

A High Performance Distributed
Data Acquisition System
for the NA48 Experiment
on CP-Violation

Nicholas Ewen Mckay



Submitted for the degree of Doctor of Philosophy
The University of Edinburgh

1995

Abstract

This thesis describes the data acquisition methods employed by the NA48 experiment at CERN. A brief overview of the physics behind NA48 and a description of the experimental apparatus are given in the introduction. The specification and the implementation of the data acquisition system is then detailed along with a discussion of the design ‘philosophy’ that influenced the design choices. The section of the data acquisition system on which I have been working, the Data Merger, is discussed in some detail. Next, my main contribution to NA48, the Input Buffer, is described along with its performance in laboratory tests and data taking runs. Alternative solutions to the problems of acquiring data are also discussed along with examples of data taking schemes from similar experiments. Finally, the conclusions that can be drawn from the design and performance of the NA48 data acquisition system are discussed.

Declaration

This work represents the efforts of many members of the NA48 collaboration at CERN, the European Center for Particle Physics. I have been an integral part of the small team of five people who have designed and developed the data merger, and have been personally responsible for the Input Buffer. The writing of the thesis has been entirely my own work.

Acknowledgements

I would like to acknowledge the Particle Physics and Astronomy Research Council for their financial support during my work at Edinburgh.

My main supervisor, Dr Ken Peach, has been a great source of help and encouragement over the last four years. I would like to thank him, Alan Walker and David Candlin for their help and support.

The help of research Associates Dr. Owen Boyle and Dr. Elizabeth Veitch has been greatly appreciated. I would particularly like to thank Dr. Boyle for his work on the data merger at CERN.

The Input Buffer board, the major product of my research, was laid out by Andrew Main at Edinburgh University. The late Peter McInnes was a great help during the design stages of the board.

The ready wit of the departments postgraduate students, Bruce Hay, Grahame Oakland and Mark Parsons has done much to make my time at Edinburgh enjoyable.

At CERN Dr. Robert Maclaren has arranged for me to have the use of his laboratory and equipment. I would also like to thank Jean-Pol Matheys for his work on the development of software for the data merger and Phillipe Brodier-Yourstone for his work on the optical links.

Contents

1	Introduction	1
1.1	Overview	1
1.2	Measurement of ϵ'	4
1.2.1	The NA48 Beam	4
1.2.2	The NA48 Detector	7
1.2.3	The NA48 Trigger System	12
1.3	Dataflow Requirements	15
2	NA48 Dataflow	17
2.1	Overview	17
2.2	Dataflow hardware	21
2.2.1	Optical Links	21
2.2.2	Data Merger	24
2.2.3	HIPPI Link	25

2.2.4	The Workstation Farm	27
2.3	Dataflow Control Software	29
2.3.1	The Run Control Program	29
2.3.2	The Workstation Farm Resource Manager	30
2.3.3	Slow Control	30
3	NA48 Data Merger	31
3.1	General arrangement of the Data Merger	31
3.2	Implementation Choices	34
3.2.1	Bus Implementation Choices	34
3.2.2	Circuit Board Implementation Choices	36
3.3	Overview of the Data Transfer Protocol	37
3.4	Input Buffer	38
3.5	R-path backplane	38
3.6	FIFO Output Formatter	40
3.6.1	Overview	40
3.6.2	General Arrangement of the FOF	41
3.6.3	R-path and token interface	43
3.6.4	Additional data generated by the FOF	43
3.6.5	The HIPPI transfer protocol	44

4	Input Buffer Design	47
4.1	Overview	47
4.2	Design Philosophy	48
4.3	Design Timetable	50
4.4	Input Buffer Logic Circuitry	50
4.5	Xilinx Design	52
4.5.1	Xilinx Layout	52
4.5.2	Xilinx Configuration	53
4.5.3	Design Entry	55
4.5.4	Xilinx Fuctionality	55
5	Test Results	60
5.1	Input Buffer Commissioning	60
5.2	Input Buffer Testing	66
5.2.1	Optical Link Data Transfer	67
5.2.2	R-Path Data Transfer	69
5.2.3	Summary	72
6	Alternative Solutions to the Dataflow Problem	74
6.1	Dataflow Architecture	74

6.1.1	Bus-based Architectures	74
6.1.2	Switching Network Based Architectures	76
6.1.3	Ring Based Architectures	78
6.1.4	The NA48 Dataflow Architecture	79
6.2	Dataflow Solutions from Other Experiments	80
6.2.1	The Atlas DAQ Scheme	80
6.2.2	The DART DAQ Scheme	81
7	Conclusions	84
7.1	Dataflow	84
7.1.1	Initial Tests	84
7.1.2	Further Laboratory Tests	86
7.2	Input Buffer	86
7.3	Concluding Remarks	87
A	VME Operation	89
B	Optical Link Interface	93
C	R-Path Interface	97
D	Token Handling	100

E Front Panel LEDs	102
F Xilinx Design	104
F.1 VME Interface	104
F.2 Optical Link and R-Path Memory Control	113
G Simulation Results	120
G.1 Dataflow Simulation	120
G.2 Input Buffer Simulation	124

List of Figures

1.1	Schematic overview of NA48 experiment	5
1.2	Read-out Structure of LKr Calorimeter	10
1.3	The triggers in respect to the dataflow scheme.	13
2.1	Hardware Components of the Data Acquisition System	18
2.2	Layout of Data Merger	25
2.3	Level 3 Workstation Farm Architecture	28
3.1	Photograph of the 6U crate used to house the DMC.	32
3.2	Photograph of the 9U crate that is used to house the Input Buffers and the FOF.	33
3.3	An example of BTL signals on the CERN backplane.	36
3.4	Block diagram of the FOF.	42
3.5	The structure of a FOF HIPPI connection.	45
3.6	The structure of an event packet.	46
4.1	IB board layout	49

4.2	Timescale for Input Buffer Development	51
4.3	Block Diagram of XC4000 Configurable Logic Block	54
4.4	Block Diagram of Xilinx	56
4.5	Generation of Write Enable for even memory	58
5.1	Initial IB Block Diagram	61
5.2	Prototype IB Test Set-up.	62
5.3	Dataflow Hardware Setup for the 1994 Test Run	64
5.4	IB Write-in Cycle from Optical Link	68
5.5	\overline{XOFF} de-asserted and transfer resumed.	68
5.6	Start of transfer of Chequer Board pattern from IB to FOF.	70
5.7	End of transfer from IB to FOF.	70
5.8	RP_HAVEDATA and $\overline{RP_NODATA}$ at the start of a data transfer to an empty IB.	72
6.1	Data Acquisition Structures (a) Bus (b) Crossbar switch (c) Ring.	75
6.2	ATM Switching Architecture	77
6.3	A Multi-Ringlet SCI Data Merger	78
6.4	Block Diagram of the KTeV DA System	82
7.1	A plot from the 1994 run showing a hit in the hodoscope together with a hit in the tagger.	85

A.1	IB VME Interface	90
B.1	Channel Memory Structure	95
C.1	R-Path Interface	98
C.2	R-Path Data Transfer	99
D.1	Token Input/Output Circuit	101
F.1	IB Xilinx Top Level Schematic	105
F.2	Xilinx VME decoding	106
F.3	Xilinx Read/Write strobe multiplexer circuit	109
F.4	Xilinx CSR circuit	110
F.5	Xilinx Counters and Protocol Circuit	111
F.6	Xilinx Memory Address Generation	112
F.7	Xilinx Write Enable Circuit	114
F.8	Generation of Write Enable for even memory	115
F.9	Xilinx token circuit	116
F.10	Even Memory Read Signals	118
G.1	Verilog Schematic of the NA48 Dataflow System	121
G.2	Simulation of the NA48 Dataflow System	123
G.3	Verilog Schematic of the Dataflow with an Expanded IB	125

G.4 Input Buffer write-in Cycle from Optical Link.	126
G.5 Input Buffer read-out Cycle over R-Path.	127

List of Tables

6.1	Critical ATLAS front end DAQ parameters	80
A.1	VME Addressing Scheme	91
A.2	Input Buffer Control and Status Registers	91
B.1	OL protocol signals	94
E.1	IB Status LEDs	103
F.1	Channel 0 VME Decoding Scheme	107
F.2	Memory Addressing Scheme Implemented in the Xilinx FPGA	118
G.1	Simulation Trigger Rates	122

Chapter 1

Introduction

1.1 Overview

The NA48¹ experiment at CERN has been designed to measure the magnitude of direct CP violation to an absolute accuracy greater than 2×10^{-4} [1].

Prior to 1964 it was believed that all weak decays were symmetric under CP, where the CP operator is defined to be the combined operations of charge conjugation and parity transformation. However, in 1964 the occurrence of CP violation was observed in the neutral kaon system [2].

Although produced by the strong interaction, kaons decay via the weak interaction. Therefore, the production and decay eigenstates can be different. The strong eigenstates of neutral kaon production are also eigenstates of the strangeness operator \hat{S} . That is:

$$\hat{S} |K^0\rangle = |K^0\rangle \tag{1.1}$$

¹The NA48 collaboration is the 48th experiment to be approved for construction at the North Area of the CERN SPS. The collaboration is made up from physicists and engineers from Cagliari, Cambridge, CERN, Dubna, Edinburgh, Ferrara, Mainz, Orsay, Perugia, Pisa, Saclay, Seigen, Torino and Vienna.

$$\hat{S} |\overline{K^0}\rangle = -|\overline{K^0}\rangle \quad (1.2)$$

The CP eigenstates of the neutral kaon decay are different:

$$K_1^0 = \frac{1}{\sqrt{2}}(K^0 + \overline{K^0}) \quad (1.3)$$

$$K_2^0 = \frac{1}{\sqrt{2}}(K^0 - \overline{K^0}) \quad (1.4)$$

Where the convention chosen is that the K_0 transforms to $\overline{K^0}$ under the CP operator while $\overline{K^0}$ transforms to K_0 .

K_1^0 is CP-even and decays to either $2\pi^0$ or $\pi^+\pi^-$ while K_2^0 is CP-odd and predominantly decays to $3\pi^0$, $\pi^+\pi^-\pi^0$, $\mu\pi\nu$ or $e\pi\nu$. All of these reactions obey CP symmetry. However, the K_2^0 may also decay to $2\pi^0$ and $\pi^+\pi^-$. These reactions are CP-violating.

Christenson *et al* [2]. deduced from this that the real weak eigenstates are mixtures of K_1^0 and K_2^0 , (assuming CPT) ²;

$$K_S^0 = \frac{1}{\sqrt{1+|\epsilon|^2}}(K_1^0 + \epsilon K_2^0) \quad (1.5)$$

$$K_L^0 = \frac{1}{\sqrt{1+|\epsilon|^2}}(K_2^0 + \epsilon K_1^0) \quad (1.6)$$

where the short-lived K_S^0 is mostly CP-even and the long-lived K_L^0 is mostly CP-odd. The mixing between K_S^0 and K_L^0 allows the long-lived K_L^0 to oscillate into

²The CPT Theorem (the combination of CP and Time symmetry) is implicit in quantum field theories such as the standard model of particle interactions. For example the masses and lifetimes of particle – anti-particle pairs are equal. This has been tested in the neutral kaon system to one part in 10^{18} .

K_S^0 and then to decay into a CP-even state. This is termed ‘Indirect’ CP-violation and the constant ϵ provides a measure of its magnitude.

The ratio of CP-symmetric to CP-violating decays is described by η_{+-} and η_{00} , where Γ is the partial width of the given decay:

$$|\eta_{+-}|^2 = \frac{\Gamma(K_L \rightarrow \pi^+\pi^-)}{\Gamma(K_S \rightarrow \pi^+\pi^-)} \quad (1.7)$$

$$|\eta_{00}|^2 = \frac{\Gamma(K_L \rightarrow 2\pi^0)}{\Gamma(K_S \rightarrow 2\pi^0)} \quad (1.8)$$

In the case of indirect CP-violation the 2π decays of the K_L arise from the K_1 component, and therefore $\eta_{+-} = \eta_{00}$.

However, it may also be possible for the K_2^0 state to decay directly to a CP-even state. If this were to occur there could be different amounts of CP-violation in the $2\pi^0$ and $\pi^+\pi^-$ decay channels. This is termed ‘direct’ CP-violation and is described by the double ratio of η_{+-} and η_{00} :

$$R = \left| \frac{\eta_{00}}{\eta_{+-}} \right|^2 = \frac{\Gamma(K_L \rightarrow 2\pi^0)/\Gamma(K_L \rightarrow \pi^+\pi^-)}{\Gamma(K_S \rightarrow 2\pi^0)/\Gamma(K_S \rightarrow \pi^+\pi^-)} \approx 1 - \frac{1}{6} \text{Re} \left(\frac{\epsilon'}{\epsilon} \right) \quad (1.9)$$

The parameter ϵ' is zero if only indirect CP-violation takes place in the neutral kaon system but in general is non-zero if direct CP-violation is present. It is this parameter that NA48 aims to measure. Previous measurements made by NA31 at CERN [3] and E731 at Fermilab [4] have yielded different results. The published result of NA31 of $\text{Re}(\epsilon'/\epsilon) = (2.0 \pm 0.7) \times 10^{-3}$ is three standard deviations away from zero, while the result from E731 ($\text{Re}(\epsilon'/\epsilon) = (0.74 \pm 0.59) \times 10^{-3}$) is consistent with zero. NA48 hopes that by increasing the accuracy of the measurement by a factor of 5 the value of ϵ' can be determined.

1.2 Measurement of ϵ'

In order to measure the double ratio R given in Equation 1.9, the interesting events must be sorted out from a large background of decays that are unimportant. The decays that are of interest are:

$$\begin{aligned}K_L &\rightarrow 2\pi^0 \\K_L &\rightarrow \pi^+\pi^- \\K_S &\rightarrow 2\pi^0 \\K_S &\rightarrow \pi^+\pi^-\end{aligned}$$

To identify these decay modes we must first be able to differentiate between K_S and K_L . In NA48 this is done by generating two beams, one of long-lived kaons (K_L) and one of short-lived kaons (K_S).

1.2.1 The NA48 Beam

Figure 1.1 shows a schematic overview of the beam line and the detector the major parts of which are detailed below:

The kaon beams are formed by directing a proton beam onto a beryllium target. The charged particles in the resulting shower are swept away from the beamline by magnets so that only neutral particles (kaons, neutrons, photons, neutrinos and lambdas) are present at the output.

The particles move along the beam pipe which passes through the detector. Only the particles which decay while in the beam line are detected. Of the neutral particles only kaons produce decays which are observed by the detector. Neutrons, gammas and neutrinos are all stable in the beam line and so will not be seen. Lambdas will decay into a proton and a charged pion, which decays almost instantly into two photons.

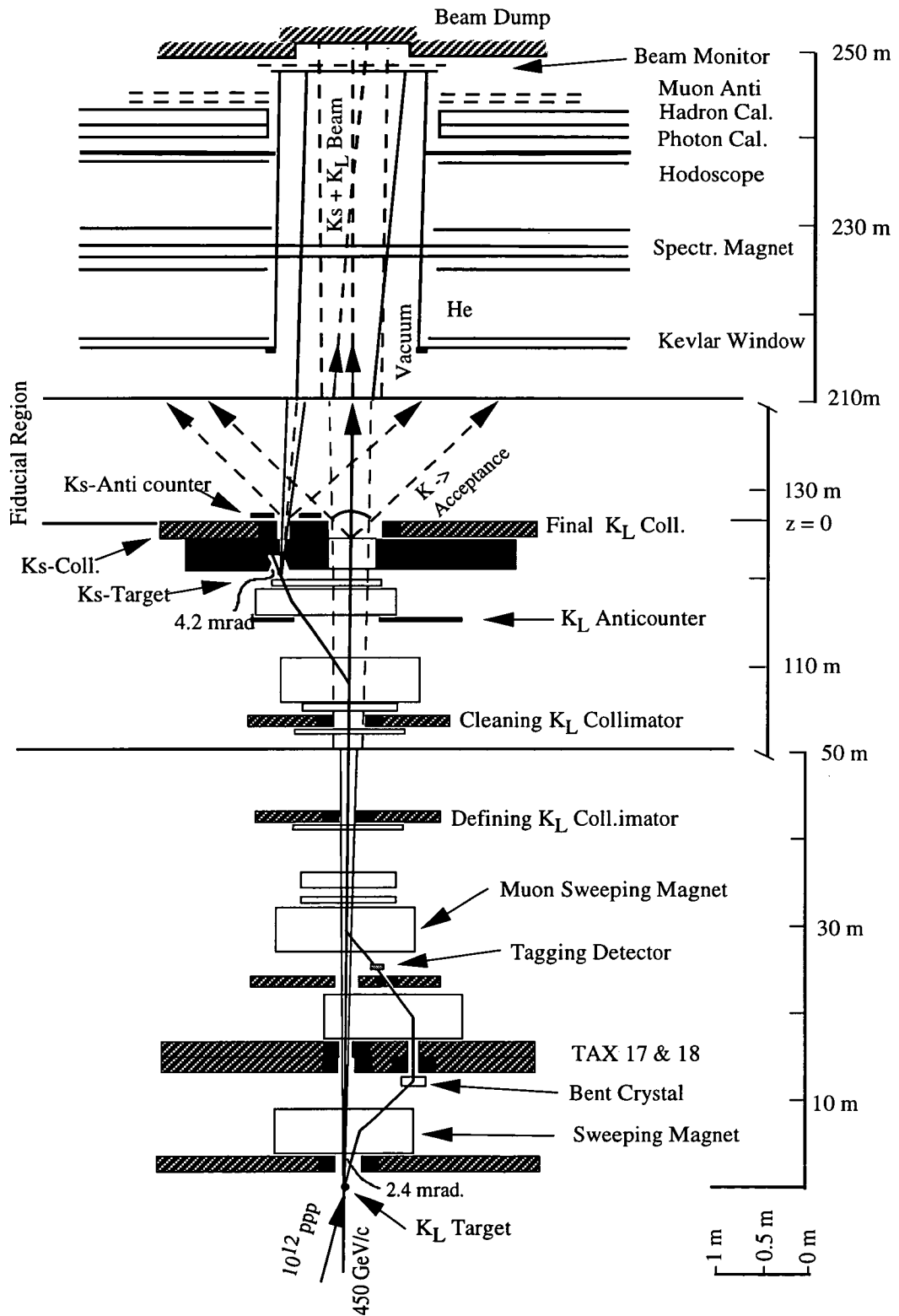


Figure 1.1: Schematic overview of NA48 experiment

The K_L Target

The K_L target is 240 m away from the detector. After the protons hit the beryllium the charged particles at the output are swept away by a magnet and the kaon beam collimated to produce a narrow beam. When the beam reaches the beginning of the decay region some 130 m away it will consist of only K_L particles as the K_S decay length in the laboratory frame at 100 GeV is only 5 m. The proton beam for the K_S is formed by deflecting some of the non-interacting protons away from the main beam and directing them through a bent crystal onto a second target only 120 m away from the detector.

The Proton Tagging Counter

To identify an event as being a K_S the protons from the second beam are tagged upstream from the target. If an event is seen to be accompanied by output from the tagger it is declared a K_S event, while if the tagger does not fire the event is labelled a K_L .

The tagger measures the time of flight between the tagging of a proton and the hodoscope [5]. Events arriving at the hodoscope in coincidence with a hit in the tagger within a certain time window are identified as K_S . The tagger consists of two sets of twelve staggered scintillation counters arranged alternately in horizontal and vertical orientation. The light from the plastic scintillator is read out through PhotoMultiplier tubes to two trigger counters. A system of Analogue-to-Digital Converters, sampling every 1.1 ns, is used to digitize the analogue pulses from the counters. The system provides a high rate capability (3×10^7 protons per 2.7 second spill), a time resolution of under 500ps and a high efficiency. After tagging the protons are deviated back towards the K_L line and sent to the K_S target.

The K_S Target

The protons are deflected from the K_L line by a magnet 110 m downstream from the first target and hit the second (the K_S target) at $z=0$ m. Before this the proton beam is deflected by 4.2 mrad by a sweeping magnet so that the K_S beam intersects with the K_L beam at the subdetectors. This beam will consist of both

K_S and K_L at the detector but in this short range K_L decays are unlikely.

1.2.2 The NA48 Detector

The detector has four main functions:

Firstly, to reduce the volume of uninteresting data that is sent to the data acquisition system, the detector must readout only on events of interest. The subdetectors give information to the trigger system (detailed in Section 1.2.3) on what particles they have detected during an event. A trigger is only sent to the readout electronics when a potentially useful decay is observed. The second function of the detector is to allow the reconstruction of the particle parameters, i.e. position, lifetime, energy and momentum. Thirdly the types of particle (i.e. electrons, pions, muons) in the decay must be able to be identified and lastly incomplete tracks and single tracks (single muons for example) must be rejected.

The detector is made up of several subdetectors, the most important of which are:

Liquid Krypton (LKr) Calorimeter Gives information on the energy and position of photons (γ), positrons (e^+) and electrons (e^-).

Hodoscope Reports on the x,y position of charged particles.

Hadron Calorimeter Reports on the energy and position of charged pions (π^+ , π^-).

Muon Veto Detects muons (μ).

Magnetic Spectrometer Enables the calculation of the momentum of charged particles.

The events that are of interest to us produce 2 pions. $K_L \rightarrow 2\pi^0$ and $K_S \rightarrow 2\pi^0$ generate 4 photons which are detected in the LKr Calorimeter. The $K_L \rightarrow \pi^+\pi^-$ and $K_S \rightarrow \pi^+\pi^-$ reactions produce two hits in the Hodoscope.

As well as detecting these reactions we must also identify the other ‘un-interesting’ decays in order to suppress them:

$K_L \rightarrow \mu\pi\nu$ The μ in this reaction deposits a little energy in the LKr Calorimeter. The μ however will also trigger the Muon Veto thus preventing this decay from being taken as a two pion event.

$K_L \rightarrow e\pi\nu$ This gives two hits in the Hodoscope but also a shower caused by the electron in the LKr Calorimeter.

$K_L \rightarrow 3\pi^0$ This produces 6 photon showers in the LKr Calorimeter.

$K_L \rightarrow \pi^+\pi^-\pi^0$ Two hits appear in the Hodoscope but 2 photons are also seen in the LKr Calorimeter

The Anticounters

These surround the beamline and detect particles (usually photons) which are outside the acceptance of the magnetic spectrometer and the LKr calorimeter. They each consist of several planes of scintillator, material which produces light when hit by most particles. The light energy from each scintillator is amplified by a photomultiplier tube before being sent to a Flash ADC, where the light is converted to electrical pulses. The output from the FADCs is used by the trigger logic to veto decays which are incomplete due to particles being outside the detector acceptance.

The K_S anticounter is situated in the K_S beam and has a photon converter in front of it to veto decays that occur upstream from the anticounter. This is used mainly to set the absolute length and energy scales of the whole detector.

The Magnetic Spectrometer

This subdetector consists of a large (2.4 m) dipole magnet and four high precision, high rate drift chambers [6]. The first two drift chambers are in front of the magnet. These give the vector of a particle prior to it being subject to the magnetic field (which corresponds to a kick of around 250 MeV/c transverse momentum). The second pair of drift chambers give the vector after the particle has been deflected by the field. Knowing the two vectors and knowledge of the magnetic field strength the momentum of the particle can be deduced.

Each drift chamber consists of four gas-filled double planes each with a central hole for the beam pipe. Each of the planes has 256 sense wires spaced on a 1 cm grid. Ionisation of the gas by a particle in the vicinity of these wires produces a voltage drop on the wires which propagates to the read-out wires at each end. From the voltage drop the path of the ionising particle can be deduced. The signals from the sense wires are routed to a TDC system. As only 2 out of 256 wires are hit in a typical $\pi^+\pi^-$ event this system performs zero suppression on the data before sending it for storage in a time stamp-addressed ring buffer. This buffer is then accessed by the Level 2 Charged Trigger electronics (Section 1.2.3).

The Hodoscope

The NA48 hodoscope is a system of scintillation counters which provides the x, y position and, with a high accuracy, the time of arrival at the detector (t_0) of charged pions resulting from kaon decays in the beam pipe. The subdetector is formed by two planes of counters. The first plane contains 64 horizontal counters while the second plane consists of 64 vertical counters [7]. Each of the planes is divided up into four quadrants of 16 counters. The hodoscope has an accuracy of around 10 cm in the horizontal and vertical directions, transverse to the beam.

Scintillation light from each counter is recorded by a Photomultiplier tube. The Hodoscope trigger system reads the outputs of the PMs and produces a trigger (Q_x) when it sees an event which appears to correspond to two charged pions (indicating that a $K_L \rightarrow \pi^+\pi^-$ or a $K_S \rightarrow \pi^+\pi^-$ decay may have taken place). As a charged pion pair is represented in the Hodoscope as four hits on different counters (two on each plane), four independent time measurements are produced. The resolution of the event time, t_0 , defined as being the average of these four timings, is around 0.5 ns.

The Liquid Krypton Calorimeter

The LKr Calorimeter [8] is designed to give excellent space, energy and time resolution as well as coping with very high particle rates. The read-out structure of a LKr Calorimeter quadrant is shown in Figure 1.2.

The read-out cells are formed on parallel ribbons of Kapton, 75 μm thick, clad in

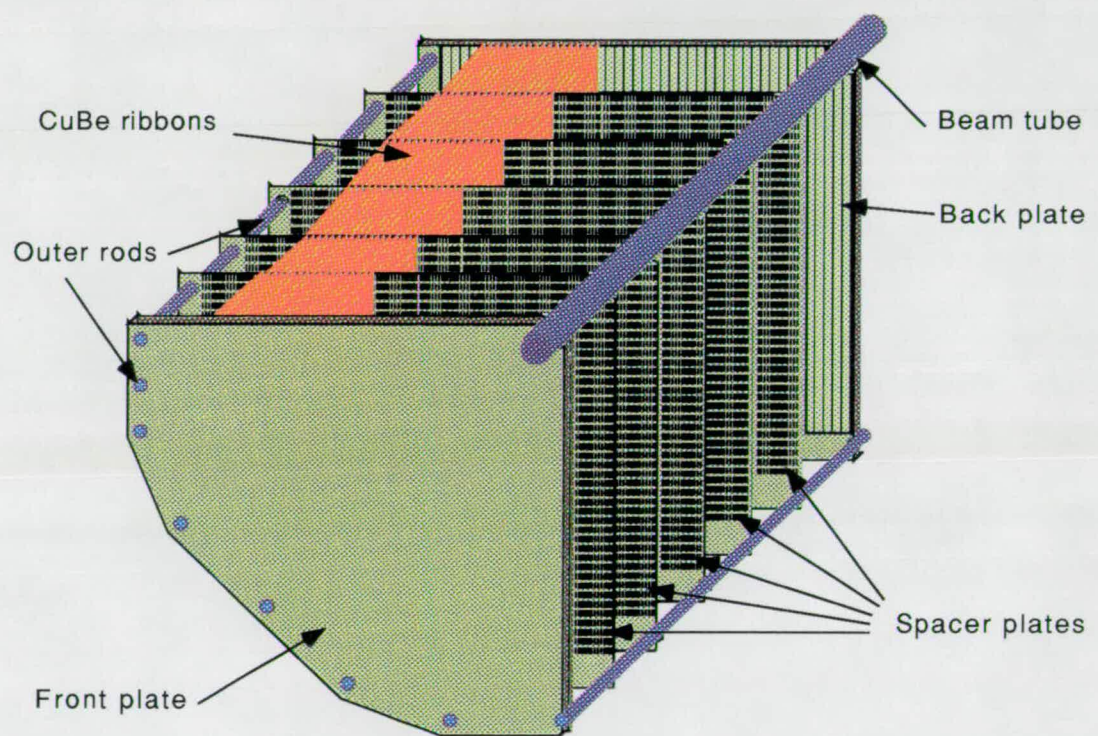


Figure 1.2: Read-out structure of the LKr Calorimeter

copper strips, 19 mm wide and 17 μm thick. The strips are 10 mm apart. The cross-section of the read-out towers is 2 cm \times 2 cm. Altogether there will be about 13500 towers. This structure ensures the high rate capability, good space resolution and photon separation that are needed. The use of liquid krypton in a quasi-homogeneous structure (the only other materials present are thin read-out ribbons) minimizes sampling fluctuations. This is required to achieve the very high energy resolution required.

The photons collide with the Liquid Krypton causing electromagnetic showers which produce ionisation. The electrons liberated by this ionisation are subject to a transverse electric field (the anode ribbon is at +5000 V) and cause an induced electric current to flow along the ribbons as they drift. The current is proportional to the size of the charge. It rises quickly and falls away linearly over 3 μs as the electrons combine at the anode.

The output from the read-out ribbons is put through a shaper which is sensitive to changes of slope in the current waveform. This produces a 100 ns wide pulse which is sent to the read-out electronics.

The Hadron Calorimeter

The hadron calorimeter (HAC) consists of an iron/scintillator sandwich of 1.2 m total iron thickness [9]. It is divided into two sections, each consisting of 24 steel plates, 25 mm thick, of dimensions $2.7 \times 2.7 \text{ m}^2$. Each plane is made up from 44 separate strips of dimensions 1.3 m \times 11.9 cm \times 4.5 mm. Consecutive planes are aligned alternately horizontally and vertically forming a quadrant structure.

The light from the scintillator is fed through photomultiplier tubes and then to a bank of FADCs. There are 176 readout channels. The analogue signals are integrated and the information on the particles energy is fed to the trigger system. The x, y position and the energy of the particle in the HAC is input to the data acquisition system.

The Muon Veto

The Muon Veto [10] consists of three planes of plastic scintillator - the first two

are 1 cm thick and the third is 6 mm thick. The planes are each sited behind 0.8 m of iron which is sufficient to stop most particles except muons from reaching the detector. The first plane consists of eleven horizontal strips, the second of eleven vertical strips. There is a hole in the middle of each plane to enable the beam pipe to pass. These planes output the x,y position of a particle passing through the subdetector. The third plane is used mainly for efficiency calculations. The light from the scintillator is read out to the muon electronics through Photomultiplier tubes attached to each end of the strips.

1.2.3 The NA48 Trigger System

NA48 has three separate levels of triggering. The Level 1 and Level 2 triggers are implemented in hardware. The third level, software filtering of the data, is implemented in the Alpha workstation farm. Figure 1.3 shows the position of the trigger levels in relation to the readout stages [11].

The Neutral Trigger

The neutral trigger [12] monitors the signals coming from the LKr Calorimeter and sends a trigger to the trigger supervisor whenever the criteria for a $2\pi^0$ event have been satisfied. The calorimeter readout is sampled and a trigger decision sent to the level 1 trigger supervisor every 25 ns. The rates are expected to be around 25 $K_L \rightarrow \pi^0\pi^0$, 1500 $K_L \rightarrow 3\pi^0$ and 100 $K_S \rightarrow \pi^0\pi^0$ per 2.7 second spill.

Four quantities are computed for each 25 ns time bin and these are used to make the trigger decision:

- The number of peaks in the LKr Calorimeter. Good events have 4 or fewer peaks (e.g. $K_L \rightarrow \pi^0\pi^0 \rightarrow 4\gamma$ produces 4 peaks). The trigger rejects events with more peaks in order to remove $K_L \rightarrow 3\pi^0 \rightarrow 6\gamma$ background.
- The total energy E . This must be above a certain threshold for the event to be considered good.

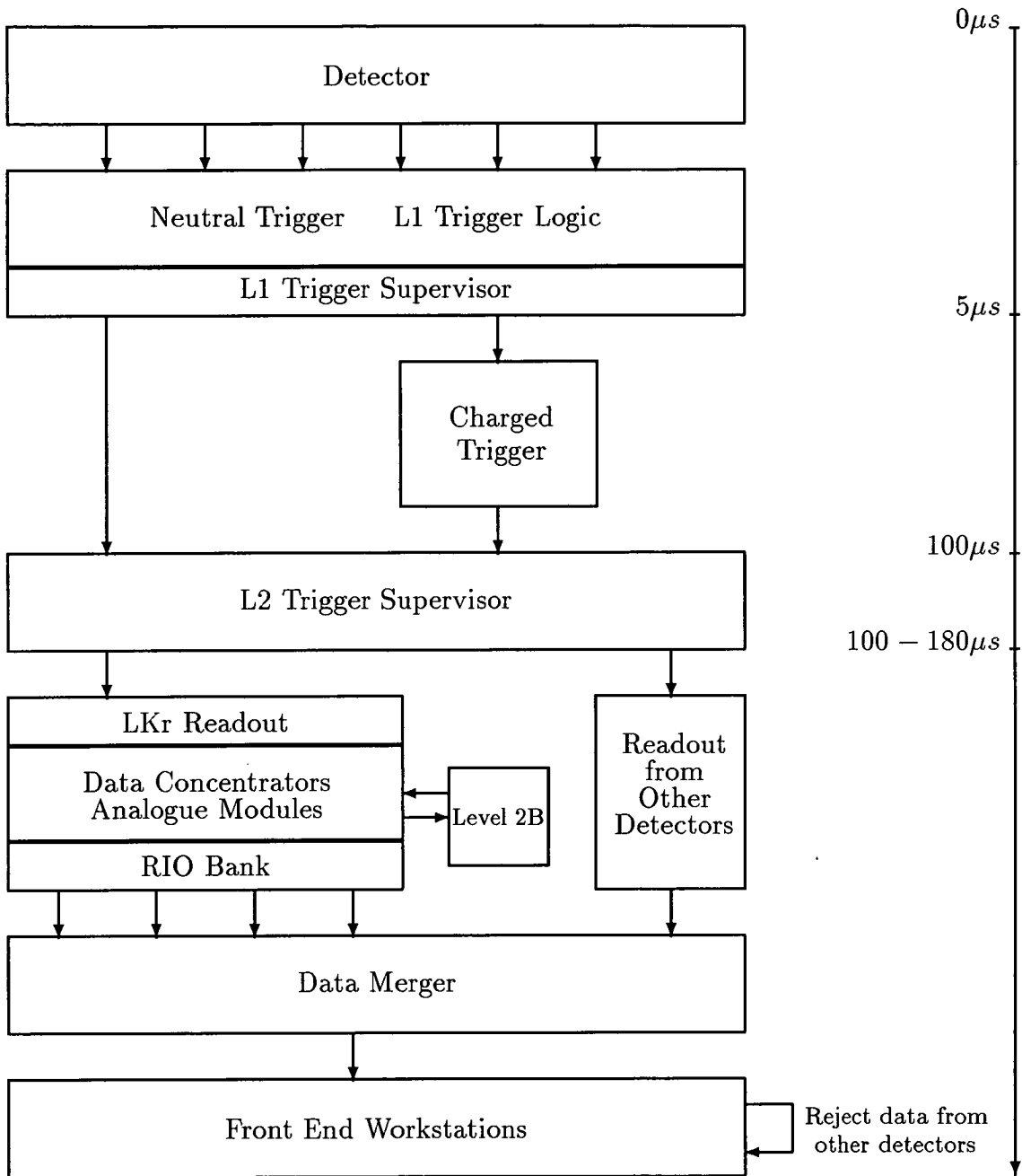


Figure 1.3: The triggers in respect to the dataflow scheme. The timing in relation to the event time is shown on the right hand side.

- The centre of gravity $C = \sqrt{(m_{1x}^2 + m_{1y}^2)}/E$ where $m_{1x} = \sum x_i E_i$ and $m_{1y} = \sum y_j E_j$. Where the sum is over i (strip number in the x projection) and j (strip number in the y projection). C should be close to zero. Single tracks will not pass this cut and therefore will be rejected.
- The proper lifetime of the kaon. This cut is performed to remove the $K_L \rightarrow \pi\mu\nu$, $K_L \rightarrow \pi e\nu$ and $K_L \rightarrow \pi^+\pi^-\pi^0$ decays. The lifetime in these events will usually not be a low enough value to cause a trigger as the neutrino (ν) or the charged pions will not be seen by the subdetector.

The Charged Trigger

The Level 2 Charged trigger [13] identifies the important charged pion events and the background $K_L \rightarrow \pi e\nu$ decays. Input to the trigger is provided from the hodoscope and the magnetic spectrometer. The hodoscope readout electronics produce a trigger pulse each time there is activity in opposite quadrants of the subdetector, the Q_x condition. This indicates that a 2 pion event may have occurred.

For events satisfying the neutral conditions, a fast DSP-based VME processor farm performs a part-reconstruction on the particles detected in the first, second and fourth drift chambers. It then calculates the momenta of all tracks, the vectors for all particle pairs and finally the 2π mass. This takes place in the 100 μ seconds after the decay. The processor farm passes its information on the event to the trigger supervisor.

The Trigger Supervisor

The trigger supervisors give decisions for each 25 ns time slice based on the information sent to them from the sub-detectors [14]. They correlate the trigger data received and dispatch a trigger decision to the read out controllers of each sub-detector. The information at the sub-detectors is digitized and stored in circular memory buffers where it remains for a minimum of 200 μ s. When the trigger

request arrives at the sub-detector the buffers are accessed and the information for the probable 2 pion time slice is sent to the data acquisition system.

The Level 2B Trigger

The Level 2B trigger is included to solve a bottleneck problem at the LKr Calorimeter. This sub-detector provides by far the most data for each event and more rejection power is needed at the Calorimeter in order to prevent the data flow system becoming swamped. The Level 2B trigger works entirely downstream from the rest of the Level 1 and Level 2 triggers. It acts only on triggers issued by the Level 2 trigger supervisor and is highly integrated into the read out of the LKr Calorimeter. When a trigger is received at the Calorimeter data is fetched from a circular memory buffer. As the data is being read out a summary is sent to the Level 2B computer which performs cuts and decides whether an event should be accepted or rejected. If the latter is the case the Calorimeter sends only dummy data to the data acquisition system while the rest of the sub-detectors send full events. The event is finally killed at the Level 3 stage in the workstations.

1.3 Dataflow Requirements

A ‘spill’ is a stream of protons sent to the experiment by the SPS. Each burst lasts for 2.7 seconds and there is one burst every 14.4 seconds. The amount of data sent from the detector during each burst depends on the trigger rate and the size of the events. The rate of events estimated to be processed after the Level 2 trigger is 1.5 kHz from the K_L beam and 0.15 kHz from the K_S . In addition to these there are calibration triggers (0.2 kHz) and ‘downscaled’ events (1 kHz). The latter are used to monitor the trigger.

The event length is dominated by data from the LKr Calorimeter. This depends on the zero suppression algorithm and the data compression algorithm used in the subdetector readout electronics. The largest amount of data is sent during a $2\pi^0$ event. With four photon showers contained within a 15 by 15 array of cells, around

900 cells are read out. For each of these 8-10 time samples are recorded to give the amplitude and timing of the pulses caused by the particle showers. With data compression the volume could be reduced by a factor of 2 or 3. The other detectors generate modest amounts of data for each trigger. The Spectrometer gives around 100-200 bytes, the hodoscope, anti-counters, muon veto, tagger, triggers and the hadron calorimeter contribute roughly 2 kbytes in total. Thus the total data volume per event is between 7 kbytes and 20 kbytes.

These figures are estimates, as at the time of the dataflow specification the detector was not yet built or tested, but they give an indication of the event rate and size. Taking the averages of these figures (say a trigger rate of **3 kHz** and an event size of **10 kbytes**) the data acquisition system has to process 30 MBytes/sec (75 MBytes/burst). However, some overhead must be built in due to the possibility that the trigger is not as efficient as planned or that new trigger categories or low downscaling factors are decided upon.

Therefore it was decided in 1993 to make the specification[15]:

100 MBytes/sec **and** 10 kHz

Which corresponds to 250 MBytes and 25 k events per burst. In the next section the hardware designed to satisfy this specification is discussed.

Chapter 2

NA48 Dataflow

2.1 Overview

The Dataflow scheme is fed data from the read out controllers of each sub-detector. It merges this data into an event packet which contains all the information needed to decipher the kaon decay that has been triggered upon. The data is then input into a farm of workstations and storage units where detailed analysis and filtering can take place.

The Dataflow scheme consists of several related subsystems as shown in Figure 2.1. The components are usually separate but some tasks can be split between different machines.

The principal stages are:

The Front End Electronics Each sub-detector has its own specially commissioned read out electronics. Data collected at most detectors is fed into the dataflow system via a VFIFO module. This takes data from VME and outputs it to an Optical Link Source (OLS) module. Each sub-detector uses the same OLS and VFIFO boards so the rest of the system is the same for each sub-detector.

The Fibre Optic Links These transfer asynchronous data from the subdetector read out crates to the Data Merger across a distance of 200 m.

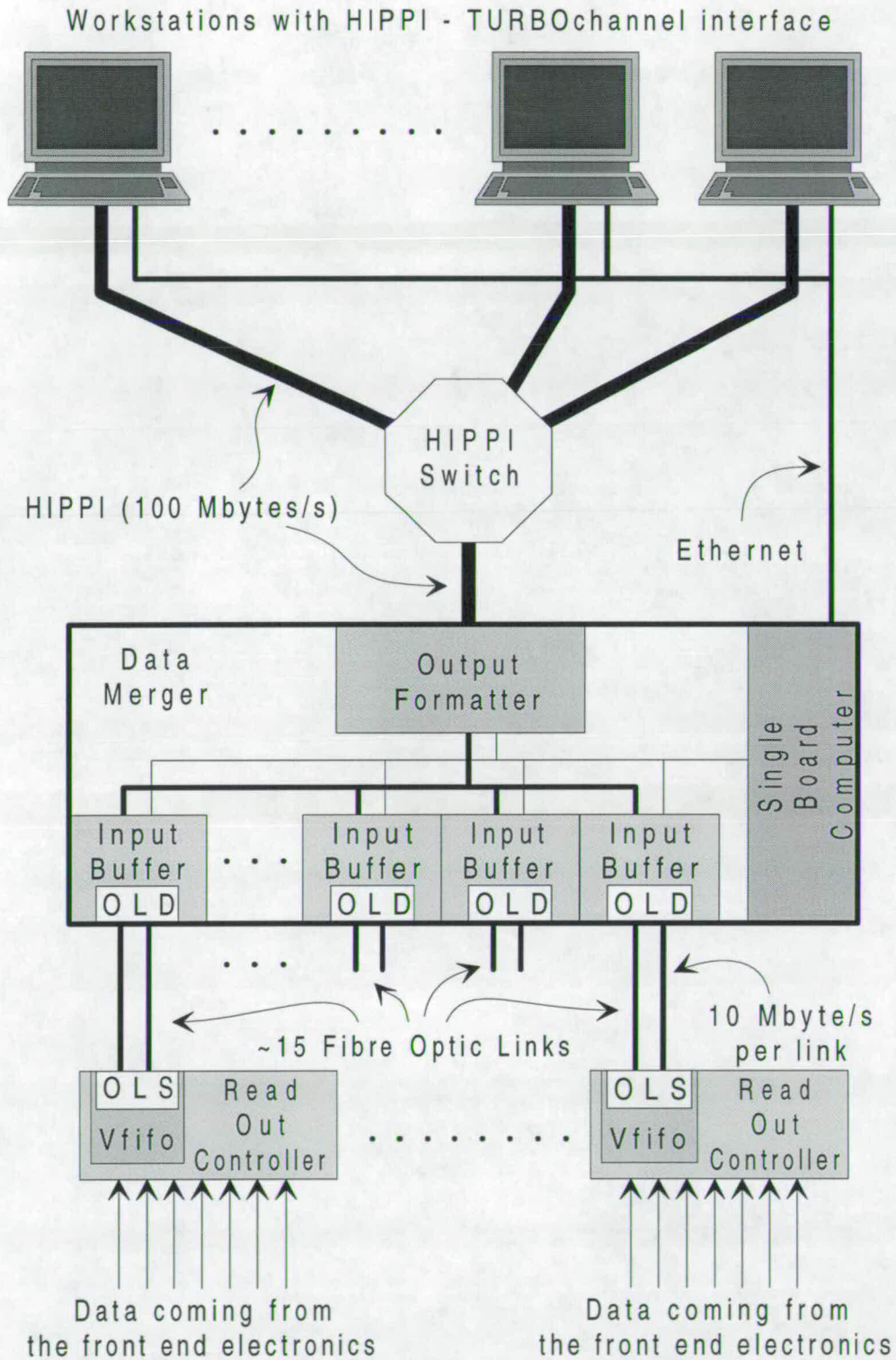


Figure 2.1: Hardware Components of the Data Acquisition System

The Data Merger Each sub-detector outputs a sub-event when a trigger is issued. In the Data Merger the sub-events are buffered and merged together across a backplane to form a full event. The event contains the information from every sub-detector for that trigger. The Data Merger output device formats the event and transmits it to the workstations.

The Data Merger Controller This is a Single Board Computer which sets up and monitors the Data Merger. It also controls the Front End Workstations (FEWS), which receive the data over the HIPPI and Turbochannel links.

The Workstation Farm The workstation farm receives the data from a complete burst. The data is reformatted and then distributed around the farm for Level 3 filtering and eventual recording onto storage media. Various 'special' events containing monitoring and calibration information are sent to special tasks. Software services for the workstation farm are provided by the resource manager.

The Control Program This provides an interface between the dataflow system and the people on shift in the control room. It monitors the condition of the system, reporting errors and responding to requests from the shift crew.

Slow Control This sets and monitors the detector and various components that are not involved directly in the data taking each burst. For example the beam and hall conditions, gas pressures and high and low voltages are set and maintained by the system.

The data acquisition system is also indirectly affected by the following components:

1. *The Common Clock*

The common clock is distributed to all subdetectors as well as the Control Program. This has a base frequency of 40 MHz.

2. *The Trigger Supervisor and the Read Out Controllers*

As summarised in Section 1.2.3 The Level 2 Trigger Supervisor is responsible

for accepting an event as interesting, setting off the read out process. It also distributes an event number to the read out controller of each subdetector. This number is checked by software in the Alpha Farm to ensure the event has been merged correctly. The readout controllers collect and buffer the data from each part of the experiment. When all the data is collected from one event they output it to the Optical Links.

3. The Beam Control System and the SPS timing signals

As well as controlling and monitoring the beam, the Beam Control System provides timing signals which split the spill into two stages. One stage is handled by the central data acquisition system and the other is the responsibility of the local data system at the subdetectors.

4. The Local Subdetector Computer

This communicates with the Control Program to provide local requests for actions. For example the shift crew must warn each Local Subdetector Computer of the need to prepare for a new run.

The design of the system makes it possible to provide higher throughput by increasing the parallelism at the stage which is constricting the speed of data transfer. If the workstations are too slow to process the data then another one may be added to quicken the acquisition rate (The HIPPI switch can be configured for up to 8 output devices). Similarly more Optical Link channels can be provided for a subdetector if the volume of data from it is too great to be collected in the required time. Furthermore some resilience is built into the system (i.e. data taking can continue at a reduced rate if a component, a workstation for example, is not available during a run)

The aim of the dataflow design was to produce a scalable data acquisition system which can cope with the high data rates the experiment requires. There are two basic constraints on the system. Firstly the data is driven by the 40 MHz clock at the subdetector level. To buffer data at this rate requires fast and expensive static RAMs. Likewise, when the data is assembled into events the total volume of 100 MBytes every second requires the use of unconventional backplane technol-

ogy. Secondly no elements of the dataflow hardware can be allowed to introduce deadtime into the system. To achieve this the basic architecture consists of consecutive stages in which data is transferred from buffers to FIFO memories over point to point links. There is no horizontal linkage between the transfers from different sources until the Data Merger stage as to do this would necessarily introduce deadtime. Instead each link is controlled by an $\overline{\text{XOFF}}$, a signal sent from the destination to the source across the link. The source stops sending data when the destination asserts this line. This occurs when the destination has nearly filled its buffer (There must be enough room left to accomodate the data sent while the $\overline{\text{XOFF}}$ is still in transit). It is the responsibility of the trigger supervisor to ensure that the trigger rate does not overload the system. An overload results in the $\overline{\text{XOFF}}$ propagating back to the subdetector which must stop taking data (This is treated as an error condition). No error correction is performed by the system, only error detection takes place. An event in which a bad word is found is rejected by the Level 3 filter. This removes deadtime but also demands that the error rate is kept extremely low.

The buffer-to-FIFO architecture logically isolates each section of the dataflow from the others. The internal on-board logic between each buffer-FIFO connection can be altered without affecting the rest of the system. In particular the development of the system downstream from the Front End Workstations (FEWS) in the workstation farm is completely independent from the development of the parts of the system that lie upstream.

In the next Section the main parts of the Dataflow are described.

2.2 Dataflow hardware

2.2.1 Optical Links

The optical links used in the NA48 experiment were designed by Robert McLaren, Leslie McCulloch and Phillipe Brodier-Yourstone of the ECP division at CERN [16].

Each optical link carries two uni-directional channels, a 32-bit data channel and a 32-bit control channel. NA48 only uses one bit of the control channel as an end of event bit (D39).

The task of an optical link is to deliver data from the ReadOut Controller (ROC) of a sub-detector to the Data Merger. In general, each sub-detector has one Optical Link (excepting the LKr calorimeter which requires 8). Each link is a duplex pair which delivers data at 10 Mbytes/sec over a distance of around 200 m and provides electrical isolation. To the hardware that interfaces to the Optical Link it looks like an I/O register where words are clocked in and out one at a time. Although connected to many different sub-detectors, all links are the same; the input to the optical link is defined as the point at which the experiment's data is presented in a common format.

Electrical Specification

Sub-events to be transferred by the optical link must be formatted as 32-bit wide words. In addition to the user data, an additional bit (called 'D39') is set high on the last word of the sub-event. The user presents data, provides a strobe and then must wait for an acknowledgment strobe back from the optical link source before changing the data.

The Optical link source (OLS) runs a VCO at 265.625 MHz which is phase-locked to the user clock. Incoming 8-bit characters undergo 8-10 bit encoding. The 10-bit characters are then serialised and the bit-stream is transmitted over the optical fibre at this rate. This is a purely serial line with no clock pulse. When the link is first established, the transmitter sends synchronisation characters which lock the phase of the receiver clock to that of the transmitter. One of the reasons for the 8-10 bit encoding scheme is to ensure that all characters transmitted consist of an even mix of 1s and 0s. This avoids transmitting "DC" codes (such as 11111111) and ensures that there are frequent changes of state which keep the clocks synchronised. The Destination (OLD) deserialises the bit-stream and reforms the 10-bit characters which it decodes to recover the original bytes of data. These are then grouped in a 40-bit wide register (the original 4 data bytes plus a control byte

consisting of parity bits and D39) and then strobed into the Data Merger. When the source is not transmitting user data it sends a stream of idle characters to the destination. These keep the clocks in synchronisation [17].

On the last word the source sets the top data bit (D39) high. This is termed the *end_of_event* bit. It is low for all other transfers.

Flow control

In the Data Merger an Input Buffer (IB) receives the data from the OLD. The IB contains a 2.5 Mbyte buffer memory. If this begins to fill to capacity, the IB signals to the OLD by asserting the line $\overline{\text{XOFF}}$. This can happen asynchronously. When the OLD sees this signal, it sends an $\overline{\text{XOFF}}$ character back to the OLS using the other fibre of the duplex pair. This causes the OLD to inhibit sending acknowledge strobes to the ROC thus preventing data from being read in to the OLS. When the IB has cleared some space, it de-asserts $\overline{\text{XOFF}}$ and the destination sends an XON character to the source which releases the inhibit on the acknowledge strobe and the data transfers resume.

The IB asserts $\overline{\text{XOFF}}$ when it is nearly full, leaving sufficient space to read in any words still in the optical pipeline. Similarly, it de-asserts $\overline{\text{XOFF}}$ only when it has cleared space well below the nearly full level. This prevents the $\overline{\text{XOFF}}$ signal from fluttering.

Error Checking

The optical link monitors itself for errors in several ways. It inputs error information to the IB which stores it for read out at the end of each sub-event.

- **Parity Checking** The parity of each byte decoded is checked against the parity bit contained in the control byte. If a mis-match occurs, the OLD sets the PERR line to the IB.

- **Table Error** If a 10-bit character does not correspond to a valid code (of the 1024 possible 10-bit codes only 256 are used) the table error bit is set.
- **Running Disparity** The 8 to 10-bit encoding system uses two look-up tables to ensure that each bit changes every cycle. If a bit is at the same level for two consecutive words then the OLD sets the RDERR line to the IB. This could mean that the 10-bit character has been corrupted during the transfer.
- **Sequence Error** D38 is the sequence bit and this must toggle between each word transferred. If two consecutive words have the same value of D38 an error is detected and the SERR line is set in the OLD.
- **Hardware Errors** The optical link cards are designed to detect an open fibre connection and will set the Link Status line in this eventuality. In addition, if the power available in the laser exceeds a maximum value a Laser Fault line will be set.

2.2.2 Data Merger

The sub-event data generated by the NA48 sub-detector elements arrives at the Data Merger [18] in a bank of IBs. There is one IB serving each sub-detector. The IBs are read out sequentially over the R-path backplane, working at 100 MBytes/sec, to the FIFO Output Formatter (FOF) which concatenates the sub-events and feeds the complete event to the workstation farm via the HIPPI link.

The sub-events are time ordered and each sub-detector produces data for each trigger.

Figure 2.2 shows a schematic of the Data Merger Each IB outputs data onto the R-Path when it receives a token. This is a 40 ns wide logic high ECL pulse which originates in the FOF and is passed to each IB via lemo cable before returning to the FOF. When the IB that holds the token has output a full event the token is passed to the next board in the chain. As the sub-events are time ordered and each sub-detector produces data for each trigger a full event is built up at the FOF from the data from each of the sub-detectors.

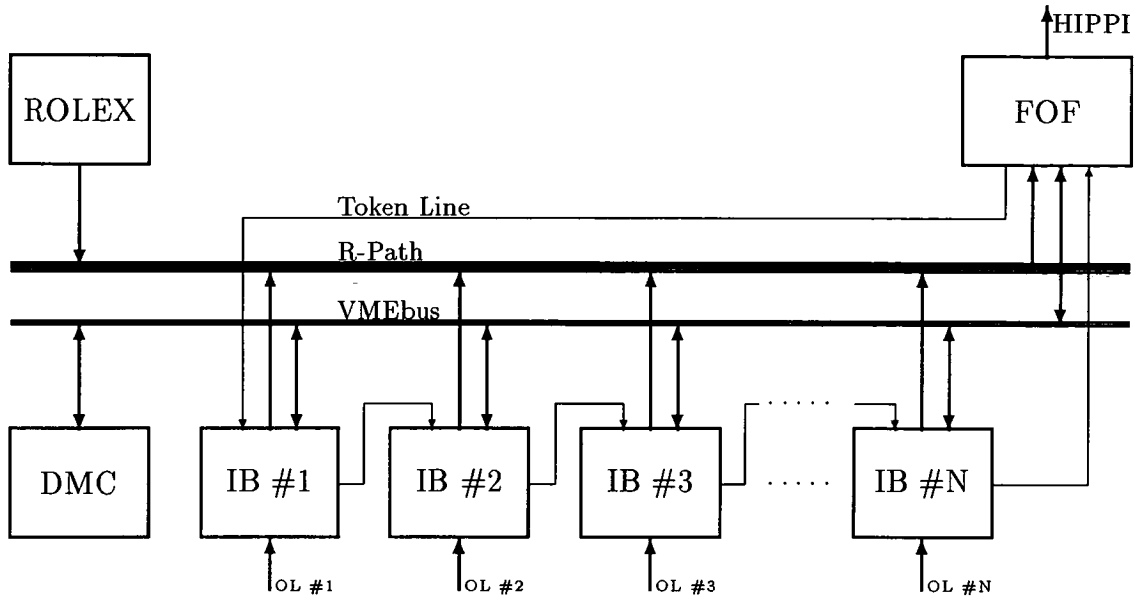


Figure 2.2: Layout of Data Merger

The FOF stores the events in a FIFO buffer before reading out to the HIPPI link at the appropriate time. The output can be sent to one of several workstations. The Single Board Computer (DMC) selects the data destination.

The DMC controls the system through VME. The DMC is a single board computer that receives instructions from the central run control program and reports back on the performance of the data merger. Each IB and the FOF are VME slave modules, i.e. they can respond to requests for information from a master module (in this case the DMC) but can not request the bus themselves.

The Data Merger is described in more detail in the next Chapter.

2.2.3 HIPPI Link

Point-to-point HIPPI links [15] are used to connect the FOF with the Front End Workstations. The High Performance Parallel Interface (HIPPI) is an ANSI stan-

standard describing a simple high-performance point to point communications link. It is designed for transmitting digital data at peak rates of 100 MBytes/sec (32 bit words) or 200 MBytes/sec (64 bit words) using multiple copper twisted-pair cables at distances of up to 25 meters.

The connection from source to destination comprises of one or more 'packets' of unlimited size. Each of these is divided into 'bursts' which can contain from 1 to 256 words. Error detection is based on a parity bit which is sent along with each byte of data and a Length/Longitudinal Redundancy Checkword (LLRC) which is sent at the end of each data burst.

The FOF is linked to a number of FEWS via a HIPPI crossbar switch. HIPPI provides a limited addressing feature that is used by the FOF and the switch to determine the destination of each packet. At the start of each connection the FOF transmits an address (the I-field) which is used by the switch to decide which workstation it will connect.

The workstation farm consists of a number of DEC TURBOchannel machines. An interface from HIPPI to TURBOchannel was developed as a joint project by CERN and DEC [19]. The interface uses a high speed FIFO to take data from the link. It then transfers the data to the workstation memory in a series of DMA transfers. So that the transfers are not hampered by any software intervention the board uses a scatter-gather table to translate logical addresses in HIPPI to physical ones in the workstation memory. Any errors that are detected are logged in a 'history' memory, the contents of which are sent at the end of each event.

Transferring the data block between the source and destination requires the cooperative work of the FOF, the HIPPI to TURBOchannel interface and the HIPPI links and switch. In addition, the Data Merger Controller controls the sending of the data over the HIPPI link by generating the I-field address of the destination workstation. This is sent to the FOF before the connection is made. The data distribution processes running in the workstation must also be able to free enough memory for all the data it receives.

2.2.4 The Workstation Farm

The 1995 hardware set-up of the workstation farm is shown in Figure 2.3. The system consists of seven workstations, arranged in front-end (FEWS), back-end (BEWS) and rear-end (REWS) groups. Each FEWS – BEWS pair is connected by two SCSI-2 busses and in turn to the REWS. Five 2 GByte disks are attached to each of the busses.

All of the workstations have equal access to the disks and the filtering and event reconstruction can be performed by any of the CPUs, however each group of workstations also has specific tasks to perform:

The Front-End Workstations highest priority task is to collect the data from the HIPPI to TURBOchannel interface and store it in memory. They also check if the data is intact before writing the data from memory to disk.

The Back-End Workstations concentrate on reading events from the disks to perform monitoring, event reconstruction and Level 3 filtering.

The Rear-End Workstation discards events that have not been triggered, re-formats the events into a ZEBRA structure ¹, appends information from the Level 3 trigger to the raw data and transfers bursts of data to tape.

The processed data is sent from the REWS to either the local backup system, consisting of several DLT 2000 tape drives, or the central CERN computing facility. It is here that the data is stored for future analysis.

In the future more workstations may be added to cope with increases in data volume.

¹ZEBRA is a CERN program which allows the creation of structures and dynamic memory allocation in Fortran[20].

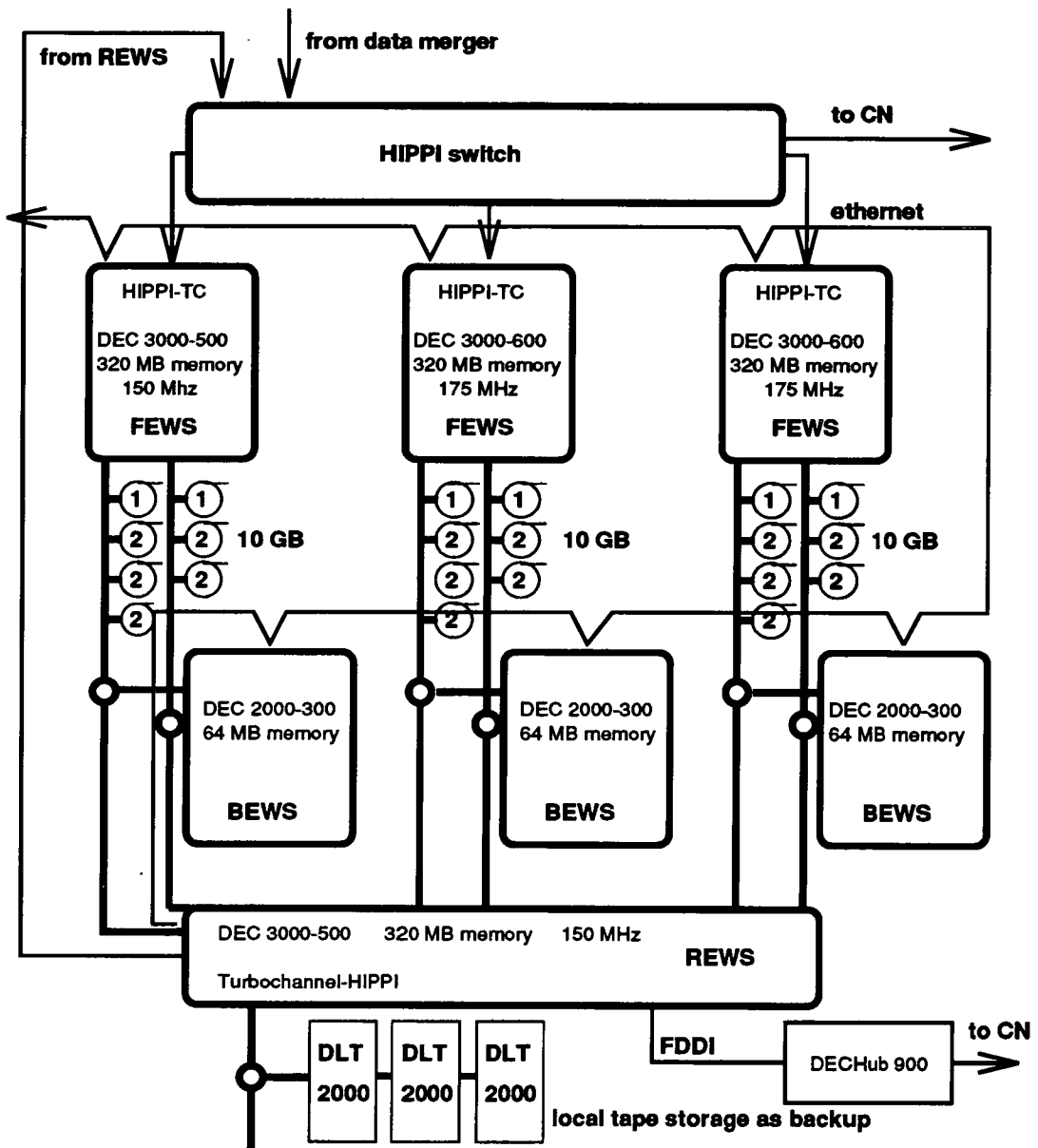


Figure 2.3: Level 3 Workstation Farm Architecture

2.3 Dataflow Control Software

2.3.1 The Run Control Program

The hardware components of the dataflow are set up and monitored by the Run Control Program. This program supervises and coordinates the activities of all the subcomponents in order to allow data to be taken. Its basic task is to start and stop runs, i.e. to configure the dataflow components at the start of data taking and to reset them afterwards. Each Single Board Computer (SBC) in the system runs its own control component (called the 'harness') which interfaces with the central component (called the 'control program').

The harness program running on each SBC monitors the activity of the dataflow electronics with which it interfaces. It performs the following functions:

- The initialization of the dataflow electronics. For example, the Data Merger SBC (the DMC) resets the IBs and the FOF by writing 'reset' to the control registers of the two boards. It also primes the FOF for data taking.
- Monitoring of events. The SBCs can access areas of memory directly to analyse the data at different stages in the dataflow chain.
- Error reporting. Dataflow components can indicate to the SBC that an error has occurred (via a VME interrupt for example). The SBC reports the error to the run control program.
- Calibration and checks of the electronics. These take place out of burst when data is not being processed by the system.

The run control program separates the data taking into 'runs'. A run is defined as a set of contiguous spills taken under identical running conditions. A new run is started whenever the conditions change. At the start of a run the CP sends a 'configure' signal to all the harness running components in the dataflow system and the harness sends back a 'system ready' signal to the CP. The communication

takes place over ethernet. During data taking, status checks, burst statistics and error information are transmitted to the CP. When the end of run command is received from the CP the SBC sends back run statistics and performs any clean up that is necessary.

2.3.2 The Workstation Farm Resource Manager

The workstation farm resource manager program controls the data flow from the FEWS to the permanent data store by managing the intermediate buffer storage and routing data through filter and reconstruction processes.

2.3.3 Slow Control

The slow control program monitors those variables that are not directly connected either to the physics data or to the dataflow hardware path. Examples of these variables are:

- The voltage and current of the power supplies.
- The temperatures of the electronics crates and the sub-detectors.
- The pressure of the gasses in the drift chambers.
- The monitoring and control of the high voltage supplies.

Chapter 3

NA48 Data Merger

3.1 General arrangement of the Data Merger

Two photographs of the Data Merger system are given in Figures 3.1 and 3.2 . The Data Merger is housed in two 21-slot VME crates. Crate 1 is 6U and contains a FIC 8234 single-board computer which acts as the Data Merger Controller (DMC). This crate also contains a BIT-3 repeater output board which allows the DMC to access Crate 2. These two boards are all that are required in Crate 1 for the Data Merger to function. However, Crate 1 may also contain various VME units for test and diagnostics such as a VFIFO, a VME-HIPPI interface, a SLATE pattern generator or a disk drive for the FIC. None of these units are essential for the operation of the Data Merger.

Crate 2 is 9U by 400 mm and is fitted with a standard VME backplane in J1 and J2. The R-path backplane occupies the J3 position and features 192-pin *Metral* connectors. Slot 1 is occupied by the Read Out Latch Enable Transmitter (ROLEX). This module generates the clock signals which control the data transfer on the R-path bus. The ROLEX has selectable clock speeds and a range of LEDs which indicate the state of the backplane protocol signals. Slot 2 is occupied by the slave board of the BIT-3 repeater. In this way, the DMC can access transparently, over VME, boards in Crate 2. Slots 3 - 20 are available for Input Buffer boards and Slot 21 is occupied by the FOF.

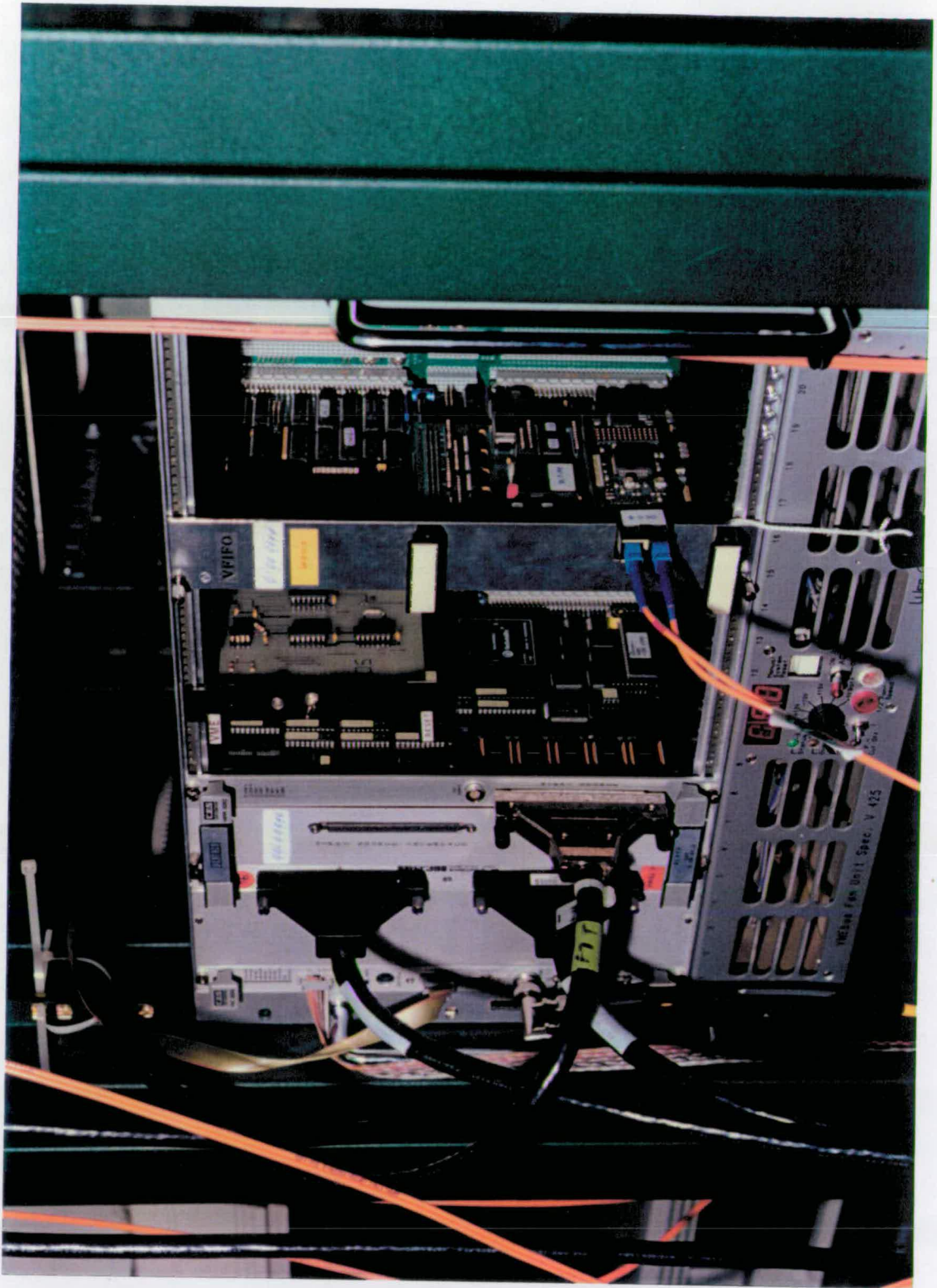


Figure 3.1: Photograph of the 6U crate used to house the DMC.

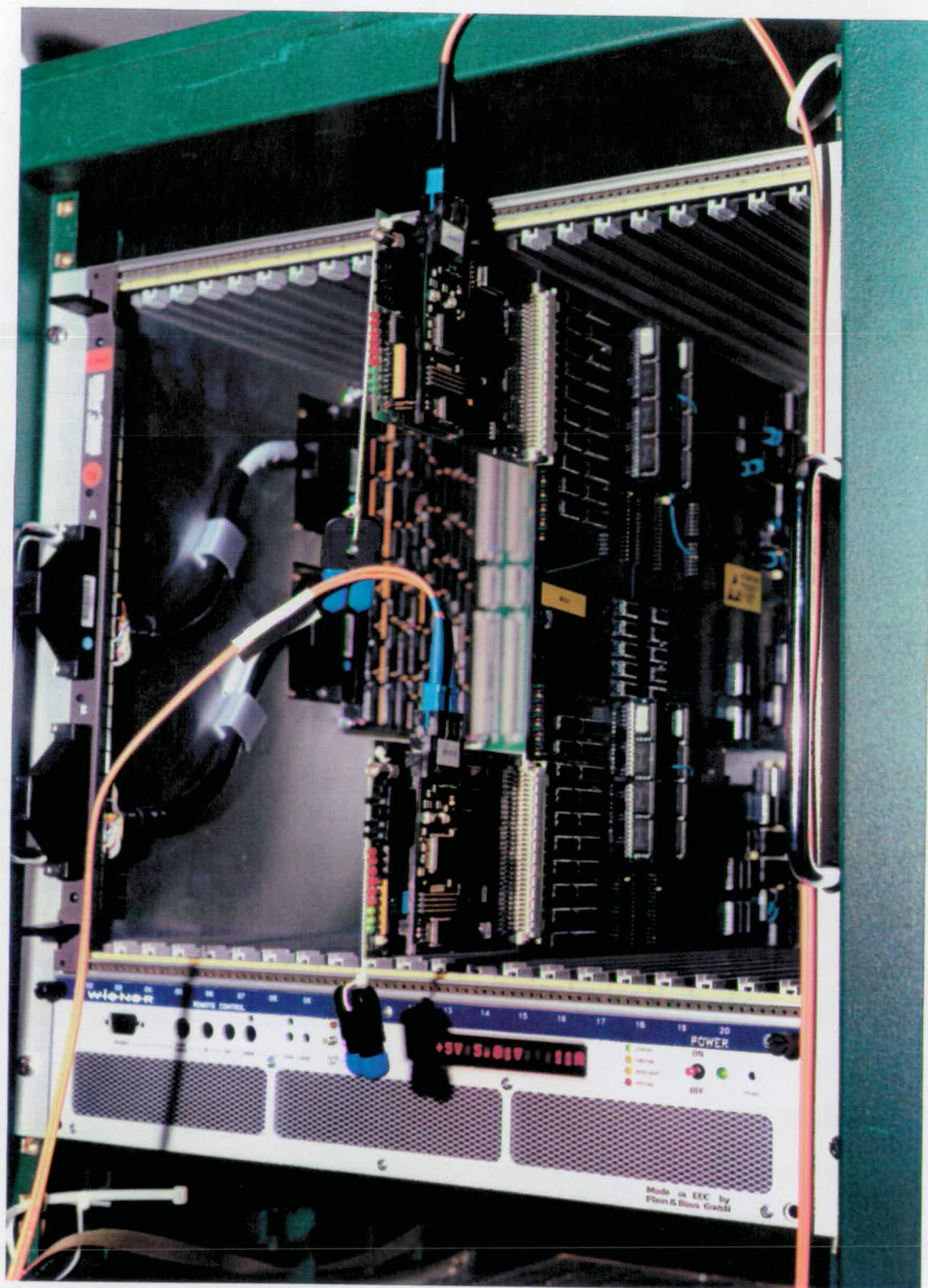


Figure 3.2: Photograph of the 9U crate that is used to house the Input Buffers and the FOF (the FOF is not present as it had not delivered to CERN at the time the photograph was taken).

3.2 Implementation Choices

3.2.1 Bus Implementation Choices

The set-up detailed in the previous Section was arrived at after the consideration of other possible systems.

The decision to make the Data Merger cards VME modules was taken at an early stage of the design. VME is a well developed bus standard which is widely used in High Energy Physics experiments at CERN and elsewhere. It has wide commercial support and it is a straightforward task to design circuit board interfaces to the system. However, the VMEbus is not capable of transferring data at the required speed. The maximum bandwidth for 32-bit wide data transfers is 40 MBytes/sec (moving up to 80 MBytes/sec in VME64, a 64-bit version where the data width is doubled by using the address lines as data lines [21]).

A study was carried out to assess whether any commercially available bus systems were capable of data transfers at the required 100 MBytes/sec rate. This showed that there were only two bus systems on the market which could meet the data rate requirement, **Fastbus** and **Futurebus+**.

Fastbus was found to have several advantages. It is capable of data transfer at rates in excess of 150 MBytes/sec[22], it is tailored for data acquisition in high energy physics experiments and it is a well defined IEEE standard. However, there is poor industrial support for the system and it is rarely pushed near its bandwidth limit.

Futurebus+ can run at speeds of up to 400 MBytes/sec in 32-bit mode and is designed to be easily integrated into other systems. However at the time of the study it was still a developing standard and was not due to be a full industrial system until 1993[23] .

It was concluded that going to a full **Fastbus** or **Futurebus+** implementation for the Data Merger would be expensive and would make integration with the rest of

the experiment difficult as the majority of the collaboration had already started designing in VME. Trying to fit a Fastbus or Futurebus+ backplane to the VME modules was also deemed unsuitable as the board and crate sizes are not compatible with each other.

This led to the conclusion that the only way to implement the Data Merger was to use a 9U VME crate with the R-Path backplane situated in the 'spare' J3 position.

Before the backplane was selected the choice of bus driver chips to be used was made. Although the full Futurebus+ standard was not finalised when the initial decisions were made the bus transceiver chips for the developing system were already available. The Backplane Transceiver Logic (BTL) interface circuits are open-collector drivers with a very low capacitance (5 pF). This technology has several advantages over TTL for bus driving applications. Firstly, bus settling time (the time required for crosstalk and reflections to subside before a receiver can reliably sample the bus) is eliminated if the backplane is properly terminated [24]. Secondly, the voltage levels in BTL are 1 V (logic low) and 2.1 V (logic high), these compare with the voltage swing of 3 V for TTL. The lower voltage swing reduces drive current requirements and induced crosstalk between lines.

Originally it was decided to use these transceivers along with a standard CERN backplane. A test set-up was constructed where a driver board transmits a chequer-board pattern along 8 lines of the backplane to multiple receiver boards. An example of 25 MHz operation is shown in Figure 3.3. These tests showed that the backplane was not ideally suited to our needs, increased loading of the backplane degraded the signal quality and cross-talk occurred between adjacent lines. The solution that was arrived at was to manufacture a backplane for the Data Merger.

The R-Path backplane was designed and built by Litton backplanes. It offers good protection from cross-talk and is tuned to the voltage levels that are used by the Futurebus+ transceivers.

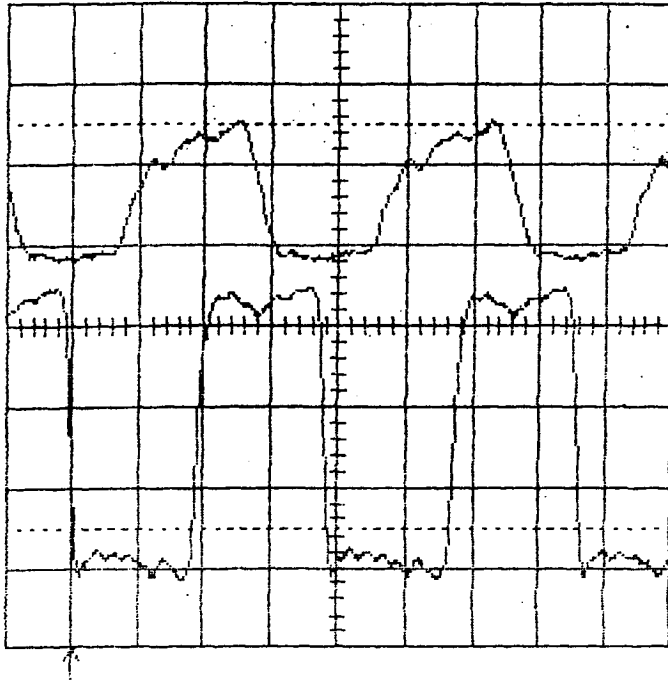


Figure 3.3: An example of BTL signals on the CERN backplane. The top trace shows a 25MHz frequency signal on the backplane, the lower trace shows the output of a receiver chip. $T/\text{division} = 10 \text{ ns}$ and $V/\text{division} = 1 \text{ V}$.

3.2.2 Circuit Board Implementation Choices

The IB and FOF boards have to be 9U high. This gives a large board area (400 mm x 365 mm) for the logic circuitry.

The majority of the IB control circuitry is implemented in a Xilinx Field Programmable Gate Array (FPGA). The logic circuit (described in Appendix F) is constructed by using the Xilinx range of logic and routing resources. The FPGA provides fast control of the board without occupying a large area. 256k x 32-bit SIMM static RAM modules are used to buffer the data from the Optical Links. These have a small footprint and are easy to address. Differential ECL is used for the token logic. This is less susceptible to noise than TTL and therefore less likely to produce a 'phantom' token if the token line is noisy. The remainder of the IB uses standard fast TTL logic circuits.

Due to the large area available and the space saved through using the FPGAs and SIMMs it was decided to put two IB channels on each board.

3.3 Overview of the Data Transfer Protocol

Data transfers are controlled at three levels of hierarchy; by spill, by event and by sub-event.

At spill level, the DMC receives signals on a serial line from the SPS computers. One second before the spill commences, the DMC receives the SPS signal, *warning_of_warning_of_ejection*. This is taken as the *start_of_spill* signal. The DMC then 'arms' the FOF over VME. This sets up the Data Merger for the coming spill. When the spill is over, the SPS signal, *end_of_ejection* is received by the FOF. The FOF then finishes reading out the IBs and then sends a *data_cleared* signal, via lemo, to the DMC. It then breaks its connection to the HIPPI switch. A spill may contain several thousands events.

An event is the combined output of all the sub-detectors when a trigger is issued. At event level, the data transfer is controlled by the token-ring. The FOF sends out a token when full sets of sub-events are waiting in the IBs. The token is passed along among the IBs and each IB can send its data out onto the R-path only when it is in possession of the token. When the last IB in the ring has been read out, it passes the token back to the FOF thus completing an event read out cycle. An event consists of several sub-events, one from each IB.

A sub-event is the block of data generated by any one sub-detector when a trigger is issued. The data arrives at the IB from the Optical Link as a series of 40-bit wide words. The words consist of 32 bits of user data (D0 - D31) and 8 bits of control data (D32 - D39). The control block is made up of 5 parity bits (one for each byte), 1 *end_of_event* bit and 2 reserved bits. At sub-event level, the words are transmitted unchanged from one IB to the FOF over the R-path backplane. The IBs participate in turn and each transfer follows the same protocol. A sub-event consists of several words of data (up to around 3k bytes).

3.4 Input Buffer

The Input Buffer is a dual channel 9U by 400mm board which sits in crate 2 (see Figure 3.2). Each channel takes events from one optical link and stores them until it receives a token. When the token arrives the IB sends an event to the FOF across the R-Path. The board also adds a word count and the OL error bits to the data. The word count is added as an extra word and the error bits are placed in the control byte of the last word. The IB memory consists of two banks (ODD and EVEN) of 40 bit-256 kByte static RAM. The control signals to these memories come mainly from a Xilinx FPGA chip. When one bank is being written to by the OL, the other can be read out through the R-Path. This ensures no break in data transmission to the FOF if data arrives over the OL when the module has the token.

The IB has a simple VME slave interface that is used for diagnostic purposes. The control byte is multiplexed down onto the 32-bit data word at the VME interface.

The design and testing of the input buffer has formed the main part of my work during my PhD and a more complete discussion of its function will be given in Chapter 4.

3.5 R-path backplane

The R-Path carries the 40 bit data words from the IB to the FOF. As well as the 32 bit data word and 8 bit control word the R-Path backplane also carries the following signals:

- **DATA_VALID** : This is a strobe that is sent by the IB to the FOF along with each data word transferred. The signal goes low on the backplane for 20 ns while the data word is being sent. The FOF uses this to latch the data into its buffers.

- **$\overline{\text{DATA_ACK}}$** : The FOF returns an acknowledge signal to the IB for each word that it receives.
- **$\overline{\text{RP_NODATA}}$** : The R-Path NoData line on the bus is driven low by an IB when it does not hold a complete sub-event. When the board receives an end of event word from the Optical Link it de-asserts this line. As the bus is wired-or the line does not go high until all the IBs have de-asserted their $\overline{\text{RP_NODATA}}$ s. A high tells the FOF that a complete event has arrived in the Data Merger buffers and hence it can send a token to read out the system.
- **RP_HAVEDATA** : This signal performs a similar function to the $\overline{\text{RP_NODATA}}$ line. It is asserted (driven low on the backplane) by any IB that has data in its memory. This data does not necessarily need to be full sub-event. A low on the RP_HAVEDATA line warns the FOF that one or more IBs contain data to be read out even if a full event is not present in the system.
- **$\overline{\text{RP_XOFF}}$** : The $\overline{\text{XOFF}}$ line is driven low by the FOF when its buffers are full. This forces the IB to stop sending data.
- **+5 V** : There are 30 5 V pins on the backplane. These provide extra power to the IB and FOF boards as well as a supply to the ROLEX.
- **-5.2 V** : 21 pins provide power for the ECL token circuitry.
- **+2.1 V** : The BTL transceivers require each line on the backplane to be terminated by a $33\ \Omega$ resistor to 2.1 V and a capacitor to ground. This supply provides the termination voltage.

The *start_of_spill* signal from the SPS causes the DMC to set a bit in a register in the FOF. This action is carried out over VME and is asynchronous with any R-path signals. When this bit is set, the FOF monitors the signal $\overline{\text{RP_NODATA}}$ (set high when all the IBs have an event in memory). When this happens the FOF sends out a token along the token line. Upon receiving the token each IB sends out one event and passes the token to the next IB in the chain. The IB sends a DATA_VALID pulse along with every word it transmits and the FOF sends a DATA_ACK back to the IB for every word that it receives. If there is an error and the number of DATA_ACK edges does not tally with the number of DATA_VALID

edges then the IB does not send on the token and the data merger stops. Each IB drives RP_NODATA low when it has cleared its memory.

During the transfer, it may happen that the FOF fills its buffer. If so, like the IB to the OL, it sets $\overline{\text{XOFF}}$ low. This can happen asynchronously.

On seeing the $\overline{\text{XOFF}}$ signal go low, the IB finishes the transfer of the current word. That is, the data remains valid until the next rising edge of the R-Path clock (RP_CLOCK1), and DATA_VALID continues to strobe. At that rising edge of RP_CLOCK1, the address counters in the IB freeze and the data is not changed. DATA_VALID finishes low. This continues while $\overline{\text{XOFF}}$ is held low.

When the FOF has cleared sufficient space, it sets $\overline{\text{XOFF}}$ high (again asynchronously). On the next rising edge of RP_CLOCK1, a new data word is presented by the IB and DATA_VALID starts to strobe again.

When the last IB in the chain completes its read out, it passes the token back to the FOF. On receiving the token back, the FOF knows that a complete set of sub-events has been read out.

When the protons from the SPS are directed away from our experiment, an *end_of_ejection* signal from the SPS is sent to the FOF. When the FOF receives this signal, it could be in the middle of an event read out cycle. It completes this read out then checks RP_NODATA. If it is low, it sends off another token and reads out the event and so on until RP_NODATA goes high. At this point, since the spill is over, no new events can arrive.

3.6 FIFO Output Formatter

3.6.1 Overview

The FIFO Output Formatter (FOF) was designed and built by Senerath Galagedera and Ben Brierton at the Daresbury Rutherford Appleton Laboratory.

The FOF is the output component of the Data Merger. It recognises when a complete set of events is waiting in the IBs and issues a token. It then receives data from the R-path backplane, buffers it in FIFO memories and sends it out again over a HIPPI link to the workstation farm. The FOF also adds event headers, word counts and error information and re-formats the data for HIPPI transmission. The transmission over HIPPI and the reception on the R-path are decoupled and asynchronous and can proceed simultaneously. The FOF can exert flow control by asserting the $\overline{\text{XOFF}}$ signal in the R-path backplane.

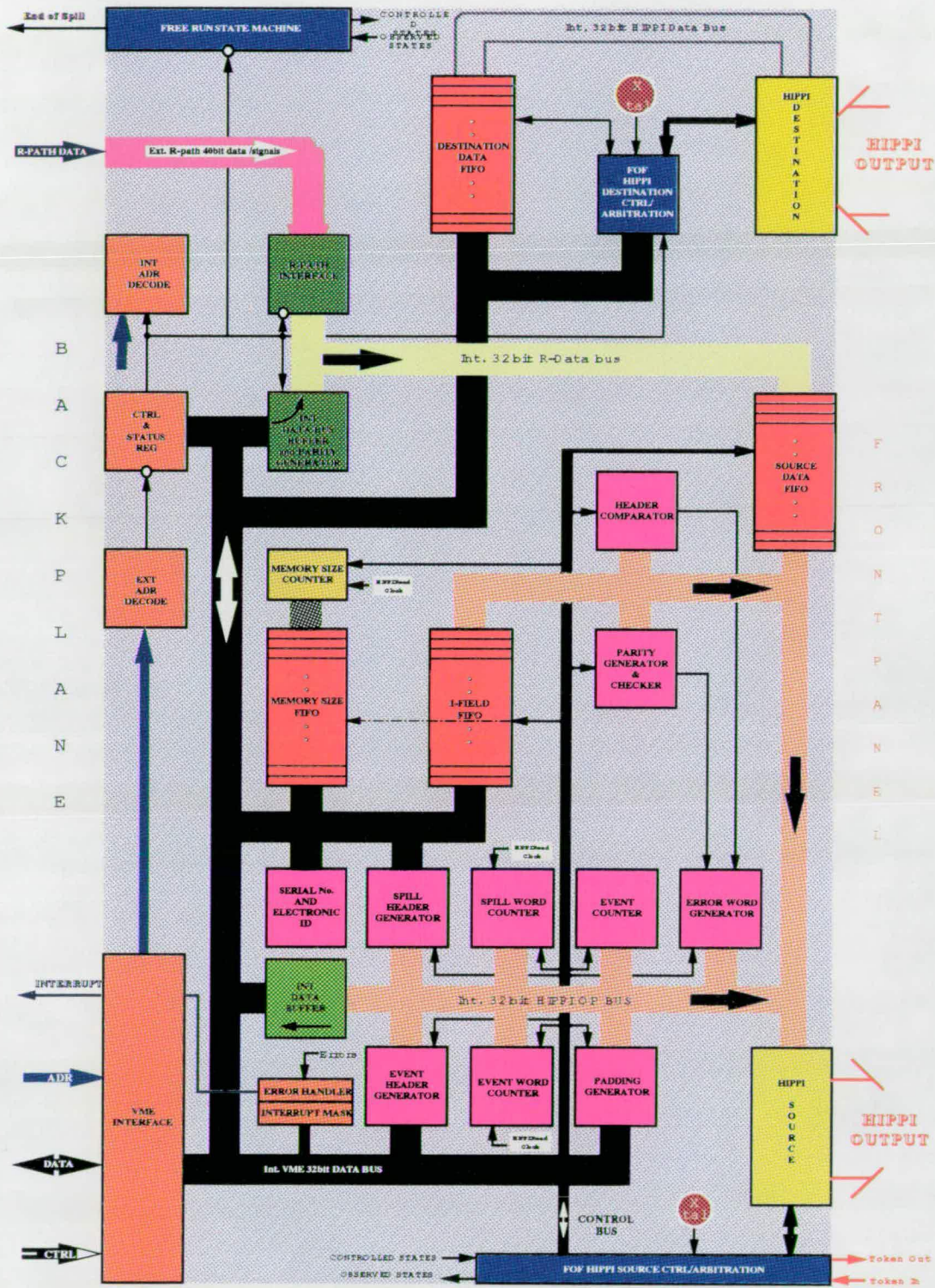
3.6.2 General Arrangement of the FOF

The FOF is a $9\text{U} \times 400\text{ mm}$ board which is mounted in the right-most slot of the Data Merger. A block diagram of the FOF is shown in Figure 3.4.

In normal operation, data is received from the R-path backplane via Futurebus+ transceivers and stored in a source data FIFO. At the appropriate moment, the data is shifted out of this FIFO to a HIPPI source chip which transmits the data out of the FOF via a HIPPI cable. The HIPPI source can also receive data from the I-field FIFO, the memory size FIFO and from various data generators. The data generators are contained in a Xilinx FPGA and generate the spill header, the event header, the padding and the error word. There are also an event counter, spill word counter and event word counter in the Xilinx the contents of which can also be sent to the HIPPI source.

The FOF is also equipped with a HIPPI destination chip which can receive HIPPI data and write it to the destination FIFO. By connecting the source to the destination with a cable, the FOF can send data to itself for self-test purposes.

The VME interface allows access to the data generators, the memory size FIFO, the I-field FIFO, the destination data FIFO and to the control and status register via VME.



The Data Merger Read Out Controller

Figure 3.4: Block Diagram of the FOF

3.6.3 R-path and token interface

Data arrives on the R-path backplane synchronously with the strobe `DATA_VALID` and the two R-path clocks, `RP_CLOCK1` and `RP_CLOCK2`. The latter is used to gate the data through the R-path transceivers and `DATA_VALID` is used to latch the data into the source data FIFO. The `DATA_VALID` signal is returned by the FOF as `DATA_ACK`.

The data transmitted along the R-path is accompanied by parity checking bits which were originally generated by the Optical Link Source (OLS). The FOF generates new parity bits for each data byte which arrives and compares the new bits with those from the backplane. Any time a mis-match occurs, an error counter is incremented. At the end of the event, the contents of the error counter are written to the error word generator.

If the source data FIFO becomes full during the transfer, the FOF drives the $\overline{\text{XOFF}}$ signal low. This inhibits whichever IB is sending data and thus pauses the dataflow on the backplane. When the source data FIFO has cleared, the FOF removes $\overline{\text{XOFF}}$ and the dataflow resumes.

The token is generated in the FOF and consists of an active-low pulse in differential ECL of 40 ns duration. This pulse is transmitted from a front panel socket along co-axial lemo cable. The returned token from the last IB is received by a complementary circuit and converted to TTL.

3.6.4 Additional data generated by the FOF

As well as the user data, the FOF generates the following additional blocks of data which are used for book-keeping, debug and analysis:

- **The I-field:** This is a HIPPI packet which is read by the HIPPI-switch and which tells the switch through which output to route the next HIPPI connection. The I-field is written to the I-field FIFO by the DMC before the spill begins.

- **Spill Header:** This is a serial number for the spill and is sent at the beginning of the connection.
- **Event Header:** This contains a serial number for the event.
- **Spill Word Count:** The total number of words sent during a spill. This includes all protocol and book-keeping words.
- **Event Word Count:** The total number of words in an event.
- **Padding:** A string of between 2 and 255 words consisting of an incrementing number sequence (i.e. 00, 01, 02 etc.). The size of the string is determined by the contents of the padding register which is loaded by the DMC.
- **Error Word:** The number of parity errors detected in the event.

3.6.5 The HIPPI transfer protocol

The various blocks of data sent by the FOF are formatted according to the HIPPI protocol. 32-bit parallel HIPPI which consists of differential-ECL signals transmitted over twisted-pair wires is used. HIPPI data is organised at four levels of hierarchy; connection, packet, burst and word.

- **Connection:** A connection can have an indeterminate duration (several seconds). During a connection many packets are sent.
- **Packet:** A packet has an indeterminate duration and the delay between packets can be indeterminate. A packet can contain any number of bursts and one short burst.
- **Burst:** A burst consists of exactly 256 words and is thus of a fixed duration. There can be an indeterminate delay between bursts. A short burst has between 1 and 255 words. There can only be one short burst per packet and this must be either the first or the last burst of the packet.
- **Word:** A word is the data which is read on the 32 data lines during one clock-cycle. The data is changed at a fixed rate of 25 MHz and so the time between adjacent words is 40 ns.

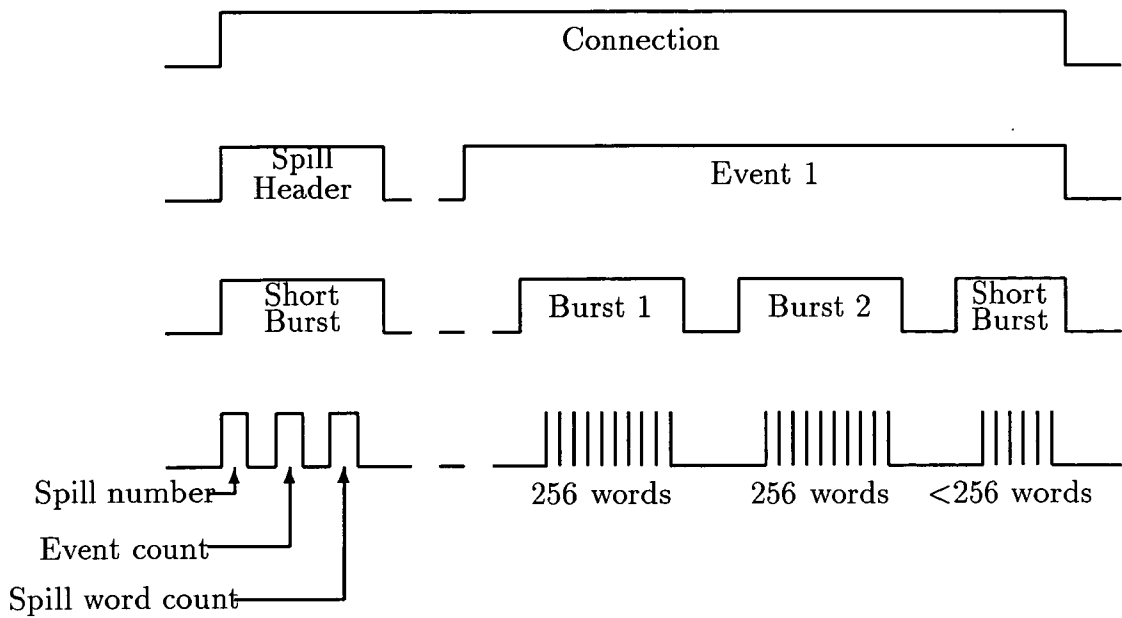


Figure 3.5: The structure of a FOF HIPPI connection.

In the Data Merger, a HIPPI connection is established for the duration of a spill (2.7 seconds).

The first and last packets are the spill head packet and the spill tail packet. These two packets consist of only three words (i.e. one short burst of three words). The words are; spill header (serial number), event count (zero in the spill head packet) and spill word count (3 in the spill head packet). The structure of the connection is shown in Figure 3.5.

The intermediate packets are event packets. An event packet consists of; event header (serial number), padding words (2-255 words), data words (the user data — perhaps many thousand words), padding words repeated, error word and event word count. The packet is split up into as many bursts of 256 words and up to one short burst as required. The structure of an event is shown in Figure 3.6.

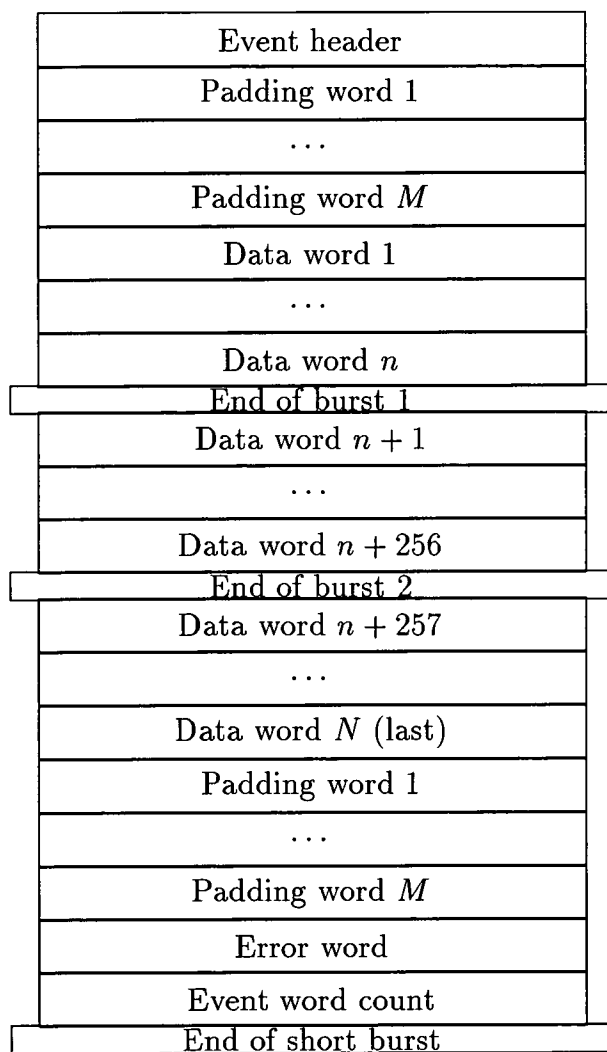


Figure 3.6: The structure of an event packet. In this example the data words start in the first burst and continue until the middle of the third burst. The third burst is a short burst.

Chapter 4

Input Buffer Design

4.1 Overview

As stated in the previous Chapter the Input Buffer (IB) takes data from a sub-detector via an optical link and transmits it at 100 MBytes/sec over the R-Path to the Output Formatter (FOF). Each board contains two IB channels, each of which consists of an optical link interface, a token handling circuit, two banks of static RAM, a memory control circuit implemented in a Xilinx FPGA and logic to control the flow of data through the board. Common to each channel is a VME interface for diagnostics and a TTL/BTL interface to the R-Path. For each channel there are two data paths – the write data path which is multiplexed between the OL and the VMEbus and the read data path which is multiplexed between the R-Path and the VMEbus.

Figure 4.1 shows a diagram of the board layout illustrating the position of the chips and the connectors. On the front of the board are two 96-pin Eurocard connectors, labelled OL0 and OL1 into which are plugged the OL receiver cards. In between these are mounted 10 LEDs per channel and four lemo sockets for the token ring. The token passing logic is also situated at the front of the board. The rear panel is fitted with two standard 96-pin Eurocard connectors, J1 and J2, which interface to the VMEbus and one 192-pin Metral connector which interfaces to the R-Path.

Data arrives from the OL in the form of 40-bit wide words. Each word consists of 32 bits of data and 8 bits of control information (5 parity bits, 1 end_of_event bit and 2 reserved bits). The words are stored in two banks of fast static RAM. The two banks are termed the ODD and EVEN memories. The board data memory is implemented in the SIMM modules U21, U23, U61 and U63. These modules contain 8 4-bit by 256k static RAM chips arranged in parallel. For the control memory, the 8-bit wide static RAMs U22, U24, U62 and U64 are used. The memory control and addressing is provided by two Xilinx FPGA chips, U27 and U67, which control channel 0 and channel 1 respectively. Apart from the above ICs the rest of the chips shown in the diagram are mainly multiplexers and latches which are used to route the data through the board.

4.2 Design Philosophy

During the initial design phase several requirements were set down for the IB, these were:

- The design should be modular, i.e. The different areas of the board are independent from each other and each is testable without relying on any input from the others.
- The design should be flexible in case changes in the design specification are necessary at a later date or the initial version does not work.
- Plentiful diagnostic resources should be put on the board to aid with debugging.

To satisfy the first requirement each area on the board was designed to interface with the Xilinx. As the logic in the FPGA can be changed, it can be routed to test one area at a time. The circuits that interface to the Optical Link, R-Path and VME can be tested independently from each other. For example, the board can take data from the OL and output it to the VMEbus without an interface to the R-Path being present.

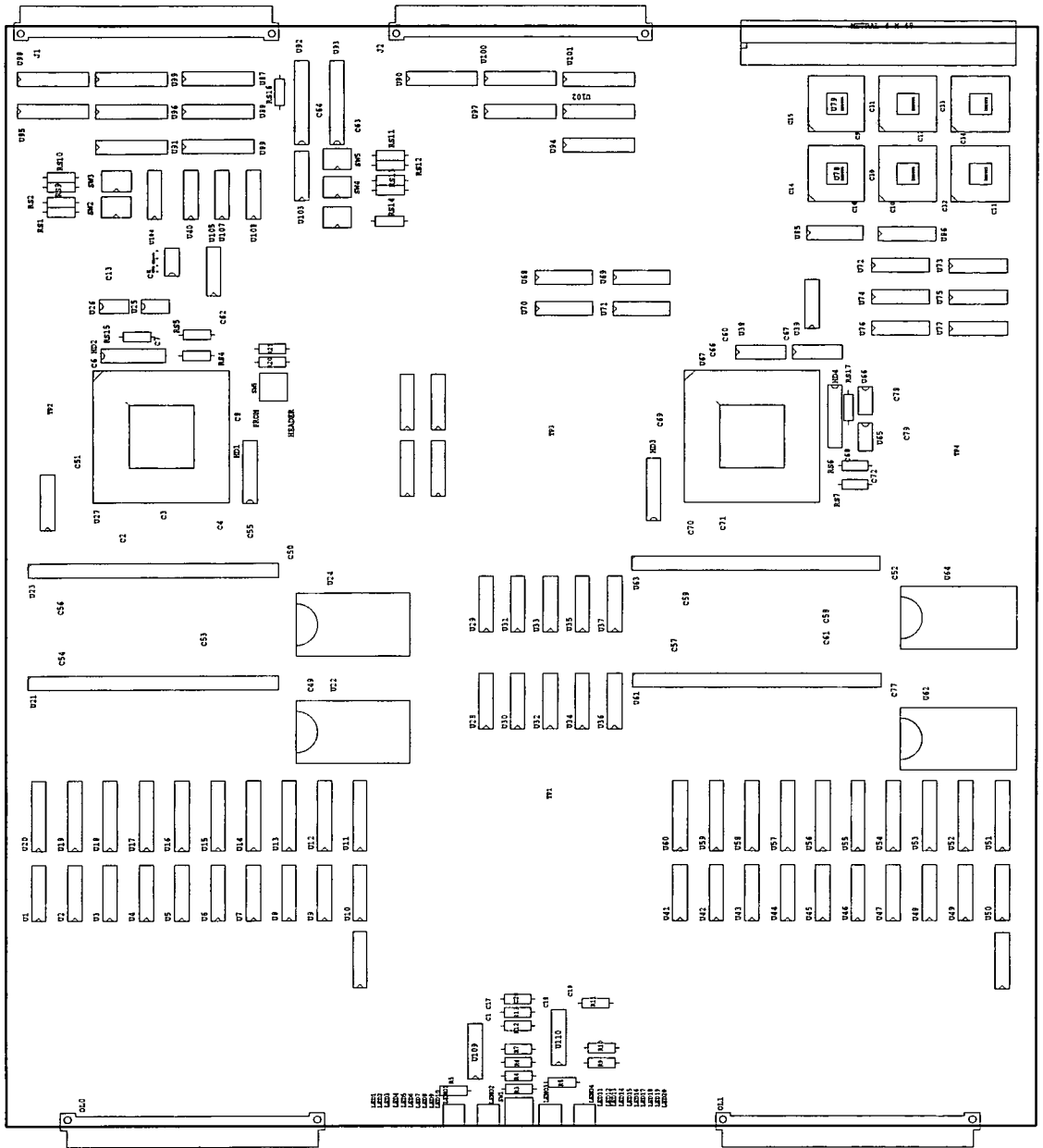


Figure 4.1: IB board layout

Putting the majority of the control logic into the Xilinx FPGA also makes the IB design flexible as it can be rerouted an unlimited number of times. However, signals internal to the Xilinx cannot be probed while debugging the board. To overcome this problem 10 LEDs and a 16 pin header were routed to spare pins on the Xilinx of each channel. Any line internal to the FPGA can be taken out to the spare pins where it can be analysed. In the final design the LEDs are defined as shown in Table E.1 and the header pins are left undefined.

The VMEbus interface was included to provide a way of looking at the contents of the IB memories. Data sent from the OL can be sent to a VME master to show if the OL interface is working. A test board which stores data sent from the IB across the R-Path before reading it out over VME was also built. The data from this board shows how well the R-Path interface performs. The FOF also includes a VME interface for diagnostic purposes.

4.3 Design Timetable

Two versions of the IB have been produced. The ‘prototype’, which interfaces only to the OL and the VMEbus, and the ‘pre-production’ board which fully satisfies the specification. A third ‘production’ board is to be made which will incorporate some minor changes from the pre-production board. The timescale for the design and manufacture of the completed IBs is shown in Figure 4.2. In this Chapter the design of the pre-production IB is described.

4.4 Input Buffer Logic Circuitry

The board can be split into six main logical sections:

- **The VME interface** consists of a series of trancivers which interface to the VMEbus data lines, address buffers which interface to the VME address lines, logic circuitry to decode the address sent to the board by the DMC

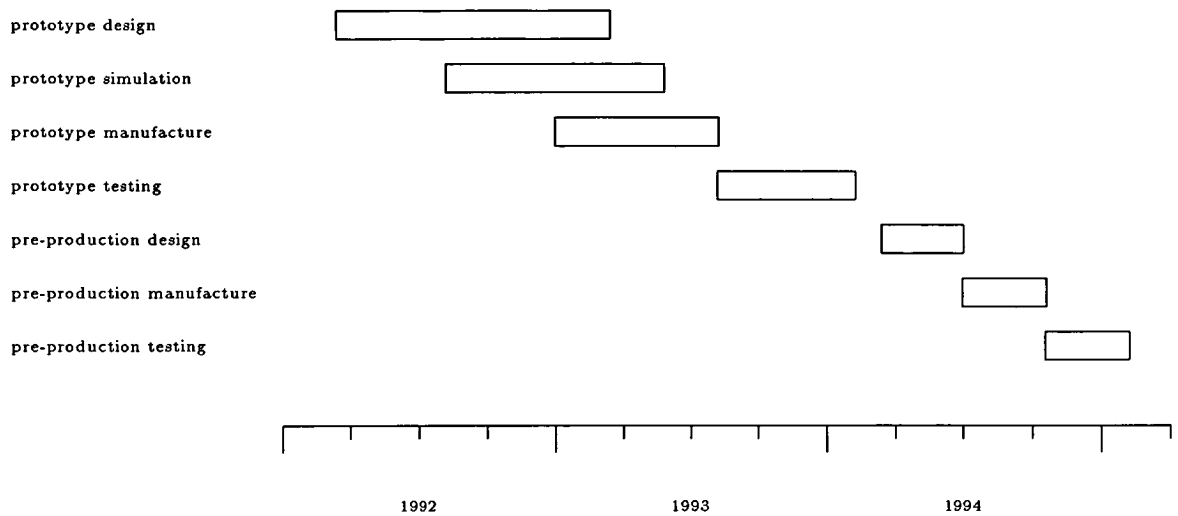


Figure 4.2: Timescale for Input Buffer Development

and a commercially available chip from PLX technology which handles the reception and transmission of the VME protocol signals.

- **The Optical Link interface** is controlled by logic circuitry contained in the channel Xilinx FPGA. On the board the data from the OL arrives at the IB on a 96-pin eurocard connector and is multiplexed with the VME data before being sent to the memories.
- **The R-Path interface** is simply a series of BTL buffers which interface to the R-Path data and protocol lines. The signals that control the transmission of data to the bus are again mostly contained in the programmable chip.
- **The token handling circuit** consists of two ECL-to-TTL converters and two TTL-to-ECL converters. The token is passed around the ring as a logic high ECL pulse. This is converted at the IB to TTL where logic inside the Xilinx uses it to start the transfer of data. The channel generates a TTL token_out pulse at the end of its data transfer which is converted to ECL and passed to the next channel.
- **The front panel LEDs** are included to give an indication of the board status. These are useful for debugging purposes.



- **The memory controller** is contained within the channel Xilinx FPGA. It controls the flow of data to and from the memories.

The first five of these sections are described in detail in Appendices A to E.

As was stated previously, the memory controllers are implemented in two Xilinx FPGAs. In the following section the architecture of the Xilinx and the programming methods that were used are described.

4.5 Xilinx Design

4.5.1 Xilinx Layout

The Xilinx XC4008pg191-6 Field-Programmable Gate Array comprises of three major configurable elements [25]: Configurable Logic Blocks (CLBs), Input/Output blocks (IOBs) and interconnections. These elements can be re-programmed an unlimited number of times. XC4000 chips range from the XC4002 (a 64 IOB, 64 CLB device) to the XC4020 (a 240 IOB, 900 CLB device). The XC4008 was chosen as it is the smallest device that can accomodate all the signals that we wish to generate.

Each FPGA is structured as a matrix of CLBs interconnected by metal segments with programmable switching points. The segments form a grid of horizontal and vertical lines that intersect at a switch matrix between each (or every other) CLB. The matrices consist of programmable n-channel pass transistors. By programming the connections between the lines at the intersections the desired routing between the blocks can be implemented.

Figure 4.3 shows a block diagram of an XC4000 CLB. The block has thirteen inputs and four outputs which connect to the programmable interconnect resources outside it. The two logic function generators with outputs labeled F' and G' can implement any boolean function of their four inputs. In addition to these another function generator can combine F' and G' with an outside signal H1. The function

generator outputs can leave the CLB through the X or Y outputs or be routed to one of two D-type flip-flops. The flip-flops latch the data on the edge of the Clock (K) and have Chip Enable (EC) and Set (SD) or Reset (RD) inputs. The source of the flip-flop data input is programmable. A direct input (DIN) can be latched in place of a function generator output.

The IOBs provide the interface between the external pins and the internal logic. There is an IOB for each package pin and each can be configured for input, output or bi-directional signals. An output enable input to each IOB can be used to produce 3-state output signals.

4.5.2 Xilinx Configuration

The Xilinx has six configuration modes selected by a 3-bit input code applied to three special input pins M0, M1 and M2. On the Input Buffer two of these modes can be selected.

Master Serial Mode

By setting M0, M1 and M2 to ground the master serial mode is selected. In this mode the configuration data is downloaded from one or more PROMs in synchronisation with a clock supplied by the Xilinx. This process takes place when the board is first powered up. As the PROMs are not re-programmable this mode is only used when the Xilinx design is stable.

Slave Mode

Slave mode is selected when 5V is applied to all three mode pins. Here the Xilinx reads the data from a serial line at the rising edge of a clock supplied by the programming device. Xilinx supply a download cable (XChecker) which can interface the device to the I/O port of a computer. As the configuration data is stored on the computer this means that the configuration can be changed without

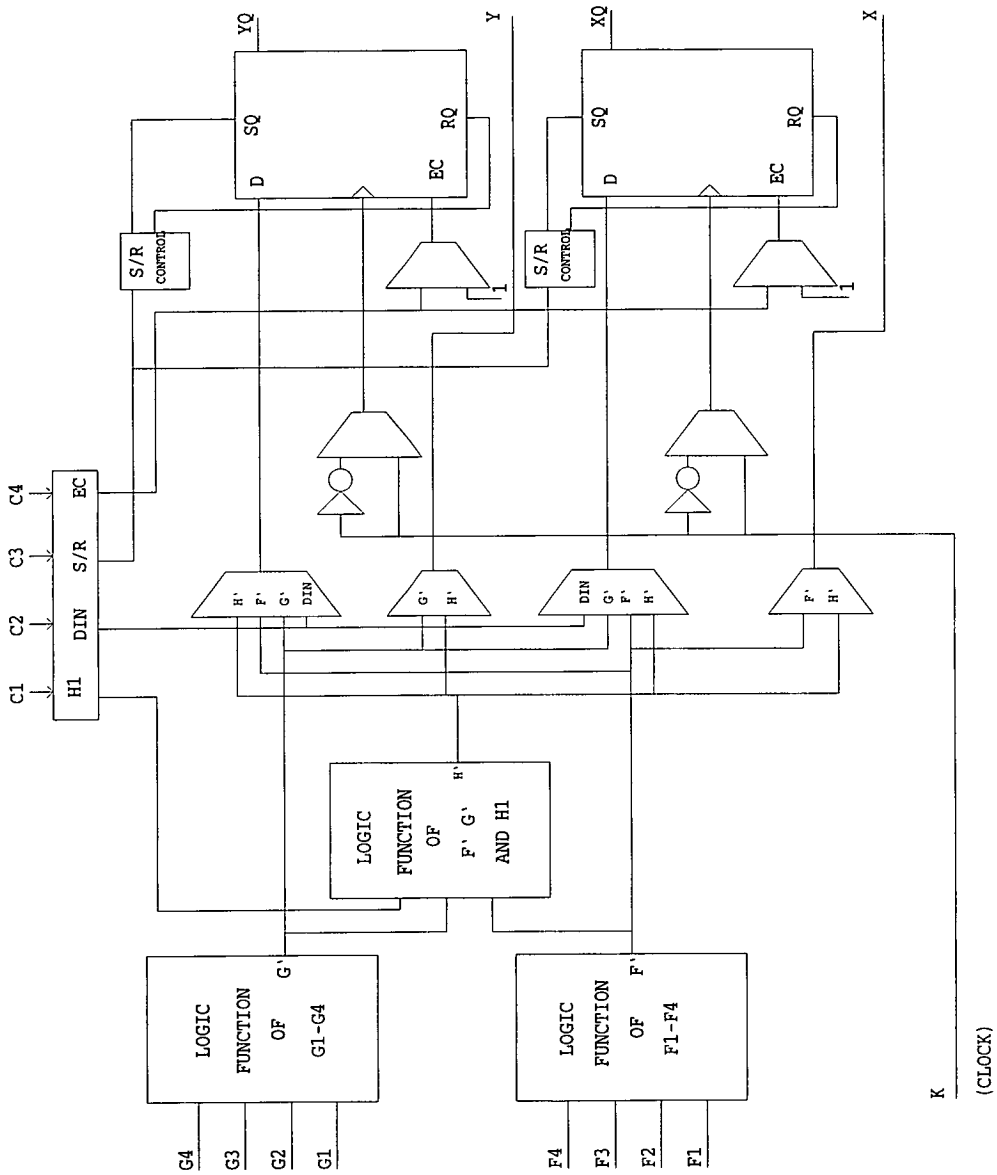


Figure 4.3: Block Diagram of XC4000 Configurable Logic Block

programming more PROMs. This mode is used when the Xilinx design is under development.

On the IB a jumper changes the voltage on the mode pins between 0 and 5 V. Two Xilinx XC1765 PROMs and an XChecker header are provided. During the debugging of the IB the slave mode is used to program the devices, but in the experiment the FPGAs are configured from PROMs.

4.5.3 Design Entry

The design is entered using the schematic-capture package OrCAD Schematic Design Tools [26]. From the schematics produced the Xilinx XACT software creates a .lca and a .bit file. The .lca file is a version of the schematic that is fully routed for a particular Xilinx FPGA (in this case the XC4008pg191-6). This file is used to make the .bit file which contains a bitstream that can be downloaded into the device. In addition, Xilinx provides a design editor (XDE) from which direct changes to the .lca file can be made [27]. Editing of the device takes place mainly at the schematic level, but small changes can be made directly from XDE to avoid running the routing program. This saves time, as the routing of the final schematic takes approximately 3 hours. However, all changes must be mirrored in the schematic to avoid confusion.

4.5.4 Xilinx Fuctionality

The functions of the Xilinx can be split into the four areas illustrated in Figure 4.4.

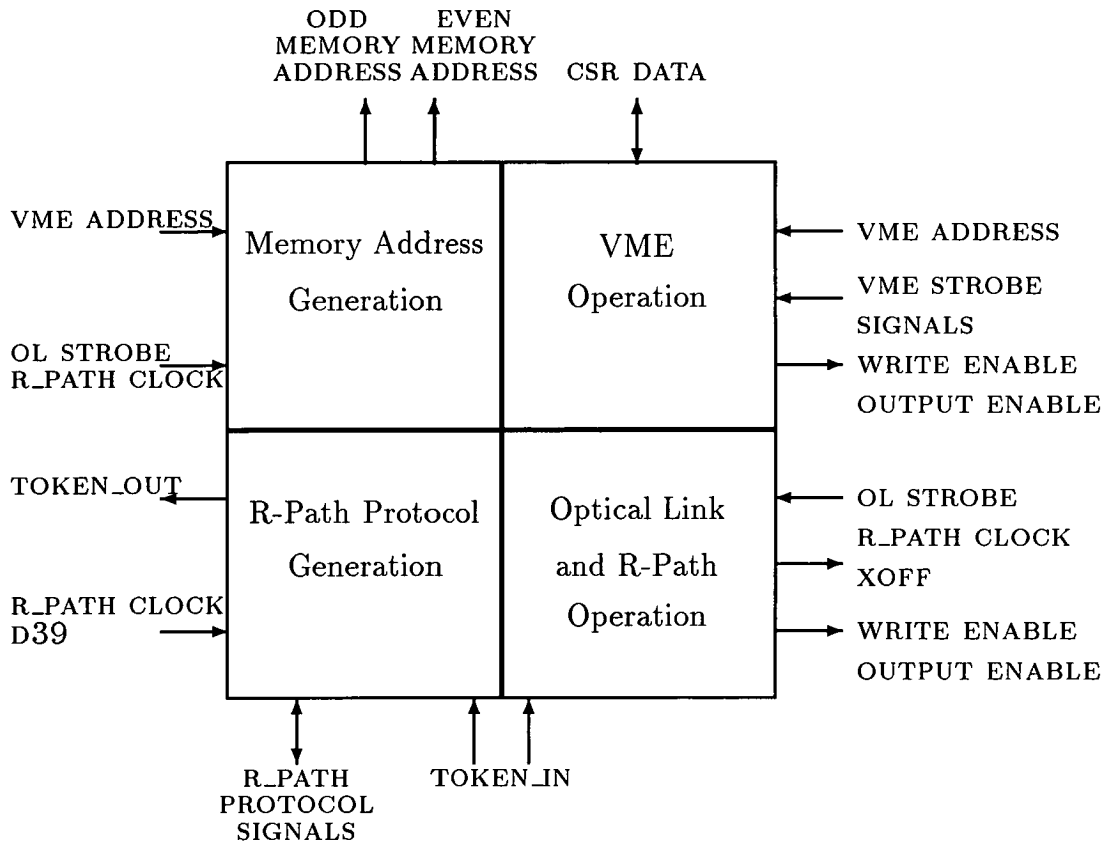


Figure 4.4: Block Diagram of Xilinx

VME Operation

When the board is addressed over VME the two FPGAs decode the VME address lines A20-A23 to determine which part of the board is being accessed. The addressing scheme is shown below.

Base Address +	Board Area
0	Channel 0 CSR
0x400000	Channel 0 Data Memory
0x600000	Channel 0 Control Memory
0x800000	Channel 1 CSR
0xc00000	Channel 1 Data Memory
0xe00000	Channel 1 Control Memory

If the memories are addressed then the relevant Xilinx sends the required write enable or output enable pulses to the memories. The Control and Status Registers (CSRs) are implemented in the FPGAs. If these are addressed during a read cycle then the relevant Xilinx sends the data from its internal registers to the VMEbus. If they are written to the data is latched into the registers in the FPGA.

Optical Link and R-Path Operation

During data taking each channel takes data from the OL and sends it to the R-Path. The OL and R-Path interface section generates the control signals that are sent to the memories during the IB read and write cycles. The OL write cycle is governed by a strobe signal that is sent to the channel along the optical link. Write enable pulses are sent to the odd and even memories alternately on the falling edge of the strobe. In addition this section of the Xilinx contains circuitry to stop the data taking when the IB becomes full. This is done by sending an $\overline{\text{XOFF}}$ signal to the OL destination module.

The transmission of data to the R-Path is governed by a clock that is generated in the ROLEX module. When the channel has the token, i.e. it is sending data to the R-Path, an output enable is sent to the odd or the even memory during each

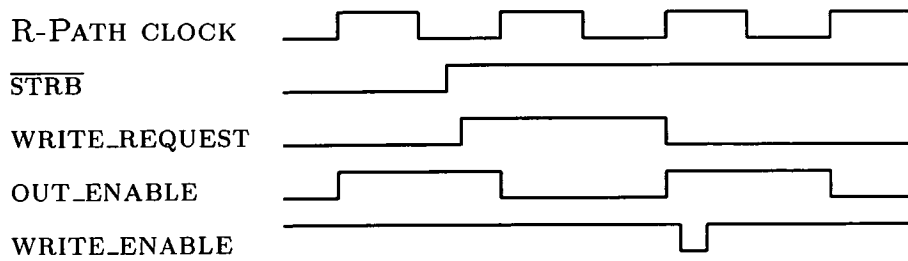


Figure 4.5: Generation of Write Enable for even memory

clock cycle.

The R-Path clock runs at 25 MHz independently from the OL strobe, and therefore the two are not in synchronisation with each other. For the IB to be able to write and read simultaneously one operation must take precedence over the other. As the OL strobe is slow in comparison with the R-Path clock, the write must be queued until the memory has completed a read cycle. To do this the strobe pulse is latched when it is seen in the Xilinx creating a ‘write_request’ signal. This is held until the memory that is being written to has been read from. At this point a short (approximately 15 ns) write enable pulse is sent to the memory. Figure 4.5 illustrates this process.

At the end of the transfer a word count is output from the CSR section of the Xilinx to the R-Path. This helps in the data reconstruction process. After the word count has been transmitted the token is transferred to the next module in the chain.

Memory Address Generation

The two banks of static RAM present in each channel are given addresses from the channel Xilinx. Three different addresses can be sent to each of the memories:

- **The write address.** This is generated in the Xilinx and then sent to a memory when a word is to be written into that memory from the Optical Link.

- **The Read address.** This address is also generated in the Xilinx and is sent to a memory when a word is to be output from that memory to the R-Path.
- **The VME address.** The VME address lines A2 – A19 are sent to each memory during a VME write or read cycle.

R-Path Protocol Generation

The R-Path protocol signals described in Section 3.5 are also routed to the FPGAs. Each Xilinx generates the following signals:

- **$\overline{\text{DATA_VALID}}$** : The Xilinx drives this line high on the IB for 20 ns while the data word is being sent.
- **$\overline{\text{RP_NODATA}}$** : The Xilinx outputs a high on this line when the channel memory contains at least one full event.
- **RP_HAVEDATA** : This line is low on the IB when the channel contains any data, i.e. the memory is not empty.

In addition to these signals the FPGA also handles the R-Path signals that are sent from the FOF to each IB:

- **$\overline{\text{DATA_ACK}}$** : Data acknowledge is a strobe signal that is sent by the FOF for each word that it receives. The Xilinx compares the number of acknowledges with the number of data valids at the end of each event. If the numbers do not tally then the token is not sent.
- **$\overline{\text{RP_XOFF}}$** : The FOF drives this line low on the R-Path (high on the IB) when its data buffers are full. The Xilinx on the channel that is sending data then stops the transfer after the next full cycle.

A more detailed description of the logic internal to the Xilinx is given in Appendix F. The next Chapter describes the testing of the IB.

Chapter 5

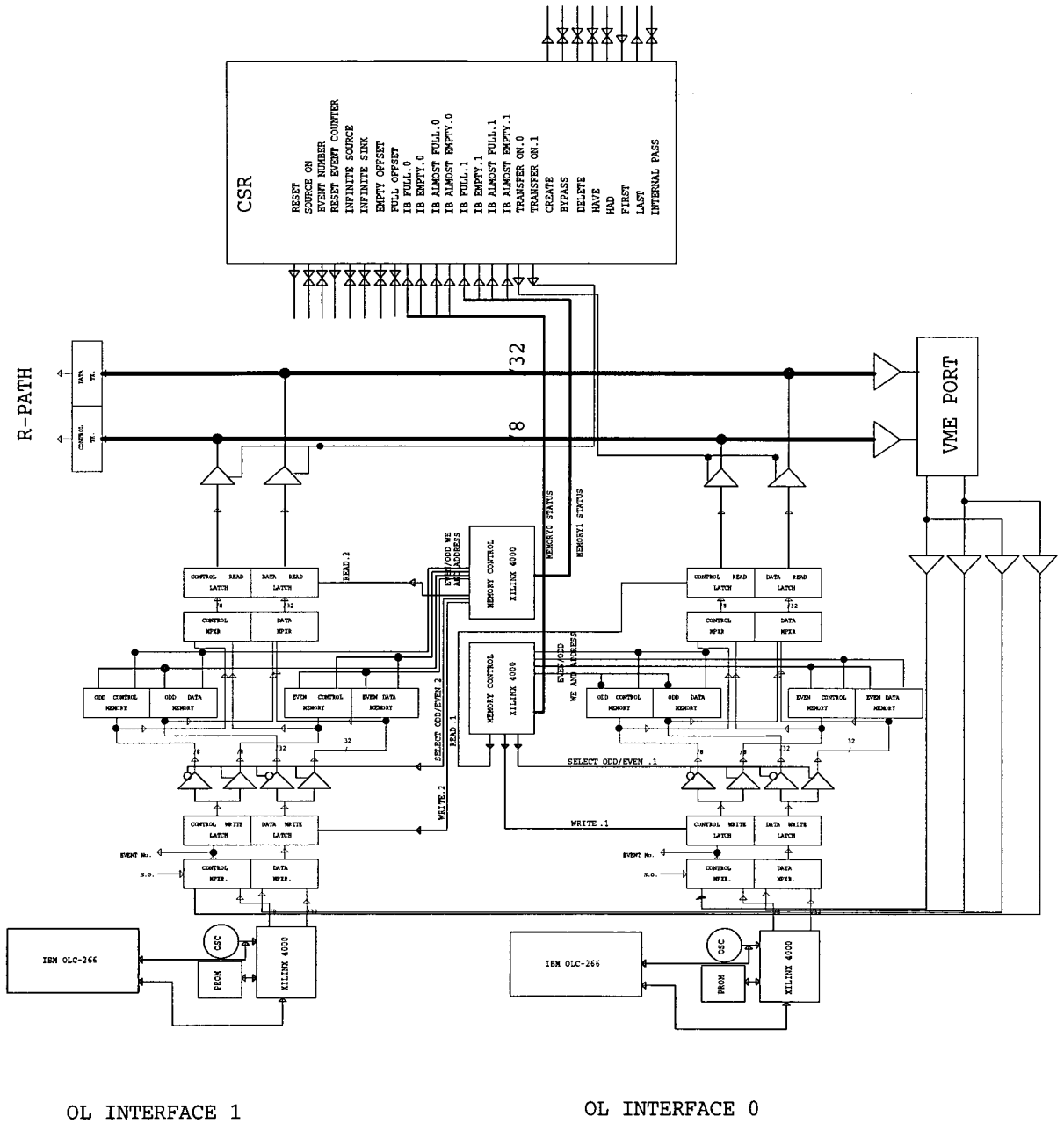
Test Results

5.1 Input Buffer Commissioning

The basic IB block diagram, shown in Figure 5.1, breaks the design into 5 separate areas. These are the OL interface, the memory banks and associated logic, the VME port, the R-Path interface and the control and status registers. From this diagram the logic circuits to implement each section were designed. In addition, the control signals required from the Xilinx to each section were defined.

Many refinements were made to the basic design during the design process. For example the hardware for the OL destination, made up of an IBM OLC-266 optical receiver card, a voltage controlled oscillator and a 4000 series Xilinx with a PROM, was originally included on the IB board. However, in the final design it was decided to put the destination on a separate card which plugs into a 96-pin Eurocard connector on the front of the IB. This enabled separate development of the OL and IB and also allowed the use of different technologies on the two boards. However, the destination board sticks out from the crate, making a protective screen around it essential. It was also decided that the CSRs would be implemented in the FPGA chip in order to make the choice of register bits changeable.

During the layout and manufacture of the prototype simulation of the memory bank structure and the OL and R-Path interfaces took place. Also the Xilinx circuitry was investigated on a test board. The board consisted of a socket, from



OL INTERFACE 1 OL INTERFACE 0

Figure 5.1: Initial IB Block Diagram

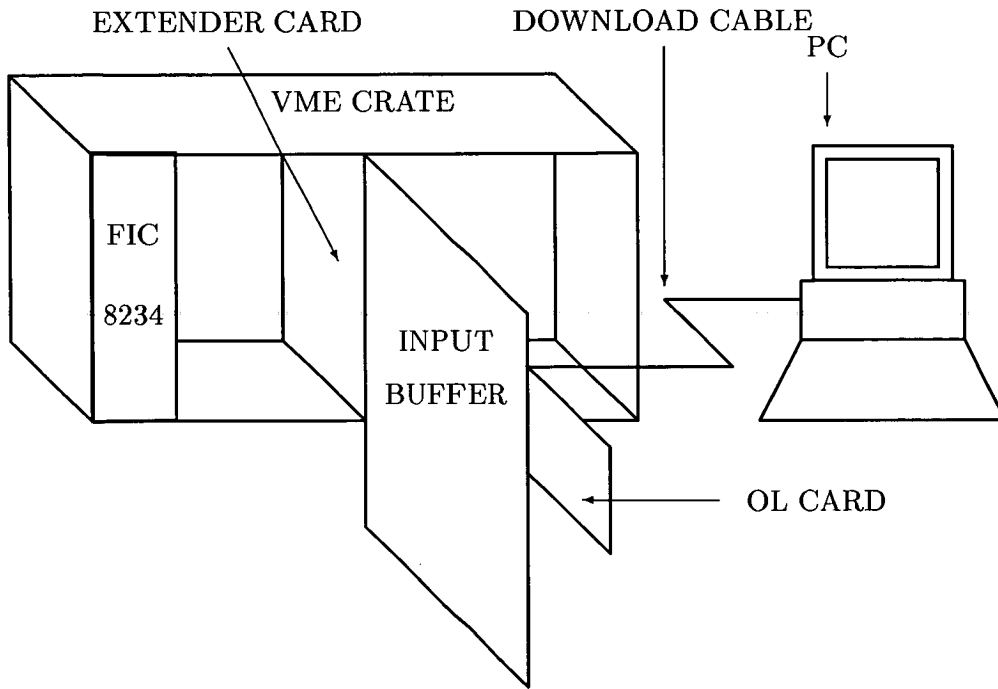


Figure 5.2: Prototype IB Test Set-up.

which a Xilinx FPGA could be inserted and removed, and header pins for downloading .bit files from a PC. Test pins were wire-wrapped to each pin on the chip. Using this setup the OL, R-Path and VME circuitry internal to the Xilinx was tested.

The prototype of the IB was delivered to Edinburgh in November 1993. The system used to debug the board is shown in Figure 5.2. The 9U high IB board was plugged into a 6U VME crate via an extender card. This enabled easy access to the board with an oscilloscope. The Xilinx were programmed from the bit files stored on the PC using the XChecker download cable. A FIC 8234 module from Creative Electronic Systems was used as the VME single board computer. This was accessed over ethernet from a workstation. This set-up enabled full testing of the OL interface, memory banks and VME section of the board.

The R-Path interface was not populated as the decision to build a custom R-Path backplane was not made until after the design was sent for manufacture. The original backplane used 96-pin Eurocard connectors whereas the custom built R-Path uses 192-pin Metral connectors. The prototype was designed to interface with

the original backplane and hence did not have R-Path capabilities. However, Some R-Path testing could take place using a pulse generator to provide RP_CLOCK1 and looking at the signals at the input to the R-Path area.

To test the memory banks the VME interface was populated and the channel 0 Xilinx was inserted. Programs to write test patterns to the module and read them back were run from the FIC. Once the memories were tested the rest of channel 0 was then built up with the exception of the R-Path interface. The Optical Link interface was tested firstly with an OL emulator, which sent a variable number of data words each accompanied by a strobe, and then with OL source and destination modules. At this stage the IB was capable of receiving words across the OL, storing them and sending them across the VMEbus backplane to the master module. The second channel was then populated and tested in the same order as channel 0.

Two prototype boards were built and debugged by August 1994. These were used in a NA48 test run in September 1994. During the run data was taken from two subdetectors, the hodoscope and the tagger. The setup for the run is illustrated in Figure 5.3.

In this system the data from each spill is sent from the sub-detectors to the IBs and then across VME to the Data Merger Controller. The data is then sent by the DMC to the VME to HIPPI board, a 6U high board consisting of a VME interface and a HIPPI source. A workstation with a HIPPI to Turbochannel interface is used to store and process the data.

This arrangement does not merge data. The OL and VME interfaces on the IB cannot function simultaneously and so the data from each 2.7 second spill must be stored and then read out in the subsequent 12 seconds before the next stream of protons is sent to the experiment. The merging of the data from the two sources is carried out in the workstation.

In the 1994 test run the system processed 10^6 events (1 kByte/event) over a 4 week period.

After these tests one of the prototype boards was delivered to the Seigen group to allow them to test the electronics for the Drift Chamber readout. The Seigen

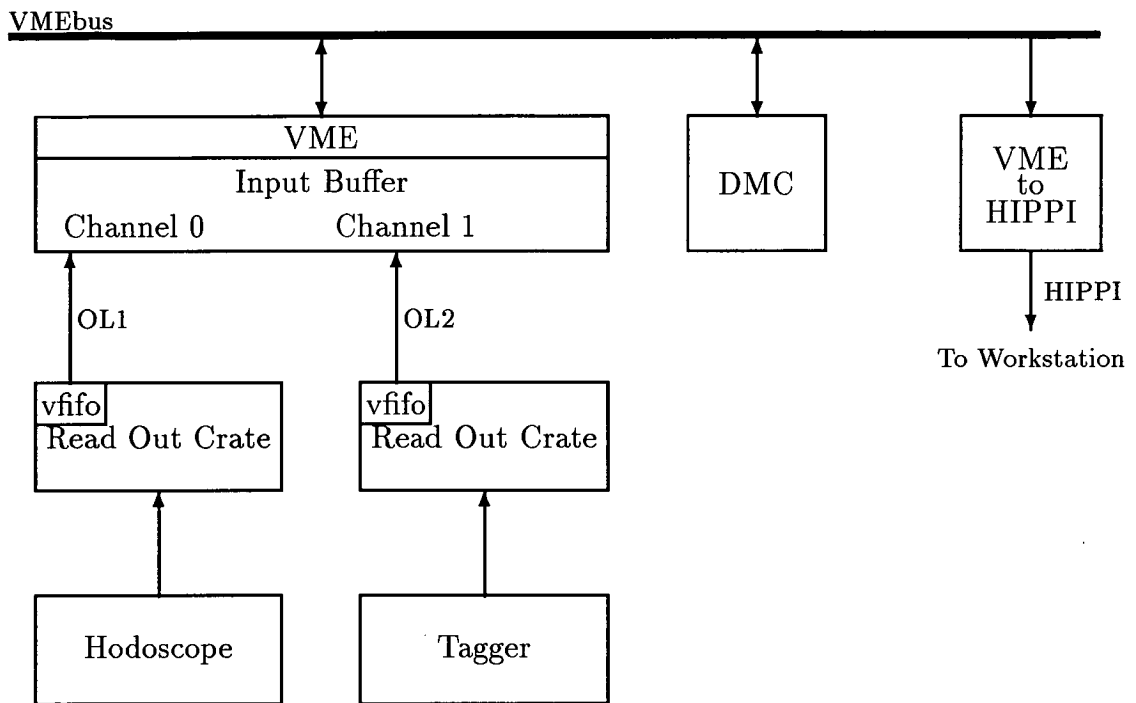


Figure 5.3: Dataflow Hardware Setup for the 1994 Test Run

group used the IB as a sink for the OL data and a VME readout module.

Several modifications to prototype were necessary before the IB could be integrated into the full Data Merger system. These modifications were finalised in March 1994 and fell into the following categories:

- Correction of tracking errors from the prototype.
- Addition of front-panel switches, spare logic, ground pins, token-ring sockets and front-panel mounting holes and changing the LED configuration.
- Conversion of CSR bus on the IB to 20-bit from 16-bit.
- Conversion of J3 connector to 192-pin Metral from 96-pin Eurocard.

Two pre-production boards were built and populated in November 1994. The testing of these boards, plus the R-Path, was carried out at CERN during December and January 1995. The test set-up was modified to enable R-Path data transfers by putting the IB in a 9U crate with the R-Path backplane in the J3 position. The 6U crate containing the FIC interfaced with the IB through a Bit-3 repeater card. This card connects the J1 and J2 backplanes on each of the crates together so that from the point of view of the VME modules the system is contiguous.

The pre-production board performed all the functions of the prototype by the start of December 1994. To debug the R-Path area of the IB a test board (the 'FOFETTE') was built with VME and R-Path interfaces. One bit in an on-board control register switched between VME and R-Path operation while another sent a token signal to a lemo socket on the front of the board.

To test the IB R-Path data transfer the FOFETTE was put in R-Path mode and a token sent to the IB. The data transmitted across the R-Path was stored in a small static RAM on the test board and then read out through VME. This test revealed that the data and the DATA_VALID pulses at the start of the transfer took approximately 200 ns to settle. This problem was traced to the temporary R-Path terminations that were being used in the absence of the FOF¹.

¹The FOF terminates one end of the R-Path bus.

The FOF arrived at CERN in March 1995 and was integrated into the Data Merger system. Firstly data was loaded into the IB and transmitted along the R-Path to the FOF, which was then interrogated by the DMC. This test showed no errors in the user data but an extra, random word was sent by the IB at the start of every transfer. The full system (vfifo to optical link to IB to R-Path to FOF to HIPPI to a workstation) was then tested. Several problems were found:

- The IB always transmitted an extra word at the start of the event. This problem was due to the IB starting to transmit DATA_VALID pulses too early.
- The last user word (the end of block) was not seen in the workstation. This was due to the FOF switching from the IB data to the padding data too early. The first padding word after the event was also not seen due to the above problem.
- The spill word count sent from the FOF was incorrect.

These minor problems did not affect the user data which was transmitted correctly. The R-Path protocol lines detailed in Section 3.5 were also shown to work.

The final commissioning of the Data Merger was carried out in May 1995. The problems found in March were eliminated and a setup to test the full dataflow system was put together. The tests are set to start in mid-July 1995 and real data will be taken in August of this year.

5.2 Input Buffer Testing

The testing of the pre-production boards was completed in early 1995. This Section shows the performance of a working channel.

5.2.1 Optical Link Data Transfer

Writing data from the Optical Link into the Input Buffer

The Optical Link transmits 32-bit data to the IB at 10 MBytes/sec. The IB uses the OL strobe (STRB), which is low for 100 ns in the middle of each data word, to clock the data into its two banks of memory. The generation of the memory write strobes is shown in Figure 5.4. The following signals are displayed:

1. STRB is shown on Channel 1. The falling edge of this signal enables the memory write cycles.
2. The Write Enable Odd (WEO) strobe goes low for 15 ns on every alternate cycle. Every odd word from the OL is written in to memory on the WEO rising edge.
3. The Write Enable Even (WEE) strobe is active during the input of each even word.

Optical Link Transfer Interrupted by $\overline{\text{XOFF}}$

When the IB is almost full it drives the $\overline{\text{XOFF}}$ line to the OL destination low. When this reaches the OL source the data transfer is stopped until the IB has cleared sufficient space in its buffers to allow the transfer to recommence. When this happens the IB drives $\overline{\text{XOFF}}$ high and the OL source resumes transmitting

Figure 5.5 shows the IB de-asserting the $\overline{\text{XOFF}}$ line. Channel 1 on the oscilloscope displays the $\overline{\text{XOFF}}$ signal while channels 2 and 3 show the memory write strobes, WEO and WEE recommencing after the $\overline{\text{XOFF}}$ line goes high.

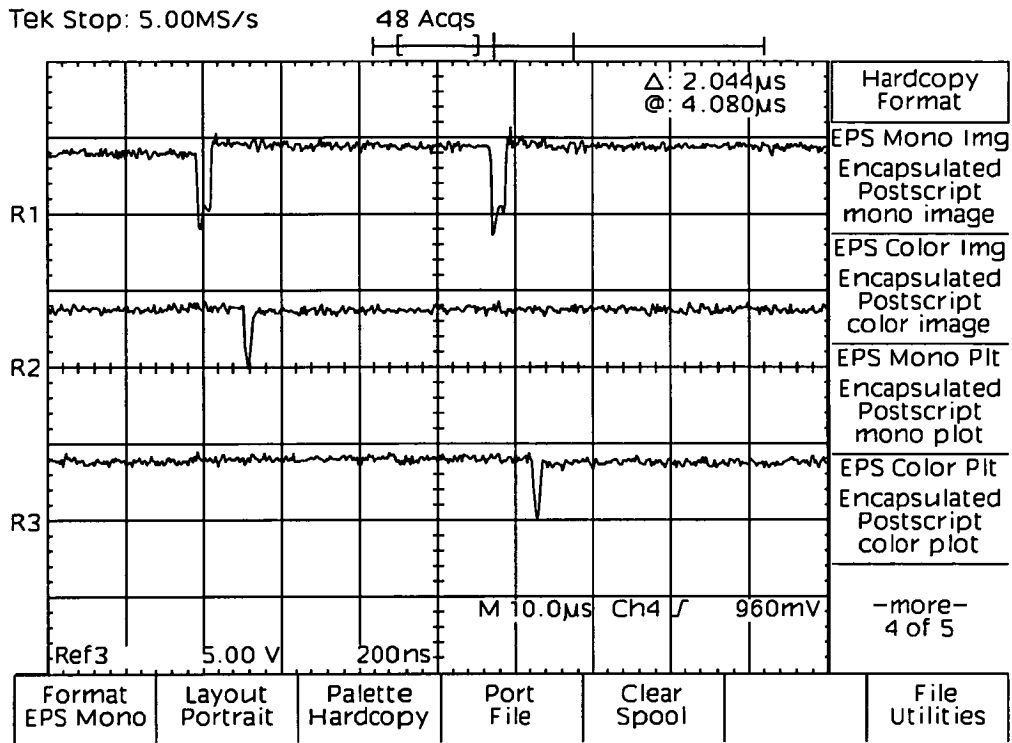


Figure 5.4: IB Write-in Cycle from Optical Link

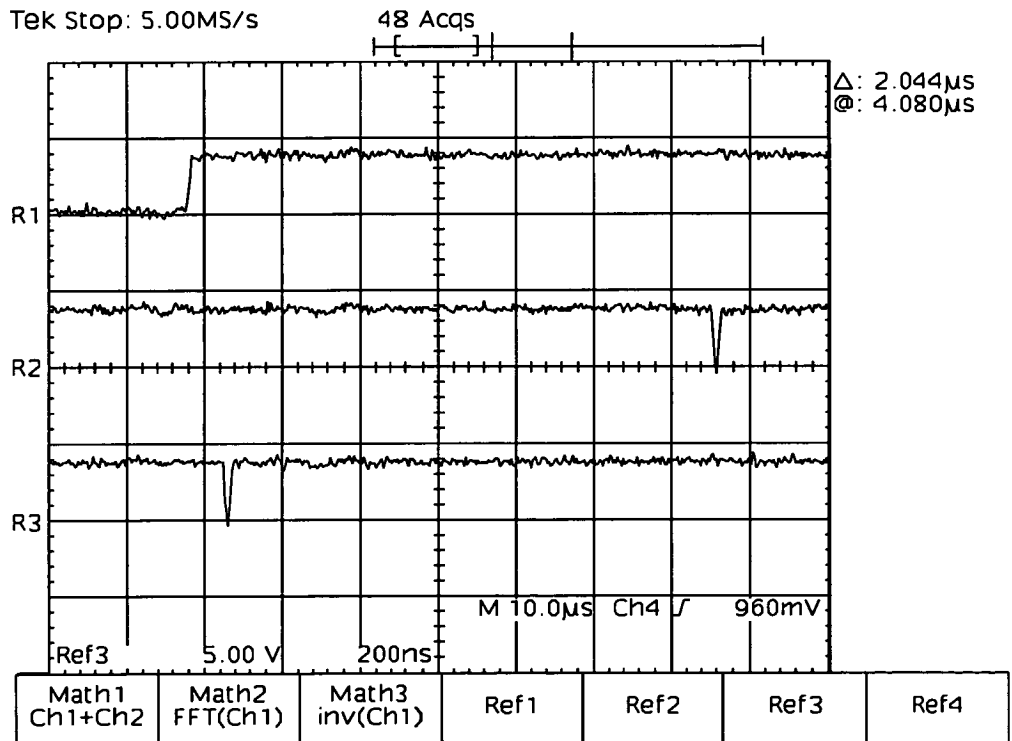


Figure 5.5: \overline{XOFF} de-asserted and transfer resumed.

5.2.2 R-Path Data Transfer

Start of Transfer

When the IB receives a 40 ns logic high pulse on the `TOKEN_IN` line it drives `DATA_AVAILABLE` low which in turn switches on the R-Path data transceivers. Figure 5.6 shows the start of a data transfer across the R-Path from an Input Buffer to the FOF. In Figure 5.6 the oscilloscope displays the following signals:

1. Channel 1 shows the `HAVE_TOKEN` line going high, turning on the IBs data transceivers.
2. `DATA_VALID` goes goes high 10 ns after the data is placed on the bus. The FOF uses the rising edge of this signal to latch the data into its FIFOs
3. Data line D0 toggles every cycle. The data transmitted in this instance was a 'chequerboard' pattern (i.e. `aaaaaaaa, 55555555, aaaaaaaaa, ...`).

End of Transfer

At the end of transfer the end of event word with bit D39 set high is transmitted to the FOF. After this the 20-bit word count is transmitted and the token passed on to the next IB. Figure 5.7 shows the following signals at the end of the cycle.

1. Channel 1 shows the D39 line going high as an End of Block (EOB) word is transmitted to the R-Path.
2. `DATA_VALID` latches the EOB word at the end of the data block, then the IB word count approximately 250 ns later.
3. `TOKEN_OUT` goes high for 40 ns after the word count has been transmitted. This is connected to the `TOKEN_IN` input of the next IB in line or, if this is the last IB, to the FOF

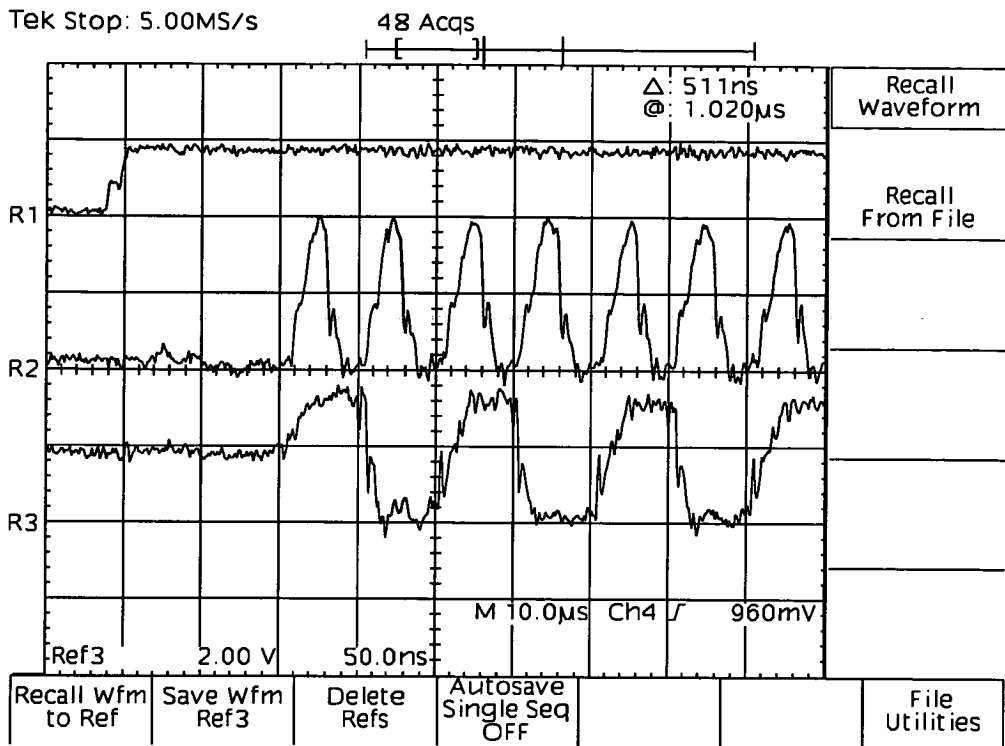


Figure 5.6: Start of transfer of Chequer Board pattern from IB to FOF.

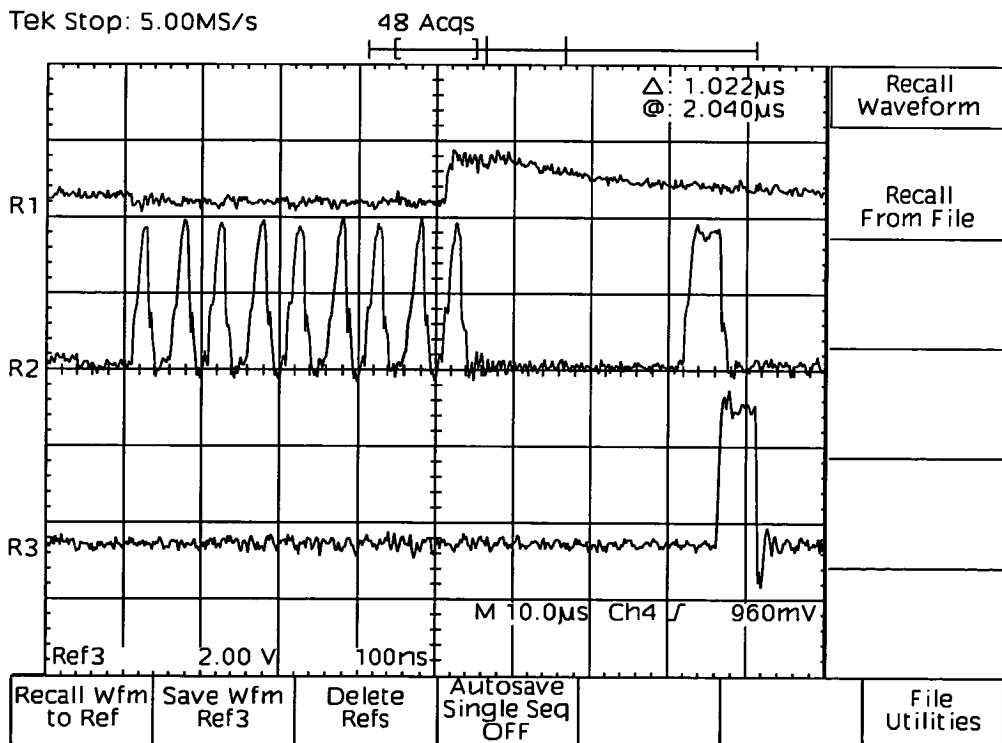


Figure 5.7: End of transfer from IB to FOF.

Transfer interrupted by $\overline{\text{XOFF}}$

When the FOF asserts $\overline{\text{XOFF}}$ the IB must finish the transfer that is in progress and then stop. This indicates that the FOFs buffers are full. After the FOF has cleared room for at least one event it de-asserts $\overline{\text{XOFF}}$ and the IB resumes transmitting data.

R-Path Protocol Signals

In addition to generating the DATA_VALID strobe and monitoring the $\overline{\text{XOFF}}$ line, each IB drives $\overline{\text{RP_NODATA}}$ and RP_HAVEDATA. These are wired-OR lines on the backplane which inform the FOF on the state of the IBs buffers. These signals are held low until every IB de-asserts its input, when this happens the lines float to 2.1 V.

When $\overline{\text{RP_NODATA}}$ is high all the IBs contain at least one event, i.e. each IB de-asserts its input when the first EOB arrives from the OL and re-asserts it again when the number of events read out over R-Path is equal to the number of events read in over the Optical Link.

RP_HAVEDATA warns the FOF that one or more IBs contain data to be read out even if a full event is not present in any of them. The IB asserts its input on the falling edge of the first strobe from the Optical Link and de-asserts it when the last data word is read out.

Figure 5.8 shows the IB asserting RP_HAVEDATA at the start of a read in from the OL and de-asserting $\overline{\text{RP_NODATA}}$ at the end. The oscilloscope traces show:

1. $\overline{\text{RP_NODATA}}$ goes high² at the end of the cycle, i.e. when one event is present in memory.

²The logic level on the board is the inverse of that on the backplane. As the BTL chips have open collector outputs the lines on the backplane are driven low when the inputs to the transceivers are high.

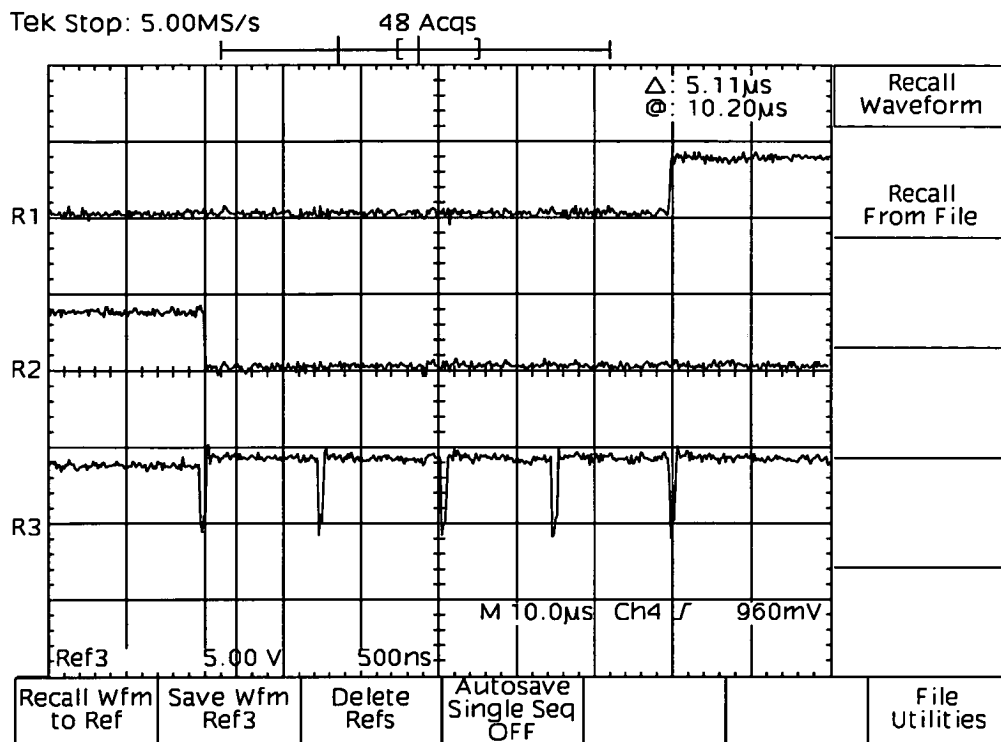


Figure 5.8: $\overline{RP_HAVEDATA}$ and $\overline{RP_NODATA}$ at the start of a data transfer to an empty IB.

2. $\overline{RP_HAVEDATA}$ goes low on the IB when the first data word is input.
3. STRB from the OL. 5 data words are transmitted, the last an EOB.

When the IB has emptied both $\overline{RP_NODATA}$ and $\overline{RP_HAVEDATA}$ return to their quiescent state. $\overline{RP_NODATA}$ goes low and $\overline{RP_HAVEDATA}$ goes high when the last word is clocked out of the IB.

5.2.3 Summary

These tests show that the Input Buffer is capable of accepting data from an Optical Link, buffering it until readout can take place and then transmitting it to the FOF over the R-Path backplane at speeds of around 100 MBytes/sec. At the end of the test processes described in this Chapter the production series design is finalized and the system as it stands is ready for full integration into the dataflow.

In the next Chapter the dataflow architecture is compared with other possible data acquisition methods.

Chapter 6

Alternative Solutions to the Dataflow Problem

Simply stated, the main task of the dataflow system is to merge data from N sources (N sub-detector elements) into M destinations (M data processors). In this Chapter alternative ways of achieving data transfers between multiple sources and destinations are discussed. The survey of different approaches to data gathering contained within this Chapter is not exhaustive, it is included to illustrate the fact that there are other approaches to data acquisition implementation.

6.1 Dataflow Architecture

Figure 6.1 shows the three basic types of dataflow architecture. These are the bus, the crossbar switch and the ring based methods.

6.1.1 Bus-based Architectures

Most HEP experiments currently use bus-based dataflow systems to collect data. In the past, event assembly was under the control of the data acquisition computer. When an event occurred an interrupt was sent from the trigger to the computer which then read the data in the sub-detector modules, usually over the CAMAC bus.

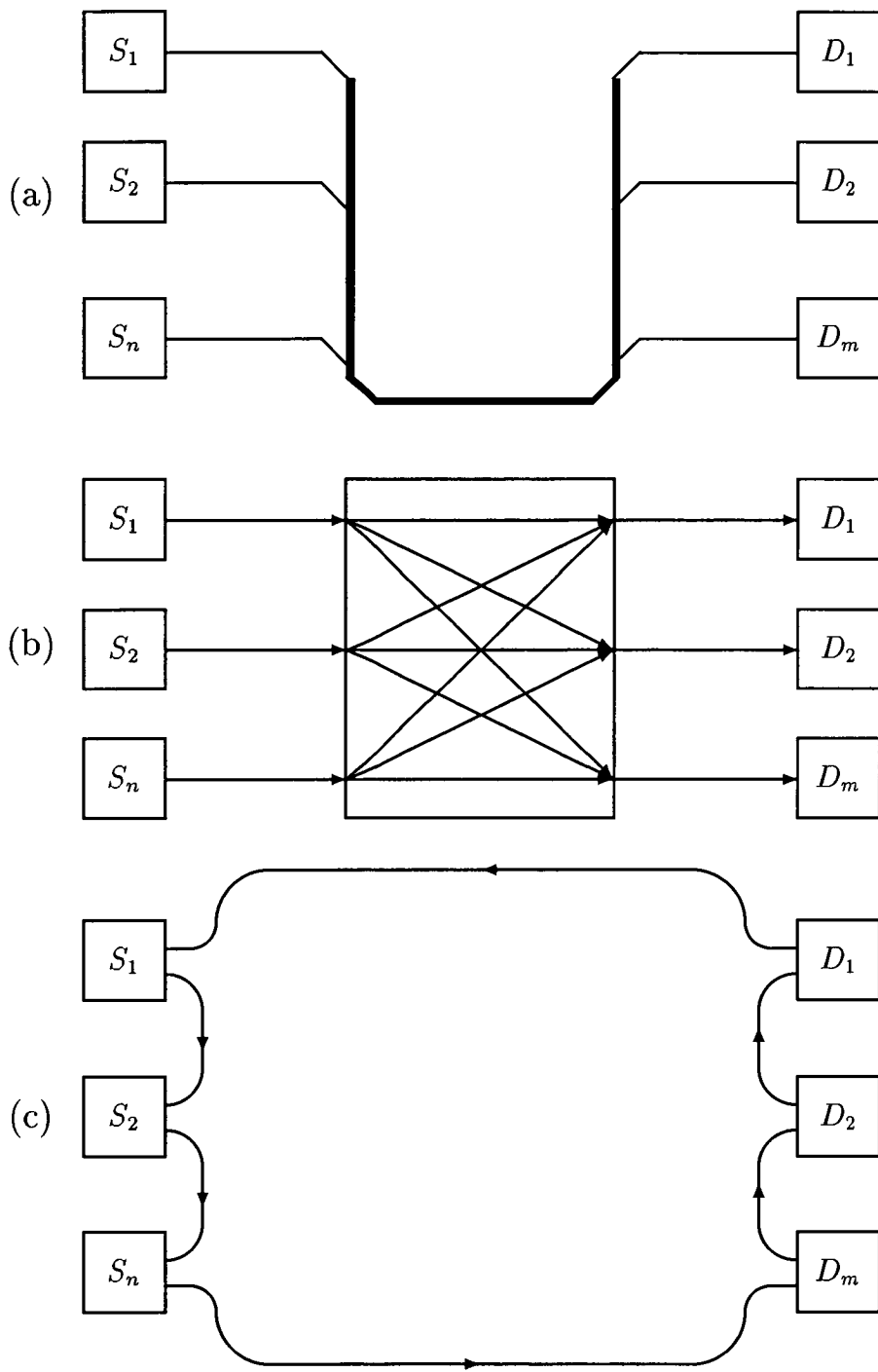


Figure 6.1: Data Acquisition Structures (a) Bus (b) Crossbar switch (c) Ring.

More recently, event builders installed in the data acquisition bus have been used to perform the data merging. The event builder can run a minimal operating system which enables rapid responses to interrupts, and in addition, free the data acquisition computer from the task of constructing the events.

The problem with bus-based architectures is that they are not scalable. The bandwidth of the bus limits the data rate of the system. Added to this the protocol overhead on busses such as VME and FASTbus introduces deadtime into the system. The time needed to gain bus mastership and to address and scan each source of data is of the order of μ seconds. As the number of sources increases so does the length of deadtime.

The problems with bus-based systems have led some designers to abandon them in favour of multiple, independent point-to-point links arranged in either crossbar or ring structures.

6.1.2 Switching Network Based Architectures

The crossbar switch scheme illustrated in Figure 6.1(b) routes data from the N sources to the M destinations via multiple, interconnected point-to-point links. The path the data takes is defined by a destination identifier in the data stream.

Point-to-point link schemes have several advantages over the more traditional methods[28]. Firstly the system bandwidth is limited only by the number of links present, as the capacity of the system is reached more links can be added to handle the extra data. Secondly point-to-point links have a very low protocol overhead, for example the data on the NA48 optical links is transferred with only a strobe signal. Thirdly they are typically less expensive than bus-based systems. The RD31 project at CERN is currently investigating such a scheme, using Scalable Asynchronous Transfer Mode (ATM) technology, for future experiments.

Figure 6.1.2 shows an example of the type of architecture that RD31 is investigating. The data is transferred from multiple sources (e.g. sub-detector elements) to multiple destinations (e.g. workstations in a processor farm).

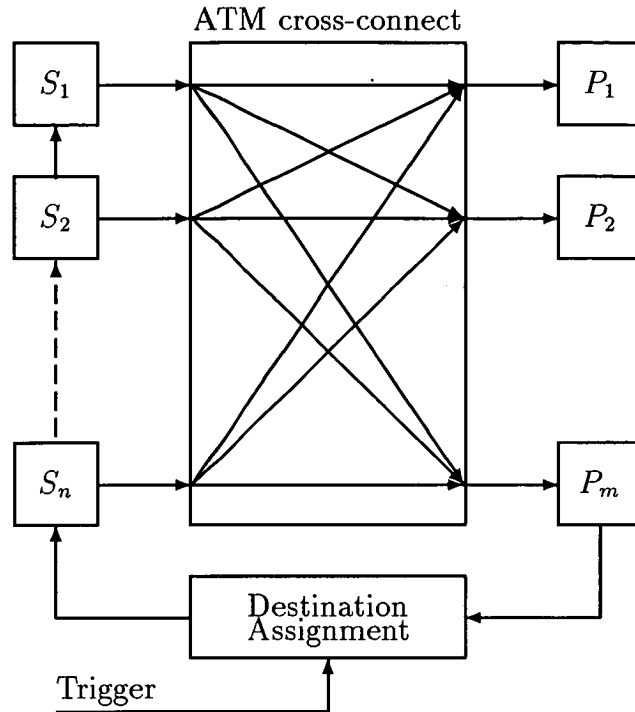


Figure 6.2: ATM Switching Architecture

ATM data is segmented into short, fixed length cells each consisting of 48-bytes of data accompanied by a 5-byte header. A 24-bit label in the header identifies which logic connection the cell belongs to. Each source uses M different labels to identify the connections to the M destinations. When a trigger is sent to the system the destination assignment logic generates and broadcasts the information used by the sources to select the appropriate connection for the event.

The use of the ATM links removes the bus bandwidth constraint on the system. If capacity is reached then more links can be added. They are, however, expensive and their use in NA48 is ruled out for this reason. However, their use in broadband telecommunications networks may drive the cost down in time for their use in the next generation of data acquisition systems.

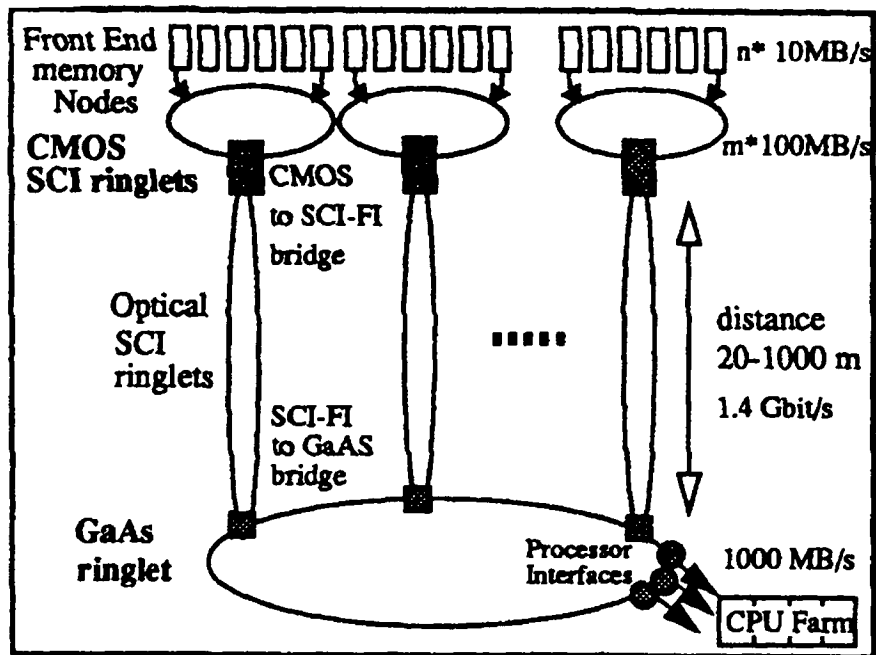


Figure 6.3: A Multi-Ringlet SCI Data Merger

6.1.3 Ring Based Architectures

In ring-based systems each device is connected to its nearest neighbours by point-to-point links. Data packets arriving at each node are either processed or passed onto the next device in line depending on the value of a node identifier in the data stream. Figure 6.3 shows an example of such a system using the Scalable Coherent Interface (SCI) technology.

SCI provides bus like features between SCI nodes in a ringlet. Uni-directional point-to-point links interconnect the inputs and outputs of the nodes in the network. The data is transmitted in the form of packets consisting of a header address (16-bit node identifier and 48-bit internal node address) followed by 16,64 or 256 bytes of data and a CRC trailer. The data packets transmitted to the node are directed either to the output link or to an input FIFO. Packets generated by the user logic at the node are queued until no data is in a by-pass FIFO and then sent.

The raw bandwidth per link is around 1 Gbytes/sec along differential ECL cable[30], but this is reduced by the ratio of packet overhead to packet data. However, two or more nodes in a ringlet may be receiving and transmitting data at the same time, giving a overall system bandwidth higher than the bandwidth per link.

The distance between nodes is limited to a few tens of meters. For the longer distances typical in HEP experiments the ringlets can be interfaced to long distance fibre optic links, operating at a reduced rate of 1.4 Gbits/sec, through bridges. By doing this networks can be built for HEP applications where data must be transmitted across hundreds of meters, (in the case of NA48, 200 meters from the detector to the data merger electronics).

The RD24 collaboration at CERN[31] is studying applications of the SCI standard for the Large Hadron Collider. To date they have tested a two-node SCI ringlet based on a R3000 RISC processor and a DMA node on a MC68040 processor bus. In these tests the DMA node achieved a data rate exceeding 100 Mbytes/sec.

This technology is still new and in some respects not ideally suited to HEP experiments. The small packet length is not ideal for NA48 where the sub-detectors transmit 100s of Kbytes per event. In addition, the bandwidth is significantly reduced when each module in a ringlet is transmitting to the same destination, a situation that is common in HEP.

6.1.4 The NA48 Dataflow Architecture

The NA48 dataflow is a hybrid of the bus-based and switching network-based designs. The optical links and the HIPPI links and switch give the data acquisition scalability despite the fact that the data from the sub-detectors is merged across a backplane bus.

The R-Path bus is different from commercially available systems such as VME in two important respects:

- The R-Path operates a very simple protocol. Data transfers are accompanied

Clock Frequency	40.08 MHz
Maximum average level 1 trigger rate	100 kHz
Level 1 pipeline length	$>2.5 \mu\text{s}$
Raw data readout time	To introduce deadtime $\leq 1\%$

Table 6.1: Critical ATLAS front end DAQ parameters

only by a strobe signal (`DATA_VALID`), which the FOF uses to latch the data into its buffers. The FOF returns a `DATA_ACK` pulse, but this is not a true handshake as it is simply the `DATA_VALID` returned. The lack of a complex handshaking protocol means that deadtime is not introduced to the dataflow.

- Each module does not have to request bus mastership. The token passing scheme cuts out the bus arbitration processes common in commercial bus systems. These can take of the order $5 \mu\text{seconds}$ whereas the token is passed between modules within nanoseconds.

These features reduce the deadtime introduced by the Data Merger to almost zero¹

6.2 Dataflow Solutions from Other Experiments

6.2.1 The Atlas DAQ Scheme

The ATLAS[33] experiment at the CERN LHC will start early next century and is currently discussing a hybrid dataflow scheme with some similarities to the one employed by NA48. The main parameters that the system will be designed to satisfy are given in Table 6.1.

The high trigger rate and the need for a system which does not introduce a great

¹The time taken to pass the token between modules and the padding words added by the FOF introduce a small amount of deadtime.

deal of deadtime has led the experiments trigger/daq steering group to consider an architecture that is based on point-to-point links and on-line data merging.

Data from each subdetector is firstly presented to the level 1 trigger pipeline which connects the subdetector to a ‘derandomizer’ where the data is stored before being sent to a front-end link. The link sends the data to a read-out driver module where data from different parts of the subdetector is merged and then transferred to a readout buffer over another point-to-point link. The readout buffer stores the data and performs error detection and recovery, local pre-processing for level 2 data and the extraction of data for level 2 and level 3.

The event building will be based on a high-speed switching network that interconnects many data sources (readout buffers) and data destinations (level 3 processing units). The ATLAS group expects that switching systems with the required performance will be available by the turn of the century and therefore the event builder could be based on an industry standard protocol. This would enable the experiment to take advantage of advances in the industry supported hardware.

6.2.2 The DART DAQ Scheme

Other current experiments have faced similar dataflow problems to NA48. For example, KTeV, based at Fermilab in the United States, is another experiment that hopes to measure $\frac{e'}{e}$. The data acquisition problems that were faced in this experiment were very similar to those faced by NA48. The solution, the DART data acquisition system, uses VME backplanes to form the hub of the data acquisition system [32].

The DART system architecture is similar to that of NA48 in that the read out of the sub-systems takes place in parallel and each sub-detector is read out independently of any other. The event building architecture is also scalable. Figure 6.4 shows a block diagram of the KTeV data acquisition system.

The data is transferred from the Front End Crates, via RS482 cable, to a bank of DDD modules. These modules provide for 40 MBytes/sec input to a 40 kByte dual

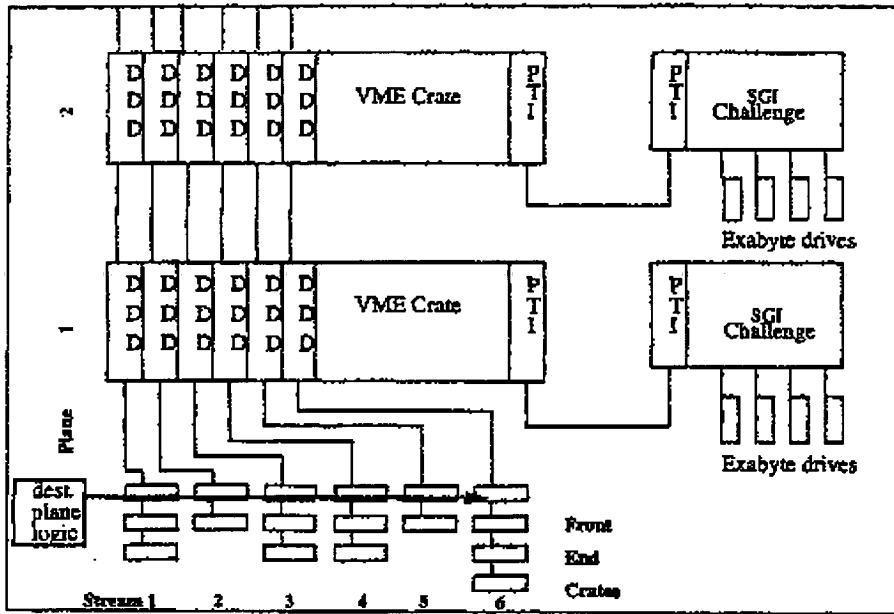


Figure 6.4: Block Diagram of the KTeV DA System

ported data FIFO memory and its output to a commercial 68040 processor board over VSB². An address word in the data stream selects one of the several VME crates that are linked in series with each other. From the crates the data is transferred to workstations before selected events are written to tape. Communication between workstations takes place over Ethernet.

DART operates at a maximum data rate that is dependent on the number of VME crates in the system. The backplanes act as parallel event builders to deliver the maximum data throughput (160 MBytes/sec. in the KTeV experiment). This solution passes the responsibility for assembling the events to the SGI Challenge multi-processor machines. These map the data in the DDD modules into their own memory then select interesting events and write them to Exabyte storage units. The SGI Challenge machines effectively perform the data merging.

Both DART and the NA48 data acquisition system have their advantages and

²VSB is an extension of the VME protocol that uses the user defined pins on the J2 backplane as address and data lines

disadvantages. The DART system requires a large amount of CPU power to scale the data down before storage can take place. In NA48 the spill size is of the order of 256 MBytes whereas in KTeV up to 3 GBytes can be delivered in one spill. The small size of the spill in NA48 enables relatively inexpensive workstations to be used to store the data. In KTeV the workstations have to assemble the events and select interesting ones to be stored, which requires a large amount of CPU power. DART, however, uses commercially available hardware whereas NA48 has to use many custom built modules.

Chapter 7

Conclusions

The performance of the whole dataflow system and its constituent parts has been tested both in the laboratory and during data taking runs. In this Chapter the conclusions that can be reached from these tests are discussed.

7.1 Dataflow

7.1.1 Initial Tests

During September 1994 the NA48 data acquisition system was shown to perform successfully in collecting data across 200 m from a reduced detector setup. An example of data from the 1994 run collected over VME is shown in Figure 7.1. The central diagram shows a hit in the hodoscope, i.e. the output of a photomultiplier tube when hit by a particle from a K_S decay. The third diagram is from the TDC and shows a flip flop that toggles when hit: this is a high to low transition. These two plots show that the K_S decays to $\pi^+\pi^-$, one of the pions giving the outputs from the hodoscope and TDC. The fact that it is K_S particle that has decayed is shown by the hit in the tagger (200 m away from the detector) shown in the first plot. All of these plots share the same time stamp, showing that they are generated from the same decay.

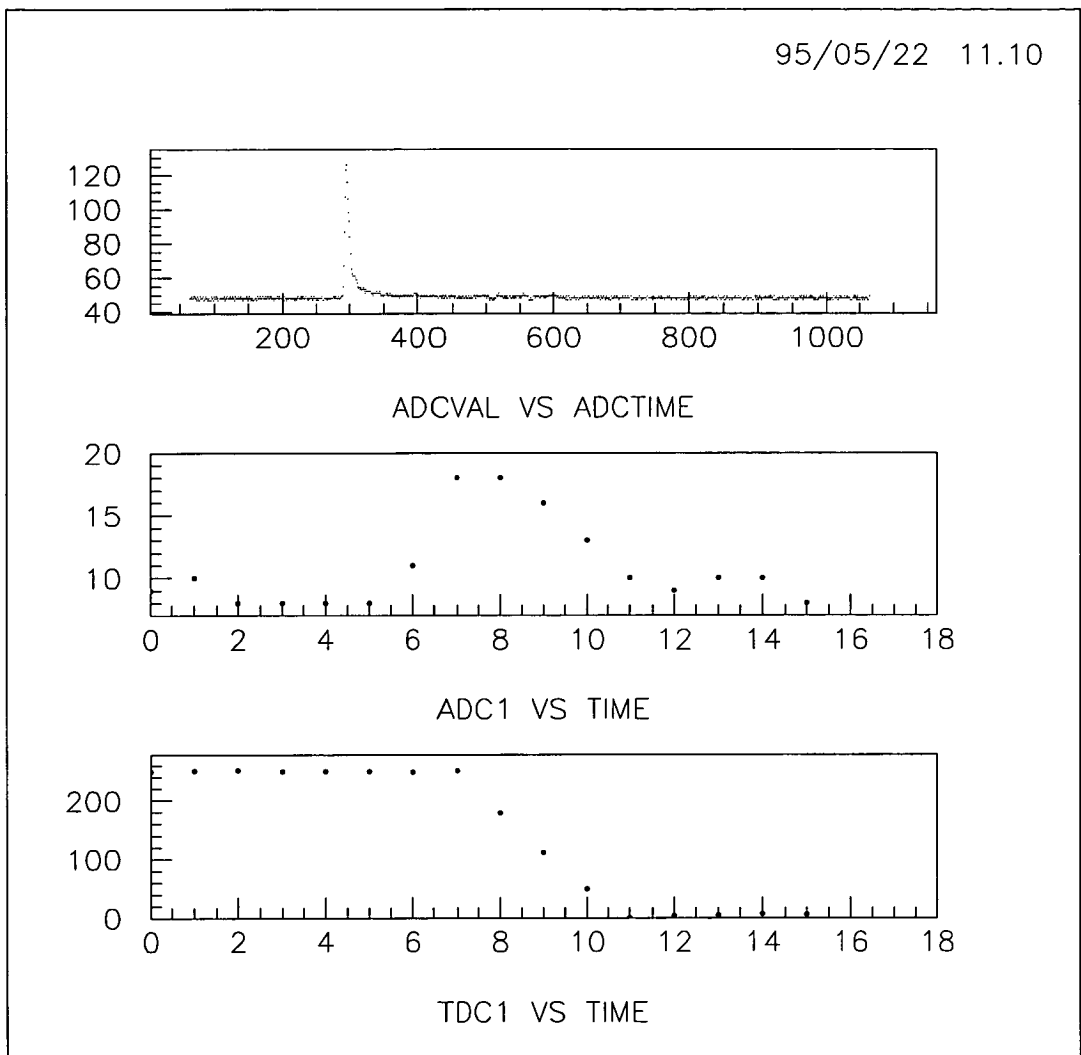


Figure 7.1: A plot from the 1994 run showing a hit in the hodoscope together with a hit in the tagger.

The control of the system through VME proved functional. The Data Merger harness program supervised the system during the data taking run in 1994. The full run control program will be in place for next year's beam time.

The general architecture of the dataflow system was also shown to be suitable. The scalability required is provided by the point-to-point links. In addition, the buffer-to-FIFO architecture enabled each component of the dataflow to be developed and tested independently. This year's enlargement of the detector will result in modification to the number of optical links and workstations used but the general architecture will remain unchanged. The simulation results detailed in Appendix G show that next year, when the LKr Calorimeter will be present, the dataflow should function at full capacity.

7.1.2 Further Laboratory Tests

The central part of the Data Merger, the R-Path, was shown to perform sustained data transfers at the required 100 MBytes/sec. The R-Path was first tested and shown to perform correctly with the FOFETTE before the FOF was delivered to CERN in March 1995. The IB and the FOF proved capable of interfacing at the required speed and the full R-Path protocol is to be tested during the next data taking run.

In the March 1995 tests the FOF was shown to be able to read in data from the R-Path and format it into HIPPI before delivering it to a HIPPI test box. In August 1995 the full data acquisition system will be in place. This system will send data from the sub-detectors to the front end workstations via the HIPPI switch in the August data run.

7.2 Input Buffer

The prototype used in September 1994 was developed to test the VME section, memory function and optical link interface, as well as the support for dual chan-

nels. The pre-production board was swapped into the system in place of the prototype for laboratory testing from October 1994 to March 1995. This board was used to debug the R-Path and the token passing mechanism.

The major IB requirement, the buffering and transfer of data from the optical link to the R-Path at 100 MBytes/sec, was achieved.

The IB, the design and testing of which is described in this thesis:

- satisfies the design criteria described in Section 4.2
- is modular
- contains adequate diagnostic resources
- is flexible, this flexibility being shown by the frequent re-routing of the Xilinx during the debugging phase to solve problems on the board.
- performs reliably at 25 MHz.
- and was produced within the experiment's budget.

7.3 Concluding Remarks

In this thesis the design and development of an advanced data acquisition architecture has been described. At the centre of this architecture is the data merger. In the data merger the required 100 MBytes/sec data rate is achieved on the R-Path backplane using BTL technology with a very light protocol.

The basic 'buffer-to-FIFO' architecture that is described here ensures that the system is flexible and scalable up to the bandwidth of the R-Path. The input stage of the data merger, the input buffer and the output stage, the FIFO output formatter are required for the system to operate at full capacity.

The dataflow has been shown to work in laboratory tests carried out in the early part of 1995. Simulation results show that when the experimental apparatus is complete the data will be collected at the required rate.

The architecture compares well with other modern data acquisition systems. Other collaborations such as KTeV have achieved similar results using different technologies but at the cost of expensive CPUs. The hybrid structure of the NA48 data acquisition combines speed and economy. The experiment will take data using the system described in this thesis with a subset of the detector in August 1995 and with the full detector in 1996.

Appendix A

VME Operation

Figure A.1 shows the VME interface to the IB.

The address on the bus is latched into the IB on the falling edge of the AS* signal. The board address is set by 8 switches which set the P inputs to the 74LS520 comparator, the Q inputs being provided by the address that the FIC asserts on the VMEbus (A24 – A31). If the FIC addresses the board and provides the correct address modifier codes (AM0 – AM5) then the ADEN* input on the VME2000 chip is de-asserted. The VME2000 (a commercial chip from PLX technology) then drives MODSEL* low to indicate that the IB is selected. This signal activates either the data receivers or the transmitters depending on whether a write or read cycle is specified. The signal WRITE* is low during a write cycle and high during a read. The addresses A21 - A23 are input to the Xilinx chip of each channel where they are decoded to provide access to either the data memory (32-bit), the control memory (8-bit) or the channel control and status registers (CSRs). The addressing scheme is detailed in Table A.1 where xx is the board address. The internal Xilinx configuration is discussed in Appendix F.

The output from the data and control memories of each channel are connected together and are routed to the output side of the VME section on lines R(0-39). As the VME data bus is only 32 bits wide, the 8-bit control memories (R32-R39) are multiplexed with the top 8 bits of the data memories (R24-R31). The control memories data is output to the VME bus lines D24-D31 when A22 is high in the VME address. If A22 is low then the data memories output R0-R31 is sent to the

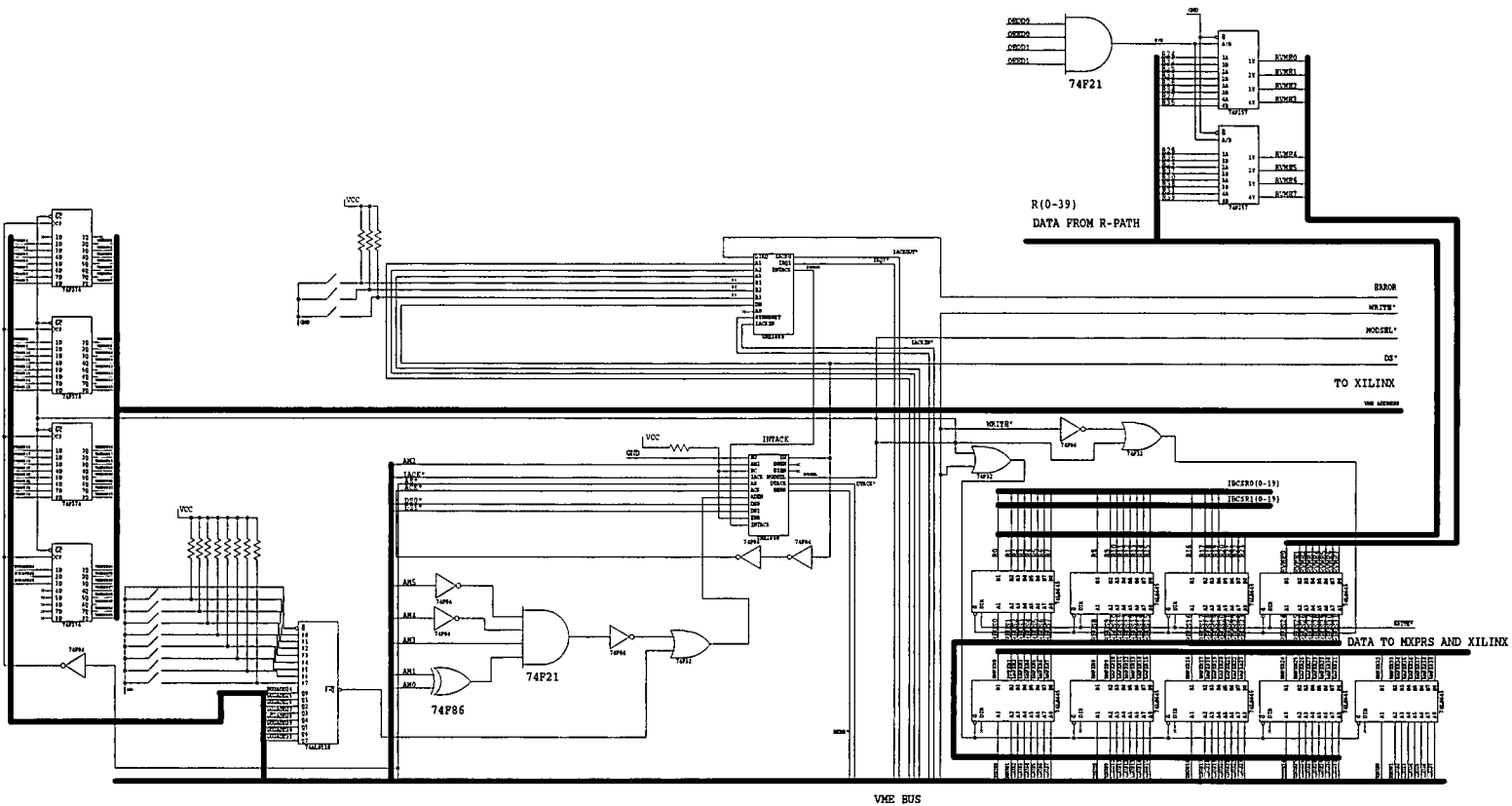


Figure A.1: IB VME Interface

Area of Board	VME Address Range	AM0-5
Channel 0 CSR	\$xx000000 - \$xx00000c	\$09,\$0D
Channel 0 Data Memory	\$xx400000 - \$xx5ffffc	\$09,\$0D
Channel 0 Control Memory	\$xx600000 - \$xx7ffffc	\$09,\$0D
Channel 1 CSR	\$xx800000 - \$xx80000c	\$09,\$0D
Channel 1 Data Memory	\$xxc00000 - \$xxdffffc	\$09,\$0D
Channel 1 Control Memory	\$xxe00000 - \$xxffffffc	\$09,\$0D

Table A.1: VME Addressing Scheme

		Channel 0 / Channel 1	
Control	Register	\$xx000000 / \$xx800000	Write/Read
	Bit 0	Reserved	
	Bit 1	Reset	
	Bit 2	Source On	
	Bit 3	Reset OL	
	Bit 4	$\overline{\text{XOFF}}$ to OL	
20-bit Word Count		\$xx000004 / \$xx800004	Read Only
20-bit Event Count		\$xx000008 / \$xx800008	Read Only

Table A.2: Input Buffer Control and Status Registers

bus.

When writing to the board the VME data lines D24-D31 are sent to the control memories while the data memories are sent the data on lines D0-D31.

The VME data lines D0-D19 are also sent to each channels CSR lines. There are three sets of registers associated with each channel, these are described in Table A.2.

The control register bit Source On sets up the board to write and read from VME (SO = 1) or to read the OL and write to the R-Path (SO = 0). Bit 4 forces the IB to send $\overline{\text{XOFF}}$ whether it is full or not, thus stopping any data transfer from the OL. For simplicity the IB accepts only single D32/A32 transfers. The VME

interface was originally planned solely for testing and so it was felt that it was not necessary to implement block transfers or D64 operation.

Appendix B

Optical Link Interface

Data from the optical link arrives at each channel on a 40-bit wide bus as shown below.



The 40-bit word is divided into a 32-bit data word D0-D31, 5 parity bits (P0-P4) and the D39 bit which, when high, indicates an end of event. The OL and IB also use the protocol signals listed in Table B.1.

The data from the OL is first multiplexed with the VMEbus data lines (see Figure B.1) under the control of SO (CSR bit 3). The data is then fanned out to two sets of buffers (odd and even) which are latched by \overline{LEE} and \overline{LEO} from the Xilinx. On the odd and even banks the 40-bit wide buffer outputs are split and fed to a 32-bit wide SRAM (the data word RAM) and an 8-bit wide SRAM (the control word RAM). The two RAMs are written to as a 40-bit wide memory i.e. they share the same address (00A0-18 and 0EA0-18) and write enable signal (\overline{WEO} and \overline{WEE}). The Figure shows the channel 0 structure, channel 1 is identical apart from the names given to the nets, e.g. 0EA0-18 is replaced by 1EA0-18 in channel 1.

The protocol signals listed in Table B.1 are routed to the Xilinx chip. \overline{XOFF} , RT

STRB	The data is valid on the low going edge of the STRB. Data is latched into the memories using this signal.
PERR	Parity Error. The OL checks the incoming data parity
RDERR	Running Disparity Error. A high on this line indicates that there has been an error in the 8 to 10-bit encoding.
SERR	Sequence Error. Indicates that a word has been lost. The OL toggles D38 on each word sent, if two consecutive D38s are the same the sequence error bit is set high
LINKST	Link Status.
LASFLT	Laser Fault.
LINK_UP	Link is up and running.
XOFF	An IB to OL signal which inhibits further data transfer.
REBOOT	The OL reboots if the IB drives this line high. This is connected to bit 3 in the IB control register
RT	OL Reset.

Table B.1: OL protocol signals

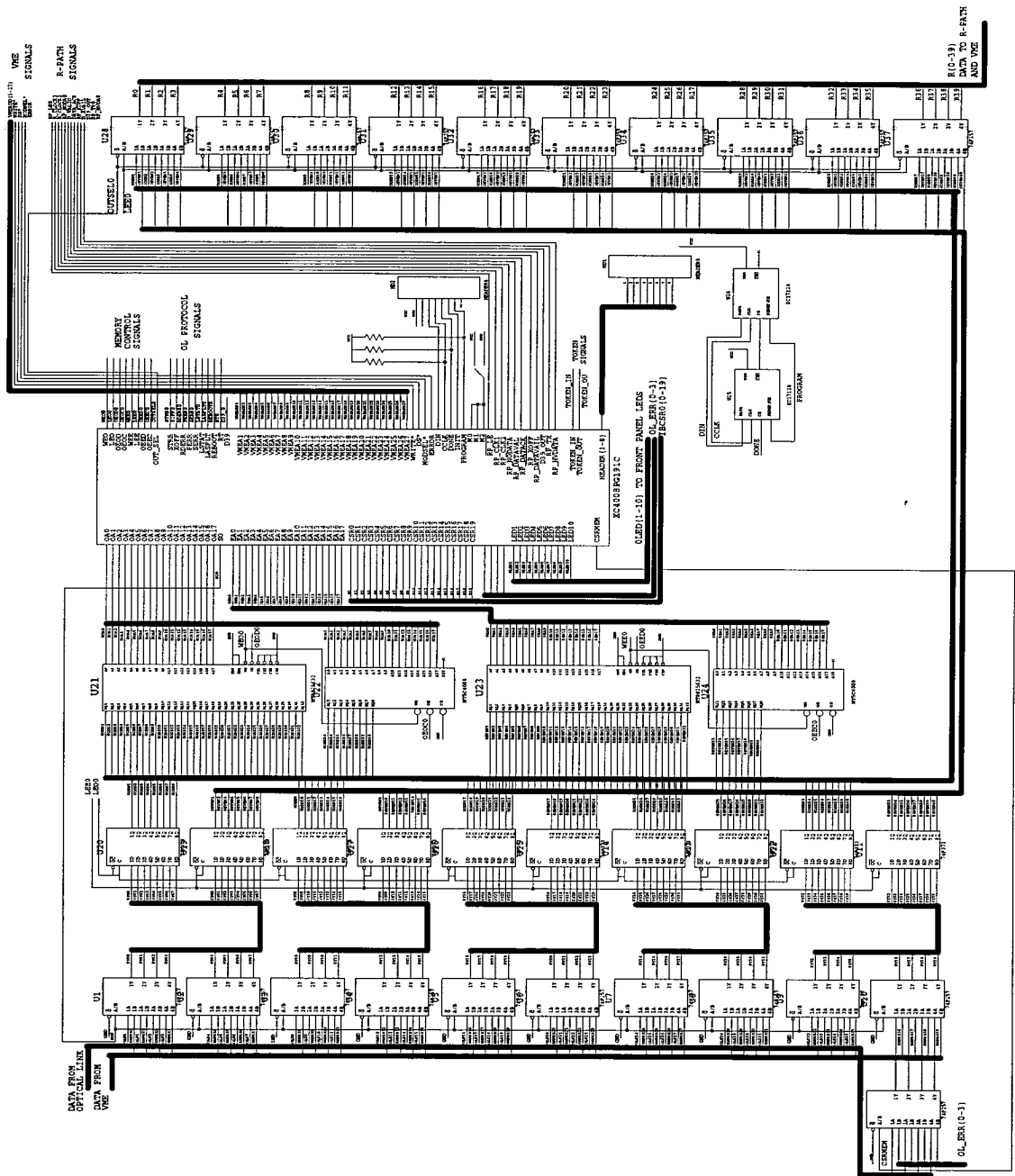


Figure B.1: Channel Memory Structure

and REBOOT are generated in the chip and the error signals PERR, RDERR and SERR (OL_ERR(0-2) in Figure B.1) are latched and written into the control memory when CSRMEM is set high. This happens on the last word of each event. This means that when the data is output to the FOF the OL error bits are transferred in the end of event word.

The STRB signal is used to generate the write and latch enable signals.

Appendix C

R-Path Interface

The R-Path interface is common to both channels. The channel that has the token at any given time has control of the output bus. Figure C.1 shows the output side of the board.

The 32 data lines plus parity are routed to 4 DS3886 Futurebus+ transceiver chips. The final parity bit P4 and the rest of the protocol lines are transmitted to the R-Path via a DS3883 chip, a non-latching equivalent of the DS3886. The two clock signals, RP_CLOCK1 and RP_CLOCK2, and \overline{XOFF} are inputs to the IB and are fed through a receiving DS3883. The latch signal RP_LE and the IB generated protocol signals are derived in the FPGAs.

When the token (a 40 ns long high pulse) arrives on the Channel TOKEN_IN line the channel drives RP_TX low only if it holds data. If it is empty then the token is passed on to the next channel. RP_TX turns on the R-Path data transceivers. The circuitry in the Xilinx increments the addresses to the memories and sends output enables to the odd and even banks in turn. The data from the odd and even memories is multiplexed together to form a 40-bit bus at the R-Path interface. The multiplexers are activated by OUT_SELECT and switched by \overline{LEE} , which is the same signal that switches on the even input latches. When it is low the even memory can be written and the odd memory read, when it is high the opposite is the case.

The data is latched out of the IB on the rising edge of RP_LE until the D39 line at

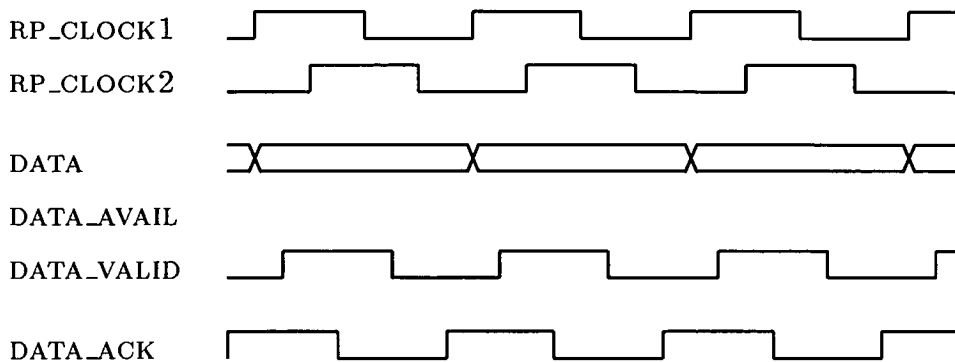


Figure C.2: R-Path Data Transfer

the IB output goes high. This indicates that the last word of the event is being sent. The channel then stops incrementing the memory addresses and prevents any further output from the multiplexers. The Xilinx then increments its internal word count and places it on the bus together with a RP_LE. The TOKEN_OUT line is then driven high for 40ns (sending the token) and HAVE_TOKEN is de-asserted.

Each word that is placed on the bus is accompanied by a DATA_VALID pulse derived in the Xilinx. The FOF uses this signal to latch the data from the IB into its buffers. This signal goes high $\frac{1}{8}$ th of a clock cycle after the data is latched onto the bus. The FOF sends back to the IB a data acknowledge pulse (DATA_ACK) for each word it receives. This is not a true handshake protocol as the FOF simply returns the DATA_VALID pulse to the IB. A full handshake would require the IB to wait for the DATA_ACK to arrive before sending the next word, which would introduce deadtime. However, the DATA_ACK does tell the IB that the FOF is present in the system and is receiving data. The timing diagram in Figure C.2 shows the relationship between DATA_VALID and the transmitted data at the output of the IB. The timing of the DATA_ACK pulse is dependent on the position of the IB.

Appendix D

Token Handling

The token travels between IB channels along co-axial lemo cable. For maximum flexibility each channel has an input and an output lemo socket. This means that one channel on a board can be involved in data taking while the other is by-passed, i.e. not connected in the token chain. The token ring is based on differential ECL, due to speed and noise considerations, so each channel has an ECL to TTL converter (a MC10H125 chip) at the token input and a TTL to ECL converter (a MC10H124 chip) at the output. The token conversion circuit is shown in Figure D.1. The token_in line for channel 0 is generated from the ECL token pulse on the lemo1 connector, the token_out is sent out to the lemo2 socket. Channel 1 takes its token input from lemo3 and outputs on lemo4.

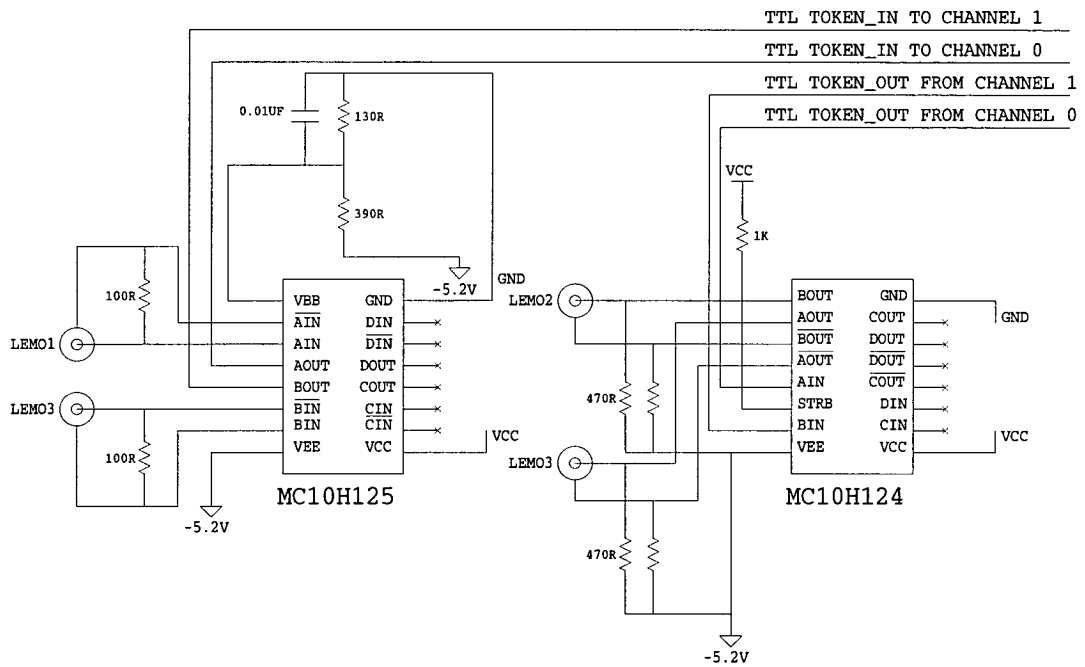


Figure D.1: Token Input/Output Circuit

Appendix E

Front Panel LEDs

Each Xilinx drives 10 LEDs at the front of the board. These, listed in Table E.1, show the status of the channel.

RED LEDs	
Error	High on VME bus error and when DATA_ACK is not received from FOF
Full	High when Channel memory is full
YELLOW LEDs	
Almost Full	High when Channel memory is nearly full ($\overline{\text{XOFF}}$ sent to OL)
GREEN LEDs	
Empty	High when Channel memory is empty
HAVE_TOKEN	Channel is transmitting to the R-Path
$\overline{\text{SO}}$	Source On. High when the channel is in Optical Link/R-Path mode. Low during VME operation
$\overline{\text{XOFF}}$	$\overline{\text{XOFF}}$ to the Optical Link
$\overline{\text{RP_XOFF}}$	$\overline{\text{XOFF}}$ from the FOF
RP_NODATA	High when channel does not contain a full event

Table E.1: IB Status LEDs

Appendix F

Xilinx Design

The top level schematic of the IB Xilinx design is shown in Figure F.1. This is split into 6 lower level schematics which contain the logic circuit itself.

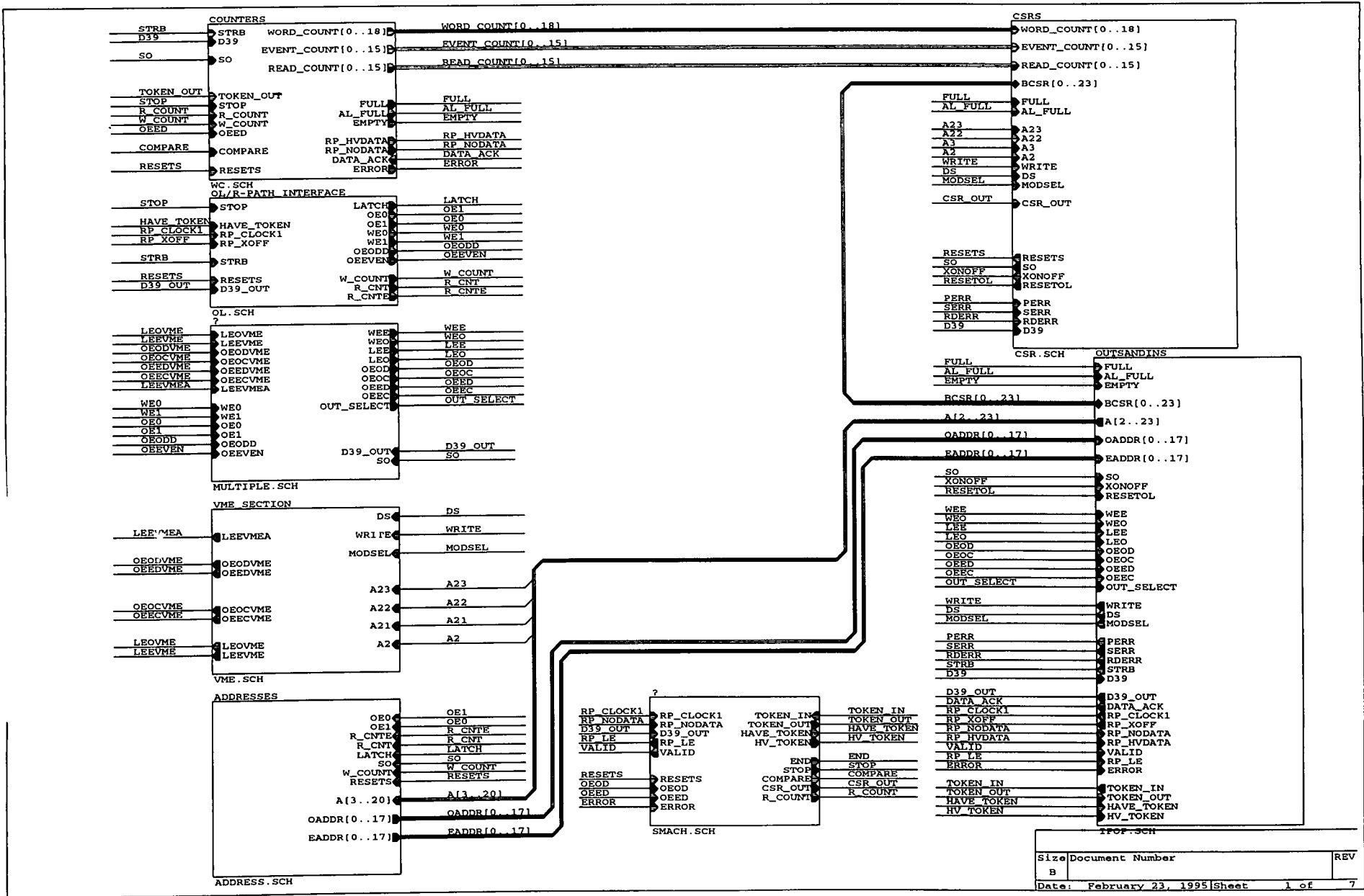
F.1 VME Interface

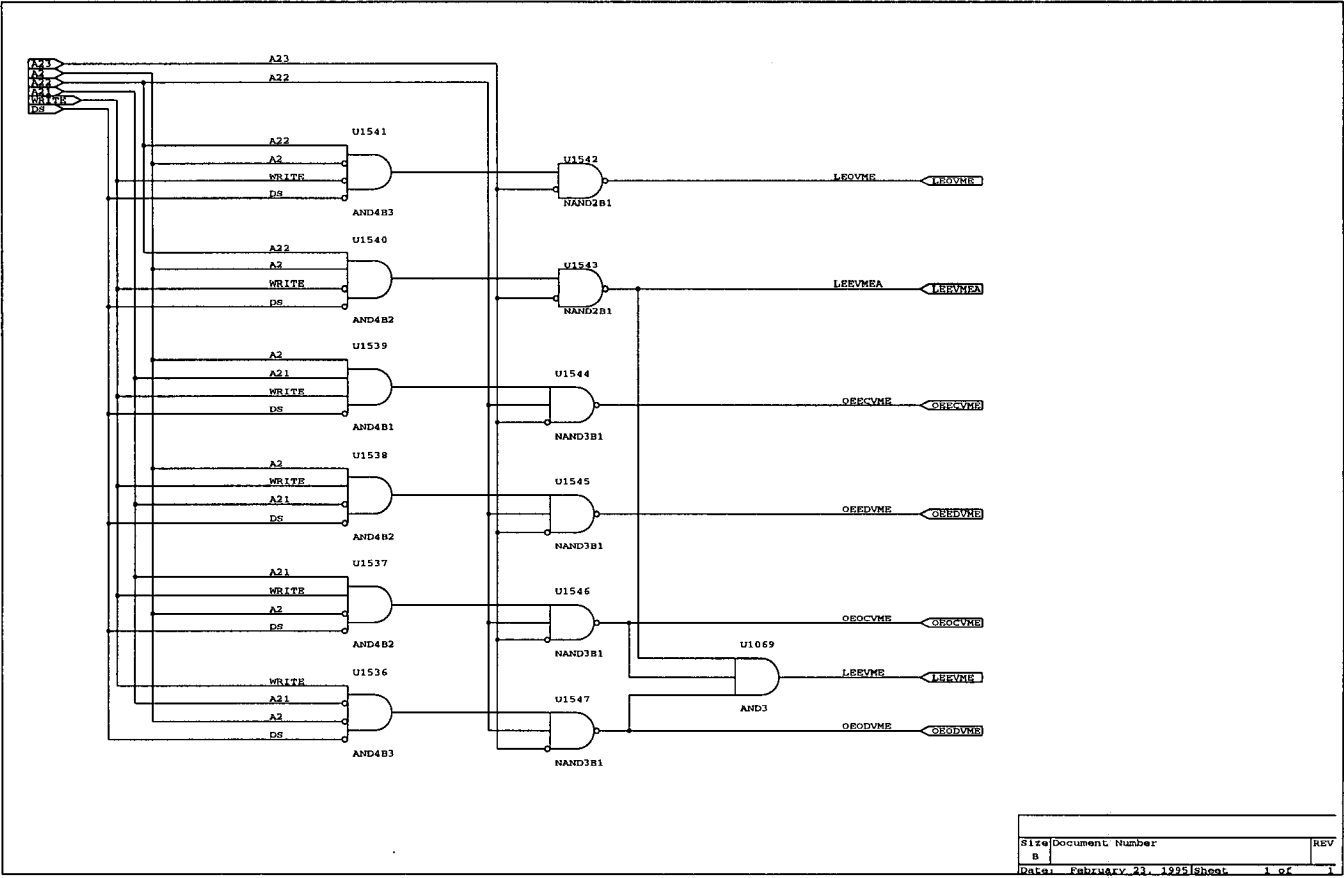
The VME addressing scheme discussed in section A is implemented as logic in the VME_Interface section of the schematic (Figure F.2). The VMEbus signal `WRITE*` along with `DS*` from the VME2000 chip and the addresses A21 - A23 and A2 are combined to produce output enables for the odd (data and control) and even (data and control) memories and the latch enables for the even and odd banks (common to data and control). The decoding scheme for channel 0 is shown in Table F.1.

The memory write enable signals (`WEOVME` and `WEEVME`) are generated directly from the latch enables (`LEOVME` and `LEEVMEA`). The inverter chains in Figure F.3 delay the low going edge of the latch pulses until the data has arrived at the memory. The latch and write pulses are multiplexed with the optical link and R-Path pulses and the data is then written in when `WEOVME` or `WEEVME` goes low.

The EVEN memory VME latch enable signal (`LEEVME`) controls the select input of the output multiplexers as well as the even latches. The latch enable goes low

Figure F.1: IB Xilinx Top Level Schematic





Size	Document Number	REV
B		
Date:	February 21, 1995	Sheet 1 of 1

Figure F.2: Xilinx VME decoding

Output Enable

	A23	A22	A21	A2	WRITE*	address
ODD memory data	0	0	1	0	1	\$400000-\$5ffffc (00 and 08)
EVEN memory data	0	0	1	1	1	\$400000-\$5ffffc (04 and 0c)
ODD memory control	0	1	1	0	1	\$600000-\$7ffffc (00 and 08)
EVEN memory control	0	1	1	1	1	\$600000-\$7ffffc (04 and 0c)

Latch Enable

	A23	A22	A21	A2	WRITE*	address
ODD memory	0	X	1	0	0	\$400000-\$7ffffc (00 and 08)
EVEN memory	0	X	1	1	0	\$400000-\$7ffffc (04 and 0c)
	0	X	1	0	1	\$400000-\$7ffffc (00 and 08)

Table F.1: Channel 0 VME Decoding Scheme

with the ODD memory output enable in order to switch the output to ODD. The signal that the EVEN write enable is derived from (LEEVMEA) does not take the ODD output into account.

The configuration file for channel 1 is identical but the line A23 is inverted before it reaches the Xilinx and so it only responds when A23 on the VMEbus is high.

The VME master addresses the IB CSRs by holding A22 and A21 low. Logic in the CSR_Section (Figure F.4) decodes WRITE* and the address to produce WRITE_SELECT and READ_SELECT pulses. The CSR lines (BCSR0 - 20) are defined as bi-directional. During a write cycle data placed on these lines is latched through the control register on the rising edge of the WRITE_SELECT pulse. On a read cycle the tri-state buffers are enabled on the falling edge of READ_SELECT. The data at the input of the buffers is dependent on the A2 and A3 VME address lines. These switch the buffers between three sources of data as shown below:

Address A3 A2 CSR Output

\$00	0	0	Control Register
\$04	0	1	Word Counter Output
\$08	1	0	Event Counter Output
\$0c	1	1	Reserved

The counter outputs are derived in the Counters section of the Xilinx (Figure F.5). The Word counter and the Event counter are clocked by the OL strobe and D39 respectively. In the Addresses section (Figure F.6) the VME addresses A3 - A20 are multiplexed with the OL and R-Path addresses and sent to the memories. The VME address line A1 is set low during D32 transfers and A2 is used to switch between the ODD and EVEN memory banks. VME addressing is selected by setting Source On (Bit 3 of the Control register) high.

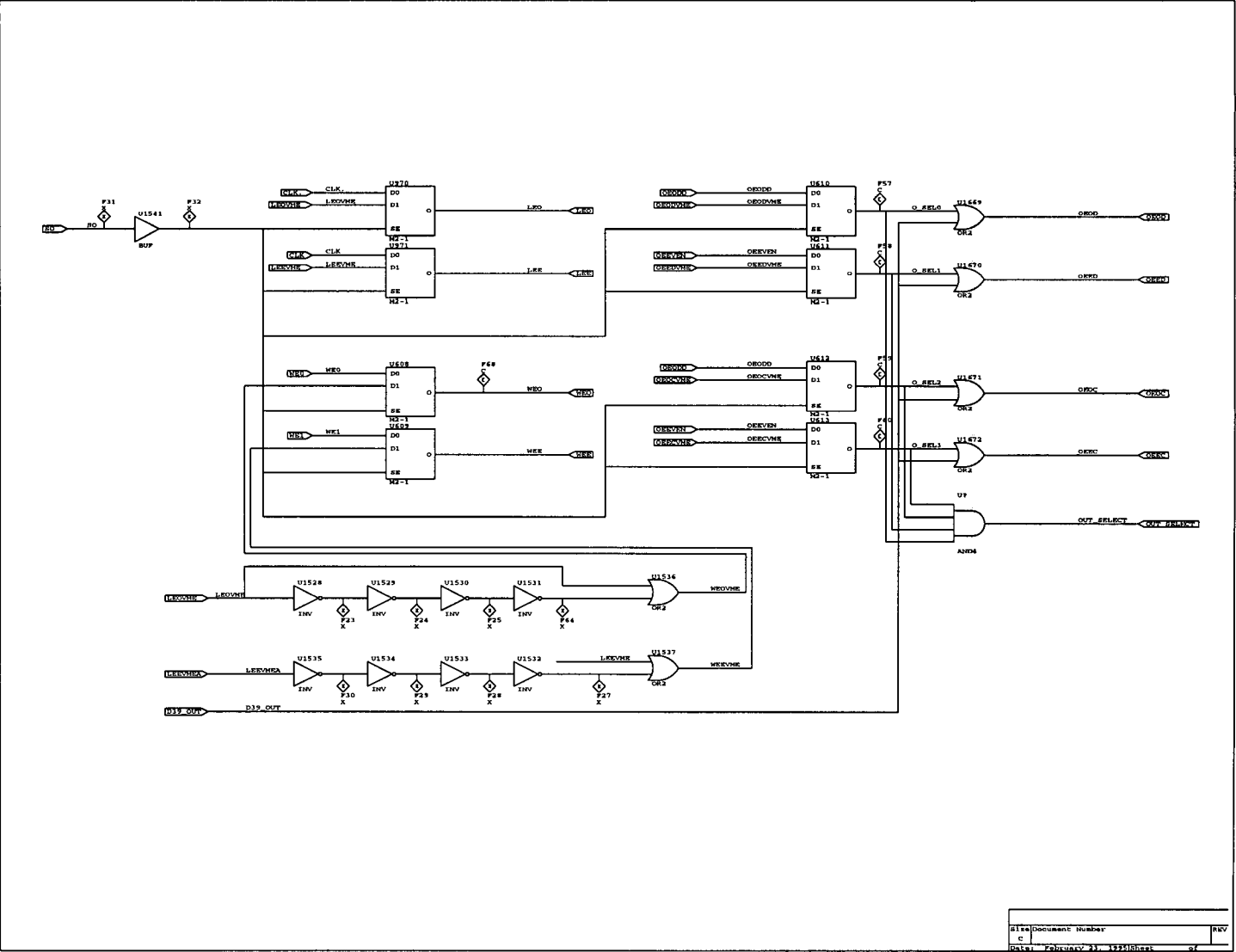


Figure F.3: Xilinx Read/Write strobe multiplexer circuit

Site/Document Number	REV
C	
DATE: February 23, 1999	Sheet 2 of 2

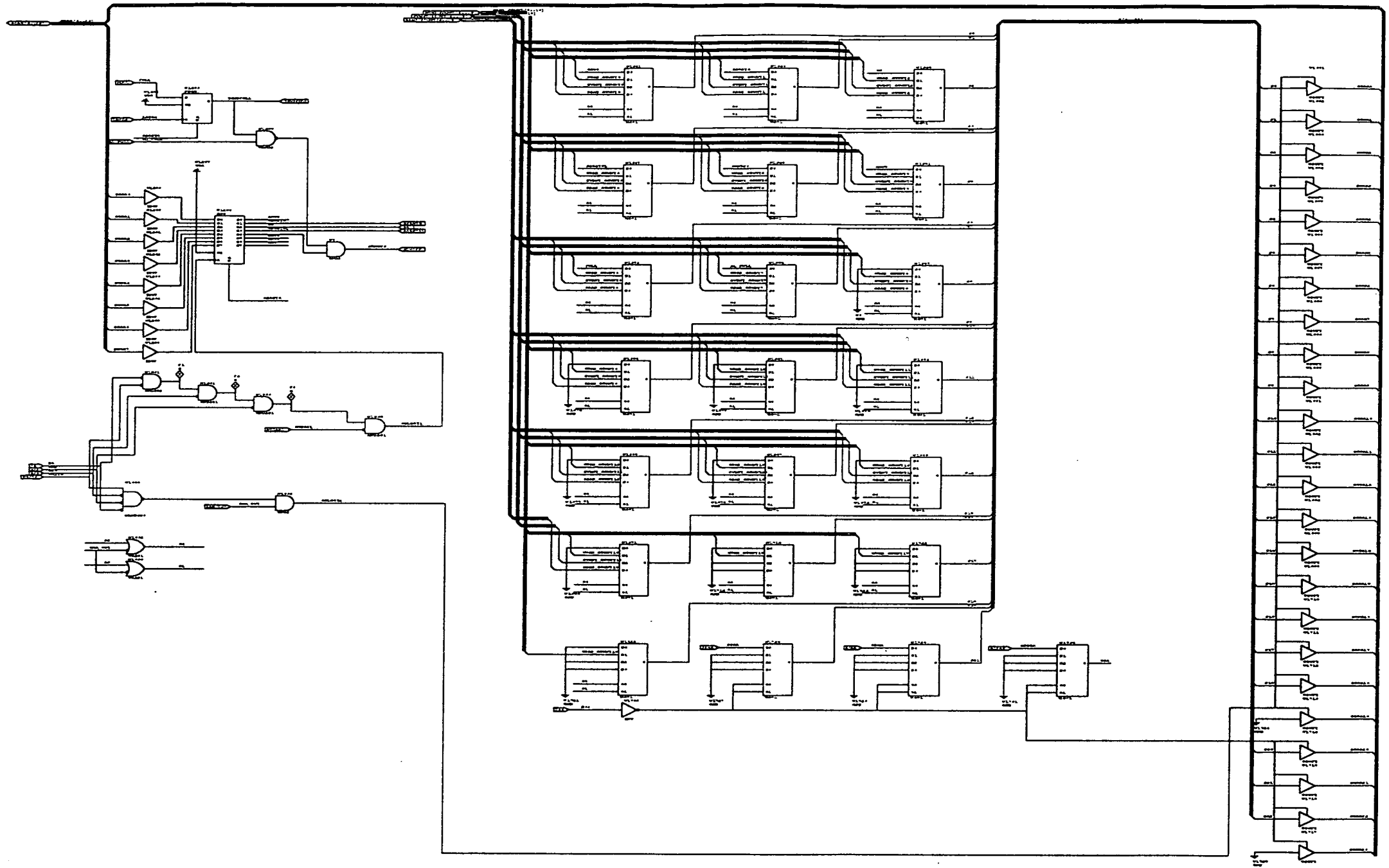


Figure F.4: Xilinx CSR Circuit

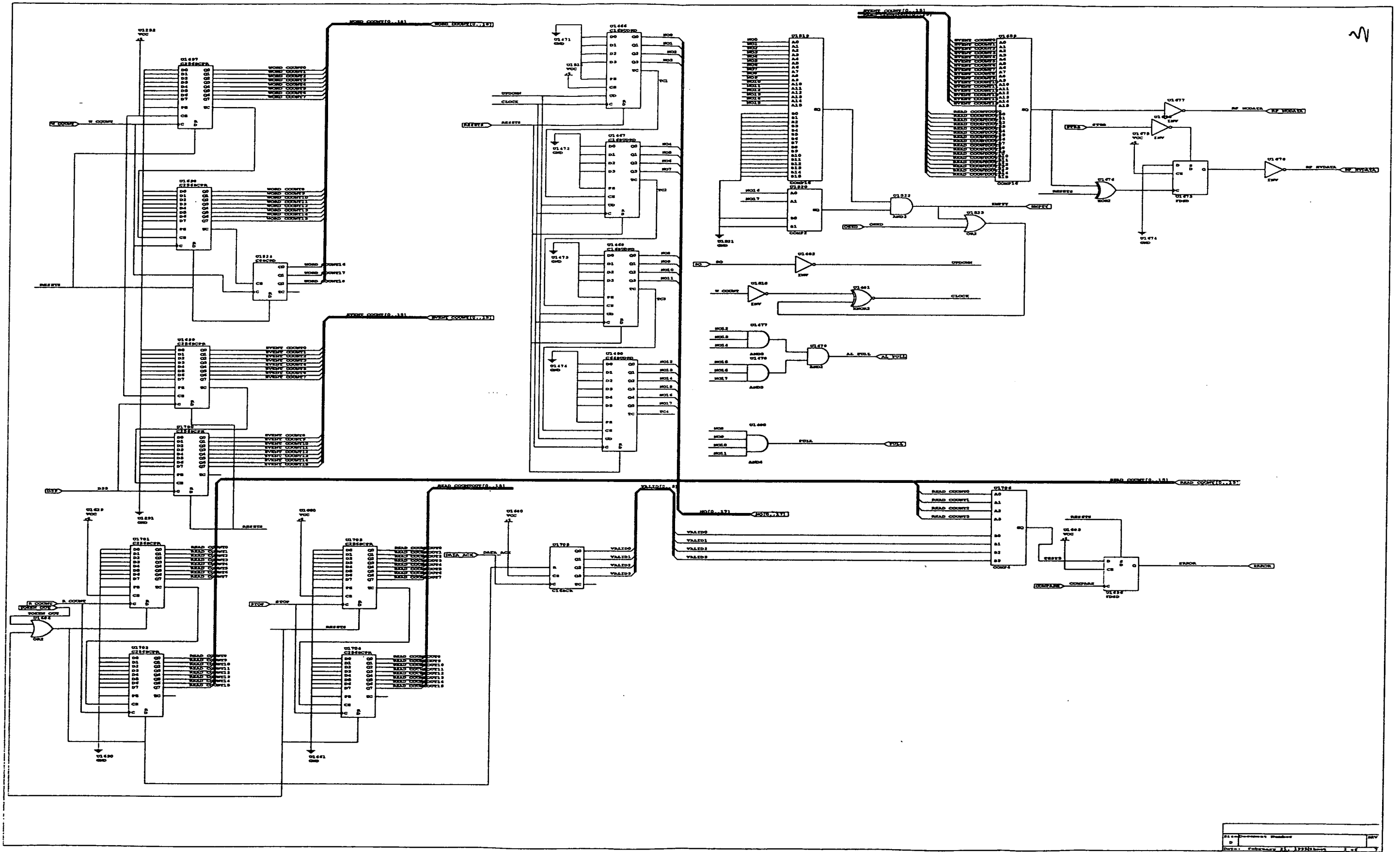


Figure F.5: Xilinx Counters and Protocol Circuit

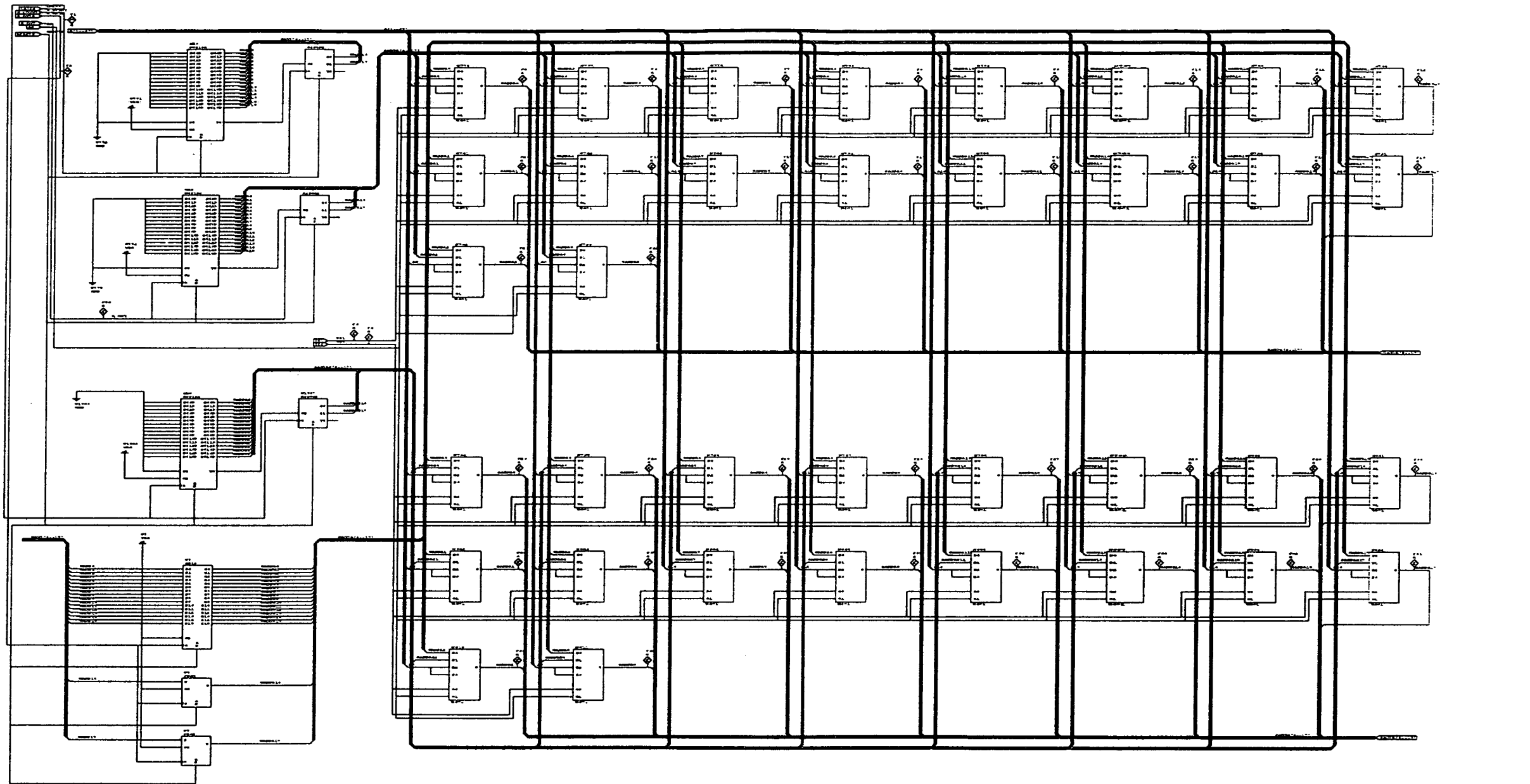


Figure F.6: Xilinx Memory Address Generation

F.2 Optical Link and R-Path Memory Control

The signals that are sent to the memories from the Xilinx during data taking are derived from the OL strobe (STRB) and the R-Path clocks (RP_CLOCK1 and RP_CLOCK2). STRB is a 10 MHz, 66% duty cycle signal which goes low 33 ns after the data is presented at the input to the IB and high 33 ns later. The R-Path clocks have a 50% duty cycle and run at 25 MHz when data is being taken at 100 MBytes/sec. The STRB and R-Path clocks are completely independent of each other. As the IB must be able to take and send data simultaneously the write enable and output enable signals sent by the Xilinx must not create contention at the memories. To do this the read operation is given precedence over the write, due to the short time (40 ns) of a R-Path data cycle.

In figure F.7 STRB enters the schematic from the left and is input to two XOR gates. The output of gate U1524 is $\overline{\text{STRB}}$ while the output of U1525 is a delayed STRB. Because the delay through each gate should be similar, the inverted and un-inverted strobes should change level in synchronisation with each other. The inverted strobe clocks through a WRITE_REQUEST signal from the flip-flop U574.¹ It is also routed to the clock input of the flip-flop U5 which toggles every write cycle. The output of U5 is then used to select which memory (ODD or EVEN) the data word is written to.

The output enable signals are derived from RP_CLOCK1. This is initially halved in frequency to give an 80 ns period and then inverted by the same arrangement of XOR gates as the strobe. The inverted clock (labelled OE1) is low during an even memory read and the un-inverted version (OE0) low during an odd memory read.

As STRB and RP_CLOCK1 are independent of each other the write request signal could arrive at any time in relation to OE0 and OE1. To ensure that the write does not occur during a read cycle it is queued until the relevant output enable has gone high.

In the even memory case OE0, WRITE_EVEN and WRITE_REQUEST are ANDed

¹Each flip-flop is clocked on a rising edge and reset or set on a logic high.

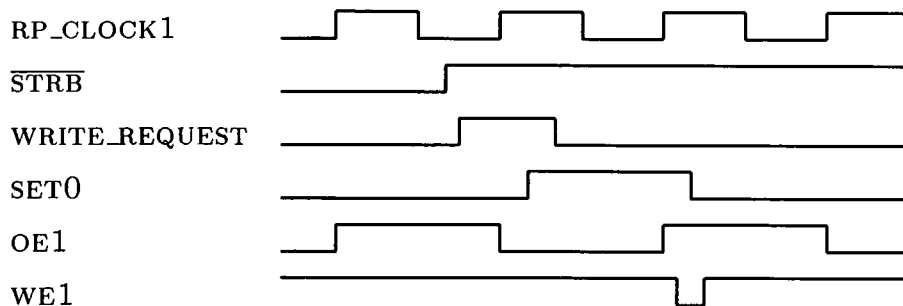


Figure F.8: Generation of Write Enable for even memory

together and the resultant signal used to clock the flip-flop U1639. A high on the output line (SET0) indicates that OL data has arrived and will be written to the even memory. The WRITE_REQUEST is removed at this point. However, the high will not be passed through to WE1 until OE1 has gone high indicating that the even memory is not being read. When this happens U1639 is reset, giving a short (approximately 15 ns) pulse at the write enable output, WE1. This process is illustrated in Figure F.8. For the odd memory write enable OE0 and OE1 are swapped around and WRITE_ODD takes the place of WRITE_EVEN in order to produce WE0.

The OL write address to both memories is the same. The write counter in the address section (Figure F.6) is clocked on the rising edge of W_COUNT and latched through to the Xilinx output when LATCH goes high. W_COUNT counts up the address on the rising edge of every second strobe while LATCH passes the data on the falling edge at the start of the write cycle.

When the IB receives the token the HAVE_TOKEN line is driven low. This signal is generated in the token circuit section of the Xilinx schematic (shown in Figure F.9).

TOKEN_IN is first ANDed with RP_NODATA, if this signal is low (i.e. there are no events in the IB memory) then the IB passes the token on to the next IB without setting HAVE_TOKEN low. HAVE_TOKEN is clocked by RP_CLOCK1 and input to a 4-input OR gate (U1702 in Figure F.7) and a 2-input OR gate (U1665). The output of U1665 goes low, enabling the transmission of the memory output enable signals (OEODD and OEEVEN) to the Xilinx output.

When the output of U1702 goes low C_STOP is enabled. When this signal is low the rising edge of RP_CLOCK1 clocks the flip-flop U1639. This flip-flop generates the signals R_CNT and R_CNTE. These are combined with RP_CLOCK1 to produce the odd and even output enable signals, OEODD and OEEVEN.

The odd and even read address counters, in Figure F.6, are incremented on the rising edge of R_CNT and R_CNTE respectively.

The circuit shown in Figure F.3 multiplexes the write and output enables with those from the VME section. The multiplexer outputs are switched by the SO signal in the CSRs. When SO = 0, the Optical Link and R-Path addresses and write/read strobes are passed to the Xilinx outputs, when SO = 1, the VME addresses and write/read signals are sent to the memories.

The memory addresses are also multiplexed with each other. The two address busses, OAddr0-17 and EAddr0-17, carry the write, read or the VME address to the memories. In Figure F.6 the two banks of multiplexers have as inputs the VME lines A3 - A20, the OL write address and the odd or even read address. The multiplexers are switched by SO, OE0 and OE1 as described in Table F.2.

Figure F.10 shows the ideal timing relationship between the output enable and the memory address at the even bank.

During the last word of an R-Path transfer the D39 bit goes high at the output of the IB. This bit is input to the Xilinx, where it is labelled D39_OUT. When it goes high the read address counters stop counting up and the memory output enables are inhibited. In addition a state machine in the token section of the Xilinx schematic (Figure F.9) is enabled. This generates the following signals:

ODD MEMORY ADDRESS

SO	OE0	OAddr0 - 17
0	0	ODD R_PATH READ ADDRESS
0	1	ODD OL WRITE ADDRESS
1	X	VME ADDRESS

EVEN MEMORY ADDRESS

SO	OE1	EAddr0 - 17
0	0	EVEN R_PATH READ ADDRESS
0	1	EVEN OL WRITE ADDRESS
1	X	VME ADDRESS

Table F.2: Memory Addressing Scheme Implemented in the Xilinx FPGA

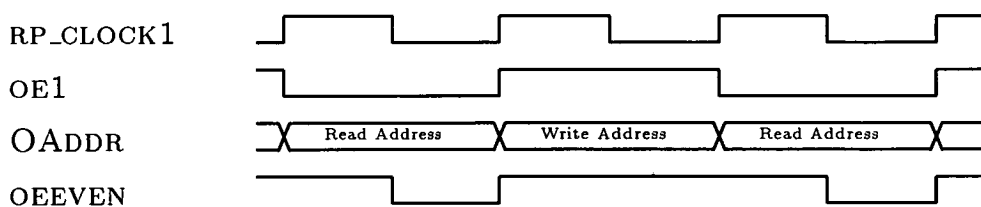


Figure F.10: Even Memory Read Signals

- R_COUNT increments the word count by one. The IB has to output the number of words it has sent to the FOF across the R-Path. The 20-bit word count itself is included as an extra word. Therefore the word count is incremented by one before being sent.
- CSR_OUT turns on the CSR output buffers. The word count is placed on the lower 20 bits. These lines are routed to the R-Path.
- RP_LETMP2 goes low to clock the word count through the R-Path transceivers. It is combined with the latch enable for the data words (RP_LETMP) to form the R-Path latch enable that is sent from the Xilinx (RP_LE).
- COMPARE samples the output of the comparator U1706. This compares the number of DATA_VALIDs sent with the number of DATA_ACKs received. If these are not equal then the token is not sent.
- TOKEN_OUT is the output of the multiplexer U1694. This is switched by the RP_NODATA line. If the IB has no events to be read out (RP_NODATA = 0) then TOKEN_IN is fed directly to the TOKEN_OUT output. If RP_NODATA = 1, i.e. there are events in the memory, then a 40 ns pulse is output on the TOKEN_OUT pin. This sends the token to the next IB in the chain.

Appendix G

Simulation Results

G.1 Dataflow Simulation

A series of simulations of the NA48 data acquisition were carried out by Dimitri Kirillov¹ at Edinburgh University during the early part of 1995. The data acquisition system was simulated using the Verilog package. A schematic of the dataflow model is shown in Figure G.1.

The Figure shows a 10 channel dataflow scheme. Each sub-detector outputs data from its readout electronics (sdet11 and sdet21) to an optical link (ol1). From the optical link data is collected by the IB (ib1) and sent across the R-Path to the FOF. Events are input to the model from the event generator (evt_gen1) which simulates the detector. Also present in the model is some trigger logic and the trigger supervisor (ts1). The clock generator (clk_gen) provides timing for the trigger system.

In the model each element of the dataflow is described behaviourly, i.e. the operation of each element not the actual physical design is described in code.

A typical simulation result is shown in Figure G.2. The simulation models 10

¹Dimitri Kirillov carried out the simulations detailed in this Appendix while on a fellowship from the International Association for the Promotion of Cooperation with Scientists from the Independent States of the Former Soviet Union. Reference Number : INTAS-93-1197.

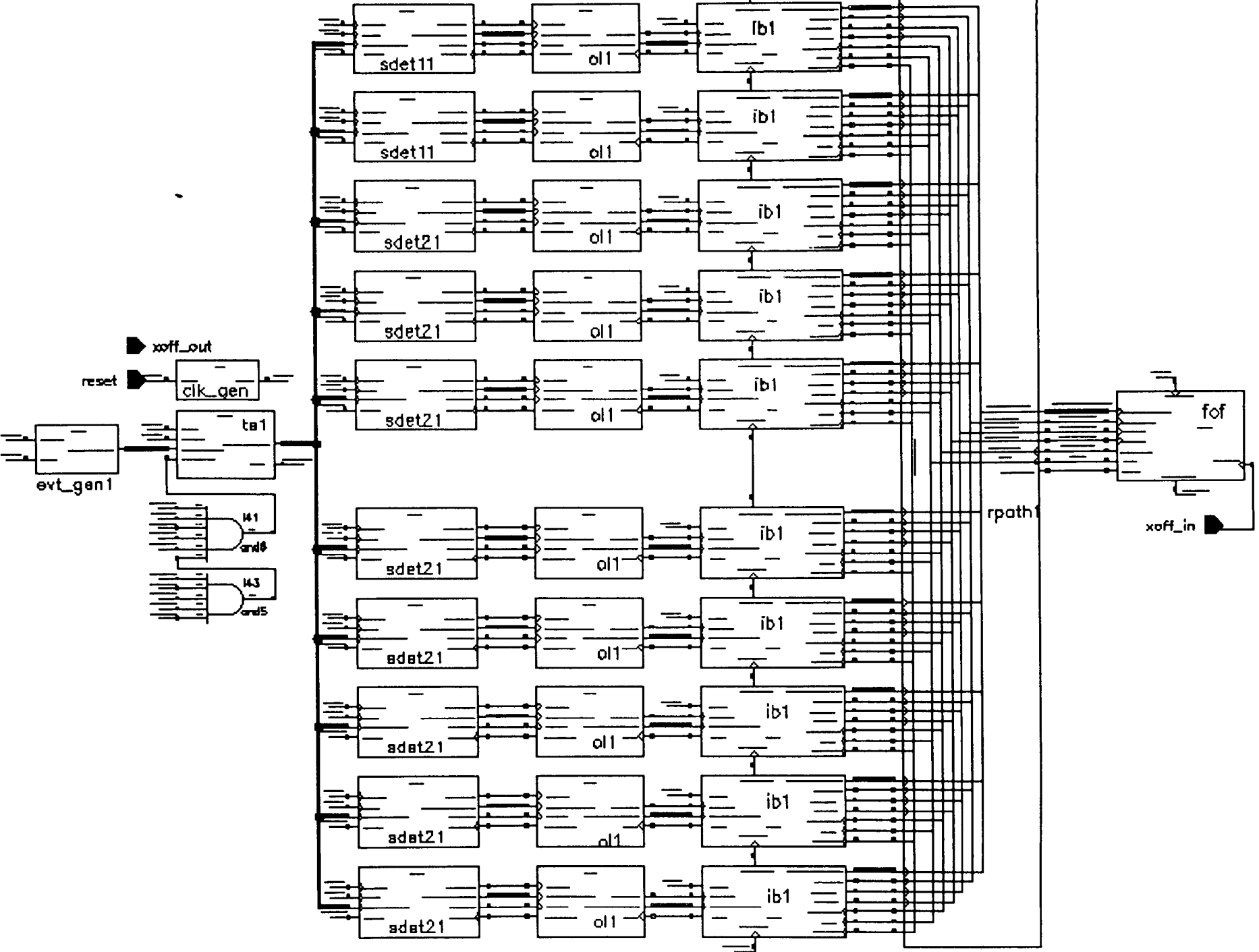


Figure G.1: Verilog Schematic of the NA48 Dataflow System

Trigger	Channel 0	Channel 1	Channels 2-9
Charged	10	50	10
Neutral	10	10	300
Calibration	10	10	6000

Table G.1: Simulation Trigger Rates

channels of data from the sub-detectors to the FOF.

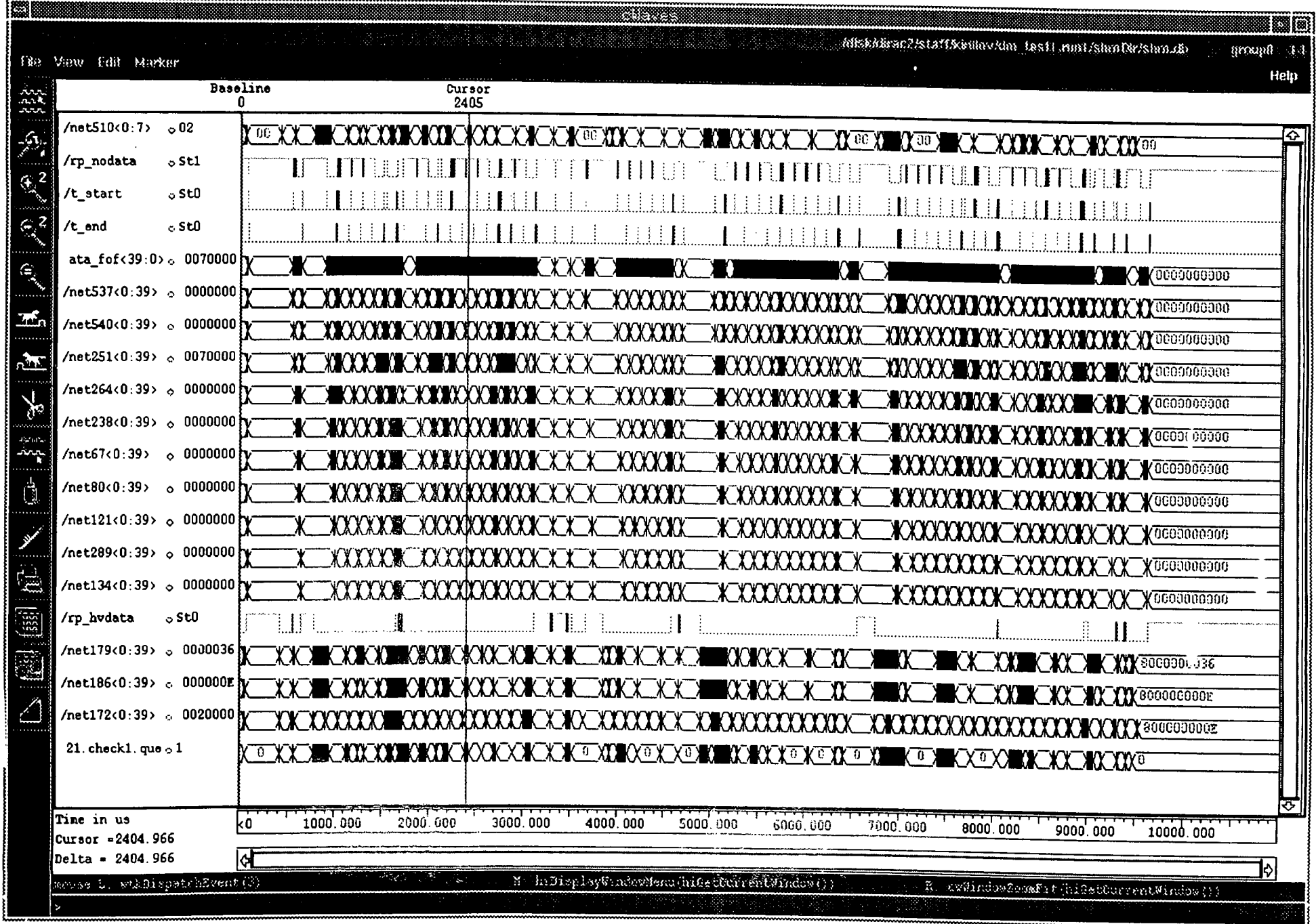
The displayed traces are:

- `/rp_nodata` — This signal is low when an event is present in every input buffer. This signal trace shows how the IBs fill and empty during the data transfers.
- `/t_start` — The ‘token start’ trace shows the token pulse leaving the FOF.
- `/t_end` — The ‘token end’ signal goes high when the token returns to the FOF after the read-out of the IBs.
- `data_fof<39:0>` — This trace displays the data at the input to the FOF.
- `/net537<39:0>` – `/net134<39:0>` — The ten traces below `data_fof<39:0>` show the data from each optical link channel at the input to the IB.
- `/rp_hvdata` — The `rp_have_data` signal is low when data is present in the IBs.

The number of 4 byte words from each trigger transferred over each of the channels is shown in Table G.1.

Channels 2-9 model the data acquisition from the LKr Calorimeter while the other two channels model the data sent from the drift chambers and another detector. The calibration data is sent during the first 5 ms of the spill. This is followed by the neutral and charged trigger information.

Figure G.2: Simulation of the NA48 Dataflow System



The simulation provides a ‘virtual’ dataflow system where new modules can be tested and their effect on the data acquisition assessed. The model also enables us to predict the point at which saturation will occur.

G.2 Input Buffer Simulation

The IB was simulated in more detail than the rest of the system. Figure G.3 shows the Verilog schematic that was used. The IB_SCH section contains all the Xilinx logic for one IB channel. The IB1_MEM1 schematic is a model of the IB memory banks and R-Path interface. The other elements of the dataflow are modelled as in Section G.1.

Figures G.4 and G.5 show an IB write-in and a read-out cycle respectively. Events are generated in the same manner as before and the R-Path clock is provided by the clk_gen1 macro.

Figure G.4 shows the following signals:

- /wee — Write_Enable for the Even memory.
- /eaddr<17:0> — Even memory address.
- /lee — Latch_Enable for the Even memory.
- /weo — Write_Enable for the Odd memory.
- /oaddr<17:0> — Odd memory address.
- /leo — Latch_Enable for the Odd memory.
- /D<39:0> — The data from the optical link model.

The data from the optical link is presented to the IB in the following format:

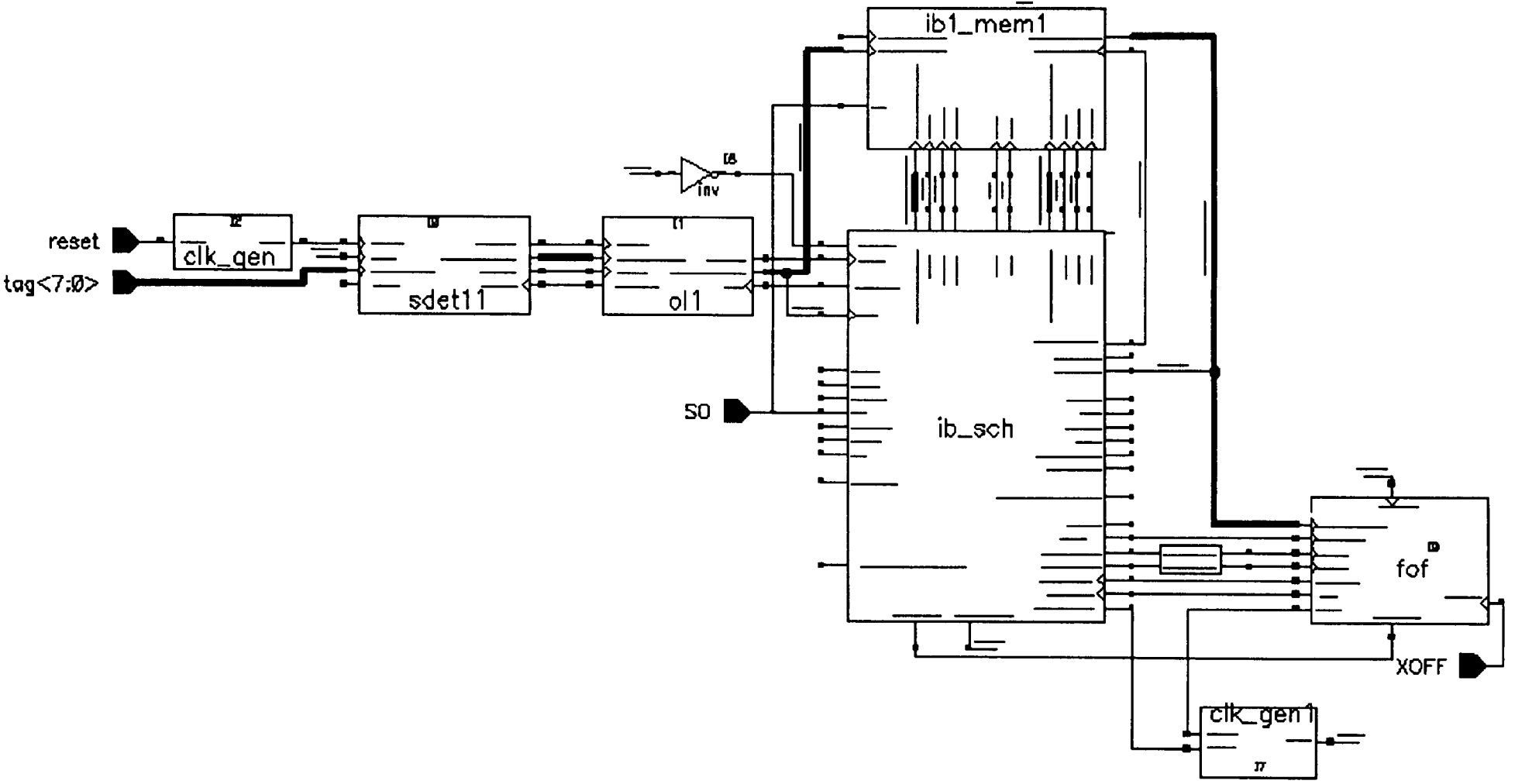
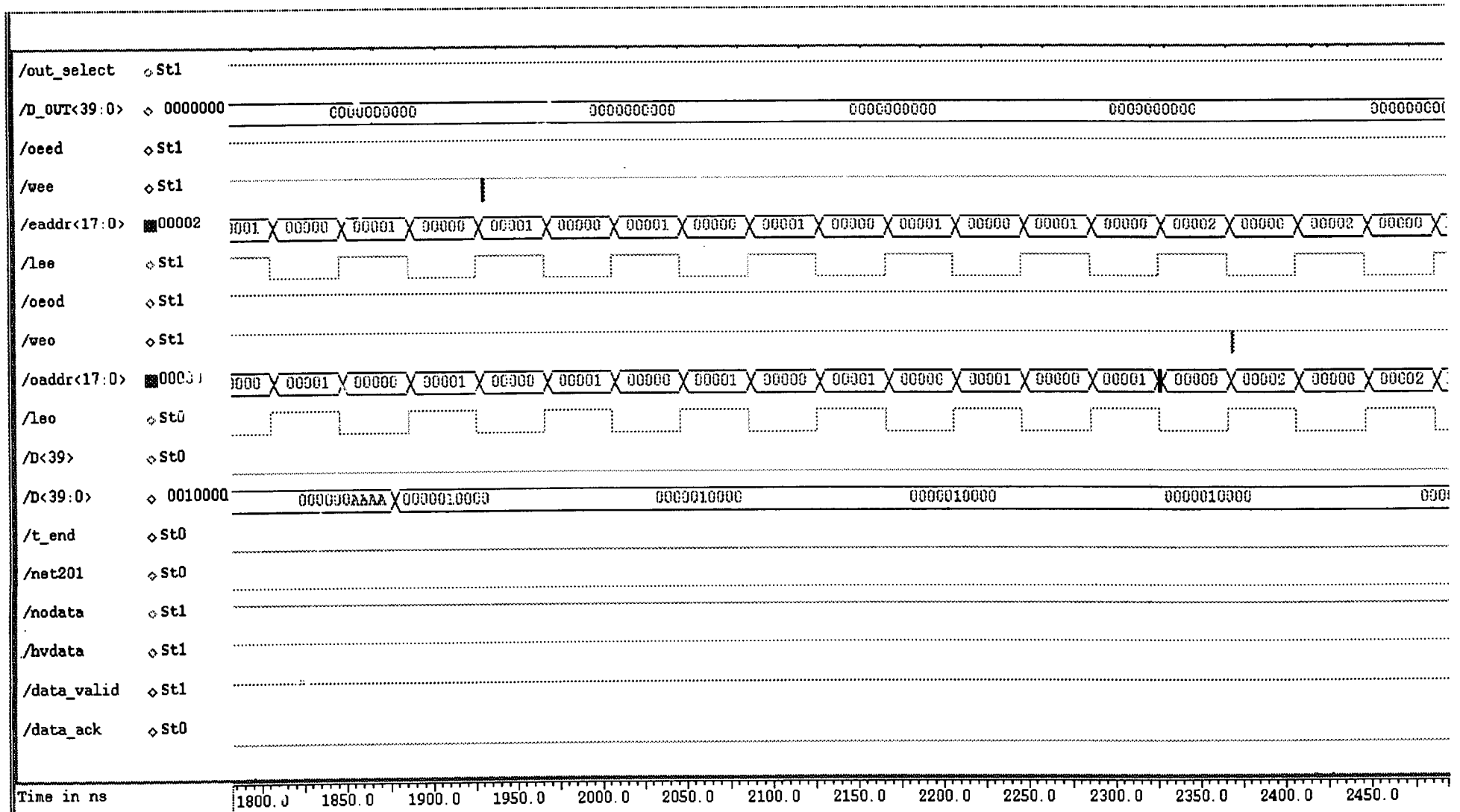


Figure G.3: Verilog Schematic of the Dataflow with an Expanded IB

Figure G.4: Input Buffer write-in Cycle from Optical Link



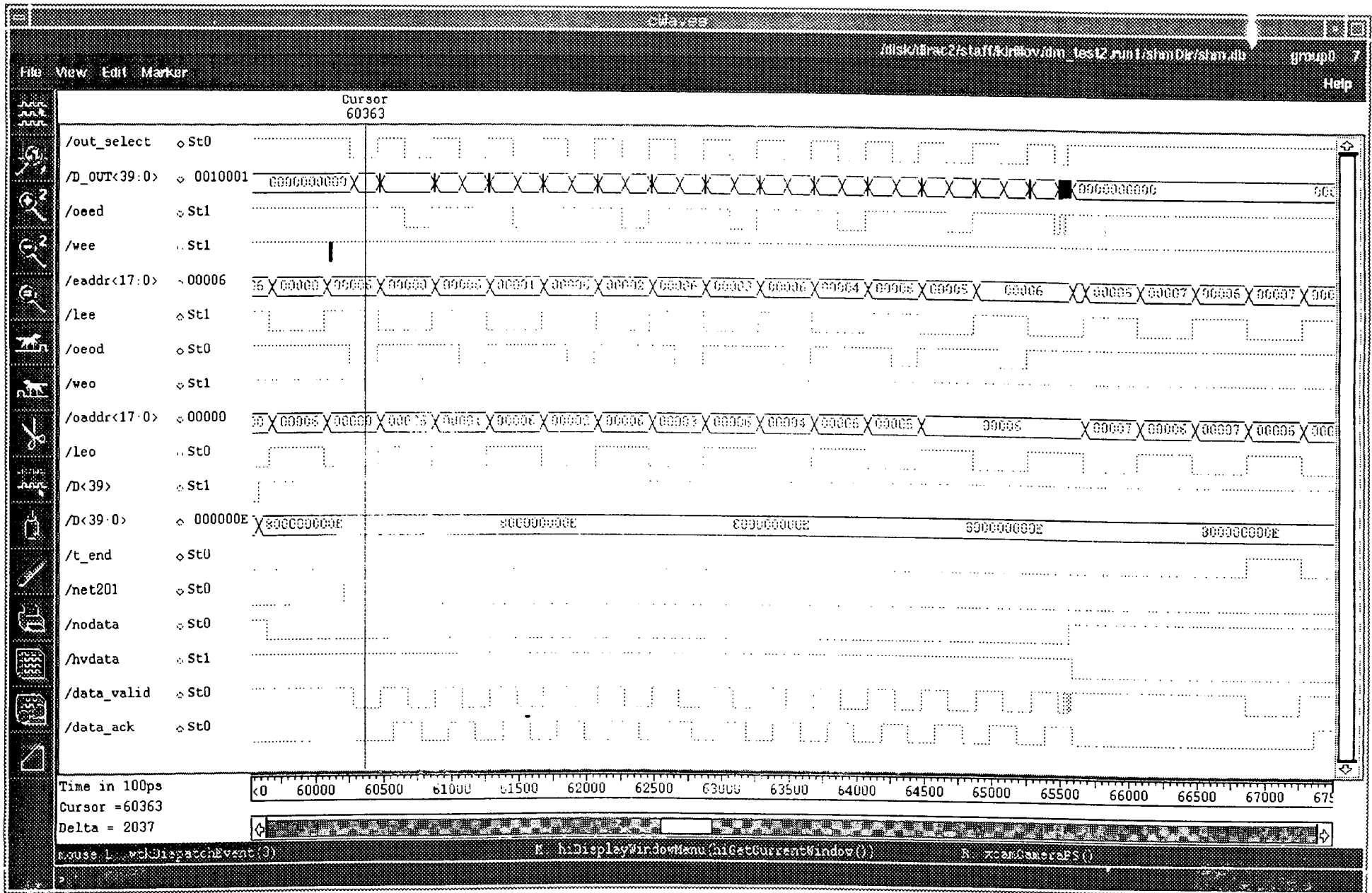


Figure G.5: Input Buffer read-out Cycle over R-Path

Word 1	Block Identifier
Word 2	Trigger Word
Word 3	Time Stamp
n Words	User Data
Word n+4	Block Length

In the case of this event the block identifier is 0x10001, the trigger word is 0, the time stamp is set to 0xaaaa and the user data consists of 10 words of value 0x10000. The block length word has the D39 bit set high indicating that it is the end of an event.

The addresses are multiplexed between the read address (which is permanently at 0000 as no read-out has taken place) and the write address. In the figure the third word of the event is written into the even memory at address 0001 and the next word to the odd memory address 0002. The read-out signals such as `D_OUT<39:0>` and `/data_valid` are not active until a token is sent to the IB. When this happens the read-out shown in Figure G.5 commences.

The following signals are shown in Figure G.5:

- `/out_select` — This clocks the data from the memories through the read latches.
- `/D_OUT<39:0>` — Data output on the R-Path.
- `/oeed` — Output Enable for the **E**ven **D**ata memory.
- `/oeod` — Output Enable for the **O**dd **D**ata memory.
- `/net201` — Token_in line.
- `/t_end` — Token_out line.
- `/nodata` — R-Path no_data line.
- `/hvdata` — R-Path have_data line.

- `/data_valid` — The `data_valid` strobe from the IB.
- `/data_ack` — The `data_acknowledge` strobe back from the FOF.

The token input to the IB (shown on the trace labelled `/net201`) sets off the R-Path read-out. The data is output from each memory by the output enable signals and then clocked through the read latches by the `out_select` signal. The data is then transmitted to the R-Path along with the `data_valid` strobe. The R-Path `no_data` signal goes low when a full event is present in the buffer and high when the data has been read out. The `have_data` line goes high when the IB channel contains data and low when the data has been read out.

Bibliography

- [1] G.D.Barr *et al.* Proposal for a Precision Measurement of ϵ'/ϵ in *CP* Violating $K^0 \rightarrow 2\pi$ decays. *CERN Proposal* (July 1990).
- [2] V.L. Fitch J.H. Christenson, J.W. Cronin and R. Turlay. Evidence for the 2π Decay of the K_2^0 Meson. *Physical Review Letters*, 13(4):138-140 (1964).
- [3] G.D. Barr *et al.* A New Measurement of Direct *CP* Violation in the Neutral Kaon System. *CERN Preprint*, CERN-PPE/93-168 (October 1993).
- [4] B. Winstein and L. Wolfenstein. The Search for Direct *CP* Violation. *Reviews of Modern Physics*, 65(4):1113-1147 (October 1993).
- [5] P.Grafström *et al.* A Proton Tagger for the NA48 Experiment. *CERN Preprint*, CERN-PPE/94-11 (January 1994).
- [6] P. Buchholz. *CP* Violation in the K System : Future Experiments in Europe. *NA48 Note*, (1993).
- [7] Perugia and Cagliari Groups. The NA48 Charged Hodoscope. *NA48 Note*, NA48/94-37 (December 1994).
- [8] V.Fanti *et al.* Performance of an Electromagnetic Liquid Krypton Calorimeter. *CERN Preprint*, CERN-PPE/93-210 (December 1993).
- [9] H. Burkhardt *et al.* The Beam and Detector for a High-Precision Measurement of *CP*-Violation in neutral-kaon decays. *CERN Preprint*, CERN-EP/87-166 (September 1987).
- [10] B.Hay. The NA48 Muon-Veto System. *Edinburgh University Report* (1994).
- [11] G.D. Barr and W. Funk. The Level 2B Trigger. *NA48 Note*, NA48/94-11 (March 1994).

- [12] G.D. Barr *et al.* Description of the NA48 Neutral Trigger Electronics. *NA48 Note*, NA48/94-4 (1994).
- [13] S.Anvar *et al.* NA48 Charged Trigger and Data-flow. *NA48 Note*, NA48/94-13 (1994).
- [14] F. Bertolino *et al.* The NA48 Trigger Supervisor Design. *IEEE Transactions on Nuclear Science*, February 1994.
- [15] G.D. Barr *et al.* NA48 Data-flow Description. *NA48 Note*, (1993).
- [16] P. Brodier-Yourstone *et al.* NA48 Fiber Optic Link Proposal. CERN-PPE (July 1993)
- [17] P. Brodier-Yourstone. The Destination Module of a Fiber Optic Link using ANSI Fiber Channel Modules. *Diploma Thesis*, CERN-PPE (December 1993).
- [18] O. Boyle and N.E. Mckay. R-Path Protocol Specification for the NA48 Data Merger. NA48, (February 1995).
- [19] A. Pastore *et al.* The HIPPI to TURBOchannel Connection. CERN-PPE, (1993).
- [20] CERN Program Library, The ZEBRA System. CERN CN,ECP,PPE. (1993).
- [21] W. D. Peterson. The VMEbus Handbook. VFEA International Trade Association. (1993).
- [22] P. L. Borrill. High Speed 32-bit Busses for Forward Looking Computers. *IEEE Spectrum*, (July 1989).
- [23] C. F. Parkman. Summary of Backplane Bus Characteristics. CERN, (November 1990).
- [24] J. Martinez. BTL Tranceivers Enable High-Speed Bus Designs. *Electronic Design News*, (August 1992).
- [25] Xilinx Inc. The XC4000 Data Book (1992).
- [26] Orcad Inc. Schematic Design Tools User Guide (1992).

- [27] Xilinx Inc. XACT Reference Guide Vol. 2 - The XACT Design Editor (1992).
- [28] K. Peach. Hardware and Software Issues for Future Experiments. *Proceedings of the Conference on Computing in High Energy Physics*. (April 1994).
- [29] M. Letheren *et al.* An Asynchronous Data-Driven Event-Building Scheme Based on ATM Switching Fabrics. *IEEE Transactions on Nuclear Science*. (Feb 1994).
- [30] J. Bogaerts *et al.* SCI Based Data Acquisition Architectures. *IEEE Transactions on Nuclear Science*. (1992).
- [31] H. Müller *et al.* First Experience with the Scalable Coherent Interface. *IEEE Transactions on Nuclear Science*. (Feb 1994).
- [32] R. Pordes *et al.* Fermilab's DART DA System. *Proceedings of the Conference on Computing in High Energy Physics*. (April 1994).
- [33] Atlas Trigger-DAQ Steering Group. Trigger and DAQ Interfaces with Front-End Systems: Requirement Document (Draft 1.4). (July 1995).

Summary of the 1995 NA48 Physics Run

Nicholas Mckay

October 3, 1995

1.1 Overview

Between the submission of this thesis and my final examination a NA48 physics run took place at CERN. During this run data was taken from seven subdetectors; the magnetic spectrometer, the hadron calorimeter, the hodoscope, the tagger, the anticounters, the muon veto and the AKS. Data from each of these was input to the data acquisition system described in Chapters 3 and 4 of this thesis. In this addendum some of the first physics results are presented. It should be noted that the analysis of the data is currently at an early stage and that more full analyses are being carried out by postgraduate students from several of the participating institutions for their PhD theses.¹

1.2 Data Acquisition System Performance

During the run several data acquisition goals were reached:

- The system was run at the nominal trigger rate of 10 kHz with 2 subdetectors, the magnetic spectrometer and the hadron calorimeter.
- The system reliably acquired data at up to 8 MBytes/sec.
- In the 30 days of data taking that took place 555 GBytes of data was collected from approximately 87000 bursts.

Before the run commenced two IB problems were identified:

- When writing and reading at the same time the data from the even memory of each channel was found to be occasionally corrupted. This was due to the write enable pulse being sent to that memory before the data had settled. The problem was solved by feeding the WEE signal into a two 74F04 inverter chain before sending it to the memory.
- The IB to Optical Link XOFF circuit had been left in the configuration that was used during the previous run, i.e. an up/down counter was

¹Many thanks to Bruce Hay and Eddie Mazzucato for providing the plots that are used in this section.

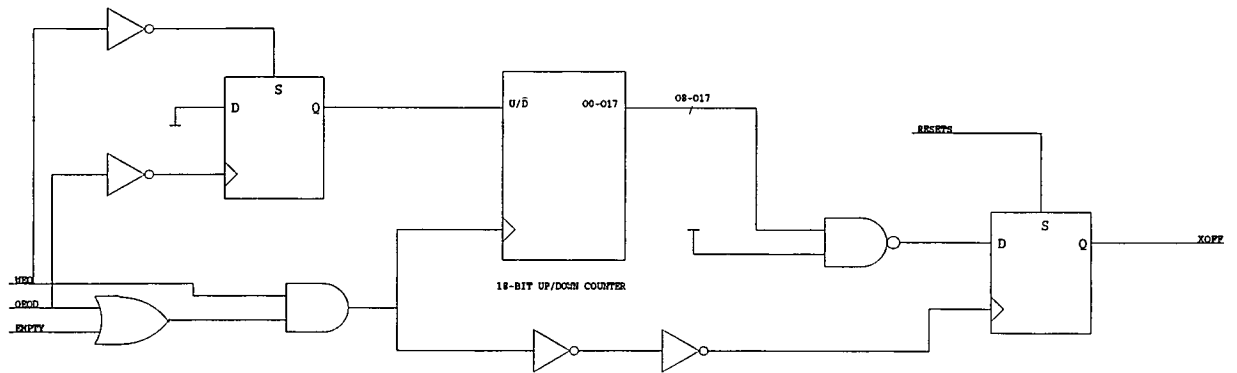


Figure 1.1: New IB to OL XOFF Circuit.

clocked up as data was read into the IB and down when it was read from VME. This configuration didn't work when the data was output to the R-Path. The circuit was changed to that shown in Figure 1.1 using the XDE program. The new configuration worked successfully.

The Data Merger was fully functional during the run. The Optical Link, token passing and R-Path protocol schemes proved successful and the basic architecture was shown to work. Five minor problems were encountered, none of which affected the ability of the system to take data, these were:

- The word count from each IB was always one more than it should have been. This was fixed in software by subtracting one from the word count.
- An extra 'buffer' word had to be included at the end of each sub-event as the IB always output one word after the end-of-event word.
- Another buffer word had to be included at the start of each sub-event as the FOF always corrupted the first data word of each event.
- The last IB word count was sometimes overwritten with the first padding word, and the first padding word was sometimes overwritten with the last IB word count. This was due to a race condition in the FOF and was fixed in software.

None of these problems caused any corruption in the data and all signs are that the data that was collected is good. The data acquisition rate was limited by the amount of events that the slowest readout system could send during the run, as each sub-detector has to send the same amount of sub-events so that the events are not mixed. The maximum rate was around 4000 events per burst. The next two sections summarise some of the early results.

1.3 K_S Beam Data

Figure 1.2 shows a distribution of the measured kaon mass. The data used to calculate the mass was taken from the information sent by the spectrometer during a K_S beam run.

The K_S particle has two main decay modes, $K_S \rightarrow \pi^+\pi^-$ and $K_S \rightarrow \pi^0\pi^0$. The first of these decays is seen as two charged tracks in the spectrometer while the second is only detected in the calorimeter. From the spectrometer information the momentum of the two charged pions is known and from this, assuming the pion mass, the mass of the kaon can be calculated.

In addition to the $K_S \rightarrow \pi^+\pi^-$ decay the $\Lambda \rightarrow p\pi^-$ decay also produces two tracks in the spectrometer. To differentiate between the two decays, the ratio between the momenta of the charged particles, P_r (plotted against the kaon mass in Figure 1.3), is calculated.

The calculation is carried out assuming that both the charged particles are pions. In the $K_S \rightarrow \pi^+\pi^-$ case this is true and the kaon mass is correct at around 0.48. In addition p_r is small as the two pions will have a similar amount of momentum. However, when the P_r calculation is performed on a Λ decay a proton is mis-labelled as a pion and the calculated mass varies depending on the large momentum of the proton. This leads to the two separate regions seen in the plot and from this plot a cut can be made on p_r at around 3 to remove the contribution from the Λ decay. Because the two decay modes can be easily differentiated the Λ mass can also be calculated from the data. A distribution of the calculated mass is shown in Figure 1.4. The background seen in each of the mass plots is mostly caused by the mis-identification of the type of decay.

1.4 K_L Beam Data

Figure 1.5 shows a plot of the square of the tangential component of the momentum (P_T^2) against the invariant mass of the charged vertex, assuming that the particles are charged pions.

The long-lived kaon is CP-odd and decays into three particles under CP-invariance. These decays give a non-zero value of P_T as some momentum 'goes missing' in a neutrino or a neutral pion, e.g. $K_L \rightarrow \pi^+\pi^-\pi^0$ or $K_L \rightarrow \pi e \nu$. However, if the tangential component of the momentum is zero then a CP-violating decay may have taken place, e.g. $K_L \rightarrow \pi^+\pi^-$. In Figure 1.5 a dark spot at around $M_c = 0.5$ shows these possibly CP-violating decays. Figures 1.6 and 1.7 show two sections through this plot. The first shows the slice from $P_T^2 = 0$ to $4 \times 10^{-4} [GeV/c]^2$ while the second shows the slice from $P_T^2 = 4 \times 10^{-4} [GeV/c]^2$ to $8 \times 10^{-4} [GeV/c]^2$. In both slices the contribution from $\pi^+\pi^-\pi^0$ decays is shown as a peak at around $M_c = 0.36 GeV/c^2$ and the contribution from the semi-leptonic decays appears as a background from $M_c = 0.35 GeV/c^2$ to $0.55 GeV/c^2$. Only the first plot shows a peak at around $M_c = 0.48 GeV/c^2$ which is the contribution from the CP violating $K_L \rightarrow \pi^+\pi^-$ decay, the second plot does not show this peak as P_T^2 is set too high.

The contributions from the $\pi^+\pi^-\pi^0$ decay and the semi-leptonics can be separated from each other by performing a cut in $(p'_0)^2$, where p'_0 is defined as the longitudinal momentum of the kaon in the Lorentz frame in which the longitudinal momentum of the two charged tracks is zero. In the 3 pion case the neutral pion will go undetected and thus the longitudinal momentum of the two charged tracks will add up to less than that of the kaon. This gives a positive definite value of p'_0 . In the case of a semi-leptonic decay ($\pi e \nu$ or $\pi \mu \nu$) an electron or a muon will be mis-labelled as a pion and a neutrino will go undetected. This mostly results in an imaginary value of p'_0 and so $(p'_0)^2$ is predominately negative. Figure 1.8 shows a distribution of $(p'_0)^2$ calculated from 30 bursts of K_L beam data.

By performing more detailed analysis on the K_L data from the 1995 run it is hoped that three of the four main K_L decay modes can be differentiated. This will enable us to calculate their branching ratios for the K_L beam in the NA48 experiment. In addition the branching ratio of the CP-violating $K_L \rightarrow \pi^+\pi^-$ decay could also be calculated from the data that has been acquired.

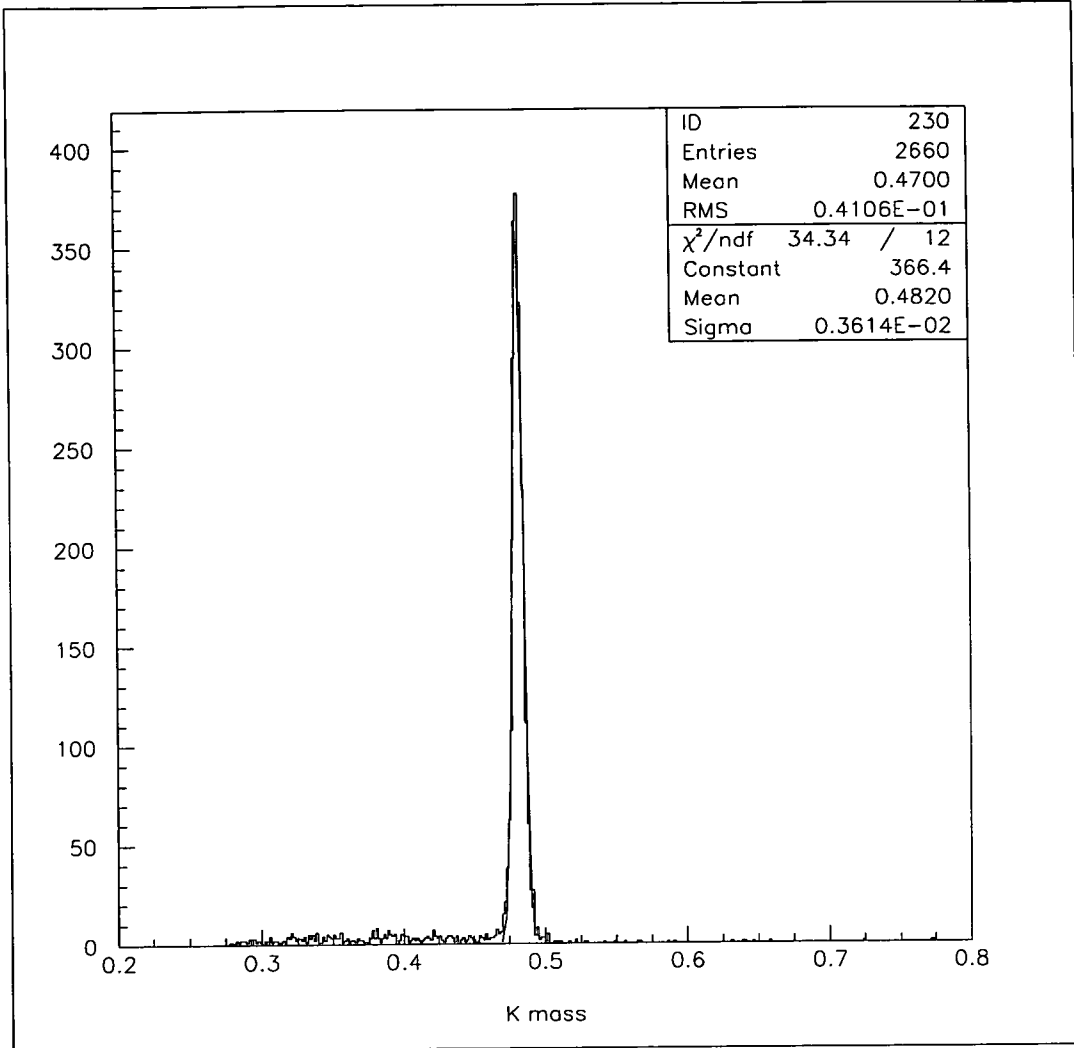


Figure 1.2: The Measured Mass of the Kaon.

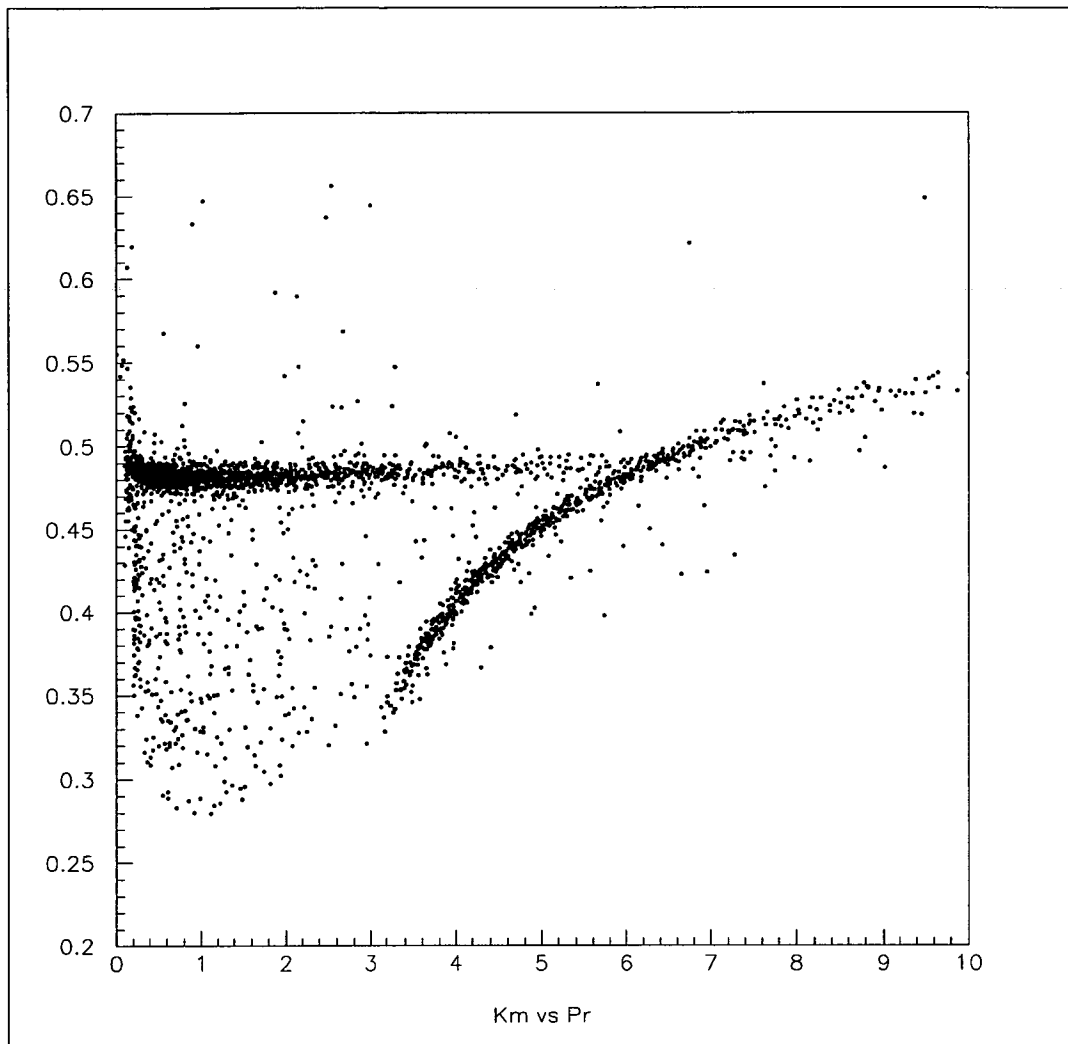


Figure 1.3: Plot of the Ratio of Momenta of the two Charged Particles in the Spectrometer against the Kaon Mass.

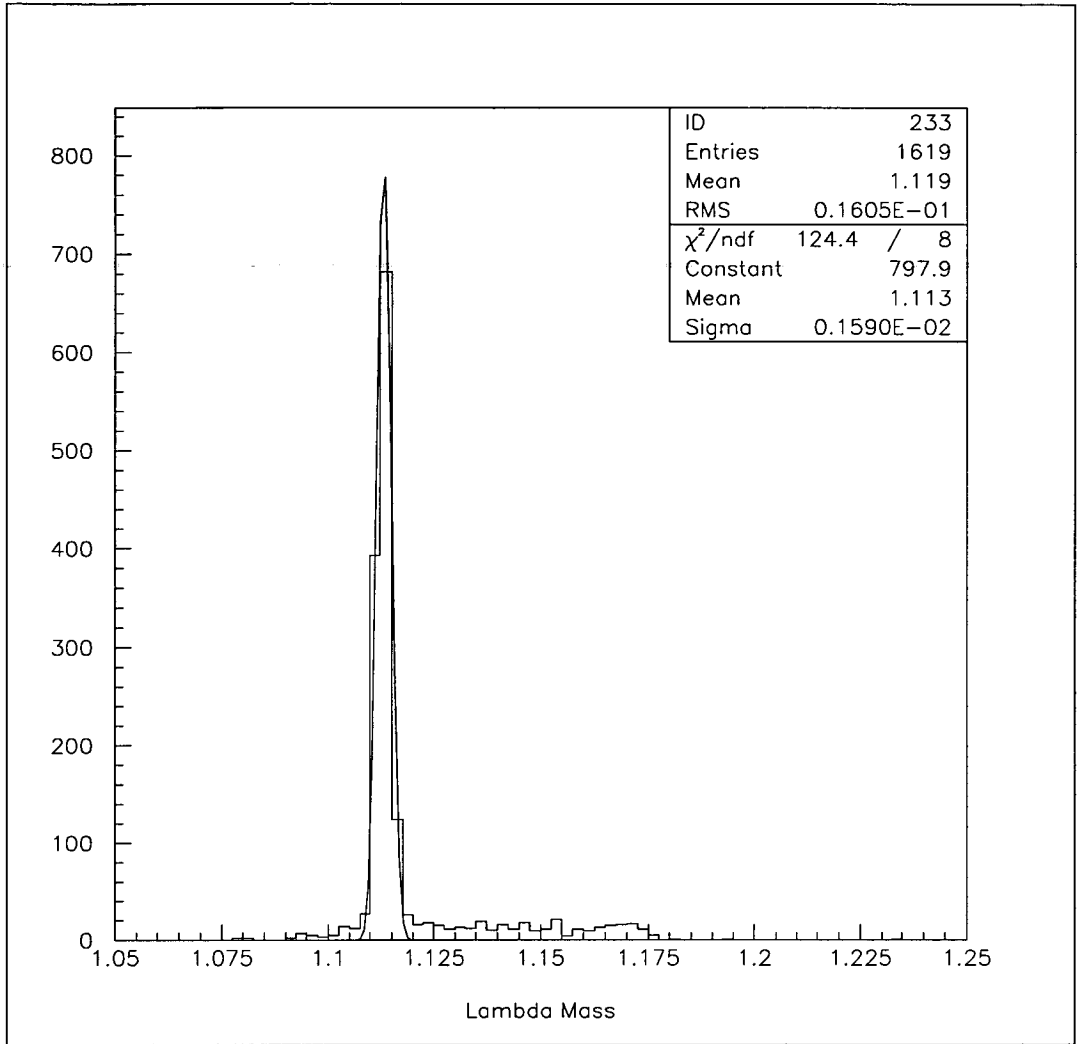


Figure 1.4: The Measured Mass of the Lambda.

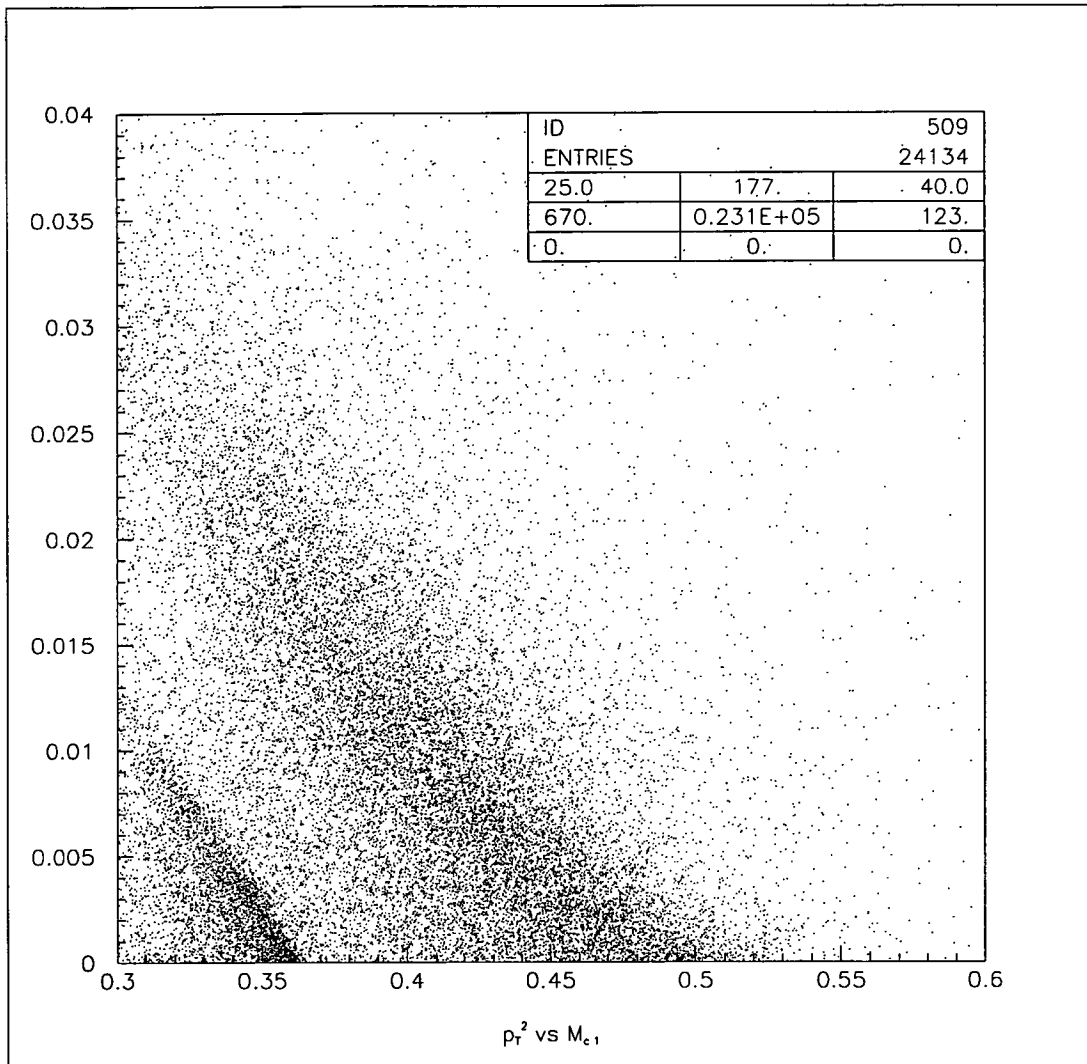


Figure 1.5: The Momenta of the Decay Particles plotted against the Invariant Mass of the Charged Vertex.

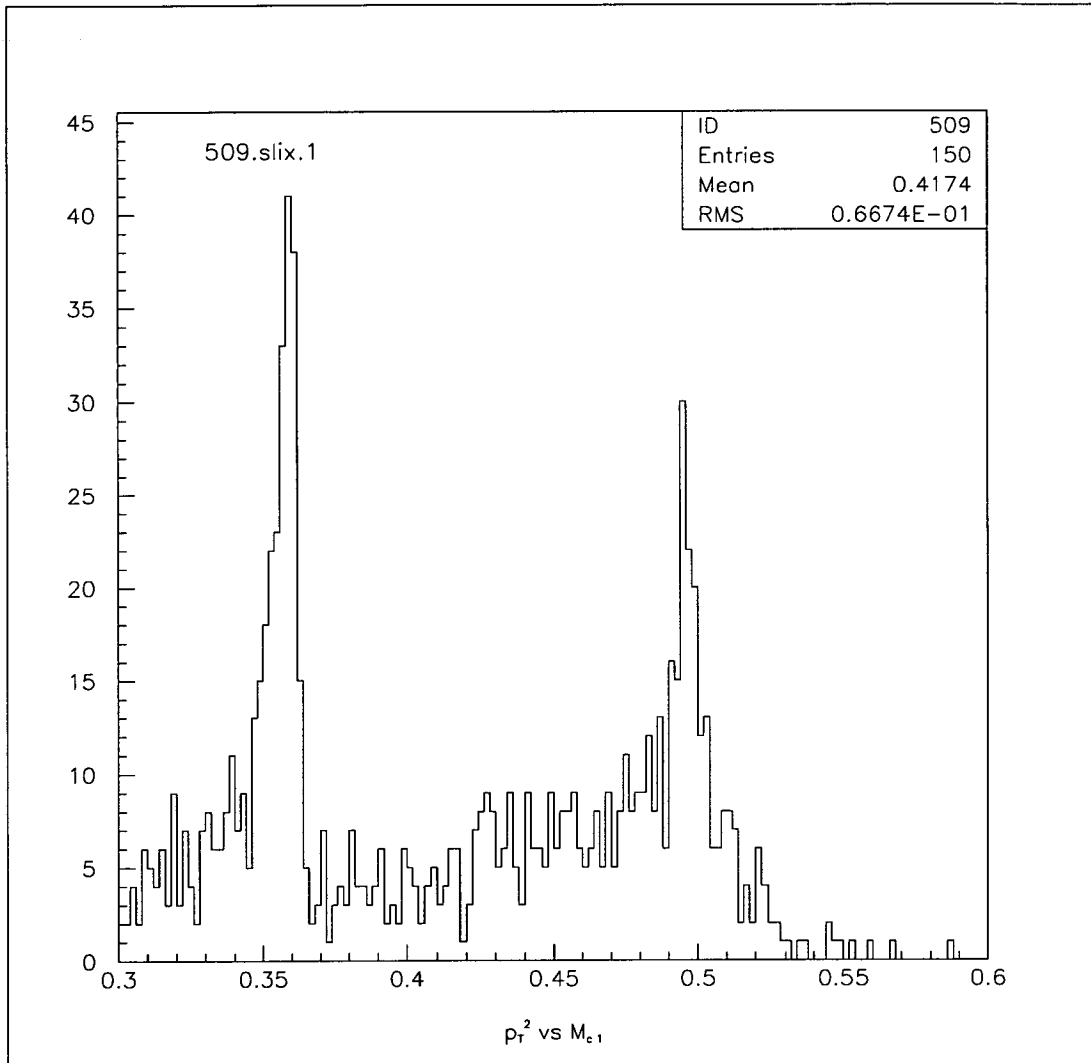


Figure 1.6: Section Through the Previous Figure Showing a Contribution from CP Violating Decays.

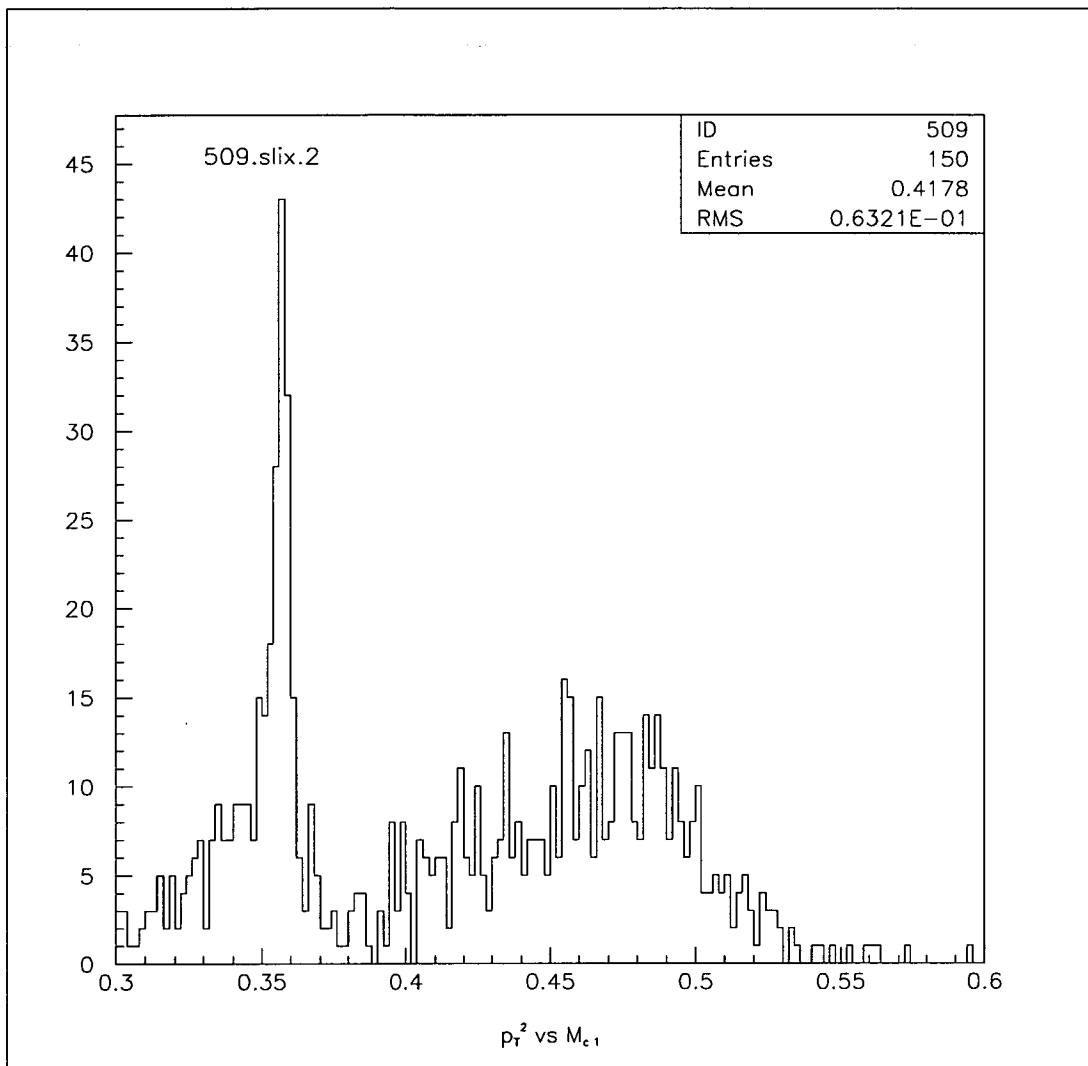


Figure 1.7: Section Through Figure 1.5 Showing no CP-Violation at Higher P_T^2 .

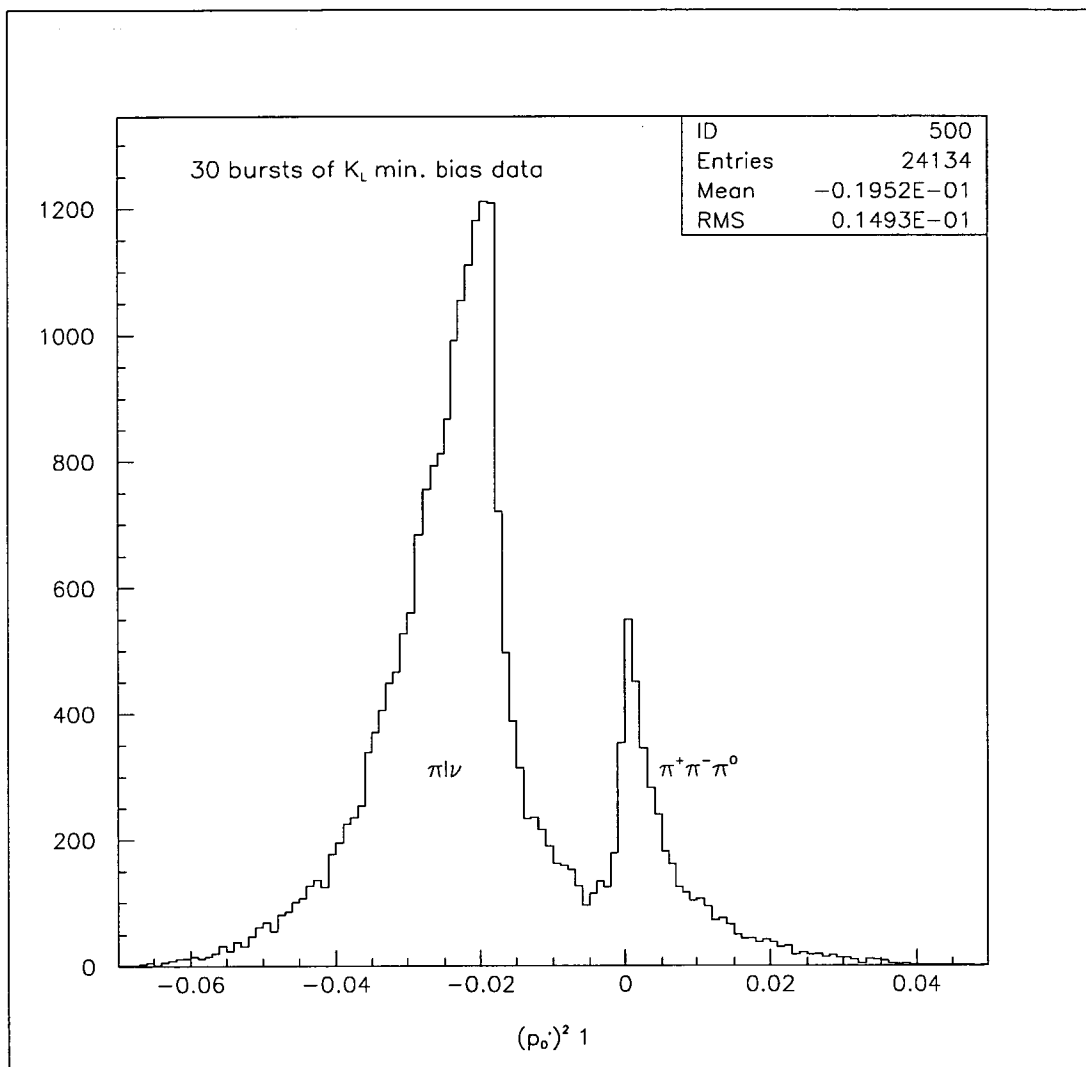


Figure 1.8: Distribution of $(p'_0)^2$.