure 3: Overview of the components which form the KTeV cluster counting system.

3.1 Front End Electronics

The front end electronics produce the 3100 input signals used by the cluster counter. Each CsI crystal is viewed by a photomultiplier tube which is digitized with an innovative digital base. [5, 6] The design of this digitizing scheme does not allow us to easily pick off the anode signals from the photomultiplier tubes without introducing noise into the readout system. We considered using the digital output of the base to generate the 3100 signals. However, this option adds too much delay to the formation of the level one trigger. The solution adopted by KTeV is to use the dynode output signal from the last stage of the photomultiplier tube which closely matches the output anode signal although with opposite sign. Each of the 3100 dynode signals are sent to comparators clocked at 53 MHz, the frequency of the Fermilab accelerator. The output from each comparator is stored in a FIFO which contains information from the past 2047 events. After receipt of a level 1 trigger all 3100 signals are transmitted to the Column Alignment Buffer boards (CAB boards) in a single strobe. The signals need to travel approximately 70 feet and are transmitted on ribbon cable, using differential ECL levels.

3.2 Column Alignment Buffer Boards

The Column Alignment Buffer (CAB) boards serve to latch the data from the front end electronics, sort it into columns, and send it to the Cluster Counter. There are sixteen CAB cards with two different flavors: A and B. The CAB A boards handle columns which only contain large CsI blocks. These are labelled cards 1, 2, 15 and 16 in Figure 1. CAB B boards handle the regions of the calorimeter which contain both large and small blocks. Data to all of the CAB cards is latched on a single strobe. The output of these latches is multiplexed so that only one of the four columns of data handled by each card is transmitted to the cluster processor at any given time. The input data to each CAB card does not arrive in columns so a major function of each CAB card is to unscramble the data and rearrange it into well-ordered columns.

The 64 columns of CsI bits are divided into four regions with each region handled by a single crate of CAB boards. These regions are indicated in Figure 1. Each crate contains four CAB cards. Inputs to each crate enter through a custom backplane which handles fifty two 34-conductor flat ribbon cables. The four boards in a crate share a single control bus which traverses the front panel. The control signals originate from the Cluster Counter Controller. A second front panel bus, which is common to the entire crate, transmits data from the CAB system to the Cluster Counter. During each clock cycle a single column of data is transmitted from each of the four CAB crates. The two buses are both differential ECL, and are terminated by plug-in resistors on the last board in the chain.

One channel of the CAB card can be seen in Figure 4. The control section consists of a single 8-bit command register. The first four bits of the command register are Clear, Latch, Test, Pattern, and the last four bits are the address of the column of data to send to the Cluster Counting Unit. The Test bit sets each CAB board into test mode, transmitting a fixed data pattern. There are two test patterns, a checkerboard pattern and its inverse, which depend upon the state of the Pattern bit. The two least significant bits of the address are used by the multiplexers. The two most significant bits are combined with the slot number of the board to determine whether each CAB board should enable its output drivers or not.

Data is sent from the CAB crates in the order shown in Table 1. During the first and last cycles

Figure 4: A single channel of the CAB system.

some of the crates do not send any data. This is done so that columns of data which are required by adjacent processors may be shared. During the last clock cycles column 64 is sent twice. Since this column is one of the “ghost” columns, sending it twice has no effect on the final result.

Table 1: Order of CAB column data.

Clock Cycle	Crate 1	Crate 2	Crate 3	Crate 4
1	–	17	–	49
2	16	18	48	50
3	15	19	47	51
4	14	20	46	52
5	13	21	45	53
6	12	22	44	54
7	11	23	43	55
8	10	24	42	56
9	9	25	41	57
10	8	26	40	58
11	7	27	39	59
12	6	28	38	50
13	5	29	37	61
14	4	30	36	62
15	3	31	35	63
16	2	32	34	64
17	1	–	33	64

3.3 Cluster Counting Unit

The Cluster Counter Unit (CCU) executes the cluster counting algorithm and produces a 4-bit cluster count. Figure 5 shows the top level diagram of the CCU. The CCU measures 20.5 by 16.5 inches and resides in a custom crate. During the first stage of the cluster counting process the 64 columns of data from the CAB cards are clocked into the CCU. Four columns of data are sent simultaneously, one column from each crate of CAB boards. In the CCU board each column of data is received via ECL differential receivers and then latched. The latched output is transmitted via four 64-bit buses to four daughter cards which contain the actual cluster counter processors. For three of the buses data may be sent simultaneously to two daughter cards. This allows us to handle the column overlaps between the different processors.

The daughter cards are mounted on the CCU board via four 50-pin connectors. Each of the four daughter cards contains two banks of FIFOs and a cluster counting processor. The 64-bit column of data is fed simultaneously into one of the banks of FIFOs and the cluster counting processor. The FIFOs are used to store the bit pattern of hit crystals for later readout. The second bank of FIFOs sits at the output of the first bank and is used to buffer events which have been selected for readout. When an event has been accepted, the data is clocked from the first FIFO bank to the second where it remains until the data acquisition system is ready to read out the data, freeing

Figure 5: Top level diagram for the cluster counting unit.

up the CCU for subsequent triggers. We also use the FIFOs for loading and storing test patterns since the output of the second bank of FIFOs can simultaneously feed the first FIFO bank and the cluster processor. Each bank of FIFOs consists of four 18-bit by 256 deep IDT72205LB chips.

The cluster counting processor has been implemented on an Altera 81188 EPLD. This chip was chosen because of its large number of I/O pins and relatively high speed. After the first two 64-bit columns of data have been loaded, the Altera chip calculates 63 pattern values from each group of 2x2 blocks. The values of the patterns in Figure 2 range from -1 to +2. However, to simplify our calculations we have shifted the pattern values up by one into the range from 0 to +3. This shift introduces a fixed offset for each cluster processor which needs to be subtracted off before determining the number of clusters. The 63 pattern weights are summed together in groups of two to produce 32 3-bit numbers, which are in turn summed in groups of two to produce sixteen 4-bit numbers. This summing continues for a total of six steps until a single 8-bit number has been calculated. At the same time that this summing pipeline occurs, the next columns of data are loaded into the processor to generate new pattern values. Each resulting 8-bit number is summed with the succeeding number from the pipeline to form a 12-bit accumulator value. For each processor the number of steps required to reach a final 12-bit number is 24, seventeen steps to load in each column of data and seven more for the data to wend its way through the pipeline.

The 12-bit outputs from each of the daughter cards are transmitted to another Altera chip (EPLD8484) which resides on the CCU motherboard. This chip produces the cluster count by summing the four separate counts and then subtracting off a fixed offset of 4032 ($63 \times (17-1) \times 4$, the number of pattern values per column times the number of columns times the number of processors). We divide the output of this chip by four and send the resulting 4-bit number to the trigger system via NIM levels. The most significant bit of the result is a logical OR of the five highest bits. The result from the processor is a binary number between zero and seven with the fourth bit signifying eight or more clusters. The cluster counting process requires 33 clock cycles.

The cluster counter is read out using a protocol similar to the LeCroy FERA protocol. [7] Readout of the cluster counter is initiated whenever the KTeV trigger system decides to accept an event. For accepted events data from the first bank of FIFOs is transferred in 17 clock cycles into the second bank of FIFOs. In addition, each of the cluster processors writes its result into one of the four FIFOs. Whenever there is data in the second bank of FIFOs, a flag is sent to a readout controller which returns a readout enable signal. Data from the CCU is transferred at 20 MHz in 16-bit words into the readout controller. A total of 288 words are sent to the readout controller for every accepted event, where 272 words are for the column data and 16 words contain the processor results. Twelve of these result words contain junk data, but we read them out in order to simplify the readout microcode.

3.4 Cluster Counter Controller

The Cluster Counter Controller (CCC) generates the strobes and control sequences for both the CAB boards and the CCU board. Each crate of CAB cards receives its own instruction set along 20 conductor flat ribbon cable. The CCU receives its instructions through a 26 conductor flat ribbon cable even though it shares the same crate as the CCC. The use of cable to communicate with the cluster counting board allows us to adjust the timing delay between CAB boards and the CCU board. The instructions for the CAB and CCU signals are ECL levels. The strobes for CAB cards and the CCU board are all generated on the controller board. The CCC consists of three

major components: a decoder for interpreting CAMAC commands and trigger signals, an address generator and memories.

The CCC receives four signals from the trigger system: Start, L2abort, L2done and Readout, and generates two signals: Busy and Done. When a start signal is received, the controller begins generating instructions starting at the address stored in one of three registers. If an abort is received at any time during the cluster counting process, the controller resets and waits for another start signal. If the cluster counting sequence successfully completes, the CCC sends a done signal and waits for either an Abort or an L2done signal. An L2done signal in combination with a Readout signal initiates a second set of instructions which transfers the data in the CCU from the first bank of FIFOs to the second. This instruction set is 18 clock cycles and the starting address is kept in the second CCC register. The CCC generates a Busy signal upon receipt of the Start signal and removes the Busy signal when an Abort is received or the readout sequence is complete. If an L2done signal is received, but no readout signal accompanies this signal, then the instructions stored at the address in register three are executed. In this case the HCC sends an event place mark but no data which reduces the HCC readout time.

The CCU and CCC boards both reside in the same crate. Since this crate is quite large and contains a custom backplane, we decided not to build a standard interface (i.e. CAMAC or VME) into the controller itself. Rather, we have chosen to use two CAMAC I/O modules for communicating with the cluster controller. These I/O modules have separate 16-bit TTL level input and output ports. One of the modules is used for sending instructions to the CCU and receiving back the CCU and CCC status. The second module is used to transmit data to and receive data from the CCU. The starting addresses for the three instructions sets as well as the instructions themselves are loaded via this module. Cluster counter test data is loaded via this CAMAC module as well. Test data is written first into the controller and then clocked out, via a 16-bit port, into the cluster counter. Results and data from the CCU's second FIFO bank are read into the controller through this same 16-bit port and back into the CAMAC I/O module. For debugging purposes the I/O modules can also generate the controller clock allowing instructions to be clocked one at a time.

4 Performance

The Hardware Cluster Counting system has been assembled and tested. We have found that it performs up to expectations. At a clock speed of 20 MHz, which is dictated by the FERA readout interface, the cluster counter finishes its task in $1.65\mu\text{s}$. This time is well within the $2\mu\text{s}$ limit required by the KTeV experiment and should keep the experiment deadtime to below a few percent. The system has run continuously for over one month and found to be robust. We have tested the system using both the CAB test patterns and simulated $K \rightarrow \pi^0\pi^0$ decays and have yet to find a cluster counter error. We will continue to run these tests of the system until the KTeV experiment begins to take data in 1996. During data taking, we anticipate that the hardware cluster counter will reduce the level two trigger rate by a factor of ten relative to the level one trigger rate.

5 Acknowledgement

We would like to acknowledge the contributions of M. Haney and T. Yamanaka who devised the original cluster counting algorithm. In addition, we received invaluable assistance from the Univer-

sity of Chicago Electronics Development Group, in particular from S. Mai, F. Quinn and J. Dudas. This work is supported by the National Science Foundation and the Department of Energy.

References

- [1] K. Arisaka et. al., *KTeV Design Report* Fermilab-FN-580.
- [2] B. Winstein, *A Pure CsI Calorimeter for the Study of Kaon Decays*, Heavy Scintillators for Scientific and Industrial Applications, (Edition Frontieres, 1993), p. 279.
- [3] M. Asner et. al., *A Cluster-Finding Trigger Processor*, Nuclear Instruments and Methods in Physics Research, A291 (1990), pp. 577-586.
- [4] M.J. Haney et. al., *Cluster Counting for E799/E832 at Fermilab*, Conference Record of the 1992 Nuclear Science Symposium, (October 1992) pp. 338-340.
- [5] R.J. Yarema et. al., *A Fast, Wide Range Charge Integrator and Encoder ASIC for Photomultiplier Tubes*, IEEE Trans. on Nuclear Science, Vol 4 (August 1993), pp. 750-752.
- [6] R.S. Kessler et. al., *Beam Test of a Prototype CsI Calorimeter*, Submitted to Nuclear Instruments and Methods in Physics Research.
- [7] 1994 Research Instrumentation Catalog, LeCroy Corporation, Chestnut Ridge, NY 10997, USA.