H-643 Version 2.2

The E781 Trigger and Data Acquisition System

Jürgen Engelfried, Peter S. Cooper, Dongning Mao

April 30, 1995

Abstract

In this document we like to collect all information concerning the actual data taking for Experiment 781 (SELEX). This includes the trigger system, the frontend electronics, the readout, the event format, the filtering, and the on-line monitoring.

This document will be constantly updated. The current version can be found inside the E781 WWW server.

Contents

1 Introduction

1.1 Scope of this Document

This document is an attempt to provide an overview of the DAQ system for E781. The subsystems and components which make up the E781 DAQ are enumerated and described at the functional level. (what they do, not how they do it.) The important system parameters are also included. For more details about particular sub-systems and components refer to the references and to members of the E781 DAQ group. You can reach the group by sending an email to e781-daq@fn781a.fnal.gov.

This is intended to be a working document. The system, as of this writing, is under active design and development. New versions of this document will appear periodically - sometimes with a very short period.

1.2 Goals of Experiment 781

E781 (SELEX) is a Fermilab fixed target experiment scheduled to take first data in the 1996 fixed target run. The physics goal of the experiment is to observe a large sample of charmed baryon decays. The scale of this sample should be of order 500K reconstructed charm baryon decays summed over all decay mode and all four stable charmed baryons. There is also a set of photon physics topics (Primakoff, Hyperon form factors, exotic mesons, ...) to be studied parasitically to the main charm data taking.

The data volumes and rates necessary to reach this level of sensitivity require a data aquisition system which is beyond the present state of the art data. To summarize this requirement as a single set of parameters, the DAQ rate during the 20 second beam spill is 35 MByte/s for a total of 700 MByte of data acquired per spill.

A crucial aspect of the E781 DAQ is the use of substantial on-line computing ($\approx$ 1000 MIPS) running complicated physics algorithms on the raw data in real time to achieve a reduction of a factor of 40 in the volume of data written to tape (700 MByte/40 = 17.5 MByte/spill to tape).

1.3 Overview about the DAQ

The E781 data acquisition system is part of the DART project, a collaboration between the On-Line Systems Department of the Fermilab Computing Division (OLS) and 6 Experiments (E781, E811, E815, E831, KTeV, E835). The idea is to develop a core system common to all experiments, in the hope of reducing the manpower needed to set up individual systems for each experiment. To obtain information about the DART software and hardware, please use the DART World-Wide Web (WWW) server, which is accessible from the Fermilab Home Page and from the E781 WWW server.

This write-up will describe the actual implementation of DART in E781, together with the trigger and front end readout layout.

The design of the system is based on the following assumptions, which will be explained in detail later in this document:

1. The beam rate is 2 MHz, and the beam consists to 50 % of π^- and to 50 % Σ^-; the π^- will be tagged by the beam TRD.

2. We have a 5 % interaction length target.

3. Trigger level 1 (beam interactions scintillator hodoscopes behind the magnets) reduces by a factor of 5.

4. Trigger level 2 (more refined momentum analysis and lambda trigger) reduces by another factor 2.

5. The Primakoff trigger contributes not more than 10 % (per definition).

6. A software level 3 filter selects charm events with a secondary vertex and has a rejection of 40.
7. A typical event is 5 kByte in size.
8. The total dead time should not be more than 30 %.
9. Only SVX and PWC data (25 % of the total event size) are used for filtering.
10. The spill lasts for 20 s, and the cycle time is 60 s.

This leads (with some safety margins) to a raw trigger rate of 10 kHz or 100 μs mean distance between triggers. We have to be ready to take the next event in maximum 30 μs (on-line dead time) and to readout events in less than 140 μs (7 kHz). The limits on the readout dead time we assume is even 100 μs. The total number of events read out in one spill will be about 140000, with 700 MBytes of data to transport. 300 kBytes/s from charm events and about the same amount for Primakoff events have to be written to Exabytes. A schematic view of the whole system is shown in figure 1.

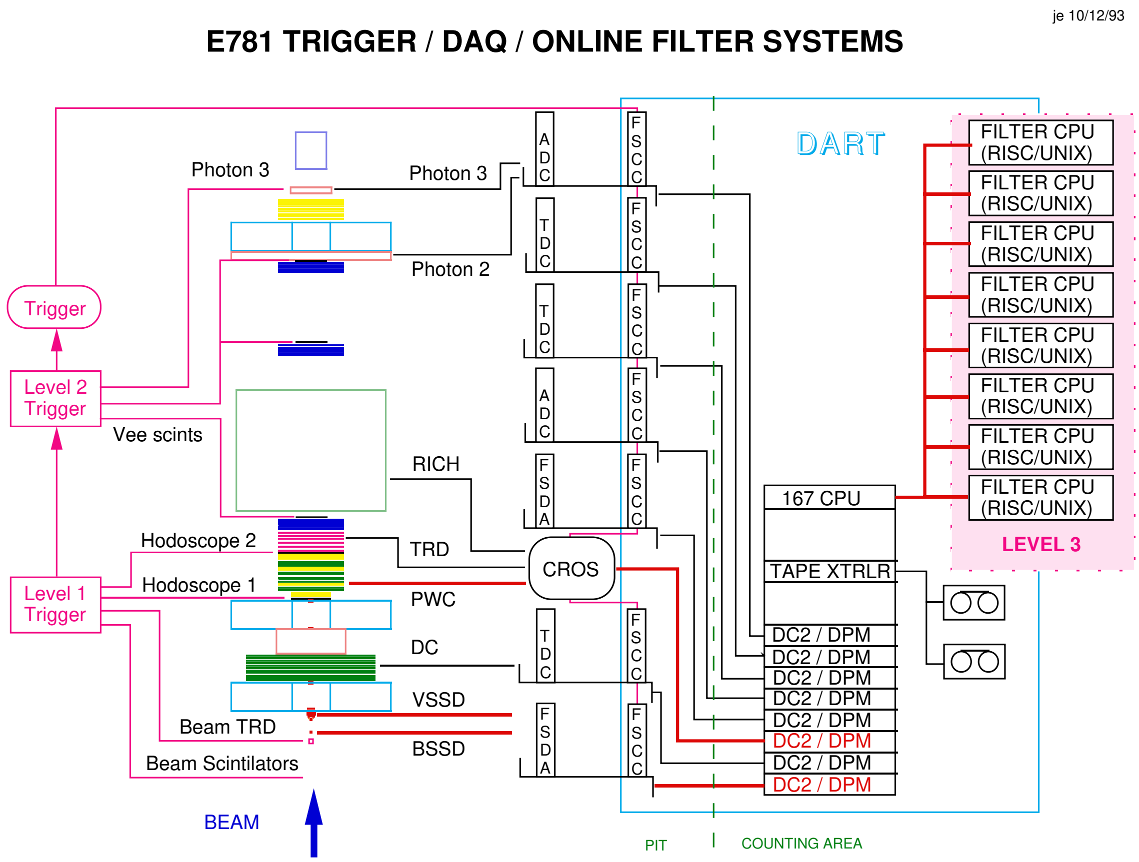

Figure 1: Schematic layout of the E781 data acquisition system

The readout of the different detector systems will take place in about ten parallel streams. The frontend consists out of Fastbus SVX Data Accumulators (FSDA) for the Silicon Microstrips detectors, Fastbus TDCs for the drift chambers, Fastbus ADCs for the photon detectors, all read out with Fastbus Smart Crate Controllers (FSCC), the Chamber Readout System (CROS) for the proportional wire chambers, the TRD, and the RICH, and the RMH system for proportional wire chambers. In addition, we will have to read out one or more CAMAC crates with scalers, latches, etc. All this frontend electronics will be located in PC4 near the actual detectors. The data will be transferred to the "Schoolhouse" (Portakamp 174) and pushed in Dual Ported Memories sitting in a VME crate. The link for each stream has a width of 32 bits and a speed of up to 40 MBytes/s. Physically it consists at the ends of a 50 leads and of a 34 leads flat ribbon cable, which are connected via optical fibers to accommodate for the total length of about 100 m.

Each DPM in the VME crate has 64 MBytes of memory, except the FSDA stream, were we need about 180 MBytes to store the 140000 events. The pointers to each event fragments are stored in the DPMs, and the Challenge will transfer part the data from two of the streams (Silicon, PWCs) and distribute them to filter processes.

After a positive decision a MVME167 (68040) CPU module will collect the pointers to all fragments of an interesting event and logs them with the help of a VME SCSI controller to one or more exabytes.

The system makes use of both the spill and the interspill period for the filter level. Nevertheless, about 1000 MIPS of computing power is needed for the filter. The transfer speed needed to transport the filter streams to the filter processors is about 3 MBytes/s. No "normal" users will be allowed on this machine(s). For run control and as central server, additional computers is used. The whole system is heavily UNIX based.

For monitoring purposes, the MVME167 is copying whole events over Ethernet (TCP/IP) to any computer which wishes to analyze events and has an Ethernet connection. Some standard check tasks will run on a separate machine constantly.

The described system is only used for actual beam data. Interspill calibration and checks have to run on the local readout controllers (FSCCs), which are full computers with Ethernet connections. There also the analysis of the events should be done (e.g. histogram the pedestal of ADCs), and only summary files (the histogram) should be transferred to the central server. To minimize the number of different computers to be support in the system, no special computers (for example PCs) will be used for in situ checks of the equipment.

2 Trigger

2.1 Introduction

We will distinguish between trigger when the beam is on for 20 s ("onspill") and triggers for calibration during the 40 s with no beam ("offspill"). The latter are special for each detector and are described later. The on spill triggers consists of the following main sources.

- Charm Trigger
- Primakoff Trigger
- Prescaled Beam triggers
- Prescaled π trigger for TRD check
- Prescaled rejected level 1 trigger
- Prescaled rejected level 2 trigger
- Pulser trigger (random) for pedestal and noise measurements
- Laser and LED trigger for on spill lead glass calibration
- LED trigger for RICH phototube checks

All non-physics triggers will run at a very low rate (a few Hz). Some of the triggers can be combined (e.g. Laser and LED).

2.2 Trigger Level 1

For the charm trigger, the level one will consist of a coincidence of the Beam Scintillator, the beam particle tag from the Beam TRD, the interaction counter and the output of the Hodoscope 1 and 2 lookup matrices. To reject interactions outside the target pulse height information from 2 vertex silicon Microstrips planes will be used. More information about the Hodoscope matrix trigger can be found in H567 and H676.

For Primakoff physics, the level one trigger will request no interaction and only one scattered track passing through the apparatus.

For timing reasons, only detectors upstream of Hodoscope 2 (shortly behind M2) will be used for trigger level 1. All detectors located more downstream have to be included in level 2.

2.3 Trigger Level 2

For the charm trigger, the momentum will be determined by a matrix, and the information from hodoscopes around the VEE chambers can be used to identify events with a Λ^0 in the final state (propably not used). An overall reduction of the combined level 1 and level 2 trigger by a factor of (at least) 10 is expected.

The Primakoff trigger, using information from the Photon detector(s), is not yet defined. Some preliminary estimates can be found in H576.

2.4 OffSpill Calibration

Each of the detectors has its own requirements in calibrating and monitoring the equipment. Between spills all these checks should be performed locally on the corresponding readout controller and only results should be written over Ethernet to the central DAQserver machine. The trigger for the calibration events has to be generated locally. Each detector group is supposed to provide the appropriate code for the check tasks. To minimize the number of different systems, the DAQ group will provide a shell structure for the programs.

3 Frontend Readout

3.1 Introduction

Here we will describe the setup for the frontends in detail. First the different readout systems (controllers and crates) will be presented, and later the specific applications for each detector are written up. The system is designed to run all streams in parallel as (nearly) independent DAQ systems, to fulfill the requirement for $< 30\,\mu$s dead time and $< 100\,\mu$s mean readout time per stream. All streams will write data in RS485/DART standard into the VME event building crate.

All the independent data sources have to generate their own dead time signal. The ready signals (or busy signals) will be anded (ored) together to inhibit triggers when any equipments is not ready to accept a new event.

3.2 The Fastbus Smart Crate Controller (FSCC)

The FSCC is a Fastbus Master consisting of four major parts:

1. A 68020 Processor from Motorola running the VxWorks operating system, with up to 4 MBytes of memory, Ethernet connection, and a UNIX-like command structure. This is a full computer. Programs are written in C (or in C++ – not recommended) and cross compiled. A 8bit Parallel Input/Output Register can be accessed by the processor.
2. A Sequencer together with a List FIFO, performing fast readout sequences on Fastbus. No conditional branches and intelligent decisions can be performed here. This is a pure readout engine.
3. A 4 bit wide Trigger Input Register with a buffer FIFO, accessible from the processor and the Sequencer.
4. A Output FIFO (16 kBytes) connected to the auxiliary port driving the RS485/DART standard at a speed of 40 MBytes/s (100 ns cycle time).

The Processor will be used for starting the list for the Sequencer, and to perform the offspill calibrations. Onspill we will use the list sequencer mode to read out data at the highest available speed from the Fastbus modules to the Output FIFO. Readout Triggers are inputted through the Trigger Input Register, which consists of a FIFO to store incoming triggers during the readout of the previous event. We plan to give a 4 bit event number as trigger signal to assure the synchronization between different readout streams. When we use the List FIFO to read out the modules, the FSCC can not generate any busy or ready signal directly, since the sequencer has no access to the parallel output register; but we can use the Trigger-FIFO-empty signal on the JO to indicate a busy FSCC. The individual Fastbus modules or some external logic have to provide the signal when they buffer multiple events. For readouts under processor control (not used in our case) the FSCC could provide this signal himself.

The FSCC prepends a 32bit header word to the data, containing the inclusive longwordcount and the 4bit event number. We will add via a list instruction a second word containing a unique FSCC identification and the spill number.

Bits	31 16	15 12	11 0
Word 1		Event Number	Longword Count
Word 2	FSCC Id	Spill Number	

Table 1: FSCC Header words

3.3 The CAMAC Smart Crate Controller (SCC)

This module is used as readout controller for a few CAMAC modules. The program (readout list) running on this module has to be cross-compiled (on a VAX in this moment) and downloaded via a serial link. It will output over an additional adapter board 16 bit wide data in ECL/FERA, feeding a DYC+. Further specifications are not yet known. An event number could be provided with an extra CAMAC module. The readout speed is maximum 600 ns/word, with an up to now unknown offset to start the readout. In any case, the readout of the whole crate has to be done in $< 30\,\mu$s.

3.4 Damn Yankee Controller (DYC+)

This module, build by the Fermilab Physics Department, converts ECLine signals from FERA or similar systems to RS485/DART. It has a 16 kByte multi event buffer, generates a leading 16 *mboxbit* exclusive word count, and copies a 4 bit trigger input to the header. We will use this module in E781 to buffer data from CROS, RMH and CAMAC sources (SCC). The DYC+ packs 16 bit words in 32 bit words. Several DYC+s can be chained into one stream, even mixed with FSCCs.

Bits	31 16	15 12	11 0
		Event Number	Word Count

Table 2: DYC+ Header word

3.5 The Chamber Readout System (CROS)

This system consists of Receiver-Memory-Latch units (16RML32) with an internal delay (circular shift register of depth 16), sitting in CAMAC crates, but not following the CAMAC standard. One module holds 32 channels, the input signals are differential ECL signal on two 34 pin connectors. Please note, that the connector is reversed compared to normal differential ECL (pin 1 and 2 are ground, not 33 and 34). The crate controller (CSC) reads out the latches with 100 ns/32 bit and converts the hits in a pipeline to 16 bit words and stores them in a buffer memory. One can specify the number of slices to be read out; only slices with at least one hit in the 32 channels are read out. All crates are read out in parallel. All CSCs are then read out by a microprogrammable module (CSB.V) with a speed of 100 ns/16 bit, and outputs data in ECL/FERA to a DYC+, and/or stores them in an internal memory to be read out via CAMAC. Up to 16 CSCs can be read out by one CSB.V.

For the deadtime we will use the busy signal from the CSC. This enables us do use a "double buffering" scheme, since the data are buffered in the CSC and the 16RML32 can accept the next event.

We plan to use 2 readout controllers (one for TRD and RICH, one for the PWCs) and chain the DYC+s from CROS and RMH into one stream (total rate $\approx$ 9 MByte/s)

The CROS system produces several different headers depending on the selected sparsification mode in the CSC (see table 3). The first longword will be a CSB.V identification, followed by an event number, also generated by the CSB.V, than follows the header from the first CSC, followed by the data. The CSB.V appends the event length at the end. Please remember that also the DYC+ produces a header word.

3.6 RMH

The frontend RMH modules are discriminator and latches; one module holds 32 channels. The system uses dedicated CAMAC crates with up to 22 modules per crate. Up to 16 crates can be branched together for the readout.

Bits	15	14	13	12	11	10 4	3 0
CSB Id Word 1	CSB Id						
CSB Id Word 2	0						
CSB header	1	1	1	1	Event Number		
CSC header	1	1	0	0	0	Sparse Mode	Crate Number
CSB trailer	1	1	1	0	Event Length		

Table 3: CROS Header words

Every crate holds up to 22 modules, a Crate Controller which will receive and distribute the trigger signal (Gate), and a Crate Encoder which connected to the Branch Receiver for the readout. The RMH modules have no internal delay; the (analog) signals have to arrive at the same time as the gate. The analog signals have a typical width of 20 nsec, the gate has to be applied at least 20 nsec before the analog signal and has a typical length of ≈ 50 nsec. The threshold can be adjusted via a screw on the module. Since it is only a latch, no Clear signal is needed before the next gate can be accepted after a rejected event.

The readout is performed with 3 Camac modules with are connected internally: the Camac Interface, the System Encoder, and the Branch Controller. The data will be output to a DYC+. The readout works at a speed of 200 nsec/hit. For the data format see table 4. The station number is calculated like $\mathrm{station} = (\mathrm{crate} - 1) * 22 + \mathrm{slot}$. The crate number is defined via a switch on the Crate Controller. No headers are provided by this system.

Bits	15	14	13 5	4 0
SE Id Word 1	SE Id			
SE Id Word 2	0			
RMH Data	Flag	/	Station Number	Channel Number

Table 4: RMH Header and Data words

As deadtime signal we will use the BUSY from the SE. No double buffering is possible.

For Offspill calibration issues the system can also be read out via CAMAC.

3.7 Silicon Microstrips (FSDA)

All silicon detectors (85000 channels) will be read out via the FSDA. The readout of the SVX chip is controlled via the SRS, a CAMAC module, also synchronizing the FSDA. The FSDA receives the data from the SVX and, after an additional zero suppression and mapping, puts them in its output memory, which can hold several events. About 20 modules are needed to readout all microstrips. The estimated time for this readout is about $25 - 30\,\mu$s (600 ns/hit, plus $\approx 7\,\mu$s overhead). The FSDA has a multi-event buffer.

The FSDAs itself are then read out via an FSCC in megablock transfers. The FSDAs can be read out with 250 nsec/32 bit , with an overhead of 1.64μsec and a 60 nsec switch time from one FSDA to the next. It should be possible to put all FSDAs in one Fastbus crate and still meet the 100 μsec readout requirement, but we will split it in two crates.

The SVX chip is an integrating device and has to be cleared from time to time. During this clear time we have to veto the main trigger.

Each FSDA provides a 32 bit header with its id, an exclusive 16 bit word count and an event number.

The busy signal is provided by the SRS for this system. OffSpill calibration will be performed on the local FSCC.

Bits	31 28	27 24	23 16	15 12	11 8	7 0
Header	Plane Id		Status	Event Number	Word Count	
Data Form 1	Amplitude	Strip Number		Amplitude	Strip Number	
DataForm 2	Strip Number		Amplitude	Strip Number		Amplitude

Table 5: FSDA Header and Data words

3.8 Drift Chambers (TDCs)

The discriminated signals from the more than 5000 wires of the drift chambers are digitized in the 10C6 Fastbus TDCs from Philips. One modules has 32 channels, in total we will use about 170 modules in 8-10 Fastbus crates. The 10C6 has no internal buffer (only a LIFO), the events have to be read out by the FSCC immediately, before a new event can be accepted. The conversion time is about $7\,\mu$s. The hit information is stored as 16 bit words. With 640 channels per crate and an occupancy of 15 % (6 hits and one header per module) this leads to a total dead time of $27\,\mu$s, when the data are read out with Megablock transfer (250 ns/32 bit) under processor control.

We have to use special crates with the termination resistors reduced from $50\,\Omega$ to $35\,\Omega$, because the drivers on the Fastbus port of the 10C6 are not strong enough.

Each 10C6 provides a 32 bit header with an exclusive 16 bit word count and an event number, which will always be 0.

Bits	31	30 26	2521	20 16	15	14 10	9 0
Header				Geogr. Addr.	# Channels		Event Id
Data		Channel Id	Data			Channel Id	Data

Table 6: 10C6 (TDC) Header and Data words

For the deadtime we will use the TriggerFIFO-not-Empty of the FSCC, since the contents of the Trigger FIFO is copied to the Output FIFO after the readout is finished.

We can connect up to four FSCCs to a common stream (the transfer time would be $59\,\mu$s ($27\,\mu\text{s} + 4 * 80 * 100\,\text{ns}$) under normal occupancy), but we will use two or three streams (total flow ≈ 25 MByte/s)

OffSpill calibration will be performed on the local FSCCs.

3.9 Lead Glass (ADCs)

The analog signals from the 2000 lead glass blocks will be digitized by LeCroy 1881M ADCs. One 1881M contains 64 channels, so 32 modules or 2 Fastbus crates are used. The 1881M has an onboard buffer to store up to 64 events. The conversion time is $10\,\mu$s and the information for each hit is stored in a 32 bit word. To read out a module, a "load next event" command has to be issued to each module via a Fastbus broadcast, which takes about $5\,\mu$s. The Fastbus crate has to be tailored to minimize this time. The 1881M provides a 32 bit header word with an inclusive longword count and an event number.

The modules will be read out separately by the FSCC in block transfer mode or Megablock mode. We need about $50\,\mu$s to read out 20 modules with 10 % occupancy to the FSCC. The FSCCs can be chained into one stream (total flow ≈ 10 MByte/s).

The auxiliary card of the Fastbus crate has to be trimmed to allow a broadcast time of about $5\,\mu$s.

The busy signal will be provided either by an external logic or by the ORed CIP (conversion in progress) signals.

Bits	31 27	26	25 24	23 17	16	15	14	13	12 7	6 0
Header Form 1	Geo Addr					P			Buffer #	Word Count
Header Form 2	Logical Addr.					P			Buffer #	Word Count
Data	Geo Addr	P	Buffer #	Channel #				Data		

Table 7: 1881 (ADC) Header and Data words

The OffSpill calibration will be performed on the FSCCs.

3.10 Beam TRD (???)

This detector is used in the trigger. Very little is known about the readout of this system, but only a few channels have to be read out, either in a pattern unit or in an ADC.

3.11 TRD (CROS)

3072 channels are connected to 96 modules in 6 crates. For readout, see the discussion above.

3.12 Proportional Wire Chambers (PWC and LPWC) (RMH)

10300 channels are connected to 16 crates in 5 racks. The readout should be organized similar to the CROS readout. There is no buffering possible, so the events have to be read out immediately.

3.13 Proportional Wire Chambers (DPWC) (CROS)

The 4736 channels are connected to 152 modules in 9 crates. For the discussion of the readout, see above.

3.14 RICH (CROS and ADCs)

The discriminated signal from 2848 phototubes of the RICH detector are connected to 89 CROS modules in 5 crates. For the discussion of the readout, see above. The analog signals of some channels (40?) will be fed to one 1881 Fastbus ADC, sitting together with the leadglass ADCs.

3.15 Hadron Calorimeter

We will use the existing E715 hadron calorimeter. It has 17 wire chambers with 48 wires each, and 50 scintillators. The chambers are read out with latches (816 channels in Camac or CROS). For the scintillators, we need 50 ADC channels.

3.16 Miscellaneous (CAMAC)

All additional things, like scalers to measure the differential beam rate, a scaler to measure the time, latches with trigger bits, and other standard modules we can place here. If the readout will be too slow, we have to chain the crates with DYC+s as buffer modules.

3.17 Summary of Frontends

In table 8 we show the summary for the frontend modules used for each detector, together with it's (guessed) readout size. In table 9 we show the actual location of the detectors and the associated electronics in PC4. The zero point of the coordinate system is the exit of the hyperon magnet.

Detector	System	Module	# Modules	Event Size [Bytes]	Conversion Time [μsec]	Readout Speed [μsec]
SVX	Fastbus	FSDA	20	1200	25	< 100
DC	Fastbus	10C6	130	2000?	7	25
Leadglass	Fastbus	1881	35	800	12	80
TRD	CROS	Latches	96	48	0	???
(L)PWC	RMH	Latches	325	296	0	???
DPWC	CROS	Latches	155	200	0	???
RICH	CROS	Latches	89	200	0	???
RICH	Fastbus	1881	1	160	16	???
Had. Cal	???	???	???	???	???	???
Misc.	CAMAC	Scalers	< 20	< 100	0	< 30

Table 8: Summary of Frontend Electronic

3.18 Summary of streams

We can combine ("chain") an arbitrary number readout controllers in one stream. In pratice, the maximum data rate has to be < 22 MByte/s, the data have to fit into the DPMs, and the chained readout controllers should be not too far apart, since we have to pass a readout token.

Table 10 contains a possible distribution of readout controllers to streams. The event format (see later) will not depend on this distribution, so we can easely adapt it to the needs once the beam is on.

3.19 The Connection to the "Schoolhouse"

All the described system are writing data on the RS485/DART cables, flat twisted pair cables, one with 50, the other with 34 pins carrying differential TTL signals. The pin specification for the cables can be found in Table 11.

For E781, the length of the cables will be about 100 m, which is too long to use a normal cable. The layout is shown in fig. 2. For most of the distance we use a glass fiber, with converters from RS485 to fiber and back on both ends. The fiber link itself is unidirectional and can operate at a maximum speed of 120 MByte/s, which is much faster than the speed of the RS485 cable (40 MByte/s). The converter from TTL to fiber and back is a commercial product[1], where the RS485 (differential TTL) to TTL converter and back is build at FNAL. Both converters are put together in one NIM module. Most of the data flow in one direction, but there is one signal (on pins 37/38), the WAIT signal, which is issued by a nearly full FIFO on the DC2, which has to be sent back to the data source. To do this, all WAIT signals from the 10 streams are collected together and an additional fiber link is operated in the other direction. A fiber optic patchpanel in the Schoolhouse and in PC4 (close to the start of the RICH) are used to collect and distribute the signals.

In the Schoolhouse, the data are received by a DC2 with the cable connector DM115[2]. The DC2 has a $4\,k * 32$ bit wide FIFO and pushes the data with a 68380 DMA controller with a maximum speed of 22 MByte/s (12 MByte/s with simultanious VME access) over VSB into VME/VSB Dual Ported Memories.

[1] Finisar
[2] Access Dynamic, Almogordo, New Mexico

#	Detector Name	Cable Length(cm)	Detector Z(cm)	ReadOut Z(cm)	Folding Point Z(cm)	Chans /Cbls	Cable Type	Rack# Type
5	BSSD	1482	318	500	0	8192/ 6	Flt.Rbbn	l3-1
7	VSSDX1	1391	409	500	0	48128/ 31	Flt.Rbbn	l3-1
8	TR SSD	1312	488	500	0	8192/ 6	Flt.Rbbn	l3-1
9	LASD	1464	664	500	0	6144/ 4	Flt.Rbbn	l3-1
10	PWC1	8100	735	96	-2984	2560/ 80	Mlt.Cndt	rmh1
11	PWC2	8100	805	292	-2851	2560/ 80	Mlt.Cndt	rmh2
12	PWC3	8100	875	356	-2784	2560/ 80	Mlt.Cndt	rmh3
13	DC	7054	804	1414	3986	344/ 22	Flt.Rbbn	l3-3
14	PHOT-1	6600	1032	922	3663	630/ 630	PK50	l3-2
15	LASD	1842	1042	500	0	10752/ 7	Flt.Rbbn	l3-1
16	DSSD	2028	1228	500	0	10752/ 7	Flt.Rbbn	l3-1
18	DPWC	1647	1420	1767	0	4736/ 296	Flt.Rbbn	l3-6
19	DDC	1500+4987	1815	1414	3593	1040/ 65	Flt.Rbbn	l3-3
21	TRD	1615	1580	1895	0	3072/ 192	Flt.Rbbn	l3-8
22	RICH	1426	2010	2136	0	2848/ 178	Flt.Rbbn	l3-9
23	VEEA	1500+4156	3205	1478	4472	1300/ 82	Flt.Rbbn	l3-4
24	VEEB	1500+3772	3807	1478	4280	1300/ 82	Flt.Rbbn	l3-4
25	PHOT-2	4946	3980	922	4274	726/ 726	RG58	l3-2
26	VEEC	1500+3490	4410	1606	4203	1300/ 82	Flt.Rbbn	l3-5
27	LPWC	8100	4919	4152	7635	2624/ 82	Mlt.Cndt	rmh4
28	PHOT-3	6600	5208	922	5415	328/ 328	PK50	l3-2
29	PHOT-4	6600	5400	922	5511	328/ 328	PK50	l3-2
30	NCAL	6843	5865	922	0	500/ 500	RG58	l3-2

Table 9: Location of detectors and associated electronics

The firmware running on the DC2 writes the data in ascending order into the memories and fills in a header with the event number, an inclusive byte count of the data received, a spillnumber, and a stream id. A event directory with pointers to the events is also located in the DPMs. At the start of a spill, all pointers are reset to zero.

3.20 Raw Data Format

3.20.1 General Structure

We[3] present a few tables to describe the raw data format. The general concept is to subdivide the event record into blocks. These blocks contain either raw data, or possible calculated quantities about the event; also Monte Carlo blocks are possible. The first word of each block contains a block id in bits 16-31, and an inclusive byte count in bits 0-15. The scheme is shown pictorially in table 13.

The very first word of the event record is the inclusive byte count for the entire event.

The first block (block id 0) is the event header. It contains some important event descriptors in fixed positions at the beginning of the event record. This block can grow in length arbitrarily, but new information can only be appended.

Block ids 1-31 are reserved for raw data streams. Each stream contains data from one or more readout controllers. The general format for these subblocks is briefly described in the next section. More detailed descriptions of the data format from the individual frontend modules can be found in earlier sections.

[3] We acknowledge the contributions of Linda Stutte to this section

Number	Detector	Readout Controller	Event Size [Bytes]	Data Rate [MBytes/s]	Total Size [MBytes]	Comment
1	Camac	SCC/DYC+	60	0	8	
2	Silicon	2 FSCCs	1200	9	170	
3	Photon 1&2	FSCC	600	6	84	10 % Occup
4	Photon 3&4	FSCC	420	4	60	10 % Occup
5,6,7	DCs	8 FSCCs	2560	19	360	15 % Occup
8	TRD/RICH	CROS/DYC+	250	5	105	
	DPWC	CROS/DYC+	200			
	LPWC	2 RMH/DYC+	300			

Table 10: Summary of Streams

Pin Pair	Cable 1		Cable 2	
1 / 2	data(0)	-data(0)	data(16)	-data(16)
2 / 3	data(1)	-data(1)	data(17)	-data(17)
5 / 6	data(2)	-data(2)	data(18)	-data(18)
......	......	......	....	....
29 / 30	data(14)	-data(14)	data(30)	-data(30)
31 / 32	data(15)	-data(15)	data(31)	-data(31)
33 / 34	dstrobe	-dstrobe		
35 / 36	astrobe	-astrobe		
37 / 38	WAIT	-WAIT		
39 / 40				
41 / 42	EOE	-EOE		
43 / 44	16/32*	-(16/32*)		
45 / 46				
47 / 48				
49 / 50				

Table 11: Pin Assignment on RS485/DART cables

Further block id assignments will be allocated as needed.

3.20.2 Readout Controller Headers

Every readout controller (FSCC, DYC+, CROS, RMH, SCC) will produce a 2 word header associated with the data read out. The first word contains the length of the transmitted data in some unit and a 4 bit event number which will assure synchronization. The second word contains a readout controller id and for the FSCC the spill number, otherwise 0.

These readout controller ids will be assigned once and forever to facilitate later unpacking of the event. Table 14 contains the assignments.

Bits	31	30	29	28 24	23 0
	0	0	0	Stream Id	Byte Count

Table 12: DC2 Header word

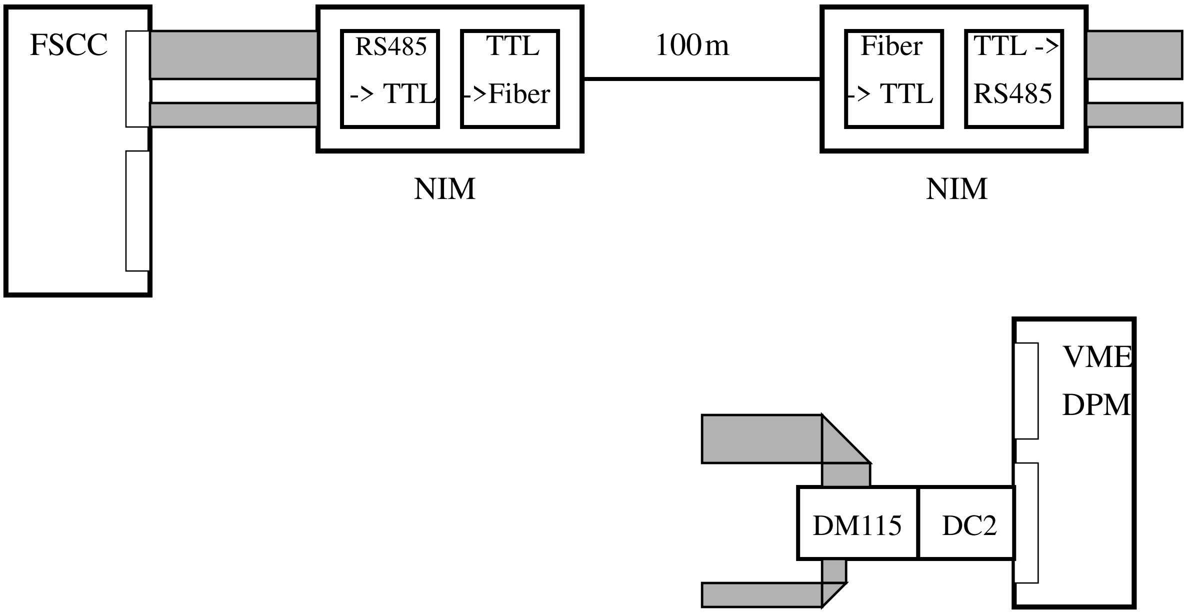

Figure 2: The RS485/DART connection between PC4 and the Schoolhouse

Bits	31 16	15 12	11 0	
	Event Length			Inclusive Byte Count
	0	Eventheader Length		Inclusive Byte Count
	Run Number			increasing from 10000
	Spill Number in Run			Reset at begin of run
	Event Number in Run			Reset at begin of run
	Trigger Number in Spill			Reset at begin of spill
	Trigger Mask			
	Trigger Mask			
	Trigger Mask			
	Trigger Mask			
	Trigger Mask			
	Trigger Mask			
	Spare			
	Spare			
	Spare			
	Spare			
	Stream Id	Length		Block Header for Streams (id=1,..,31)
		Evt#	Length	FSCC or DYC+ Header
	Readout Controller Id	Spill Number		Spill number only from FSCCs
	Data from this Readout Controller			See individual tables
	. .			
		Evt#	Length	FSCC or DYC+ Header
	Readout Controller Id	Spill Number		Spill number only from FSCCs
	Data from this Readout Controller			See individual tables
	. .			
	. .			
	Stream Id	Length		Block Header for Streams (id=1,..,31)
	. .			

Table 13: Event Format

Detector	Crate	Readout Controller	Identification	Readout Planes
Misc.	CAMAC	SCC/DYC+	0x5331 = 'M1'	Scalers and Latches
	CAMAC	SCC/DYC+	0x5332 = 'M2'	Scalers and Latches
Silicon	FSDA 1	FSCC	0x4631 = 'F1'	
	FSDA 2	FSCC	0x4632 = 'F2'	
Photon	ADC 1	FSCC	0x4131 = 'A1'	Photon 1,2
	ADC 2	FSCC	0x4132 = 'A2'	Photon 3,4
Drift Chambers	TDC 1	FSCC	0x5431 = 'T1'	
	TDC 2	FSCC	0x5432 = 'T2'	
	TDC 3	FSCC	0x5433 = 'T3'	
	TDC 4	FSCC	0x5434 = 'T3'	
	TDC 5	FSCC	0x5435 = 'T5'	
	TDC 6	FSCC	0x5436 = 'T6'	
	TDC 7	FSCC	0x5437 = 'T7'	
TRD, RICH	CROS	CSB.V/DYC+	0x4331 = 'C1'	TRD, RICH
DPWC	CROS	CSB.V/DYC+	0x4332 = 'C2'	DPWC
LPWC	RMH	RMH/DYC+	0x5231 = 'R1'	
	RMH	RMH/DYC+	0x5232 = 'R2'	

Table 14: Assigned Identification for each Readout Controller

4 Event Building and Filtering

4.1 Introduction

In the Schoolhouse, all data arrives over the fibers and RS485 links into VME/VSB Dual Ported Memories (DPM) by way of DM115/DC2 VSB interfaces. In one spill, we collect about 700 MBytes of data. The VME crate has therefor about ten DPM modules with 64 MByte or 128 MByte memory. The VME crate holds in addition a MVME167[4] with a 68040 processor running the VxWorks[5] operating system to control the VME crate (see below), a RIMFIRE 3500 SCSI controller[6], a VMETRO[7] display module, a VME Output Register to control the DM115 and the DC2s, and a VME to VME crate interconnect. A layout of the crate is given in Fig. 15. At least five slots in the VME crate are still free.

M	V	V	R		D	D	D	D	D	D	D	D	D	D	D
V	M	M	i		P	P	P	P	P	P	P	P	P	P	P
M	e	E	m		M	M	M	M	M	M	M	M	M	M	M
E	t	t	f												
1	r	o	i		1	6	6	6	6	6	6	6	6	6	6
6	o	V	r		2	4	4	4	4	4	4	4	4	4	4
7		M	e		8										
		E													
					S	S	S	S	S	S	S	S	S	S	S
					1	1	2	3	4	5	6	7	8	9	A

Table 15: Layout of the VME crate

The pointers to the event fragments are generated by the DC2 and are kept in the DPMs. The Filter machine, a twelve processor SGI Challenge L[8], will copy part of the event used in the level 3 filter or the whole event and make the filter decision. The logging will either be performed via the Rimfire SCSI controller, controlled by the MVME167, or from the filter machine.

4.2 DC2 / DM115

The DM115 receives the data coming over the RS485/DART cables (see table 11) and pushes the data in the DC2, which has a $4\,\mathrm{K} * 32$ bit FIFO. The heart of the DC2 is a Motorola 68340 DMA controller, which writes the data from a buffer FIFO over VSB into the Dual Ported Memories sitting in the VME Crate. Both modules are plugged in from the back of the VME backplane. The maximum data rate with the DMAC is 22 MByte/sec, which is reduced to ≥ 12 MByte/sec when the DPM is accessed from VME at the same time.

4.3 The Dual Ported Memories

We will use MMI-6390D[9] DPMs with 64 and 128 MByte and VME D64 capability. The measured maximum transfer rate from and to the memory with a MVME167 as controller is 40 MByte/s for D64 block transfers. The data from a whole spill will be stored in these memories. The lower 256 Bytes are reserved for the handshake between the MVME167 and the DC2, the following nearly 1 MByte contain the pointers to the event fragments. All the rest will be filled up linearly. If one memory gets filled up, a Buffer Almost Full

[4] Motorola
[5] Wind River System
[6] Ciproco Inc., Plymouth, MN
[7] ??
[8] Silicon Graphics, Mountain View, California
[9] Micro Memory

(BAF) signal is given out by the DC2. This signal will be used to inhibit further triggers and a error message will be issued. After the processing of all events from one spill is finished, the pointers are reset.

4.4 The MVME167

The heart of this module is a Motorola 68040 processor, running the VxWorks Real Time Operating System. It contains 32 MByte of memory, a DMA controller for fast data transfer, and a SCSI controller (not used by us). The module can be accessed over Ethernet or via a terminal port. A 8 bit parallel input/output register can be used to give signals to the processor, which will be used in our case for the BAF from the DC2 and the Spill Gate.

Depending if we log from VME or from the Filter machine, the MVME167 will perform several tasks. In any case events will be distributed from here to monitor tasks running on backend workstation over Ethernet (HOISTing). Also some bookkeeping concerning the use of the DPMs will be done here. If we log from VME, a logger process is organizing and controlling the logging of the data via the RIMFIRE controller to the Exabyte. All the processes are connected via DFM queues.

4.5 The Connection to the Filter Processors

About 25 % of the data from charm events have to be transported to the filter processors; this translates to a speed of about 3 MByte/s. The SGI Challenge has an internal VME crate. We can use a VME to VME Crate Interconnect. Two types are currently under investigation: BIT3 Model 413[10] and PT-VME940[11]. Both were successfully operated on a Challenge. The maximum speed is 25 MByte/s with an overhead of 250 μsec for the BIT3 and 40 MByte/s with an overhead of 1 msec for the PT respectively. UNIX device drivers for both interfaces exist and support the use of DFM on the filter machine.

4.6 The Filter Processors and Processes

To filter 140000 events in 60 s we need about 1000 MIPS of computing power (500 MIP ms/event). The most convenient way is to have this in one box with only one connection to the central VME crate. We plan to use a SGI Challenge with twelve processors. A first machine with two processors, which will be used for tests and as the central server, is installed since December '93 (fn781a). The real filter machine will be purchased in FY'95.

The filter data will be transported via the VME to VME Crate Interconnect to this machine. The pointers to the event fragments will be provided by the DC2, the transport will be done by the Challenge. Due to the overhead for a DMA transfer, we have to transfer blocks of event fragments (about 1000 events in a 1 MByte block). The transfer process on the Challenge will distribute the event block to the filter processes via DFM. The results are collected later and either send to a logging process or to the MVME167. Details of the procedure are currently being worked out.

[10] Bit3 Computer Cooperation, Minneapolis, Minnesota
[11] Performance Technology Inc

5 Monitoring

Any computer with an Ethernet Interface and TCP/IP software can be used to monitor the experiment. The MVME167 will distribute events to these computers. The user can request events of a certain type (e.g. trigger mask); to request ALL events of a special type, the program has to run under the control of the data acquisition, since this request can slow down or kill the whole data acquisition.

The most important monitor tasks (direct monitoring of the hardware, e.g. wire maps etc.) will be combined into one program and run on a central machine to assure constant checking of the apparatus.

For the OffSpill checks, the frontends will write results files to the central server machine. Programs running on other computers can analyze these data via reading the files. The data acquisition group will provide the read and write routines.

6 Message Reporting

We will use the MURMUR Message reporting facility. All processes on all computers (with permission) can send messages to a central server process. There the messages are routed according to the corresponding database entries to different screens and/or certain actions (e.g. executing a shell script) are taken. We will define a procedure to enter messages in the central database.

7 Other Literature

Most of these documents arte updated frequently. Please make sure to use always the newest version. You can find the documents in WWW, either in the E781 server (H–Notes) or in the DART server.

H460: The E781 experiment layout, J.Lach, J.Russ

H567: Preliminary work on the E781 trigger system electronics, J.Chilton, V.Smith

H576: E781 Photon Trigger Rate Estimates using E761 data, P. Cooper

H???: FSDA User Manual, S.Stepanov, P.Neoustuev

PN???: The FSCC Users Manual

HN???: The SCC technical description

PN???: The DC2/DM115 specification, O. Trevizo

HN???: The DYC+ Specifications, S. Hansen

IN???: The DC2 software requirements, J.Anderson, O.Trevizo, T. Yamanaka

DS225: The DART Data Flow Manager Design, D.Berg, D. Black, J. Engelfried, D. Slimmer

PN484: The DFM Programmer's Guide, D.Berg, D.Black, J.Engelfried, D.Slimmer

IN346: Using BIT3 on SGI Crimson and Challenge Processors, J.Engelfried

PN482: DFM HOIST, G.Oleynik, R.Pordes, D.Slimmer, J.Streets, A Majewska, C.Guss

PN429: Hoist Programmer's guide, D.Brown, G.Oleynik, R.Pordes, J.Streets, D.Slimmer

PN472: Murmur Quick Reference Guide, C. Moore

PN476: Camac IEEE Library for CES8210 CAMAC Driver, C.Moore, M.Vittone, E.Stern, J.Kim

Specification for CROS and RMH Readout Controllers, J.Engelfried, Oct.12,1993

H665: Timing Tests of the Impact Parameter Trigger on RISC Workstations; M.Procario, J.Russ

H676: E781 Hardware Trigger Preliminary Design; A.Kulyatsev, P.Cooper, A.Kozhevnikov, J.Russ

PN499: An IRIX Device Driver for the BIT3 Crate Interconnect, J.Engelfried

H616: PWC, TRD and RICH channel, RML, CSC crate and racks needs N. Terentyev

H669: Racks and Cable Layout of E781 at PC4, P.Cooper, D.Mao, J.Russ

A short Description of the RMH Readout System for teh E704 MPWCs, A.Bravar, Y.Onel, A.Penzo

H722: A Re-Evaluation of the E781 Charm Trigger, P.Cooper, A.Kulyavtzev, J.Russ

H728: Proposal for E781 Raw Data Format, J.Engelfried, L.Stutte