

This is to certify that the

thesis entitled

Control of the Hardware Trigger at the
DØ Experiment

presented by

Philippe Laurens

has been accepted towards fulfillment
of the requirements for

M.S. degree in Physics

A handwritten signature in cursive script, appearing to read "M. A. Abolins", written over a horizontal line.

Major professor

(M.A. Abolins)

Date 11 - 12 - 86



RETURNING MATERIALS:
Place in book drop to
remove this checkout from
your record. FINES will
be charged if book is
returned after the date
stamped below.

JUL 26 2015

CONTROL OF THE HARDWARE TRIGGER
AT THE DO EXPERIMENT

By

Philippe Alain Laurens

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Physics and Astronomy

1986

ABSTRACT

CONTROL OF THE HARDWARE TRIGGER AT THE D0 EXPERIMENT

By

Philippe Alain Laurens

This thesis describes the control of a large electronic system. The system presented in this document is the hardware first level trigger of the large scale High Energy Physics experiment called D0. The construction of D0 is, at this date, still in progress. The experiment will be located at FERMI NATIONAL ACCELERATOR LABORATORY near Chicago, Illinois.

From a general point of view, I will introduce and explain the functions and characteristics of a trigger system.

In the special case of D0, different problems encountered are presented. The solutions and choices made in the design of the trigger are justified.

The discussion will focus on the control of the trigger system. The design and construction of the fast and reliable internal communication structure and the connection with its control processor have been my principal concern over the last fifteen months and will be described in detail.

ACKNOWLEDGMENTS

First, I would like to thank Professor Maris A. Abolins, who made possible these fifteen months of work and study at Michigan State University.

I would like to express my gratitude to Dan Edmunds and Donn Shull, who generously provided guidance, patience, and friendship.

And finally, I want to thank Sarah Lindsay for her love and support.

TABLE OF CONTENTS

LIST OF TABLES.....	v
LIST OF FIGURES.....	vi
I. INTRODUCTION.....	1
I.1. FERMI NATIONAL ACCELERATOR LABORATORY.....	1
I.2. THE DO EXPERIMENT.....	4
II. FUNCTIONNALITY AND PERFORMANCES OF A TRIGGER.....	6
II.1. FUNCTIONNALITY.....	6
II.1.1. TIMING.....	6
II.1.2. DECISION.....	7
II.2. NECESSITY.....	7
II.2.1. DATA ACQUISTION TIME.....	8
II.2.1.1. DETECTION AND STORAGE.....	8
II.2.1.2. DATA COLLECTION.....	9
II.2.1.3. DATA PREPROCESSING AND STORAGE.....	9
II.2.2. BEAM CROSSING RATE.....	10
II.3. QUALITY ESTIMATION.....	12
II.3.1. CRITERIA.....	13
II.3.2. SOFTWARE.....	15
II.3.3. HARDWARE.....	16
II.3.4. COMBINATION.....	17
II.4. CHOICE OF TRIGGERING CRITERIA.....	17
II.4.1. PARAMETERS.....	17
II.4.2. EXPERIMENTATION TOOLS	18
III. THE DATA ACQUISITION AT DO.....	19
III.1. DATA READOUT.....	19
III.2. DO TRIGGERS.....	21
III.2.1. LEVEL 0 TRIGGER.....	22
III.2.2. LEVEL 1 AND LEVEL 1.5 TRIGGER.....	22
III.2.2.1. DETECTOR TRIGGERS.....	22
III.2.2.2. TRIGGER FRAMEWORK.....	22
III.2.3. LEVEL 2 TRIGGER.....	26
IV. INPUTS TO THE DO FIRST LEVEL TRIGGER.....	29
IV.1. DATA.....	29
IV.1.1. DATA TO INCLUDE IN THE TRIGGER DECISION.....	29
IV.1.2. DATA NOT INCLUDED IN THE TRIGGER DECISION.....	29
IV.2. CONTROL.....	30
IV.2.1. CONTROL LINES.....	30
IV.2.2. CONTROL COMMANDS.....	32

LIST OF TABLES

Table 1. COMPARISON OF PROPAGATION TIME THROUGH A BASIC LOGIC NOR GATE FOR DIFFERENT LOGIC FAMILIES.....	44
---	----

LIST OF FIGURES

Figure 1.1. THE TEVATRON.....	2
Figure 3.1. PROGRAMMABLE ANDOR NETWORK.....	24
Figure 5.1. TRIGGER FRAMEWORK INPUTS AND OUTPUTS.....	37
Figure 7.1. CARD DIMENSIONS.....	46
Figure 7.2. CONTROL COMPUTER BUS.....	48
Figure 7.3. FIRST LEVEL TRIGGER BUSSES.....	49
Figure 7.4. TIMING AND SYNCHRONIZATION BUS.....	50
Figure 7.5. TYPICAL RACK OF THE FIRST LEVEL TRIGGER SYSTEM.....	52
Figure 7.6. ADDRESS DECODING AND DATA PATH.....	53
Figure 7.7. SPECIFIC MOTHER-BOARD BUS.....	54
Figure 7.8. COMMUNICATION INTERFACE FUNCTIONS.....	58
Figure C.1. BUS BUFFER CARD LAYOUT.....	77
Figure D.1. MOTHER-BOARD DRIVER CARD LAYOUT.....	78
Figure E.1. COMMUNICATION INTERFACE CARD LAYOUT.....	79
Figure E.2. COMMUNICATION INTERFACE CARD INPUTS AND OUTPUTS.....	80
Figure E.3. VMX DRIVER CARD LAYOUT.....	81
Figure F.1. BUILDING A VAXELN SYSTEM WITH VAX/VMS.....	85

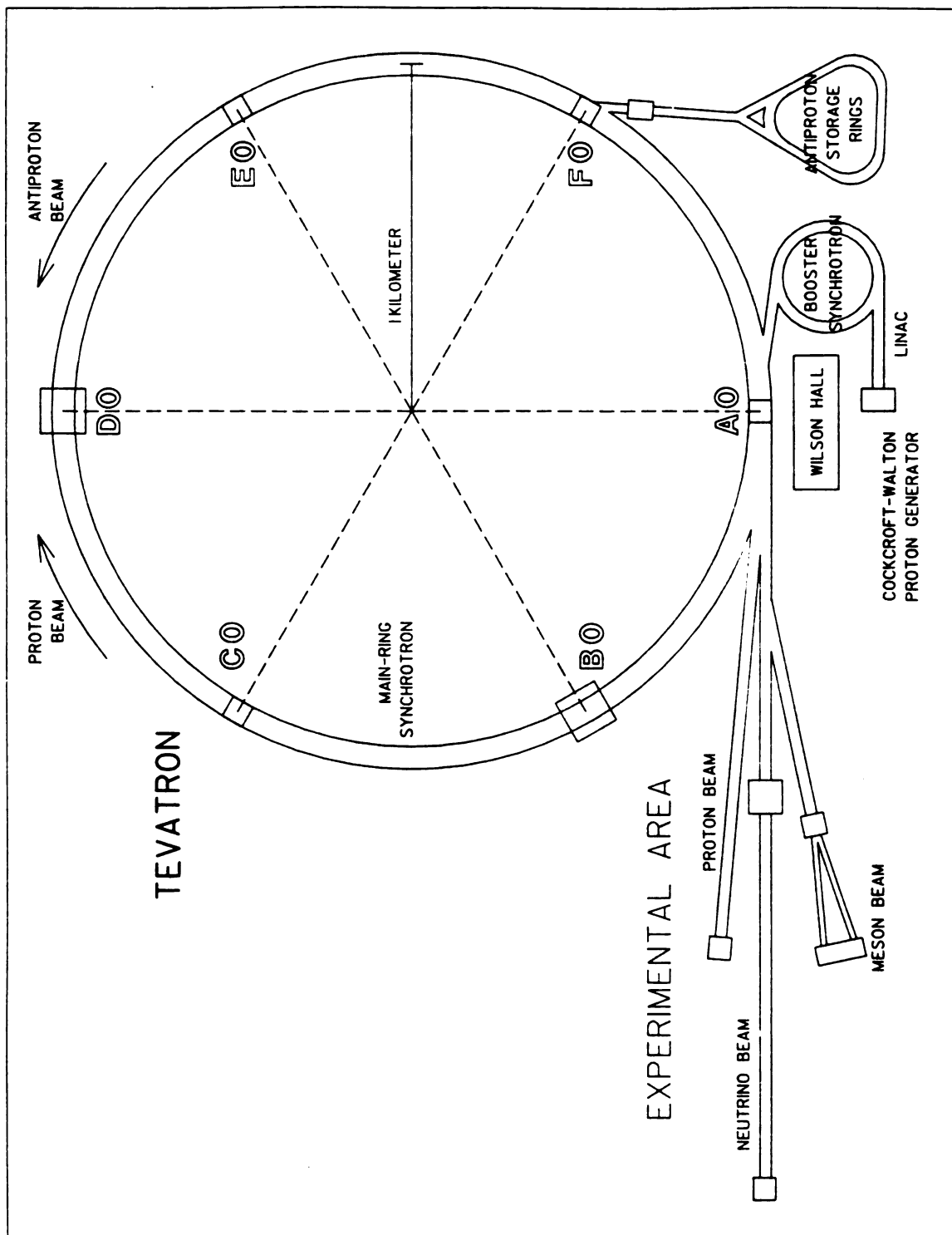
I. INTRODUCTION

I.1. FERMI NATIONAL ACCELERATOR LABORATORY

The first synchrotron built at the National Accelerator Laboratory (Batavia, Illinois) began operating in 1972 and could produce a single beam of protons with a maximum energy of about 400 billion electron volts (400 GeV). One million electron volt protons (1 MeV) are produced by a Cockcroft-Walton generator, accelerated up to 200 MeV in a 145 meter linear accelerator and led into a booster synchrotron which increases their energy to 8 GeV (figure 1.1). The proton beam is then switched to the main ring (2 Kilometers in diameter) and accelerated up to 400 GeV. After acceleration, the beam is deflected from the ring to the different experimental areas. For further description of the original accelerator see reference 2.

In the same tunnel, a new synchrotron ring has been added to the main ring. The new ring uses superconducting magnets which double the maximum strength of the magnetic field and hence the maximum energy of the beam. The new proton synchrotron took the name Tevatron in relation with the attainable energy of one trillion electron volts (1TeV).

The Tevatron has also been modified to enable it to run as a proton-antiproton collider. Two beams of 1TeV particles are travelling in the same beam pipe, but in opposite directions. The counter rotating beams are not continuous, but are made of a succession of groups of particles travelling in packets. They are synchronized with respect to each other and collide at definite geometrical points depending upon the



THE TEVATRON
Figure 1.1

number of packets in the ring.

The antiprotons are produced at a slow rate from protons accelerated in the main ring synchrotron and directed to a fixed target. Some of the particles produced in the collision are antiprotons. They are sorted, cooled and stored in a storage ring. The cooling process involves reducing their relative random velocities in order to produce antiprotons with identical energies. They are also divided up into packets (called bunches) before being introduced back into the main ring and then into the Tevatron in the opposite direction of the proton packets.

The proton antiproton collider existing at Fermilab is a unique facility, the energy of 2TeV is not attainable in any other existing machine. It is superior to the CERN collider because it has a higher energy, higher luminosity and a higher antiproton production rate. For further description of other accelerator see reference 3.

An experiment which is intended to look at the products of proton-antiproton collisions must be built on the accelerator ring and the detector must surround the beam pipe at the precise interaction point.

The accelerator ring is made up of 6 identical sectors. They are referenced by a letter, A through F. An additional digit is appended to the letter to identify geographical locations within the sector. With 6 proton packets and 6 anti-proton packets, the interaction points are located at A0, B0, C0, D0, E0, F0 (figure 1). The protons are led into the machine at the A0 point.

Only two of the 6 interaction points will be used for large experiments. At these two points the beams are strongly focused to obtain higher interaction probability. Several experiments of smaller scale are already scheduled or are in place on the collider ring at C0, E0, and F0. At the B0 location a large scale experiment called CDF is already in place.

I.2. THE D0 EXPERIMENT

The experiment discussed here is being built at the geographic site D0. The experiment is known by the name D0 and the Fermilab experiment number E740.

The appendix A gives the list of all the institutions officially participating in D0. It also includes the names of the high energy physicists from Michigan State University having responsibilities in D0.

Building an experiment like D0 is a very expensive project. The cost of D0 is currently estimated to be 50 million dollars. In order to justify the construction of D0, three kinds of arguments have been presented. First, such an expensive accelerator must be as fully exploited as possible. Second, D0 is expected to bring confirmation of new results and discoveries with improvement in precision. Last, but not least, a rich harvest of new results and discoveries is expected at this high energy regime (see reference 1).

An additional use of D0 is that it will test the general design of detectors for the next series of high energy experiments. An even larger collider is being designed, it uses two rings of superconducting magnets with a 52 mile circumference. The accelerator is called SSC

(standing for Superconducting SuperCollider) and is designed to accelerate two counter rotating beams of protons up to 20 TeV. The funding necessary for the construction of SSC is still under discussion. For further description see reference 4.

The detector being built for D0 has three major components. The central detector includes a Vertex Detector, a Transition Radiation Detector, and Drift Chambers. The Calorimeter, segmented into 3 pieces, surrounds the central detector. The outer-most detector consists of the muon bending magnets and the Muon Detector. Every detector is designed to cover as large a solid angle as possible. The whole detector and the front-end electronics are placed on a rolling platform. The detector can be placed in the ring or transported in the assembly hall.

II. FUNCTIONALITY AND PERFORMANCE OF A TRIGGER

II.1. FUNCTIONALITY

The description of the main functions of a trigger in a data acquisition context will show the crucial role of the trigger system in the performance of D0.

The functionality of a trigger can be subdivided into two different categories: timing of the acquisition sequence and decision making.

II.1.1. TIMING

The first and probably most familiar function of a trigger is the ability to initiate and synchronize the data acquisition sequence.

As an analogy we can take the example of an automatic camera. In this case, the request will be the action of the operator on the mechanical trigger button. The data acquisition sequence will consist in exposing the film to the proper amount of photons.

In response to the operator's request, the electronic trigger controlling the camera is expected to activate the photosensitive cell, select the proper lens aperture, open the shutter, start integrating the flux of light passing through the lens, close the shutter, wind the roll of film up to the next exposure and return to a waiting state, ready to answer the next operator's request. These activities must be performed in a certain sequence in order to produce the expected picture and it is the function of the trigger to initiate and synchronize them.

II.1.2. DECISION

The decision-making function can be represented as the ability of the trigger to select and update the state of its outgoing control signals upon request. The decision will depend upon the values of the input data and control signals at the time of the solicitation. The answer must be unique and predictable according to the set of rules the system has been designed and programmed to follow.

Let us again use the analogy of the camera: after the operator's request, the internal trigger of the camera is expected to select the optimum aperture of the lens according to the current light intensity and respecting the different selections made by the operator. The system may then decide to fire the flash if the light level is too low, or possibly warn the user or even refuse to take the picture instead of wasting the film when the required conditions are not met.

II.2. NECESSITY

There are two timing constraints along with their orders of magnitude which need to be presented here in order to illustrate the critical requirements of a trigger for D0. The first notion to introduce is the data acquisition time including data collection, on line processing, and storage time. The second number presented and justified will be the beam crossing rate which determines the decision time.

II.2.1. DATA ACQUISITION TIME

Three steps must be distinguished. The data acquisition time is to be subdivided into a detection time, a data collection time and a data pre-processing time. The final stage will be the storage on a magnetic tape of a self-sufficient and exhaustive description of the selected event.

II.2.1.1. DETECTION AND STORAGE

This is the first step inherent to the data acquisition.

Detection

The data acquisition starts in the detectors. In the calorimeter, for example, the ionization of the liquid argon molecules, caused by the passage of high energy particles, is sensed on copper-plated fiberglass boards. The information appears as a voltage at the output of the preamplifiers a few 100 nanoseconds after the collision.

Analog Storage

The charge must then be detected and stored before the next beam crossing in the detector. This is typically achieved with an analog storage of the detected voltage across a capacitor. There will be one capacitor for each signal requiring storage. There are hundreds of thousands of channels at D0. This storage is accomplished in parallel on all the channels and repeatedly at every beam crossing (i.e., every 3.5 microseconds).

II.2.1.2.DATA COLLECTION

A/D Conversion

The next step is the data type conversion from analog to digital for every channel to be read out. This operation is not done in parallel on the total number of detector channels, but sequentially in subsets of the full detector. It would indeed be very costly and space-consuming to place one A/D converter per channel. As we will see later, this operation is also not performed after every beam crossing.

Sequential Readout

Once digitized, the information will be sequentially collected from all the parts of all the detectors and gathered in a block of data called an event. This operation requires a few milliseconds.

II.2.1.3. DATA PREPROCESSING AND STORAGE

The on-line data preprocessing sequence can be defined as the task of transforming the collected data from raw and difficult to use to corrected and meaningful. This possibly requires certain calibration or offset corrections, reordering, reformatting, adding extra parameters and context variables, and producing physics variables. Simultaneously, physics variables are extracted and monitored to help control the experiment.

The final step of the data acquisition will be the storage and archiving of the preprocessed data. This will take place on the host computer which is crucial to the control of the experiment. In order not to saturate this computer whose control and monitoring tasks are

highly essential to the experiment, the host computer data acquisition rate should not exceed about 1 processed event per second (1 Hertz).

Past this point, the recorded event will be considered as usable for physics calculation and off-line analysis.

II.2.2. BEAM CROSSING RATE

The acquisition time above justified applies only to the recording of those events considered as containing "interesting physics". I should immediately point out that this is a very subjective notion which will probably evolve over the live time of the experiment, but which implicitly defines uncommon and infrequent events.

It would, of course, not be reasonable to try to execute a complete data acquisition sequence as described above for every beam crossing in the collider. It would limit the maximum rate of processed events to approximately 1 Hz. Such a mode of data acquisition would essentially imply observing, without learning, well-known and well-understood events. It would take several years of processing time before one could obtain some knowledge about those interesting events which are also statistically extremely rare.

The obvious solution to this problem is to increase the beam crossing rate and the number of particles per bunch in the tevatron. This will make the statistically rare events happen more frequently. Unfortunately, it will also require a very efficient trigger system capable of deciding very quickly whether an event is worth recording or not. The data acquisition would only occur for those events considered interesting.

The intensity of the beam is characterized by a variable called luminosity. The luminosity L of head-on colliding beams is defined as the product of the number N_1 of particle per bunch in one beam multiplied by the number N_2 of particles per bunch in the other beam and by the frequency f at which the two beams collide and divided by the cross-sectional area A .

$$L = f \times \frac{N_1 \times N_2}{A} \quad (1)$$

The disadvantages and limitations of an increase in the luminosity are of two kinds. Increasing excessively the number of particles per bunch will make statistically more frequent the complex and multiple interactions. If too many inelastic interactions happen within a single beam crossing, the products are much harder to track and recognize. The event will be hard to analyze and reconstruct from the data, especially because of the finite spatial resolution of the detectors. Multiple inelastic collisions cannot be differentiated from a single collision if the vertices are too close to each other to be differentiated.

Increasing the beam crossing rate will consequently decrease the time left for the trigger decision. The number of variables that can be processed and taken into account in the shorter period of time between beam crossings will get smaller, thus tending to degrade the ability of the trigger system to recognize the interesting events.

An increase in interaction rate will also induce a higher proportion of events ignored during the data acquisition dead time. The dead time of the data acquisition is due to the few milliseconds duration of the acquisition sequence. All information about every beam

crossing happening while the data of a previous event is being collecting is ignored and lost until the data acquisition is freed again and able to record a new set of data (this subject will be further discussed in chapter III).

The beam crossing frequency at Fermilab is about 300,000 Hz, corresponding to a time between interaction of 3.5 microseconds. The luminosity in the collider at the D0 site is expected to be:

$$L = 10^{30} \text{ cm}^{-2} \times \text{sec}^{-1} \quad (2)$$

for a beam diameter focused to about one millimeter. At this luminosity, non-elastic collisions are expected to occur with an average rate of 50,000 Hz and "interesting physics" is expected at a rate lower than 1 Hz.

We can now define the goal of the trigger at D0 as lowering the high-beam crossing rate down to a suitable acquisition rate without causing dead-time by on-line sorting and discarding of all uninteresting events.

II.3 QUALITY ESTIMATION

We have shown in II.2 how important the role of a trigger is to the D0 experiment. In fact, an exceptionally efficient trigger needs to be built. In order to evaluate the quality of this trigger, let us now scan different criteria and review what can be expected from hardware and software types of triggers.

II.3.1. CRITERIA

There are a number of obvious parameters to take into account during the evaluation of a trigger or any other device. We will not deeply discuss here the global cost, the complexity or the physical size, which are some of them. Let us instead focus on the qualities which are special to the trigger system.

Decision Time, Dead Time

We have already seen that the decision time will be one of the designer's constraints. We have also defined the dead time as a variable to minimize.

Reliability and Testability

Another quality which needs to be emphasized is the extremely high reliability expected from such a device.

Indeed, the whole experiment requires a running trigger structure. If a detector or a portion of the data acquisition quits working, the rest of the detectors and the data acquisition can keep gathering data. If a part of the trigger structure quits working, the rest of the experiment will lose synchronization or will be saturated with unsorted events. Clearly, the whole experiment will stop when the trigger structure fails. Therefore, the trigger system must be extremely reliable. It should also be designed to continuously perform auto-tests and in case of failure immediately diagnose what parts or cards need to be replaced.

Programmability, Flexibility

The kind of physics studied at D0 will evolve over the lifetime of the experiment. Some of the parameters of the detectors will be modified or new parts will be added, in other terms, the trigger will be expected to be receptive to different values or types of parameters. Then the trigger system at D0 (like any other incompletely defined problem) will have to be flexible, programmable, and expandable in order to follow almost any possible modification of its requirements.

Efficiency

After these general rules, the quality of the decision must now be evaluated: the trigger must be efficient.

The trigger must recognize all the "interesting events" without missing any but it must also make the minimal number of mistakes and select only these events. The more information is processed in making the decision, the more complex the system will be but also the more complete and accurate the decision will be. We will later come back to the programming and evaluation of the trigger decision.

This notion appears under two aspects. The trigger efficiency will depend upon the way the trigger is built, but also upon its programming and configuration. Exactly like a problem solved on a computer, the quality of the answer depends both upon the quality of the machine and the accuracy of the algorithm used.

We will now review what can be achieved with the two different types of triggers.

II.3.2. SOFTWARE

The best and most complete judgment which can eventually be formulated about an event will certainly come from an algorithm running on a powerful computer having access to all the information collected about this event.

It has the advantages of being infinitely programmable, flexible and expansible, but the data acquisition structure must allow the computer access to the data of every event to be processed.

If it is chosen to build a trigger from a program running on line, a large amount of time will be spent writing and testing the program, but the work involved may be deducted from the even larger task of writing the off-line software analysis programs. For a triggering program, the analysis is pushed only far enough to make a judgment about the content of the event without fully processing it. The long development time involved for a triggering algorithm can hence be considered as needed in any case. The partial information obtained will be very useful to classify the events and will be added to the raw data at archiving time.

The (very strong) limitation on a software trigger comes from the slow decision time. The amount of data to be processed is very large and all operations are performed sequentially. A complete data acquisition sequence is needed for each processed event. All these facts lead to a maximum processing rate of a few events per second as

explained before. This must be compared to a beam crossing rate of 300,000 Hz.

Despite all the advantages described, a software trigger is not sufficient in the special case of D0.

II.3.3. HARDWARE

The strongest constraint on the design of the trigger system is the processing rate of 300,000 Hz or an event every 3.5 microseconds.

Once the detection time is subtracted from this already short time, the time left for a decision does not allow any serial or iterative process. All operations must be performed independently and in parallel on a hardware device specially designed for this purpose.

Moreover, the information must come directly from the detectors, bypassing the rest of the data acquisition, reserving a complete readout only for the selected events.

The flexibility of a hardware trigger is intrinsically limited. The design must be done in a very careful way in order to build the most general purpose device possible.

A fast reaction time can be achieved with a hardware type of trigger. It is, however, counterbalanced by a higher complexity, a larger size, and a higher cost. The decision made will be limited by the number of variables processed, and the efficiency of such a device will not be one hundred percent. A hardware trigger is not sufficient for an experiment like D0.

II.3.4. COMBINATION

Neither a software nor a hardware trigger can match the specifications described. The appropriate solution is a combination of the two using the following scheme.

A hardware trigger makes an incomplete decision within the 3.5 microseconds. A readout is initiated for each event selected by the hardware trigger. A software trigger makes the final and accurate decision. The events passing the two trigger decisions are sent to the host computer for final processing and storage.

The hardware trigger of D0 is called the First Level Trigger; it is expected to lower the 300,000 Hz interaction rate down to an average rate of about 200 Hz. The software trigger is called the Second Level Trigger; it is expected to make the final triggering analysis to achieve a final rate of a few Hertz.

II.4. CHOICE OF TRIGGERING CRITERIA

II.4.1. PARAMETERS

Since the amount of information and the number of operations is limited, the efficiency of such a system will very strongly rely on the choice of the observed variables and criteria.

Different Possible Parameters

There are many criteria that may be included in the trigger decision. The peak of energy measured within a certain solid angle, the energy sum over the whole set of channels, the number of detected jets

of particles, the way the different detector channels are averaged and grouped before being submitted to the trigger decision are some of the available possibilities. They must be studied and evaluated, and a choice must be made before the corresponding hardware can be designed.

II.4.2. EXPERIMENTATION TOOLS

The computer simulation plays a very important role in the design of the detectors and the design of the triggers.

Events are generated with a program called ISAJET (reference 14). Other programs are used to simulate the response of the detectors to these events. The geometry and physical characteristics of the detector components are given as input to the program which then predicts the energy deposited in every cell.

Programs have been written at MSU to evaluate the performance of the trigger system. They associate Monte Carlo event generation with detector simulator programs to compute the efficiency of the trigger in recognizing the events as a function of the variables observed and the chosen triggering criteria (ref. 5,7).

Monte Carlo studies are also used to evaluate the second level trigger and the proficiency of the off-line analysis software to analyze the events. They are again used to confirm or analyse the conclusions deduced from the collected events.

III. THE DATA ACQUISITION AT D0

The D0 data acquisition system is described below with specific information presented about the calorimeter section.

III.1. DATA READOUT

At every beam crossing, the information coming from every channel of the calorimeter detector is stored on a capacitor. Most of the time, this information will be overwritten at the next beam crossing. When the trigger decides to initiate a data acquisition sequence, the voltage across the capacitor must be saved to be read out. It must not be overwritten by the next events until all capacitors are read out.

Some dead time appears as soon as the events cannot be recorded by lack of storage capacitors. Indeed, the number of lost events is not negligible. With a data collection time of the order of 10 milliseconds there would be thousands of events lost at every data acquisition sequence.

In order to diminish the importance of this dead time, the calorimeter detector is designed to be able to hold two sets of data before being saturated. This is called the Double Buffering of the data acquisition.

The data is recorded on one set of capacitors until the trigger decides that the information about the last event should be saved and read out. At that time, the data will continue being recorded on the other half of the double buffer. If the other half is already holding data and therefore not capable of recording new events, the data will

quit being recorded and will be lost, creating dead time. Because the interesting events occur randomly in time, missing useful data will inevitably happen, even with a double buffered system.

The probability of saturating two sets of capacitors is now much smaller than in the case of a single set; useful data will be lost only when three "interesting events" occur over a time interval smaller than the data readout time. One could imagine further decreasing this dead time by having even more sets of storage capacitors, but the decrease in dead time induced does not compensate for the increase in cost, and two sets of capacitors have been considered sufficient.

I here will briefly digress in order to prevent confusion. There is, in fact, another place in the data acquisition system where a double set of capacitors is used, but for a completely different purpose. They are part of the so called before-after differentiation (base-line subtractor). In this case, one of the capacitors holds the voltage before the interaction, and the other capacitor holds the voltage after the interaction. The voltage difference between the two capacitors will, therefore, hold the information about the interaction time only. This voltage will not be affected by previous interactions.

After detection, the information takes two different paths. The first path leads to the storage of the analog data on the capacitors of the double buffer. The other path goes directly to the trigger which will receive and operate on the analog signals independently of the rest of the data acquisition system.

The first level trigger will be described in detail in the next section. Let us just assume here that depending upon the decision of the trigger, the data acquisition will stop with the storage on the capacitors or will be pushed further. If such a decision is made, the analog voltage must be converted to its corresponding digital value and be read out.

The data acquisition is divided into 32 different geographical sections. Each of these sections receives its own control signal from the trigger system to initiate the A/D conversion. This operation will be performed sequentially over a set of channels to reduce the cabling and the size and cost of the converter devices. The converted data is placed in contiguous blocks of memory.

The digitized data, spread over all the different sections, is then collected on eight data cables and forwarded to the second level trigger. The second level trigger will be described in the next section.

The event is further processed by the software trigger. If the decision of the second level trigger is also positive, the block of data is sent to the host as a DECnet message over the ETHERNET link.

III.2. D0 TRIGGERS

The trigger system of the D0 experiment is divided into three different layers. They are called Level 0, Level 1, and Level 2 Trigger, respectively.

III.2.1. LEVEL 0 TRIGGER

The Level 0 trigger generates a veto signal for the Level 1 trigger when no proton-antiproton collision occurs in the detector. With an accurate timing of the beams entering both ends of the detector, the position of the vertex (interaction point) inside the detector is also computed and will be used as an input to the Level 1 trigger.

III.2.2. LEVEL 1 AND LEVEL 1.5 TRIGGER

The first level trigger is the combination of the detector triggers and the trigger framework.

III.2.2.1.DETECTOR TRIGGERS

The detector triggers analyze the data coming directly from the detectors and produce outputs sent as inputs to the trigger framework. The calculations made evaluate variables such as the total deposit of energy, the peak value of the energy, the angular position of the jet(s),etc. The efficiency of the trigger depends on the choice of the calculations and their accuracy.

A detector trigger is, for example, the calorimeter trigger which will be built by Michigan State University and integrated in the same structure as the trigger framework. The other detector triggers are the TRD (Transient Radiation Detector) and the muon trigger.

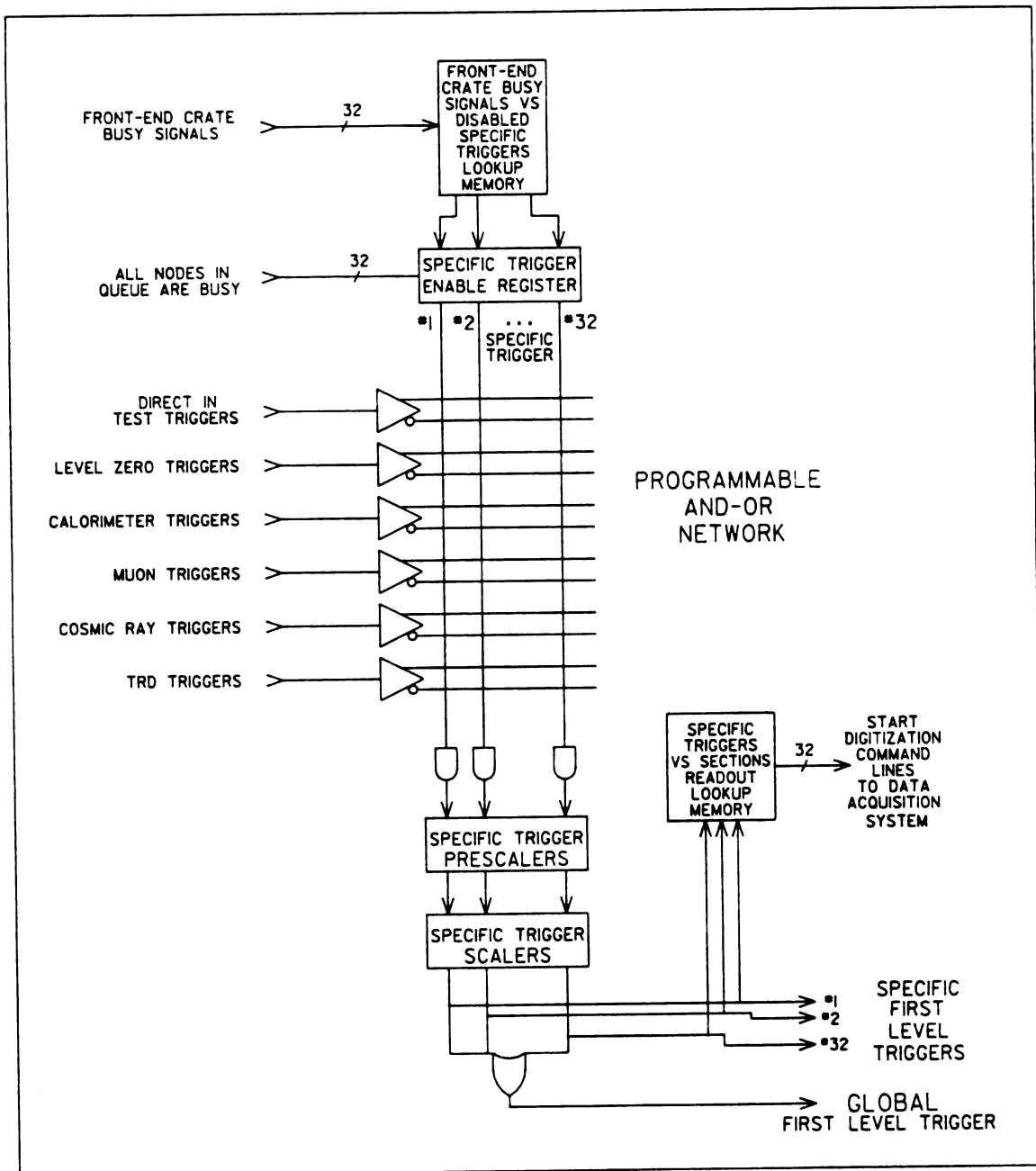
III.2.2.2.TRIGGER FRAMEWORK

The trigger framework is the central part of the trigger system which receives all the control signals and all the information from the detector triggers (reference 6). The trigger framework makes the trigger decision and provides the timing and control signals.

At the D0 experiment, the trigger decision is not unique. There are 32 specific trigger decisions made in parallel and a global trigger decision which is the logical sum of the 32 specific trigger decisions. Or, in other words, the global trigger decision is positive as soon as at least one of the specific trigger decisions is positive.

The subdivision of the first level trigger into 32 specific trigger decisions must not be confused with the geographical subdivision of the data acquisition in 32 different sections. The specificity of the trigger decision holds a physical meaning, but the subdivision of the data acquisition is just a logical grouping of detector channels. There is a programmable conversion table in the first level trigger system which translates the 32 specific trigger decisions into 32 start-digitizing commands sent to the data acquisition sections.

The trigger decision is essentially produced by a large and/or gate array (figure 3.1). Every channel of every specific trigger input is programmable. there are 256 inputs to the network and all specific trigger decisions are made independently and in parallel. A specific trigger can be programmed to require a certain logical state on each particular channel, or certain inputs can be programmed as not participating in the trigger decision.



PROGRAMMABLE ANDOR NETWORK
Figure 3.1

The need for an intermediate trigger level has been anticipated. This is an extension of the first level trigger called the Level 1.5 trigger. The specific triggers can be programmed to wait for more information after a positive specific trigger decision at Level 1.

If the complement of information received brings a negative Level 1.5 decision, a signal is sent to clear the previous Level 1 decision and to abort the event processing. In case of a positive Level 1.5 decision, the processing of the event completes normally and the specific trigger is automatically reenabled to process the next event. The muon detector is an example of a device which cannot give full information within 3.5 microseconds.

It should be noted that the Level 1 trigger must send a positive decision before the next beam crossing time and then eventually negate it. It cannot wait and send an accurate decision because the information would be overwritten at the next beam crossing time.

The information passed to the trigger framework is also forwarded to the second level trigger in order to permit verification and refinement of the first level decision and facilitate access to the data read by the data acquisition system.

The trigger framework also includes a timing generator which generates timing and synchronization signals for the rest of the experiment. These timing signals are programmable and are extremely flexible. For example, a timing generator card can be synchronized with the accelerator clock or can also use its on-board oscillator to replace the accelerator clock for test purposes during the time the Tevatron is

not functioning. The control signals provided are also programmable in a fixed sequence through a PROM or can as well be derived from external signals.

The first level trigger framework and the calorimeter trigger are programmed and controlled by a microcomputer. The computer used is a MicroVAX running programs under a DEC VAXELN system downloaded by the host computer (see appendix F). This computer is referred to as the First Level Control Computer. Its function will be further described in the next chapter.

III.2.3. LEVEL 2 TRIGGER

The First Level Trigger described in the previous paragraph is expected to reduce the interaction rate of 300,000 Hz down to a triggering rate of about 200 Hz. Hence, it is necessary to lower this rate by another factor of 200 before sending the data to the host computer. A computer analysis of the event is now necessary to perform a finer and final decision. Two hundred events per second correspond to a processing time of 5 milliseconds per event. During such a short period of time a single processor could barely reproduce the Level 1 trigger calculation.

Instead of being a single processor, the second level trigger will be a network of microprocessors working in parallel. The network has been chosen to be 50 MicroVAX II processors running their programs under a VAXELN system downloaded from the host computer (see appendix F).

The different processors are called the second level trigger nodes and are linked by an Ethernet link. A node processes until completion one event at a time. The nodes are divided in specialized subsets constituting different queues. After making a decision about one event, the processor will either send the set of data to the host computer, or discard the event and immediately start processing the next event waiting in its queue.

The manager of the second level trigger nodes will be another MicroVAX running under VAXELN. It is called the Second Level Trigger Supervisor. The second level trigger supervisor is connected to the second level processors by a set of control lines driven between DRV11-J cards connected to the Q-Bus of the Microvax nodes and by the Ethernet link.

The second level supervisor receives the 32 specific trigger decisions. As soon as an event is selected, and depending upon the specific kind of trigger(s) involved, the supervisor assigns the processing of the event to one of the 50 second level nodes.

The 50 independent nodes are not totally equivalent, but may be loaded with different specialized programs better designed for the analysis of a certain type of event. The repartition of the number of nodes per type of event is chosen in order to match the statistical rate of each event type and thus minimize the dead time. A queue is assigned to each version of triggering software and is characterized by the number of busy and idle nodes. The management of the queues is the task of the second level supervisor.

The First Level Trigger Data BLock is received by the selected node along with the rest of the data collected by the data acquisition system. The second level node can start processing the Data Block received within 1 millisecond after the beam crossing causing the event before it receives all the data. Because of the fixed format of the data block, the processor is able to very quickly learn the parameters of the event (e.g. the angular position of the jets, etc.) and thus save precious time by reducing the task of unpacking the data from the main acquisition system.

If the event passes the second level trigger decision, the complete block of data (i.e. the data block plus the data from the main acquisition system) is transported to the host computer as a DECnet message over the Ethernet link.

IV. INPUTS TO THE DO FIRST LEVEL TRIGGER

The first level trigger receives data and accepts control signals to include in the trigger decision. It also accepts data to transmit to the second level trigger.

IV.1. DATA

IV.1.1. DATA TO INCLUDE IN THE TRIGGER DECISION

The input data for the first level decision comes from the detector triggers. It forms the input to the programmable and/or network. Each bit of this digital information is compared in parallel for the 32 specific triggers to the requirements programmed into the network for this channel.

A specific trigger will make a positive decision if the programmed conditions are simultaneously satisfied for every input to the network. There is no intrinsic limit to the number of positive specific trigger decisions for a given beam crossing. This number will depend upon the programming of the network and on the nature of the event.

IV.1.2. DATA NOT INCLUDED IN THE TRIGGER DECISION

The first level trigger is designed to accept external data and to incorporate the information in the first level trigger data block described in the next chapter. The data block will be transmitted to the second level nodes and will be part of the data received by the host for archiving.

This mode of data collection has two advantages: first, it is a convenient way to include a small amount of dispersed data. Second, the first level trigger system is the only place in the whole experiment where information about the beam crossing which directly preceded the selected event is recorded. This information could be important in understanding the influence of previous crossings on the trigger decisions.

An example of this type of information is the vertex location detected by the scintillator monitoring the vertex location.

IV.2. CONTROL

IV.2.1. CONTROL LINES

The first level trigger accepts control signals from various parts of the experiment. For example, it is necessary to disable the specific triggers initiating the readout of a temporarily saturated section of the data acquisition system.

As described earlier, the Level 0 Trigger acts as a veto condition on the First Level Trigger which is expected not to issue any positive decision in case of a negative Level 0 result.

It is vital for the experiment that the trigger and the data acquisition system stay synchronized to successfully record the selected events. From each of the 32 geographical sections of the data acquisition system comes a veto signal called "Front-End Crate Busy". This signal is in its active state when both halves of the double-buffered storage capacitors are busy holding data currently being

digitized or not yet digitized.

In such a case, as already described, the trigger system is expected to stop sending "start-digitization" commands. The first level trigger has a programmable conversion table describing which of the specific triggers must be disabled depending upon which front-end crate is busy.

There is another case where the specific triggers need to be disabled. As described earlier the second level nodes are divided into specialized queues. When a queue is full, the second level supervisor must stop the corresponding specific triggers. There are 32 disable specific trigger lines. It is the responsibility of the second level supervisor to assert the proper lines when a certain queue is full.

The Data Block is read on a data cable identical to those collecting the digitized data from the different geographical sections of the detector. The Data Block Builder has been designed to appear to the acquisition system just like a digitization crate. The data block builder is the only device connected to the data cable serving it.

The cards controlling the data cables and the cards digitizing and holding the data from the geographical sections of the detector all match the VMEbus specifications (consult reference 8). For this reason, the data block builder records the data block on a VME memory board and signals to the data cable sequencer when the data is available on the memory card. In return, the Data Cable Sequencer Card is expected to signal when the memory board is available for a new data block.

IV.2.2. CONTROL COMMANDS

The first level trigger system is programmable by means of the First Level Control Computer. The control computer makes the translation between the commands sent and the location and value of registers to read or write in the first level trigger system in order to satisfy the request.

Typically, the commands are sent by the host computer to the control computer. The commands can consist of different types of messages.

- The host computer can request an initialization of the whole trigger system.

- The host computer can ask the control computer to load the trigger system with new physics variables.

- The host computer can request the control computer to read certain variables from the trigger system.

- The host computer can request that the control computer disable or enable a certain specific trigger.

- The host computer can demand a complete stop of the trigger system.

All these operations are performed by the control computer upon a request from the host computer. These messages are carried over the Ethernet link also used to down-line load the operating system running on the control computer.

The Second Level Supervisor also needs to send commands to the Control Computer. The specific triggers may be programmed to automatically disable themselves after every event they select. They must be reenabled by the Control Computer before they can generate another specific trigger decision. The request will come from the

second level supervisor at the completion of the processing of the event by a second level node.

These messages are also carried over the Ethernet link. They are not related to the disable specific trigger control signals described earlier and carried over a dedicated cable. A mode of operation using the automatically disabled specific triggers will introduce more dead time than a mode using the dedicated cable to control the triggers, especially if the ethernet link is saturated.

V. OUTPUT FROM THE DO FIRST LEVEL TRIGGER

The first level trigger plays a decisive role in the synchronization of the whole data acquisition system.

V.1. CONTROL SIGNALS

I already mentioned some of the control signals generated by the first level trigger system. The trigger decision itself is a combination of 32 specific trigger decisions and a global trigger decision. These signals are sent to the second level supervisor. Depending on the way the trigger system has been programmed, the physical meaning of the combination of the specific trigger decisions can be decoded by the second level supervisor computer, and a processor with a specialized program can be assigned to the processing of this particular type of event.

At the same time that the 32 specific trigger decisions are sent to the second level supervisor, the first level trigger sends start-digitization commands to the 32 different sections of the data acquisition system. The mapping of the 32 start digitization commands versus the 32 specific trigger decisions is fully programmable in the first level trigger framework. The current beam crossing number is sent, for synchronization purposes, to the data acquisition system along with the start-digitization signals.

A Clear Most Recent Trigger Signal can be sent to the data acquisition sections. Its effect is to abort the digitization sequence initiated at the last trigger decision.

All these signals correspond to the normal function of the trigger. If abnormal behavior is noticed by the control computer, a warning or an alarm message is sent over the Ethernet link to the host computer. For example, the control computer will warn the host computer when a specific trigger accumulates an abnormal amount of dead time or if a loss of synchronization is detected.

V.2. TIMING AND SYNCHRONIZATION SIGNALS

The first level trigger system generates two sets of timing signals. A private set of timing and synchronization signals internal to the trigger system is used to synchronize the operation of the different parts of the trigger. For example, the double buffering of the trigger system is controlled by three of these timing and synchronization signals. There are 32 internal timing and synchronization signals which are supported by the trigger system.

The Trigger system also generates external timing signals which are used to synchronize the operation of the other devices in the experiment. For example, the base line subtractors in the data acquisition are synchronized by means of these timing signals. There are 32 external timing and synchronization signals available to the rest of the experiment.

One Master Timing Generator Card is assigned for each set of timing signals. A Master Timing Generator Card is synchronized with the accelerator clock. But it also can be programmed to replace it if the detectors need to be tested independently of the accelerator. The timing signals can follow a cyclic pattern reproduced at every beam

crossing or can be controlled by external signals.

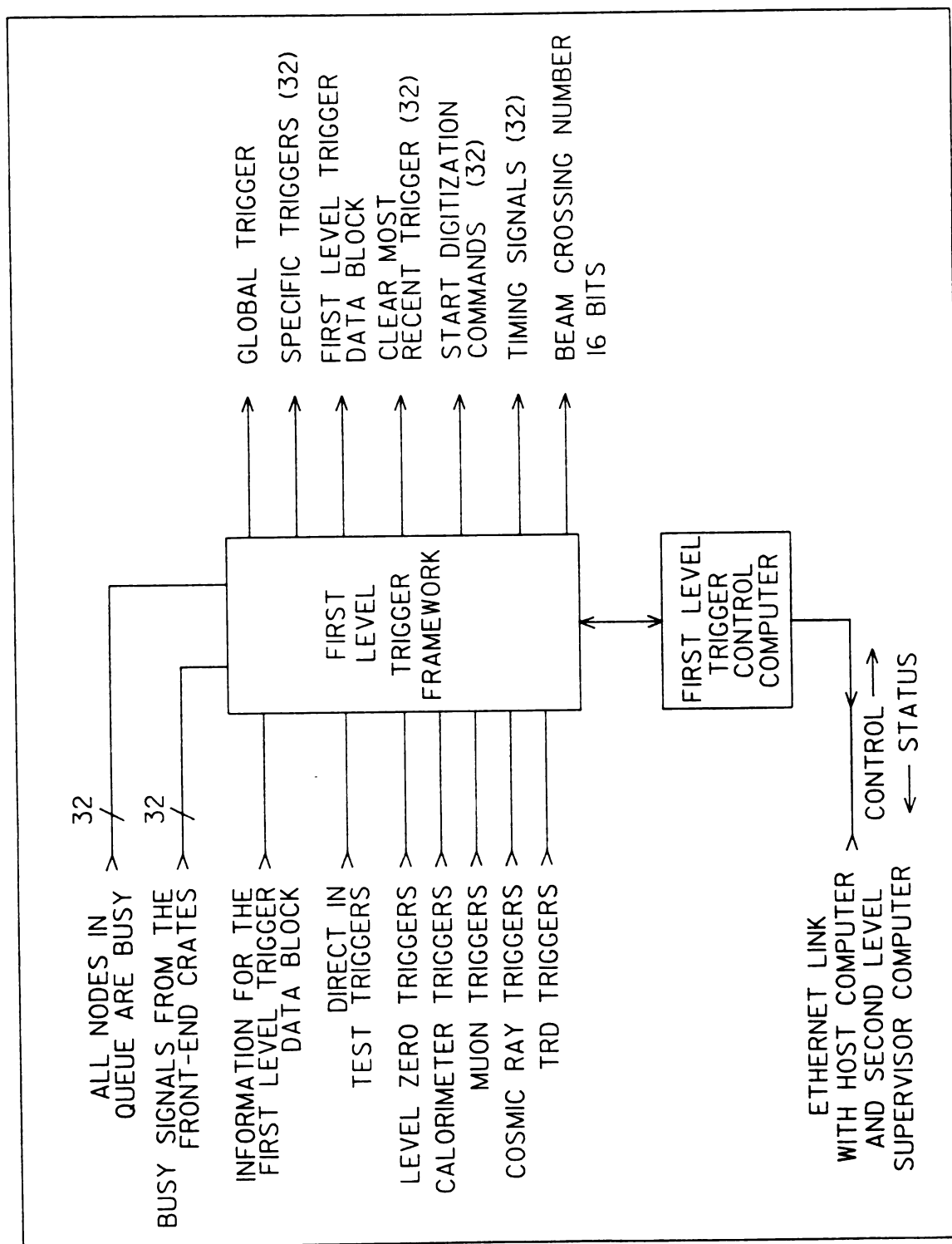
V.3. THE DATA BLOCK

The calculations made by the detector triggers are not lost, but rather included in the set of data describing the event and transmitted to the second level trigger.

The Data Block is a fixed format, 8 kilobyte (8,096 times 8 bits) block of data. The Data Block Builder reads a fixed sequence of registers in the Trigger System and copies the contents of these registers to the VME memory module. A new Data Block Builder Cycle is started at every global trigger decision. Building the Data Block takes less than one millisecond.

The set of data transmitted is called the first level trigger data block. It is a block of fixed length and fixed format. It contains all the information submitted as input to the trigger and/or network and each of the 32 specific trigger decisions plus the values of all the counters associated with each trigger. It also contains other information generated by the detector triggers and external variables sent from other sources.

A very important function of the Data Block is to preserve information about the beam crossing which immediately preceded the selected one. In fact, information from the same sources is collected for the previous and for the current beam crossing.



TRIGGER FRAMEWORK INPUTS AND OUTPUTS

Figure 5.1

The data block is used for many different purposes. The information about the input status to the and/or network along with the trigger decision allows on-line testing of the correct operation of the network.

The monitoring of the different scalers in the trigger framework gives direct information about the triggering rate, dead time, etc. The current and previous buffering of the beam crossings makes possible the evaluation of the hysteresis of the detector.

The data block content is systematically used by the software trigger to get a crude description of the event before it receives the complete data from the main data acquisition system.

The current and previous buffering of the trigger system must not be confused with the so-called double buffering of the whole data acquisition system. The first level trigger system is double buffered, but both halves of the double buffers must be capable of holding and updating the information on the previous crossing associated with the information on the current event. In this sense, the trigger system registers must function as a four-level buffer.

The inputs to and outputs from the trigger system are summarized in figure 5.1.

VI. THE FIRST LEVEL TRIGGER CONTROL COMPUTER

The first level control computer, a MicroVAX, used to program, control, monitor, and test the first level trigger system. A parallel output interface connects it to the Communication Interface Card which drives the control bus of the trigger system.

VI.1. CONTROLLING

The control computer can perform the control operations requested by the host computer or the supervisor computer.

During the normal operation of the trigger and the data acquisition systems the specific triggers are frequently enabled and disabled according to the availability of the data acquisition crates and the availability of specialized level two nodes to process the different types of events. Some of these requests are directly sent by the second level supervisor over a dedicated cable. But it is still sometimes necessary for the control computer to enable or disable a specific trigger. For example, the automatically disabled triggers will be reenabled by the control computer after a request from the supervisor computer.

VI.2. PROGRAMMING

The trigger system is programmable and the control computer is the interface where the transition from physics requirements to register contents is performed. The control computer can write any writable register of the trigger system. Moreover, for testing and verification, any register that can be written to can also be read.

Programming the trigger system is necessary during the initialization or during any kind of calibration when the specific trigger characteristics have to be modified.

VI.3. MONITORING

The control computer also performs monitoring functions. It can compute statistical variables. By directly observing the status lines coming from the communication interface card of the trigger system, the control computer can evaluate the triggering rate, or the fraction of the time when the double buffer of the trigger system is full.

The control computer can also request and read a copy of the next data block to be built. The bandwidth of the communication between the control computer and the trigger system does not allow the data block to be read while it is being built. A data block spy function associated with the data block builder can be activated by the control computer to copy the next data block while it is being built. When the data block spy is full, the control computer reads the data block through programmed I/O.

The access to the data blocks allows the control computer to calculate and monitor the dead time and the triggering rate of each specific trigger.

VI.4. TESTING

The proper operation of the whole experiment relies on the quality of the first level trigger system. To match the criteria of reliability presented in paragraph II.3.1., it is clear that the trigger system must

be designed to be fully testable. Off-line diagnostics should be able to very quickly list down to the board level which parts of the system are defective and require replacement.

Moreover, on-line testing programs should constantly watch the operation of the trigger framework. When an anomaly is detected by the hardware monitoring program, alarm messages can be sent to the host and the supervisor computer. A new initialization or a new configuration may then be requested or even a shutdown and off-line tests could be performed.

VII. FIRST LEVEL TRIGGER COMMUNICATION STRUCTURE

The communication structure, described here, supports both the trigger framework and the calorimeter trigger. Both devices are being built at Michigan State University.

Over the last fifteen months, I designed , built, and tested the cards supporting the communication links and the control of the first level trigger system and its control computer. All the cards of the hardware trigger have been designed on the Intergraph CAD system, in the electronic shop of the Physics/Astronomy Department (see appendix B).

VII.1. PHYSICAL IMPLEMENTATION

The physical implementation of the trigger system does not follow any existing standard. We will describe here why and in what sense the structure of our system is different from other common specifications.

A few points need to be mentioned. First of all, the electromagnetic noise in and around the detector requires a careful design of the communication links. The long signal lines have to be protected from background electromagnetic noise. The solution chosen for this problem is the use of differential lines terminated by a load matching the characteristic impedance of the line. The noise is picked up equally by the two wires constituting the line. The voltage difference between the two conductors is used to carry the information and this is not altered by the noise.

The design of a piece of equipment with a reaction speed of the order of a microsecond requires high speed electronic technology. This constraint will influence the choice of integrated circuits to use and also influence the logical structure of the signal path. All operations must be performed in parallel to the largest possible extent rather than serially.

The table 1 gives the propagation time in TTL and ECL gates. The logic family to use cannot be the classic TTL family or more generally any family where the transistors building the circuits are driven to their saturation points. In fact from the table 1 it is obvious that the optimal choice would be to build a system in ECL (Emitter Coupled Logic) technology.

The advantages of the ECL family are the fast propagation and the availability of many integrated circuits with differential inputs and outputs. The disadvantages of this family are the high power consumption, the requirement of external pull down resistors and the small number of complex circuits available. The ECL family is a non-saturated type of logic. Thus the transistors building the circuit never reach saturation even when the inputs and outputs are at a stable, defined electrical state. The power consumption is higher for non-saturated logic than for saturated logic; and, therefore, a system using ECL logic needs bigger power supplies and also an higher capacity cooling method.

LOGIC TYPE	NAME	MIN.	TYP.	MAX.	UNIT
TEXAS INSTRUMENTS TTL, Standard Logic	SN7402		12	22	ns
TEXAS INSTRUMENTS TTL L, Standard Low-Power Logic	SN74L02		35	60	ns
TEXAS INSTRUMENTS TTL LS, Low-Power Schottky Logic	SN74LS02		10	15	ns
TEXAS INSTRUMENTS TTL S, Schottky Logic	SN74S02		5		ns
TEXAS INSTRUMENTS TTL ALS, Advanced Low-Power Schottky Logic	SN74ALS02	3		12	ns
TEXAS INSTRUMENTS TTL AS, Advanced Schottky Logic	SN74AS02	1		4.5	ns
TEXAS INSTRUMENTS CMOS HC, Silicon-Gate Complementary MOS Logic	SN74HC02			20	ns
MOTOROLA MECL 10K, Emitter Coupled Logic	MC10102	1		3.3	ns
MOTOROLA MECL 10KH, Emitter Coupled Logic	MC10102	.7		1.7	ns

COMPARISON OF PROPAGATION TIME THROUGH A BASIC
LOGIC NOR GATE FOR DIFFERENT LOGIC FAMILIES.

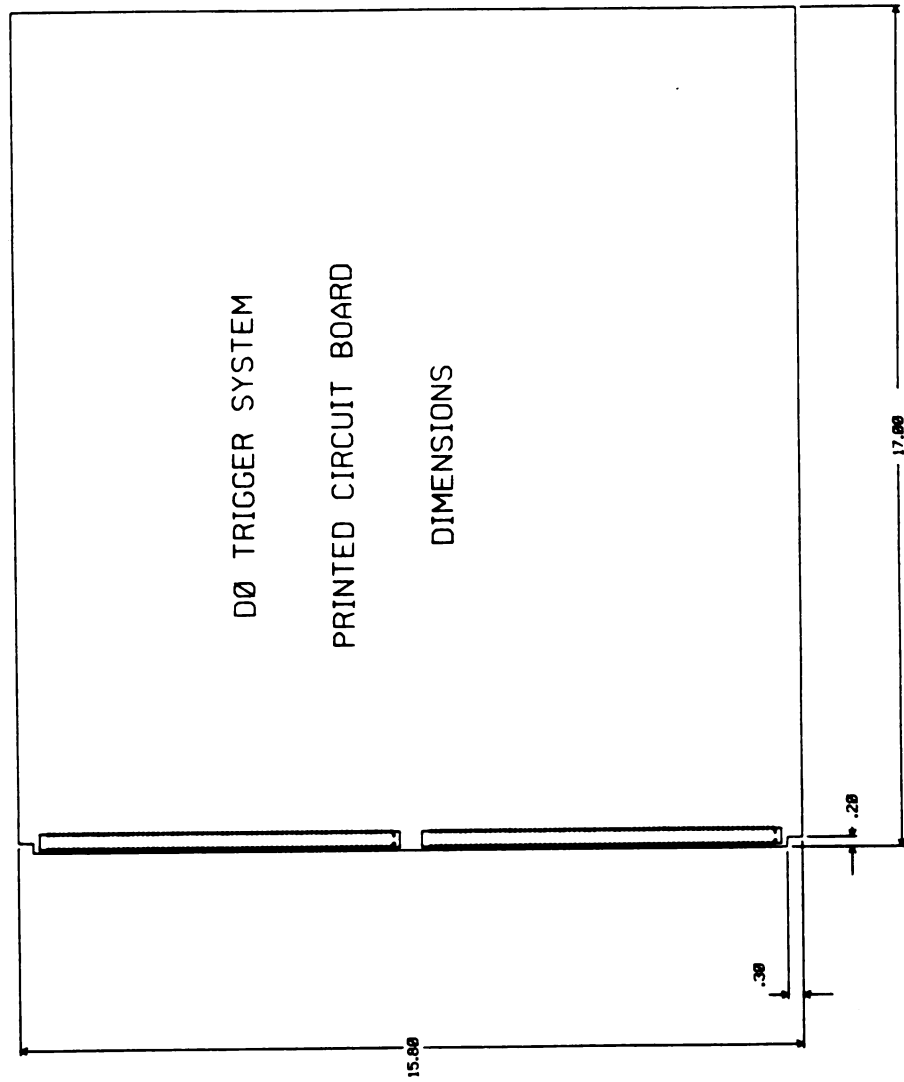
Table 1

It is not efficient to build the trigger with only ECL logic circuits. All the board intercommunication structure is supported by ECL chips, most of the higher complexity logic (like counters, memory, etc.) uses TTL/AS or TTL/ALS families.

The other constraint brought into the design of the trigger system is the number of board interconnections required in the trigger framework and the calorimeter trigger. Two contacts are needed for each differential signal. A card with 100 inputs needs 200 contacts on its connectors not counting the 40 or 60 contacts for data and address signals and the 10 or 20 pins necessary for the power supplies of the board.

There is no existing standard matching these requirements. The FASTBUS standard, designed for high-speed data acquisition systems and multiprocessor communication, would satisfy the need for ECL differential communication and would provide a good physical implementation. However, all board interconnections would have to be carried on external cables linking the cards to each other. The design of the cards would be made harder by following the complex FASTBUS specification, but all the multi-master features supported by FASTBUS would never be used.

The final choice has been made in favor of the design of a custom standard for the communication structure in the trigger system. The cards connect to the backplanes via two dual in-line, board-mounted connectors which provide 280 contacts. Twenty-four contacts are used for the power supplies and 128 differential signals can be carried via the backplane on each board. The cards are 15.8 inches wide and 17.0



CARD DIMENSIONS
Figure 7.1

inches long (see figure 7.1). The cards are placed horizontally in the crates and the crates are placed vertically in the racks in order to allow effective crate lengths larger than the rack width.

The crates have been designed and built at MSU. There are twenty slots per mother-board, two mother-boards per crate, and two crates per rack. The maximum number of cards per rack is thus 80. The cooling is provided by 18 fans on one side of each crate to blow cold air across the boards with water-cooled heat exchangers placed between the racks. The power supplies must provide the +5 volts used for the TTL logic family and the -5.2 volts for the ECL logic family. They are assembled from commercially available power supply modules of high-power density. The power consumption of a single board of the trigger system can exceed 50 watts.

VII.2. COMMUNICATION LINKS

All communication links are carried by ECL signals on differential lines. Twisted-pair flat cables with characteristic impedance of 110 Ohms are used to carry these signals, and special backplanes have been designed and built at MSU to carry differential signals via the backplane board interconnections.

All data transfers made by the control computer or by the data block builder are supported by the Control Computer Bus. The Control Computer bus is the only access to the contents of the registers in the trigger system.

The maximum data size of the different registers spread over the trigger system is limited to 8 bits. The address of a register on a card in the trigger system is the aggregate of the address (8 bits) of the mother-board holding the card, the address (6 bits) of the card in the mother-board and the address (8 bits) of the function within the card. The full address is then 22 bits long. A separate control line specifies whether the operation on the register is a read or a write operation. A strobe line is used during a write operation to specify the point in time when the data must be stored.

Name : Control Computer Bus
 Type : ECl differential signals.
 Number : 32 signals

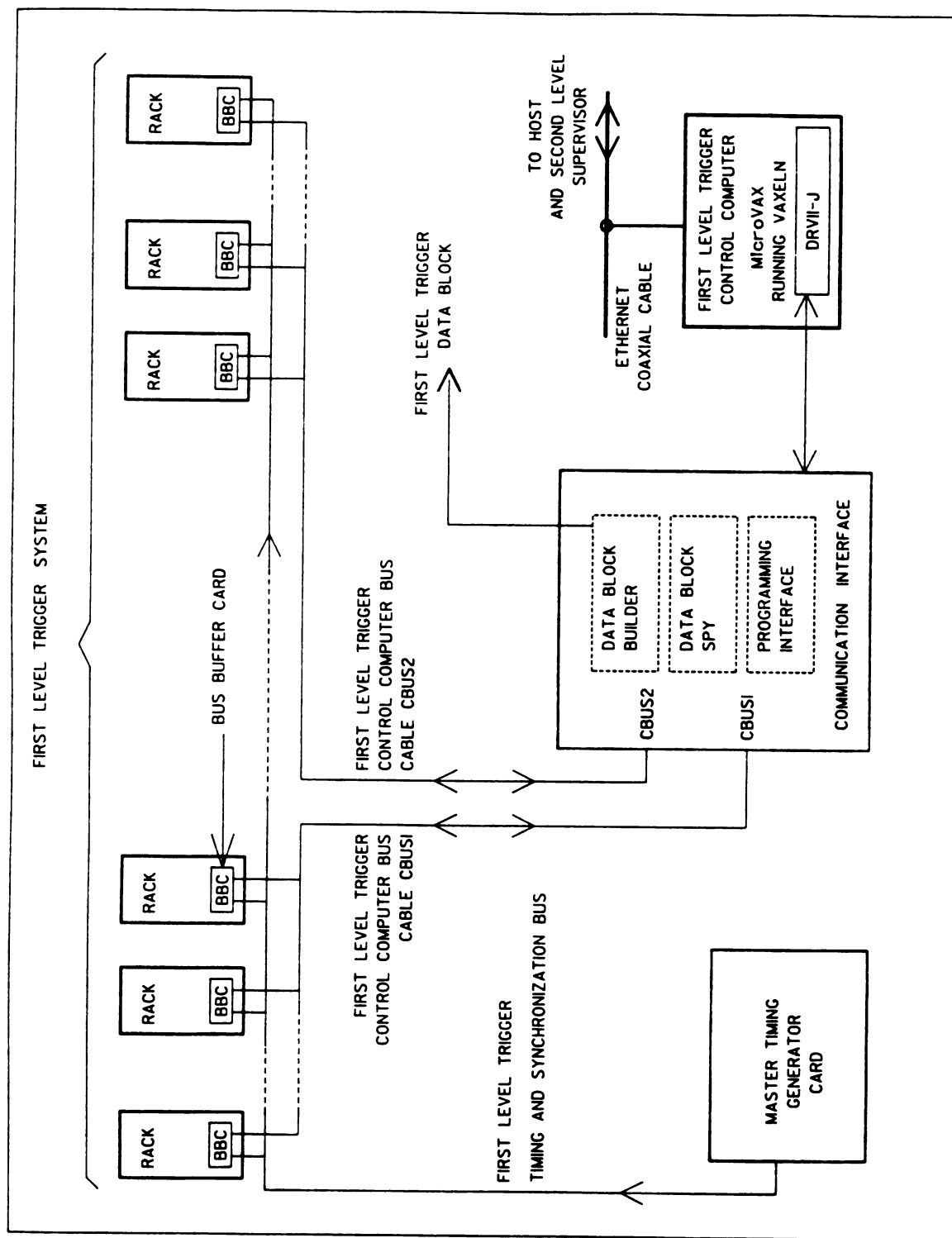
Function :

Mother-Board Address	8 bits	256 different possible values
Card Address	6 bits	64 different possible values
Function Address	8 bits	256 different possible values
Data (Bidirectional)	8 bits	256 different possible values
Direction	1 bit	High for read, low for write
Strobe	1 bit	Active falling edge

TOTAL	32 bits	

CONTROL COMPUTER BUS
 Figure 7.2.

The control computer bus is driven by the data block builder during a data block builder cycle or by the programming interface during a programmed I/O request from the control computer. Both of these functions are located on the Communication Interface Card which is the only driver device on the control computer bus cable. Two types of I/O operations may be performed on this bus: fast readout during a data block builder cycle, and slow programming of the trigger system by the



FIRST LEVEL TRIGGER BUSSES
Figure 7.3

control computer.

To meet the overall timing requirement of the D0 experiment, the data block must be built within 1 millisecond of the trigger decision. In order to build the data block, 8,192 read operations are performed on the control computer bus. Because of the large physical size of the trigger system, the propagation time in the cables is important during a transfer cycle. During a read cycle, the address must travel through the control computer bus cable and the communication cards of the Trigger system, a register must recognize the request as being addressed to it, it must place the requested data on the bus, and the data must travel back through the communication cards and the control computer bus cable. Even with strategically placed ECL logic and differential lines, a read cycle could not be performed in the necessary 100 ns. The choice has been made to implement the control computer on two cables called control computer bus cable CBUS1 and cable CBUS2 (figure 7.3). During a data block builder cycle both cables are used in parallel, and each individual read cycle is performed in 200 nanoseconds.

All timing and synchronization signals are distributed on the Timing and Synchronization Bus. This bus is a set of 32 differential ECL signals generated by a Master Timing Generator Card.

Name :	Timing and Synchronization Bus
Type :	ECL differential signals
Number :	32 signals
Function:	Timing and synchronization of the functions of the calorimeter trigger and the first level trigger framework.

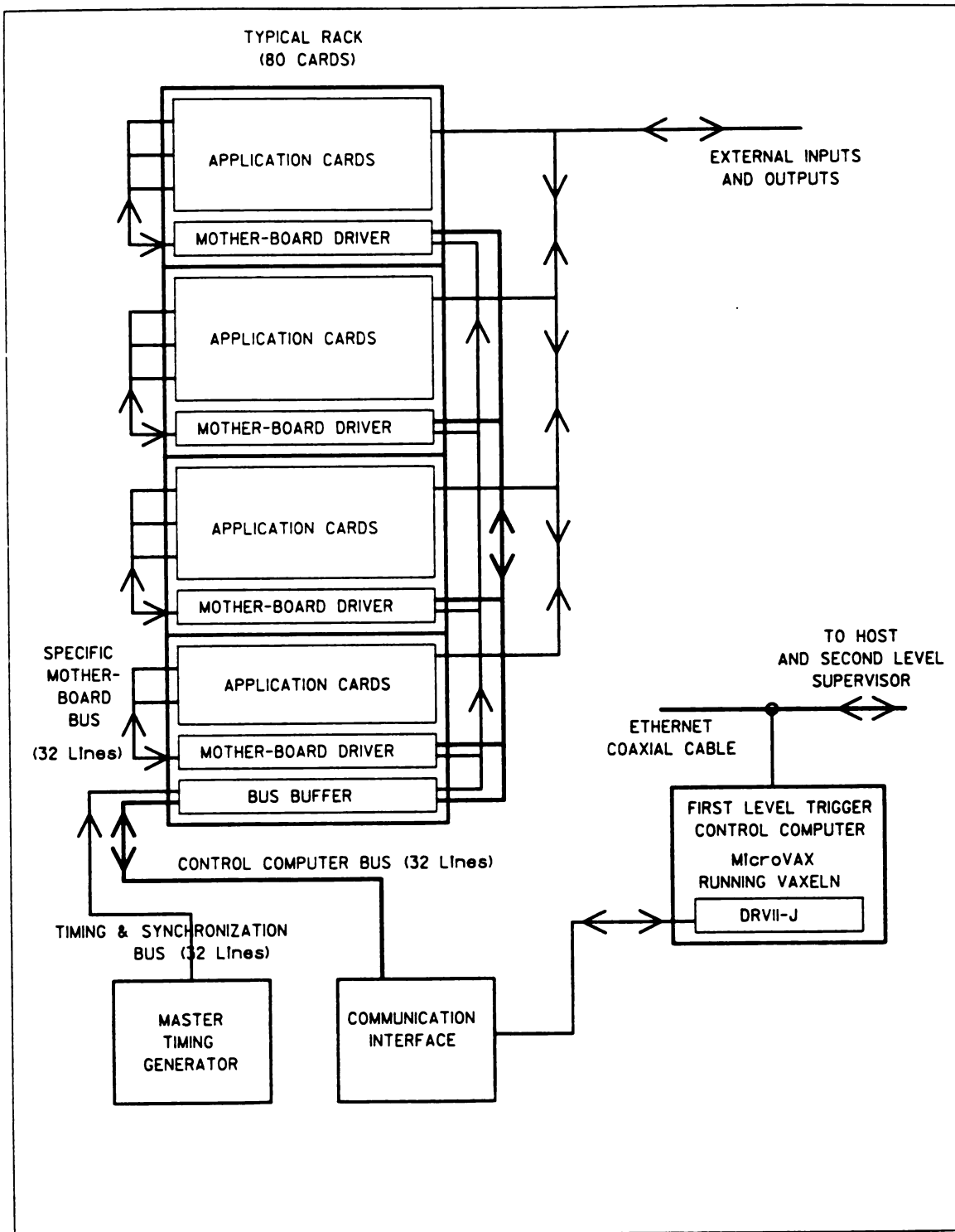
TIMING AND SYNCHRONIZATION BUS
Figure 7.4.

The control computer bus and the timing and synchronization bus are received in every rack by the Bus Buffer Card for that rack. Both busses are forwarded, unmodified by the bus buffer card, to the mother-board driver cards of the rack. There are typically 4 mother-boards and hence 4 Mother-board driver cards per rack.

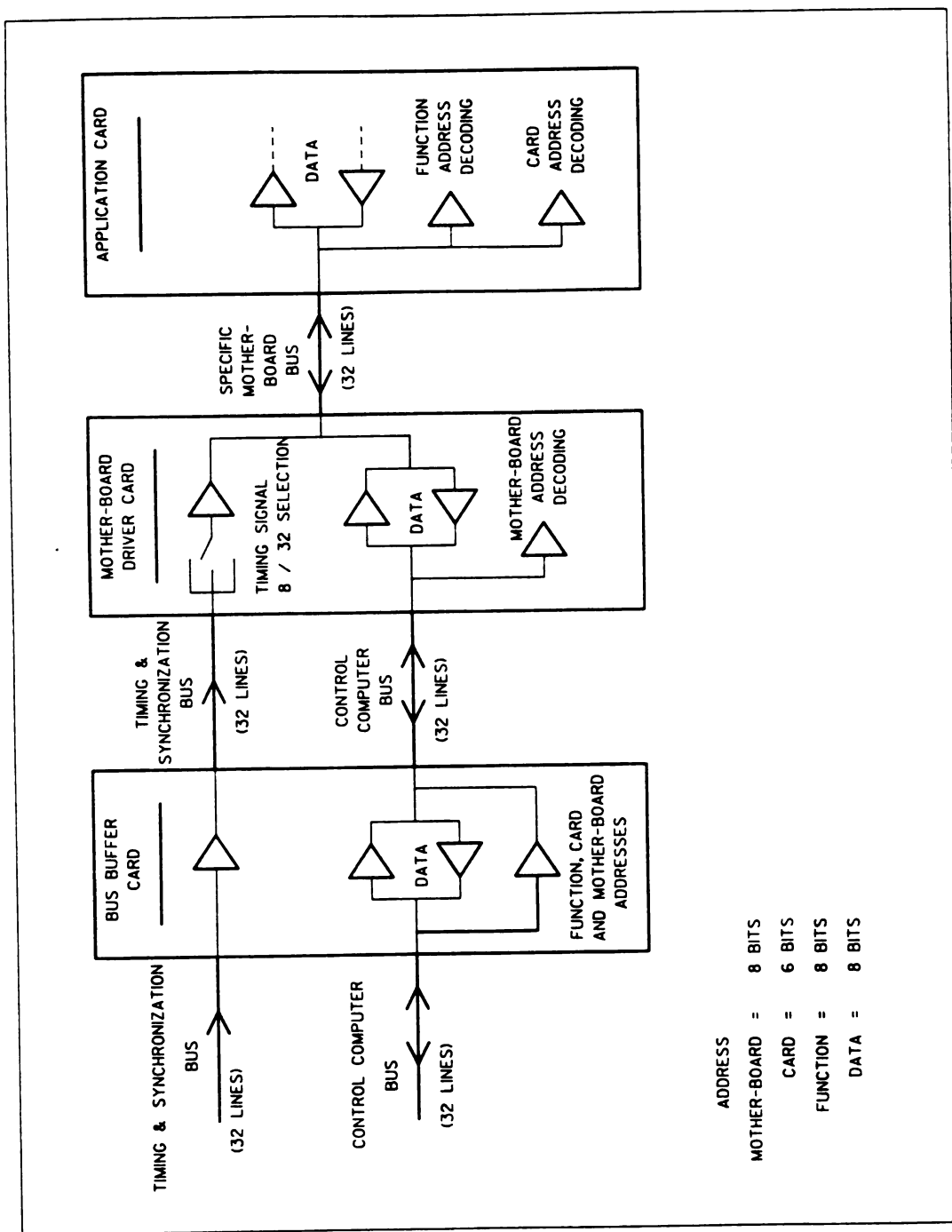
The mother-board address is decoded on the mother-board driver card. The bus transmitted by the mother-board driver card to the application cards is derived from the control computer bus (without the mother-board address) and includes only 8 out of the 32 lines from the timing and synchronization bus.

The bus so created is called the Specific Mother-Board Bus; there are as many different Specific Mother-Board busses as existing mother-board driver cards. A mother-board driver card transfers the control bus to the application cards when the mother-board address present on the control bus matches the address set in a configuration switch on the mother-board driver card. All the specific mother-board bus lines are driven to a default low differential ECL state when the mother-board is not selected, with the exception of the direction line whose default state is high (for read direction).

The path of the busses is presented in figure 7.5 and 7.6.



TYPICAL RACK OF THE FIRST LEVEL TRIGGER SYSTEM
Figure 7.5



ADDRESS DECODING AND DATA PATH
Figure 7.6

Name : Specific Mother-Board Bus
 Type : ECl differential signals
 Number : 32 signals

Function :

Card Address	6 bits	default state card address 0
Function Address	8 bits	default state card address 0
Data (Bidirectional)	8 bits	default state 0
Direction	1 bits	default state hig for read
Strobe	1 bits	default state low inactive
Timing Signals	8 bits	always transmitted
TOTAL	32 bits	

SPECIFIC MOTHER-BOARD BUS
 Figure 7.7

VII.3. COMMUNICATION CARDS

VII.3.1. BUS BUFFER CARD

The First Level Trigger Bus Buffer Card is designed to receive and transmit the Control Computer Bus and the Timing and Synchronization Bus. Typically, one of these cards is included in every rack of the trigger system. Both busses run on two 32 pair twisted flat cables and are distributed to all the Bus Buffer Card of the First Level Trigger System. In every rack, both busses are buffered by the Bus Buffer Card and distributed to every Mother-Board Driver Card of the rack. The outputs of the drivers are pulled down by 56 Ohm resistors connected to VTT (-2.0 V), matching the characteristic 110 Ohm impedance of the differential line. The two cables carrying the two busses inside the rack are terminated by 110 Ohm resistors across every differential line.

Every differential signal is displayed on the edge of the board by a pair of red Light Emitting Diodes (LED's) to provide a visual check of the electrical state of every bit of each bus. The LED's normally work when the timing and synchronizaton signal number 32 is in a high differential state. When this signal is in the low state, all LED's are simultaneously turned on (this feature permits the test of the LED's and diminishes the electromagnetic noise produced by the card).

VII.3.2. MOTHER-BOARD DRIVER CARD

In each rack of the First Level Trigger System the Bus Buffer Card disributes the First Level Trigger Control Computer Bus and the First Level Trigger Timing and Synchronization Bus via two 32 twisted-pair flat cables to every Mother-Board Driver Card in the rack.

On each mother-board, the Mother-Board Driver Card distributes the Specific Mother-Board Bus to every application card on that mother-board via either the backplane or a 64 conductor flat cable running in front of the crate.

All bits of every bus are ECL differential signals. The driver outputs are pulled down by 56 Ohm resistors connected to VTT (-2.0 V), matching the 110 Ohm characteristic impedance of the differential lines. The specific mother-board bus used (on the front cable bus or the backplane bus) are terminated with 110 Ohm resitors across every differential line.

When the backplane is used to carry signals other than those on the specific mother-board bus, a cable can connect the front connector of the mother-board driver card to the specific mother-board bus front connector of all application card in the mother-board in a daisy chain running in front of the crate.

The 32 differential signals composing the Specific Mother-Board Bus distributed to the cards plugged into a given mother-board are:

- The Card Address
- The Function Address
- The Bidirectional Data
- The Strobe bit
- The Direction bit, and
- 8 signals from the First Level Trigger Timing and Synchronization Bus.

The selection of the 8 timing signals to be transmitted on the specific mother-board bus is made on the card. Two sets of wire-wrapped pins permit the selection of the timing and synchronization signals.

The 8 bits of the Mother-Board Address present on the First Level Trigger Control Computer Bus are constantly decoded on the Mother-Board Driver Card. The address of the mother-board is selectable on the mother board driver card with an 8 bit switch.

When the decoded mother-board address matches the switch-selected mother-board address, all signals, including the 8 bidirectional data bits, are transmitted to the application cards.

When the mother-board address, on the First Level Trigger Control Computer Bus does not match the 8-bit address set on the mother-board driver card switch, both the Backplane and the Front Cable Specific Mother-Board Bus stay in the defined default state:

- Card Address 0
- Function Address 0
- Data 0
- Strobe bit in low, inactive state, and
- Direction bit in high, read state.

The 8 Timing and Synchronization Signals selected on a given Mother-Board Driver Card are always transmitted to the mother-board, whether or not the proper mother-board address is present on the First Level Trigger Control Computer Bus.

VII.3.3. COMMUNICATION INTERFACE CARD

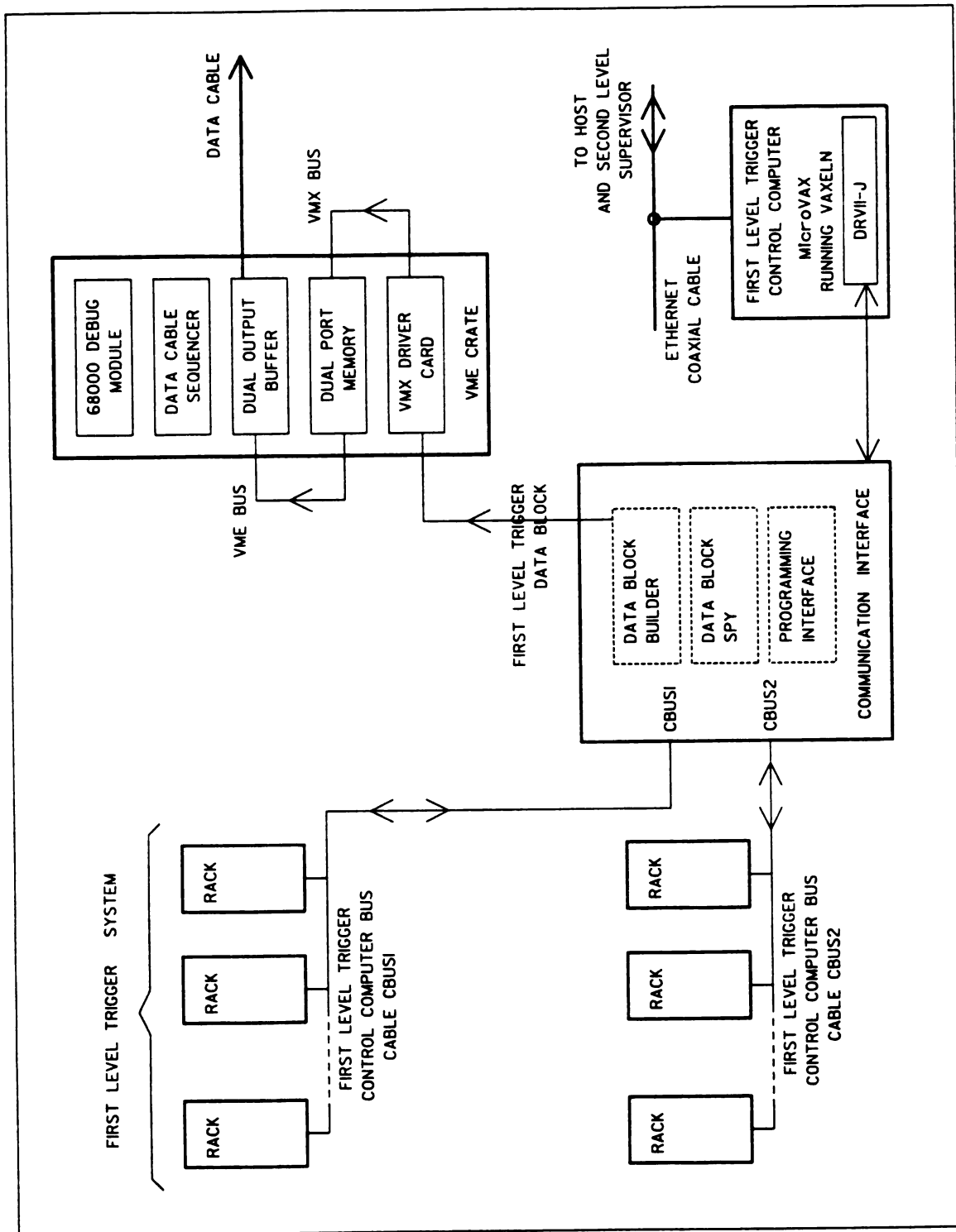
VII.3.3.1. GENERAL

The Communication Interface Card performs three different functions (figure 7.8):

- The PROGRAMMING INTERFACE Function: this function allows the First Level Trigger Control Computer to write and read back any programmable register of the First Level Trigger System.

- The DATA BLOCK BUILDER Function: after a positive Level 1 Trigger decision or after a request from the Control Computer, the Communication Interface Card quickly reads (in less than a millisecond) a fixed set of registers of the First Level Trigger System. The block of data built from the content of these registers is called the First Level Trigger Data Block.

- The DATA BLOCK SPY Function: the Data Block Spy Function allows the Control Computer to access, upon request, a copy of a Data Block. The Data Block Spy records a whole Data Block while it is being built



COMMUNICATION INTERFACE FUNCTIONS
Figure 7.8

without disturbing the Data Block Builder and does not affect the Data Block Cycle time. Later, the copy of the Data Block recorded can slowly be read by the Control Computer.

VII.3.3.2. ENVIRONMENT

The Communication Interface is connected to:

1. The First Level Trigger Control Computer

via a DRV11-J Card.

The First Level Trigger Communication Interface links the First Level Trigger Control Bus to a Digital Equipment DRV11-J Card which is used as a parallel input/output interface to the MicroVAX First Level Trigger Control Computer. The DRV11-J Card provides 64 TTL input/output data lines to communicate with a Q-bus. The First Level Trigger Communication Interface is connected to the DRV11-J card via two 50-wire flat cables.

2. A VME Dual Port Memory Module

via the VMX Driver Card.

The First Level Trigger Communication Interface also connects the First Level Trigger Control Bus to the VMX Driver Card. This VMX Driver Card resides in a VME crate which is part of the Data Acquisition System. It multiplexes data and addresses sent by the Communication Interface in order to match the VMX specifications. It writes the Data block into the Dual Port VME Memory Module via the VMX bus. The Dual Port VME Memory Module will then be read by the Dual VME Output Buffer

via the VME Bus.

3. The First Level Trigger System

via the two Control Buses CBUS1 and CBUS2.

The First Level Trigger Control Computer Bus is the only means of programming or reading the First Level Trigger System. It is split into two 32-pair twisted flat cables. They are called First Level Trigger Control Computer Bus Cable CBUS1 and CBUS2. Each one of these two cables distributes the bus to one half of the Trigger System. In each rack of the Trigger System the Control Computer Bus is received by the Bus Buffer Card of the rack. Each signal is driven on a differential ECL line. The Communication Interface Card provides the pairs of 56 Ohm termination resistors connected to -2V. Both cables of the Control Computer Bus should be terminated by 110 Ohm resistors across each differential pair.

The Communication Interface also receives from the Trigger System:

- One of the 8 Start Digitizing Cables , and
- The Timing and Synchronization Bus distributed by the Master Timing Generator Card.

VII.3.3.3. CONTROL BUS ARBITRATION

In general, the Control Computer Bus is left available to the Data Block Builder in order to immediately service a request for a new Data Block, but anytime the Control Computer needs to use the bus, it may generate a Bus Control Request to the Communication Interface Card. When such a request is received, the Data Block Builder will disable itself from starting a new Data Block, but will complete a current Data

Block Cycle in progress. The Control Computer has no way of stopping the completion of a Data Block Cycle.

During a programming sequence, the control computer can select either the Control Bus Cable CBUS1 or Cable CBUS2. The Control Computer selects one function at a time and uses only the cable serving this function.

During a Data Block Cycle, the two cables are utilized in parallel. The Data Block Builder synchronously drives the two cables and simultaneously sends two different addresses on the cables. Two different registers from the Trigger System are addressed at the same time, and two different bytes of information are received at the same time.

Whenever any one of the two cables is unused, every line on the cable except the direction line is maintained in a low differential ECL state. This default state selects:

- The Mother-Board Address 0
- The Card Address 0
- The Function Address 0
- The Read Direction (high), and
- A low (inactive) Strobe Signal.

VII.3.3.4. THE FIRST LEVEL TRIGGER DATA BLOCK BUILDER

The Data Block is a fixed format, 8-kilobyte block of data. The Data Block Builder reads a fixed sequence of registers in the Trigger System and copies the content of these registers to the VME memory module. The sequence of registers to be read is recorded in 3 PROMs for each bus.

A new Data Block Cycle is started when a Start Digitization Command is generated by the First Level Trigger System on the Trigger-Acquisition Synchronization Cable allocated to the Data Block Builder. When the Data Block Builder finishes its cycle, it sends a Data Block Completed Signal to the Dual VME Output Buffer Card.

When a new Data Block Cycle is initiated, two different addresses are generated on the board from two sets of PROMs and sent in parallel on both First Level Trigger Control Buses CBUS1 and CBUS2. The Communication Interface assumes that the 8 bits of data corresponding to the contents of the addressed registers are received within 200 nanoseconds. Every 16-bit word obtained by concatenating the two bytes of data is latched and sent to the Dual Port VME Memory module while two new addresses are sent on the control bus cables.

Building the 4k x 16-bit word Data Block takes approximately 800 microseconds with an internal cycle time of 200 nanoseconds.

When the Data Block Spy Function is activated by the Control Computer, the data are simultaneously written into an on-board RAM. A Spy Next Data Block Command will activate the Data Block Spy Function for only one complete Data Block Cycle. A request occurring during a Data Block Builder Cycle will be stored and the record will automatically be synchronized with the beginning of the next Data Block Cycle. If a Data Block being recorded by the Data Block Spy is aborted before completion, the Data Block Spy will automatically reset and enable itself for the next cycle until it has been able to record a full data block.

The Communication Interface Card sends the Data Block to the VMX Driver Card which writes every word of the Data Block in a dual port VME/VMX memory module (MVME214) using the VMX bus. The Data Block may be aborted in case of a negative level 1.5 decision, but a successful Data Block is recorded all at once in a single Data Block Cycle. It will be read from the Dual Port Memory Card by the Dual Output Buffer Card using the VME bus. The Data Cable Sequencer and the Communication Interface exchange CONTROL signals in order to synchronize the reading and writing of the Dual Port VME Memory.

The Data Block Builder looks to the First Level Trigger exactly as any detector system section. It is served by its own Trigger-Acquisition Synchronization Cable.

The First Level Trigger Data Block Builder does not require a special type of Dual VME Output Buffer. The Data Block Builder looks to the Data Cable 0 Sequencer exactly as any other Detector System (except that only one section is connected to this Data Cable Sequencer).

VII.3.3.5. TESTING FEATURES

For testing purposes, the Control Computer can initiate a new Data Block Cycle. The Start New Data Block Command is sent through the DRV11-J card directly to the Data Block Builder. This command does not require the Control Computer Bus. A Data Block Cycle already in progress will not be affected. When such a request occurs during a Data Block Cycle, the request will be stored in the same way as an external Start Digitizing Command, and the Front-End Busy Signal will be asserted.

In the case of a Data Block Cycle initiated by the Control Computer, it is important to note that the rest of the Acquisition System does not receive any Specific Trigger Decision Signal or any Start Digitizing Command from the Trigger Framework, but that it will receive a Data Block Completed Signal from the Communication Interface Card.

The Control Computer can also stop the communication of any data to the Dual Port VME Memory Module. When the Disable Data Block Sendout Signal is active, the 16 data lines, the 12 address lines, and the accompanying clock signal going to the VME memory module are left in a high impedance electrical state. However, a Data Block Completed Signal will be sent to the Dual VME Output Buffer. The Disable Data Block Sendout Signal is not synchronized with the beginning of a Data Block and must be carefully used.

The Control Computer can, at any time, instantaneously stop the First Level Trigger System from any further triggering by forcing the Front-End Busy Signal issued by the Data Block Builder to an asserted state.

The Control Computer can stop the activity of the Data Block Builder. When the Disable Data Block Builder line is asserted, no Start Digitizing Command coming from the Trigger-Acquisition Synchronization Cable will be serviced or even stored. The Data Block Builder will, however, respond to a Start New Data Block Command coming from the Communication Interface Card.

VII.3.3.6. PROGRAMMING THE FIRST LEVEL TRIGGER SYSTEM

Before a run, the First Level Control Computer has the ability to initialize, program, and test the First Level Trigger System.

The programming operation is slow, and the Cycle Time is chosen according to the "slowest" function to be programmed in the system. The registers are first written by the Control Computer via programmed I/O through the DRV11-J interface. They are then all read back and compared to their expected value.

During a run, the Control Computer occasionally needs to communicate with the Trigger System (e.g. upon request from the Second Level Supervisor Computer to enable an automatically disabled Specific Trigger). To become master of the Control Bus, the Control Computer first sends a Bus Control Request to the Communication Interface Card and then waits for the Control Bus Freed status line to be asserted before starting to use the bus. The Control Computer can also unmask the Control Bus Freed Interrupt and wait for an interrupt signal after sending the Bus Control Request.

The Data Block Builder Cycle can be delayed if the Control Computer uses the Control Bus. A Data Block Cycle takes approximately 800 microseconds. After a Level One Trigger Decision, the corresponding half of the double buffered registers is expected to be cleared within 1 millisecond.

It clearly means that, during a run, the Control Computer should never use the Control Bus longer than 200 microseconds without checking that no new Data Block Cycle is waiting to be serviced. If one (or two) Start Digitization Command(s) is (are) waiting to be serviced by the Data Block Builder, then the New Data Block Cycle Requested status line is in an asserted state.

Indeed, 200 microseconds should be more than sufficient for any maintenance and control sequence needed during a run. A Start Digitization Command received on the Trigger-Acquisition Cable during this period of time will always be stored and serviced as soon as the Data Block Builder recovers the control of the bus. Of course, this time restriction does not apply when the whole Trigger System is stopped for a complete initialization or programming sequence.

VII.3.4. VMX DRIVER CARD

This card has been designed to write the data block in a VME memory card. The data block is built by the Communication Interface Card with a cycle time of 200 nanoseconds. The data and addresses are sent in parallel on a flat cable and are accompanied by a clock signal.

The VME memory card used to store the data block is a MVME214 dual port static memory card from the Motorola Company. The VMX Driver card will perform the necessary operations to match the VMX bus specifications (ref. 8,9) in order to successfully write the data into the memory module through its VMX port. The VMX lines will only be driven when the data block builder function is active.

The main manipulation is to multiplex the data and address on the 32 multiplexed data/address lines of the VMX bus. The other aspect of the operation is to generate and receive the different timing signals in order to operate as a master on the VMX bus.

The VMX base address is set on three switches located on the VMX Driver Card. The data block is 4,096 words of 16 bits long, therefore the data block will be recorded in 8,192 consecutive addresses of the dual port VME memory starting at the base address set on the VMX driver card. The address coming from the Communication Interface Card must be shifted by one bit (or in other terms multiplied by two) in order to obtain even addresses on the VMX bus capable of holding 16-bit words.

The VMX Driver Card takes the mastership of the bus at every Data Block Builder Cycle. All line drivers of the card are in a high impedance state whenever the data block builder is inactive. The effective address sent on the bus is obtained by multiplying by two the address coming from the communication interface card and adding this number to the base address selected on the switches.

For every new word of the data block signaled by a rising edge occurring on the clock line, the VMX Driver Card will send the transfer size, direction, and the address strobe signals notifying all cards on the VMX bus that a transfer request is made.

In response, the VME dual port memory card is expected to recognize the address as being located on its board. It should send to the VMX driver card the acknowledgment signals corresponding to its port size which is a longword port (32 bits transferred in a single cycle) in the

case of the MVME214.

At the reception of the acknowledgment, the VMX driver card replaces the 32 bits of address on the VMX bus by the 16 bits of data and sends the data strobe signal.

The dual port memory card writes the data present on the bus into the two consecutive memory locations needed to record the word of data and sends the acknowledgment signaling the normal completion of the write operation.

The VMX driver card then terminates the cycle by clearing the strobe lines. The two cards and the bus are then ready for the next data block word and the next active edge on the clock line.

In fact, the data size and transfer are not modified throughout the data block cycle and are kept in their proper state while the data block builder is active. The success of the operation relies on the availability of the memory card. The communication interface card must verify this condition and never initiate a data block builder cycle when the VME dual output buffer card is reading the VME memory card using the VME access of the memory.

The other assumption is that the VMX write cycle is always completed within the 200 nanoseconds available for each transfer.

VII.3.5. DIGITAL EQUIPMENT CORPORATION DRV11-J PROGRAMMED I/O CARD

The MicroVAX Control Computer accesses the First Level Trigger Communication Interface by programmed I/O through a DRV11-J card connected to its Q-BUS. The DRV11-J Card is a parallel input/output interface card providing 64 TTL input/output data lines accessible by the CPU. The First Level Trigger Communication Interface is connected to the DRV11-J card via two 50-wire flat cables.

The 64 bidirectional lines are divided into four independent 16-bit ports. There is a control and a status register for each port.

VII.4. SOFTWARE

The Bus Buffer Card, the Mother-Board Driver Card, and some of the application cards were finished before the completion the Communication Interface Card and before I received the MicroVAX control computer; a temporary method of testing the trigger cards was needed.

The temporary solution selected was an IBM PC computer with a parallel I/O interface card. I designed a wire wrapped interface card to drive the control computer bus from the parallel I/O interface of the IBM PC. It was then possible to write BASIC programs on the PC to perform the read and write cycles on the control computer bus. I have written a number of test programs which have been of great help (and are still used) to test the application and communication cards of the system.

The final software running on the First Level Control Computer has not been written at this time because its final specifications are still being developed by the DO software design team. However, I have already written the core of the final software running on the MicroVAX to control the trigger system.

The MicroVAX II control computer has been connected to the trigger system and all the communication structure is in place. Another MicroVAX II with a VAX/VMS operating system acts as the host computer for the purposes of building and down-line loading the VAXELN systems to the control computer. I have finished a skeleton of the final software in order to verify and test the proper operation of the communication cards and of the application cards already built. A large amount of code has already been written; the lowest level of subroutines controlling the operations on the DRV11-J card are written in EPASCAL, but the main part of the programs is written in FORTRAN.

VIII. CONCLUSION

The construction at Michigan State University of the first level trigger system of the D0 experiment is not yet finished.

The trigger framework is very close to being complete. The communication and synchronization structure has already been built and the large logic network generating the trigger decisions and start-acquisition command is close to completion.

On the other hand, the construction of the Calorimeter Trigger, has not yet been started. The fact that the Calorimeter Trigger will have the same physical implementation, the same communication structure and will be serviced and controlled by the same Control Computer and the same software as the trigger framework will help the design and construction of this detector trigger at MSU in 1987.

We can already predict from the actual state of progress of the Trigger at MSU the existence of a running trigger framework for the coming 2000-channel test at FERMILAB in January 1987.

APPENDICES

APPENDIX A

THE D0 COLLABORATION

The D0 collaboration is spread over the United States and also includes a group of European researchers. The persons from Michigan State University part of the D0 experiment are :

Maris Abolins	(Senior person in charge of the first level trigger)
Ray Brock	(In charge of the cosmic ray scintillator veto)
Dan Edmunds	(Designing the first level trigger)
Jim Linnemann	(In charge of the calorimetry software for the second level trigger and for the off line analysis)
Dan Owen	(In charge of the end cap calorimetry)
Bernard Pope	(Senior person of the MSU calorimeter group)
Harry Weerts	(In charge of the construction of the calorimeter circuit boards)

The following list covers the names of the different national laboratories and universities involved in the design and construction of the D0 experiment. There are about 200 persons participating to this experiment.

- BROOKHAVEN NATIONAL LABORATORY, NY
- BROWN UNIVERSITY, RI
- COLUMBIA UNIVERSITY, NY
- FERMI NATIONAL LABORATORY, IL
- FLORIDA STATE UNIVERSITY, FL

- LAWRENCE BERKELEY LABORATORY, CA
- UNIVERSITY OF MARYLAND, MD
- MICHIGAN STATE UNIVERSITY, MI
- NEW YORK UNIVERSITY, NY
- NORTHWESTERN UNIVERSITY, IL
- UNIVERSITY OF PENNSYLVANIA, PA
- UNIVERSITY OF ROCHESTER, NY
- CEN with a group of researchers from SACLAY, FRANCE
- STATE UNIVERSITY OF NEW YORK, NY
- YALE UNIVERSITY, CT

APPENDIX B

THE INTERGRAPH ELECTRONIC DESIGN SYSTEM

The design of all the electronic cards and other hardware parts built by MSU for the D0 experiment have been executed on the Intergraph Computer Aided Design (CAD) workstation located in the Electronic Shop of the Physics/Astronomy Department.

The Intergraph system consists of a dual screen graphics workstation, a VAX 11/750 (Digital Equipment Corporation computer) modified to receive additional hardware, and dedicated software installed with the VAX/VMS Operating System on the VAX 11/750. The VAX holds the graphics files and operates on the elements in the files. The workstation receives the operator's commands and displays and manipulates on the graphics screen the elements held in the files.

The basic CAD system IGDS (Intergraph Graphics Design System) has been used for the design of mechanical parts such as the cardfiles of the trigger system or the design of the calorimeter detector boards. An additional and specialized software package has been purchased for the design of electronic cards. This specialized package IEDS (Intergraph Electronic Design System) makes possible the design of the high-quality electronics needed for D0.

The following description illustrates the use of IEDS in designing electronic circuit cards. The first step of the design will be to define and describe the function, requirements, and performance of the

card to be designed. The description must be as accurate and complete as possible. This work must take place before and must not interfere with the sketch of the electronics involved.

The second step will transform the functional description into its equivalent electronic schematic. Some other products can be purchased from Intergraph or other companies to simulate the operation and performance of the card from the electronic schematic. MSU did not purchase this additional package.

The final schematic will be drawn on the workstation. The Intergraph software includes a number of libraries describing all the electronic components needed. The library, most commonly used at this point in the design, holds the schematic representations of the components.

The schematic must be carefully completed and all the electrical connections must appear on the drawing. The IEDS software builds a list of the electrical nets recorded in the schematic.

The physical design of the card begins with an outline of the board. The different connectors and elements are placed in a separate file holding the board layout. A library contains the cross references between the elements placed in the schematic and the elements placed in the board file. Each element now has the proper physical dimensions and connections of the corresponding electronic part. All elements placed in the drawing of the board are recognized by the software and compared to the ones placed in the schematic. The IEDS software provides an automatic placement program to help the designer in his work.

The software reproduces for the designer the connections recorded in the schematic file; it places in the board file a straight cosmetic line for every connection in the schematic file. The designer then manually transforms each one of the straight lines into a real trace drawn on the different layers of the board. The IEDS software is also able to automatically place the traces on the board, but the quality of the work depends upon the complexity of the board, the number of layers of conductors, and the placement of the components.

The IEDS software checks that the electrical connections placed in the board file are identical to the connections placed in the schematic file. The software performs the other very useful task of checking the board for short circuits and traces that are too close to each other.

The final result of this design is a graphic representation of the electronic card to be built. A magnetic tape is then produced from the graphics file and sent to a photoplotting company to produce at a one-to-one scale the pictures of the board connections on photographic film. There will be one photographic mask per layer of traces, one for the holes to drill and copper-plate, and additional masks for silk screens or solder masks.

The mylars are sent to another company which will use them to drill and plate the holes in the circuit board and etch the traces on the copper layers of the final board.

Before the era of Computer Aided Design, the schematics were produced on drafting tables and the boards were laid out with black tape on mylar film at a magnified scale before being photographically reduced. The main advantages of such a CAD system over the older drafting technique are the high quality and high density achievable with careful work, and the assurance that the board produced will match the schematic. Any modification of the layout during the board design is eased by global element manipulations instead of laborious displacement of black tape.

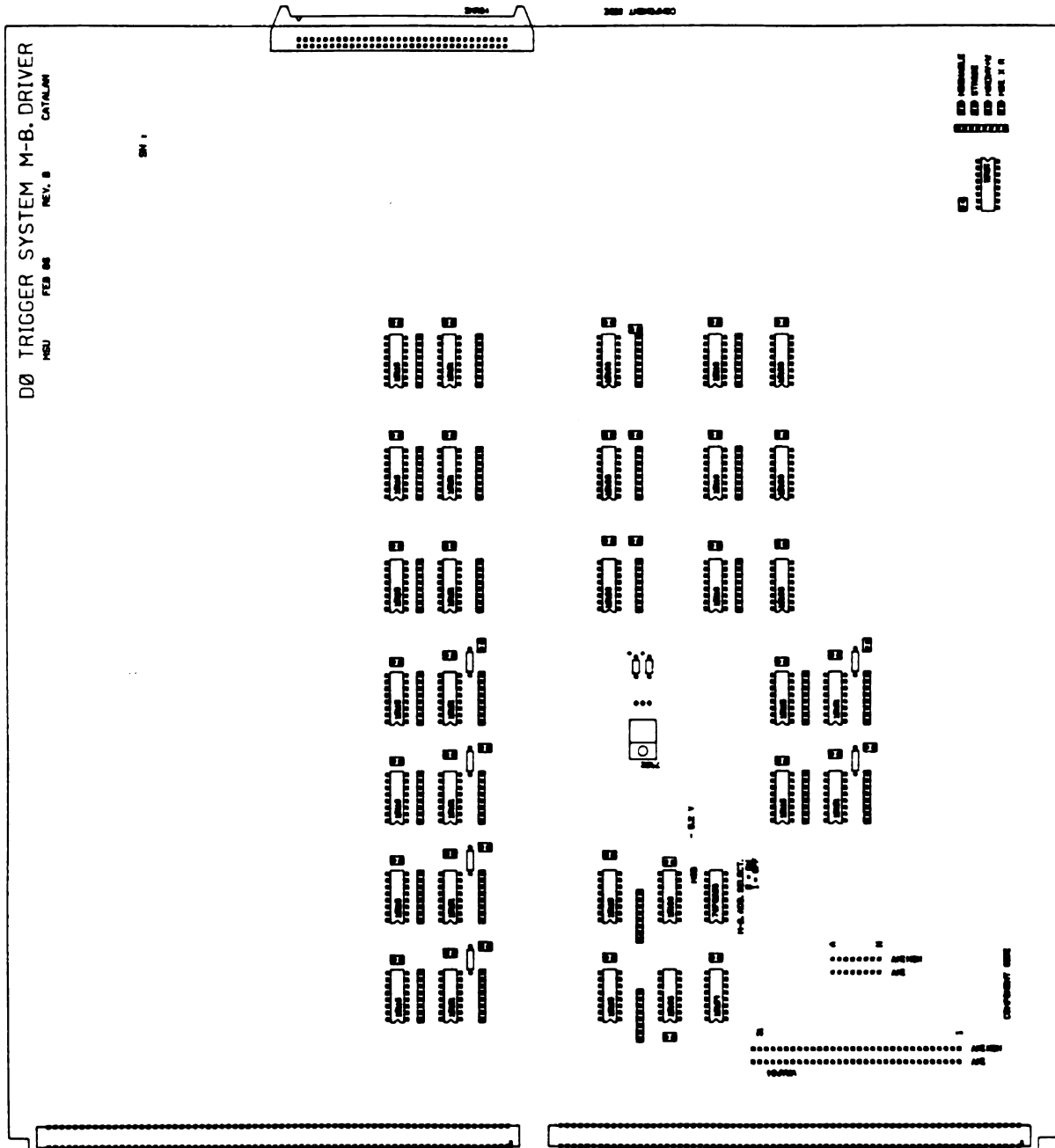
The automatic component and trace placement must be used with care in the case of the high-quality hardware required for the D0 experiment. Indeed, the software does not have the required efficiency to handle high density layout on double-side circuit boards. The use of ECL and TTL logic requires the presence of a double net of power connections in the middle of the signal traces; moreover, a very carefully laid out ground structure is a prerequisite for obtaining the high performance expected from ECL circuitry.

BUS BUFFER CARD



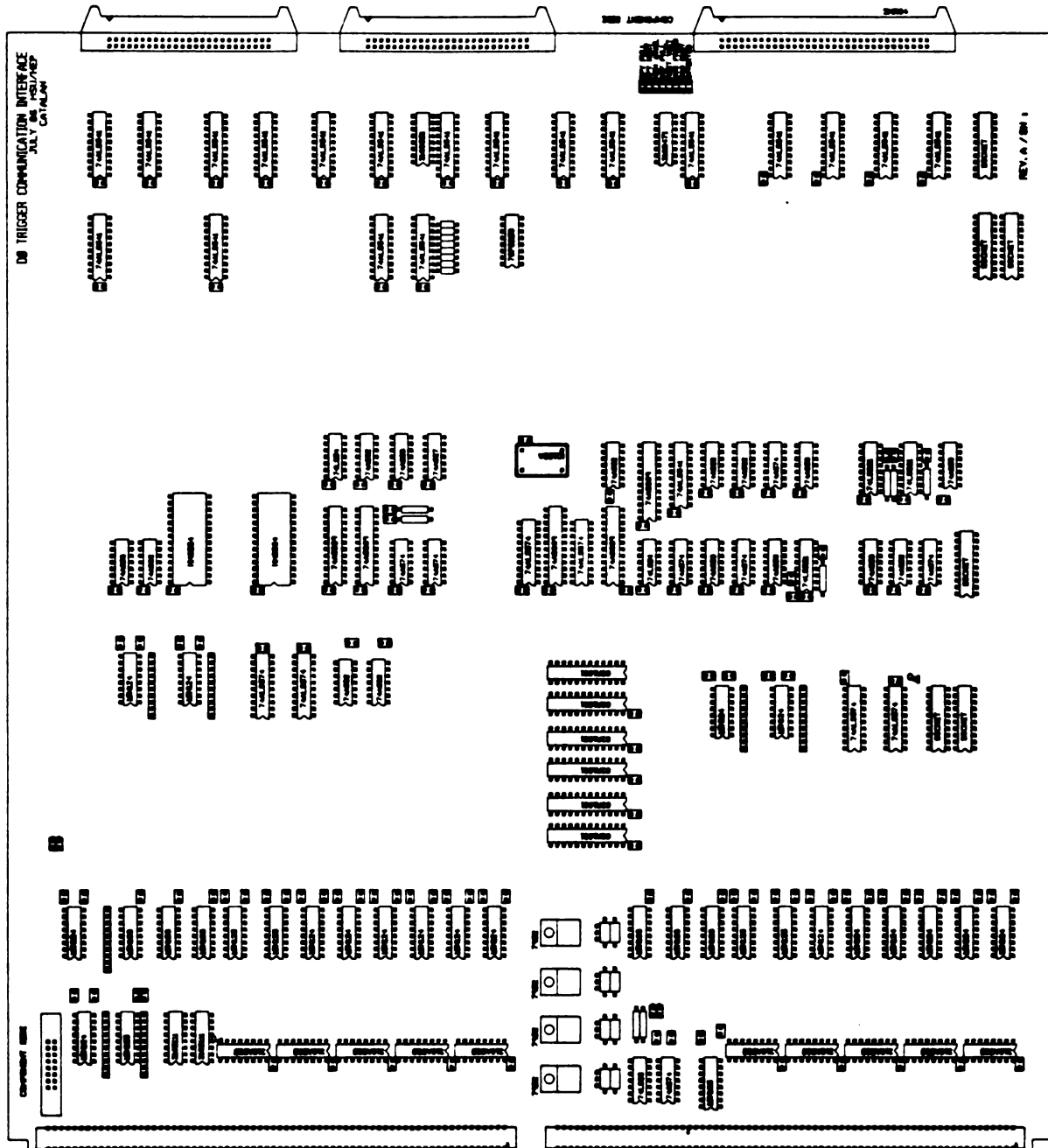
BUS BUFFER CARD LAYOUT
Figure C.1

APPENDIX D MOTHER-BOARD DRIVER CARD

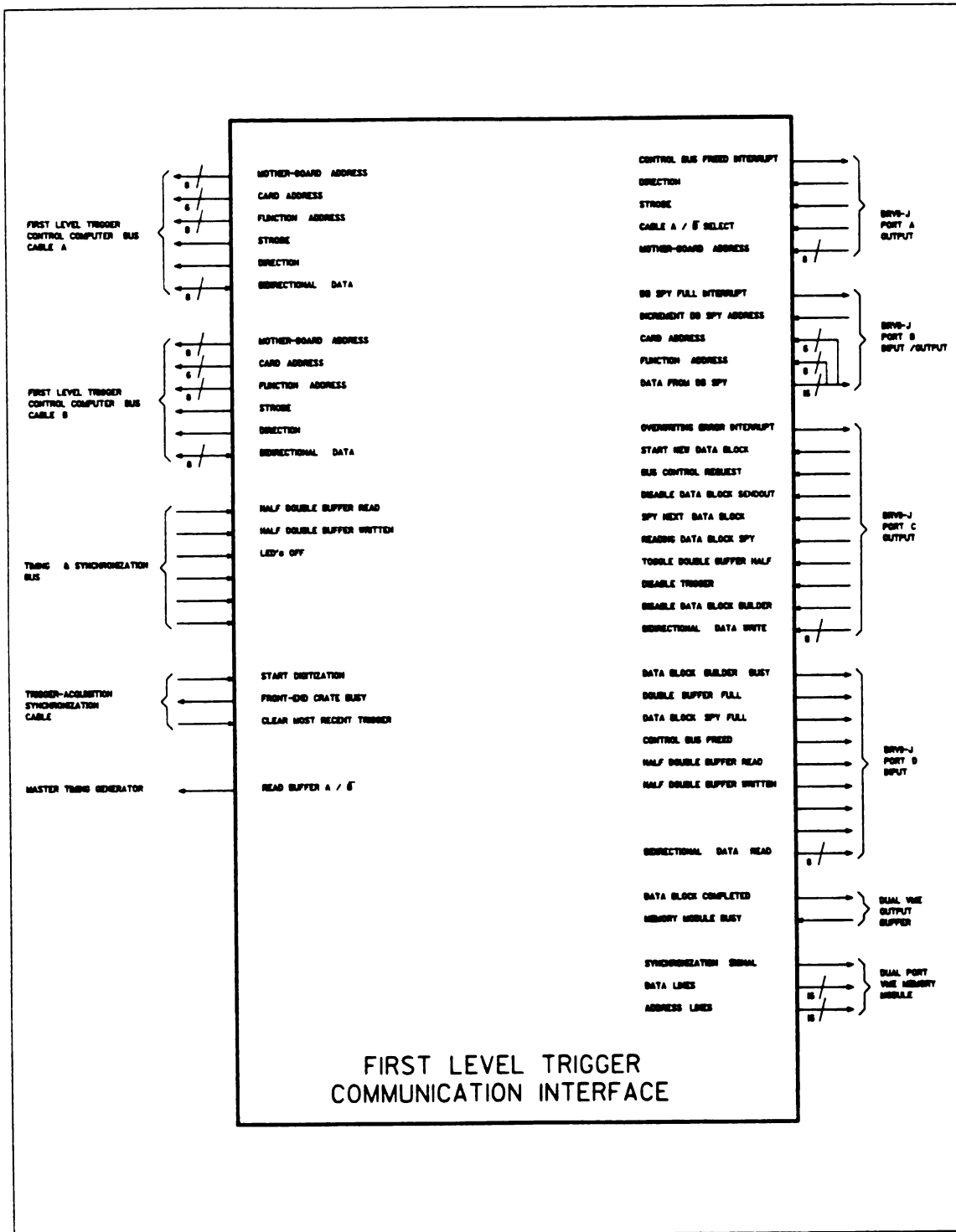


MOTHER-BOARD DRIVER CARD LAYOUT
Figure D.1

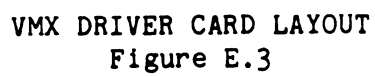
APPENDIX E COMMUNICATION INTERFACE CARD AND VMX DRIVER CARD



COMMUNICATION INTERFACE CARD LAYOUT
Figure E.1



COMMUNICATION INTERFACE CARD INPUTS AND OUTPUTS
Figure E.2



APPENDIX F

INTRODUCTION TO VAXELN

DEFINITION

VAXELN is a software product sold by Digital Equipment Corporation for the installation with a VAX/VMS or MicroVMS operating system. It is not an operating system by itself, but a tool for the development of dedicated, real-time operating systems for VAX processors.

VMS

VAX/VMS is a DEC software product purchased by most of the VAX owners to run as the operating system on their machine. An operating system is the software necessary to transform a single processor and its hardware peripherals into a multitask, multiuser machine offering in a secure and shared environment a complete set of software tools for file processing, program development and program execution.

MicroVMS is the abbreviated version of the VAX/VMS operating system specially developed to run on MicroVAX microcomputers. It basically performs the same functions than VAX/VMS but occupies a smaller space on a smaller machine.

The owner of a VAX machine and VMS operating system can buy and install hardware extensions for his machine, for example, new mass storage devices (tape drives, disk drives), memory expansion cards, Ethernet controller card (for networking purpose), etc.

A VAX owner can also buy new software products to install on his machine, for example, a high level language compiler (FORTRAN, PASCAL..), or the DECnet software to network his machine, or any kind of application programs to handle accounting, graphics, etc.

NETWORKING

DEC offers a number of hardware communication devices to permit different VAX systems to communicate with each other. The main networking software product sold by DEC is called DECnet.

A user on a VAX computer can use DECnet to access files stored on another system or send messages to other users on the network. DECnet allows a direct connection to another node on the network, so the user can work on the remote system as he would do on its own. However, exactly like on his own system, the proper authorizations and privileges are required to access other systems or files from remote machines.

The different systems networked are called nodes and have addresses on the net. The machines periodically exchange messages to indicate their availability on the net.

The communication link used to support DECnet can be as simple as a phone line connecting I/O cards in the two machines. Another communication link commonly used is called ETHERNET. It is designed for high speed communication on a local network and is perfectly fitted to transport the DECnet messages.

VAXELN

VAXELN (or simply ELN) is a software product for use with a VMS operating system. It provides the tools necessary to develop real-time application programs and to build a system including these programs ready to run on another (or the same) VAX processor.

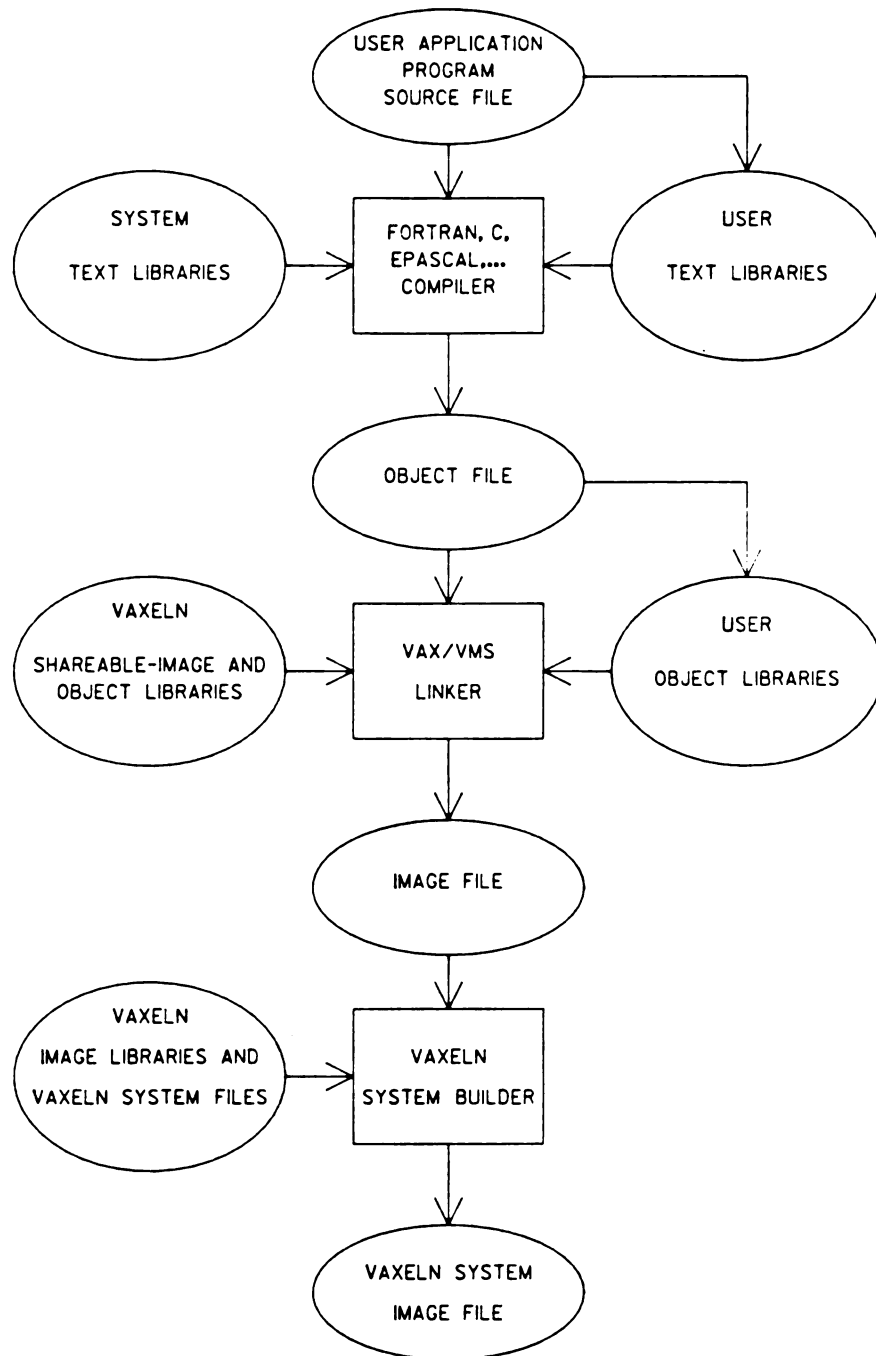
The system image that one builds with ELN holds the application programs and the minimum layer of software to interface it with the VAX processor hardware and to manage the execution of the programs. In general the ELN system overhead and system security are much lower than with VMS.

An ELN system is particularly designed for real-time applications where the response time to external events is critical.

An ELN system is not flexible and will run only the programs included at the time it was built. It is mainly used to solve well understood, dedicated problems.

The preferred high level programming language for creating programs to run on an ELN system is a superset of PASCAL called EPASCAL. It allows the program to perform the connection to and handling of hardware devices connected to the processor, the creation of subprocesses, etc. The VAX C programming language can also perform these tasks.

However, the latest versions of VAXELN also support FORTRAN as a high level programming language. In fact object modules obtained from different compilers (like FORTRAN and EPASCAL) can be linked together to produce the executable system image.



BUILDING A VAXELN SYSTEM WITH VAX/VMS

Figure F.1

A typical ELN application needs two VAX processors in the same network. The node running VMS and holding the VAXELN software kit is called the Host. The node which will run the ELN application is called the Target.

The programs are compiled and linked under the VMS system, the system is built on the host node and then down-line loaded via the network to the target node. The programs can be accessed with terminals directly connected to the target node or remotely from the host computer. The ELN application can also create and access files on the host system as well as on its own peripheral devices. Once booted with an ELN system a target node can run independently from its host.

The down-line loading to the target node via the network can be avoided by building the system on a mass storage media (like a hard disk or a set of floppy disks) and carrying it from one machine to the other. A single processor can also be used under VMS to build the ELN system and then shutdown and rebooted with the ELN system.

The most convenient and probably most efficient configuration uses a host node (MicroVAX, VAX 11/730, 11/750 or 11/780) with VMS and VAXELN, a target node (MicroVAX or set of MicroVAXs) and an ETHERNET link between the nodes.

LIST OF REFERENCES

- 1 DO DESIGN REPORT (November 1984)
The DO Experiment
At The Fermilab Antiproton-Proton Collider
- 2 SCIENTIFIC AMERICAN (February 1974) by R.R. Wilson
The Batavia Accelerator
- 3 SCIENTIFIC AMERICAN (January 1980) by R.R. Wilson
The Next Generation Of Particles Accelerators
- 4 SCIENTIFIC AMERICAN (March 1986) by S. Wojcicki
The Superconducting Supercollider
- 5 DO NOTE 259 : (September 1985) by M.Abolins, D.Edmunds, J.Linnemann
Level One Trigger Studies For DO
- 6 DO NOTE 328 : (February 1986) by M.Abolins, D.Edmunds, J.Linnemann
DO Trigger Framework
- 7 DO NOTE 427 : by M.Abolins, D.Edmunds, J.Linnemann
A Monte Carlo Study of The TRD Trigger
- 8 VMEbus Specification Manual (Motorola) MVMEBS/D2
- 9 MVMX32bus Design Specification (Motorola) MVMX32BS/D1
- 10 DIGITAL EQUIPMENT CORPORATION
VAXELN Installation Manual
VAXELN User's Guide
VAXELN Application Design Guide
VAXELN Pascal Language Reference Manual
Programming in VAX FORTRAN
- 11 MOTOROLA INCORPORATION
MECL System Design Handbook, Fourth Edition, 1983
Motorola MECL Device Data
- 12 TEXAS INSTRUMENT
Standard TTL (Volume2), 1985
The TTL Data Book (Volume3), 1984
- 13 INTRODUCTION TO HIGH ENERGY PHYSICS
Second Edition, D. Perkins, 1982
- 14 ISAJET, Monte Carlo event generation program
written by Frank Paige.