A design study of the trigger and data read-out system for the ATLAS end-cap trigger muon chamber

Motoo FUKUDA
Department of Physics, Tokyo Metropolitan University

January 10, 1997

Abstract

On the recent field of high energy physics experiment, undiscovered elementary particle based on Standard Model is only "Higgs". It is essential for accomplishment of Standard Model to find Higgs particle. In this background, the ATLAS collaboration is now proceeding to search for Higgs and lots of physicists all over the world are studying physics, detectors, electronics etc.

We will observe 14 TeV proton-proton collision reaction in ATLAS using Large Hadron Collider (LHC) in CERN. Since some of the main Higgs search modes include muons, it's very important to identify the muon including event, that is to say, excellent muon detector must be needed. The Thin Gap Chamber (TGC) is one of the excellent muon detectors. It has good performance of shortness of time jitters, then it is adopted to End-cap muon chamber for trigger decision and track data detection in the ATLAS experiment.

In this thesis, the design of trigger and data read-out system for the TGC is proposed. This proposal is satisfied with the physical, electrical, and whole ATLAS data acquisition (DAQ) requirements or specifications. Based on this proposal, production of the prototype slave board of the TGC read-out system has been constructed to estimate the performance of functions of trigger and read-out system. This prototype board has been made upon the VME bus standards and is available to confirm hardware logic and performances of slave board for TGC read-out system. The performance of the board is described in this thesis.

Acknowledgments

I wish to thank many individuals who have helped my work.

I especially would like to thank my supervisor, Professor Chikara Fukunaga, for his valuable guidance and continuous support. I would like to thank Professor Ryosuke Hamatsu for his guidance in the collaboration.

I am indebted to Professor Osamu Sasaki, Mr. Masahiro Ikeno for their guidance, advice, and support. I have never accomplished this work without their help. I would like to express my gratitude to Professor Tokio Ohska and Professor Masaharu Nomachi. And I would like to thank for all ATLAS-Japan collaborator.

Finally, I would like to express my appreciation to my parents for long lasting support.

Contents

Chapter 1

Introduction

1.1 ATLAS Experiment

1.1.1 Overview

The ATLAS experiment is the next large elementary particle experiment project which will be held at CERN[1] in 2005 year. The main purpose of this project is to observe "Higgs" boson. The Higgs is considered to the origin of mass introduced in the spontaneously symmetry-breaking mechanism and is the last elementary particle which should be discovered for the accomplishment of the Standard Model (SM). It will be a great progress to find the Higgs and detailed study of it's characteristics is desired.

The LHC[2] is a proton-proton collision accelerator designed for the ATLAS experiment. It will run at the maximum beam energy of 14 TeV, which is an energy region no physician has ever explored. The LHC will be constructed to the place of the LEP[3] which is the current largest accelerator at CERN. Though the most important goal is to observe Higgs and explore the origin of the mass, a large range of physics opportunities are offered because of the high energy of the LHC. Another important goals are the searches for heavy W- and Z-like objects, for supersymmetric particles, for compositeness of the fundamental fermions and detailed studies of the top quark.

Since the cross-sections of the processes for almost interesting physics questions are small over a large part of the mass range it is important to operate at high luminosity in the ATLAS experiment. The primary goal is to operate at luminosity of 10^{34} [$\mathrm{cm}^{-2}\mathrm{s}^{-1}$]. Consequently maximization of the detectable rates above backgrounds by high-resolution measurements of electron, photons, and muons is required for all detectors. LHC runs at the bunch crossing rate of 40 MHz. In order to observe all the events efficiently, data acquisition (DAQ) system is synchronized to the bunch crossing. So high performances of the detector and electronics systems are required and the ATLAS detector system must be more complicated and bigger structure than the usual high energy experiment system. The whole view of the ATLAS detector is presented in Fig.1.

[1] European Laboratory for Particle Physics

[2] Large Hadron Collider

[3] Large Electron-Positron collider

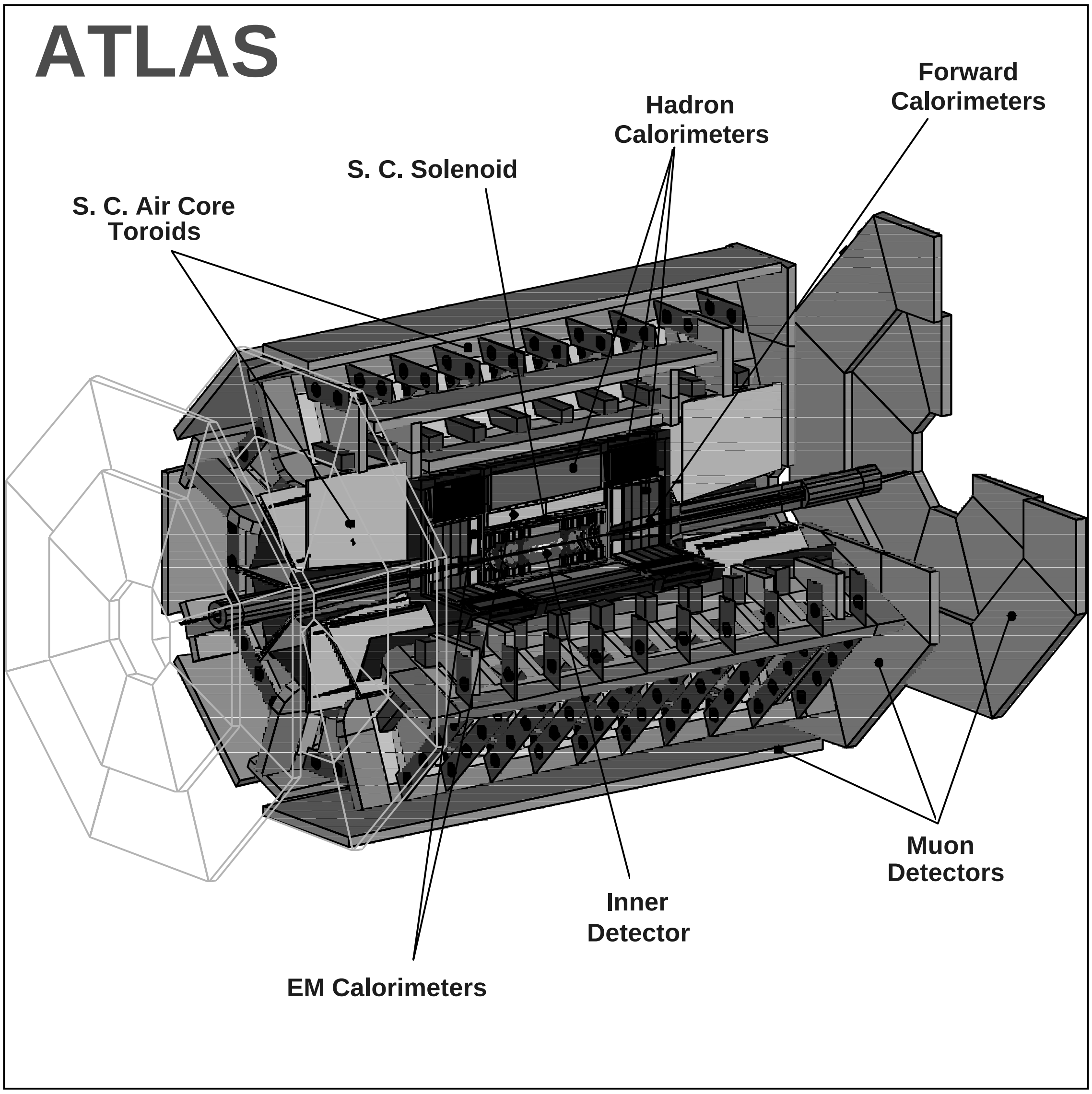


Fig. 1 Whole view of the ATLAS detector.

1.1.2 Muon detection in Higgs search

Owing to the spontaneous symmetry-breaking mechanism, we expect the existence of Standard Model Higgs boson, or of a family of Higgs particles ($H^{\pm}$, h, H and A) when considering the Minimal Supersymmetric extension of the Standard Model (MSSM). According to many studies for the ATLAS experiment, there are several observable processes for Higgs and some of the processes includes muons in a final state like as follows,

$$H \rightarrow ZZ^{*} \rightarrow 4\ell$$

$$H \rightarrow ZZ \rightarrow 4\ell, 2\ell 2\nu$$

$$H \rightarrow WW, ZZ \rightarrow \ell\ell jj, \ell\nu jj$$

where $\ell = e\, or\, \mu, j = jet$. The processes with muon in a final state is one of the most important processes because this will provide relatively clean signature. So the muon detection identification is one of the most important subject in the ATLAS.

1.2 Thin Gap Chamber (TGC)

The Thin Gap Chamber (TGC) [1][2] is the trigger decision as well as tracking detector for muons in the end-cap region of the ATLAS detector. This chamber has very thin structure with the thickness of a few centimeters. The cut-off view and outside shape of the TGC is shown in Fig. 2 and 3. Owing to good time response and lightness of the weight caused to thin structure, this chamber is adopted to the level 1 trigger[4] decision detector at outer region of the end-cap in ATLAS. This chamber is currently being developed and the optimization of the parameters is advanced in a collaboration of Japanese, Israeli institutes and CERN.

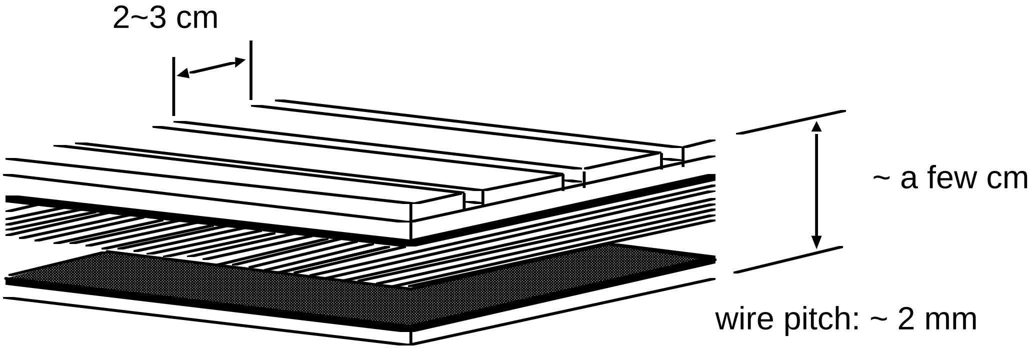


Fig. 2 Cut-off view of Thin Gap Chamber.

[4] First data selection signal. Show next chapter in detail.

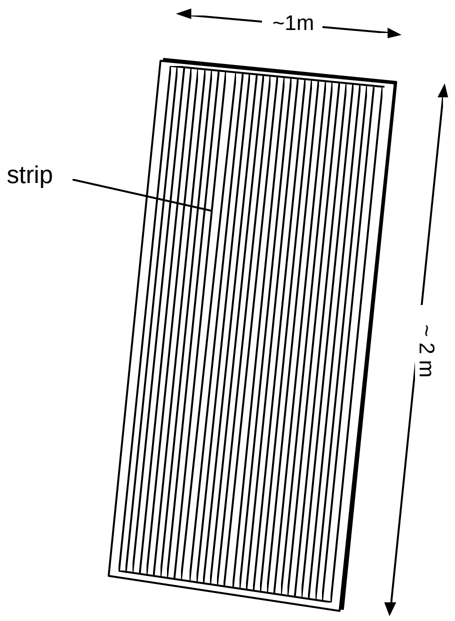


Fig. 3 Outside shape of a piece of TGC.

The rough size of TGC is about 1 x 2 meters and thickness of about three centimeters. This chamber unit has about 200 wires along the transverse direction and about 30 pieces of copper stripes on the one side of the surface along the longitudinal direction. Combined with the both signals from wires and strips, two-dimensional position information is obtained.

As the system shown in Fig.4, one layer has octagonal shape. One octant is covered with this chamber and seven layers are taken in one side of the end-cap. Based on the multiple layer structure, muon track information is obtained.

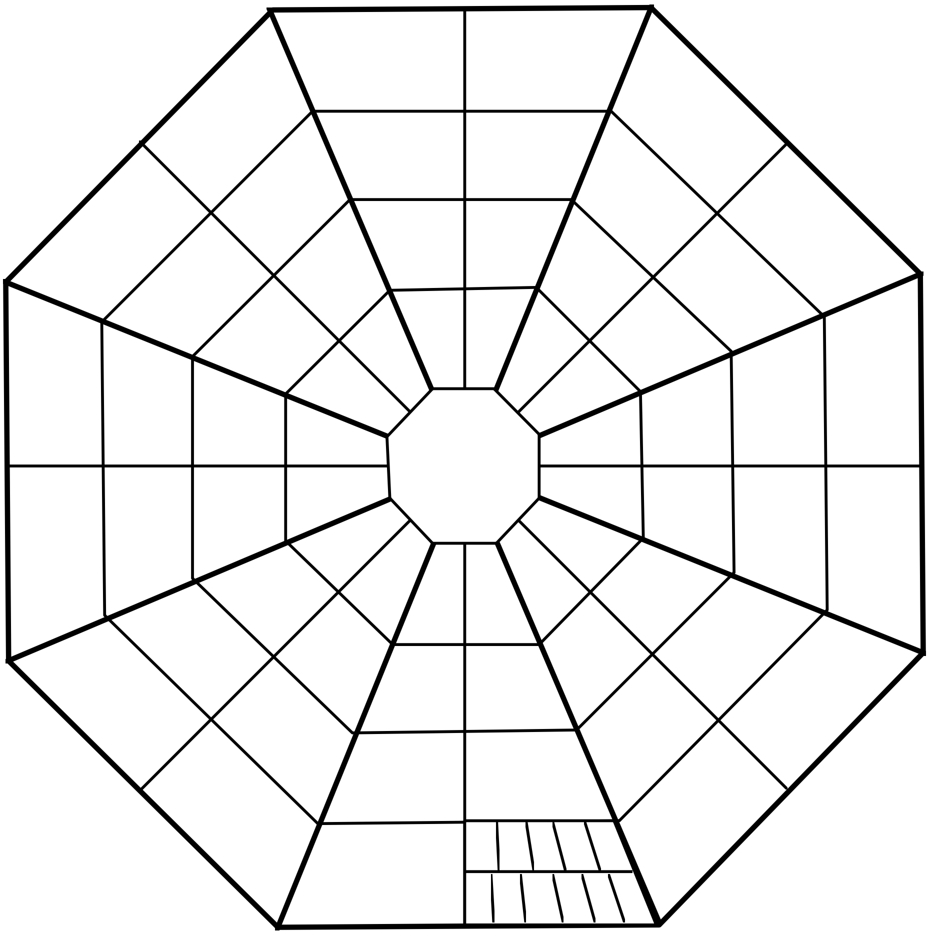


Fig. 4 End-Cap panel structure (One layer)

Chapter 2

Trigger and Data Acquisition (T/DAQ) System for the ATLAS Experiment

2.1 Trigger system

The trigger system works to judge the physically interesting event or background one at the data taking. The concept of the trigger system is presented in Fig.5.

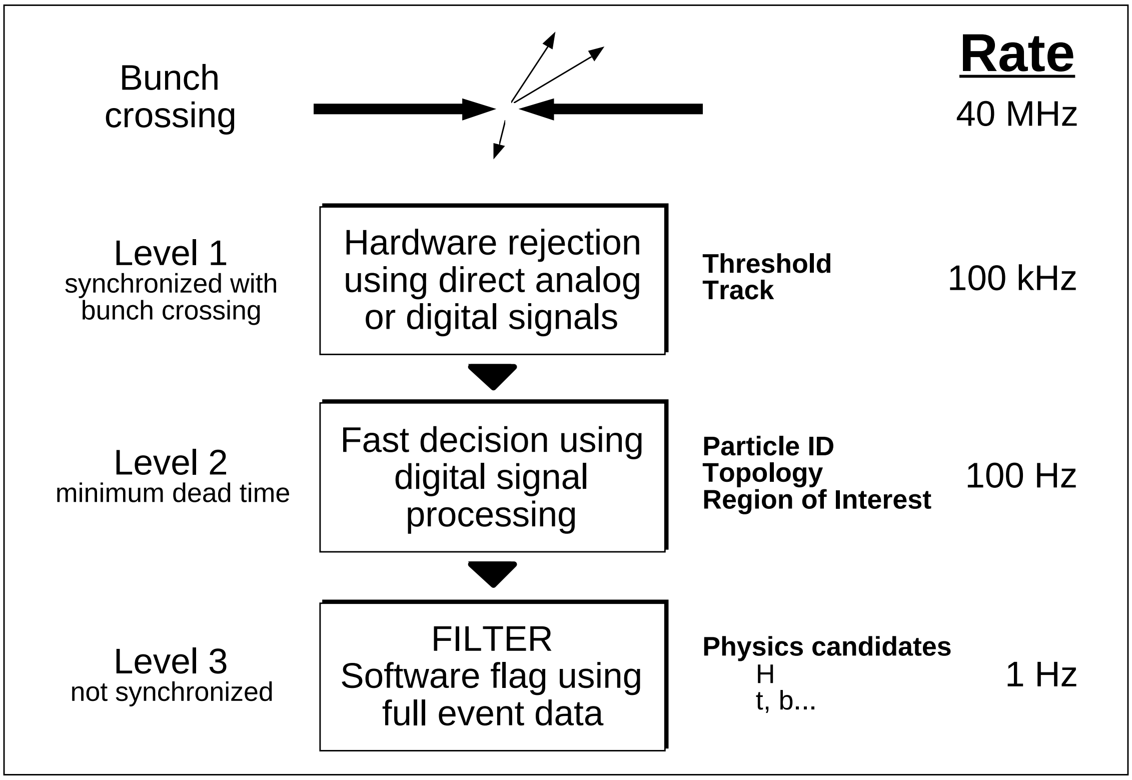


Fig. 5 The Concept of trigger system.

Trigger system is divided into three levels.

Level 1 trigger is decided with the most simplest and fastest way in the three trigger levels and is generated by “logical” hardware architecture. In this level, simple algorithms like threshold cut and/or track information are taken. All sub-detector systems must accept level 1 trigger at the maximum rate of 100 kHz and all the triggered data should be sent to the level 2 trigger region. Untriggered data is regarded as background event and thrown away at this level.

The level 2 trigger is decided by “hardware processor” with rough calculation of data whether physically interested event or not. In particular, the “Region of Interest” (RoI) data like a tracking data for an interested particle found in a level 1 triggered data are selectively calculated. The maximum rate of the level 2 trigger is ~1 kHz and the data accepted to the level 2 trigger is transferred to the level 3 processor.

The level 3 trigger is the final selection in the on-line DAQ system. This trigger is decided by “software processing” of the full event building using all the data. The maximum rate is expected to ~1 Hz. If passed the check of the level 3 trigger, the data is written down to a storage media and analyzed in off-line detailed calculation.

2.2 DAQ system

The DAQ system should be designed to take account with the whole DAQ data flow and trigger acceptance. In each level of the trigger processing, buffering memory is placed to keep the DAQ data for time long enough to decide the trigger, which is called like “level 1 buffer”. It is necessary to pool the DAQ data in order to gain time that trigger processor takes to judge an interesting event or not. For the level 1 buffer, required hold time are ~2.5 μ seconds and ~10 m seconds are required for the level 2 buffer.

In addition the DAQ system should be designed with attention to the facility of the trigger processing, event building and final detailed analysis. Therefore partitioning of the DAQ unit and format or arrangement of the DAQ data are taken care in each level. In the following section, the concept and detailed explanation of the T/DAQ components are described.

2.3 Identification of Components

The ATLAS Technical Proposal [3] has presented the concept of the read-out system. Based upon this concept, the “Trigger & DAQ Interfaces with Front-End Systems: Requirement Document” [4] attempts to define clearly the interface boundaries between the T/DAQ system and the front-end electronics subsystems as shown in Fig. 8. In this document, the T/DAQ system is partitioned into subsystems which have the same logical components and building blocks in any subdetectors.

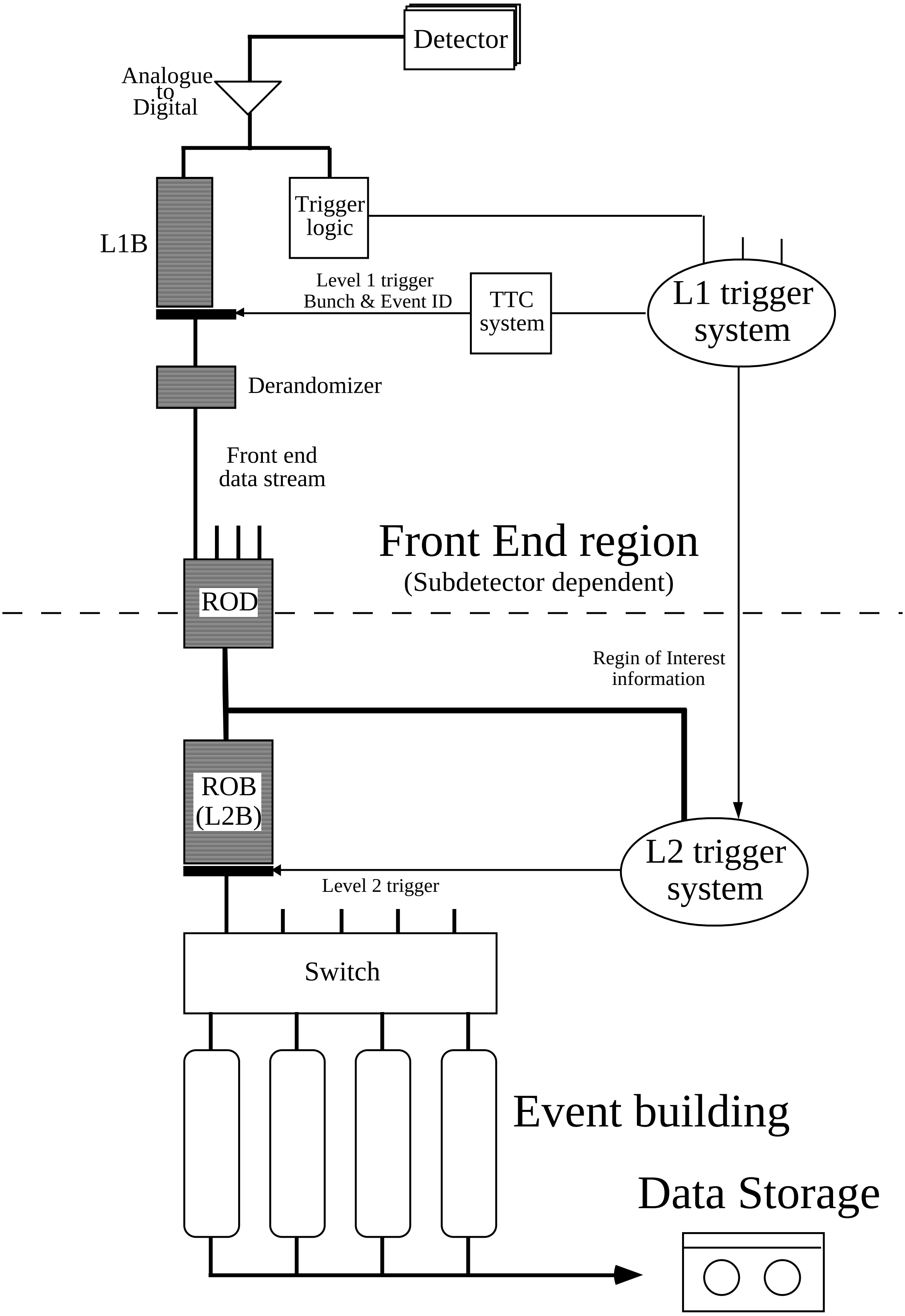


Fig. 6 Functional view of Trigger & DAQ component

Applying the same T/DAQ scheme over all detector groups, we can share the facilities for operation, maintenance and repair.

The explanation for some functional components of the T/DAQ system based on Fig.6 is described in the following subsections.

2.3.1 Front-End Electronics Subsystem

The front-end electronics subsystem includes different functional components:

- the front-end analogue or analogue-digital processing,
- the Level 1 buffer (L1B) in which information is stored and retained for time long enough to accommodate the Level 1 trigger latency,
- the Derandomizer in which the data corresponding to a Level 1 trigger accept are stored before being sent to the following level, and
- dedicated links or buses which are used to transmit the front-end data stream to the next stage.

2.3.2 Read-Out Driver (ROD)

The ROD is the functional element of the front-end system where one can reach a higher level of data concentration and multiplexing by gathering information from several front-end data streams. Elementary digitized signals are formatted as raw data prior to be transferred to the ROB.

2.3.3 Read-Out Buffer (ROB)

The ROB includes the level 2 buffer and receives and stores the raw data from the ROD. It performs the following tasks:

- data storage during level 2 and, for level 2 selected event, at least until read-out to level 3 is completed,
- buffer management,
- error detection and recovery, diagnostics and one-line monitoring, and
- data server for level 2 and level 3.

2.3.4 Level 1 Trigger System

The level 1 trigger function is provided by a 40 MHz synchronous system that receives directly from the front-end electronics dedicated “coarse” data, organized and segmented in a detector component specific way. These data are extracted from the relevant front-end electronics systems and sent to the sub-trigger processors through dedicated “level 1 links”.

- Level 1 trigger processors for the calorimeter and muon systems.
- Level 1 central trigger processor including dead time logic.
- Connection to Trigger, Timing and Control (TTC) system.

2.3.5 Timing, Trigger and Control (TTC) System

The TTC system interfaces with the Level 1 Central Trigger Processor, Detector Control System, Level 2 and/or DAQ systems and the LHC machine. This system provides timing, trigger and control signals to the Front-End Electronics Systems, RODs, Level 1 trigger processors and ROBs. These signals are sent via optical fiber distribution. It delivers all the fast signals which are synchronized to the LHC machine clock, and one which have to be compensated for different detector, electronics and propagation delays. These include the LHC clock, Level 1 Trigger Accept, Bunch ID Reset, Event ID Reset, selective resets and synchronous test commands. The system can also transmit asynchronous controls and data such as trigger type information and individually-addressed channel enable, deskew adjust and calibration parameters. All the features mentioned above are provided simply using TTC receiver chip [5][6].

2.4 Requirements of the T/DAQ System

A part of the requirement, mainly concerned with the front-end electronics, for designing the sub-detector's T/DAQ system is described in this section.

2.4.1 Level 1 Buffer

Detector groups are responsible for providing the Level 1 buffer. The Level 1 buffer has to be large enough to accommodate the maximum possible Level 1 trigger latency of 2.5 μ second presented by the T/DAQ group. If it is implemented as a pipeline architecture, which usually adopted almost in all detector groups, the information from the detectors has to be stored at 40 MHz clock rate until the Level 1 trigger decision is available.

2.4.2 Derandomizer

Detector groups are responsible for providing the Derandomizers. When a Level 1 trigger accept signal occurs, the data stored in the Level 1 buffer have to be transferred to the next level of buffering (ROD). Due to the random time of arrival of the L1A signal and the limited transmission speed, a Derandomizer stage has to be foreseen.

The size of the Derandomizer and the speed of the front-end links have to be chosen in order to limit the deadtime introduced at the level of the Derandomizer to a value less than 1% in the worst case for the sub-detector as a whole.

The occupancy of the Derandomizers has to be estimated in order to limit losses of events when they become full. The traditional way of asserting a BUSY signal to force the Trigger Processor to introduce deadtime should be dismissed in ATLAS.

2.4.3 Bunch and Event ID

It is implemented to identify uniquely the events with two identifiers, the Bunch Identifier and the Event Identifier. The Bunch ID is a 12 bit number which gives the bunch crossing count. The Event ID is a 24 bit number which gives the Level 1 trigger Acceptance number. The detector groups must add the full length Bunch and Event ID to the raw data from the ROD level. However, only in the front-end region, it isn’t necessary to handle the full length Bunch and Event ID.

2.4.4 Read-Out Driver

The ROD receives data from the Derandomizers at the Level 1 trigger rate. Whenever possible, the ROD output should match the Region of Interest segmentation or a subset of it. The ROD has to add full length Bunch and Event ID to raw data. From the different pieces of event coming from the Derandomizers, the ROD must build an event fragment which includes at least Bunch ID, Event ID, error status information, ROD module ID number and raw data. If the buffer in the ROD is nearly full, the ROD has to provide a BUSY signal in order to stop Level 1 trigger generation. A full handshake mechanism will be defined.

Chapter 3

T/DAQ System for the TGC

Any schemes and parameters of the TGC front-end T/DAQ system is now being designed under the condition of the T/DAQ requirement. In this section, the present status and some ideas or proposals for TGC front-end T/DAQ systems is described.

3.1 Condition of the Front-End T/DAQ System

TGC is literally very thin and so the arrangement space of the front-end electronics is also narrow and thin. This condition restricts the design of the front-end electronics not to take the usual way like using the crates or racks. All printed circuit boards must be arranged on the surface of the TGC. Then we have to through away using VME bus for the local DAQ bus although we thought that the VME is one of the most appropriate system.

We have been searching the other data transmit methods in place of VME, but any of these have its merits and demerits, for instance the speed of transmit, the hardness of design, the heaviness of protocol or the expense in costs. If possible, we would like to adopt the major standards in order to avoid the troubles or costs in production or maintenance.

One of the best solution satisfied with the requirement is a serial bus with high transmit speed. The merits of a serial bus are facility of cabling and failure hard topology. In general serial transmit can reduce the cable thickness compared with parallel transmit. Bus topology is hard against failure of any nodes compared with, for instance, Token-Ring topology. The demerit is to have a tendency of heavy protocols. We are trying to adopt a serial bus to the TGC front-end T/DAQ system.

3.2 Circumstances of T/DAQ System for the TGC

The T/DAQ system of the Thin Gap Chamber has also been designed in conformity with the ATLAS T/DAQ requirement. The scheme and parameters of the system is now being developed. As shown in Fig. 7, we took the idea of having a local DAQ block as a unit of front-end DAQ system for the TGC. This block must geographically be put onto the surface of chamber. One block manages total 8 to 24 chambers. One DAQ master board and 8 to 20 DAQ slave boards consist of a local DAQ block. The DAQ master includes ROD[5] function and DAQ bus master logical unit which manages the local DAQ bus. The DAQ slave includes Level 1 buffer and Derandomizer. The triggered data are transmitted from Derandomizers in a slave to ROD in master via local DAQ bus.

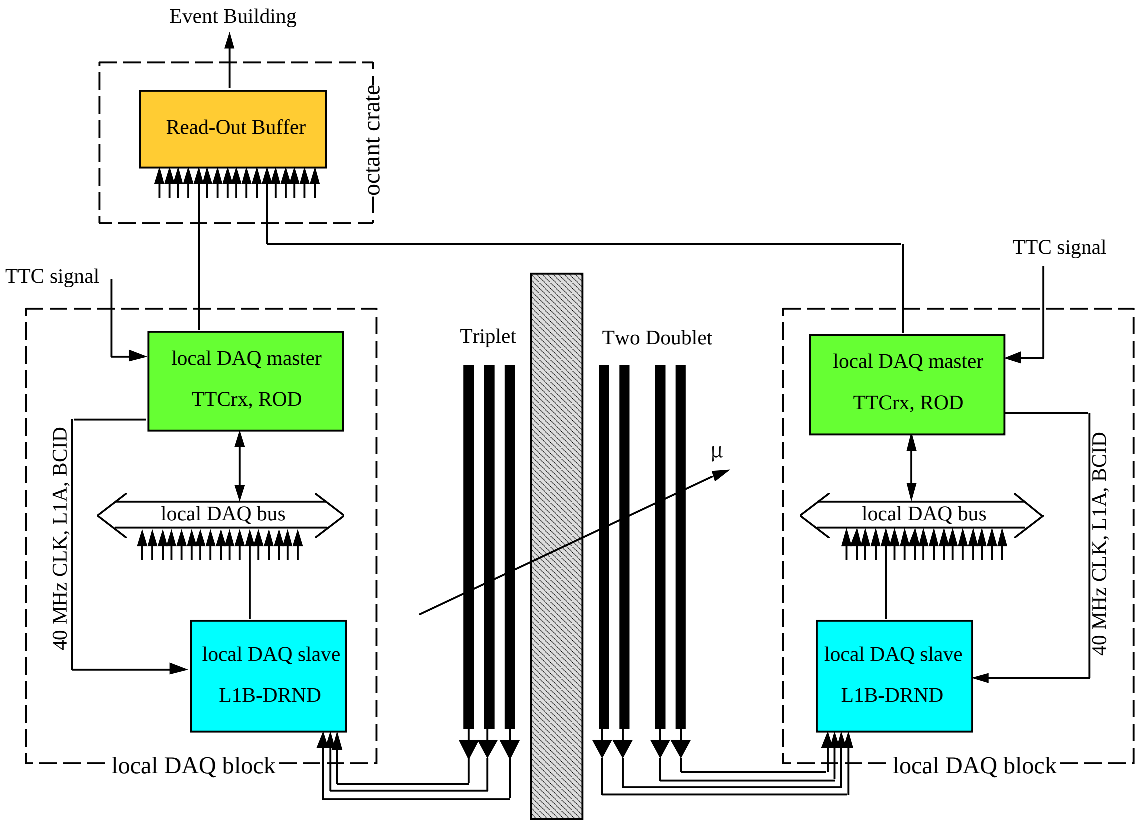


Fig. 7 TGC local DAQ block (Read-Out scheme)

We decided that each slave is responsible for 94 (for a triplet part) or 128 (for a two doublet part) channels of data, in other words, one slave board covers 32 channels x 3 or 4 layers as shown in Fig. 7 and Fig. 8. Since from 12 to 20 slave boards are placed per one local DAQ block, the maximum number of channels handled by one block is over 2000 and required typical translation speed of the DAQ bus is about 10 Mbytes per second as indicated in Fig.8. The local DAQ bus is now under development.

[5] Read-Out Driver described in the last in detail.

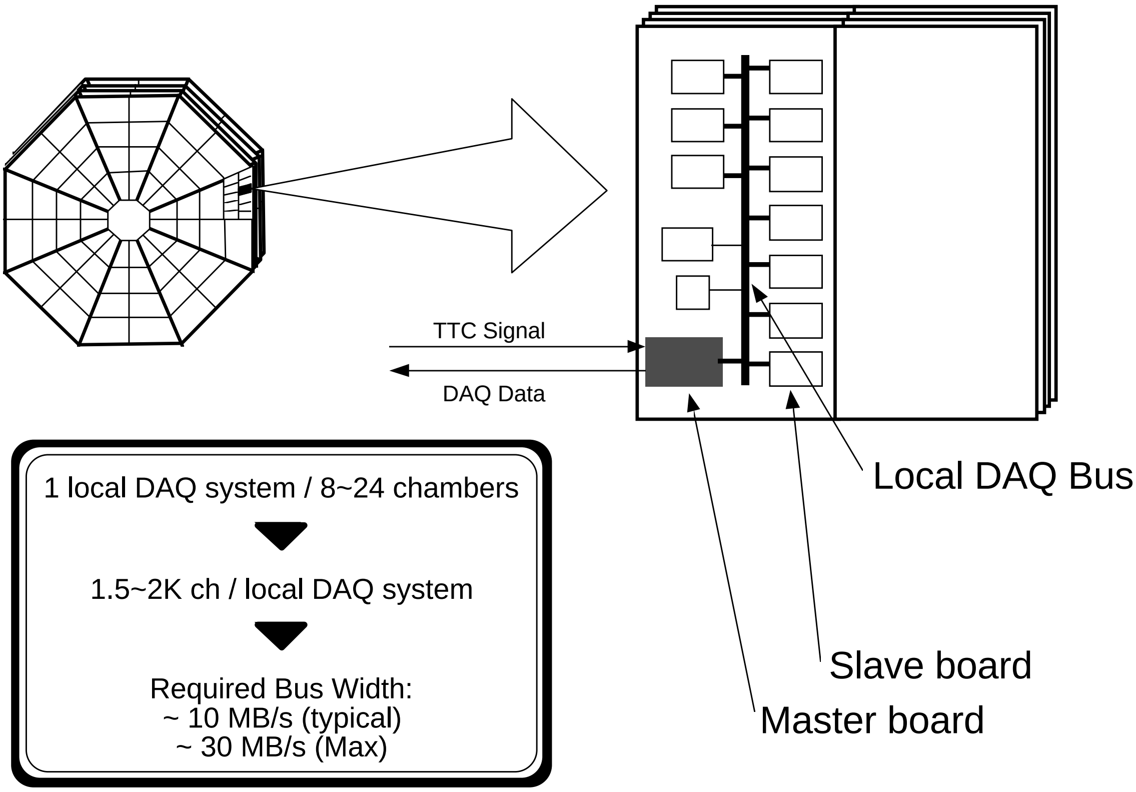


Fig. 8 Arrangement image of local DAQ unit

3.3 IEEE 1394 Serial Bus

In the several serial transfer system standards available in present electronics situation, we have taken notice of the IEEE 1394 serial bus [7]. IEEE 1394 is one of the newest data transfer protocol and expected that it is applied to main data transfer system for personal computer peripherals in place of the SCSI bus in the near future. IEEE 1394 has characteristic real time data transfer mode and provides maximum transfer speed of 400 M bit/sec. IEEE 1394 will satisfy the required condition of data transfer speed and cabling in the TGC front-end DAQ system. In addition this standard will be popular and then it is expected to become easy to get parts and developing environment because of the adoption of Personal Computer peripherals.

The author have attempted to use the IEEE 1394 serial bus for TGC front-end DAQ bus and studied the protocol and some products. However, IEEE 1394 is just fixed its specification and the products has hardly ever released from any manufacturers. So we are studying this bus and waiting for more useful products.

Chapter 4

Design of the Prototype Read-Out Slave Board

A prototype board has been produced in order to estimate the validity of the schemes, functions and parameters of the TGC local DAQ system. This board corresponds to the data read-out part of the slave board belonging to the front-end local DAQ bus. The trigger decision logic part as for the trigger chamber and the analogue to digital part are not installed.

4.1 Specifications

This board has been made of four layers printed circuit board on the VME 9U standard. The block diagram of the board is shown Fig.9 and the photograph is presented in Fig.10. Running clock frequency is 40 MHz, the same as the LHC clock. The main part of this board is Level 1 Buffer (L1B), Derandomizer and TTC signal input logic. A CPLD[6] (AMD Mach435 [8]) manages all the functional parts of this board. Level 1 buffer and Derandomizer is composed of FIFO memory[7]. The width of Level 1 buffer is 16 bit. The data come into Level 1 buffer is generated by the Pulse Pattern Generator (PPG) and sent via 16 channel flat cable. The PPG is a VME module made at KEK[8] and able to put out a 16 bit digital data from the on-board memory that can be read/write via VME. Pattern output frequency is up to 300 MHz. By making an appropriate data pattern and loading it into PPG memory via VME, an arbitrary DAQ data pattern is given into the prototype board.

The TTC signals is also generated by PPG. Although we had got the TTC receiver chip, there is no way to drive it in present. So the TTC receiver chip could not be installed in this board. The TTC signal interface block has been made of daughter board

[6] Complex Programmable Logic Device

[7] First In First Out memory

[8] National Laboratory for High Energy Physics, Japan

and in order to be able to substitute the daughter board onto the TTC receiver chip when it is available.

The main interface to readout the DAQ data is VME bus. Using VME bus makes the facility of debug, test, estimation of functions of the board. This board has been made by VME slave and available 32 bit normal data transfer and block transfer. In addition the other read-out interface, which is the IEEE 1394 protocol, is also prepared on this board to estimate the proposal mentioned in chapter 3. But this interface is now not available because interface controller chip is just recently released and inconvenience of installation still remains. This interface has also been made as the attachment of daughter board. In the near future more useful chip will be released and then this function will be available for transferring DAQ data instead of VME bus.

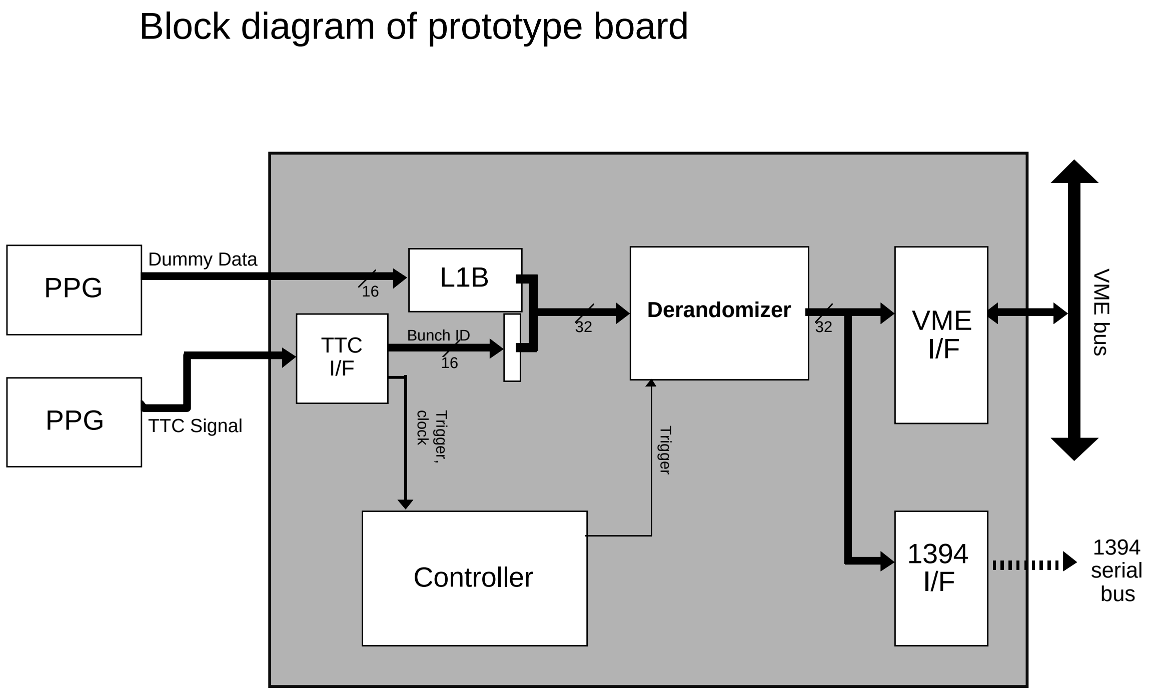


Fig. 9 Block diagram of prototype board

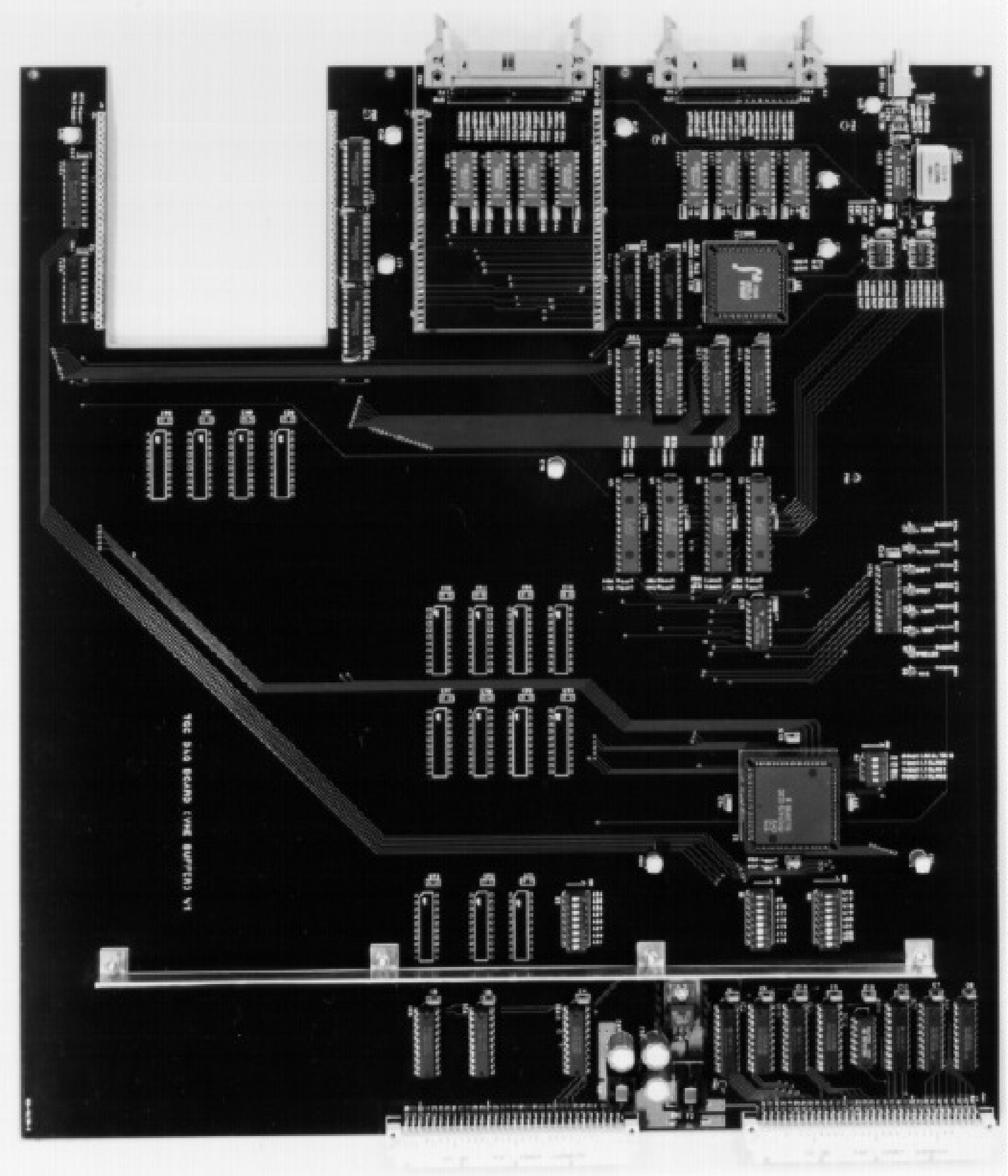


Fig. 10   Photograph of the prototype board

4.2 Functions and parameters

All the functions on this board are designed to meet the requirements of the T/DAQ system. Both Level 1 buffer and Derandomizer run synchronously with the PPG at 40 MHz.

4.2.1 Level 1 buffer

The memory size of the Level 1 buffer is 16 bit x 256 words. This corresponds to the time of 6.4 μ seconds to store the DAQ data at 40 MHz clock. Level 1 buffer behaves like pipeline and the length of the pipeline is able to adjust from 248 to 256 steps by a dip switch.

4.2.2 Derandomizer

The memory size of the Derandomizer is 32 bit x 2048 k words. Higher 16 bit correspond to DAQ data and lower 16 bit correspond to TTC signal (bunch ID and more). This memory depth has sufficient margin for the estimation and debug operation. Only when the trigger signal which is included in TTC signal is asserted, the Derandomizer stores the data which is given out from Level 1 buffer and bunch ID which given out from TTC signals. Untriggered data are never written into the Derandomizer and simply discarded.

4.2.3 TTC signal interface

TTC signal is 16 bit pattern information. The main signals are Level 1 trigger and 12 bit bunch ID. This pattern is generated for imitating the output signal of TTC receiver chip. This part is designed by the daughter board and if TTC receiver chip is available, we will change the daughter board to an appropriate one.

4.2.4 VME interface

The VME interface is used to readout the data from Derandomizer and to set various configuration (e.g. board reset, memory clear, DAQ operation switch). The 32 bit VME normal data transfer and block transfer are available.

4.2.5 IEEE 1394 interface

This interface is provided only as a connector. The prepared line is based upon the TSB11C01 and TSB12C01 chipset [9]. This chipset is released from Texas Instruments Inc.

4.3 Summary of the test run

Test run environment is shown in Fig.11. One-board Sun (UNIX) has been used as a VME master module to access all VME slave modules.

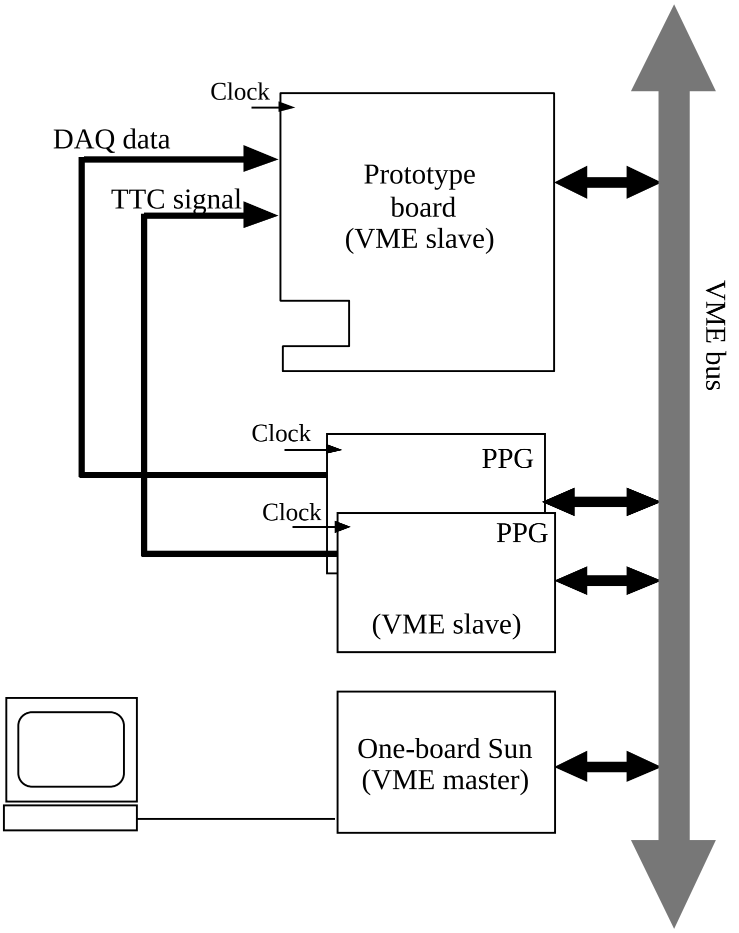


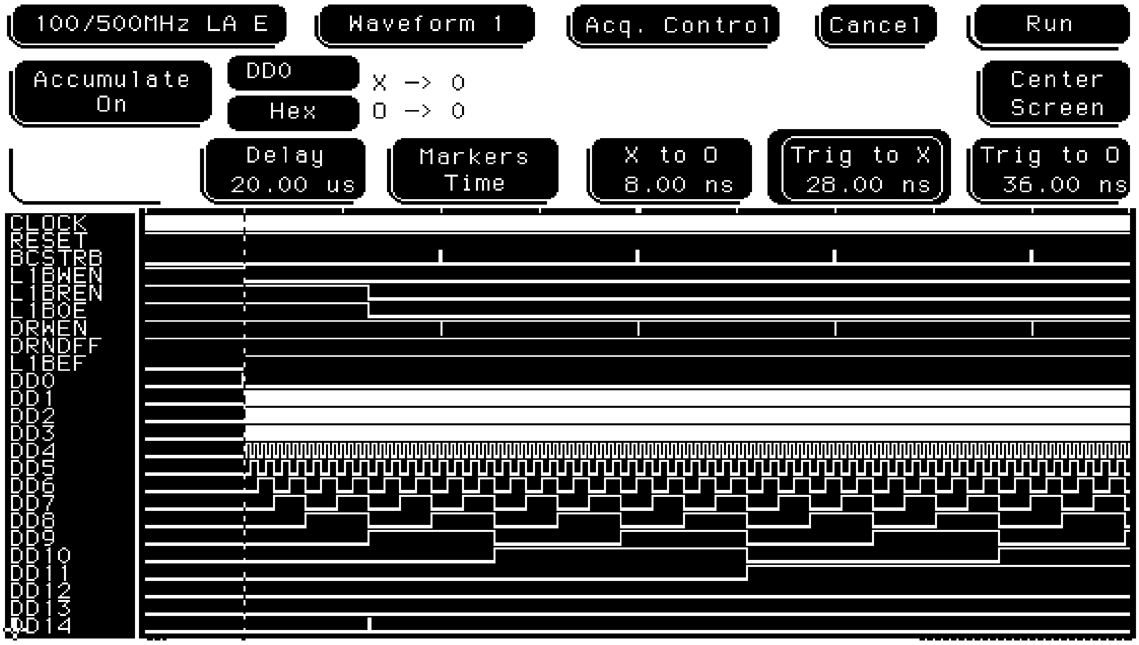
Fig. 11 Test run environment

The operation procedure is as follows,

1. fill the PPG memory with appropriate data pattern from one-board Sun,
2. put 40 MHz clock into PPG to start output of data from its memory, and
3. read out a data from the prototype board to one-board Sun via VME.

Some waveforms of the prototype slave board observed using logic analyzer and the explanation of it are presented in the figures.

DAQ operation start sequence



Accumulate On

DD0 Hex

X -> 0
0 -> 0

Delay 20.00 us

Markers Time

X to 0 8.00 ns

Trig to X 28.00 ns

Trig to 0 36.00 ns

Fig. 12 DAQ operation start sequence

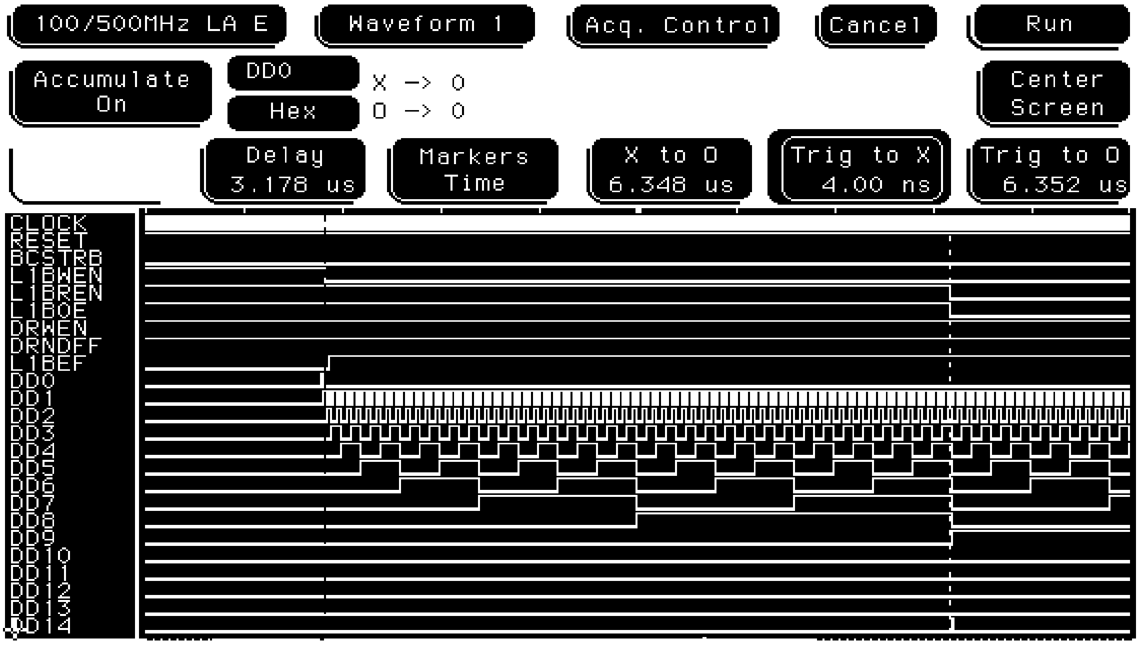
When the write enable line of the Level 1 buffer (described as L1BWEN in Fig.12) is asserted to low, Level 1 buffer starts writing successively the data from PPG with synchronization to the clock. As shown in Fig.12, read enable and output enable line (L1BREN and L1BOE) has been asserted to low after 254 clocks[9]. This means that Level 1 buffer has started putting out the data successively. Consequently Level 1 buffer behaves like pipeline that synchronously and simultaneously input and output the data.

We seem that the write enable line of the Derandomizer (corresponding to DRWEN in the waveform) has asserted to low only when the trigger line (BCSTRB) becomes high. This means that Derandomizer inputs only the triggered data and doesn’t input when not triggered.

The enlarged waveforms of Fig.12 is presented in Fig.13 and 14.

[9] This is the value when taking the data in Fig.12. 248 to 256 count is selectable.

Write enable low to read enable low



Accumulate On

DD0 Hex

X -> O

O -> O

Delay 3.178 us

Markers Time

X to O 6.348 us

Trig to X 4.00 ns

Trig to O 6.352 us

Fig. 13 Pipeline length

Fig.13 is the enlarged waveform between write enable line (L1BWEN) and read enable line (L1BREN) indicated in Fig.12. The time scale shown at the top of the waveform window is 1 micro second. This picture clearly shows the time from “L1BWEN” low to “L1BREN” low. It is 6.352 μ sec (shown in “Trig to 0” marks) which corresponds to the pipeline length of 254 steps at 40 MHz in this test. Therefore we seem that Level 1 buffer behaves like pipeline and the length of pipeline is same to the set up value of this test run.

Trigger to writing into Derandomizer

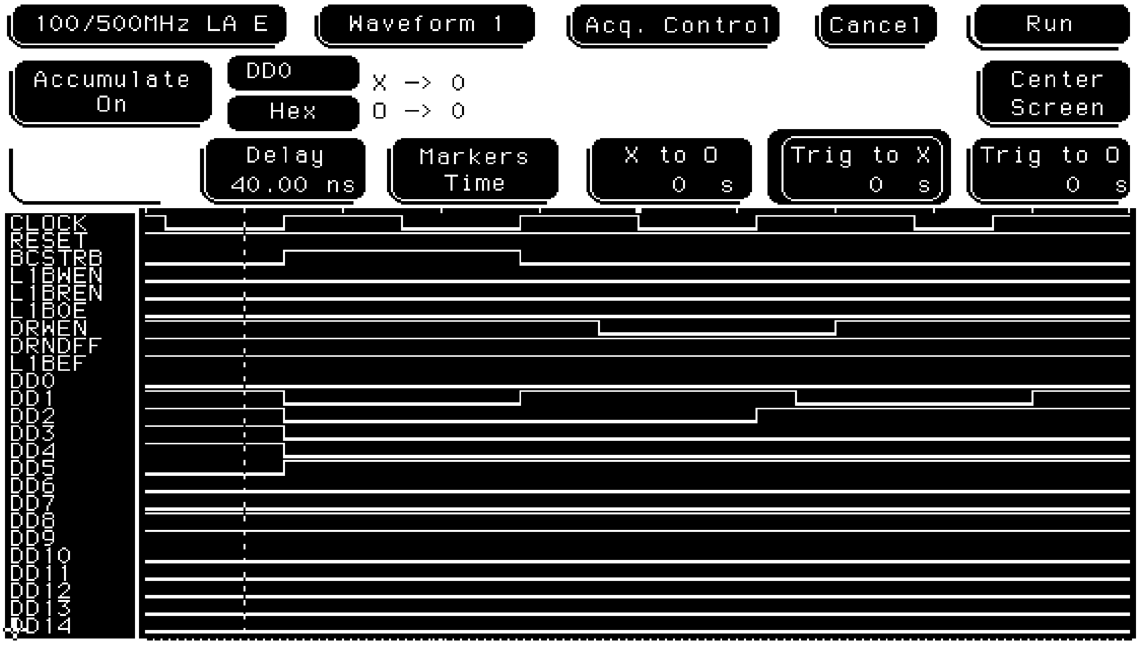
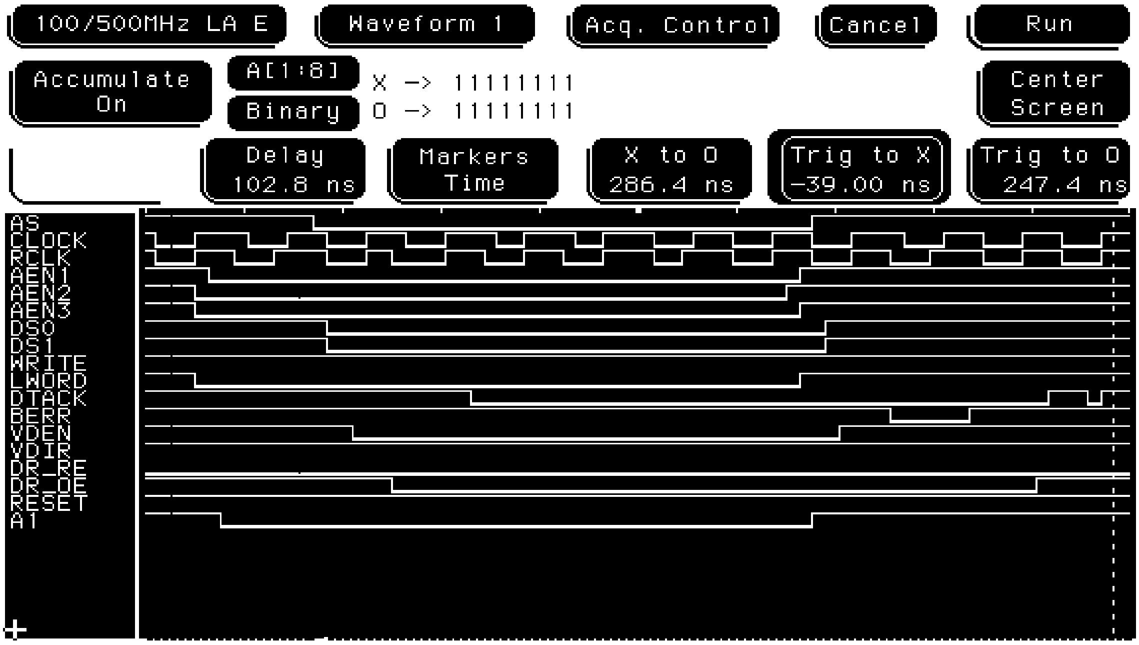


Fig. 14 Trigger to writing into Derandomizer

As shown in Fig.14, when the “BCSTRB” line which corresponds to trigger signal is asserted, write enable (DRWEN) is down to low at the next clock cycle. This means that Derandomizer writes the data from Level 1 buffer at the rising edge of the clock. DRWEN line remains to high whenever trigger (BCSTRB) is low in any clock cycle. Then we found that data the selection function by trigger was working as expectation.

VME read-out sequence



Accumulate On

A[1:8] Binary

X -> 11111111

O -> 11111111

Delay 102.8 ns

X to O 286.4 ns

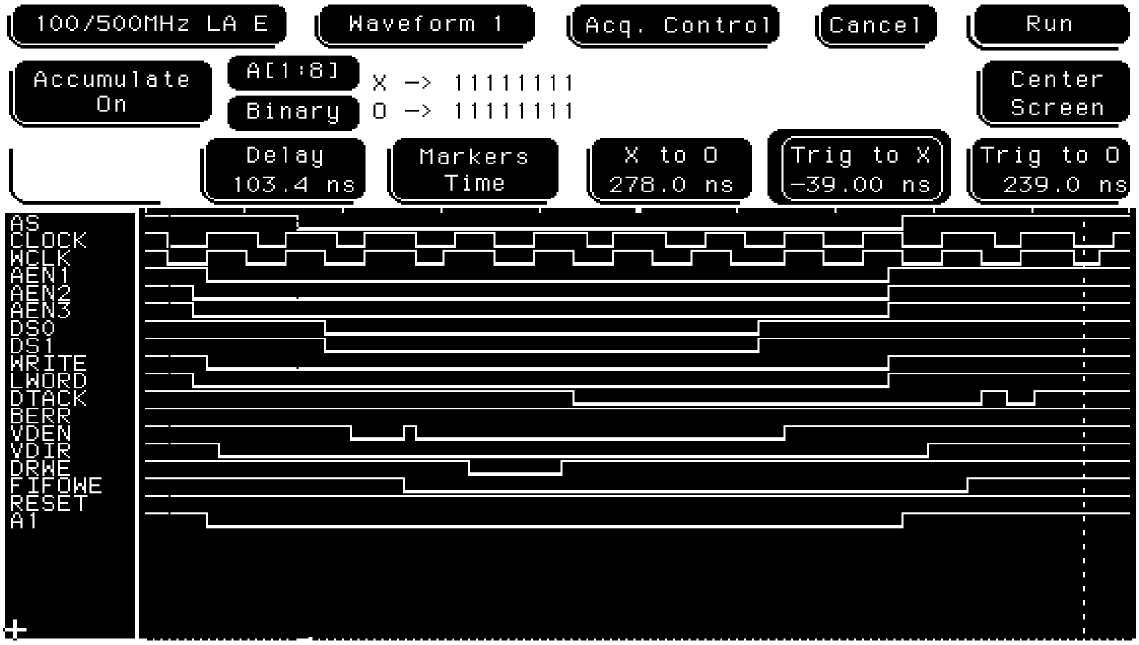
Trig to X -39.00 ns

Trig to O 247.4 ns

Fig. 15 VME read-out sequence

A VME read-out sequence of the normal transfer mode is presented in Fig.15. This waveform shows the speed of the VME interface logic. The time from “DS[0..1]” line low to “DTACK” line low is about 40 ns. This is sufficiently fast for general VME transfer logic.

VME write sequence



Accumulate On

A[1:8] Binary

X -> 11111111

O -> 11111111

Delay 103.4 ns

Markers Time

X to O 278.0 ns

Trig to X -39.00 ns

Trig to O 239.0 ns

Fig. 16 VME write sequence

A VME write sequence is presented in Fig.15. The VME write sequence is prepared in order to test and debug the Derandomizer and so practically it is not necessary the speed. The time from “DS[0..1]” low to “DTACK” low is about 70 ns.

Chapter 5

Summary

In this thesis, front-end Trigger and DAQ system for the Thin Gap Chamber was studied and prototype slave board for the local DAQ bus was produced.

We have been designing the Trigger and DAQ system in conformity with the requirement of ATLAS detector design and the condition of Thin Gap Chamber. The author have especially involved in the front-end DAQ system construction. I studied the requirement of the ATLAS DAQ and advanced the design along with the whole ATLAS DAQ system. Simultaneously the condition of the characteristics and the experimental situation of Thin Gap Chamber are considered to its design. At present, the front-end DAQ system is under development and the specifications and detailed parameters are being determined step by step.

I have produced the prototype slave board in order to verify the electronics design and to estimate the functionality of the local DAQ bus. This module is made by VME 9U board. This board includes a Level 1 buffer and a Derandomizer which are the main parts of the DAQ system. I checked these functions and found these works as expectation.

Additionally I studied and searched the data transfer methods of local DAQ bus and put a suggestion of using IEEE 1394 serial bus for it. The local DAQ bus is also now under development.

Appendix A

Hardware design specifications of prototype slave board

This is the schematics of the prototype slave board. This schematics has been drawn using OrCAD for Windows95.

Appendix B

Software design specifications of prototype slave board

This is the alpha version of program source of ABEL-HDL for designing the logic part of the prototype slave board. ABEL-HDL is the most popular language for designing the logic of a CPLD (complex programmable logic device). I used it to program the AMD Mach435 device.

```
//********************************************************
// TGC DAQ board project 1996 C.Fukunaga Lab.
// AMD Mach435
// board controller chip -- top module
// Written by    M.Fukuda Tokyo Metropolitan Univ. Oct 1996
//********************************************************


MODULE TGCDAQ


AMDMACH PROPERTY 'SYNC A14 lcount0';
AMDMACH PROPERTY 'SYNC A11 lcount1';
AMDMACH PROPERTY 'XOR_FACTOR 8';
AMDMACH PROPERTY 'GROUP B BERR_REQ';
AMDMACH PROPERTY 'GROUP D sreg_b0 sreg_b1 sreg_b2';
AMDMACH PROPERTY 'GROUP D sreg_a0 sreg_a1 sreg_a2';


DECLARATIONS


        //****** global lines ******//
        clk     pin 20;
        !reset  pin 19 istype 'reg';
        !vmeled pin 3  istype 'com';

        C       = .C.;
        X       = .X.;


        //****** VME lines ******//
        !vmereset                pin 18;
        !vas             pin 24;
        !vds0            pin 25;
        !vds1            pin 26;
        !vwrite          pin 27;
        !vlword          pin 28;
        !viack           pin 29;
        [vam0..vam5]     pin 12..17;
        vdtack           pin 30 istype 'reg,invert';
        vberr            pin 31 istype 'reg,invert';
        !vden            pin 45 istype 'com';
        vdir             pin 46 istype 'com';
        [a1..a4]         pin 33..36;
        a8               pin 37;
        ![aen1..aen3]    pin 38..40;

        DS_LAT        node istype 'reg,buffer';
        BERR_REQ      node istype 'reg,buffer';
        DERAND_READ    node istype 'com';
        DERAND_WRITE   node istype 'com';
        L1B_SWITCH     node istype 'com';
        FIFO_RESET     node istype 'com';
        SYS_RESET      node istype 'com';
        I1394_ACCESS     node istype 'com';
        AEN              node istype 'reg_D,buffer';
        ACCESS_ENB               node istype 'reg_D,buffer';
        ADS              node istype 'com';
        NO_TARGET                node istype 'com';
        DR_RD            node istype 'com';
        DR_WR            node istype 'com';
        LSW              node istype 'com';
        FRST             node istype 'com';
        SYSRST           node istype 'com';
        //****** Derandomizer control lines ******//
        !fifo_wen                pin 69 istype 'reg,invert';
        !derand_oe               pin 68 istype 'reg,invert';
        !derand_ren      pin 67 istype 'reg,invert';
        !derand_wen      pin 66 istype 'reg,invert';
        !local_trig      pin 52;
        !local_trig_on   pin 5 istype 'com';
        !derand_full     pin 7;
        !derand_empty    pin 8;
        L1A              pin 79;
        dd0              pin 78;
        BCstrobe         pin 77;
        !ttc_oe          pin 76 istype 'reg,invert';
        ttc_le           pin 75 istype 'reg,buffer';

        DR_RD_ENB                node istype 'com';
        DR_WR_ENB                node istype 'com';
        NORMAL_XFER      node istype 'com';
        BLOCK_XFER               node istype 'com';


        //****** L1 Buffer control lines ******//
        !L1B_oe          pin 82 istype 'reg, invert';
        !L1B_ren         pin 6 istype 'reg,invert';
        !L1B_wen         pin 70 istype 'reg,invert';
        !L1B_on          pin 4 istype 'reg_t,invert';
        !L1B_full                pin 9;
        !L1B_empty               pin 10;
        !fifo_length0    pin 51;
        !fifo_length1    pin 50;
        !fifo_length2    pin 49;

        lcount7..lcount0 node istype 'reg';

        L1B_COUNT        = [lcount7..lcount0];


        //****** IEEE 1394 control lines ******//
        bclk    pin 61 istype 'reg,buffer';
        I1394_CSpin 60 istype 'reg';
        I1394_CApin 59;
        I1394_wrpin 58 istype 'reg';
        I1394_int        pin 57 istype 'reg';
        I1394_aen        pin 56 istype 'reg';
        !I1394_den       pin 55 istype 'reg';
        I1394_dir        pin 54 istype 'reg';


        //****** Stete diagram control ******//
        sreg_a0..sreg_a2 node istype 'reg';
        sreg_b0..sreg_b2 node istype 'reg';
        sreg_c0, sreg_c1 node istype 'reg';

        DERAND_READ_STATE        = [sreg_a2..sreg_a0];
        DERAND_TEST_STATE        = [sreg_b2..sreg_b0];
        DAQ                      = [sreg_c1, sreg_c0];


//***********************************
//               GLOBAL LINES
//***********************************

EQUATIONS
```

```
        reset.CLK        = clk;

        reset   := vmereset # SYS_RESET # FIFO_RESET;
        vmeled  = AEN;



//************************************************
//               VME ACCESS
// Address decode, mode change, controls, etc.
//************************************************

EQUATIONS

        vdtack.CLK       = clk;
        vberr.CLK        = clk;
        DS_LAT.CLK       = clk;
        AEN.LE           = ADS;
        ACCESS_ENB.LE    = ADS;
        BERR_REQ.CLK     = clk;
        L1B_on.CLK       = DS_LAT;

        vdtack.CLR               = vmereset;
        vberr.CLR                = vmereset;
        DS_LAT.CLR               = vmereset;
        AEN.AR           = vmereset;
        ACCESS_ENB.AR    = vmereset;
        BERR_REQ.CLR     = vmereset;
        L1B_on.AR                = reset;

        DS_LAT.D         = vds0 & vds1;
        AEN.D            = aen1 & aen2 & aen3;
        ACCESS_ENB.D     = !viack & !a1 & vlword;
        ADS              = vas # vds0 # vds1;

        ////////////////////////////////////////////////////
        ////    VME Address decode
        ////
        ////    ADDRESS MAPPING
        //// xxxxx000   Derandomizer access
        //// xxxxx004   L1 Buffer run/stop (toggle)
        //// xxxxx008   Undefined reserved
        //// xxxxx00C   Undefined reserved
        //// xxxxx010   Undefined reserved
        //// xxxxx014   Undefined reserved
        //// xxxxx018   FIFO Reset
        //// xxxxx01C   SYSTEM Reset
        //// xxxxx020
        ////    :       same as of 00~1C
        //// xxxxx0FF
        //// xxxxx100
        ////    :       IEEE 1394
        //// xxxxx1FF
        ///////////////////////////////////////////////////

        DR_RD   = !vwrite & !a8 & !a4 & !a3 & !a2;
        DR_WR   =  vwrite & !a8 & !a4 & !a3 & !a2;
        LSW     =  vwrite & !a8 & !a4 & !a3 &  a2;
        FRST    =  vwrite & !a8 &  a4 &  a3 & !a2;
        SYSRST  =  vwrite & !a8 &  a4 &  a3 &  a2;

        DERAND_READ      = ADS & AEN & ACCESS_ENB & DR_RD;
        DERAND_WRITE     = ADS & AEN & ACCESS_ENB & DR_WR;
        L1B_SWITCH               = ADS & AEN & ACCESS_ENB & LSW;
        FIFO_RESET               = ADS & AEN & ACCESS_ENB & FRST;
        SYS_RESET                = ADS & AEN & ACCESS_ENB & SYSRST;
        I1394_ACCESS     = ADS & AEN & ACCESS_ENB &  a8;

        NO_TARGET                = ADS & AEN & ACCESS_ENB
                                 & !DR_RD  & !DR_WR  &  !LSW  &  !FRST
& !SYSRST;

        L1B_on.T= L1B_SWITCH;

        vberr    := DS_LAT & (
                         I1394_ACCESS
                         #
                         AEN & ACCESS_ENB & NO_TARGET
                         #
                         AEN & !ACCESS_ENB
                         #
                         BERR_REQ);

        vdir     = !vwrite;

        vden     = DS_LAT & (DERAND_READ # DERAND_WRITE);


        when(DS_LAT & (L1B_SWITCH # FIFO_RESET # SYS_RESET)) then
                 vdtack   := 1;
        else
                 vdtack   := 0;



//*********************************************************
//               DERANDOMIZER CONTROL
// Date readout and test-data write via VME bus.
// VME D32 normal and block Xfer are supported.
// BERR is asserted if FIFO is empty when readout request.
// If AM code is NOT equal to 09h, 0Ah or 0Bh, BERR is asserted.
// When written test-data, L1 Buffer should be stopped.
//**********************************************************

EQUATIONS

        derand_ren.CLK   = clk;
        derand_wen.CLK   = clk;
        derand_oe.CLK    = clk;
        fifo_wen.CLK     = clk;
        ttc_oe.CLK               = clk;
        ttc_le.CLK               = clk;
        [sreg_a0..sreg_a2].CLK   = clk;
        [sreg_b0..sreg_b2].CLK   = clk;

        derand_ren.CLR   = reset;
        derand_wen.CLR   = reset;
        derand_oe.CLR    = reset;
        fifo_wen.CLR     = reset;
        ttc_oe.CLR               = reset;
        ttc_le.CLR               = reset;
        [sreg_a0..sreg_a2].CLR   = reset;
        [sreg_b0..sreg_b2].CLR   = reset;

        local_trig_on    = local_trig;

        DR_RD_ENB        = !derand_empty & !fifo_wen & !I1394_den;
        DR_WR_ENB        = !derand_full & !L1B_on & !ttc_oe & !derand_oe
& !I1394_den;
        NORMAL_XFER      = !vam5 & !vam4 &  vam3 & !vam1 &  vam0
                           #
                           !vam5 & !vam4 &  vam3 &  vam1 & !vam0;
        BLOCK_XFER              = !vam5 & !vam4 &  vam3 &  vam1 &  vam0;


STATE_DIAGRAM    DERAND_READ_STATE
   //// State diagram for Derandomizer Read Access via VME.
   //// If VME AM[0:5] is NOT equel to 09h, 0Ah (32bit normal Xfer)
   //// or 0Bh (32bit block Xfer), then BERR is asserted.

        STATE 0:
        derand_ren       := 0;
        derand_oe        := 0;
        vdtack  := 0;
        BERR_REQ := 0;
        if (DERAND_READ) then
                if (DR_RD_ENB) then
                        if (NORMAL_XFER) then
                                 1 with {derand_ren := 1; derand_oe :=
1;}
                        else if (BLOCK_XFER) then
                                 1 with {derand_ren := 1; derand_oe :=
1;}
                        else 5 with BERR_REQ := 1;        //  bad  AM
code
                else 5 with BERR_REQ := 1;                //  bad  VME
access
        else 0;

        STATE 1:
                derand_oe        := 1;
                vdtack  := 0;
                BERR_REQ := 0;
                if(DS_LAT) then 2 with {derand_ren := 0; vdtack := 1;}
                else 1 with derand_ren := 0;

        STATE 2:
                derand_ren       := 0;
                derand_oe        := 1;
                vdtack  := 1;
                BERR_REQ := 0;
                if(!DS_LAT) then 0;
                else 2;

        STATE 3:
                derand_oe        := 1;
                vdtack  := 0;
                BERR_REQ := 0;
                if(DS_LAT) then 4 with {derand_ren := 0; vdtack := 1;}
                else 3 with derand_ren := 0;

        STATE 4:
                derand_ren       := 0;
                derand_oe        := 1;
                vdtack  := 1;
                BERR_REQ := 0;
                if(!BLOCK_XFER) then
                        0  with  {derand_ren  :=  0;  derand_oe  :=  0;
vdtack := 0;}
                else if(!DS_LAT) then 3 with {derand_ren := 1; vdtack :=
0;}
                else 4;

        STATE 5:
                derand_ren       := 0;
                derand_oe        := 0;
                vdtack  := 0;
                BERR_REQ := 1;
                if(!DS_LAT & vberr) then 0 with BERR_REQ := 0;
                else 5;


STATE_DIAGRAM   DERAND_TEST_STATE
        //// State diagram for Derandomizer Write Access via VME.
        //// This cycle is for FIFO test.
        //// If bad AM code, FIFO full or Running L1B is caught,
        //// then BERR is asserted.

        STATE 0:
                derand_wen       := 0;
                fifo_wen := 0;
                vdtack   := 0;
                BERR_REQ := 0;
                if(DERAND_WRITE) then
                        if(DR_WR_ENB) then
                                if(NORMAL_XFER) then 6 with fifo_wen :=
1;
                                else  if(BLOCK_XFER)  then  3  with
fifo_wen := 1;
                                else 5 with BERR_REQ := 1;       //
bad AM code
                        else 5 with BERR_REQ := 1;       //  bad  VME
access
                else 0;

        STATE 6:
                derand_wen       := 0;
                fifo_wen := 1;
                vdtack   := 0;
                BERR_REQ := 0;
                goto 7;

        STATE 7:
                derand_wen       := 0;
                fifo_wen := 1;
                vdtack   := 0;
                BERR_REQ := 0;
                goto 1;

        STATE 1:
                derand_wen       := 0;
                fifo_wen := 1;
                vdtack   := 0;
                BERR_REQ := 0;
                if(DS_LAT) then 2 with derand_wen := 1;
                else 1;

        STATE 2:
                fifo_wen := 1;
                vdtack   := 1;
                BERR_REQ := 0;
                if(!DS_LAT) then 0 with {derand_wen := 0; fifo_wen :=
0;}
                else 2 with derand_wen := 0;

        STATE 3:
                derand_wen       := 0;
                fifo_wen := 1;
                vdtack   := 0;
                BERR_REQ := 0;
                if(!BLOCK_XFER) then 0 with fifo_wen := 0;
                else if(DS_LAT) then 4 with derand_wen := 1;
                else 3;
        STATE 4:
                fifo_wen:= 1;
                vdtack   := 1;
                BERR_REQ:= 0;
                if(!DS_LAT) then 3 with derand_wen := 0;
                else 4 with derand_wen := 0;

        STATE 5:
                derand_wen      := 0;
                fifo_wen:= 0;
                vdtack   := 0;
                BERR_REQ:= 1;
                if(!DS_LAT & vberr) then 0 with BERR_REQ := 0;
                else 5;


//*********************************************************
//              L1 BUFFER CONTROL
// L1B behaves like a pipeline.
// The length of pipeline is variable from 249 ~ 256
// steps owing to L1B_LENGTH choiced by 3-bit dip switch.
//*********************************************************

EQUATIONS

        L1B_ren.CLK     = clk;
        L1B_wen.CLK     = clk;
        L1B_oe.CLK              = clk;
        DAQ.clk         = clk;
        L1B_COUNT.CLK   = clk;

        L1B_ren.CLR     = reset;
        L1B_wen.CLR     = reset;
        L1B_oe.CLR              = reset;
        DAQ.CLR         = reset;
        L1B_COUNT.CLR   = reset;


STATE_DIAGRAM   DAQ

        State 0:
                L1B_wen  := 0;
                L1B_ren  := 0;
                L1B_oe   := 0;
                derand_wen      := 0;
                ttc_le   := 0;
                ttc_oe   := 0;
                L1B_COUNT := 0;
                if(L1B_on) then 1;
                else 0;

        State 1:
                L1B_wen  := 0;
                L1B_ren  := 0;
                L1B_oe   := 0;
                if(!L1B_on) then 0;
                else if(dd0) then 2 with
                        {L1B_wen := 1; L1B_COUNT := L1B_COUNT + 1;}
                else 1;

        State 2:
                L1B_wen  := 1;
                L1B_ren  := 0;
                L1B_oe   := 0;
                L1B_COUNT := L1B_COUNT + 1;
                if(!L1B_on # L1B_full) then 0;
                else if(L1B_COUNT == 254) then 3 with
                        {L1B_ren := 1; L1B_oe := 1;}
                else 2;

        State 3:
                L1B_wen := 1;
                L1B_ren := 1;
                L1B_oe  := 1;
                if(!L1B_on # L1B_full # L1B_empty) then 0;
                else if(BCstrobe & !derand_full) then 3 with
                        {derand_wen := 1; ttc_le := 1; ttc_oe := 1;}
                else 3 with {derand_wen := 0; ttc_le := 0; ttc_oe := 0;}


//******************************************
//              IEEE1394 RELATED LINE
//******************************************

EQUATIONS

        bclk.CLK        = clk;
        I1394_CS.CLK    = bclk;
        I1394_wr.CLK    = bclk;
        I1394_int.CLK   = bclk;
        I1394_aen.CLK   = bclk;
        I1394_den.CLK   = bclk;
        I1394_dir.CLK   = bclk;

        I1394_CS.CLR    = vmereset # SYS_RESET;
        I1394_wr.CLR    = vmereset # SYS_RESET;
        I1394_int.CLR   = vmereset # SYS_RESET;
        I1394_aen.CLR   = vmereset # SYS_RESET;
        I1394_den.CLR   = vmereset # SYS_RESET;
        I1394_dir.CLR   = vmereset # SYS_RESET;

        bclk.D          = !bclk; // bclk = 20MHz

        // Both CA and L1A are not used lines.
        // Change the equation below when using 1394 function.
        I1394_CS := I1394_CA & L1A & vam2;
        I1394_wr := 0;
        I1394_int       := 0;
        I1394_aen       := 0;
        I1394_den       := 0;
        I1394_dir       := 0;


END
```

References

[1] S.Dado et al., “A New High Gain Thin Gap Detector for the OPAL Hadron Calorimeter”, Nucl. Inst. and Meth. A252 (1986) 511-516.

[2] G.Mikenberg, “Thin-Gap Gas Chambers for Hadronic Calorimetry”, Nucl. Inst. and Meth. A265 (1988) 223-227.

[3] “ATLAS Technical Proposal for a General-Purpose pp Experiment at the Large Hadron Collider at CERN”, December 1994.

[4] “Trigger & DAQ Interfaces with Front-End System: Requirement Document”, January 1996.

[5] J.Christiansen et al, “Receiver ASIC for Timing, Trigger and control Distribution in LHC Experiments”, Presented at the IEEE Nuclear Science Symposium, San Francisco, 21-28 October 1995.

[6] J.Christiansen et al, “TTCrx Reference Manual”, CERN ECP/MIC, March 1996, http://pcvlsi0.cern.ch/ttc/ttc_manual.pdf

[7] IEEE Standards Department, “P1394 Standard for a High Performance Serial Bus”, P1394 Draft 8.0v4, November 1995.

[8] AMD, “PAL device Data Book and Design Guide”, 1995.

[9] Texas Instruments, “Network System Products Data Book”, 1995.