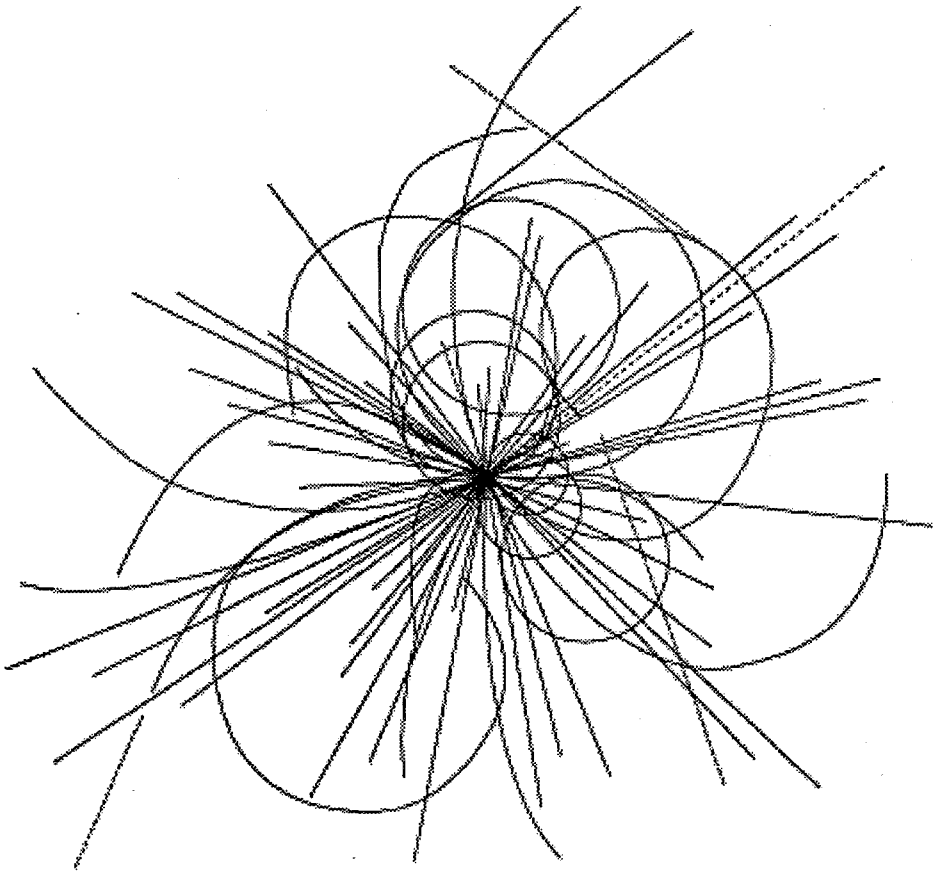
SSCL-Preprint-560

A 40 GByte/s Read-Out System for GEM



M. Bowden
J. Carrel
J. Dorenbosch
V. Kapoor

SSCL-Preprint-560
April 1994
Distribution Category: 414

Superconducting Super Collider Laboratory

SSCL-Preprint-560

A 40 GByte/s Read-Out System for GEM*

M. Bowden, J. Carrel, J. Dorenbosch and V. Kapoor

Superconducting Super Collider Laboratory†
2550 Beckleymeade Ave.
Dallas, Tx 75237

April 1994

*To be submitted to *Nuclear Instruments and Methods in Physics Research.*

†Operated by the Universities Research Association, Inc., for the U.S. Department of Energy under Contract No. DE-AC35-89ER40486.

A 40 GByte/s read-out system for GEM

M. Bowden, J. Carrel, J. Dorenbosch and V. S. Kapoor
Superconducting Super Collider Laboratory, Dallas, TX, USA*

The preliminary design of the read-out system for the GEM (Gammas, Electrons, Muons) detector at the Superconducting Super Collider is presented. The system reads all digitized data from the detector data sources at a Level 1 trigger rate of up to 100 kHz. A total read-out bandwidth of 40 GBytes/s is available. Data are stored in buffers that are accessible for further event filtering by an on-line processor farm. Data are transported to the farm only as they are needed by the higher-level trigger algorithms, leading to a reduced bandwidth requirement in the Data Acquisition System.

1. The GEM detector

GEM was to become one of the two large detectors at the Superconducting Super Collider (SSC). The GEM design emphasizes clean identification and high-resolution measurement of the high-p_T physics accessible at the SSC: Higgs physics, heavy quark production, Z', W', Supersymmetry, etc. [1].

A detailed description of the proposed detector can be found in Reference [2]. GEM specializes in measurement of photons, electrons and muons, hence its name: "Gammas, Electrons, Muons." It uses a Silicon Strip system and Interpolation Strip Chambers, a very precise electromagnetic calorimeter, a hadronic calorimeter, and a precise 4π muon spectrometer. All detector subsystems are inside a giant solenoidal magnet allowing measurement of the momenta of high-energy muons. One of the primary physics goals of GEM is to be sensitive to

*Operated by the Universities Research Association, Inc., for the U.S. Department of Energy under Contract No.~DE-AC35-89ER40486.

the distinctive decay of a light Higgs to two photons. This goal is a main driver for many of the requirements on the on-line event selection and the readout.

Collisions occur in the center of the detector at a rate of 60 MHz. In each collision, hundreds of particles are produced. To measure the characteristics of these secondary particles with sufficient precision, the detector must have excellent spatial, energy and time resolution. To deal with the large number of particles extremely fine segmentation is required. The detector has millions of individual measurement channels, that each are sampled at 60 MHz. Most electronics for measurement and digitization are mounted on front-end electronics modules inside the detector. Electronics must be extremely reliable and extensive monitoring must be built into the system. In case of errors the system must degrade in a predictable and graceful way. The number of channels in the different subsystems is shown in table 1. The total volume of information that is generated at each crossing is 2.3 MBytes. The total information flow is larger than 10^{14} Bytes per second.

Table 1
Detector subsystem data volume parameters

Subsystem	Channels k	bits/ channel	Data Size kBytes
Silicon	2500	1	313
IPC	400	10	500
Calorimeter	125	18	281
Muon wires	300	2	75
Muon strips	900	10	1125
Sum	4225		2294

The amount of information needed to describe particles differs vastly between subsystems. Channel occupancies vary significantly too. The silicon strip subsystem, for example, produces a single bit per strip when it is hit in a collision. The IPCs, in contrast, require a 10-bit pulse height measurement on each of three neighboring strips to determine a hit position. Moreover, since the time evolution of the signals on the strips is slow, pulse height values are sampled for at least

four consecutive clock periods. Particle hit positions can then be calculated by interpolating the integrated pulse heights.

The occupancy of most channels is low. Significant data reduction can be obtained by noise suppression. Obviously this increases the amount of information per hit channel, due to the addition of addresses. Table 2 shows the occupancy in each subsystem and the number of 16-ns samplings that must be measured for each particle hit. Noise suppression reduces the total amount of data needed to describe a single collision to a modest 300 kBytes.

Table 2
Data volume after noise suppression

Subsystem	Occupancy %	Samplings per hit	Data Size kBytes
Silicon	0.5	1	25
IPC	1.5	4	72
Calorimeter	4.0	9	135
Muon wires	1.0	2	6
Muon strips	0.5	4	41
Sum			279

2. Trigger philosophy

Most of the collisions are not particularly interesting. To reduce storage requirements, only a very small fraction ($\approx 10^{-5}$) of the collisions are actually recorded on mass storage. On-line real time selection picks out the collisions of potential interest. This selection, called "triggering," is typically done in multiple stages, called "Levels." Consecutive levels use information subsets of increasing size. The goal of each level is to reject uninteresting collisions. Only collisions of potential interest are accepted and passed to the next, higher stage. This way higher levels have more time available to perform more sophisticated pattern recognition.

Trigger and readout schemes of different detectors can be distinguished by the number of trigger levels, the way the triggers are implemented in hard and software, the data paths that are used to bring data to the trigger hardware, and the subsystems that participate at each trigger

level. For example, the Solenoidal Detector Collaboration (SDC) detector, GEM's competitor at the SSC, has three trigger levels that each have a separate data path. The lower two trigger levels in SDC are done in special purpose hardware with limited programmability [3].

In any detector the lowest selection stage must inspect data from all collisions. Since GEM specializes in gamma's, electrons and muons, the Level 1 trigger only needs information from the calorimeter and muon subsystems. Condensed information from these systems is transmitted from the front-end electronics to the Level 1 processor. GEM uses hundreds of optical fibers, each running at 1 GHz. The Level 1 trigger processor is implemented as a fully-pipelined digital processor. Data transport and trigger execution take a few microseconds. The maximum allowed Level 1 trigger accept rate is specified at 100 kHz.

During the operation of the Level 1 trigger hundreds of new collisions will occur. Information for them is continuously sent to the Level 1 processor pipeline. A copy of the complete information is simultaneously stored in the front-end electronics in analog or digital form. This requires an on-detector memory of about 0.5 GBytes operating at 16-ns clock speed. Information for events that are rejected by Level 1 is flushed. Data related to accepted collisions is captured, digitized, noise suppressed, and stored in small local derandomizing buffers on the front-end modules.

3. Readout architecture

The main tasks of the readout are to make data for events accepted by Level 1 available to the higher level triggers, and to move data for events accepted by the higher-level triggers to mass storage. The basic system architecture is shown in fig. 1. A detailed design report can be found in reference [4].

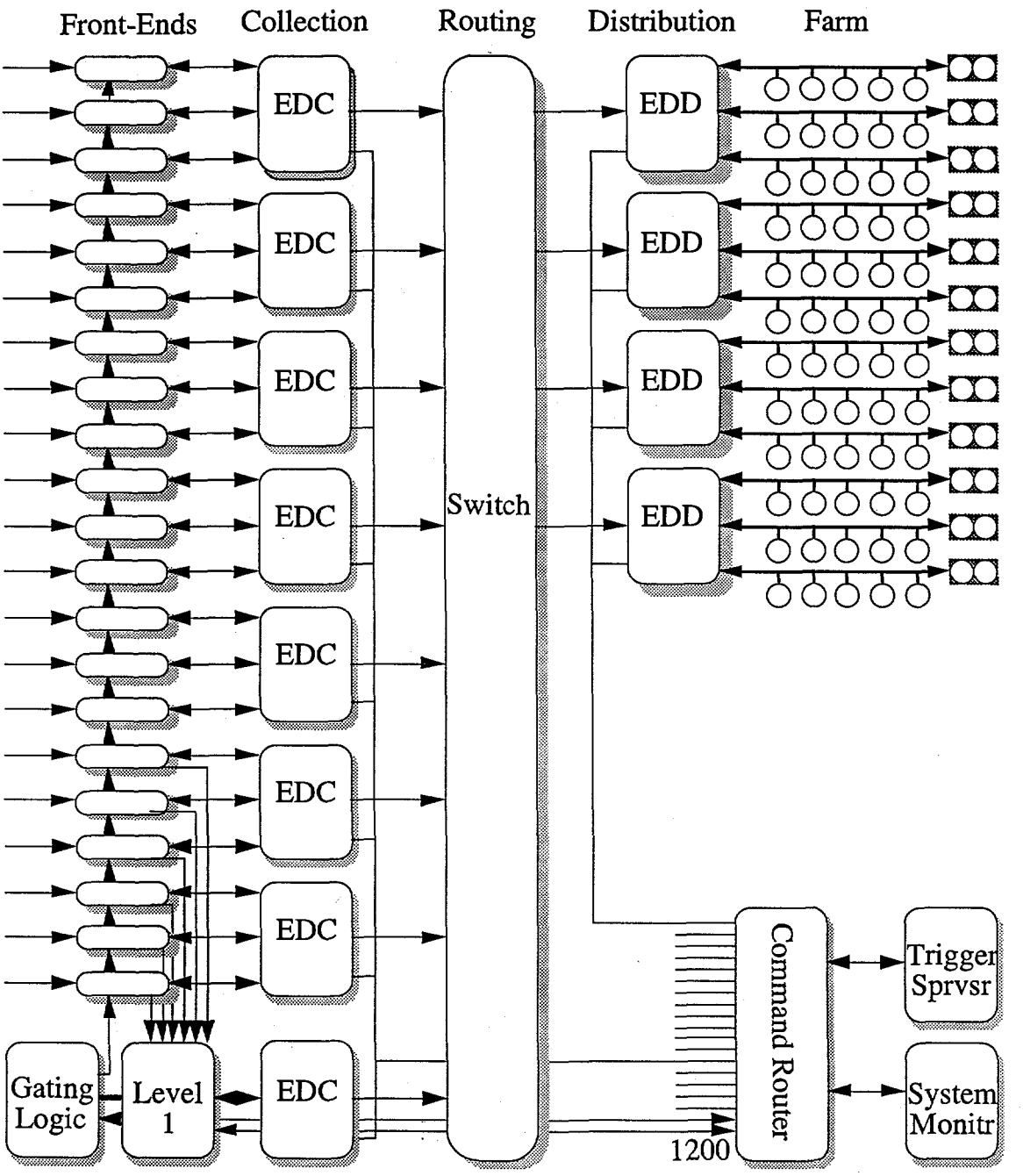


Fig. 1. GEM data acquisition architecture.

For events accepted by the Level 1 trigger, all data is moved off-detector as early as possible. Data is formatted and packetized by the front-ends. Header and control information is added and data is shipped on unidirectional serial links to large random-access data buffers in the control room of the experiment. These buffers are called the "Event Data Collectors" (EDCs). Each EDC stores data from up to 32 front-ends and services a small localized area of the detector. Typically one EDC contains less than 0.5% of the total event data.

GEM does not have separate Level 2 and Level 3 triggers. All higher level pattern recognition is done by up to 1000 CPUs organized in a large processor farm. Each collision is assigned to a specific, single processor for further treatment.

To reduce the dataflow through the switch, a processor retrieves data from EDCs only as needed by the higher level trigger algorithms. A request by a processor for event data is sent to the relevant EDCs only. The data is returned sequentially through a high-speed Event Builder switch. Data packets from the same collision from different EDCs are assembled into single event blocks before they are presented to a processor. Event block assembly is done by special purpose processors, the Event Data Distributors (EDDs). The EDDs forward the assembled blocks to the requesting processors.

Transport latencies are large. The data flow control exhibits a large degree of concurrency, with requests for data from many events (typically 10–20 000) being processed simultaneously.

The system allows up to 512 EDCs and 512 EDDs. The initial system is expected to use approximately 450 EDCs and 128 EDDs. The required number of EDCs is determined by the number of front-end modules to be supported. The required number of EDDs is determined by the data volume.

4. Event data flow control

Control of the data transport from the front-ends to the EDCs is relatively simple. It is fully driven by the Level 1 Accept signal. Event data flow from the events to the processors is request driven (fig. 2). A processor can issue a Data Request for data for a specific event from some or

all of the EDCs. The request flows through the EDD and Control Network to the relevant EDCs. The EDD retains a copy of the Data Request message for use in determining when all requested data has been delivered.

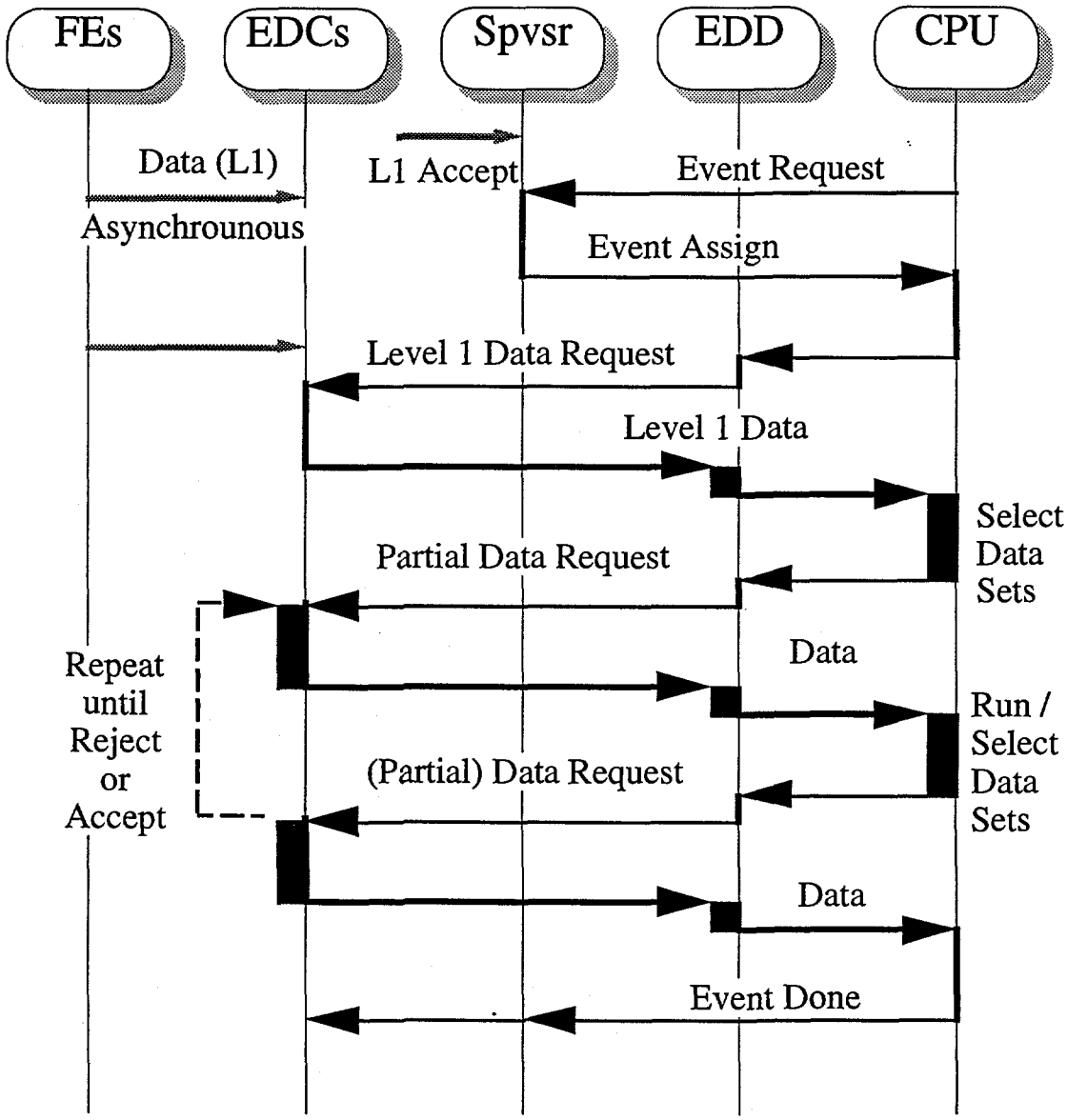


Fig. 2. Data flow control.

The EDCs will transmit the requested data through the Event Builder, back to the EDD. When data from EDCs arrives at the EDD, it is assembled. After all requested data arrived at the EDD, the assembled event is forwarded to the Processor.

The normal operating sequence is as follows: For all accepted events, the Level 1 Trigger sends a Level 1 Accept message with partition and trigger type information to the Trigger Supervisor and Gating Logic. The Trigger Supervisor adds the information to a list of events to be processed. The Gating Logic distributes the Level 1 Accept to the EDCs and all associated front-end modules. The front-end modules (and the Level 1 trigger itself) transmit data to their EDCs. Event data remains stored in the EDCs until the event is rejected or fully read out to the processor farm.

Farm processors that have sufficient CPU power and buffer space may request a new event by sending an Event Request message to the EDD. This message can be a simple request (any event), or it can be a request for an event from a specific partition or of a specific type. The EDD passes this request through the Control Network to the Trigger Supervisor.

The Trigger Supervisor matches this request with a previously received Level 1 Accept message and returns an Event Assign message through the Control Network to the EDD and Processor. The Event Assign message includes the event number of the new event.

4.1. Selective readout mode

To reduce the required data bandwidth (and cost) of the Event Builder, a selective readout option is used in the baseline design. In this mode, a processor will first request the data from the EDC connected to the Level 1 trigger. This data contains information on the trigger type and on the location of the trigger patterns in the detector.

The trigger algorithm first determines the minimum amount of data that will lead to the best possible rejection factor. The processor then issues requests for those data, suspends activity on the new event, and waits for the EDD to assemble the requested data. On arrival of the requested data, the trigger algorithm is resumed. If it is possible to reject the event, an Event Done message

is sent to the Trigger Supervisor. Otherwise more data are requested and the algorithm is suspended again.

Data requests are continued until an event is rejected or accepted. For rejected event, data in the EDCs can be overwritten, for accepted events the full data set must be transported to the relevant processor. Buffer space in the EDCs can be reused after the Event Done message.

A time-out is used to reduce the maximum latency of the higher level trigger process. It forces readout of all EDC data for an event into the processor to make space available in the EDCs. The trigger algorithms in the processor can continue after a time-out since all data are available in processor memory.

Selective readout mode does not appear difficult to implement, but it places additional requirements on the EDCs, EDDs and Control Network. The EDC buffers are larger to accommodate the increased latency, EDC memory management must allow random access to events, EDDs must assemble partial events of varying size (up to a full event), and the Control Network must provide additional bandwidth for the Event Data Request Messages. The higher latency also implies that the processors must handle many events simultaneously, requiring fast context switching and expanded memory.

4.2. Full readout mode

In the Full Readout Mode, all data for each event accepted by Level 1 is transmitted to the processors. This requires a higher bandwidth through the Event Builder, or a lower trigger rate. The advantage of this approach is that the Control Network requirements are minimal. Communication between EDDs and EDCs is limited a to single broadcast Data Request, which reduces the Control Network traffic by approximately 75%. Additional benefits include reduced memory management requirements in the EDC (events are processed in FIFO order) and reduced buffer requirements in both the EDCs and processors (as a result of decreased latency). The increased switch bandwidth requirement with the corresponding increase in the number of EDD modules increases the cost of the readout by about $2–3M.

5. Control network

The Control Network provides a connection to every EDD and EDC module, the Trigger Supervisor and the System Monitor. To reduce bandwidth requirements, it must support broadcast messages. Messages may be directed to individual modules or to any predefined group of modules. For example, when a processor needs data from a specific detector area, a readout message from an EDD might be directed to: an individual Calorimeter EDC, an individual EDC and its nearest neighbors, all EDCs in a selected sub-region of the Calorimeter (partial Calorimeter readout), all Calorimeter EDCs (full Calorimeter readout), all EDCs in a selected η-ϕ region or sub-region in all detector subsystems (partial event readout), all EDCs (full event readout), or any other combination which seems appropriate.

This allows control or readout of selected logical groupings with a single message. In particular, the Level 2 Accept or Reject message, which may be sent to all EDCs, requires only a single broadcast. Logical destination groups are programmable and are downloaded in lookup tables. Each message can be directed to an arbitrary group that can be varied for individual events. This mechanism is also used to obtain detector partitioning.

The Control Network must guarantee order of arrival of messages. i.e., successive messages from a specific EDD to a specific EDC must arrive in the original order. It can be satisfied by routing all EDD-EDC messages across a single physical data path. There is no requirement that messages sent from different EDDs to a common EDC arrive at the EDC in the order they were sent.

The Control Network interface modules will reside in a single crate called the Control Router (fig. 3). Messages are broadcast across the backplane as 64-bit single words. The destination field of the message is decoded by a routing table on each interface module. This enables the transmitter buffer for the selected output links. The routing table provides space for 64-K group codes. Any output link may be a member of any group.

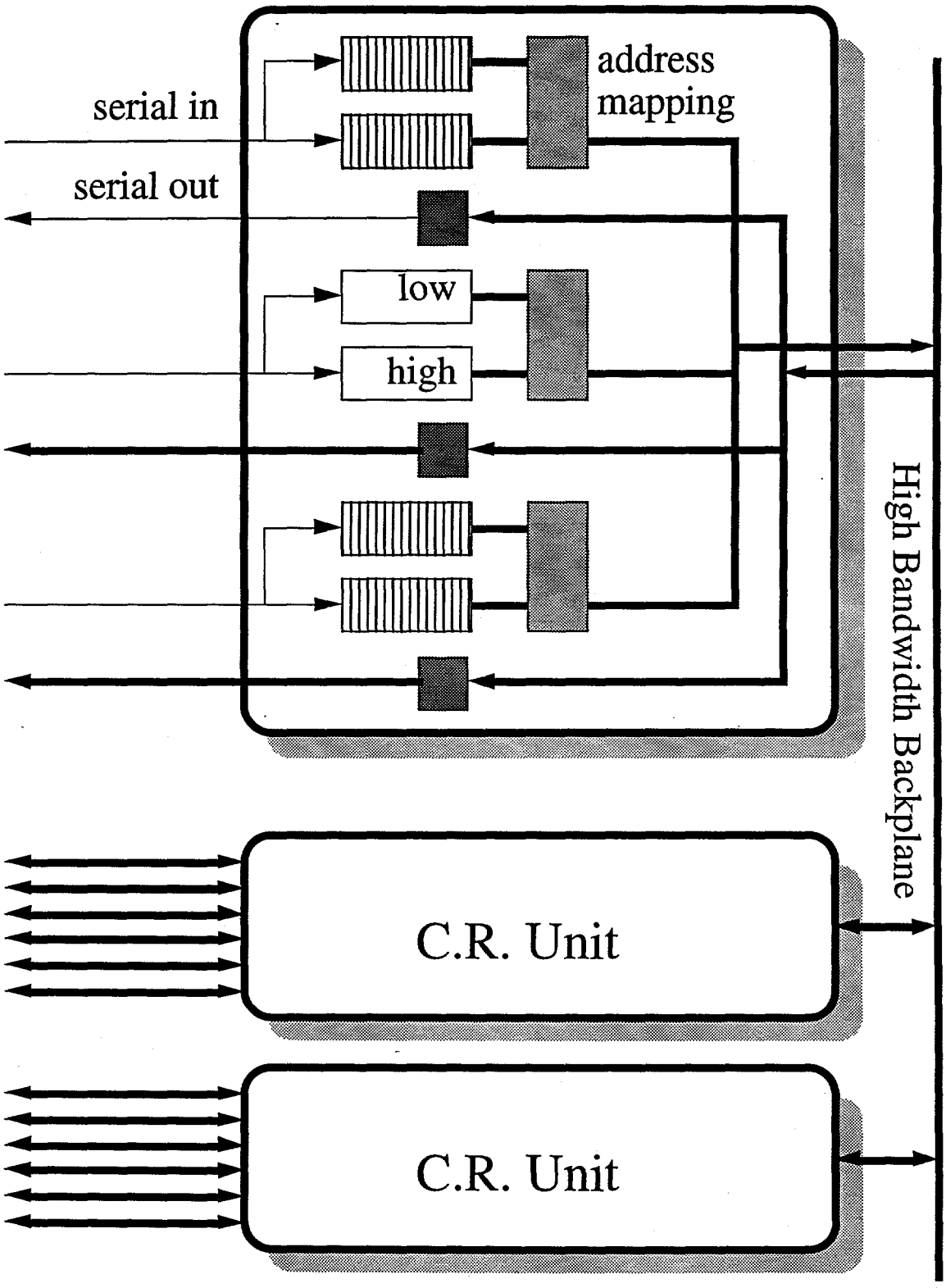


Fig. 3. Control router crate.

6. Trigger supervisor and Gating Logic

The Trigger Supervisor is a central control point in the system. Its main function is to assign event numbers to Processors. It does not directly control the data flow. However, the Trigger Supervisor can help to balance data flow into the processors by ensuring a uniform distribution of event assignments. The Trigger Supervisor can also partition the readout system. This allows for independent readout of separate detector subsystems, for example during commissioning.

Level 1 event numbers are sequential within a partition. An EDC may be assigned to only one partition. It does not need to be aware of the system partitioning. A Processor may only request data from EDCs in the partition corresponding to the assigned event.

The Trigger Supervisor receives the Trigger messages and event type information from the Level 1 Trigger system, identifies the partition and assigns the corresponding events to Processors. If the number of pending event requests from the Processors falls far behind the number of Level 1 Accepts, this indicates that the trigger rate is too high and the Trigger Supervisor may inhibit Level 1 Accepts. The Trigger Supervisor may also inhibit triggers momentarily to allow transfer of synchronous control messages to the front-end system.

The Gating Logic is responsible for distribution of the timed control signals. These signals include the 60-MHz system clock and crossing synchronous control. The gating logic can gate off the Level 1 Accept signals if buffers become too full, hence its name. The Gating Logic is responsible for all scheduling and measurement of dead time. Initialization messages to the Gating Logic are used to map control outputs to specific partitions. Functionally, independent Gating Logic, Trigger Supervisor and Level 1 triggers exist for each partition.

7. Front-end interface

The front-end modules in the GEM system are mostly isolated on-detector devices. For this reason, we have attempted to minimize the number of connections they have. The minimal front-end interface consists of precisely timed Clock and Crossing Sync signals, plus serial Control

and Data Links. Up to 32 front-ends can be connected to a single EDC (fig. 4). Each link is independent, thereby improving the robustness of the system.

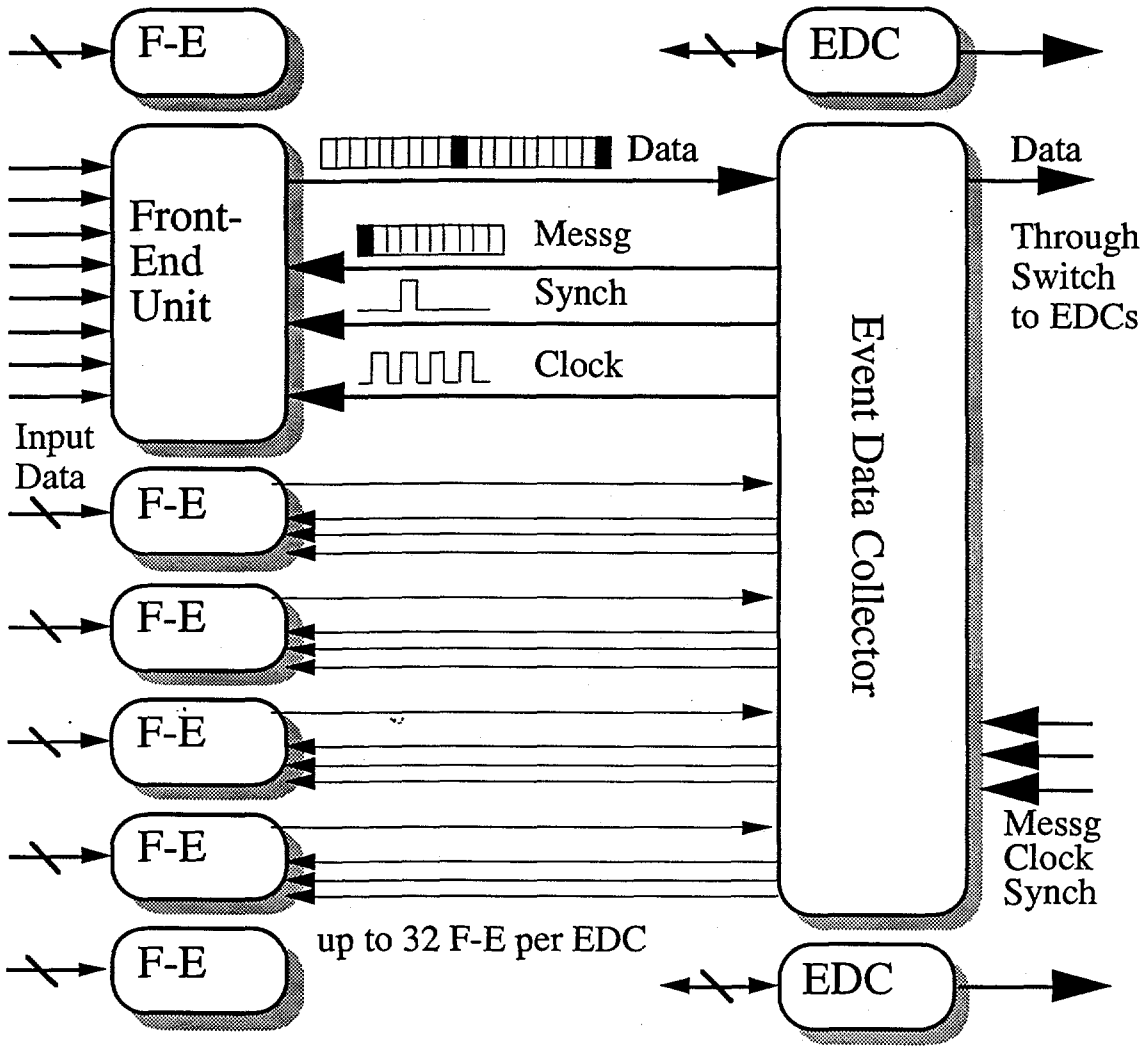


Fig. 4. Front-end connections to EDC.

Each front-end circuit has an internal Event Counter to check event synchronization (fig. 5). The Event Counter is post-incremented by the Level 1 Accept signal. The value of this counter is included in the data. It is used to verify that data blocks belong to the same event.

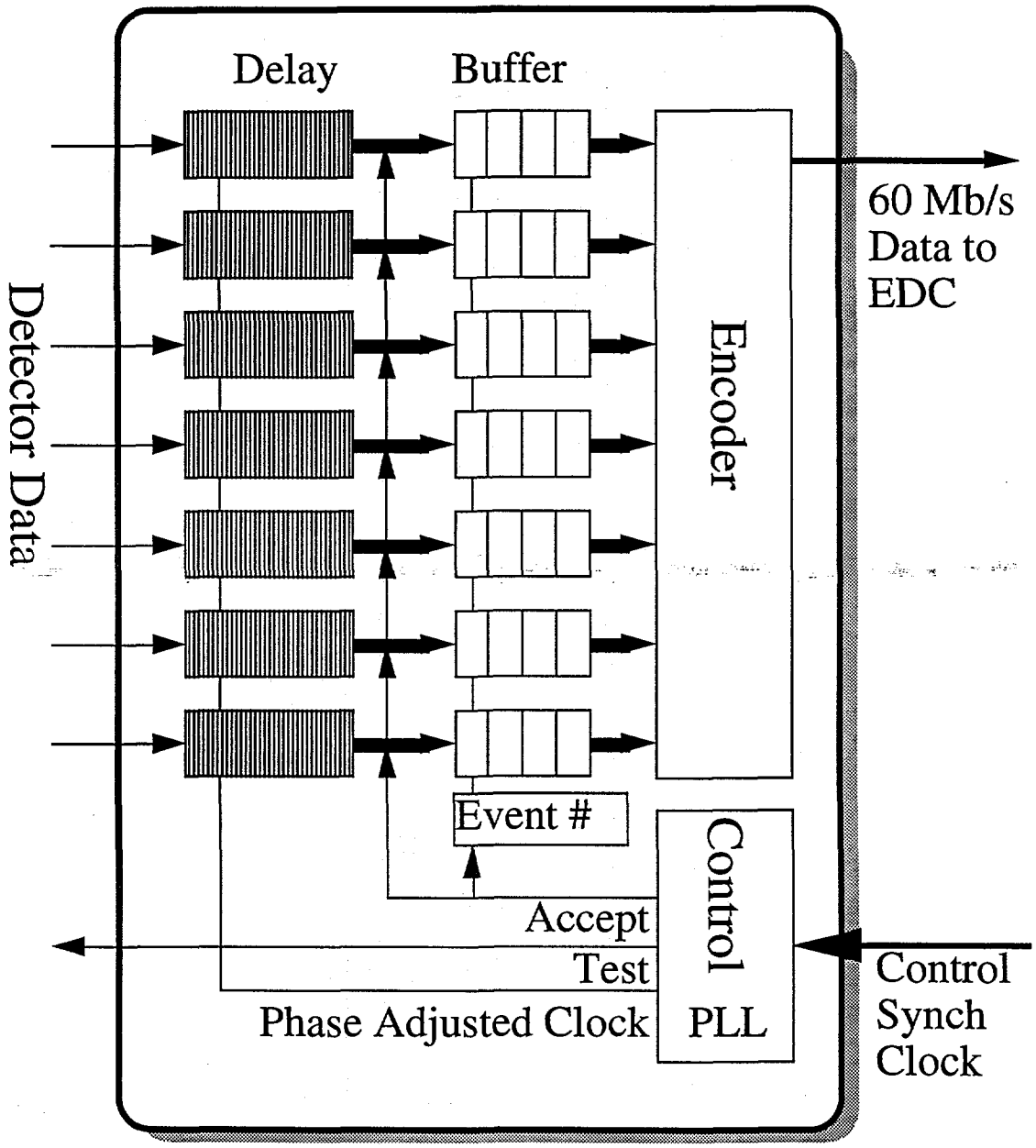


Fig. 5. Functional diagram of front-end module.

7.1. Clock and Crossing Sync

The front-end Clock is derived from the global 60-MHz accelerator clock. The Crossing Sync is a Clock-synchronous signal indicating a specific clock cycle at which an action is to take place. It is mainly used as the Level 1 Accept signal. In the baseline proposal, Clock and Crossing Sync signals are multiplexed on a single optical fiber and separated with a Phase Locked Loop.

The Clock and Crossing Sync signals are generated by the Gating Logic and transmitted via the EDCs which distribute them to individual front-end modules. A phase-locked loop at the receiving end is used to set the vernier delay for precise phasing. The Clock vernier must be set with a precision of about a ns, as Clock may be used for ADC gating and TDC time reference. The resolution requirement on the vernier adjustment of the Crossing Sync is less demanding, as Crossing Sync only has to identify specific clock cycles. However, the Synch adjustment range must span a few clock cycles.

7.2. Front-end data link

Data Links are 60 Mbps ($\leq$ 30 MHz NRZ) synchronous. It is planned to use a direct-coupled receiver for the 60 Mbps links. This reduces the overall power requirements in the front-end module by eliminating the continuous transmission of synchronization frames required in DC balanced protocols. To obtain a BER of 10^{-14} or better, the received signal should be at least 3 dB above the listed receiver sensitivity at standard 10^{-9} operation.

If a standard front-end interface IC is developed, the Data Link bandwidth may be increased to 120 Mbps. This would alleviate the possibility of Data Link overflow in some subsystems at very high trigger rates or at large event sizes. The high speed links will necessarily use an encoded (DC balanced) protocol. The link receivers in the EDCs could be implemented on daughter boards to allow the use of alternate links.

The Data Link format is bit and frame synchronous, with 1 start bit and 16 data bits per word. The protocol allows for two types of transfers, data packets and messages. The two types of transfers are distinguished by a code in the first 4 bits. One data packet is output in response to each Level 1 Accept. There may be any number of idle (zero) bits between packets.

7.3. Data packets

A data packet consists of a header word followed by a unlimited number of data words. For an empty data packet only the header word is sent. A zero in the start bit position of a multiple word event data packet identifies End-of-packet.

Normal FE data packet

st Data Code FE_Status

1	**0**	**1**	**1**	**0**	**E**	**T**	**L**	**E**	**FE_Event_Number**
1	**Data 1**								
1	**Data 2**								
1	.....								
0	**Data N (may be checksum)**								

The header contains a fixed pattern, status information and the FE Event Number. The code in the first four bits distinguishes data from messages. The next four bits contain the front-end status. The FE Event Number is used by the EDC as a check that the correct event is in the input queue and also as an additional check of the previous event's end-of-packet marker.

When running at 60 MHz, a link can send 3.5 million 17-bit words per second. Each event has an overhead of one word. Hence, at a trigger rate of 100 kHz the total overhead is 100 000 words, plus one START bit for every 16 data bits. Total overhead at maximum load is about 9%, mostly caused by start bits. Note that the average data size per event per front-end cannot be larger than 68 Bytes at the maximum trigger rate. At lower rates the packets can be larger.

One of the FE Status bits is used to denote empty data packets. A front-end can set the Throttle Request status bit when internal buffers get almost full. In response, the EDC will generate a THROTTLE REQUEST message to the gating logic, to suppress further Level 1 Accepts. The FE Data Lost status bit indicates that data from this front-end has been suppressed for this event. The FE Error bit indicates that the data may be affected by errors on the front-end board.

7.4. Front-end messages

Message packets are transmitted on the same link as event data packets. A message packet may only be transmitted between event data packets. It is always only one (17 bit) word long. It contains a four-bit code and one byte of information. Front-end modules send message packets only in response to commands. They do not initiate message transfers.

FE message

st	*Msg Code*								
1	**1**	**1**	**0**	**1**	**X**	**X**	**X**	**X**	**FE Message**

7.5. Control link

The control link is used for slow control of the front-end. The baseline design assumes that Control Link messages are transmitted asynchronously at one third the Clock rate (20 Mbps, ≤ 10 MHz NRZ) to allow oversampling. If the routing of the Control Link fiber is the same as the Clock, it should be possible to operate the Control Link synchronously at 60 Mbps (≤ 30 MHz NRZ). It may also be possible to multiplex the information onto the fiber that already carries the Clock and Crossing Synch signals.

Each message includes a start bit and 16 message bits. The Control Link message is defined as follows:

Control link message

1	**P**	**Offset (7 bits)**	**Value (8 bits)**

All messages are mapped into a generic memory space. The memory space is divided into 65 536 pages of 128 Bytes each. "P" is the Page Select bit. If zero, then page 0 is selected. Page 0 contains the most commonly used registers and is always accessible. If the Page Bit is one, then a page is selected according to the contents of the PAGE SELECT HIGH and PAGE SELECT LOW memory locations on page 0, which must be set first. If the front-end module requires fewer than 128 registers, then the page select registers need not be implemented.

Control link messages act like write commands. The "Offset" field identifies one of the 128 registers on the selected Page. The "Value" field contains the 8-bit value to be written to that register.

Read operations are implemented as split transactions. A message is sent with the "Offset" pointing to the READ register in page 0. The "Value" field contains the address of the register which is to be read. The front-end module responds by transmitting the Value of that register on its output Data Link in standard "FE Message" format:

7.6. Synchronous commands and testing

Most commands to the front-ends are asynchronous. The precise timing of their execution is not critical. A few signals, however, related to Level 1 Accepts and full speed testing, must be crossing synchronous. In these cases the Crossing Sync signal is used for synchronization. The interpretation of Crossing Sync depends on the last Control Link write to a synchronous command register.

Synchronous commands are mainly used for high speed testing. For such tests, the test type is determined by the writing to a TEST register. Following a write to the TEST register, the front-end module should generate the proper kind of test event at the next Crossing Sync. It is possible to generate a rapid succession of precisely timed tests. The number of successive Crossing Syncs to be interpreted as test events is determined by the content of the TEST LENGTH register. The precise definition of the test types is subsystem dependent. Test type 0, if implemented, should

return a pre-defined event data packet. Any Crossing Sync signal which has not been preceded by a write to a TEST register is interpreted as a Level 1 Accept.

Asynchronous command registers support commands which invoke an action in the front-end module but do not require a corresponding Crossing Sync. They include commands to restart a front-end, an ECHO command for debugging, and read and write command for internal registers, such as event number, status, front-end id, serial number and module number. Provisions are made for block reads and block writes. A set of registers is reserved for the implementation of IEEE-1149 Boundary Scan.

8. Event Data Collectors

Each Event Data Collector (fig. 6) receives data from up to 32 front-ends on individual data links. Each link is connected to a FIFO buffer with a common output bus driving a large common memory. The common memory must support simultaneous read and write. Sixteen bit wide interleaved, synchronous DRAM or Video DRAM is assumed.

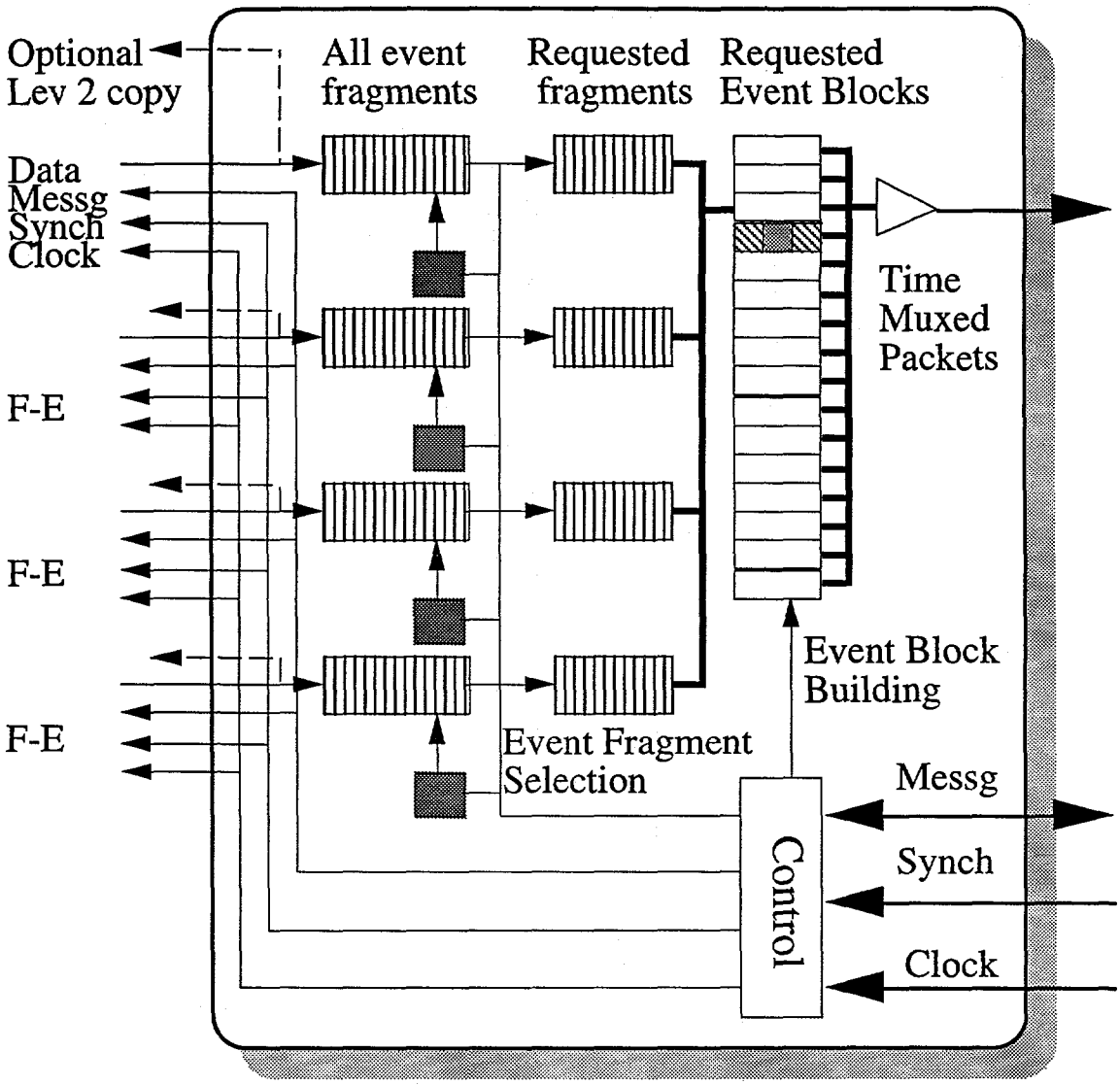


Fig. 6. Event data collector.

8.1. Optional level 2 trigger interface

Data from the front-end modules are transmitted on fiber-optic links. The EDC receivers convert these optical signals to electrical signals before decoding the serial data stream. Differential Pseudo-ECL buffers are provided on the EDC to supply a copy of the serial data streams to external devices such as an optional hardware Level 2 Trigger system. Level 2

hardware must provide the serial decoding and event building functions for the data streams which it monitors. The result of the hardwired Level 2 calculation can then be stored in another EDC for access by the Farm processors.

8.2. EDC data format

The EDC builds the FE data packets into larger event fragments. The EDC prefixes the front-end data packets with a header that contains the overall word count, a unique identifier (EDC ID), the EDC data format descriptor (EDC DD), the EDC status, and the EDC event number. The front-end event number has too few bits to be useful. It is substituted by an event number calculated by the EDC which has sufficient bits to serve as a unique event identifier while the event is processed by the Level 2 trigger. The Level 2 and Level 3 trigger may add another event number. For this format, overhead from headers is of the order of 12% at the maximum data rate.

EDC data packet

EDC Packet Word Count	
EDC ID	
EDC DD	
Status	**Reserved**
EDC Event Number (m.s. Bytes)	
EDC Event Number (l.s. Bytes)	
FE Packet 1	
FE Packet 2	
.....	
Last FE Packet	

The original front-end data packets are transformed. Start bits are stripped and the original packet is prefixed by a word count, a front-end identifier (FE ID) and a data descriptor (FE DD).

The FE ID is a unique module identifier assigned during the initialization phase. It serves as a geographical address. The FE DD describes the format of the data and the format version currently used.

Empty packets with zero status are normally left out, but optionally word count, FE ID, FE DD and the original packet header can be included. For front-ends that are temporarily inaccessible, an empty packet is inserted with the FE Data Lost status bit set to 1.

FE data packet

FE Packet Word Count								
FE ID								
FE DD								
0	1	1	0	0	T	L	E	FE Event Number
Data 1								
Data 2								
.....								
Data N								

9. Event builder switch

Several options have been considered for the switching network component of the Event Builder. These include a basic Time Division Multiplexed (TDM) switch, a general-purpose commercial switching network and a network based on the Scalable Coherent Interface (SCI) standard.

The advantage of the TDM approach is that there is no external control required. The switch is completely non-blocking with no arbitration or handshaking. No packet length, source, or destination headers are required in the data packets because the packet size and order of interconnection are fixed. The disadvantage is that the bandwidth may not be used very efficiently for random traffic patterns. Traffic in classic event building applications is fairly uniform, so the efficiency can be relatively high, but this is not necessarily the case for the selective readout option discussed earlier.

A general-purpose switch is self-synchronizing [5]. On the first set of events, all sources may attempt to send a packet to the same destination. All but one are blocked. On the second set of events, the source which was not blocked on the first set is free to send data to the second destination while the remaining sources all arbitrate again for the first destination. Eventually, all sources are skewed by approximately one event set and blocking is minimal.

An Event Builder based on SCI has also been considered. An SCI network would provide higher efficiency for the selective readout mode, but may not be cost-effective for the full readout mode [6]. For total data rates above 8 GBytes/sec, an SCI Event Builder will need a very high-speed crossbar switch as the center stage of the network.

For random traffic, a general-purpose commercial switching network (e.g., ATM or Fibre Channel) may have higher efficiency. However, existing Fibre Channel switches are optimized for large data transfers, so many events must be concatenated to reduce overhead. Concatenation of events increases the buffer requirements and eliminates the selective readout option. Source and destination headers in the data packets are required for routing. Production cost of a commercial switch is higher due to the general-purpose control and bi-directional data path, but development cost should be lower. An ATM switch may suffer from pathological traffic patterns that occur in this readout when multiple EDC data sources start to send large amounts of data to a common output EDD [7, 8].

The synchronous TDM switch is used in the baseline design because it is thought to be the least expensive (although not necessarily the most efficient) approach. It is implemented as a modular three-stage network with a maximum size of 512×512 channels. It can be built with a limited number of identical crossbar switch modules.

Each crossbar module contains a configuration memory that holds the routing information for each time slot. The interconnection pattern is typically programmed as a synchronous barrel shift rotation, forming a conventional TDM switch with equal bandwidth allocated to each combination of input and output. Inputs and outputs are 1 Gbps (100 MBytes/sec) serial links.

Switch modules are decoupled by a small FIFO on each receiver, which eliminates the need for synchronization at the bit or word level.

10. Event Data Distributors

For each data request by a processor, an EDD (fig. 7) receives data from some or all of the EDCs. It assembles the data to form a partial or full event for transfer to a processor. The EDD-Processor connection is a 1 Gbps bi-directional link. Protocol for this link is defined by the processor and is assumed to be a commercial standard. To accommodate the selective readout mode, the EDD stores the data request messages from the processors. By inspecting the stored processor data requests, the EDD can determine which EDCs are expected to supply data for a given request. Data packets from a given EDC arrives at the EDD in the same order as they are requested.

The EDD maintains a pointer queue for each EDC in the system. As data packets arrive from the EDCs they are stored in the Event Data buffer and a pointer to each packet is stored in the pointer queue corresponding to the connected EDC. The EDD Output Controller loops over the output of the pointer queues and copies one or more packets to the output link. If all data packets forming an EDC event fragment have not yet arrived, the EDD waits at that queue (subject to system time-out). If the routing table indicates no data are expected from an EDC, the EDD skips that queue.

When issuing a data request to the EDD, the processor provides a group address (for EDC mapping), event number, process identifier and address pointer (for the returned data). The EDD does not add any header information to the collected data. When all the data has been written into processor memory, the EDD writes an interrupt message with the original process identifier.

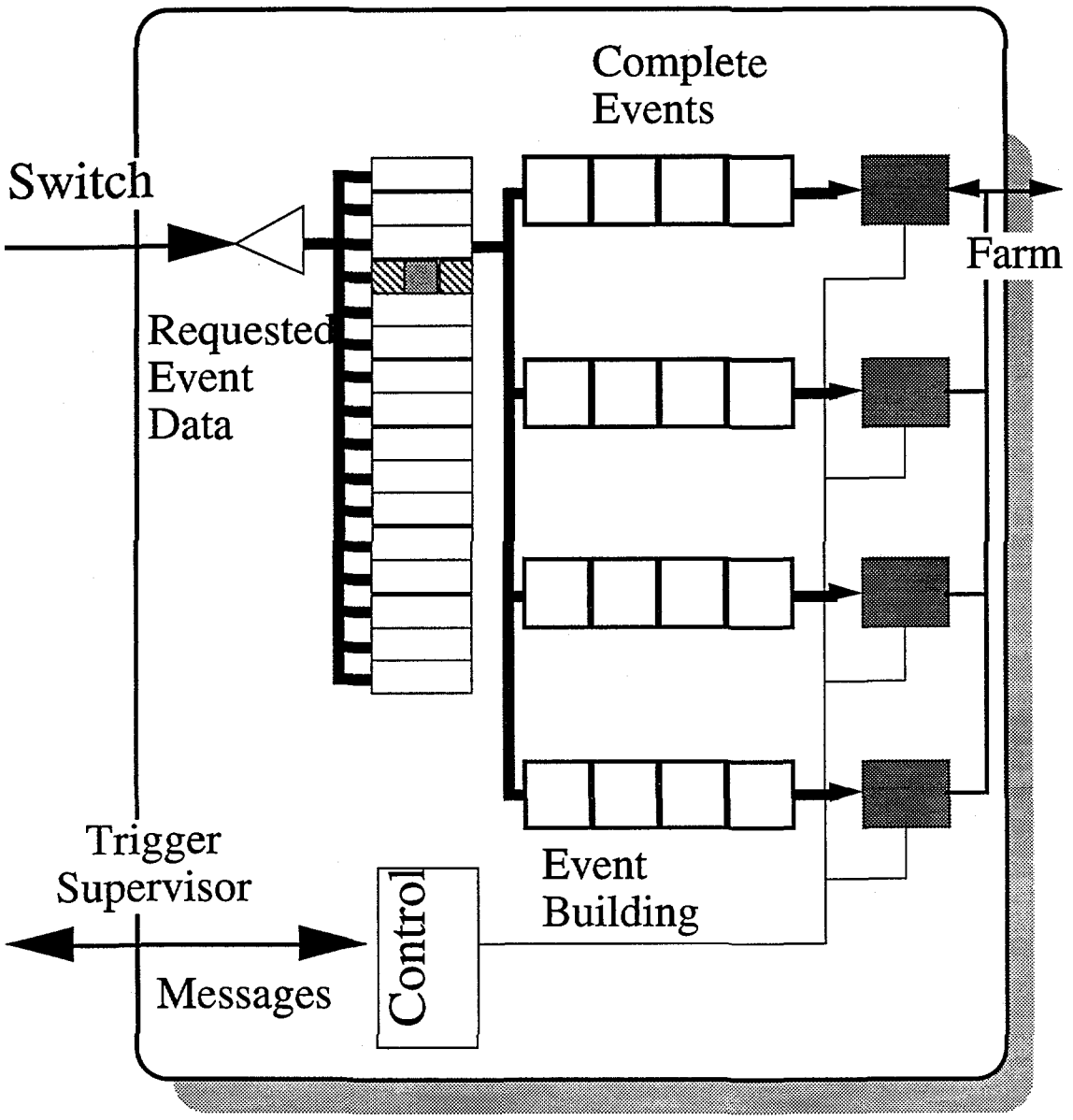


Fig. 7. Event data distributor.

11. Monitoring and error recovery

Overall coordination of the system is done by the System Monitor. Control messages are used to monitor the status of the system. Checks are performed at a rate that allows sufficiently fast error detection to avoid serious data loss.

The protocol used to check the front-ends is optimized to avoid the number of messages going back to the System monitor. The EDCs will accumulate and forward status information to the System Monitor only at low frequency or when errors have occurred. These messages will normally not add significantly to the traffic through the Control Router. Separate EDC status requests may be issued to verify the correct functioning of the EDCs themselves.

11.1. Estimated error rates

The following failure rates (Table 3) are estimates, based on published values for typical standard commercial components. These rates are given as a preliminary basis for determining the error detection / error correction requirements of the system.

Table 3
Failure rates in the different subsystems

Subsystem	Number of Links/Modules	Failure Rate per unit
FE module	10 000	1/day
FE Data Link	10 000	0.25/day
FE Data Link soft error	10 000	20/hour
EDC/EDD	640	0.20/day
EDC/EDD buffer soft error	~40 GBytes	6/day
EDD event assembly	96	0.01/day
Control Router	~20	0.02/day
Control Network soft error	~650	1/day

11.2. Front-end link errors

Most of the errors will occur in isolated front-end modules. We expect a link error every 200 seconds. Less than 10% of these errors will occur in header words or start bits. Errors in detector data are not considered critical (to DAQ system operation) and do not affect the data readout. A data checksum may be included to detect and log such errors.

For most types of serial encoding the probability of a single bit error is low compared to double bit and burst errors, making error correction techniques such as SEC Hamming codes ineffective for serial data links. Block codes (Reed-Solomon, etc.) would be more effective but are relatively expensive.

At the expected link error rate of < 1 error per 10 million events, error correction of any kind is probably not cost-effective. The event data for that link should either be flagged and discarded or the error treated as noise. Provisions must be made to make the higher level triggers aware of errors to avoid trigger biases. For header error detection, it is critical to correctly recognize the location of the End-of-packet marker. Loss of synchronization is detected with very high probability by checking the pattern word and the event number field in the header of the next event.

The event number in the header of each event serves as an incrementing pattern word check by the EDC. The wrong event number may be caused by an Event Counter error, or it may indicate an error in the location of the End-of-packet marker of the previously received event. In either case the EDC removes the front-end module from its readout list. Other front-end links connected to that EDC continue to be read out. An empty data packet with a Data Lost status flag is inserted in place of the missing event fragment in EDC memory.

11.3. Error detection and correction

At the predicted error rates one approach to error correction might be a localized reset mechanism for individual front-ends, while requiring a full system reset in response to most other detected errors.

To resynchronize a single front-end module, an EDC may use the following procedure: Stop delivery of Level 1 Accepts to the front-end module that is in error, reset the front-end module to clear its data output buffer, flush data in internal (EDC) buffers for the front-end, set the front-end event counter to a value slightly ahead of the current event number, and resume delivery of Level 1 Accepts at the appropriate moment.

The EDC will then resume accepting data from the corresponding input channel, starting with the designated event number. During synchronization, it will generate empty FE data packets with the proper status bits to flag the error.

12. Bandwidth requirements

12.1. Data bandwidth requirements

To estimate maximum data rates, we have assumed a Level 1 trigger rate of 100 kHz, an average event size of 400 kBytes, Selective Readout Mode and a Level 2 trigger that gets a rejection of 90% using 10% of the data in the EDCs. The Level 2 Accept rate is 10 kHz.

The link between a front-end and the EDC is used after each Level 1 Accept. At 100 kHz trigger rate, the average data bandwidth is 4 MBytes/s. Header words and control bits add overhead for a total of 40 Mbits/s. Since the maximum throughput of these links is only 60 Mbits/s, good load balancing is very important.

For Selective Readout Mode the switch throughput depends on the higher level algorithms that are used. If Level 2 can get a 90% rejection using 10% of the data, it will need 4 GB/s. Complete, accepted events would be read at 10 kHz, requiring another 10 kHz * 400 kB = 4 GB/s. The total data bandwidth requirement adds up to 8 GB/s, while 128 EDDs can absorb a

maximum of 12 GB/s. Event fragment headers will add less than 2%. Overhead in a barrel switch is expected to be minimal.

In Full Readout Mode the required bandwidth through the event builder switch is 100 kHz * 400 kB = 40 GB/s. As each switch output channel runs at 1 Gbits/s, this option requires the maximal number of 512 EDDs to be present, giving an aggregate maximum throughput of about 50 GB/s.

12.2. Control bandwidth requirements

Control traffic is dominated by messages related to data readout on behalf of Level 2 and 3 and by synchronization messages to the front-ends. All messages between EDDs, EDCs, Trigger Supervisor and Gating Logic are passed through the Control Router (CR), which may present a bottleneck.

In total, messages are coming into the Control Router at about 1300 kHz. Per message, about 760 ns is available on the backplane of the Control Router. Messages from the Gating logic related to synchronization will have to be routed with high priority. They occur at relatively low frequency. The link into the Trigger Supervisor has a very high rate. This may require the use of a special purpose processor for message processing.

In Full Readout mode The EDDs only issue one readout request per event. It is broadcast to all EDCs. In this mode, the traffic into the Control Router crate is reduced by about 70%.

13. Conclusions

The design of the GEM readout system meets all of the DAQ requirements. It provides a scalable system, where the most complex data handling is performed in units that reside in the accessible environment of the control room. Front-end modules are kept simple and have minimal data storage requirements. They are controlled via individual links to increase the robustness of the system. Extensive attention has been given to system monitoring and error

detection. Isolated front-end errors can be detected and flagged, while automatic localized reset procedures may provide error recovery in most cases.

Extensive simulation of alternative readout architectures and protocols has been done [9, 10, 11]. It has been shown that sufficient bandwidth is available [12]. The main concern is the limited bandwidth in the data link between front-ends and EDCs. It requires very careful load balancing over the front-end modules. Alternatively, the use of higher speed fibers can be envisaged, where needed.

We have provided a full definition of the front-end interface so that the design of the system components with the longest lead times can proceed. The readout system itself is built out of a limited number of different units to reduce engineering cost. Where applicable, options to use commercial equipment is left open. No special purpose hardware needs to be developed for Level 2 or Level 3 triggers.

References

[1] Lefmann et al., GEM Technical Design Report. SSCL, GEM-TN-93-262, April 1993.

[2] GEM collaboration, GEM Status at Closeout,. SSCL, GEM-TN-94-551.

[3] E. L. Berger, SDC Technical Design Report. SSCL-SR-1215, April 1992.

[4] M. Bowden, J. Carrel, J. Dorenbosch, V. S. Kapoor, A. Romero, GEM Data Acquisition System Preliminary Design Report. SSCL GEM TN-93-471, Sept. 1993.

[5] D. Black et al., Results From a Data Acquisition System Prototype Project Using a Switch Based Event Builder. IEEE Nuclear Science Symp., Santa Fe, NM, Nov. 5–9, 1991.

[6] H. Muller, A. Bogaerts, R. McLaren, C. Parkman (CERN), D. Linnhoefer, New buses and links for data acquisition. Nucl. Instrum. Methods A315 (1992) 478–482.

[7] J. Christiansen, M. Letheren, A. Marchioro, NEBULAS: An asynchronous self routing packet switching network architecture for event building in high rate experiments. CERN-DRDC-92-22.

[8] J.A.C. Bogaerts, R. Divia, H. Muller (CERN), J.F. Renardy (Saclay), SCI based data acquisition architectures. IEEE Trans. Nucl. Sci. 39 (1992) 85–94.

[9] E.C. Milner et al., Data Acquisition Studies for the Superconducting Super Collider. IEEE Tran. Nucl. Sci., 39, 1992.

[10] M. Botlo et al., Data Acquisition for Super Colliders. SSCL-PREPRINT-156, May 1992. Como Conference, Como, Italy, May 1992. Nucl. Phys. B 32, 1993.

[11] M. Botlo et al., Data collections studies for the Superconducting Super Collider. Nucl. Instrum. Methods A315 (1992) 472–477.

[12] James G. Branson, A Simulation of the Proposed GEM Data Acquisition System. GEM TN-94-632.