

Conceptual Design of the Readout System for the Linear Collider Digital HCAL Prototype Detector

**John Dawson, Gary Drake, Bill Haberichter,
José Repond, Dave Underwood, Lei Xia**
Argonne National Laboratory

John Butler, Menakshi Narain
Boston University

Mark Oreglia
University of Chicago

Jim Hoff, Abder Mekonani, Raymond Yarema
Fermi National Accelerator Laboratory

Edwin Nobeck, Yasar Onel
University of Iowa

Andy White, Jaehoon Yu
University of Texas - Arlington

**Version 1.13
Sept. 12, 2005**

Contents

1. Introduction	p. 3
2. Overview of Instrumentation Requirements	p. 5
3. System Design Parameters	p. 7
4. Design Choices	p. 10
5. System Architecture	p. 12
5.1 DCAL Custom Chip	p. 14
5.2 Front-End Motherboard	p. 19
5.3 Data Concentrator	p. 23
5.4 Super Concentrator	p. 30
5.5 Data Collector	p. 34
5.6 VME Computer & Readout	p. 37
5.7 Trigger Farm	p. 39
5.8 Timing System	p. 41
5.9 Trigger System	p. 47
6. Trigger and Data Rates	p. 48
7. Summary of Component Counts	p. 55
8. Bibliography	p. 56

1. Introduction

A new approach is emerging in the instrumentation of detectors for calorimetry. Traditionally, calorimeters have been designed to measure the energy deposition over a wide dynamic range. This is often done by digitizing signal pulse height (integrated current) using an ADC having 12 to 18 bits of dynamic range. Because large dynamic range is often expensive, cost/performance trade-offs usually result in each read-out channel servicing a rather large part of the fiducial volume of the detector, often including many sampling layers in transverse depth.

For the Linear Collider, it is important to measure jets with a detector that has excellent energy resolution. Significant improvement in jet energy resolution over what has been obtained with previous detectors can be achieved by applying a technique known as *Particle Flow Algorithms*. By utilizing the information from both the tracking systems and the calorimeter, these algorithms rely on the correct assignment of the energy deposits in the calorimeter to the different components of a jet, i.e. due to charged or neutral particles. The Particle Flow Algorithms work best when the detector components are specifically optimized for this technique. A requirement is that the calorimeter have extremely fine segmentation, on the order of one square centimeter, laterally and layer-by-layer longitudinally. This fine segmentation results in a large number of electronic readout channels, and renders a high-resolution measurement for each channel impractical. This leads to the consideration of a simple digital readout, where the dynamic range of a single channel is reduced to a small number of bits. Monte Carlo simulations have shown that it is possible to preserve the energy resolution of single hadronic particles using a simple discriminator with only one threshold – a 1-bit ADC! In essence, this approach trades wide dynamic range on a small number of channels, for low dynamic range on a large number of channels.

Currently, new detectors are being developed that would use this technique. These include:

A. Resistive Plate Chambers (RPCs) for the Hadron Calorimeter of the Linear Collider

This detector will use 1-cm x 1-cm pads to read out RPCs made from glass. The RPCs would be operated in avalanche mode (as opposed to streamer mode), where the smallest signal to measure is approximately 100 fC. The pads are arranged in a square array. A convenient grouping of channels is 8 by 8 pads, or 64 channels per chip. The chips will reside directly on the RPCs. (See [1-10] for a description of RPC detectors.)

1. Introduction (Cont.)

B. Gas Electron Multipliers (GEMs) for the Hadron Calorimeter of the Linear Collider

The read-out configuration of this detector will be similar to that of the RPCs described above. It will also use 1-cm x 1-cm pads for read-out. The smallest signal to measure is approximately 5 fC. Like the RPCs, a convenient grouping of channels would be 8 by 8 pads, or 64 channels per chip. The chips will reside directly on the read-out pads. (See [11-19] for a description of GEM detectors.)

Detector R&D in these areas is currently in progress. A specific project involves building prototype detectors of each of the RPCs and GEMs, and characterizing the performance in a test beam. This note will describe the basic properties and conceptual design of the electronics and readout system that will be used in these studies. A goal in the project is to develop a readout system that can be used for both. A summary of the detector properties is shown in Table 1.1.

Parameter	RPCs	GEMS
Type	Avalanche	(Gas)
Geometry	1cm x 1 cm Pads	1 cm x 1 cm Pads
Capacitance	10-100 pF	10-100 pF
Smallest Signal	~100 fC	~5 fC
Pulse Width	~5 nS	~3 nS
Rise Time	~2 nS	?
Largest Signal	~10 pC	~100 fC
Noise Rates	~0.1 Hz	?
Env. Noise Susceptibility	Low	Low

Table 1.1. Summary of Detector Properties

2. Overview of the Instrumentation Requirements

Both the RPCs and GEMs are gas detectors that are biased with a strong electric field. When a particle passes through the active detector, gas molecules in the detector can be ionized, causing them to drift in the direction of the electric field. Both detectors have 1-cm by 1-cm pads, etched on a printed circuit board (PCB), arranged laterally to the electric field. The charged ions induce an image charge on the pads. The job of the front-end instrumentation is to collect the induced charge from the pads. A schematic view of an RPC is shown in Fig. 2.1. For the purposes of describing the electronics, the principle of operation for the GEMs is similar, except that the charge gain of the detector is $\sim 10\times$ smaller. In both cases, charge is induced on the readout pads when particles ionize the gas in the detector. See Fig. 2.2.

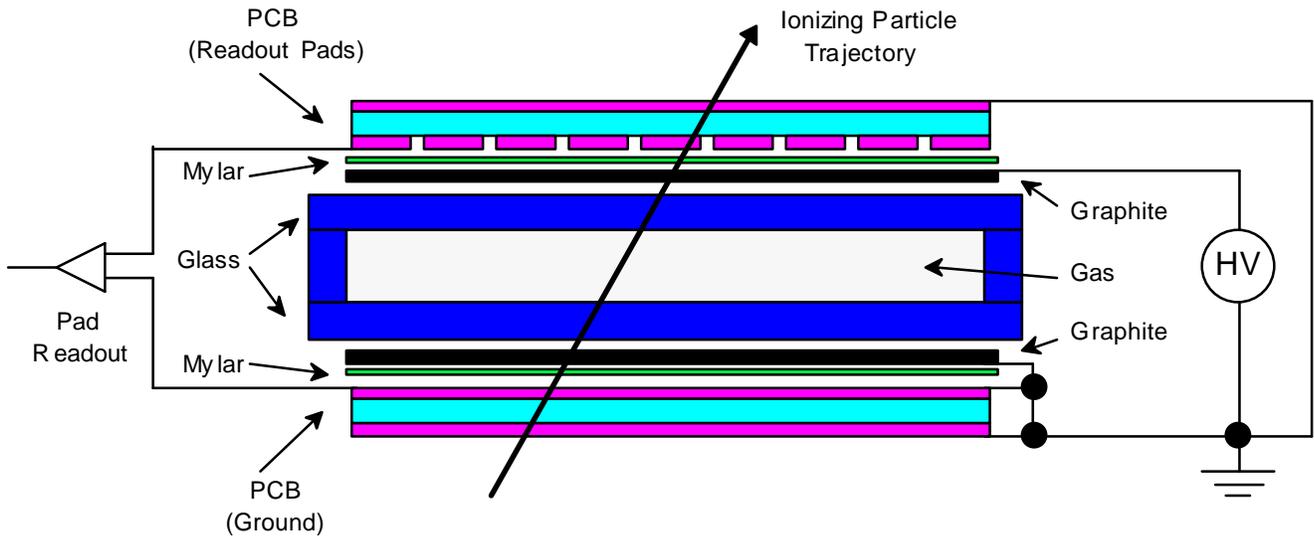


Figure 2.1. Anatomy of an RPC

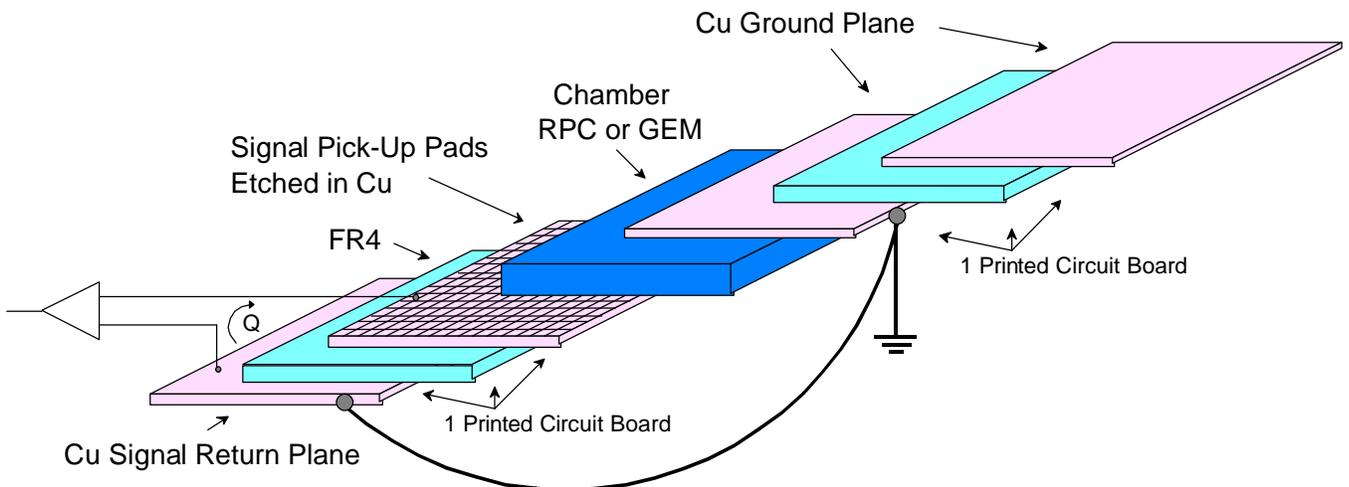


Figure 2.2. Partial Detector Plane – RPC or GEM

2. Overview of the Instrumentation Requirements (Cont.)

The prototype detector sections will be built as 1-meter by 1-meter planes, sandwiched between plates of steel. See Fig. 2.3. Each plane will have 96 by 96 pads, one square centimeter each. There will be 40 readout planes all together. Since each plane has ~10,000 pads, the full detector has ~400,000 readout channels.

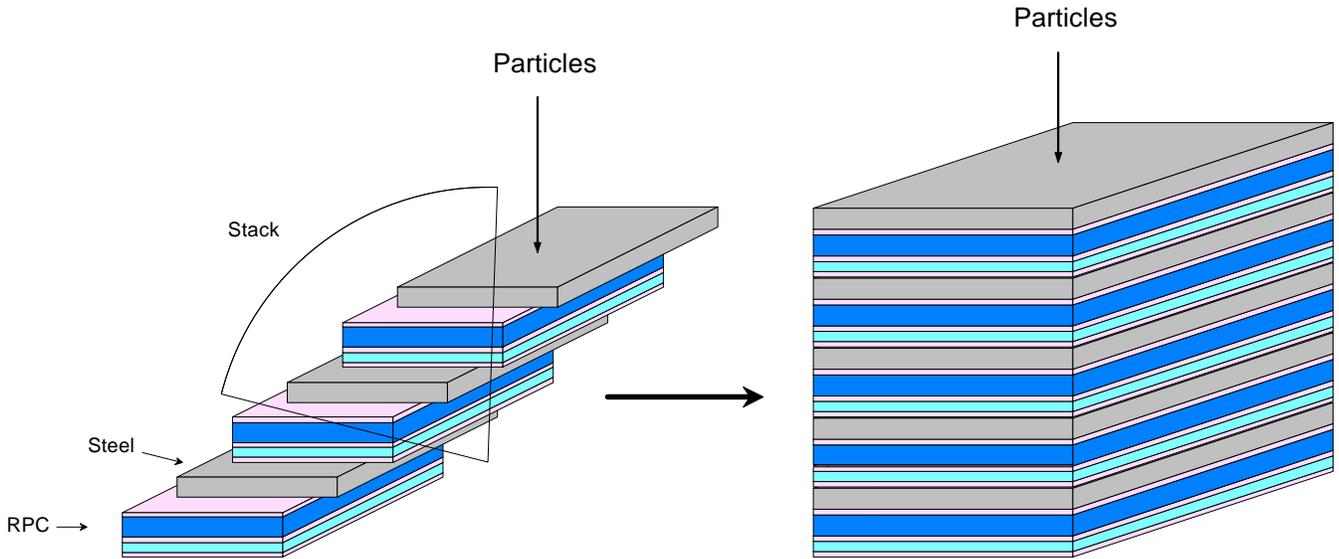


Figure 2.3. Construction of Detector Planes

When an ionizing particle goes through the detector, charge is induced in local areas of each plane for an instant of time, as shown in Fig. 2.4. By discriminating the charge deposition as hits on each plane, the trajectory of the particle, as well as other properties of the particle, can be ascertained. The primary function of the instrumentation is to record these hits as a function of time, and provide a framework for reading them into a computer for event reconstruction and analysis.

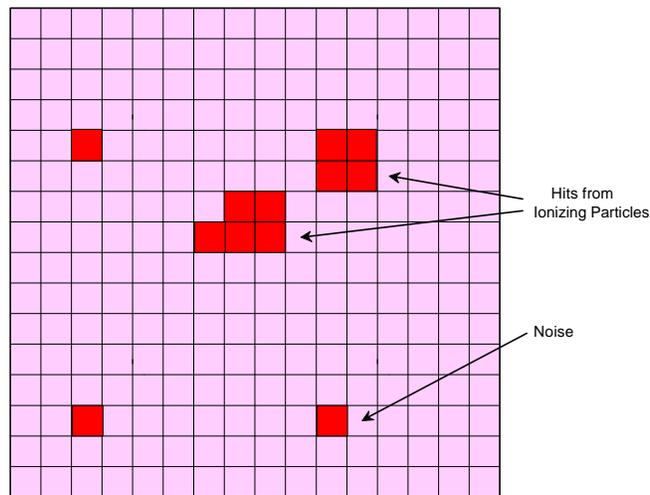


Figure 2.4. Hits on Pads (Partial Plane) at an Instant in Time

3. System Design Parameters

The conceptual design of the readout system for this application is based on the following assumptions and parameters:

3.1. Nature of Data

The nature of the measurement is to record hits from the pads as a function of time. Each pad is to have an amplifier and a discriminator with a programmable threshold, and have one bit as the output. The bit for a channel is true when the signal from the pad is over-threshold (the definition of a "hit"), and false when under threshold. Each hit shall have a time associated with it, which is synchronized over the entire detector. This reduces problems in assembling events in the higher levels of the system if there is latency in collecting data from different parts of the system.

3.2. Timing Resolution

It is sufficient for the timing resolution of events to be 100 nanoseconds.

3.3. Event Rate

The event rate for hadronic interactions foreseen for the Linear Collider is very low, on the order of one per minute. For testing the prototype detectors in a test beam, it is desirable to have a event higher rate. RPC detectors have an inherent limit in how fast a local area of the detector can recharge after being hit, requiring of order several milliseconds to recover. Furthermore, the intrinsic noise rate of RPC detectors is very low, less than 1 KHz per square meter, or 0.1 Hz per pad. GEM detectors are in principle capable of higher event rates. However, Since both detectors are candidates for this application, it will be assumed that the maximum event rate capability of the RPCs sets the performance goals. ***For the test beam, it will be arranged so that the maximum event rate will be no greater than 100 Hz.*** This is a crucial aspect of the system design, because it determines certain performance aspects such as data transmission bandwidth and multiplexing. See Section 6 for a discussion.

3.4. Live Time

While the primary goal in the development of the detector is to perform measurements in a test beam, it is desirable to have the capability to measure noise and cosmic rays as well. Again, assuming a low trigger rate, it is desirable to have the detector and electronics capable of being live 100% of the time, i.e. no dead-time from the readout electronics.

3. System Design Parameters (Cont.)

3.5. Nature of Data Acquisition

It is desirable for the data acquisition system (and trigger system) to be as simple as possible, while achieving the desired performance. Because the event rate is expected to be low, and because there is no need for complex triggers involving multiple detector subsystems, it is desirable for the system to be configured as "data push." This means that once data is stored in a local memory in the front-end electronics, it is "pushed" into the upper levels of the system without further commands or requests from the system. This reduces the complexity of the data acquisition system and the trigger system, and streamlines the data acquisition process. This does have implications for the nature and philosophy of the trigger system, as described below.

3.6. Event Triggering

The system can operate with three kinds of triggering philosophies: Self-Trigger, External Trigger, or Gated Trigger. These are described below:

3.6.1. Self-Trigger

Because the event rate for this detector is expected to be low, including the intrinsic noise rate, it should be possible to operate the system with a software trigger only, i.e. no intermediate hardware trigger. Event selection and reconstruction can be done at higher levels of the readout architecture. Signals from the detector are discriminated as a fundamental part of the architecture. It is straightforward to arrange for hits to automatically write data into local memory. Each hit is tagged with a timestamp directly on the front end to identify the occurrence in time, and addresses are added to identify the location in space. Data is read out by the data acquisition system and passed to a higher-level processor, where algorithms sort the data. Noise hits will be isolated in time and space, and are easily discarded. Data that has a clustering in time and space are processed further as possible events. It will be necessary to implement efficient algorithms for this processing. It will also be possible to overwhelm this system if the event rate is high, so care must be exercised in setting thresholds on the front-end discriminators, as well as careful control of noise.

3.6.2. External Trigger

The readout system and front-end electronics shall be capable of being triggered by an external system. The latency is expected to be no more than 2 microseconds. This will require there to be a pipeline in the front-end electronics.

3. System Design Parameters (Cont.)

3.6.3. Trigger Gate

The readout system and front-end electronics shall be capable of using a gate, to be used as a coincidence with self-triggering. This would be used in a test beam, where the approximate arrival time of particles is known, without knowing that an event actually occurred in the detector.

4. Design Choices

The following design choices have been adopted as parameters in the conceptual design of the system:

4.1. Custom Integrated Circuit for Front-End Electronics

Because of the high channel count, it is desirable to use a custom integrated circuit for the front-end electronics. The chips reside on the printed circuit board of the detector, on the opposite side of the 1 cm by 1 cm pads. Each chip services an 8 by 8 array of pads, or 64 channels, as shown in Fig. 4.1. The chip, called DCAL, performs all of the front-end processing, including: signal amplification, discrimination/comparison against threshold, recording the time of the hit, temporary storage of data, data readout, and has ancillary control functions.

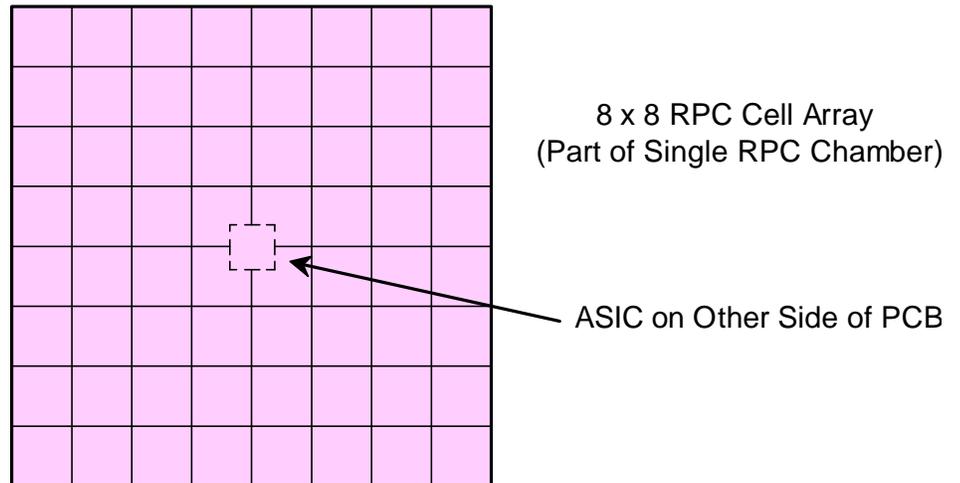


Figure 4.1. Each Custom Chip Services an 8 x 8 Array of Pads

4.2. Timing

The timing of hits in the DCAL chip shall be implemented using the concept of a "timestamp" counter. This counter is reset periodically and synchronously across the system, and advances with each 100 nS clock, which is also synchronous across the system. It is desirable to not have to reset the counters too often, but also to limit the number of bits to be read out as part of the timestamp. A reasonable compromise is taken to have the counters reset once per second. With 100 nanoseconds counting rate, the counter will need 24 bits.

4. Design Choices (Cont.)

4.3. Data Format

When a hit occurs in a chip, there are two ways to read it out: read bit pattern plus timestamp ($64 + 24 = 88$ bits), or read channel address plus timestamp ($6 + 24 = 30$). The two schemes have about the same number when three channels are hit per event. Simulations indicate that, on the average, more than three channels will be hit within a chip per real event. Therefore, the chip shall record hit pattern plus timestamp, and this forms the data when an event occurs.

4.4. Data Readout

Since the DCAL chips reside on the detector, it is desirable to reduce the number of digital readout lines to a minimum. Therefore, the readout scheme will use serial data transmission. The transmission rates at the different points in the system architecture shall be determined by the expected maximum data rates. The transmission rates will then determine the type of transmission media that is needed. Serial data transmission protocols usually require extra bits for framing. These need to be included in data rate calculations. It is desirable for the system sub-components to have buffering, to cover situations where a burst of data occurs.

4.5. Control

There are a small number functions needed for the DCAL chips (setting of threshold levels, masking bad channels, diagnostics, etc.) It is desirable to have a separate control path from the data transmission, since the latter needs to be efficient. Since digital I/O is a significant issue in this system, the control path shall also use serial transmission.

4.6. Data Multiplexing

It is advantageous to concatenate or multiplex the data streams from several DCAL chips into a single output data stream. If a DCAL chip is hit on the average of once every 10 milliseconds, and the readout time is many times faster, then there is significant excess bandwidth available. Furthermore, it is desirable to reduce the amount of back-end data processors. It is therefore desirable to multiplex data streams from the DCAL chips by a factor between 10 and 100. The level of multiplexing is to be determined by physical and logistic considerations in the setup. The appropriate addresses will be needed at collection points in the architecture to properly identify the data.

5. System Architecture

Using the assumptions, constraints, and principles outlined in Section 3 and Section 4, a readout system for the detector has been envisaged. The following is a description of the conceptual design of the readout system for this application.

The block diagram of the readout system is shown in Fig. 5.1. The system can be divided into four primary subsystems: the front-end electronics, the back end data acquisition system, the timing system, and the trigger system.

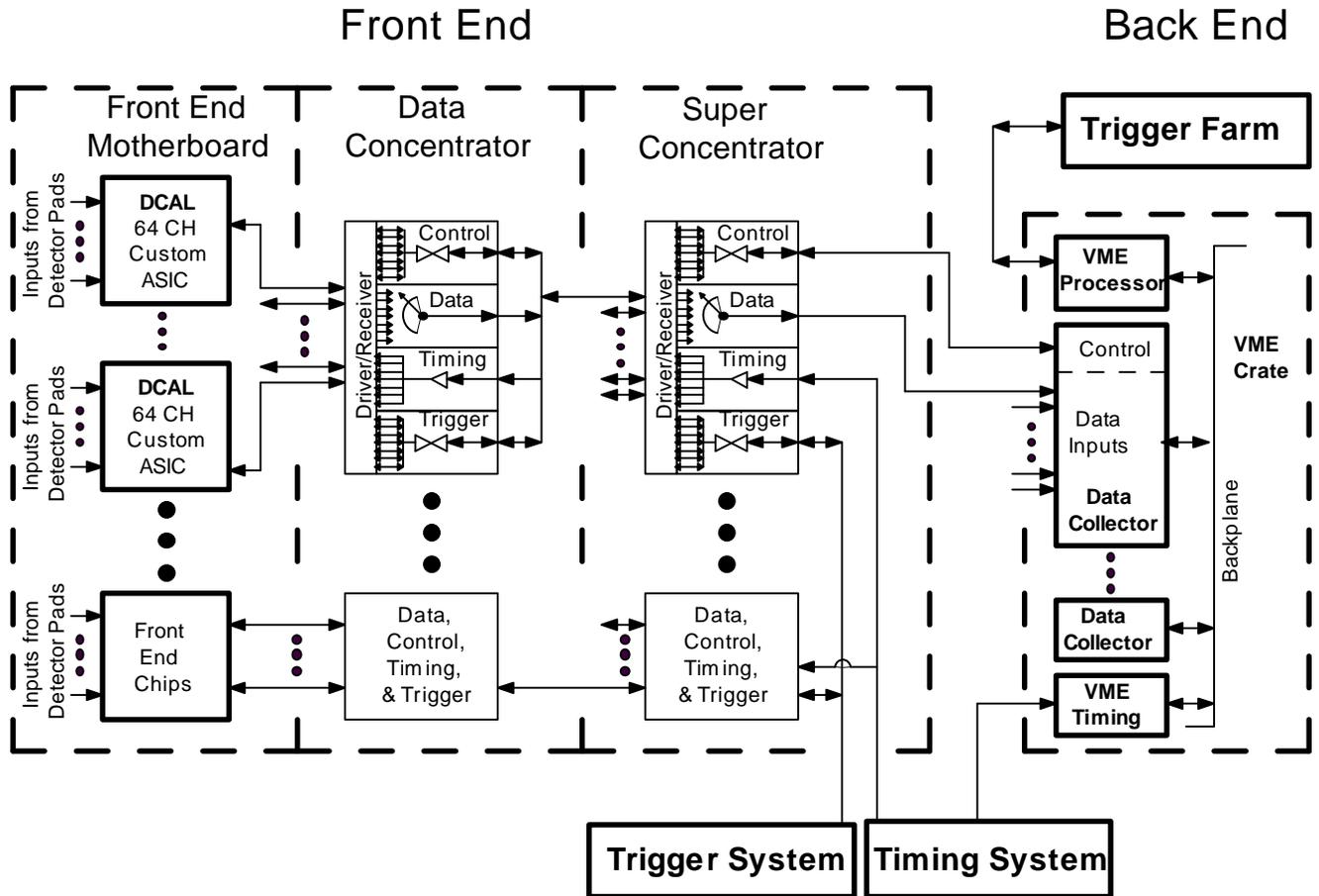


Figure 5.1. Block Diagram of the Readout System

5. System Architecture (Cont.)

The front-end electronics begins with the front-end custom chip, DCAL. It resides on the detector, and handles all of the analog signal processing, formation of data words, and temporary data storage. The chips reside on the Front-End Motherboards, which also have the detector pads as part of them. Data from the DCAL chips is sent to the Data Concentrator, which functions as a multiplexer, concatenating data streams from several DCAL chips into one. There is a further level of multiplexing by the Super Concentrators, which receive data from several Data Concentrators. These components all reside on or near the detector.

The back end is the data acquisition system and trigger farm. When the Super Concentrators have data, it is read by the Data Collectors, which resides in a VME crate. The crate hosts several Data Collectors. The data is written into buffers on these cards, and then read periodically by the VME Processor, which is a single board computer that resides in the crate. Next, the data is read from the VME Processor into the Trigger Farm, which performs event selection.

The other two subsystems, the Timing System and the Trigger System, provide support functions. The front-end electronics needs a small number of clock signals to perform the data acquisition, and these must be synchronized over the entire detector. The front-end electronics also has the capability for several kinds of trigger schemes, and these are provided by the Trigger System.

Each of the four subsystems has one or more sub components. The functionality of each component is described in the sections that follow.

5. System Architecture (Cont.)

5.1. DCAL Custom Chip

The front-end electronics is an integral part of the detector. The primary component is a custom integrated circuit called DCAL. It resides directly on the detector, and handles the entire analog signal processing functions, as well as the collection of data words. Each chip services 64 detector channels. A complete specification of the chip has been written [20], and the design work is in progress. The basic operation is described below.

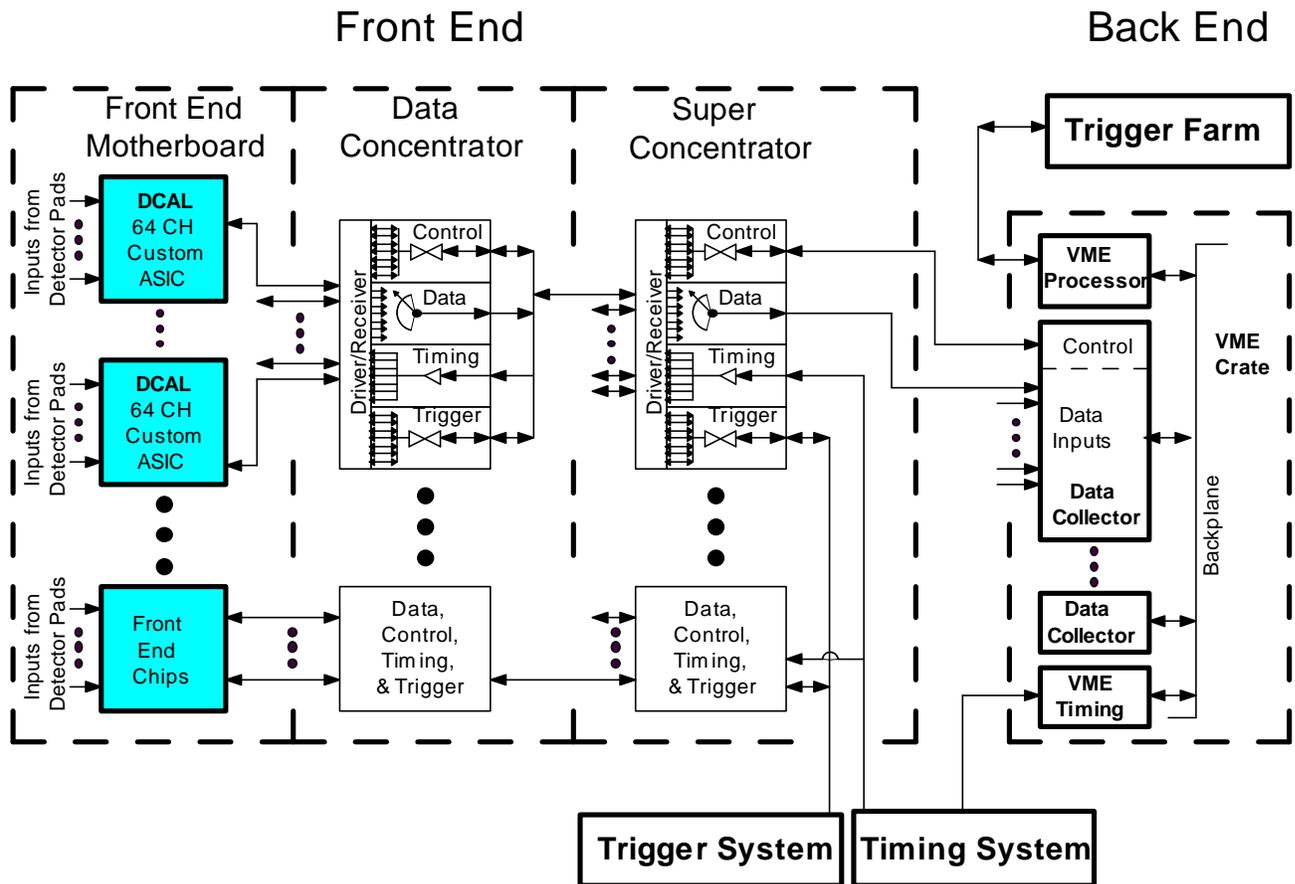


Figure 5.3. Sub-component Description: DCAL Custom Chip

5. System Architecture (Cont.)

5.1. DCAL Custom Chip (Cont.)

A conceptual block diagram of the DCAL chip is shown in Figure 5.3. The operation can be divided into several functional blocks, which will be described.

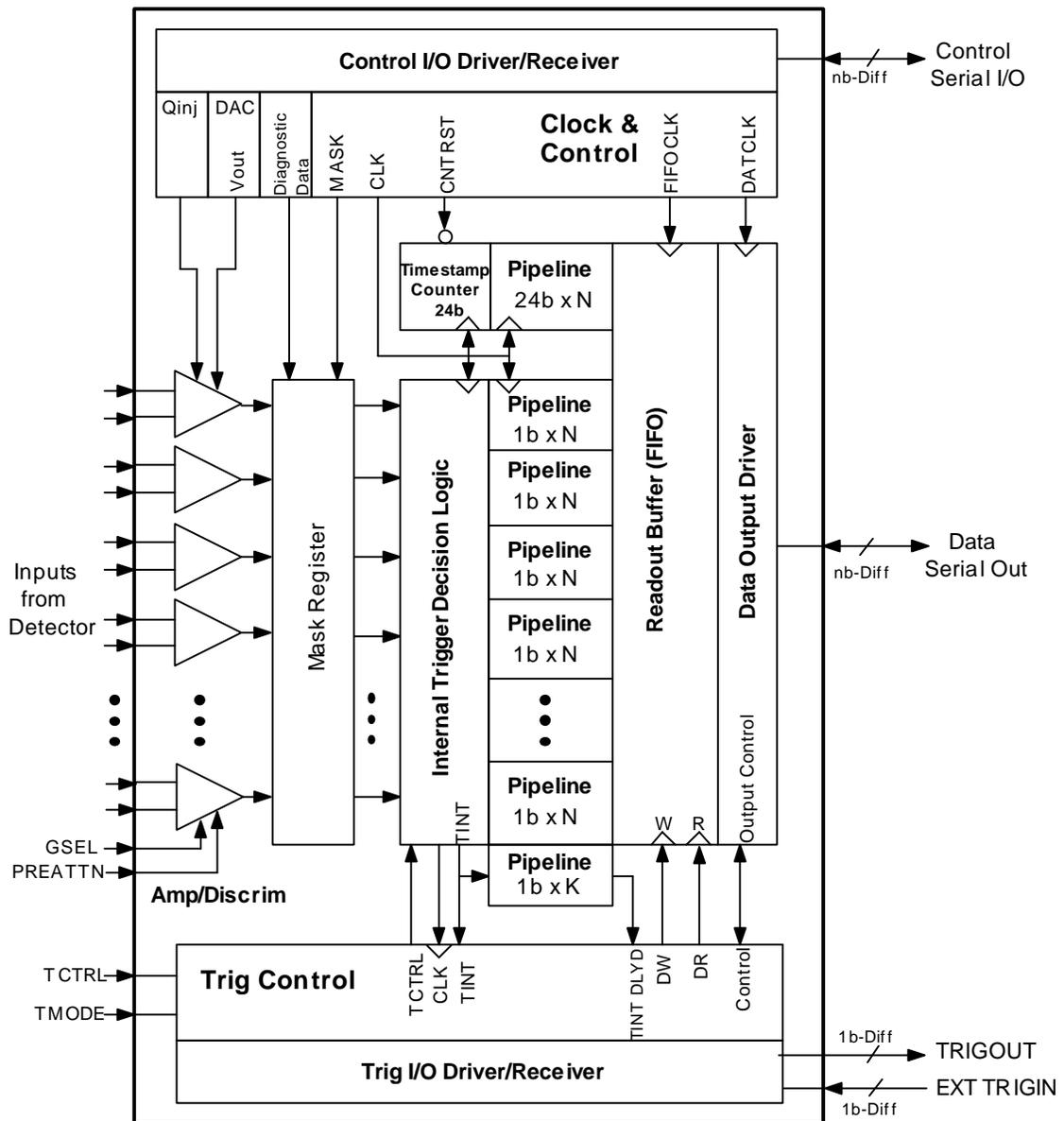


Figure 5.3. Block Diagram of ASIC

5. System Architecture (Cont.)

5.1. DCAL Custom Chip (Cont.)

Signals from the detector are charge or current pulses. When a signal is received from the detector, it is amplified and shaped. It is then passed to a discriminator, where it is compared to a threshold voltage. The discriminator fires when the signal exceeds a programmable threshold level. The discriminators are evaluated using a 10 MHz clock provided to the chip from outside. The discriminators must hold the state of the response until the end of the clock cycle (defined to be the rising edge), so that the state of the discriminators can be latched into the circuitry that follows.

The primary difference between the RPC detector and the GEM detector, from the point of view of the electronics, is that the GEM signals are approximately a factor of 10 smaller in than the RPC signals. See Table 1.1. The amplifier in the chip will have a gain switch, which can switch in an additional X10 gain to accommodate this difference in operation. See Fig. 5.4.

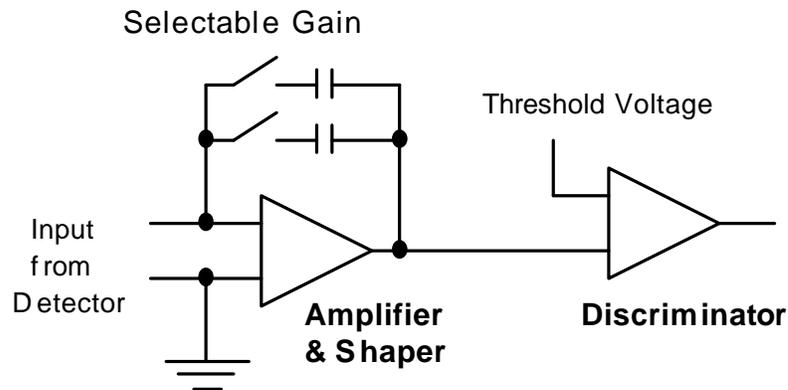


Figure 5.4. Block Diagram Front-End Amplifier

In order to reconstruct hits in later stages of the data acquisition system, the chip shall provide an event ID for each hit. The chip uses the concept of a Timestamp Counter to accomplish this. It is essentially a free-running counter, which is reset periodically (~ once per second.) The counter is advanced by the 10 MHz clock provided to the chip. The counter must have 24 bits in order to count to 10,000,000 in one second.

5. System Architecture (Cont.)

5.1. DCAL Custom Chip (Cont.)

Both the Timestamp Counter bits and the discriminator states define event data. A data word is then 64 bits from the discriminators, and 24 bits from the Timestamp Counter, for a total of 88 bits. They are written into a 20-stage pipeline, and are advanced through the pipeline by the 10 MHz clock. At the end of the pipeline, 2 microseconds in length, a decision must be made whether or not to keep the event data. This decision, called the Trigger Accept, causes the bits to be written into an output buffer, which is configured as a FIFO. From there, circuitry in the chip is activated that reads the event data from the output buffer, and sends it out of the chip. The FIFO is eight stages deep to handle a burst of hits for a short time.

The chip has three ways to capture event data. The first way uses an external trigger, provided from outside the chip. The second uses on-board circuitry to make a trigger decision internally. The circuitry evaluates the states of the discriminators, and arranges for a Trigger Accept to capture the data associated with the trigger decision as the data emerges from the end of the pipeline. The third way is a hybrid of the first two, where a trigger gate is provided, and data is acquired if a discriminator fires in the gate. The choice of trigger mode is set up through the control circuitry. The chip also uses the internal trigger to send a trigger signal out of the chip, for use by an external trigger system. The trigger output and external trigger input are configured as individual output lines.

The data output uses a serial communication protocol. It is a unidirectional link having a point-to-point connection between source (the DCAL chip,) and the receiver. The output driver uses a "data push" protocol. When data is written into the output FIFO and goes "not empty," the data output circuitry is activated. It reads the next available data word from the output FIFO, and sends the bits out of the chip serially. Eight bits are processed at a time, encoded into an 11-bit serial transmission. The extra three bits are used for framing and designation of data type. The bit rate is equal to the 10 MHz clock, or 10 Mbit/sec. It thus takes 12.1 microseconds to send out an entire data block, or 121 times the fundamental clock period. (See Section 6 for a discussion of data rates.) The chip uses LVDS (Low Voltage Differential Signal) for the output drive. LVDS is a good choice for processing digital signals in an analog environment where noise is a concern.

5. System Architecture (Cont.)

5.1. DCAL Custom Chip (Cont.)

The chip requires two precision timing signals to function. The first is the 10 MHz clock. This is treated as a precision dedicated signal, to maintain synchronization across the detector. The chip receives the clock using LVDS, like the serial data transmission. All clocking functions in the chip use this signal, including the serial transmission. The second precision timing signal is Counter Reset. Although it is active only once per second, it must be timed to within one clock cycle, as it is used to synchronize timing of all chips in the system. This signal is also received by the chip using LVDS.

Control functions of the chip are handled by a dedicated interface called the Control Communication Link. It also uses a serial communication protocol, although it may utilize several signal lines. This link has several functions. The chip needs an on-board DAC to generate the threshold voltage for the discriminators. An important control feature is the ability to mask off bad channels. This is needed to prevent noisy channels from co-opting the readout bandwidth. This is incorporated using a simple AND gate with the output of each discriminator. The masking is accomplished by loading a “Mask Register” through the control link.

Another control function is the control of charge injection. The ability to inject charge is useful for testing the basic signal processing circuitry. By providing the capability of injecting charge at a precise time over many parts of the system, the synchronization of the chip counters can be tested. If the charge injection circuitry were further enhanced to incorporate a DAC, it would be possible to study the sensitivity of the discriminators as a function of threshold voltage setting versus value of the charge injected. Clearly the latter features represent a more complex design, and are open for discussion.

One aspect of the Control Link not shown in the block diagram is that each chip on a motherboard will have a unique address, and the communication of control data will use a bus structure. The addresses will be set using resistors or jumpers on the motherboard next to each chip. Each control word will have an address encoded in it, to identify which chip is the intended destination. This reduces then number of I/O lines on the motherboard. See Section 5.3.

5. System Architecture (Cont.)

5.2. Front-End Motherboard

The Front-End Motherboard is the host for the DCAL chip, and must provide the means for getting both analog and digital signals into and out of the chip. See Fig. 5.5. As described in Section 2, the DCAL chips reside directly on the detector. Specifically, the pads of the detector are etched into one side of the printed circuit board of the motherboard. This dual functionality makes the design of the motherboard complex, and merits the designation as a primary sub-component. The functionality of the motherboard is described below.

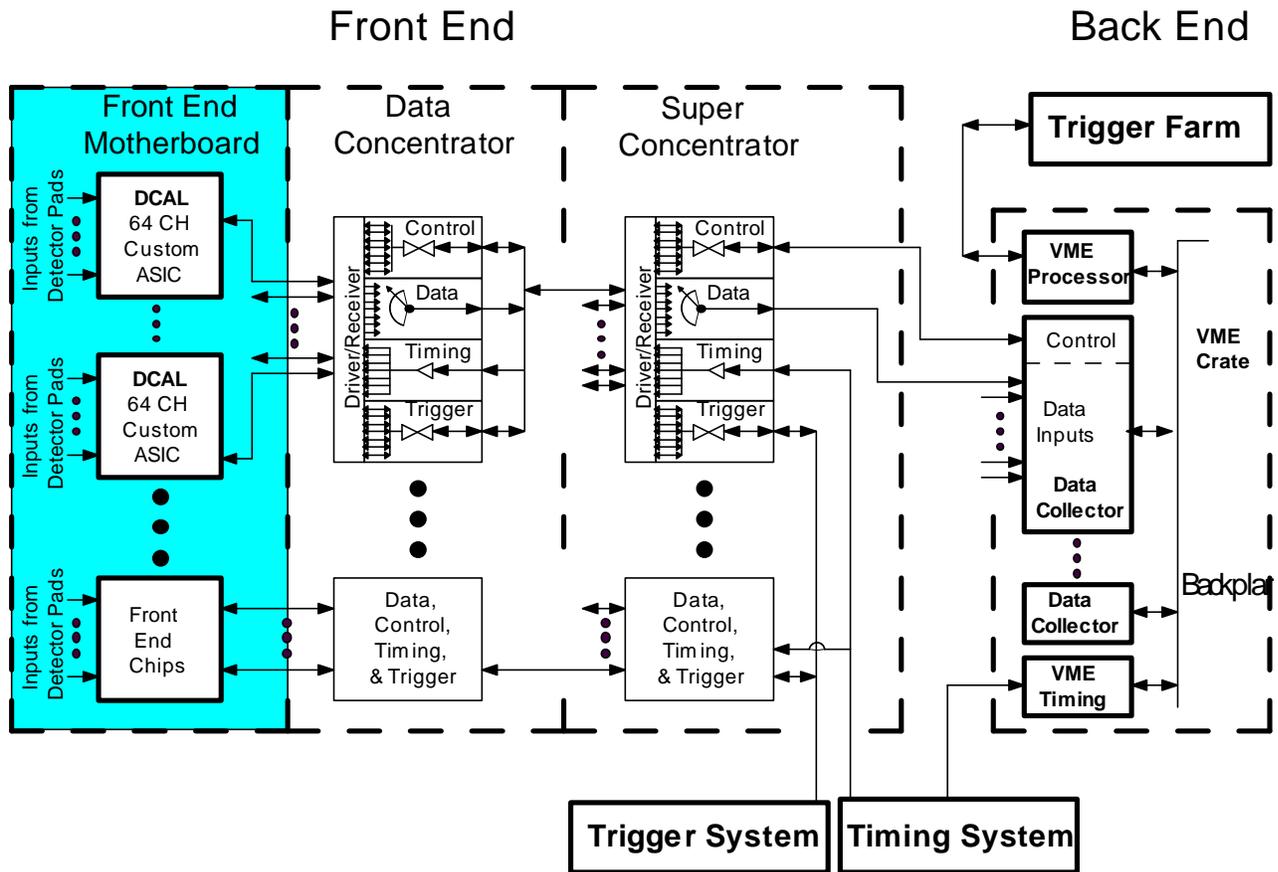


Figure 5.5. Sub-component Description: Front-End Motherboard

5. System Architecture (Cont.)

5.2. Front-End Motherboard (Cont.)

The Front-End Motherboard has two primary functions. The first is to provide the readout pads for the RPCs. The second is to host the DCAL chips. This includes: the routing of signals from the pads of the detector to the input pins of the chips; the distribution of power and ground to the chips; the distribution of clock and control signals; and the routing of the output signal lines from the DCAL chips to the receivers (to be described in Section 5.3.) This makes the design of the motherboard complex. See Fig. 5.5. A particular concern is noise pickup from digital signals in the printed circuit board by the sensitive front-end amplifiers. For this reason, the printed circuit board must have several layers of ground planes to shield against noise. The DCAL chip is designed to use LVDS signals for the transmission of digital signals into and out of the chip, which will also provide a measure of noise reduction.

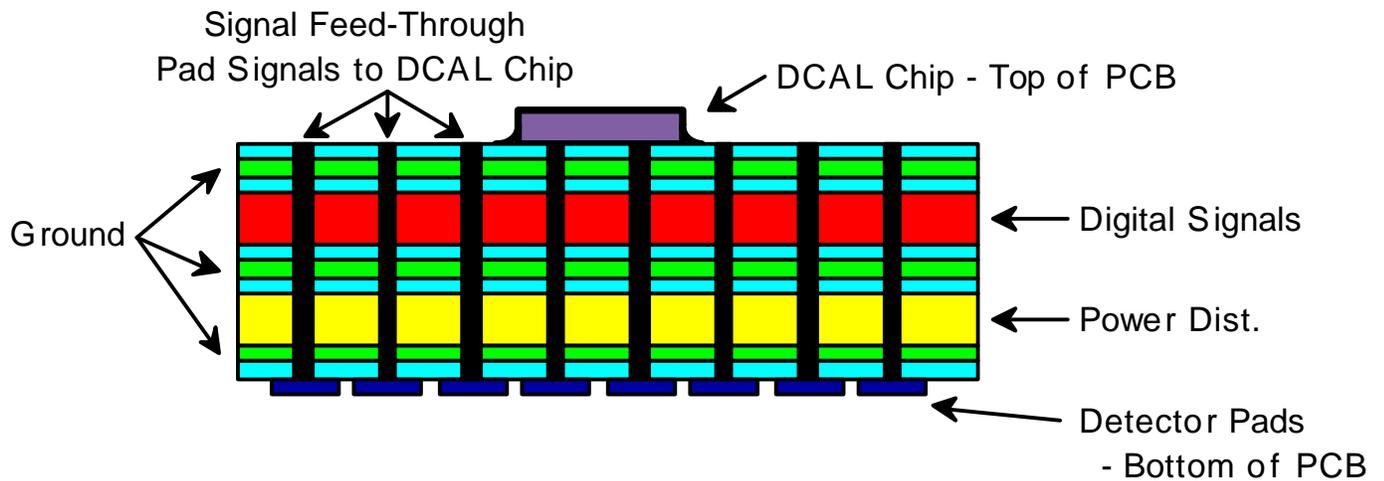
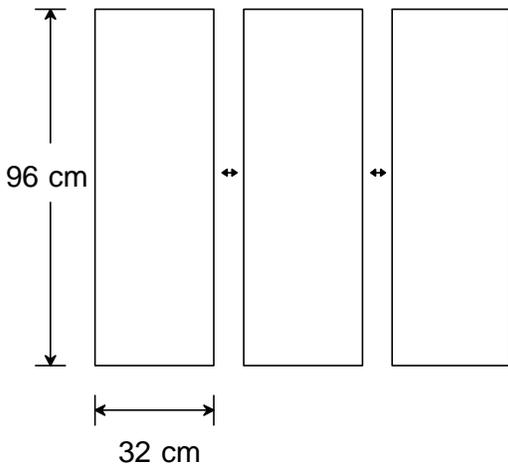


Figure 5.5. Conceptual Cross Section of the Front-End Motherboard

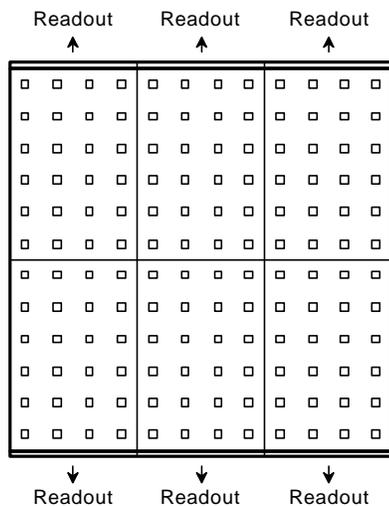
5. System Architecture (Cont.)

5.2. Front-End Motherboard (Cont.)

As described in Section 2, the prototype detector will be built using planes that are one meter-square in area. The active area of a plane will actually be 96-centimeters by 96-centimeters. The planes will be constructed from three chambers that are 32-centimeters by 96-centimeters. They are butted next to each other to form a plane. The motherboards lie on top of the chamber. Currently, it is envisaged that each motherboard would cover one-half of a chamber. See Fig. 5.6. Each motherboard is 32-centimeters by 48-centimeters, and hosts 24 DCAL chips, arranged in a 4-by-6 array as shown. (It may be desirable or necessary to implement a different physical arrangement for the motherboards given practical considerations, such as manufacture-ability, connector density, mounting, etc.)



3 Chambers Form a Plane



2 Front End Boards/Chamber
6 Front End Boards/Plane
24 ASICs/Board
1536 Channels/Board
144 ASICs/Plane
9216 Channels/Plane

Figure 5.6. Arrangement of Front-End Motherboards on a Plane

5. System Architecture (Cont.)

5.2. Front-End Motherboard (Cont.)

The digital signals on each motherboard are routed to the outer edge, where services for the motherboard are accessible, including receivers and drivers for digital signals (data, clock, control, and trigger), and power. See Fig. 5.7. This interface will be described in Section 5.3.

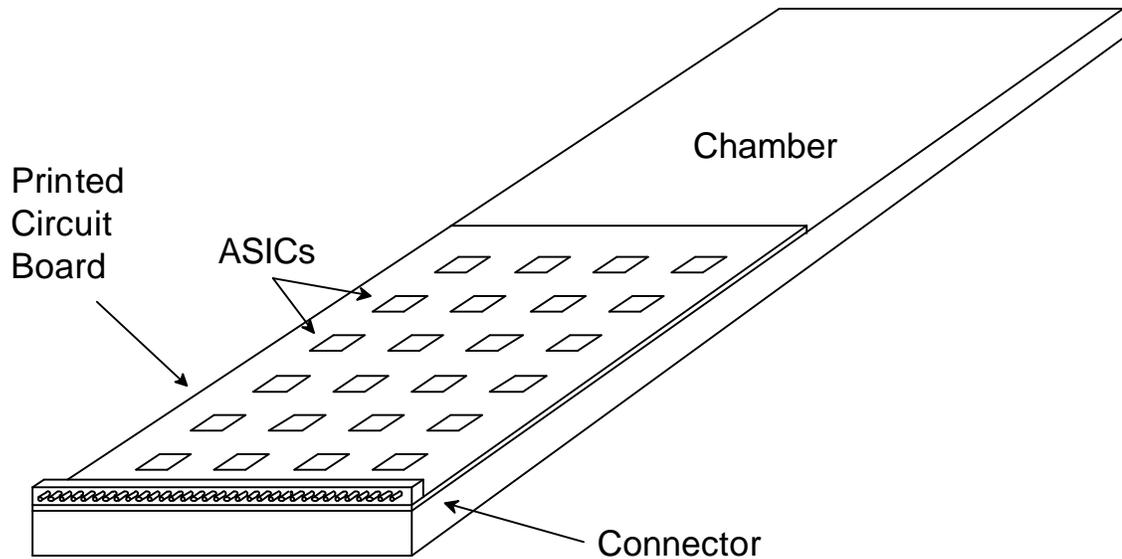


Figure 5.7. Front-End Motherboard Interface

As described in Section 5.1, the control signals will be routed to the front-end chips using a bus structure. Each chip will have a unique address on the board, which is set using jumpers or resistors that are mounted on the motherboard. The motherboard may also have other passive components around the chips, such as bias resistors and bypass capacitors.

5. System Architecture (Cont.)

5.3. Data Concentrator

The Data Concentrator is an intermediate interface between the front-end chips and the data acquisition system. The primary purpose is to concatenate serial data streams from several front-end chips into one, thereby reducing the amount of instrumentation required in the back end of the system. The Data Concentrator also provides an interface for the control, timing, and control functions needed by the chips. The functionality is described below.

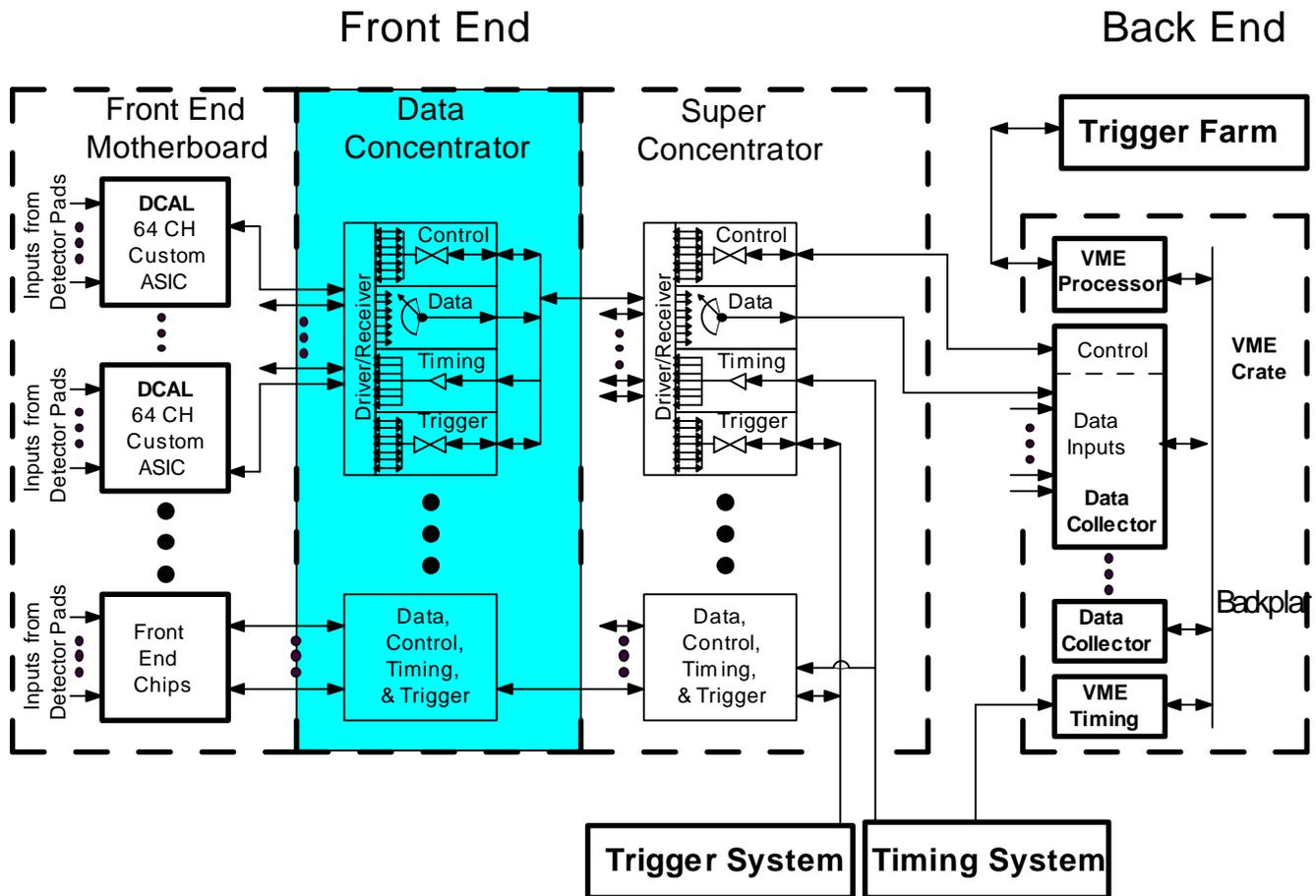


Figure 5.8. Sub-component Description: Data Concentrator

5. System Architecture (Cont.)

5.3. Data Concentrator (Cont.)

A conceptual block diagram of the Data Concentrator is shown in Figure 5.9. The operation can be divided into several functional blocks, which will be described.

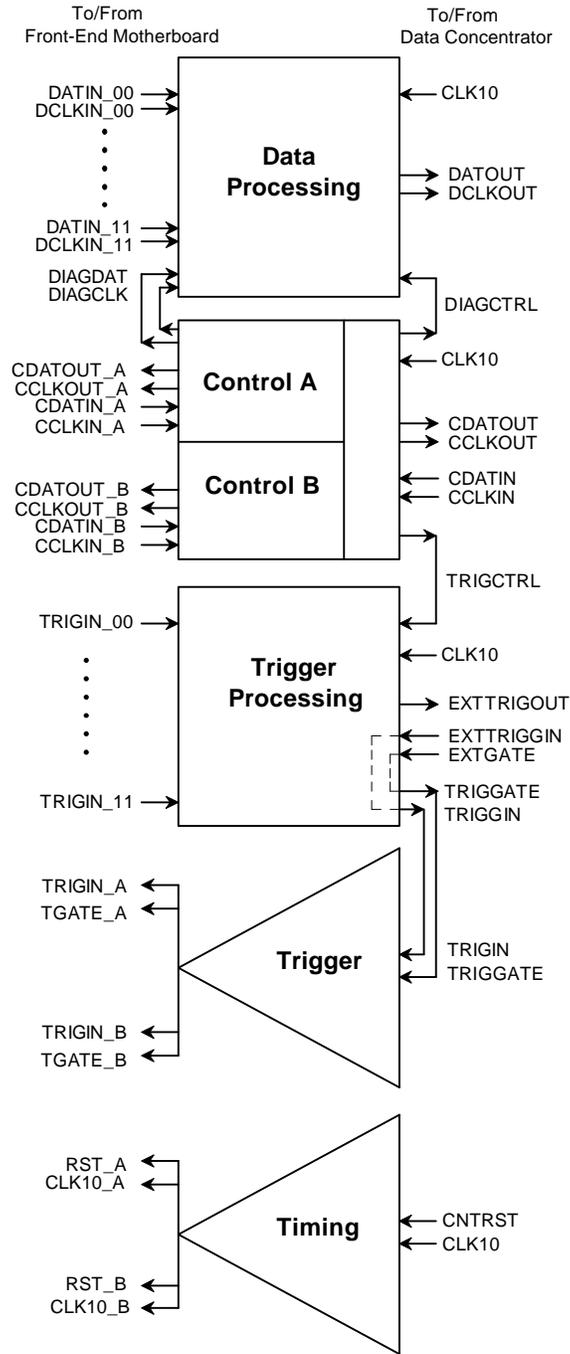


Figure 5.9. Block Diagram of the Data Concentrator

5. System Architecture (Cont.)

5.3. Data Concentrator (Cont.)

The primary function of the Data Concentrator is to concatenate the serial data streams from several front-end chips into a single output. This is needed to reduce the number of back-end data processors. The Data Processing block performs this function. For reasons that will be described later, the Data Concentrator will service 12 front-end chips. In the simplest form, it can be thought of as a data multiplexer, where several input data streams are fed into a single output stream using an efficient algorithm. See Fig. 5.10. Because the data comes in spurts, there needs to be buffering of data on the input side. In order to accommodate possible output data flow rate problems, there is buffering on the output side as well. Because the Data Processing block mixes data from several chips in a non-uniform way, another function of the circuitry is to append an address onto each data word. Both the input bit rate and the output bit rate are 10 Mbit/Sec, equal to the fundamental clock frequency. (See Section 6 for a discussion of data rates.) Both the input and output transmissions use LVDS.

Data Processing Block

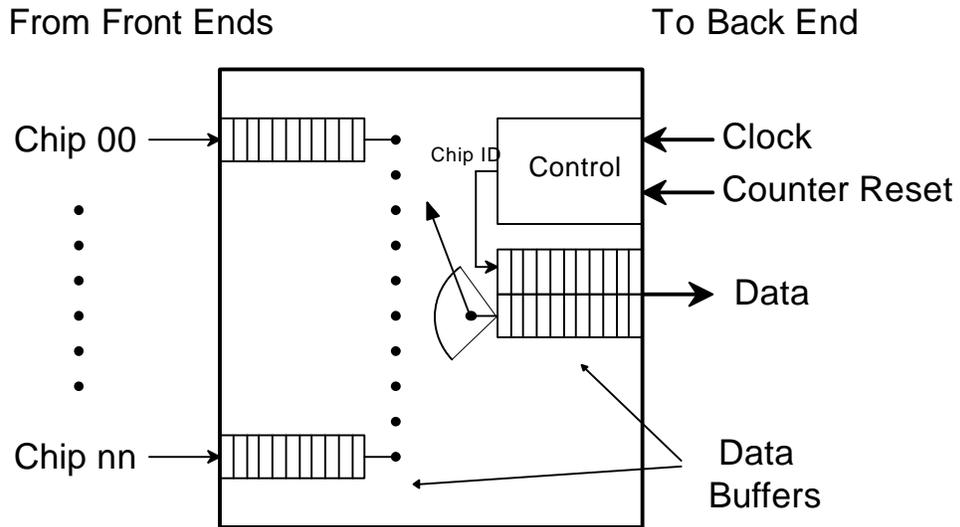


Figure 5.10. Data Processing Block of the Data Concentrator

5. System Architecture (Cont.)

5.3. Data Concentrator (Cont.)

There are two kinds of connections between the Data Concentrator and the Front-End Board. The Data Processing block is an example of point-to-point connections. Each front-end chip has a dedicated connection for the serial output data to the Data Processing block on the Data Concentrator. See Fig. 5.11. The second kind of connection is a bus connection, as shown in Fig. 5.12. This is used for many of the other functions of the Data Concentrator, including clock distribution, control, and some of the triggering.

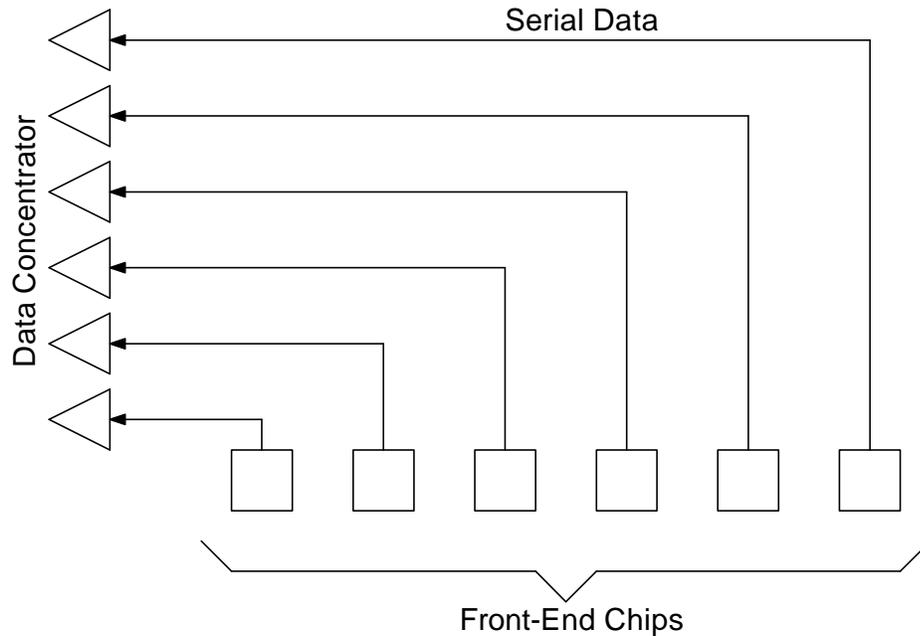


Figure 5.11. Point-to-Point Connections

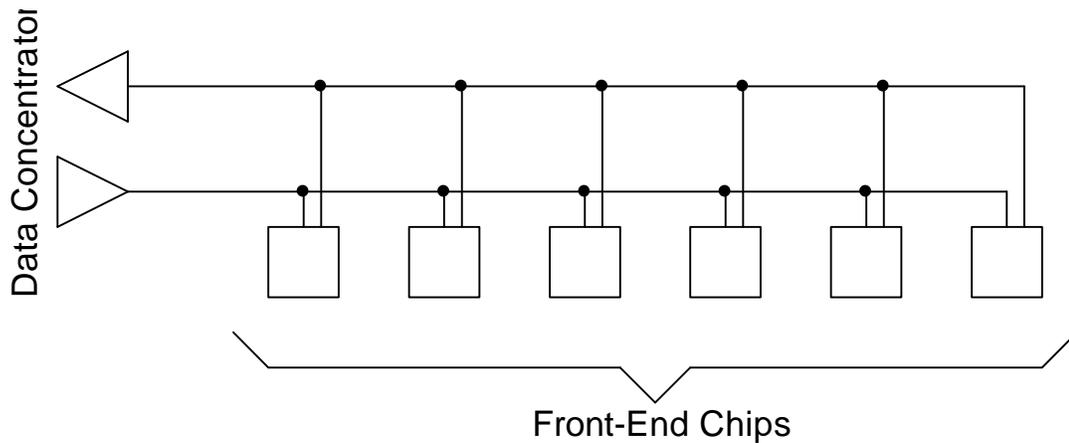


Figure 5.12. Bus Connections

5. System Architecture (Cont.)

5.3. Data Concentrator (Cont.)

The Data Concentrator must perform several ancillary functions to support the operation of the front-end electronics. One function involves the control of the front-end chips. Each chip has several registers that must be initialized prior to data taking. The communication protocol is serial. Each control word has an address encoded in the control word, which allows the use of a bus structure as shown in Fig. 5.12. Because multiple front-end chips reside on this bus, each control command must include an address to identify the target chip. Data processing and control are entirely separate paths in this architecture, as will be described in Section 5.5. It is assumed that there are no time-critical functions in the control path, so that synchronization between chips is not necessary. The Data Concentrator has no intelligence itself for the control functions. Instead, it is entirely pass-through, and has a separate connector that provides input from the intelligent part of the system. Referring to Fig. 5.9, notice that the control block is divided into two sections, A and B. As will be described shortly, each Data Concentrator services 12 front-end chips, which are arranged in two columns of six chips each. Because the control functions use a bus protocol, it is convenient to have two control buses, one for each group of six chips. The A block and B block provide this service. The signal transmission may use either LVDS or CMOS (to be determined.)

The front-end chips need two timing signals to function. One is the 10 MHz clock, called CLK10, which is the fundamental timing structure for the system. The second is a reset for the Timestamp Counter that resides in each front-end chip, and is called CNTRST. Each hit is tagged with a Timestamp, which is implemented using an internal counter that is clocked with the 10 MHz clock. In order to reconstruct events, it is important that the counters in all chips are synchronized, and this is provided by Counter Reset. Both signals are regarded as precision timing signals, and come into the Data Concentrator from an external timing system. The signal CLK10 is also used by the Data Concentrator for the various clock and timing functions needed for the operation, including the serial data transmission. Both CLK10 and CNTRST are distributed to the front-end chips using the bus structure of Fig. 5.12. The signal transmission uses LVDS.

The Data Concentrator also handles certain trigger functions. These functions will be described in Section 5.8. This functionality is mostly separate from the other parts of the Data Concentrator. The system design has a high degree of flexibility for triggering, and the use of programmable logic at this point in the architecture is desirable to enhance capabilities.

5. System Architecture (Cont.)

5.3. Data Concentrator (Cont.)

The physical arrangement of Data Concentrators is shown in Fig. 5.13. As described earlier, there are six front-end boards for each plane of the detector. In the current plan, each front-end board has two sets of Data Concentrators. For the purposes of this description, it will be assumed that two Data Concentrators service one Front-End Motherboard. It may be desirable for the design of the printed circuit board for the Data Concentrator to incorporate both sets.

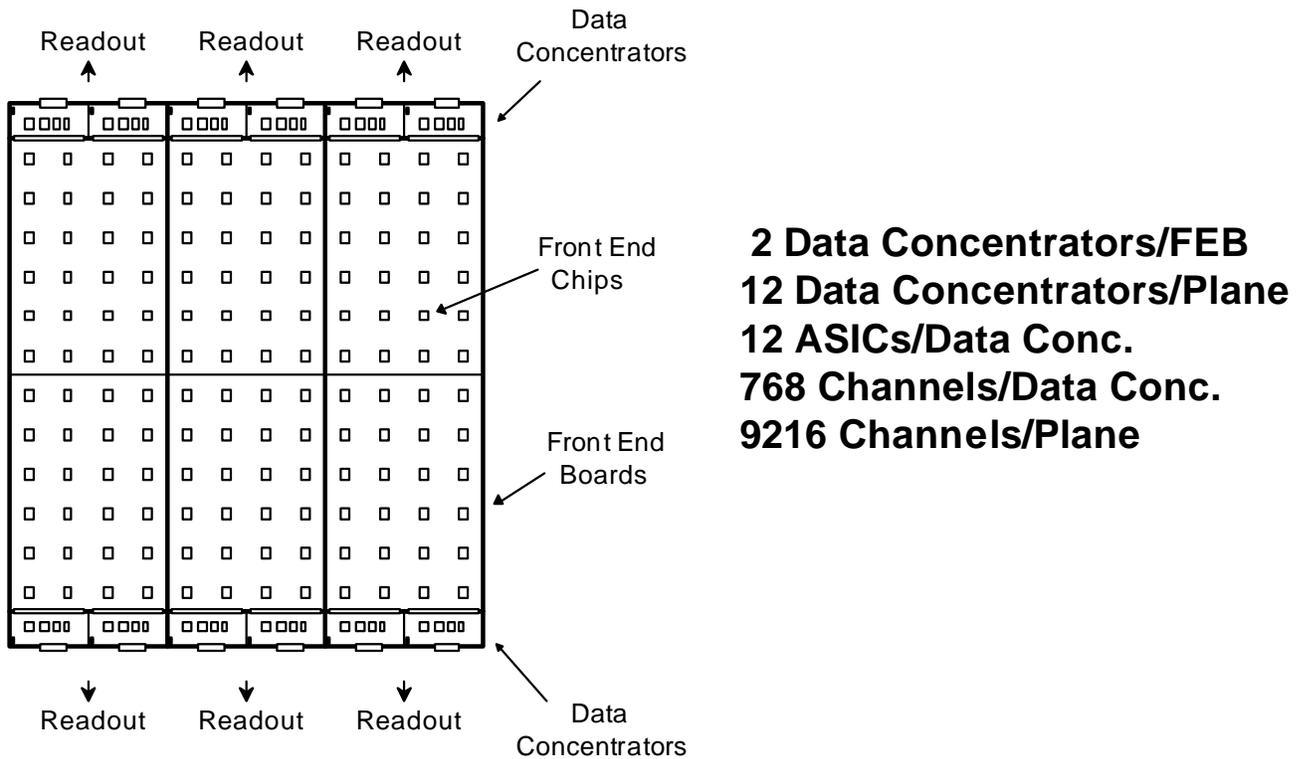


Figure 5.13. Arrangement of Front-End Boards and Data Concentrators

5. System Architecture (Cont.)

5.3. Data Concentrator (Cont.)

As described in Section 5.2, the digital signals from the motherboards are routed to the outer edge. The Data Concentrator boards plug in here. See Fig. 5.14. The interface must have a connector, since the motherboard is part of the detector, and it must be possible to service the Data Concentrators.

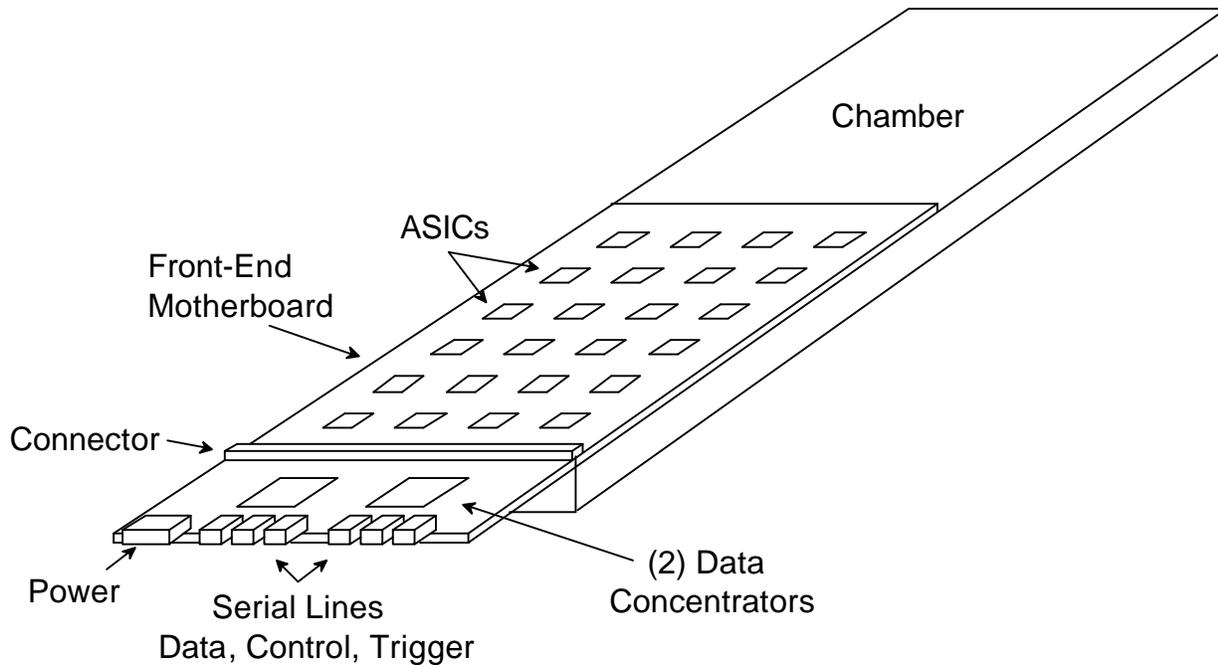


Figure 5.14. Interface of Front-End Boards and Data Concentrators

An additional function of the Data Concentrator not shown in the block diagram of Fig. 5.9 is to provide power and ground to the front-end boards. The front-end chips require 2.5V. The circuitry on the Data Concentrator is not specified here, but will have power requirements as well. The Data Concentrator board will receive power from an external power supply, both for use with on-board circuitry and for the front-end boards.

5. System Architecture (Cont.)

5.4. Super Concentrator

As indicated in Section 4, the event rates and noise rates for this detector are very low. With the architecture described thus far, there is excess bandwidth available for additional data transmission. The physical arrangement of front-end chips and Data Concentrators described previously is based on reasonable assumptions about connectors, signal line density, and I/O of programmable logic on the Data Concentrators. In order to achieve better use of the bandwidth, as well as to reduce costs with the back end electronics, the concept of a Super Concentrator is used, as shown in Fig. 5.15. The functionality is described below.

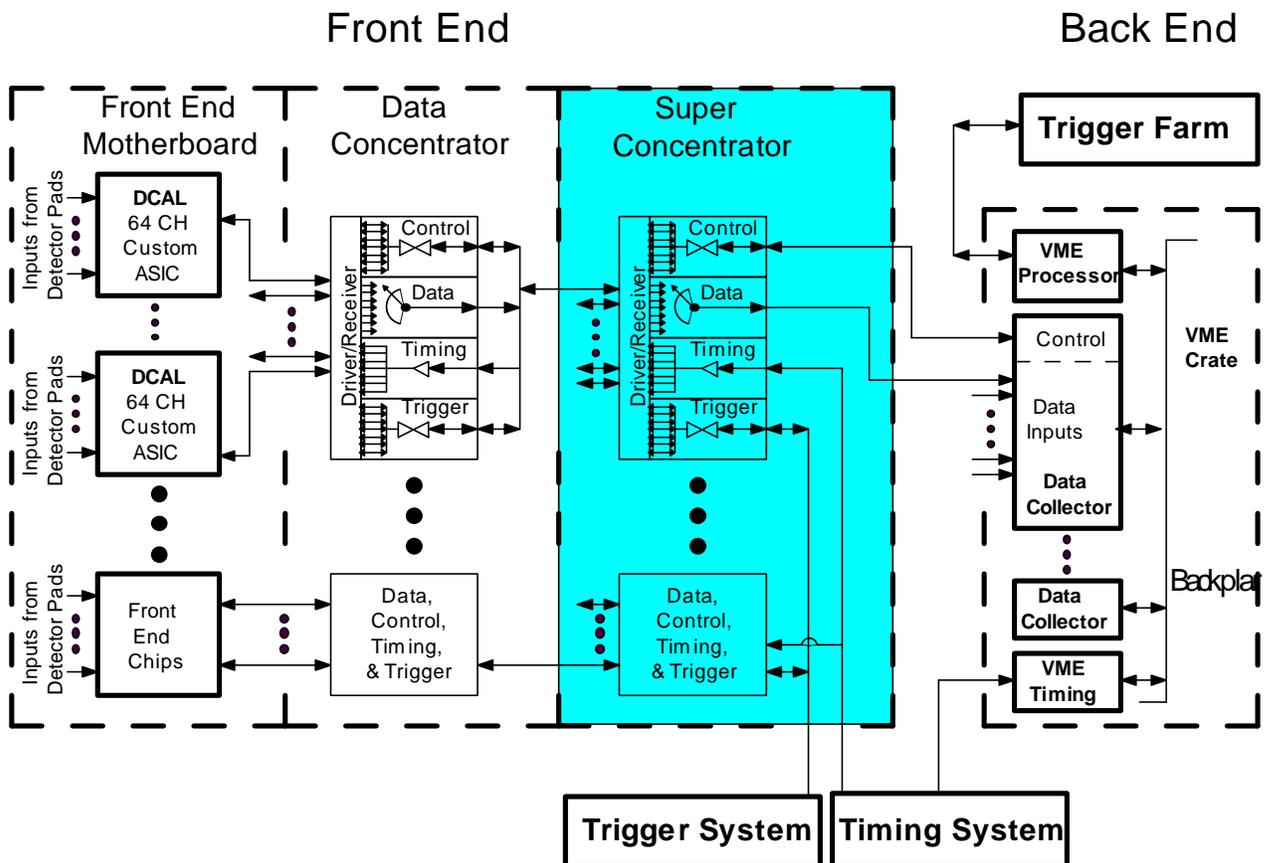


Figure 5.15. Sub-component Description: Super Concentrator

5. System Architecture (Cont.)

5.4. Super Concentrator (Cont.)

A conceptual block diagram of the Super Concentrator is shown in Figure 5.16. Notice that it is very similar to the Data Concentrator.

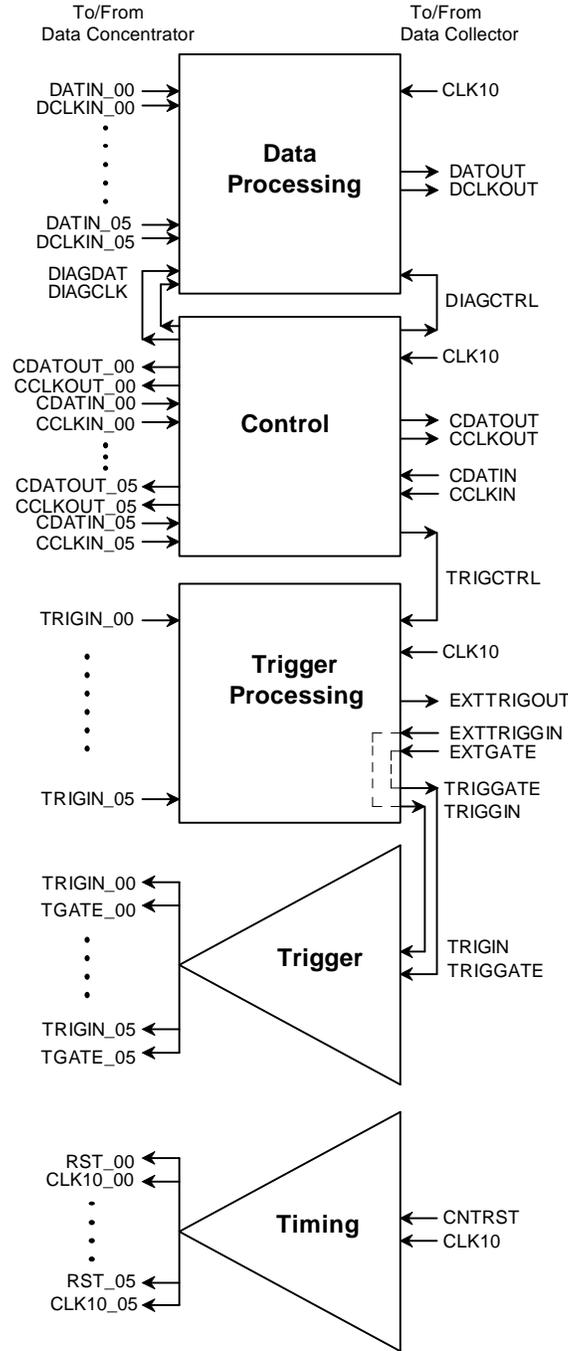


Figure 5.16. Block Diagram of the Super Concentrator

5. System Architecture (Cont.)

5.4. Super Concentrator (Cont.)

At the most basic level, the Super Concentrator multiplexes all of the functions of the Data Concentrator to an additional level. While each Data Concentrator services 12 front-end chips, the Super Concentrator services six Data Concentrators, or one side of a plane. The multiplexing of data streams is the same function as the Data Concentrator. Since an additional level of multiplexing is used, the Super Concentrator must also append an address onto each stream of data, to identify the source. The input bit rate from the Data Concentrators is 10 Mbit/Sec. Even with the high-level of multiplexing at this point in the architecture, the data rates are expected to be low enough to permit using 100Mbit/Sec on the output as well. See Section 6. LVDS is used on the input. While the data rates are into the Super Concentrator are low, it is desirable to use optical fiber to send data from the Super Concentrators to the Data Collectors. This helps to reduce noise coupling and ground loops between the front end and the back end. With optical fiber, it is desirable to use a commercial serial transceiver.

The way that the Super Concentrator handles the multiplexing of control functions is somewhat different from the Data Concentrator. The Data Concentrator has two output paths, A and B, whereas the Super Concentrator has six. Other than that, the functionality is identical. The control words must have an address associated with them to identify which Data Concentrator is the destination. The Super Concentrator must pick off that part of the control word and suppress it after the routing is determined, to avoid sending extraneous information to the front-end chips. Part of the address is also which front-end chip is the destination. That part of the address must stay intact. The signal transmission may use either LVDS or CMOS (to be determined.)

The transmission of the clock signals is a straight fan-out. No additional signal processing by the Super Concentrator is needed. Since the system architecture relies on having these signals synchronized over the entire system, it is important that care be exercised to minimize excessive random latencies in the buffering and driving of these signals. The signal transmission uses LVDS.

The handling of trigger signals will be described in Section 5.8.

5. System Architecture (Cont.)

5.4. Super Concentrator (Cont.)

A possible configuration for the physical arrangement of Super Concentrators is shown in Fig. 5.17. As indicated, the side of a plane offers a convenient place both physically and in the architecture for the Super Concentrator. The cables between the Data Concentrators and the Super Concentrator may be somewhat long if need be, although the serial transmission is 100 Mbit/Sec, and high-quality cable will be needed if the cables are long. As shown in the figure, a possible arrangement is to mount the Super Concentrator above the Data Concentrators, with the data cables folded between them. The Super Concentrator boards will need power, which is probably best made by a dedicated connection to an external power supply like the Data Concentrators. Again, implementation constraints may force the actual physical arrangement to be modified.

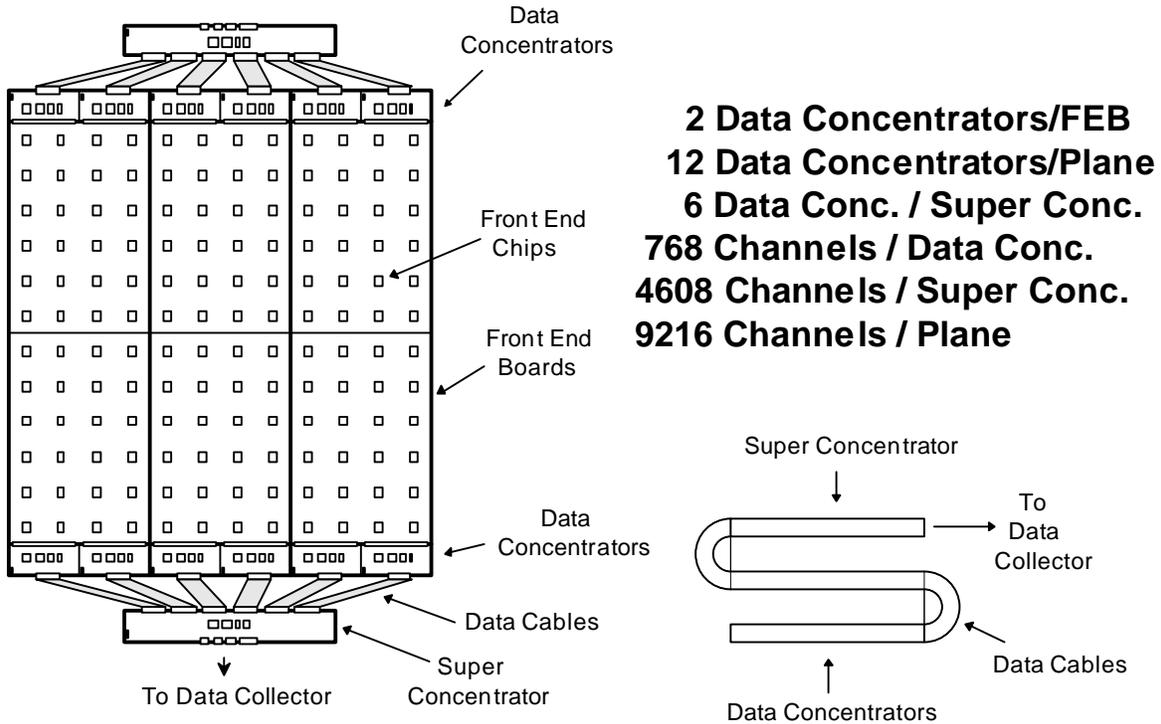


Figure 5.17. Arrangement of Front-End Boards, Data Concentrators, and Super Concentrators

5. System Architecture (Cont.)

5.5. Data Collector

Data sent from the front-end electronics is received and processed by the first component in the back-end electronics, the Data Collectors. This component receives the serial data streams from the Super Concentrators, and stores it in buffers pending readout. The Data Concentrator is also the interface for control information to and from the front-end electronics.

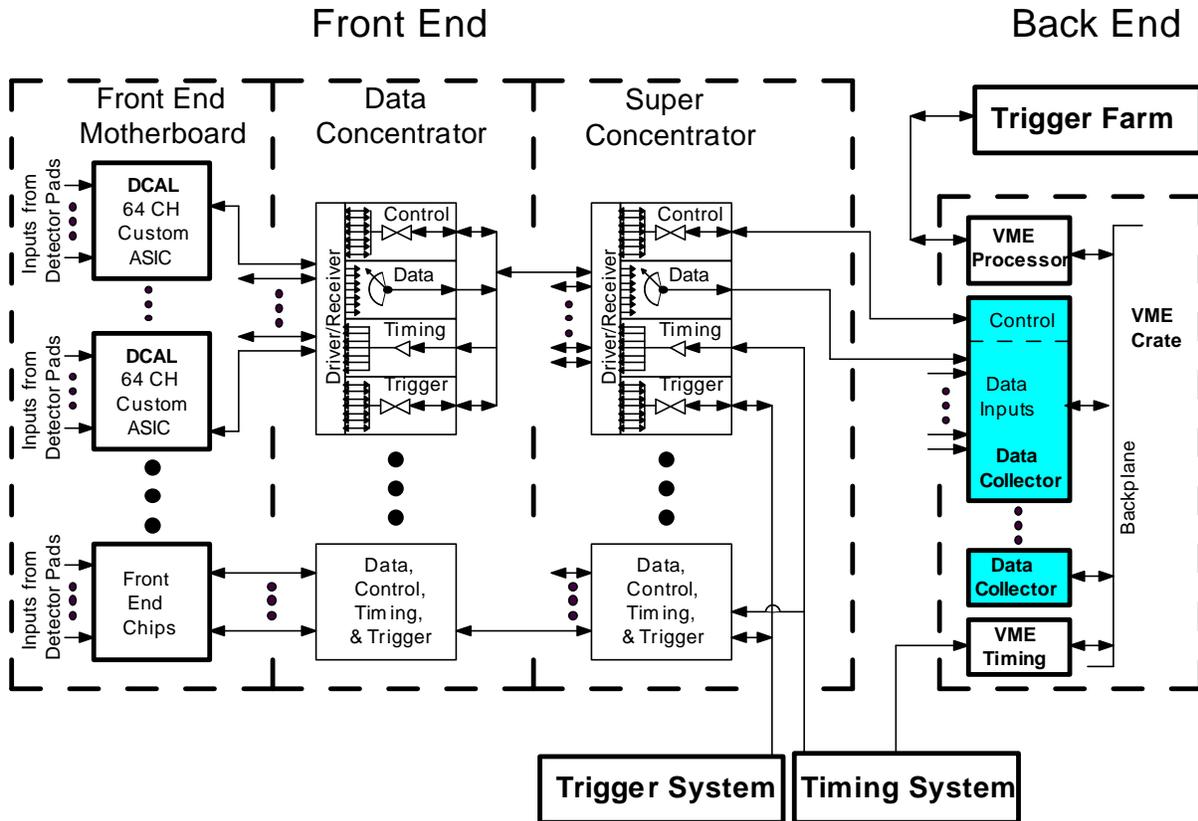


Figure 5.18. Sub-component Description: Data Collector

5. System Architecture (Cont.)

5.5. Data Collector (Cont.)

Data from the front-end electronics is received by the Data Collectors using dedicated, serial communication links. The input bit rate is 10 Mbit/sec, using optical fiber as the transmission medium. The data acquisition is "data-driven," so that data is pushed from the front-end electronics into the Data Collector without being requested. When the data is received, the Data Collector first converts it from a serial bit stream into bytes and data words. As described earlier, as data passes through each part of the system, the sub-component that receives the data appends an address onto the data. The Data Collector receives data streams from several Super Concentrators, so it must add an address to identify the source.

Each data word is then stored in one of two readout buffers on the Data Collector, making it available for readout. The arrangement of two buffers allows the Data Collector to write data to one buffer while the other is being read by a readout computer (see Section 5.6.) The buffers change state in a "ping-pong" fashion at a programmable rate, controlled by the Timing System (see Section 5.7.) The readout at this point in the architecture is dead-timeless, within certain constraints. For a given maximum data rate, the buffers must be large enough and the buffer-swap rate fast enough so that the buffers do not fill completely. This assumes that the readout of the buffers can be achieved within the buffer-swap period. The buffer-swap rate can be adjusted to optimize the data acquisition. Refer to Section 6 for a discussion of data rates.

5. System Architecture (Cont.)

5.5. Data Collector (Cont.)

The Data Collector is envisaged to be implemented as a 9U x 400 mm VME card, as shown in Fig. 5.19. The card has high-speed, dedicated communication links to receive the serial data streams from the Super Concentrator. Each Data Collector receives data from twelve Super Concentrators via front panel connectors. The input streams are uni-directional, and dedicated to writing event data. The control path for the passing control information to the front-end electronics is implemented using the J3 connector of the card as shown. An Auxiliary Card is used to fan out connections for the control to the twelve Super Concentrators serviced by the card.

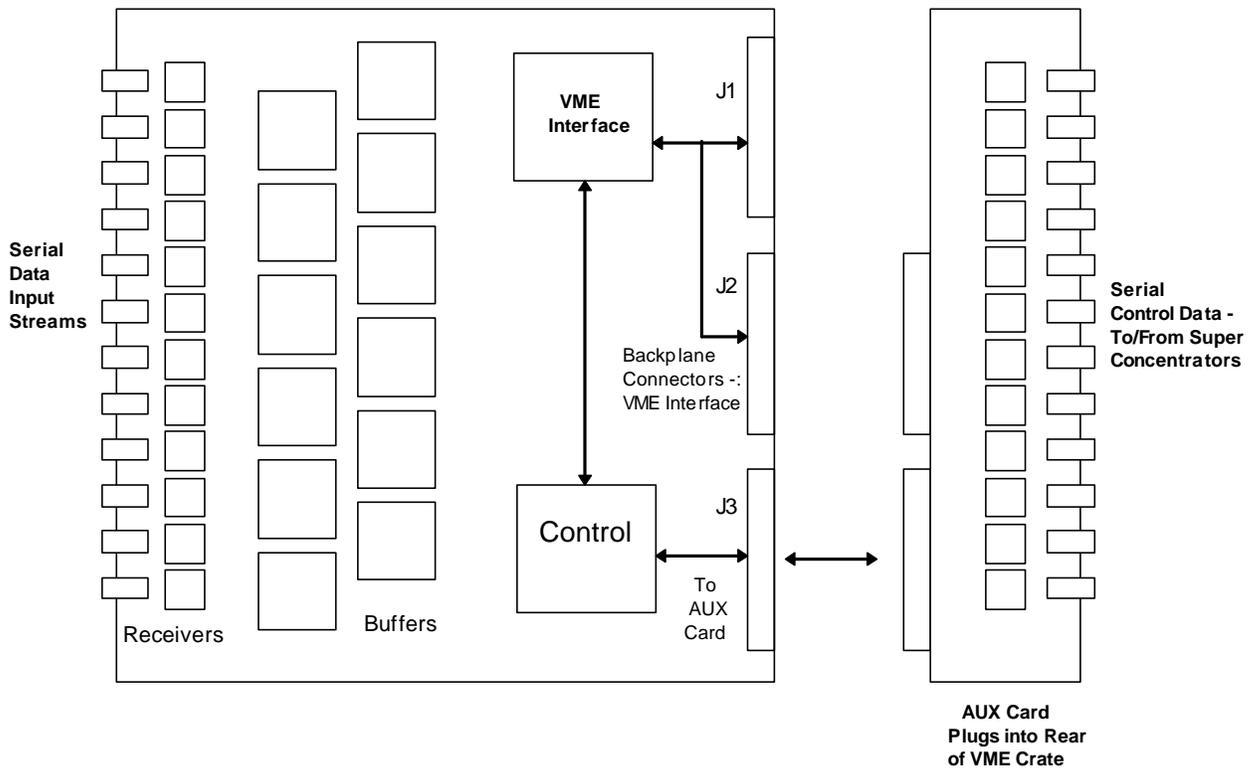


Figure 5.19. Conceptual Design of the VME-based Data Collector

For this design, each Data Collector services twelve Super Concentrators, which corresponds to 72 Data Concentrators, or 864 front-end chips, or 55,296 detector channels.

5. System Architecture (Cont.)

5.6. VME Single-Board Computer & Readout

Data is stored in buffers on the Data Collectors. It must be read from each one, brought together as a data record, and passed to the Trigger Farm for event selection and reconstruction. This is the job of the VME computer. It resides in the VME Crate, as shown in Fig. 5.20.

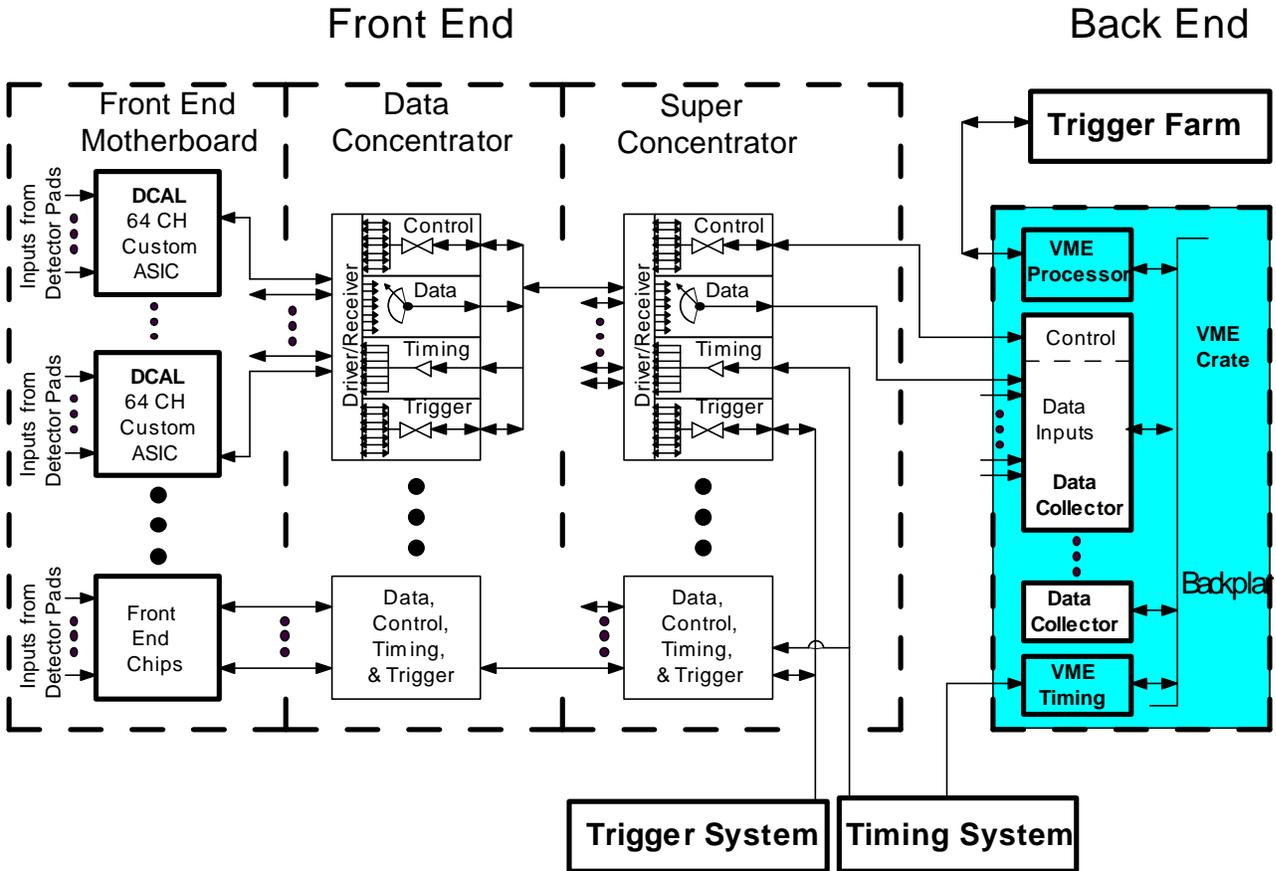


Figure 5.20. Sub-component Description: VME Computer & Crate

5. System Architecture (Cont.)

5.6. VME Single-Board Computer & Readout (Cont.)

The VME Computer can access each Data Collector in that crate through the normal VME communication protocol. The primary function of the VME computer is to periodically read data from the buffers of all of the Data Collectors in the VME Crate, and then send it to the Trigger Farm. As mentioned earlier, the timing for accessing data from the buffers on the Data Collectors is controlled by the VME Timing Module that also resides in the crate, which in turn is controlled by the timing system. The VME Timing Module issues two Interrupt Service Requests (ISRs) in an alternating fashion at a periodic rate. These are used by the Data Collectors to change the active buffer for writing data. The ISRs also alert the VME Processor that new data is available for reading. The VME computer executes an algorithm that reads data from each Data Collector, saving it in local memory. Each read cycle forms a "block" of data. An efficient protocol such as block transfers on the VME bus can be used. As before, the destination component must add an address to the data, indicating the source. No sorting or event selection is done by the VME Computer. At a selected time, such as (N) ISR periods, the data stored in local memory is sent to the Trigger Farm. A convenient period might be one second, where it is arranged for there to be a whole number of ISR periods in one second. The data sent to the Trigger Farm is called a "Time Frame." Using the real-time clock, the VME Computer could attach a header to the data transfer, identifying the period of the data. The transmission medium and protocol between the VME Computer and the Trigger Farm is envisaged to be fast Ethernet.

A second function is to arrange for control information to be sent to the front-end electronics by communicating with the Data Collectors. Using standard VME communication protocols on the VME backplane, data and address can be written to a Data Collector. The address is decoded, and a destination Super Concentrator is selected. The control data is sent from the Data Collector to the AUX Card, where it is sent to the appropriate Super Concentrator. This is a slow control function, so there are no requirements with respect to global timing. The data must have an address encoded in it, so that the destination Super Concentrator can in turn select a destination Data Concentrator. Similar addressing and decoding is then done by the Data Concentrator to select a destination front-end chip. The control link is bi-directional, so that control information can be read as well in response to a read command from the VME Computer to the Data Collector.

The VME Computer uses a modern operating system. Neither this nor the programming language is specified in this document.

5. System Architecture (Cont.)

5.7. Trigger Farm

The part of the system that performs event selection and reconstruction is the Trigger Farm. It receives time frames of data from the VME computer, and executes algorithms on them to form real events.

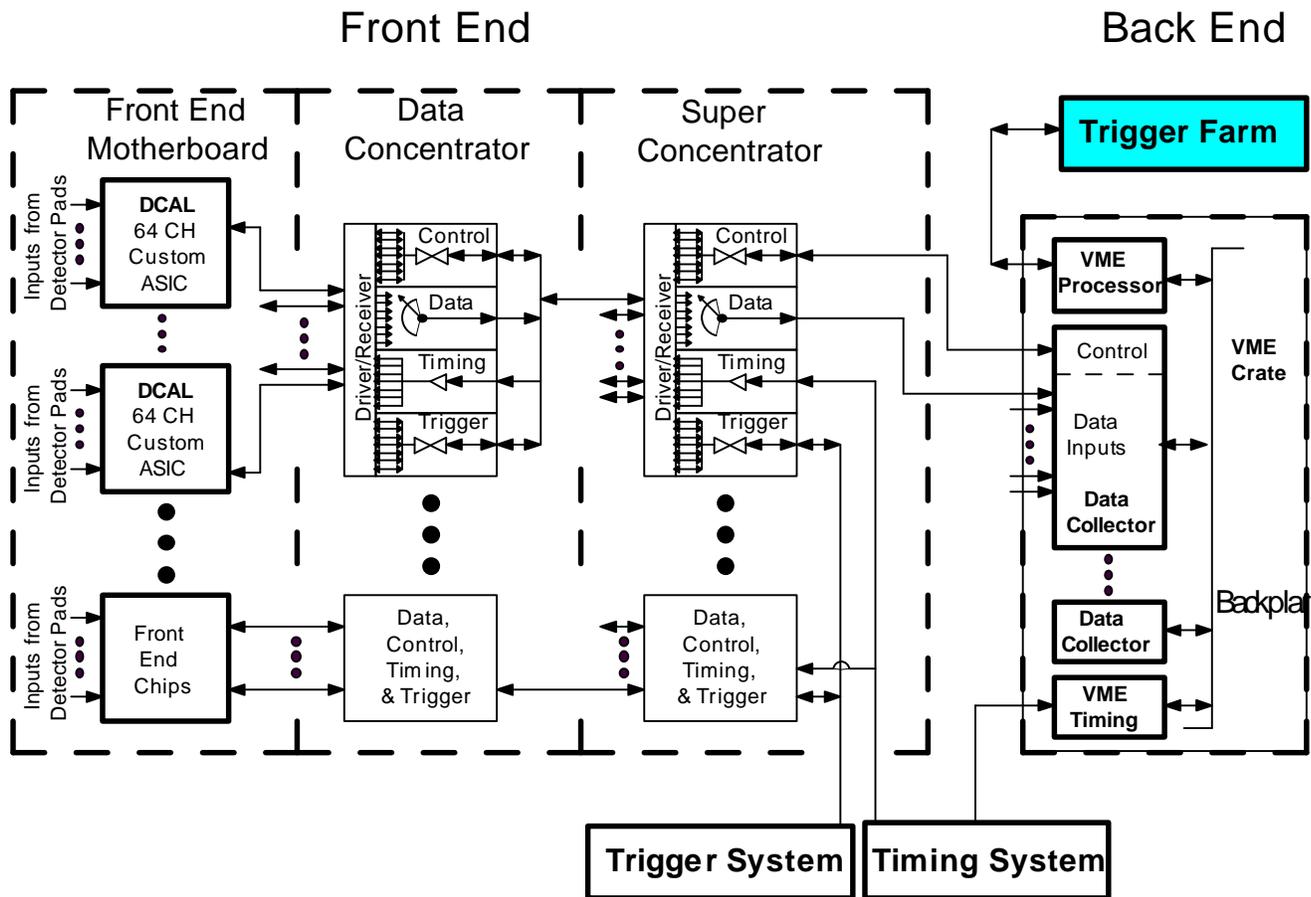


Figure 5.21. Sub-component Description: Trigger Farm

5. System Architecture (Cont.)

5.7. Trigger Farm (Cont.)

The data received by the Trigger Farm contains a mix of noise, background events, and real data. There is no zero-suppression of data in any other part of the system architecture. The Trigger Farm must separate out the real events from the junk. To do this, an algorithm similar to that used in the MINOS Experiment at Fermilab [21] might be employed. This works as follows: Each data word in a time frame consists of a hit pattern corresponding to channels in a front-end chip, and a timestamp. The algorithm first sorts the data by time order. Next the data is passed through a sliding time window, looking for a multiplicity of data words within it. Noise is easily identified, because they will be isolated in time and space. These are discarded. When a multiplicity of data words is found in a time window, the algorithm saves the data, tagging it as a "snarl," or possible event. This data is passed to a second process or processor for further evaluation. This first level of algorithm continues through the end of the time frame.

Once snarls are formed, a second algorithm is executed that calculates a trajectory in space and time. Knowing the geometry of the detector (the mapping of the physical locations of channels into channel address space), the trajectory of each snarl is compared to the known direction of the beam. For a test beam, the trajectory would point back to the direction of the beam. For colliding beams, the trajectory would tend to be orthogonal. If there is a magnetic field, this would have to be accounted for in the calculations and evaluation. If there are detector subsystems closer to the interaction point in the detector, timing and spatial information from these might be used as well. Using this analysis or others either of similar or higher sophistication, background event may be separated from beam events. The resulting data set is then either passed to a higher level in the trigger system, or written to mass storage for off-line evaluation.

The details of the hardware configuration, as well as the algorithms, are not specified in this document.

5. System Architecture (Cont.)

5.8. Timing System

The Timing System has an important role in the electronics. Event selection and reconstruction depend on having the timestamp counters in all of the front-end chips synchronized. The Timing System also controls the formation of time blocks and time frames in the back end electronics, which also has certain advantages if this is synchronized with the front-end electronics. The relationship in the system architecture is shown in Fig. 5.22.

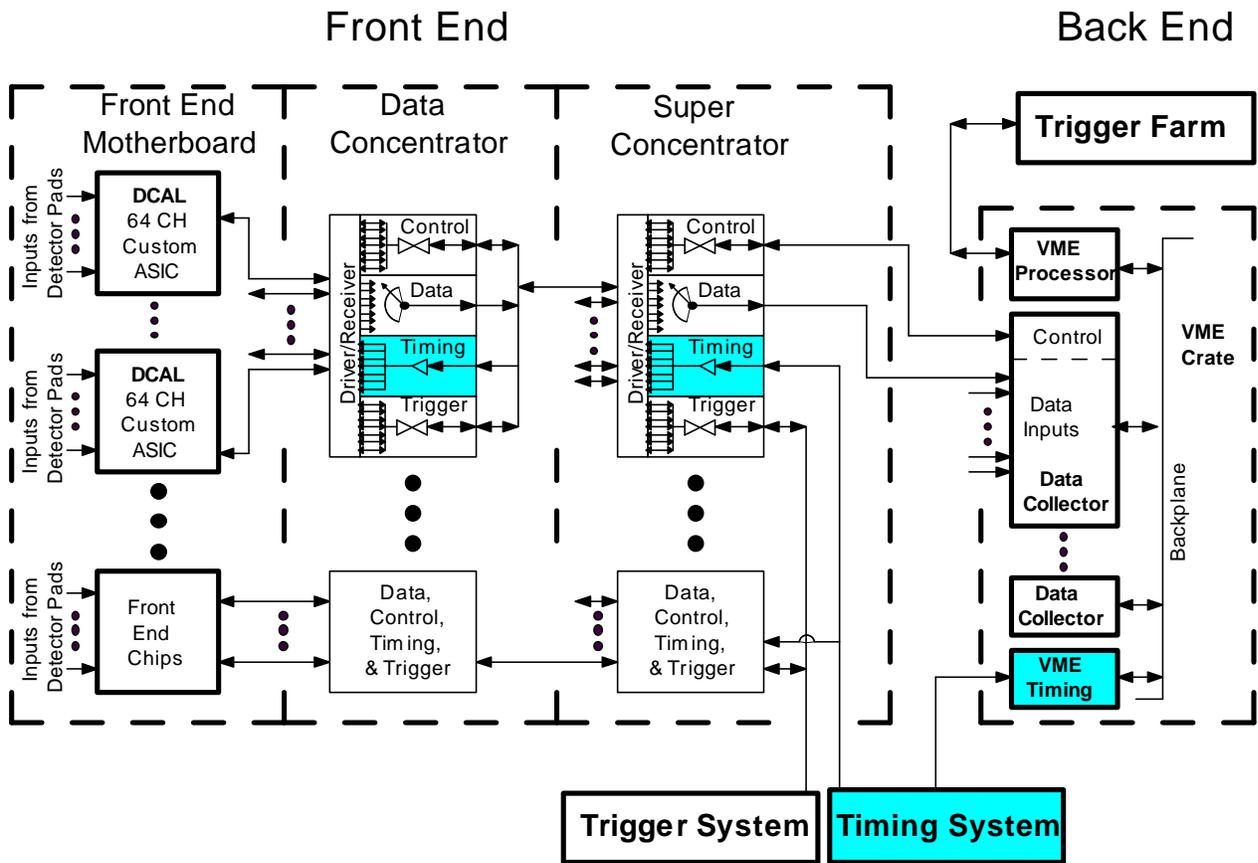


Figure 5.22. Sub-component Description: Timing System

5. System Architecture (Cont.)

5.8. Timing System (Cont.)

A block diagram of the Timing System by itself is shown in Fig. 5.23. It has several components, which are intertwined into both the front-end electronics and the back-end electronics. The function of each component, and the nature of the timing signals, will be described.

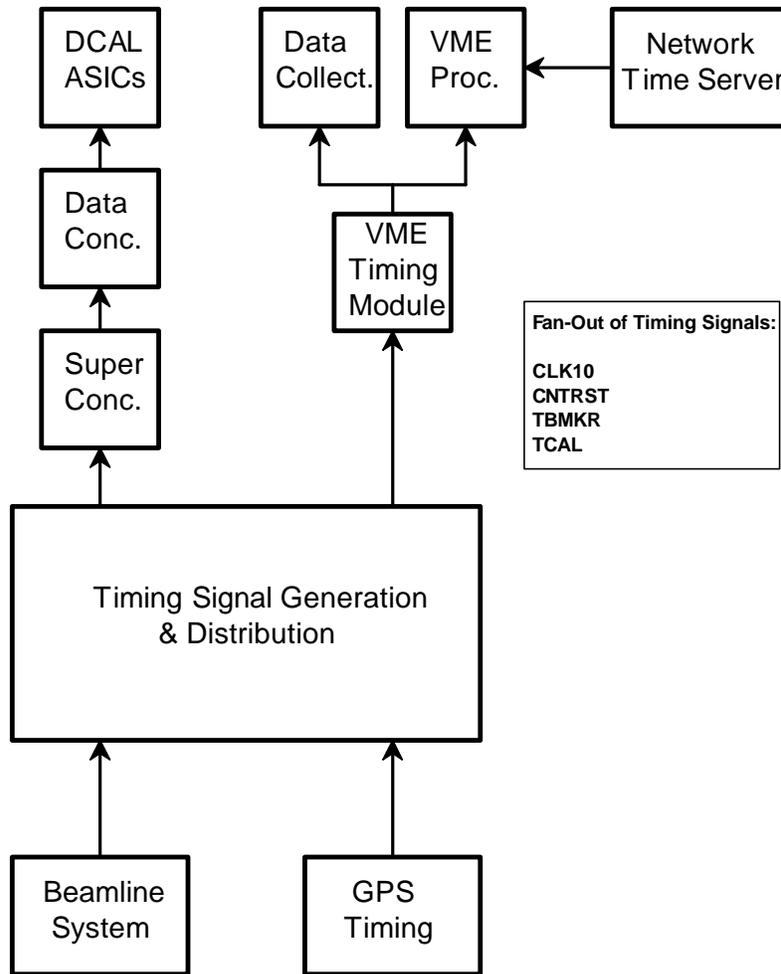


Figure 5.23. Block Diagram of the Timing System

5. System Architecture (Cont.)

5.8. Timing System (Cont.)

The timing system must provide four high-precision signals to different parts of the electronics. The first is the 10 MHz Clock, denoted as CLK10. This is the basic timing signal used by all components in the system. The front-end chips use this to advance the value of the timestamp counters, which are used as a tag on hits measured in the chips. The front-end chips also use CLK10 for other functions. The states of all comparators are refreshed on the rising edge of the clock. The serial data transmission operates at the same frequency for the front-end chip and the Data Concentrator. The output of the Super Concentrator also uses the 10 MHz clock. This clock does not need to be synchronized with that used by the Data Concentrators, although the CLK10 signal is readily available from the timing system. Both event data and control data passed to and from the front-end chips have a strobe associated with them, and this is generated from CLK10. Note that there is no requirement on the system that the high-speed clock be synchronized across the system. Asynchronous data transfers are permitted, and each system component must buffer and re-synchronize the data as appropriate.

In order to provide a high accuracy of timing, it is necessary for CLK10 to have a high-level of precision, so that the counters in all front-end chips can be synchronized. The implementation of a global source for this clock is an important aspect, but the distribution of timing signals must be done with care to avoid spoiling the precision. This includes uniform cable lengths, short cable lengths, the use of low-dispersion cable, the minimization of buffers, and the use of transceivers with low intrinsic delay.

The front-end chips receive CLK10 through the Super Concentrator and the Data Concentrator, as shown in Fig. 5.23. The logic in those components may need a clock, to perform either synchronous or asynchronous data processing. They may both use CLK10 for these functions, although it is important that the distribution of signals to the front-end chips be done carefully to not lose synchronization across the system. Note that the data processing functions on the Data Concentrators and Super Concentrators do not have to be synchronized across the system.

The back-end electronics may use the 10 MHz clock as well. While the basic data transfer and processing do not need to be synchronized with the front-end electronics, it is advantageous for the back-end to be synchronized in the formation of time blocks and time frames. This will be discussed shortly.

5. System Architecture (Cont.)

5.8. Timing System (Cont.)

The second precision timing signal is Counter Reset, denoted as CNTRST. The timestamp counters in the front-end chips have a finite number of bits, and therefore must either be reset occasionally or allowed to roll over. Having a global counter reset helps to ensure that all counters remain in sync for long periods of time, and helps to reduce the loss of data due to synchronization errors. The timestamp counters have 24 bits, and at a clock frequency of 10 MHz, this corresponds to just over one second before roll-over occurs. Having a counter reset once per second has advantages for the back end of the system as well. Like the 10 MHz clock, counter reset must be synchronized over all front-end chips. It is synchronized by the clock, using the rising edge of the clock in coincidence with the assertion of counter reset to perform the reset operation, as shown in Fig. 5.24. As shown in the figure, a convenient source for a signal with a one-second period is the one pulse-per-second (pps) signal that is available from the Global Positioning System (GPS.)

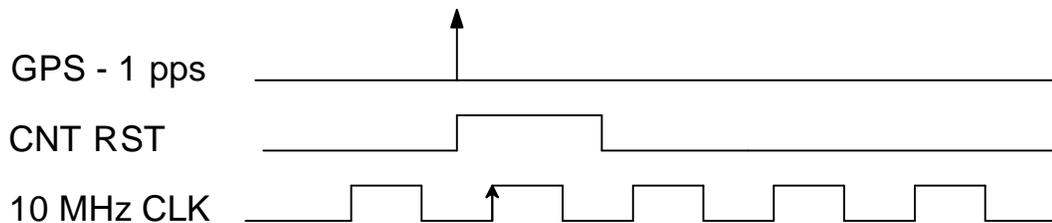
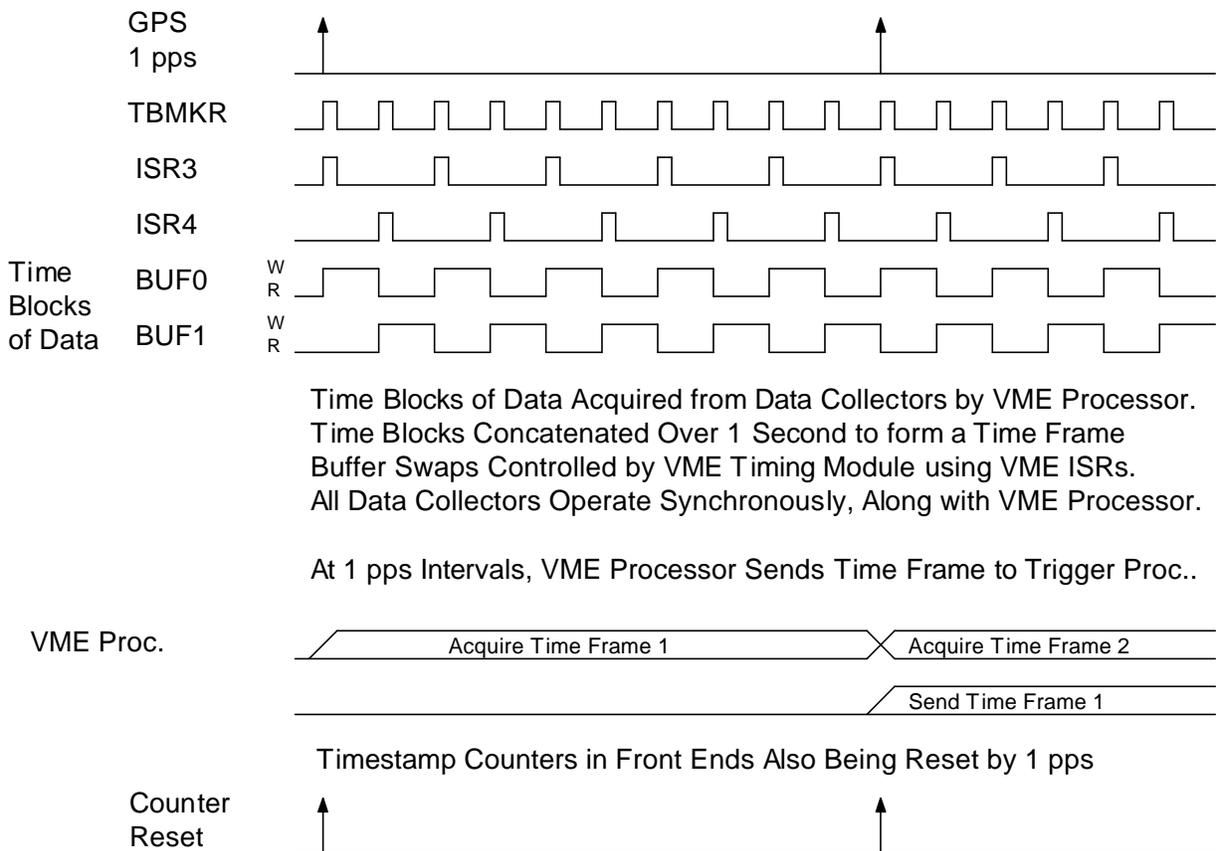


Figure 5.24. Timing Relationship Between CLK10 and CNTRST

5. System Architecture (Cont.)

5.8. Timing System (Cont.)

The third precision timing signal is the Time Block Marker, denoted as TBMKR. This signal is used by the back end electronics to control the change in data buffer states. As described earlier, each Data Collector has two readout buffers. One buffer is made available to the VME Processor for reading while new data is being acquired. At a period determined by the Time Block Marker, the states of the buffers change, occurring at the same time on each Data Collector. The VME Processor also knows when this state change takes place, and begins a new read cycle when it occurs. This is controlled by the Timing System, which sends out TBMKR at a predetermined frequency. As shown in Fig. 5.23, it is received by the VME Timing Module, a card that resides in the VME crate. This is a relatively simple module, which generates one of two Interrupt Service Requests (ISR) in ping-pong fashion on the VME backplane. Each ISR corresponds to a particular buffer state. The modules in the VME Crate respond according. The timing is illustrated in Fig. 5.25.



Time Blocks of Data Acquired from Data Collectors by VME Processor.
 Time Blocks Concatenated Over 1 Second to form a Time Frame
 Buffer Swaps Controlled by VME Timing Module using VME ISRs.
 All Data Collectors Operate Synchronously, Along with VME Processor.

At 1 pps Intervals, VME Processor Sends Time Frame to Trigger Proc..

Figure 5.25. Timing Relationship Between TBMKR and Buffer Swaps

5. System Architecture (Cont.)

5.8. Timing System (Cont.)

The period of TBMKR is programmable in the central processor of the Timing System. In principle, there are no constraints on it, as long as the buffers on the Data Collectors are not allowed to fill completely. However, the implementation of an additional level of synchronization can provide an additional level of integrity to the system. The timestamp counters are reset once per second. If the VME Processor forms time frames that are one second long as well, and if the formation of time frames is synchronous with counter reset, then the data in a time frame will have unique timestamps. The timestamps will also increase monotonically, at least approximately, in the data record. This will aid in the data sorting done by the Trigger Farm. For this to work correctly, it will be necessary for there to be a whole number of Time Block Markers in a one-second period. Also, it is desirable for the number of Time Block Numbers to be an even number, so that a new time frame always begins with the same data buffer. This is not an absolute requirement, but could be used as a check to see that TBMKR signals have not been missed.

The fourth precision timing signal is used for generating a calibration signal. Denoted as TCAL, it is used by the front-end chip to inject charge into selected channels.

In order to keep the signal distribution as simple as possible, as well as to provide a degree of flexibility in case of contingencies, it may be desirable to distribute all four timing signals as a group to all locations. This would allow the selection of a standard connector and cable for the distribution of all timing signals. In order to minimize noise, the timing signals should be distributed as differential signals, such as LVDS.

The timing system is envisaged to have a central processor or unit, as shown in Fig. 5.23. In the basic form of operation, it can stand alone, needing no external inputs to determine or control the timing signal formation. It may be desirable to use the Global Positioning System (GPS) as a means of generating an accurate pulse per second (pps). It may also be desirable to use inputs from the beamline, to provide an indicator of when beam interactions might occur.

5. System Architecture (Cont.)

5.9. Trigger System

The front-end chips can acquire data in several different ways. One way is to self-trigger, where any hit registered by a front-end channel automatically initiates data transmission to the Data Concentrator. This is useful for studying noise, as well as for measurements of cosmic rays. However, two other trigger modes are built into the system. One is the case where external detectors and hardware are used to produce a trigger. The front-end chip is designed to accommodate an external trigger, and the Data Concentrator acts as the interface. In this case, the Data Concentrator merely acts as a pass-through, sending the signal TRIGIN to the front-end chips from an external trigger source. The second case is the situation where there is no external trigger, but it is known in advance when events of interest might occur. This might be the case where beams of particles are created by an accelerator, and a time period is known where there is the possibility of an event occurring, as opposed to having actually detected an event. The front-end chips have a special mode that looks for the coincidence of a hit in a channel, and the presence of this gate signal, called TRIGGATE. When this coincidence occurs, data is acquired by the front-end chips and is automatically transferred to the Data Concentrator. Again, the Data Concentrator acts as the interface. Both TRIGIN and TRIGGATE are bussed signals. The trigger mode that the front-end chips operate in is controlled through the control path.

In certain applications, it may be desirable to have the front-end chips participate in the trigger decision. They are designed to have a trigger-out signal, called TRIGOUT. Any time a channel goes over threshold, this signal is asserted by the front-end chip. Note that this is an "OR" of all channels in a chip. This signal could be used in several ways. It might be used by the Data Concentrator to form a "local trigger," looking for coincidence among the 12 front-end chips serviced by the Data Concentrator. There might also be an external trigger system, which would receive the TRIGOUT signals from all front-end chips, and use this in making an external trigger decision. This functionality is not specifically defined in this document. However, by using programmable logic in the design, it is advantageous to incorporate flexibility should this be desired functionality. Note that the TRIGIN signals are point-to-point between each front-end chip and the Data Concentrator.

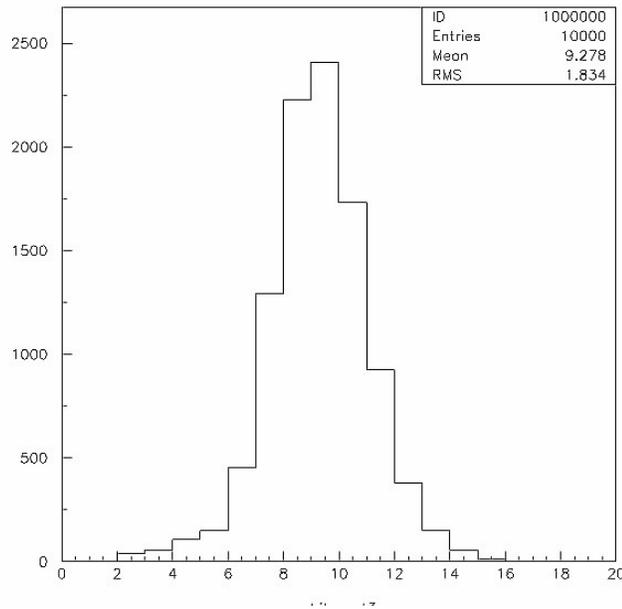
The nature of the Trigger System, and the details of performance and operation, are not specified at this time.

6. Trigger and Data Rates

In this section, the data rates and required bandwidth at each point in the system are defined and specified.

6.1. Nature of Events

As specified in Section 3, the expected event rate in the test beam shall be a maximum of 100 Hz. When beam events occur, they tend to deposit energy in local regions of each detector plane, and continue through the detector in a transverse direction with the beam. In general, multiple channels in a given chip will be hit when beam events occur, and often multiple chips will be hit at the same time on a given plane as well. Events may also span over more than one clock cycle in time over the whole detector. For a given event, the total occupancy over the entire detector is very low, but because the system architecture is configured to read out planes, the planes that have the highest occupancy set the performance needed. Simulations of the response of the prototype detector to 50 GeV pions have shown that the average number of chips hit per plane is approximately 9, but may be as high as 16 [22.] See Fig. 6.1.



**Figure 6.1. Maximum Number of Chips Hit on a Plane per Event
50 GeV Pions**

6. Trigger and Data Rates (Cont.)

6.1. Nature of Events (Cont.)

Because much of the readout system is based on plane readout, the number of ASICs hit per plane and the average and maximum rates set the performance of many parts of the system. These include the DCAL ASIC, the Data Concentrators and Super Concentrators. The performance of the Data Collector and the readout through the VME system are dependent on how energy is deposited in multiple planes as particles pass through the detector. This is complex, and difficult to characterize in terms of setting system performance. However, the problem can be divided into two parts, each of which can be addressed at a basic level. The first concerns understanding the event rate into inputs of the Data Collectors. Since the Super Concentrators write data into the Data Collector inputs, and since each Super Concentrator services half of a plane, then the nature of energy deposition into the planes sets this performance. The Data Collectors must have adequate buffering to handle the event rate. The second concerns reading the data from the Data Collectors by the VME Processor. While the amount of data collected by individual Data Collectors may be difficult to characterize, it is relatively straightforward to calculate the number of ASICs hit per event. The assumption then is that the VME processor must acquire all of this data. A plot of the total number of ASICs hit in the detector per event is shown in Fig. 6.2. The average number is 135, but can exceed 200. This information will be used in subsequent sections to set the system performance.

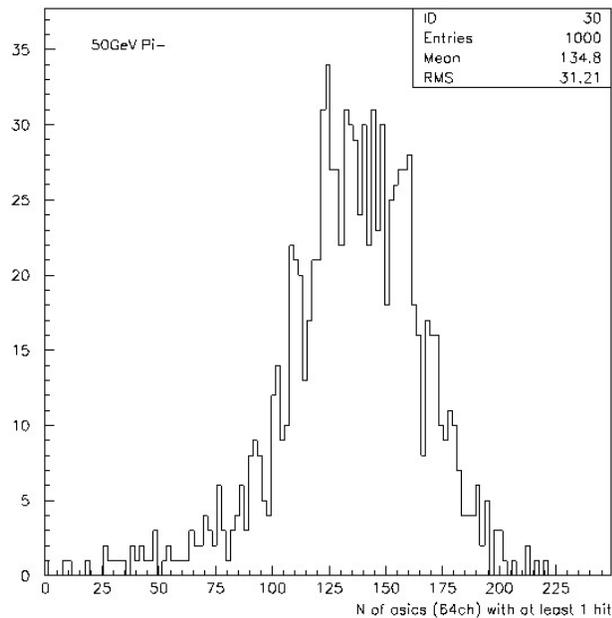


Figure 6.2. Maximum Number of Chips Hit in the Detector per Event for 50 GeV Pions

6. Trigger and Data Rates (Cont.)

6.2. Front-End Chip

The most stringent aspect of operation is when one or more channels in a chip are hit at 100 Hz. This is much greater than the intrinsic noise of the detector, which is approximately 1 KHz per square meter, or 6.4 Hz per chip. When a front-end chip receives a trigger, whether self-triggered or from an external trigger, it causes the data is stored in the output FIFO inside the chip. Each data word contains 64 bits that represent the states of the 64 comparators, plus 24 bits of timestamp, for a total of 88 bits, or 11 bytes. If the maximum hit rate is 100 Hz, then the maximum expected data rate stored in a chip is 1100 Bytes/sec.

The bits are written out of the chip using serial transmission. Eight bits are processed at a time, encoded into an 11-bit serial transmission. The extra three bits are used for framing, and they provide a means to synchronize the receiver. Thus, 121 bits are sent out of the chip per triggered hit. The output bit rate is governed by the 10 MHz clock, achieving 10 Mbit/sec. When the framing bits are removed, this corresponds to 910 KByte/sec of data. It thus takes 12.1 microseconds to send out an entire data block, or 121 times the fundamental clock period. While there is buffering inside the front-end chips to handle instantaneous bursts, the maximum data rate out of each chip is limited to 83 KHz (events per second.)

The duty factor, expressed as the ratio of the maximum expected bytes per second produced by the chip from events, to the maximum output rate in bytes per second, is 0.12%.

6. Trigger and Data Rates (Cont.)

6.3. Data Concentrator

Each Data Concentrator services twelve front-end chips, in a two-by-six array. The average maximum number of chips hit per event on a plane is nine. The spatial deposition of energy in a plane is difficult to characterize in general. The worst case for the electronics is where clustering occurs. Consider the case where hits are clustered in a 3-by-3 array of chips. The maximum overlap onto the 2-by-6 array of chips serviced by a Data Concentrator would then hit approximately half of the chips. If each chip has an average hit rate of 100 Hz, then each Data Concentrator receives hits at six times this rate, or 600 Hz. If each front-end chip has an expected (stored) data rate of 1100 Bytes/Sec, then the equivalent stored data rate into the Data Concentrators is 6600 Bytes/Sec. Again, the Data Concentrators have buffering on the inputs to accommodate data bursts. As part of the processing, the Data Concentrator takes out the framing bits to recover the 8 bits of data. Note that this figure is not the data transmission rate from the front-end chip. Rather, it is the rate at which bytes of data are accumulated by the Data Concentrator, and assumes that the transmission rate is high compared to the data rate, which is the case.

Once data is received and processed by the Data Concentrator, it must then add four bits of address to identify which of the twelve inputs the data came from. An additional four bits are added as padding, to be used for additional address identifiers later and to preserve the number of bytes as a whole number. The new data block for each front-end chip thus contains 96 bits, or 12 bytes. The data is retransmitted in bytes, adding two framing bits per byte to form 10 bits. The output transmission again uses the 10 MHz clock, sending 10 Mbits/sec, or 1 MByte/Sec of actual data. Thus, each 96-bit data word (120 bits total with framing) requires 12 microseconds to send. If six front-end chips are hit per event, then the Data Concentrator must process 72 bytes per event, and the average maximum time to send the data from an event would be 72 microseconds. This corresponds to a maximum output rate of 14 KHz (events per second.)

Because the Data Concentrator adds an additional byte to each event, the aggregate input rate of 6600 Bytes/Sec becomes 7200 Bytes/Sec after internal processing. The duty factor is the ratio of 7200 KBytes/Sec, the "processed" input event rate, to 1 MByte/Sec, the output rate, which is 0.72%.

If the number of chips hit in a plane is 16, the absolute maximum, then consider the case where the hits occur in a 4-by-4 array of chips. A Data Concentrator servicing a two-by-six array of front-end chips might then have eight chips hit. By the same arguments used above, the aggregate rate into the input of a Data Concentrator would increase to 8800 Bytes/Sec, which after internal processing would become 9600 Bytes/Sec. The time to transmit an event would increase to 96 microseconds. The duty factor would be 0.96%.

6. Trigger and Data Rates (Cont.)

6.4. Super Concentrator

Each Super Concentrator services six Data Concentrators, which corresponds to half of a plane. In Section 6.1, the average maximum number of chips hit on a plane was given to be nine. It is conceivable that an entire event could be contained in a region of the plane serviced by one Super Concentrator. This can happen for example if six chips are serviced by one Data Concentrator, and the other three chips are serviced by an adjacent Data Concentrator on the same half of the plane. Then, all of the data produced on the plane passes through one Super Concentrator. The Data Collectors send 12 bytes of data for each chip hit. Thus, when nine chips are hit, the average maximum amount of data per event sent to the Super Concentrators is 108 bytes. At 100 Hz event rate, this corresponds to 10.8 KByte/Sec. In the worst case where 16 chips are hit per plane, each event would produce 192 bytes, and the data rate into the Super Concentrator would be 19.2 KByte/Sec.

On the output side, the Super Concentrator must be capable of processing the full event at the expected average maximum input rate. Like the Data Concentrators, the Super Concentrators have buffering on the inputs to accommodate data bursts. Again, it will be assumed that the Super Concentrators decode the 10-bit transmission back into 8 bits of data, and take out the framing bits on the input. The upper four bits of the address byte that were added by the Data Concentrator are used by the Super Concentrator to identify which of the six inputs the data came from. The new data word still contains 96 bits, or 12 bytes. Thus, the output event size would contain 108 bytes per event for 9 chips hit at a time, and 192 bytes per event in the worst case.

As before, the data is retransmitted in bytes, adding two framing bits per byte to form 10 bits. The output data rate is assumed to be 10 Mbit/Sec, or 1 MByte/Sec. For the case where the data from 9 front-end chips are processed per event, then it would take 108 microseconds to process the data from one event. For 16 chips, the time would be 192 microseconds.

For 9 chips, the duty factor is the ratio of 10.8 KByte/Sec, the aggregate input rate, to 1 MByte/Sec, the output rate, which is 1.08%. For 16 chips, this would be 1.92%.

6. Trigger and Data Rates (Cont.)

6.5. Data Collector

Each Data Collector services twelve Super Concentrators. Each Super Concentrator services half of a detector plane. In general, energy deposition in a plane of the detector is localized to a small area, and most of the data from a plane will come from one of the two Super Concentrators for a given event. In order to consider the performance needed by the Data Collector, consider the worst case where all 12 inputs receive data at the maximum rate. For the case where 9 chips are hit per plane, each Super Concentrator sends data at 10.8 KByte/Sec, so the aggregate rate would be 130 KByte/Sec. The Data Collectors have buffering on the inputs to accommodate data bursts.

Each data word received at the input of the Data Collector is decoded from the serial encoding, producing 12 bytes, or 96 bits. When data is received and processed by the Data Collector, it must add an address before it is put into a readout buffer. While only 4 bits are needed, the Data Collector adds an entire byte with padding as needed, in order to preserve a standard byte format. Thus, each data word contains 13 bytes of data after processing by the Data Collector. Choosing a conservative approach for the VME transfers, each data word would be read using two 64-bit block transfers, with 3 additional bytes padded with zeros. Thus, it would take 2 VME read cycles to read each 16-byte data word. This boosts the "processed" input data rate to 173 KByte/Sec. This corresponds to the "reasonable" worst-case rate at which data is written into the readout buffers on the Data Collector.

To check these assumptions from a different point of view, consider the distribution shown in Fig. 6.2. The average number of chips hit per event is shown to be 135. If the event rate is 100 Hz, and each chip results in 16 bytes of data, then the aggregate data rate from the detector would be 216 KByte/Sec. While it is unlikely that a single Data Collector would have the majority of the data rate, the assumptions used above are within the overall expected data rates. Note that the absolute worst case where 200 chips are hit per event, if it could be sustained at 100 Hz, would produce 320 KByte/Sec.

6. Trigger and Data Rates (Cont.)

6.6. Readout Computer

As described in Section 6.1, the average maximum number of ASICs hit per event is about 135. If each ASIC hit produces a 16-byte data word, then the number of bytes per event is 2160. At the 100 Hz event rate, this corresponds to 216 KBytes/Sec. A VME crate can transfer data across the backplane at a rate of 10 MByte/Sec, although overheads can slow this down. Nonetheless, if all of the data from the detector comes through one VME crate, there is a margin of approximately a factor of 50 between the expected rate and the maximum processing rate.

There is a set of general relationships between readout buffer size, buffer swap period, and readout speed on the VME backplane. On the input side, the buffers must not be allowed to fill before changing state. On the output side, the readout buffers from all Data Collectors must be read before the buffers change state. The general relationships are:

$$[\text{input rate per Data Coll. (Byte/Sec)}] * [\text{buffer swap period}] < [\text{buffer size}]$$

and

$$[\# \text{ Data Coll. per Crate}] * [\text{buffer size}] / [\text{buffer swap period}] < [\text{VME readout speed (Byte/Sec)}]$$

For example, let the input data rate for a Data Collector be 100 KByte/Sec. Assume that there are 20 Buffer Swaps per second, or one every 50 mSec. The buffer size would need to be larger than 5 KBytes for the entire Data Collector, or 416 bytes per input channel (assuming 12 input channels.) If there are 20 Data Collectors per VME Crate, then the readout speed would need to be greater than 2 MByte/sec. This is comfortable for a typical 64-bit block transfer on a VME backplane operating with a 25 MHz clock.

7. Summary of Component Counts

The following table summarizes the number of each component needed for the system as described in this document.

Component	#/Chamber	#/Plane	# Channels per Unit	TOTAL # Units
Planes	0.333	1	9216	40
Chambers	1	3	3072	120
DCAL ASIC	24	144	64	5760
FE Bds	2	6	1536	240
Data Conc.	4	12	768	480
Super Conc.	0.667	2	4608	80
Data Collector	-	0.166	55,296	7
VME Crates	-	-	387,072	1

Table 7.1. Summary of Component Counts

8. Bibliography

- [1] "Results of tests of prototype Resistive Plate Chambers," I. Crotty, J. Lamas Valverde, G. Laurenti, M.C.S. Williams, A. Zichichi, 1994. In Pavia 1993, Proceedings, The resistive plate chambers in particle physics and astrophysics* 199-206.
- [2] "Investigation of Resistive Parallel Plate Chambers," I. Crotty, J. Lamas Valverde, G. Laurenti, M.C.S. Williams and A. Zichichi, published in Nuclear Instruments and Methods in Physics Research A 329 (1993) 133.
- [3] "The Non Spark Mode and High Rate Operation of Resistive Parallel Plate Chambers," I. Crotty, J. Lamas Valverde, G. Laurenti, M.C.S. Williams and A. Zichichi, published in Nucl. Instr. and Meth. A 337 (1994) 370.
- [4] "Further Studies of Avalanche Mode Operation of Resistive Parallel Plate Chambers," I. Crotty, J. Lamas Valverde, G. Laurenti, M.C.S. Williams and A. Zichichi, Nucl. Instr. and Meth. A 346 (1994) 107.
- [5] "The Wide Gap Resistive Plate Chamber," I. Crotty, E. Cerron Zeballos, J. Lamas Valverde, D. Hatzifotiadou, M.C.S. Williams and A. Zichichi, Nucl. Instr. and Meth. A 360 (1995) 512.
- [6] "High Rate Resistive Plate Chambers," E. Cerron Zeballos, I. Crotty, D. Hatzifotiadou, J. Lamas Valverde, S. Neupane, S. Singh, M.C.S. Williams and A. Zichichi, CERN PPE/95-93, CERN/LAA-MC 95-16, presented at Vienna Wire Chamber Conference, Feb. 13, 1995. Nucl. Instr. and Meth. A 367 (1995) p. 388-393.
- [7] "New results on High Rate Resistive Plate Chambers," E. Cerron Zeballos, I. Crotty, D. Hatzifotiadou, J. Lamas Valverde, S. Neupane, S. Singh, M.C.S. Williams and A. Zichichi, CERN/LAA-MC 95-22, presented at the "Four Seas Conference", Trieste, Italy 25 June - 1 July 1995.
- [9] "The development of the MultiGap Resistive Plate Chamber," E. Cerron Zeballos, I. Crotty, D. Hatzifotiadou, J. Lamas Valverde, M.C.S. Williams and A. Zichichi, CERN/LAA-MC 97-05, presented at the Fourth International Workshop on Resistive Plate Chambers and Related Detectors, Naples, Italy, October 1997.
- [10] "The Development of the Multigap Resistive Plate Chamber," M.C.S. Williams et al., 1998, Nucl. Phys. B -- Proc. Suppl., Volume 61, Issue 3, 1998, pp. 250-257.
- [11] "The Gas electron multiplier (GEM)," By R. Bouclier, et al., CERN-PPE-96-177, Nov 1996.

8. Bibliography (Cont.)

- [12] "The Gas electron multiplier (GEM)," By R. Bouclier, et al., IEEE Trans.Nucl.Sci.44:646-650,1997.
- [13] "GEM: A new concept for electron amplification in gas detectors," By F. Sauli. Nucl. Instrum. Methods A386 (1997) 531-53.
- [14] "Development of the gas electron multiplier (GEM)," By J. Benlloch, et al., IEEE Trans. Nucl. Sci. 45 (1998) 234-24.
- [15] "Studies of an MSGC equipped with a GEM grid as a tracking device," By W. Beaumont, et al., Nucl. Instrum. Methods A419 (1998) 394-39.
- [16] "Rate and gain limitations of MSGCs and MGCs combined with GEM and other preamplification structures," By P. Fonte, V. Peskov, B.D. Ramsey, Nucl. Instrum. Methods A419 (1998) 405-40.
- [17] "A GEM of a detector," CERN Cour. 38 (1998) No. 9 19-2.
- [18] "Further developments of the gas electron multiplier (GEM)," J. Benlloch, et al., Nucl. Instrum. Methods A419 (1998) 410-41.
- [19] "Detection of single electrons emitted by internal photocathodes with the gas electron multiplier (GEM)," A. Sharma, CERN-OPEN-99-372 (Dec 1999).
- [20] "Specification of the DCAL Chip." Version 2.0, Argonne Internal Document, Feb. 23, 2004.
- [21] "MINOS Technical Design Report," NuMI Note #337, Fermilab Internal Document, October, 1998.
- [22] Detector simulations by Lei Xia, Argonne National Laboratory, June, 2005.