receivers are not only unsychronized with the local clock but are also not aligned to the same bunch crossing. The method of synchronization we are using was originally proposed to handle the SDC fiber link word synchronization on the SSC [4]. We offer here a short description of the technique followed by a detailed description of an ASIC proposed as a local component of the global synchronization method.

In order to achieve synchronization with this method several conditions must be met.

1. A FIFO several crossings deeper than the difference in time between the shortest and longest link is required at the receiving end of each link.

2. When a "reset" or "realign" of the link is released at the sending end, the front end data entering each link must be associated with the same bunch crossing.

3. One link is dedicated to control only. This control link is slightly longer, in transmission time, than any of the other links in the system and is used to enable the outputs of all the Synchronization ASICs.

If the above conditions are met, the circuit described below should successfully synchronize any individual link with all the rest.

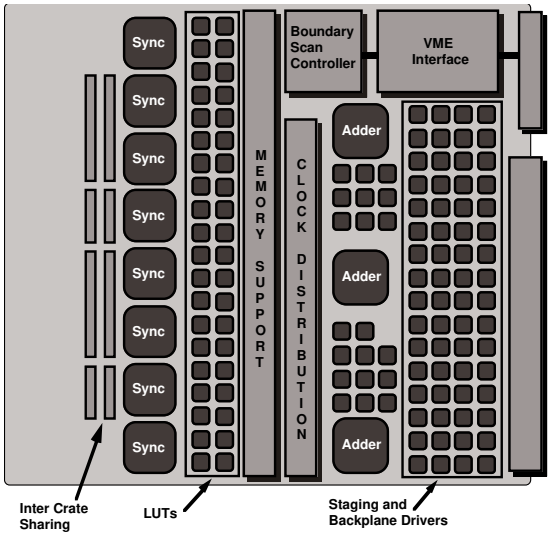

Figure 9. Front view of the Receiver card with Synchronization and Adder tree circuitry.

4.3.1 Synchronization ASIC

The synchronization ASIC is designed to receive four channels of parallel data (84 bits) from the serial/parallel converters. A block level diagram of the ASIC is shown in Figure 10. Data is transmitted from each converter along with a strobe, CLK_N, and a command/data control bit, c/d_N. These two bits

are in synchronization with the incoming data, but not with the local trigger processing system clock. CLK_N is used to clock the data into a FIFO within the Synchronization ASIC. The command bit, c/d_N, is used to switch the ASIC out of data transfer mode into synchronization mode.

When a channel is in synchronization mode it is receiving a sequence of commands designed to bring the data into the FIFO in proper alignment. Each channel has a small controller which decodes the command information into STOP, CLEAR, START, and LAST. The STOP command blocks input data to the FIFO. The CLEAR command clears the FIFO. The START command enables the FIFO input and sets up the output to be enabled synchronously with the local system clock when the enable from the control link, Enable Out, reaches the FIFO. The LAST command is used as filler to ensure there will be at least two words in the FIFO of the longest data link when the Enable Out arrives.

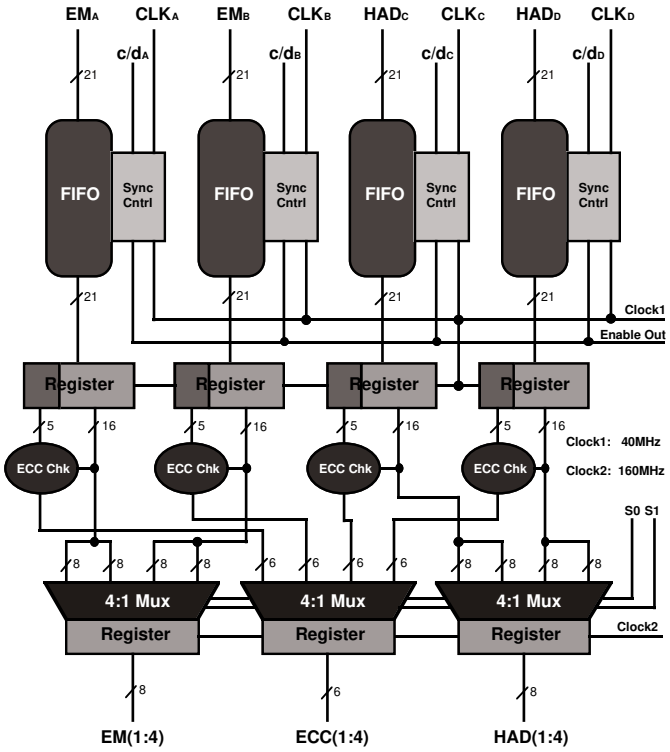

Figure 10. Block diagram of Synchronization ASIC.

The Enable Out, arriving from the control link, has been synchronized with the local system clock by the internal logic of control link's Synchronization ASIC. The Enable Out is deskewed with respect to the individual crate clocks, fed to the data links in parallel, and used to enable each synchronizer's FIFO output. All the data in the separate FIFOs is clocked out on the same crossing and, as a result of the conditions given in section 4.3, the data at the head of each FIFO is from the same crossing.

In order to achieve maximum utilization of board space, all the logic following the Synchronization ASIC is run at 160 MHz. This high data rate provides a potential savings of a factor of 4 in component volume. A significant savings is also realized by placing the multiplexing circuitry, necessary to convert the 40 MHz data flow into 160 MHz, at the output stage of the Synchronization ASIC. The four input channels, handling a total of eight pieces of 8 bit information per crossing, are placed on two output channels of 8 bits each. This 4:1 compression requires a corresponding increase in frequency, from 40 MHz to 160 MHz, to keep up with the incoming data flow.

There are also four Error Correcting Codes (ECCs) associated with the four input channels. After synchronization, each is checked against the data. If an error is detected a single bit is set, one for each incoming channel, and appended to the original ECC code. These four ECCs, along with their error bits, are transmitted out of the ASIC at 160 MHz on a single set of 6 pins. The full ECC information is made available off chip in case further error analysis is required.

The technology has not yet been chosen for this ASIC. The FIFO structure implies some static ram support. The operating frequency suggests GaAs or BiCMOS. The number of pins is on the order of 116, so the design will not be pin-limited. The final design of this ASIC will commence when the full set of synchronization protocols is determined for the CMS trigger system.

4.4 Lookup Tables

Lookup tables are required to translate the information coming from the calorimeter front end electronics, in compressed format, onto the several different scales used by the energy adder tree and the Electron Isolation logic. The Hadronic and Electromagnetic energies are individually translated into eight bits of linear E_T with a resolution of approximately one GeV. These values are summed to provide total energy in 4 x 4 trigger tower regions of the calorimeter. They are forwarded to the Jet Summary card for further combination in the total transverse energy calculation and used as the basis for E_X and E_Y missing transverse energy calculation.

The isolation processor on the Electron Isolation card requires electromagnetic and hadronic transverse energy with several dynamic ranges. The reference tower needs 8 bits of electromagnetic transverse energy with a resolution of .5 GeV and 4 bits of hadronic transverse energy with a resolution of .5 GeV. The neighboring towers need 6 bits of electromagnetic energy with a resolution of 2 GeV and 2 bits of hadronic energy with a resolution of .5 GeV. The energies for the neighboring towers are easily derived from the values provided by the lookup tables for the reference tower by truncating top or bottom bits of each value. To guard against wrap-around when the 2 bit hadronic values are generated, any value greater than the maximum is set to the maximum.

With a 6.25 nsec period the memories must have access times less than or equal to 3.0 nsec in order to handle the usual board propagation and setup and hold times. Both Cypress and IDT make memories capable of operating at the required speed. The major drawback to these memories is power dissipation. A large fraction of the power required for the Receiver card will be consumed by these memories.

Data is downloaded to the memories and read back through the VME interface. This requires support circuitry located in the area of the memory chips. The data inputs to the memories can be tied in parallel for writing the chips, but all the address lines (224) need to be individually buffered. The buffering is located near the memory in order to maintain short board traces in the high speed section of the logic. Reading the data out of the memories back to the VME interface requires buffering for all the data output lines. This buffering is provided by an 8:1 multiplexer within each Adder ASIC.

4.5 Energy Summing

The beginning of the energy summation tree is on the Receiver card. The transverse energy for each of the two 4 x 4 trigger tower regions is independently summed and forwarded to the Jet Summary card. On the Jet Summary card these E_T sums are used to continue the energy summation tree and also compared against a threshold to determine whether any sub-region contained jets. The E_T sums are applied to a set of lookup tables to generate E_X and E_Y for each 4 x 4 region. A separate adder tree is used to sum up E_X and E_Y from the regional values.

Though the input values at the top of the adder tree have only 8 bits of range, the adder tree has been designed to handle a dynamic range of 10 bits for either positive or negative values. This implies an overflow at approximately 1000 GeV, using the compressed scale described in the CMS Level 1 Calorimeter Trigger Performance Studies [5]. The exact value will depend on the resolution required of the input transverse energy for other trigger functions. Any overflow (or underflow) generated as the result of an arithmetic operation (A_{OV}) will stay in time with the data and be ORed with any other overflow that might have occurred in the same crossing. All values are handled as 11 bit *2's-complement* numbers.

A second overflow condition can also occur. The value sent to the trigger processor from the calorimeter front end electronics may be at the highest possible count. The lookup tables will be programmed to output $3FF_H$ for this particular input. This output code is recognized, at the top level of the adder tree, as indicating an input data trigger tower overflow (T_{OV}). It is handled much like the arithmetic overflow in that is it ORed with other T_{OV} overflows and passed down the tree in time with the data that generated it. If the overflow is caused by a hardware failure, the lookup table can be re-

written to zero out the affected channels. The arithmetic and tower overflow bits are handled separately through to the bottom of the adder tree.

4.5.1 Adder ASIC

The adder ASIC is implemented as a 4-stage pipeline with eight input operands and 1 output operand. There are only three stages of adder tree, but an extra level of storage has been added to ensure chip processing is isolated from the I/O. We have determined that the ASIC must work reliably at a clock period of 5.0 nsec in order to ensure safe operation at an in-circuit period of 6.25 nsec.

This ASIC uses 4 bit adder macro cells to implement twelve bit wide adders. Eleven bits are wired, left justified, to each operand of an adder. The LSB of each adder will be internally set to ZERO. The MSB is treated as a sign bit. Therefore, although the adder tree may be constructed from three 4 bit adders, the width of the operand data paths has been limited to eleven bits.

An Adder ASIC chip is designated as 'master' if it is in the top rank of the adder tree and as 'slave' if it is further down. Masters can generate Tower overflow (T_{OV}), but slaves can only propagate T_{OV}. Both masters and slaves can generate and propagate arithmetic overflow/underflow (A_{OV}). These bits are appended to each input and output operand, making all operands 13 bits wide. T_{OV} becomes the twelfth bit of the output result and A_{OV} the thirteenth bit. The data outputs of the chip are forced to $3FF_H$ when either an overflow or underflow occurs.

4.5.1.1 Detailed Description

The top of the adder tree is composed of four 12 bit adders and includes the logic required to detect and propagate T_{OV} and A_{OV}. The T_{OV} generate circuitry is a filter designed to detect the input code $3FF_H$. The A_{OV} generate circuitry examines the sign bits of the input operands and the results operand, together with the carry out, to determine whether or not an overflow or under flow has occurred. All eight of the T_{OV} bits are ORed together and all four of the A_{OV} bits are ORed together to form two separate overflow bits that are forwarded with the data in the pipeline. Edge triggered registers are used to store the results for the next stage of the adder tree. A block diagram of the Adder ASIC is shown in Figure 11.

The second stage contains two more 12 bit adders and includes the logic needed to propagate T_{OV} and to detect and propagate A_{OV}. From this point on, T_{OV} is forwarded down the pipeline from register to register. A_{OV} is generated in the same manner as in the first stage and the resulting two bits are ORed with the A_{OV} from the previous stage. Edge triggered registers are used to store the results for the next stage of the adder tree.

The third stage contains the final adder as well as a continuation of the T_{OV}/A_{OV} circuitry. The register at this level is the last storage element before the ASIC output. If either T_{OV} or A_{OV} have been detected, the output operand stored in this register has the value $3FF_H$. T_{OV} and A_{OV} are stored along with the operand. Adder Tree ASICs further down in the tree are designated "slaves" and are blocked from using the operand $3FF_H$ to generate T_{OV}. Thus we retain the identity of the tower overflow bits through the entire tree.

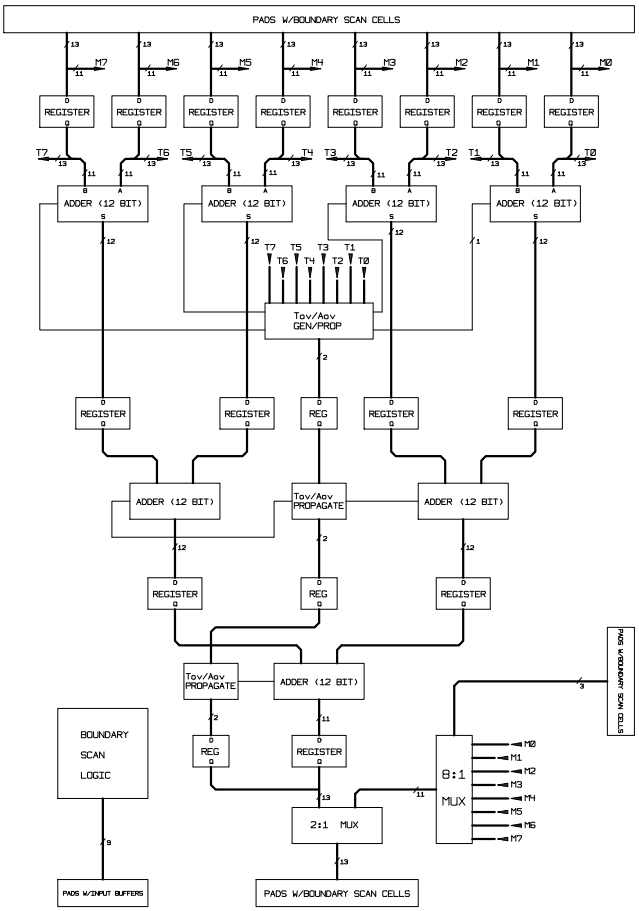

Figure 11. Block diagram of the Adder ASIC.

The last register is presented to one input of a 2:1 multiplexer before leaving the chip through the boundary scan cells and pads. The other side of the multiplexer is fed by an 8:1 multiplexer which passes any one of the eight input operands, less the two overflow bits, to the output of the ASIC. This feature was provided to minimize the external logic needed to read back the values of the lookup tables that feed the first stage of the adder tree logic.

The equations below provide definitions for A_{OV}, T_{OV}, and the output operand. Clock cycle n+1 is the time at which these three values are present at the outputs of the ASIC.

$$\left.\begin{array}{l} T_{OV}(n+1) = 1 \text{ if any input operand } OP_{IN}^{1\rightarrow 8}(n-3) = 3FF_H \text{ for the masters.} \\ T_{OV}(n+1) = T_{OV_IN}^{1}(n-3) + \ldots + T_{OV_IN}^{8}(n-3) \text{ for the slaves.} \end{array}\right\} \quad (1)$$

$$\left.\begin{array}{l} Out_{0\rightarrow 10}(n+1) = S_{1\rightarrow 11}^{7}(n), \text{ if } A_{OV}(n) \text{ and } T_{OV}(n) \text{ both equal ZERO.} \\ Out_{0\rightarrow 10}(n+1) = 3FF_H, \text{ if either } A_{OV}(n) \text{ or } T_{OV}(n) \text{ equal ONE.} \end{array}\right\} \quad (2)$$

$$\left.\begin{array}{l} A_{OV}(n+1) = A_{OV}(n) + A_{OV}(n-1) + A_{OV}(n-2) + A_{OV}(n-3) \\ \text{where} \\ A_{OV}(n-3) = \bigcup_{p=1}^{p=8} A_{OV_IN}^{p}(n-3), \\ A_{OV}(n-2) = \bigcup_{r=1}^{r=4} \left[\left(\overline{A_{11}^{r}(n-2) \oplus B_{11}^{r}(n-2)} \right) * \left(S_{11}^{r}(n-2) \oplus A_{11}^{r}(n-2) \right) \right], \\ A_{OV}(n-1) = \bigcup_{r=5}^{r=6} \left[\left(\overline{A_{11}^{r}(n-1) \oplus B_{11}^{r}(n-1)} \right) * \left(S_{11}^{r}(n-1) \oplus A_{11}^{r}(n-1) \right) \right], \\ \text{and} \\ A_{OV}(n) = \left(\overline{A_{11}^{7}(n) \oplus B_{11}^{7}(n)} \right) * \left(S_{11}^{7}(n) \oplus A_{11}^{7}(n) \right). \end{array}\right\} \quad (3)$$

S_{11}^{x} is the sign bit of the xth sum. $A_{OV_IN}^{p}$ is the overflow input for the pth operand. A_{11}^{r} and B_{11}^{r} are the sign bits of the A and B operands of the rth adder. $T_{OV_IN}^{x}$ is the tower overflow bit for the xth operand. Finally, n is the nth clock cycle in the pipeline.

4.5.1.2 ASIC Boundary Scan

The chip also contains boundary scan support. The ASIC boundary scan implementation, along with a proper board level implementation should provide full testing capability of the ASIC while it is in circuit. The boundary cells can also be used to verify circuit integrity (shorts, opens, and stuck at one/zero) at the board level. IEEE standard 1149.1 has been strictly adhered to in order to ensure compatibility with other ASIC and board level boundary scan controllers. The full JTAG controller and a major subset of the commands has been implemented. All inputs and outputs, with the exception of the five boundary scan control signals, have scan I/O cells.

The following instructions are recognized by the JTAG controller: BYPASS, IDCODE, SAMPLE/PRELOAD, EXTEST, and INTEST. BYPASS collapses the scan loop to a single bit, thus bypassing the ASIC as seen by the board level scan loop, providing faster testing of other chips on the board. IDCODE serially outputs an identification code indicating the manufacturer, part

number, and revision level of the ASIC. The four least significant bits are used to provide an address for the chip. This address will be set at the board level by hardwiring four input pins. SAMPLE/PRELOAD samples and preloads the boundary scan cells on two separate phases of the instruction. EXTEST tests the communication with the exterior by setting up values on the ASIC outputs and sampling values on its inputs. INTEST is used to test the internal circuitry. It accomplishes this by setting up known vectors on the input scan cells and capturing the results, after single stepping the core logic, in the output scan cells.

4.6 Staging Circuitry for the Electron Isolation Board

Thirty-two towers, in a 4 x 8 array, are processed on each Electron Isolation board. Data from twenty-eight neighboring towers is required to determine isolation for towers on the edge of the 4 x 8 region. Data is transferred between the Receiver card and the Electron Isolation card at 160 MHz. In order to retain point to point transmission, data going to a neighboring Electron Isolation board must be transmitted through separate drivers on separate backplane lines.

The order in which the data is transmitted to the Electron Isolation card is important. Since connector pins are at a premium, it is necessary to ensure that all lines have useful data for each of the four 6.25 nsec cycles making up a 25 nsec crossing. Board space is also at a premium, so it is necessary to limit the amount of circuitry committed to staging the data for the backplane. The most efficient way to satisfy these goals simultaneously is to pay careful attention to the order in which the fibres are connected to the input channels of the Receiver card. This, in conjunction with the multiplexing of data in the Synchronization ASIC, handles most of the data staging. The remaining staging is handled by a minimum amount of extra buffering on both the Receiver and Electron Isolation cards. More detail is provided below in the section describing the Electron Isolation processing.

4.7 Outputs

All outputs in the trigger data path are driven by ECL registers with differential outputs. Individual outputs are pulled down to V_{TT} through 390Ω. The pulldown is used to provide signal levels on the board outputs or backplane when the receiving card (Electron Isolation or Jet Summary) is not plugged into the backplane. In normal operation the receiving card will provide a 50Ω termination to the transmission line.

Even at 160 MHz there are a considerable number of outputs to deal with. The Electron Isolation card directly associated with a given Receiver card must have 32 towers of 8 bit electromagnetic and 4 bit hadronic information per crossing. Transmission of this information requires 192 pins in differential mode. In addition the neighboring cards to the left, right, top, and bottom, as shown in the top half of Figure 7, need 24 towers of 6 bit

electromagnetic and 2 bit hadronic information per crossing. These neighbors require an additional 96 pins. Finally the corner neighbors must have 2 bits of hadronic information transmitted each crossing. Since the corner information must go to four separate cards it is necessary to send it at 40 MHz, using 16 pins in differential mode. These numbers total to 304 pins required to transmit data from one Receiver card to several Electron Isolation cards.

The adder tree information is sent to the Jet Summary card. There are two totals, one for each of the 4 x 4 trigger tower regions covered by a Receiver card. These two 13 bit numbers will be multiplexed onto a single set of 13 differential pairs at 160 MHz. One of the remaining two 6.25 nsec data frames is used to transmit the quiet and minimum ionization information (4 bits), determined for the two 4 x 4 trigger tower regions, to the Jet Summary card. The remaining bits in that data frame and all the bits in the fourth 6.25 nsec data frame are presently undefined, but may be used in support of error logging and/or the tau trigger.

Every Receiver card shares its data with at most 6 Electron Isolation cards within the same crate. In addition each Receiver card sends some of its data off crate at 40 MHz to two or three neighboring crates. Crate to crate communication is handled by special cables running between the Receiver cards. This distributes the inter-crate buffering among the eight Receiver cards in a crate rather than attempting to put it all on one or two special cards at the ends of each crate. The most amount of information shared between two Receiver cards in different crates is carried on 204 twisted pair (102 in each direction) at 40 MHz.

4.8 Boundary Scan

The board level boundary scan is still under design. The boundary scan commands supported by the Adder ASIC have already been described, and the board level controller will support them as well. The following is an outline of functions under consideration.

To the extent it is possible, all registers on the outputs of the board, driving backplane lines, and all registers immediately following the Sync ASIC, will be fashioned after a boundary scan cell. These scan registers exist so that, through a combination of shifting and single stepping, we will be able to set up or capture data on the board boundaries. We do not expect to have enough room to implement the extra level of storage necessary to hold all inputs or outputs at a defined state while testing the board logic.

The board level boundary scan controller will have a hardwired program that is entered on power-up or by command through the VME interface. This program will use vectors stored in a PROM on the board to perform a minimal test on intra-board circuit integrity and make simple tests of data paths within the ASICs. The vectors will check continuity, stuck-on ones and stuck-on zeros, and shorts for those nets in the data processing path. The

results of the applied vectors will be read back into the boundary scan controller via the scan loop and compared with the expected results, also stored in PROM.

The boundary scan controller interprets commands sent through the VME controller to perform the individual JTAG instructions implemented in the ASICs, captures the resulting data, and sends it back to the crate controller on request. Several boundary scan loops will be implemented rather than a single long loop in order to minimize the time required for tests of a specific class.

4.9 VME Interface

The VME interface is a full implementation of the VME specifications, excluding READ/MODIFY/WRITE. Interrupts will be supported as these are useful during the boundary scan operations. At the moment, the implementation of Block Transfers is under consideration. While they may be useful for the memory operations, the extra logic may not be justified.

4.10 Clock Distribution

The on-board clock distribution is under design. The initial concept is discussed here. We will continue this development through board level simulation and test set-ups.

Synchronization of the incoming data with respect to the board clock is effectively handled by the Sync ASIC. The only requirement is that the global enable used by the synchronization chips must be in proper phase relationship with the board clock. Therefore, the board clock need not be adjusted to coincide with any particular phase of the incoming data.

The most successful clock distribution scheme is likely to be one in which the clock travels parallel to the data path towards the outputs on the backplane. All of the trigger data flow on the board is in the direction from the Sync ASIC to board output. Also, there are no re-entrant adds within the adder tree. As long as both the clock and the data take roughly the same path, it can be assumed that the extra delay of clock to data out, plus any combinational logic in the data path, is sufficient to ensure the clock will always arrive far enough ahead of the data at each successive storage element to guarantee hold times at that element. Setup times must be carefully determined since they are easily overrun by too much logic between registers.

Locating the clock distribution circuit near the center of the board could lead to the clock being distributed both parallel and anti-parallel to the flow of the data. Consistent distribution of the clock anti-parallel to the data will work, and could be safer, if there is known to be sufficient time between all levels of storage to allow for the small loss in cycle time due to the propagation of the clock on the board.

5. Electron Isolation Card

The Electron Isolation Card, shown in figure 12, receives data at 160 MHz in a staged fashion from at most nine Receiver Cards and performs the isolated electron algorithm described in the introduction. Some of the data originates from Receiver cards in neighboring crates, but is transmitted through the local Receiver cards. The Electron Isolation card is 9U x 280mm and resides in the front of the crate, offset from the Receiver cards by 1.52cm. The electron isolation algorithm is performed on this card and the final results sorted to identify the 4 highest rank electron isolation candidates.

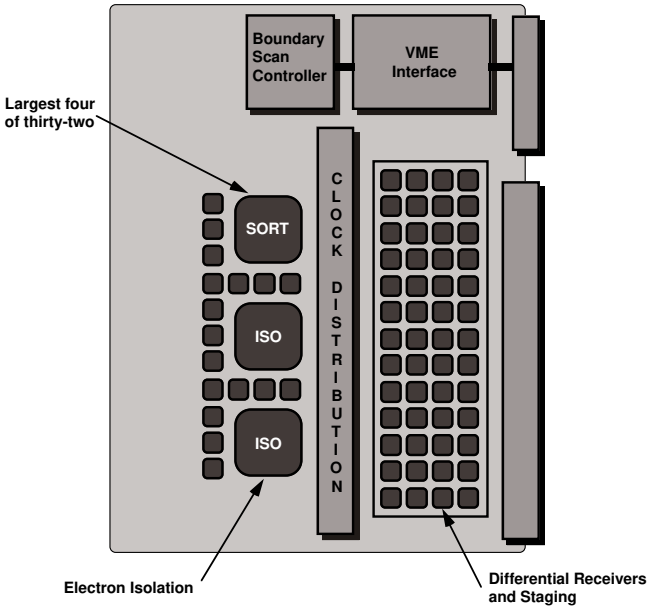

Figure 12. The Electron Isolation Card.

The Electron isolation algorithm will be implemented in a custom ASIC. The results from the electron identification ASIC are sorted in a second ASIC (Sort ASIC) and the top four candidates from the 4 x 8 trigger tower region are transferred to the Jet Summary card. The Jet Summary card does a further sort using the same Sort ASIC to output the top four electron, jet, and isolated hadron candidates to the global trigger processor crate. The total E_t, E_x and E_y information from the crate region is also forwarded to the global trigger processor.

5.1 Inputs

All data received by the Electron Isolation card comes from local Receiver cards in differential mode. Terminations for the lines will be on the cards rather than the backplane. Each card processes 32 towers of electromagnetic and hadronic information organized in a 4 x 8 array. Data is also required from the neighboring 28 towers to determine isolation on the boundaries of