E897/BTeV

BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA

FACULTAD DE CIENCIAS FISICO MATEMATICAS

POSGRADO EN OPTOLECTRONICA

THESIS

"The Development of Software to Characterize the Fermilab Pixel Readout Chip for the BTeV experiment"

Submitted in satisfaction of the requirements for the Ph. D. in Physics with Specialty in Optoelectronics

Present

M. C. Maria Aurora Diozcora Vargas Treviño [1]

TREVINO

Advisor

Dr. Marleigh Sheaff [2]

December 2000

---

[1] *Scholarship holder from CONACyT, MEXICO/Work developed in a stay at Fermilab, Batavia, Ill, USA*
[2] *University of Wisconsin, Madison, USA/CINVESTAV, MEXICO*

ACKNOWLEDGEMENTS

I want to thank God for allowing me to be able to meet this enormous challenge in my life.

I want to express my most sincere acknowledgement to my husband Sergio Vergara Limon for all his help in the development of this work and for loving me so much, and to my daughters Teresita and Aurorita for their patience and love.

I want to acknowledge my parents Lic. Marciano Vargas Minor and Lic. Alejandra Concepcion Treviño Lopez, because for their help and love for me.

I whish to thank my advisor Marleigh Sheaff for all her help and advice during the curse of this project.

I whish to thank S. Zimmermann for advice during the course of this project and G. Chiodini for technical assistance. I also want to express my appreciation to the Fermilab staff and management, with special thanks to J. Appel, S. Kwan, and E. Barsotti. This work would not have been possible without the support and encouragement of A. Cordero and A. Fernandez of FCFM/BUAP. The support of the U.S. DOE and CONACyT, Mexico, is also gratefully acknowledged.

TABLE OF CONTENTS

CHAPTER THREE

CHARACTERIZATION

CHAPTER FOUR

SOFTWARE

CHAPTER FIVE

RESULTS OF THE CHARACTERIZATION

INTRODUCTION

High-energy physics is the science of the fundamental nature of matter. Studying subatomic particles and forces gives us a key to understanding the simple physical laws that govern the universe. In 1965, the United States Joint Committee on Atomic Energy (JCAE) and the National Academy of Sciences (NAS) approved a frontier high energy physics project to develop a 200 GeV Accelerator. In 1967, Robert R. Wilson was chosen by URA (Universities Research Association, Inc) as the first Director of the National Accelerator Laboratory (NAL). In 1974, the Laboratory was renamed in honor of Enrico Fermi as Fermi National Accelerator Laboratory. Physicists use accelerators of higher and higher energies to probe deeper and deeper inside the nucleus. Like a more powerful microscope, the higher and higher energy accelerators enable the investigation of smaller and smaller distances, by now even distances inside the proton. Higher energy accelerators also enable the production of heavier and rarer particles.

The search for new particles using the Fermilab accelerator produced a discovery in 1977, the first evidence for the bottom quark. Later on the first superconducting accelerator was constructed in the same tunnel as the original one, called the main ring. This new accelerator was later transformed into a proton-antiproton collider. The beam of particles begins as negative hydrogen ions in the Cockcroft-Walton accelerator. They continue to the Linac (Linear accelerator). As the beam of negative hydrogen ions enters the third accelerator, the circular Booster, both electrons are stripped off leaving a proton beam. Finally the protons are injected into the Main Ring. The antiproton source was essential to produce the proton's opposite particle. These antiprotons could then be steered into collision with protons and observed in specially designed detectors. The energy of these collisions would be close to 2 TeV in the center of mass. In 1985 the beam reached 800 GeV, and the first collisions of protons and antiprotons (combined energy of 1.6 TeV) were observed at the Colliding Detector at Fermilab (CDF). With the highest energy yet achieved the most powerful superconducting accelerator in the world the began the search for the most exotic particle within reach: the top quark. Two specialized detectors were constructed by large teams of experimenters at CDF and at DZero. In 1995 both the CDF and Dzero teams announced the top's discovery.

Under the command of John Peoples until 1999 and now under Michel Witherell the most powerful particle accelerator on earth, Fermilab's Tevatron, gives scientists from all over the United States and the world the opportunity to work together on experiments to try to understand the laws of nature. The Tevatron accelerates protons and antiprotons in a giant underground ring. When proton and antiproton collide at close to the speed of light, they make a tiny fireball of pure energy as intense as the big bang, when the universe was a trillionth of a

second old. Some of the energy turns into matter, according to Einstein's famous equation, $E = mC^2$, yielding sprays of particles that may hold answers to our questions about the laws and origin of the universe.

As the program at Fermilab moves forward the particle physics field focuses inside the quark and beyond. Fixed-target and colliding-beams experiments continue their searches on the frontier. In order to contribute to a deeper understanding of the heavy quarks, bottom and top. Fermilab has approved the BTeV experiment, E897. The new Fermilab's Tevatron will produce more than 400 billion b-flavored hadrons per year and 10 times as many c-flavored hadrons per year. A hadron is a particle made of strongly-interacting constituents (quarks and/or gluons). These include the mesons and baryons. Such particles participate in residual strong interactions. The heavy-flavored band a hadrons will be an excellent resource with which to investigate CP violation, quark/anti-quark mixing and rare decays. BTeV will be well positioned to answer the most crucial questions in heavy flavor physics. BTeV will use a powerful magnet, called SM3,which already exists at Fermilab. The other important parts of the experiment include the vertex detector, the RICH detectors, the EM calorimeters, and the muon system.

The pixel detector is a multi-plane pixel device that will sit inside the beam pipe. This silicon pixel detector has been proposed to be used in the first level trigger of the BTeV experiment. Thus, the BTeV pixel detector is central to the capability of the BTeV experiment. The major components for the pixel detector system will be the sensor, readout chip, sensor-to-readout-chip connection, mechanical support and cooling, high-density interconnection between the readout chips and the control chips on the detector plane peripheries, and the multiplexing and data transmission to the first-level trigger.

The silicon pixel detector will be composed of 62 pixel planes of 100x100mm each, divided into 31 stations with two planes each, placed perpendicular to the colliding beam and installed a few millimeters from the beam. VLSI pixel readout chips containing front-end electronics for every pixel sensor will be bump-bonded to the detector. Since this detector will be employed for on-line track finding for the lowest level trigger system, the pixel chip will have to read out all detected hits. The pixel chip development involves a succession of steps and submissions toward a chip that meets the BTeV requirements. VLSI chips with pixel unit cells of the same dimensions have been designed and built to instrument the sensors. The chips resulting from these steps have been dubbed FPIX0, FPIX1, and so on. The FPIX1 represents the first step towards the final pixel readout architecture. The FPIX1 chip is a column-based pixel chip with $50\mu m \times 400\mu m$ pixel cells arranged in an array of 160 rows by 18 columns for a total of 2880 pixel cells. Current FPIX1 prototypes have sensitive areas of up to 59.2 mm^2. In order to gain experience on the technical issues Fermilab has made a five-readout-chip sub-assembly with flexible cable interconnections, called the Fermilab multi chip module, or MCM.

Since each FPIX1 chip contains 2880 channels, and the experiment will use about 13000 chips, the task of characterizing them is challenging and potentially time consuming. We developed software within the LabVIEW framework to control a set of instruments to perform bad pixel map, threshold and noise tests on a full chip in a few hours of running. We also developed software to characterize the Fermilab Pixel module. My thesis work was performed as part of the research and development needed in order to build the pixel detector.

The main goal of the thesis is:

"To develop a test stand including the test setup and software to characterize the Fermilab Pixel Readout chip (FPIX1) for the BTeV experiment"

In order to research the main goal of my thesis, we will describe as an abstract all the chapters containing in my thesis as follows:

In chapter one we give a brief explanation of the BTeV experiment. First we discuss the requirements of the experiment, explaining its goal. We also include a brief explanation of the parts of the experiment, e.g., the pixel detector.

In chapter two we describe the FPIX1 chip and the multi chip module (MCM). We give an extended explanation of how to handle and operate this chip. We give details on how we designed the patterns to control the chip in order to download and readout information from it. We show real photographs of the chip and also for the multi-chip module that contains five FPIX1 chips connected in daisy chain. As for the single FPIX1 chip we describe the correct way to download to and readout information from the MCM, describing all the signals that allow us to characterize it.

In chapter three we cover in detail the characterization of the FPIX1 chip, explaining the different tests that we performed on it, including the bad pixel map test, threshold uniformity test without sensors and with sensors bump-bonded to it, the reproducibility test, hit studies and other studies of interest. Later in this chapter we describe the test setup arranged to carry out all the characterization.

In chapter four we will describe the first test stand used (setup and software) to characterize the FPIX1 chips, which took about 157 hours (6.5 days) of running to characterize only one FPIX1 chip, which means 786 hours (32 days) of running to characterize 5 FPIX1 chips connected in daisy chain. The final version after an optimization of the test setup and software takes only 6 hours of running to characterize an MCM with 5 FPIX1 chips connected in daisy chain. And it could be possible decrease the running time injecting more than 17 cells at a time, because the most recently test using a computer with a 750 MHz processor speed shows that the process time is of only about 2 hours of running to characterize a MCM with 5 FPIX1 chips connected in daisy chain. Also, we show the development of the software used in the last version of the test stand. The software developed utilizes the LabVIEW (Laboratory Virtual Instrument Engineering Workbench) framework. LabVIEW is a development environment based on the graphical programming language G. It is integrated fully for communication with hardware such as GPIB, VXI, PXI, RS-232, RS-485, and plug in data acquisition boards. The software is a very important part of the tests, because it is by means of the software that we can control all the phases in the test. We describe the data acquisition system and the data processing system.

In chapter five we report the results obtained in the tests that we described in chapter three. Finally we give the references show and two appendices. As appendix A we show the layout and pin assignment for the single FPIX1 chip as mounted on the single-chip test board, and as appendix B, we show the layout and pin assignment of the multi chip module (MCM) assembly.

With the automatic characterization of the MCMs we are contributing to develop the first prototype of the half pixel plane for the BTeV Pixel Detector.

With this work we did the follow presentations and publications

Presentations:

1. M.A. Vargas, M. Sheaff, S. Vergara, "The development of software to characterize the fermilab pixel readout chip for the BTeV experiment", Poster presentation, SOMI XIV CONGRESO DE INSTRUMENTACION, Tonanzintla, Puebla, MEXICO 1999.
2. S. Zimmermann, S. Kwan, G. Cancelo, G. Cardoso, S. Cihangir, D. Christian, R. Downing, M. A. Vargas, et al., "Development of high data readout rate pixel module and detector hybridization at Fermilab", Poster presentation, PIXEL 2000, Genoa, June 2000.
3. G. Cancelo, S. Vergara, M.A. Vargas, et al., "Fiber optic based readout for the BteV's pixel detector", Poster presentation, LEB Sixth workshop on electronics for the LHC experiment, Cracow, Poland, September 2000.

Publications:

1. M.A. Vargas, M. Sheaff, S. Vergara, "The development of software to characterize the fermilab pixel readout chip for the BTeV experiment", Preceedings of the SOMI XIV CONGRESO DE INSTRUMENTACION, pp. 608-612, Tonanzintla, Puebla, MEXICO 1999.
2. M.A. Vargas, M. Sheaff, S. Vergara, "The development of software to characterize the Fermilab pixel readout chip, FPIX1, for the BTeV experiment", Accepted to NIMA, November 2000. Fermilab preprint FERMILAB-PUB-00-244E, November 2000.
3. S. Zimmermann, S. Kwan, G. Cancelo, G. Cardoso, S. Cihangir, D. Christian, R. Downing, M. A. Vargas, et al., "Development of high data readout rate pixel module and detector hybridization at Fermilab", Preceedings of the PIXEL 2000, Genoa, June 2000.
4. G. Cancelo, S. Vergara, M.A. Vargas, et al., "Fiber optic based readout for the BteV's pixel detector", Preceedings of the LEB Sixth workshop on electronics for the LHC experiment, Cracow, Poland, October 2000.

CHAPTER ONE

THE BTeV EXPERIMENT AT FERMILAB

The subject of the BTeV experiment is to learn more about the bottom and charm quarks. The experiment will use the Fermilab Tevatron. The accelerator will produce more than 400 billion b-flavored hadrons per year and 10 times as many c-flavored hadrons per year. These particles participate in residual strong interactions. The heavy flavored hadrons will be an excellent resource with which to investigate CP violation, quark/anti-quark mixing and rare decays. With all the results expected from the experiment, BTeV will be well positioned to answer the most crucial questions in heavy flavor physics.

1.1 THE BTeV EXPERIMENT

To understand better the subject of the BTeV experiment, we can see the atom's structure. The atom is know to contains many more elementary particles inside. The structure within the atom [1] is shown as Fig. 1.1.

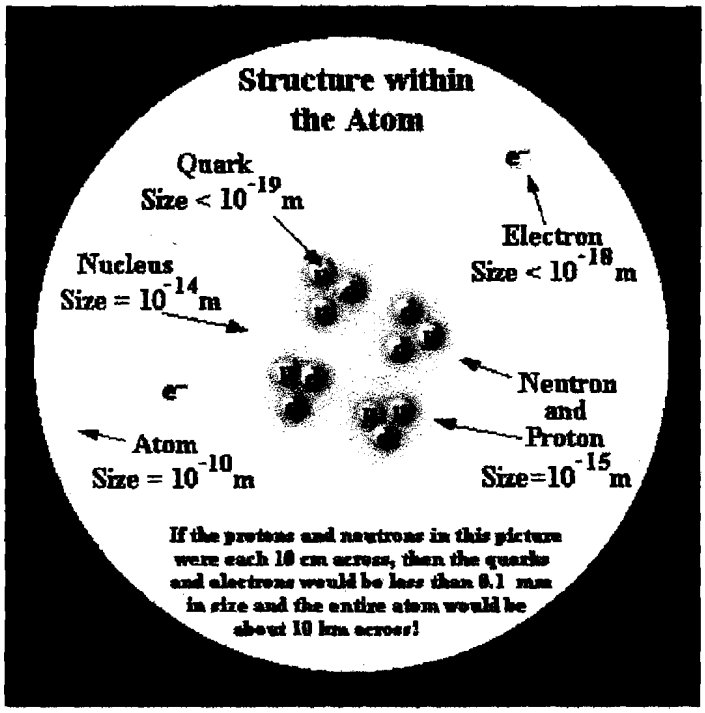


Figure 1.1: The structure within the atom (this picture is a conception of an artist in a work made for Fermilab).

We can see that the neutrons and protons contain quarks. Also, inside the atom there are leptons. The elementary particles and force carriers [1] are shown in Fig. 1.2.

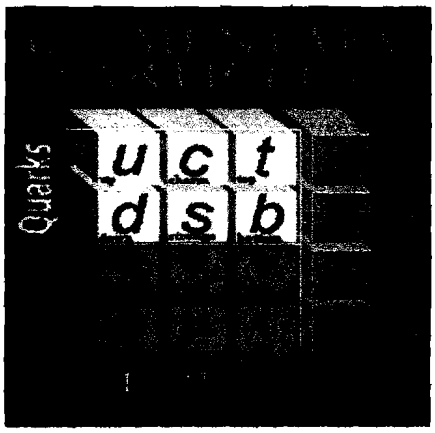


Figure 1.2: Elementary particles.

The BTeV experiments will use a powerful magnet, this magnet will generates 2.6 T-m. The other important parts of the experiment include the vertex detector, the RICH detectors, the EM calorimeters and the muon system [2]. A sketch of the future apparatus is shown as Fig.1.3.

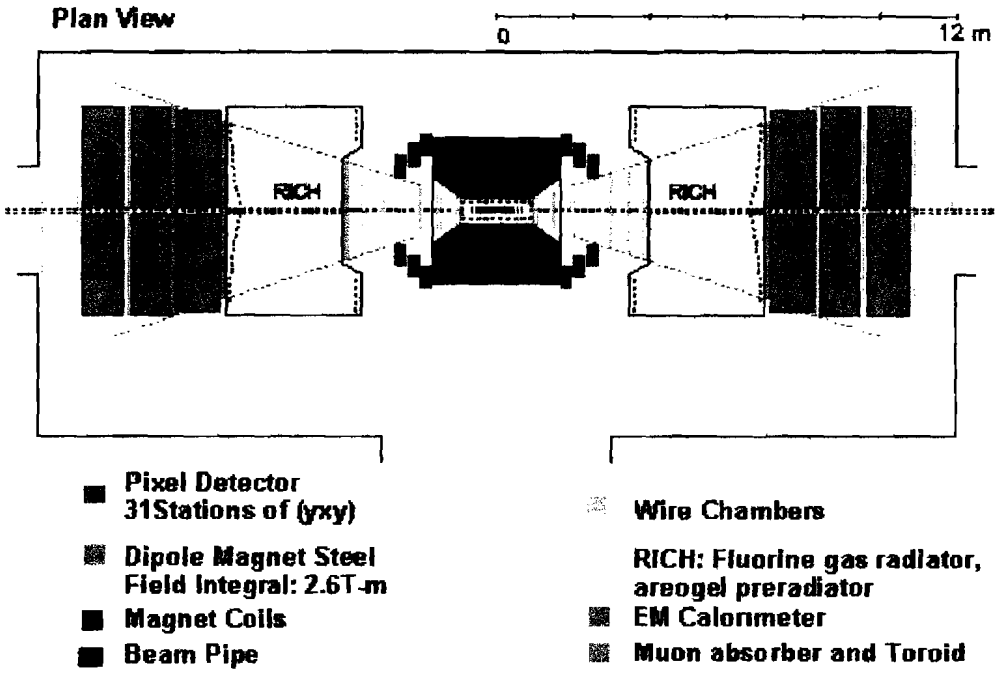


Figure 3: Sketch of the future BTeV spectrometer.

The vertex pixel detector provides the high resolution space points near the interaction, which are used both online and offline to reconstruct tracks and associate them with their parent vertices. In designing the vertex tracking system, one must consider the long interaction region (IR) of the Tevatron, which has a $\sigma_z = 30cm$.

1.2 PIXEL PLANE

The BTeV experiment is going to use silicon pixel detectors because they provide high precision space points and they are quite radiation hard. Radiation hardness allows the detector elements to be placed very close to the beam approximately 6mm (in vacuum, separated from the beam only by a thin RF shield). The measurements of space points, with very little noise, provides superior pattern recognition, allowing the experiment to reconstruct tracks and vertices in real time, and trigger on events containing reconstructable heavy flavor decays. The long interaction region forces one to have a rather long vertex detector with many pixel planes.

The pixel detector will contain about thirty million rectangular pixels, each 50 μm X 400 μm. Each pixel sensor is read out by a dedicated electronics cell. The pixel sensors are arranged in a bidimensional array, and the readout electronic cells are contained in a bidimensional array of unit cells of the same size as the sensors in the Fermilab pixel chip (FPIX). The two are connected by bump-bonds. The basic building block of the pixel vertex detector is a hybrid assembly called the Multi-Chip Module (MCM) which consist of a linear array of FPIX readout chips connected in daisy chain, bump-bonded to a sensor and wire bonded to a flexible printed circuit which carries I/O signals and power.

The baseline vertex detector will consist of a regular array of 31 "stations" of "planar" pixel detectors distributed along the interaction region, see Figure 1.4.

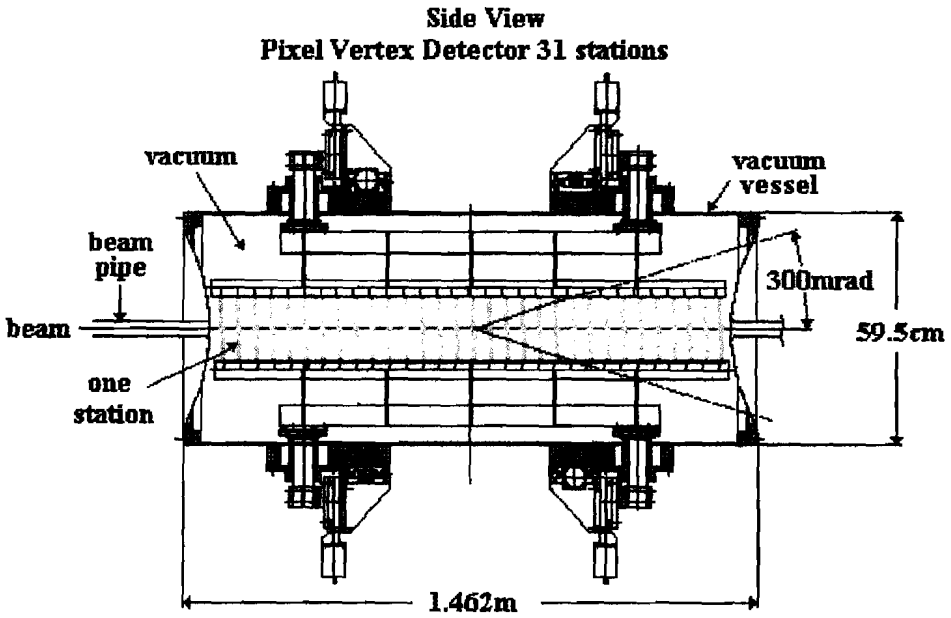


Figure 1.4: Side view of the future Pixel Vertex Detector, including the vacuum vessel and its mechanical support.

Each future station will contain one plane with the narrow pixel dimension vertical, and one with the narrow dimension horizontal, and is composed of two 5 cm X 10 cm halves. The half stations are mounted above and below the beam and are arranged so that a small square hole is left for the beams to pass through [2]. Two diagrams showing how the MCMs will be arranged on the pixel planes are shown in Fig. 1.5 and Fig. 1.6.

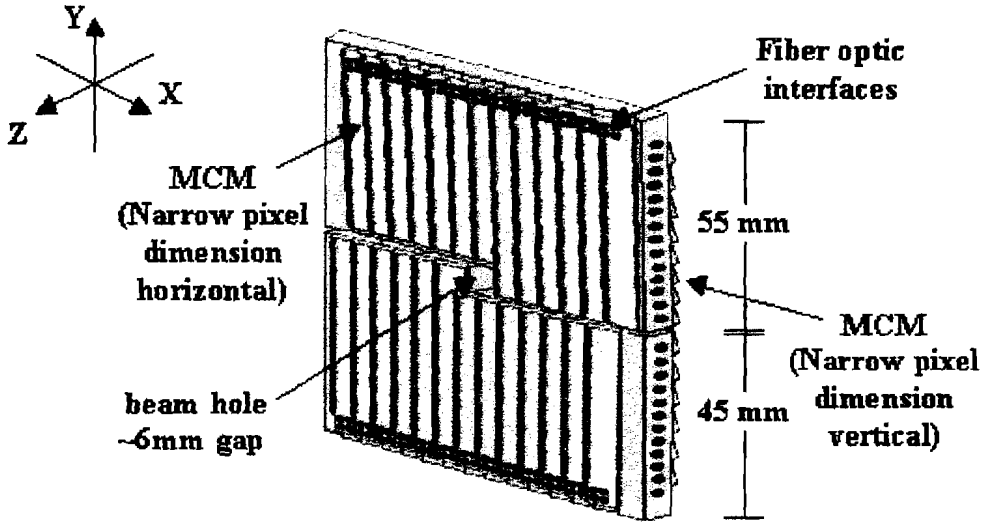


Figure 5: Future station composed by two 5 cm X 10 cm half stations. The station contains two planes, one plane with the narrow pixel dimension vertical, and the other with the narrow pixel dimension horizontal.

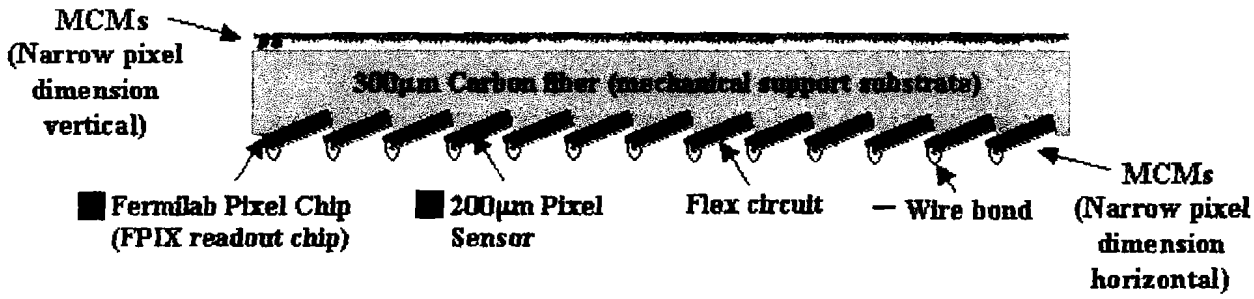


Figure 1.6: Future station top view.

The elementary charged particles generated by proton-antiproton collision will be bent by the magnetic field generated by means of the SM3 magnet. They will produce tracks in the pixel vertex detector. The curvature of a track will allow its momentum to be measured. Track will be used to from vertices. The RICH detector is made to use the *Čerenkov* effect to know the mass of these elementary particles. With this information many of the B particle decays will be able to be fully reconstructed.

1.3 REQUIREMENTS FOR THE BTeV VERTEX DETRECTOR

The BTeV experiment has very stringent requirements for the Pixel Vertex detector. The first constraint is that all the 31 stations must tolerate a high radiation dose. This dose has been calculated by people in the experiment. It depends on the radial distance from the beam, see Fig.1.7.

We can see from figure 1.7 that the FPIX chips near from the beam will receive about 1Mrad during one year of running and the electronics sitting on the perimeter of the pixels planes will receive about 100kRad.

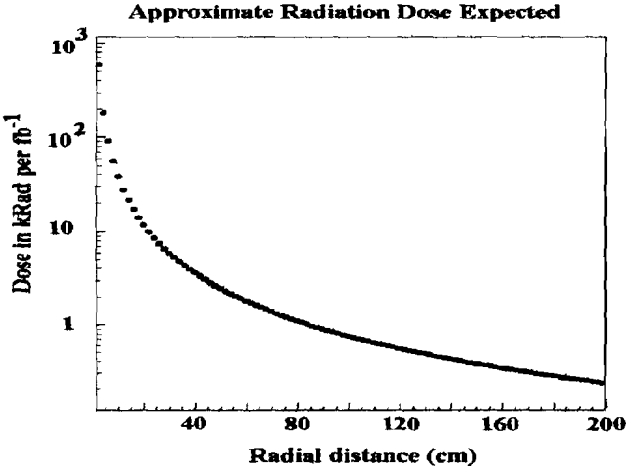


Figure1.7: Expected radiation dose at the Tevatron for one year running [3].

Another constraint associated with the pixel planes is their mass, they should be as light as possible to decrease particle interactions with the material. Strong interactions are the source of unwanted secondary particles. Electromagetic interactions and strong interactions causes

scattering, which increases the error in the track's reconstruction [4]. The mean scattering angle (in milliradians) is approximately given by Eq. 1.1.

$$\Theta = \frac{13.6}{P}\sqrt{\frac{X}{X_0}}, \tag{1.1}$$

where P is the momentum of the particle in GeV/c, X_0 is the radiation length and X is the thickness of the material in question.

Table 1 gives the radiation length of different materials.

Material	X_0 (mm)
Water	360
Beryllium	350
Carbon fiber	250
Beryllium oxide	143
Diamond CVD	123
Silicon	100
Aluminum	89
Nickel	15.7
Copper	14
Silver	8.7
Gold	3.4

Table 1: Radiation length of materials [4].

From Table 1 we can see that a particle traversing an aluminum sheet will have an approximately 2.5 times smaller mean scattering angle than in a copper sheet of same thickness.

The experiment requires that the FPIX chips read out all detected hits. Since the pixel detector is used for the main first level trigger and the number of hits is quite large, it is necessary to provide significant readout bandwidth to transfer all data from the pixel chips to the trigger electronics. Simulations performed mapped the number of group hits per chip in a half plane [3], considering a pixel chip with active area of 8 mm X 8 mm ($64mm^2$), each chip with 3200 pixels cells of 50×400 μm. The pixel chip uses 17 bits to deliver all the information about the position of a hit (row and column), time stamp, ADC and chip ID.

To achieve the high bandwidth required for the readout, ultra-high-speed digital optical links of 1-2 Gbps will be employed on each detector half-plane. A second type of digital optical link, which will send the command and control signals to the FPIX chips from the counting room to each detector half-plane, can operate at lower speed (~100 Mbps).

Both the emitters of the ultra-high-speed optical links and the receivers of the lower speed optical links will be about 7 cm from the beam if they are mounted on the detector planes as proposed in the baseline design. In this case, these devices will receive about 100 Krad of

radiation per year of running and will suffer some radiation damage. A further, rather stringent requirement is that they must operate inside the beam pipe in vacuum at $\approx -5^{o}C$ [1]. Other solutions, which place the optical circuits outside of the vacuum vessel about 25 cm from the detector, are also being studied.

To satisfy all the BTeV requirements, the Pixel Vertex Detector will have all the properties shown in Table 2.

Property	Value
Pixel size	rectangular: 50 μm × 400 μm
Plane Dimensions	10 cm × 10 cm
Central Square Hole Dimensions (adjustable)	nominal setting: 12 mm × 12 mm
Total Planes	62
Total Stations	31
Pixel Orientations (per station)	one with narrow pixel dimension vertical & the other with narrow dimension horizontal
Separation of Stations	4.25 cm
x-Plane to y-Plane Separation (within station)	5.0 mm
Total Station Depth (incl cooling, supports)	6.5 mm
Sensor Thickness	250 μm
Readout Chip Thickness	200 μm
Total Station Radiation Length (incl RF shielding)	2%
Total Pixels	3×10^{7}
Total Silicon Area	$\approx$0.6 m^{2}
Readout	analog readout (3 bits)
Trigger	signals are used in Level 1 trigger
Rate Requirements	time between beam crossings is 132 ns.
Noise Requirement	desired: $< 10^{-6}$ per channel/crossing required: $< 10^{-5}$ per channel/crossing
Resolution	better than 9 μm
Radiation Tolerance	$> 6 \times 10^{14}$ particles/cm^{2}
Power per Pixel	<60 μWatt
Operating Temperature	~-5 °C

Table 2: Properties of the Pixel Vertex Detector.

1.4 CONCLUSIONS

Until now we have a idea of the experiment and its requirements, now in the following chapter we will describe one of the principal integrated circuit which is one of the important parts of the pixel vertex detector, the FPIX1 chip. We will study the chip itself, describing the way to download and acquire information from it. Also we will describe the Multi-chip Module (MCM) that contains five FPIX1 chips connected in daisy chain that represent the first step to have a complete detector.

CHAPTER TWO

THE FPIX1 CHIP AND THE MULTI CHIP MODULE (MCM)

2.1 THE FPIX1 CHIP

2.1.1 Fermilab Pixel Chip 1 (FPIX1)

FPIX1 is the second in the sequence of submissions aimed at developing a chip that meets the BTeV requirements. It is a VLSI chip with pixel unit cells of size $50\,\mu m \times 400\,\mu m$ arranged in an array of 160 rows by 18 columns for a total of 2880 pixel cells. The readout and control architecture is column based. The block diagram of the FPIX1 chip and a photograph of the chip are shown in Fig.2.1 [5]. The FPIX1 chip can be divided into three mutually dependent parts, the pixel cell, the End of Column (EOC) Logic and the Chip Control Logic.

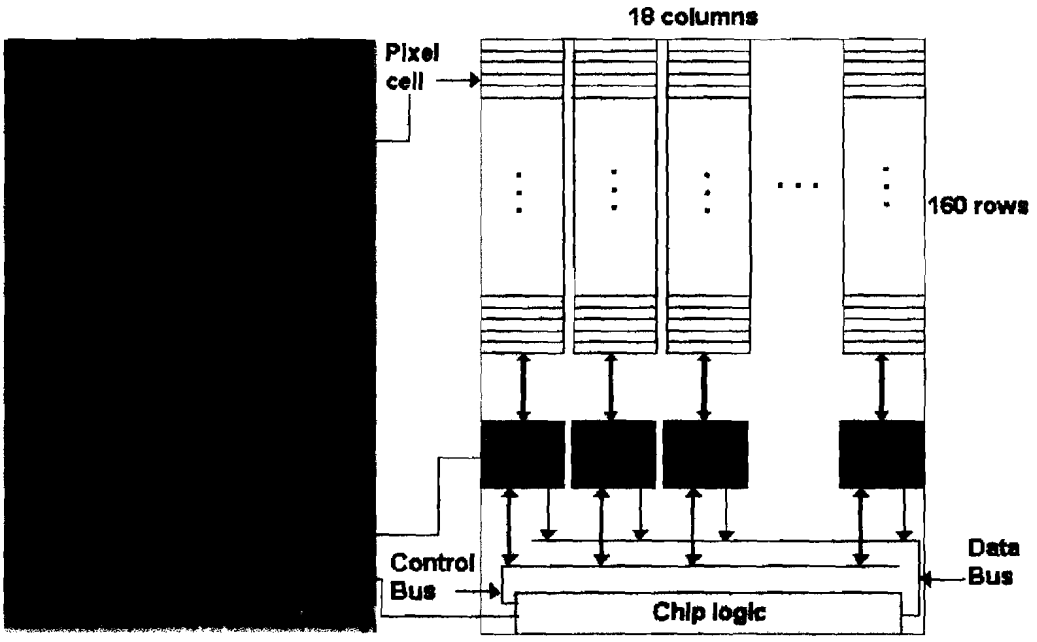


Figure 2.1. FPIX1 Block Diagram

To better understand the front-end of the pixel cell we are going to analyze a basic front-end for a pixel sensor. A basic front-end contains a black box circuit, of which the purpose is to eliminate the effect of the leakage current on the integrator amplifier. This integrator amplifier is used to integrate the current generated when a charged particle crosses the pixel sensor. Then the signal output of the integrator amplifier will represent the charge produced by the particle crossing. The equations that result from an analysis of the basic front-end circuit are:

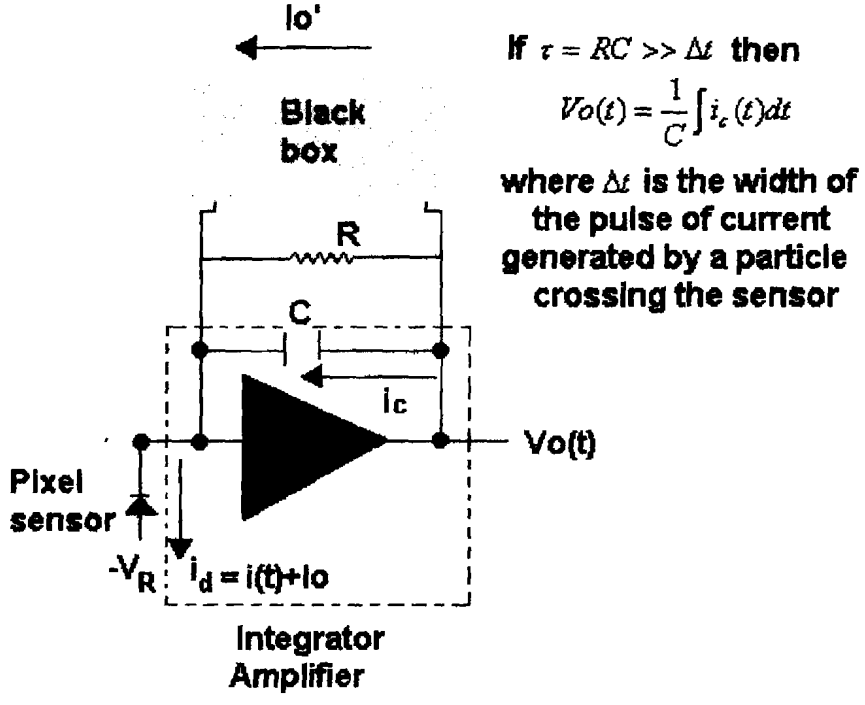


Figure 2.2 Basic front-end circuit.

Performing node analysis we obtain:

$$-[i(t)+Io]+Io'+i_c+i_R=0 \tag{2.1}$$

where i(t) is the current generated by the particle crossing, Io is the leakage current and Io' is the black box current. Then this equation can be divided in two cases the dc and the ac cases, for the dc case we have:

$$-Io+Io'=0\ ;\ Io'=Io \tag{2.2}$$

In consequence the current that flows by the black box must be equal to the leakage current denominated by *Io*. This condition is necessary and means that the leakage current doesn't flow thought the integrator amplifier. For the ac case we have:

$$-i(t)+i_c+i_R=0; i_c=i(t)-i_R, \text{ or, } i(t)=i_c+i_R \tag{2.3}$$

and

$$i_c=C\frac{dVo(t)}{dt}; i_R=\frac{Vo(t)}{R} \tag{2.4}$$

where *Vo(t)* is the voltage output of the integrator amplifier. This voltage will be a pulse generated by the particle crossing. Taking the integral on both sides of the equation we have:

$$i(t)=\frac{dVo(t)}{dt}+\frac{Vo(t)}{RC} \tag{2.5}$$

Solving the differential equation we can obtain the natural response, setting the current *i(t)* equal to zero:

$$Vo_n(t)=Vo\exp\left(-\frac{t}{RC}\right) \tag{2.6}$$

Then we can assume that the action of the resistor is to discharge the capacitor after the current generated by the particle crossing the sensor is integrated. If we don't put the resistance in the circuit then the capacitor always will be charged, i.e., if other particle crosses the sensor then the amplifier doesn't integrate the charge from zero. Instead the amplifier will begin to integrate the charge starting from the voltage level that has the capacitor, which is the level of the maximum voltage output of the amplifier produced by the first particle. With the incorporation of a resistance in the circuit we are avoiding this problem. For better understanding we show in figure 2.3 the shape of the integrator output without and with this resistance.

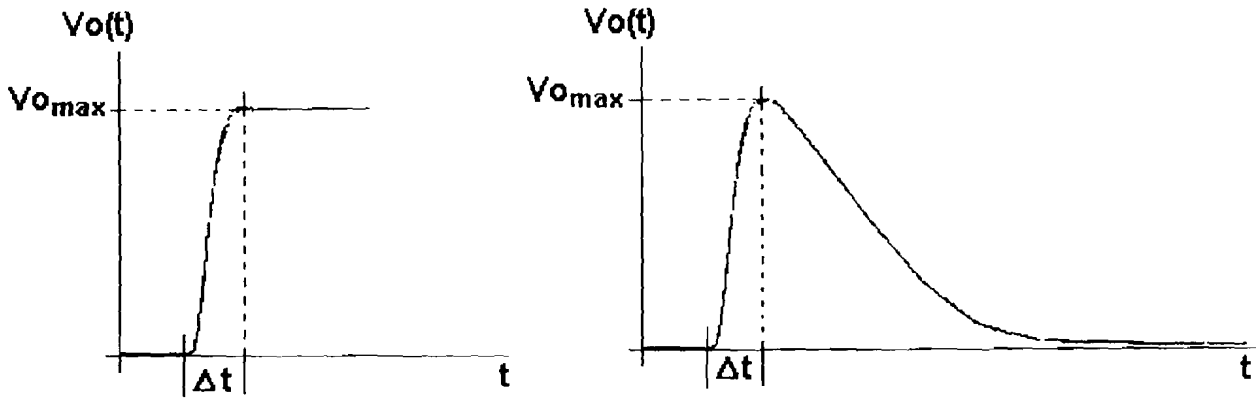


Figure 2.3 Shapes of the integrator output, a)without resistance and b) with resistance.

Following the analysis if $\tau = RC >> \Delta t$ then the voltage output of the integrator amplifier will be:

$$Vo(t) = \frac{1}{C}\int i_c(t)dt \tag{2.7}$$

and how

$$Q(t) = \int i_c(t)dt = \int i(t)dt \tag{2.8}$$

where $Q(t)$ represents the electrical charge as a function of time. This function will be a pulse generated by the particle crossing, resulting in the following relationship:

$$Vo(t) = \frac{1}{C}Q(t) \tag{2.9}$$

Then,

$$Vo_{max} = \frac{1}{C}Q_T \tag{2.10}$$

where Vo_{max} is the maximum voltage of the pulse produced by the corresponding particle and Q_T is the total charge generated by the particle.

With this result we can see that there exists a relationship between the voltage output of the integrator amplifier and the charge collected by the sensor when a charged particle crosses it. Then we can see how important it is to characterize each front-end of a complete FPIX1 chip to assure a uniform response from cell to cell.

As an example we shown in the figure 2.4 a real signal obtained characterizing a front-end of the first prototype for the readout chip called FPIX0 [6]. The FPIX0 and the FPIX1 have essentially the same front-end electronics.

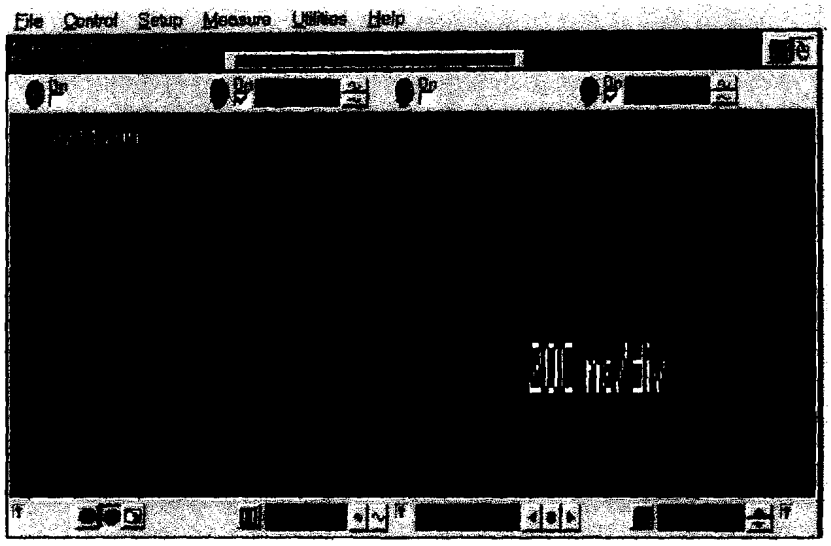


Figure 2.4 A real signal obtained characterizing the front-end of the first prototype for the readout chip called FPIX0.

The Pixel cells hold the front-end (Fig. 2.5) electronics and the digital interface to the EOC logic. The front-end contains a charge-sensitive amplifier and a second amplification stage; the output of the second stage connects to a flash ADC and a discriminator. The discriminator output is asserted when the signal at the input of the discriminator is higher than the threshold (Vth0). The pixel cell contains a digital interface with two major components, the command interpreter and the pixel token and bus controller.

The command interpreter has four inputs corresponding to the four EOC command sets. These commands are presented by the EOC logic simultaneously to all pixel cell interpreters in a column. When an interpreter is executing the input command and the hit output from the discriminator is asserted, the interpreter associates itself with the particular EOC set and simultaneously it alerts the EOC logic to the presence of a hit via the wire-or'ed HfastOR signal. After that, the information is stored in the cell until EOC set issues an output or reset command. When this command is an output command, the interpreter issues a bus request and asserts the wire-or'ed RfastOR signal. Then the balance of the readout proceeds synchronously with the ReadClk. The EOC logic provides a column token, the token quickly passes pixel cells with no information until it reaches a cell that is requesting the bus. The data is composed of the ADC count bits [3:1] and the row address radd [7:0]. As the hit pixel is read out it automatically resets itself and withdraws its assertion of the RfastOR. This signal returns to its inactivated state while the rest of the hit pixels are being read out [5]. The action of the synthetic resistor is that it acts like a resistor for small signals and like a constant current source, discharging the feedback capacitor, for large signals.

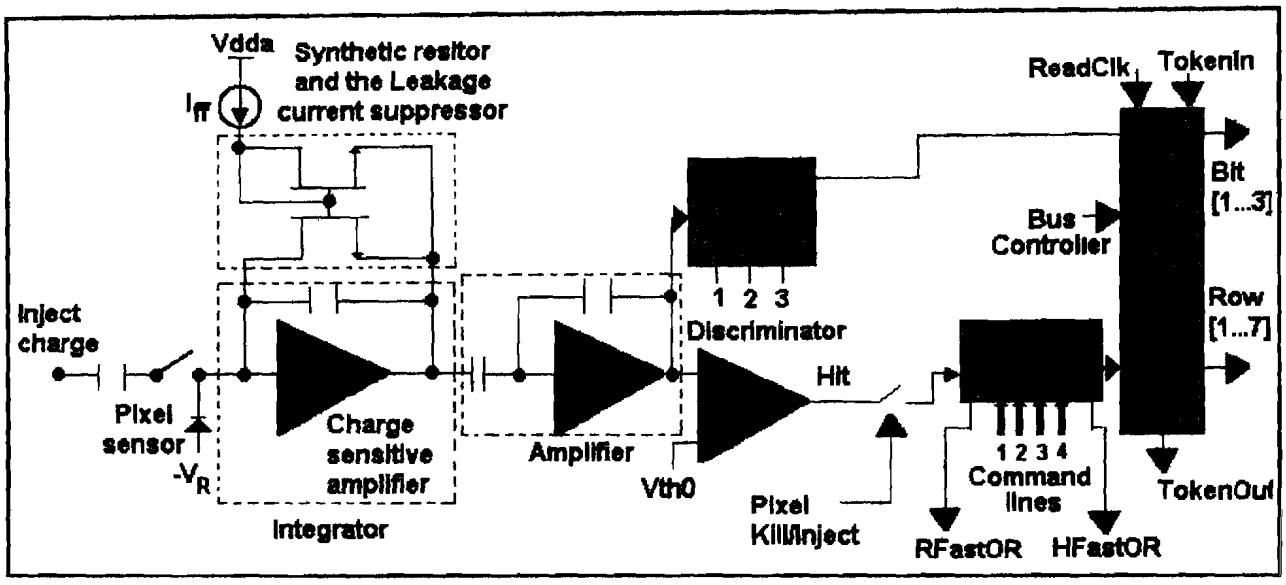


Figure 2.5. Front End pixel cell.

The end of column logic consists of a Priority Encoder and four EOC command set. The EOC command set consists of a timestamp register, a state machine to generate the EOC commands and two comparators. The priority encoder selects one EOC set to send a input command; when there is a hit in some pixel cell, the HFastOR signal is asserted and the state machine assign the EOC set. Then the EOC set responds by latching the current BCO (CBCO) in its EOC timestamp register, later on the EOC set send an idle command at the next rising edge of the BCO clock. With this feature all pixels in a particular column hit in the same clock period are associated with a single EOC set. The priority encoder assigns the next EOC set to send the input command to the column when the rising edge of the BCO clock is present.

We are not losing data in pixel cells in a column when there is a hit present in a column in up to four different crossings because the EOC logic has four EOC sets. When the EOC set is asserted it is waiting for matches with its stored timestamp BCO (SBCO). If this match is between the requested BCO (RBCO) and the SBCO, the EOC set broadcasts the output command and if the match is between CBCO and SBCO, it broadcasts the reset command. This defines the reset delay. A second machine is implemented inside the column token and bus controller. This second state machine is used to control the access to the EOC data bus; this access is arbitrated by an EOC token. When there is a match between the RBCO and the latched CBCO, the column controller issues the Ctkin token to the column and waits for the EOC Token In (ETkin) from the chip logic. When the ETkin is asserted, the column controller enables the pixel data onto the internal data bus and keeps it asserted until all the hits in the pixel cells in the column are reading out. The column controller now passes the EOC token to the next EOC logic by asserting ETkout. With the assertion of Etkout as soon as Ctkout is received by the column controller, we perform a full clock speed readout of the pixel data, even when the chip finishes the read out of one column and starts to read out the next. A scheme of the EOC logic is shown as Fig. 2.6.

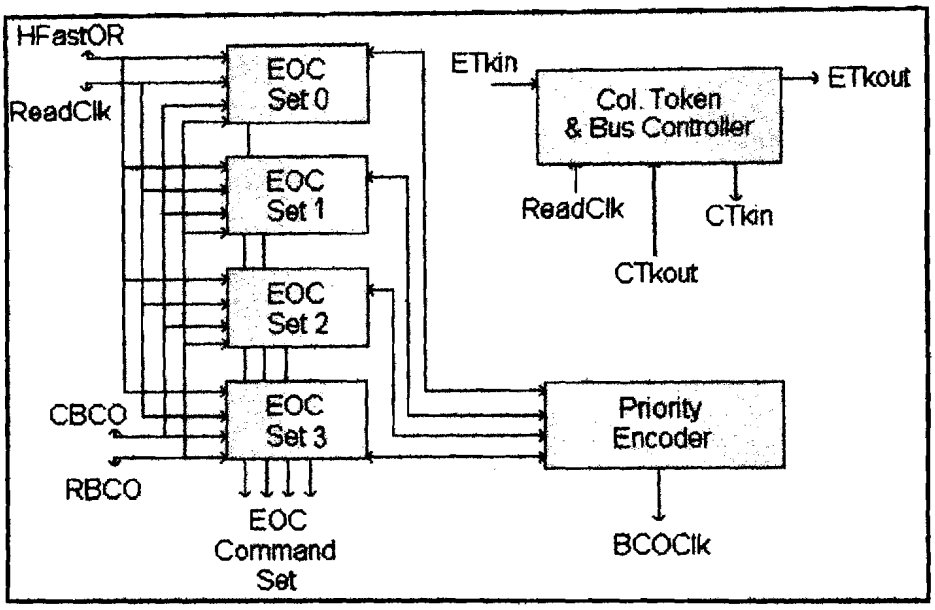


Figure 2.6. End of Column Logic.

The chip control logic consists of the current BCO counter (CBCO), the readout BCO counter, a multiplexer and a chip controller, and controls the features associated with the whole chip. The CBCO increments synchronously with the BCOClk and then is delivered to the EOC logic. The multiplexer multiplexes the readout BCO counter or the external Request BCO (in case the external trigger is used); the output of the multiplexer forms the RBCO number that is delivered to the EOC logic. When the chip is operated in continuous readout mode, the multiplexer connects the Readout BCO counter to the RBCO, and this counter will provide the timestamp number that should be used to compare with the timestamp latched in the Timestamp registers. Other functions of the chip logic include external bus arbitration, which is done by a token passing from chip to chip. The configuration is programmed using a serial bit stream to set features like pixel cell kill (to disable pixels) and pulse inject select (for enabling programmable pixel cells to accept charge inject directly into the front-end using an external voltage source).

2.1.2 Pad description and physical dimensions

The FPIX1 chip contains 87 pads. We show all the pads in the following tables [7].

Twelve pads are providing ground:

Function	Pad number	Pad name	Description
Ground (0V)	1,87	**Vssa**	Pixel analog circuitry
	9,79	**Vssd**	Pixel commands interpreters and drivers
	13,19,25	**Vsub**	Substrate voltage
	15,21	**Vssd-eoc**	End of column Logic
	51	**Vss-pad**	Pad circuitry
	5,81	**Vss-comp**	Pixel A to D conversion

Nine pads supplies power:

Function	Pad number	Pad name	Description
Power (3.3V)	3,85	**Vdda**	Pixel analog circuitry
	11,77	**Vddd**	Pixel commands interpreters and drivers
	17,23	**Vddd-eoc**	End of column Logic
	53	**Vddd-pad**	Pad circuitry
	7,83	**Vddd-comp**	Pixel A to D conversion

One pad is the input for the charge injection:

Function	Pad number	Pad name	Description
Inject Charge	2	**InjectIn**	Analog input for test charge

Six pads give the current bias:

Function	Pad number	Pad name	Description
Current bias	4,10	**Ivbbnl**	Decoupling point and override for internal bias
	6	**Ivfb**	Feedback current (5nA)
	8	**Ivbbp**	Master bias current (7 μ A)
	12	**Ivbbpl**	Decoupling point and override for internal bias
	14	**Ivbb-comp**	Decoupling point and override for internal bias

Three pads are the threshold voltages for the ADC and one pad represents the threshold voltage of the discriminator:

Function	Pad number	Pad name	Description
Thresholds	16	**Vth3**	Most significant threshold for ADC
	18	**Vth2**	Threshold for ADC
	20	**Vth1**	Threshold for ADC
	22	**Vth0**	Threshold for hit comparator

Two pads are the bias voltage for the two different sensors i.e., p-stop and n-guard.

Function	Pad number	Pad name	Description
Bias voltage	24	**Vpstop**	Bias voltage for the detector p-stop
	26	**Vnguard**	Bias voltage for the detector n-guard

Fifteen control signals, which we can program, control and readout the FPIX1 chip:

Function	Pad number	Pad name	Description
Control signals	27/28	**TokenIn/TokenIn***	Input for the chip token
	29	**TrigAcc**	Trigger mode: Trigger signal; Continous mode: Accept (3.3V) or reject (0V) hits.
	30	**ShiftIn**	Input for the scan paths
	31	**DataRes**	Reset signal
	32	**ProgramRes**	Reset signal
	33	**ShiftClk**	Shift clock signal
	34/35	**ReadClk*/ReadClk**	The read out clock
	36/37	**BCOClk*/BCOClk**	The beam cross over clock
	50	**LoadKill**	Load kill pattern into pixels on falling edge
	76,78,80	**Reg2, Reg1, Reg0**	Scan path selection bits (Reg2 is MSB)

Thirty-two output signals

Function	Pad number	Pad name	Description
Output signals	52	**ChipHasData**	Diagnostic path, it uses CMOS levels
	82	**ShiftOut**	Output form scan path
	38-49, 54-75	**Data outputs (17 differential signals)**	Data output signals
	84/86	**TokenOut*/TokenOut**	Chip token output

The size of the FPIX1 chip is 7.5mm by 12.5mm [4]. Pads 1 and 87 are 17.3 microns in from the outer edge of FPIX1. The bottom row of pads, which are the odd number pads, are 39.2 microns from the bottom of FPIX1. The FPIX1 pads are on a 169.2 micron pitch with 79.2 microns between pads, the pads themselves are 160 microns by 90 microns. A sketch is shown in fig. 9. A block diagram of the FPIX1 chip is shown as a Fig.2.7.

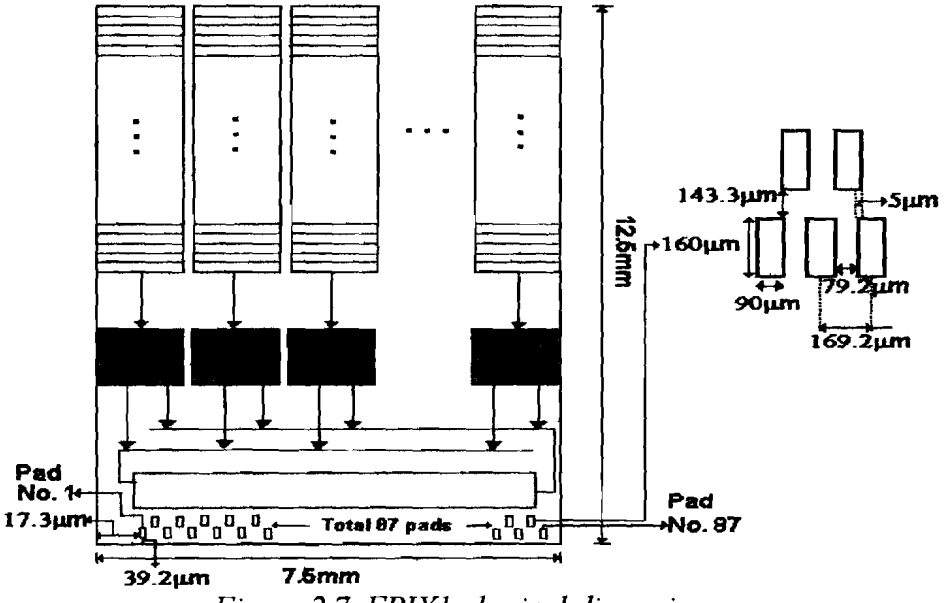


Figure 2.7. FPIX1 physical dimensions.

The most important signals are the control signals. With these signals we can download information to the chip. Most of these control signals are CMOS i.e., 3.3V represents a logical 1 and 0V represents a logical 0. The first control signals are **Reg [2:0]** that represent the scan path selectors for the **ShiftIn** signal; they are CMOS inputs. **ShiftIn** is a CMOS input for the scan paths. At the rising edge of the **ShiftClk**, the value at the **ShiftIn** will be scanned into the path chosen by the **Reg0**, **Reg1**, and **Reg2** signals. **ShiftClk** is the clock signal; at the rising edge the contents of the **ShiftIn** signal are scanned into whatever scan path is selected by **Reg0**, **Reg1**, and **Reg2** signals. **DataRes** is a reset level signal. When this signal is high, the BCO counters are reset to zero; the End of column registers are reset to empty and the chip stops outputting data. **ProgramRes** is a reset level signal. When this signal is high, the mask registers are set to zero; the chip is reset to continuous mode; the BCO lag is set to 2 and the ChipID is set to zero. **LoadKill** is the signal which latches the kill pattern that has been scanned into the pixel array on its falling edge. It is a CMOS level signal and it is kept at logical 1 during the kill scan, and dropped to a logical 0 to latch the kill pattern. **TrigAcc** is a dual-mode signal. If the chip is operating in triggered mode, it is the trigger signal. If the chip is operating in continuous mode, this pad is the Accept/Reject signal. In continuous mode, when this signal is logical 0 the chip rejects (ignores) incoming data, and when is logical 1 the chip accepts incoming data. This is a single ended CMOS signal.

To get the information acquired by the chip, we need to send the set of signals that control the readout of the chip information and then process it. Most of these signals are low voltage differential signals, or LVDS, (Vhi≅ 1.75V; Vlo≅ 1.55V). **ReadClk*/ReadClk** is a free-running LVDS level differential clock. The simulated frequency was 26MHz, but the real frequency will be whatever the chip can handle. **BCOClk*/BCOClk** is an LVDS representation of the beam crossover signal. **TokenIn/TokenIn*** it is an LVDS input. When the signal is high (Pad 27=1.75V; Pad 28=1.55V), the chip can take the bus. When this signal is low (Pad 27=1.55V; Pad 28=1.75V), the chip cannot take the bus and its data outputs are tri-stated.

The **ShiftOut** represents the scan output for each of the scan paths selected by **Reg0**, **Reg1** and **Reg2**. This signal is a CMOS level. This signal will be driven to the **ShiftIn** of the next chip when you connect several chips in daisy chain. **TokenOut*/TokenOut** represents the output of the chip token signal.

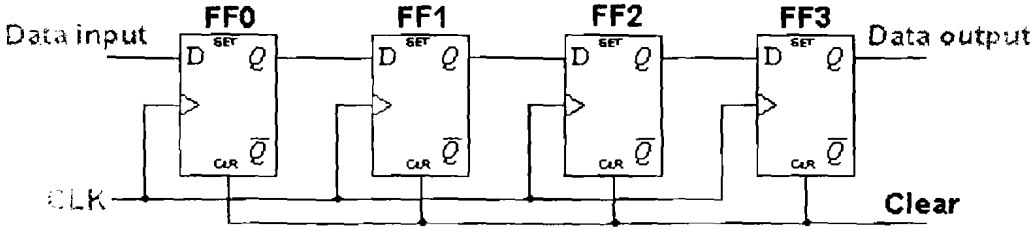
The FPIX1 chip delivers its information in the following data output format [7]. The data outputs are tri-statable LVDS outputs. Data 16 is the data valid signal; with this signal you can know if the chip has data to send. When the chip has data, the Data valid signal will be a logical 1 (3.3V) if there are no data this signal will be a logical 0 (0V). When there are data available, it will come in two forms i.e., control and data words; by means of Data 15 we can know whether the word presented is a control or a data word. Data 15 is a logical 0 in a control word and a logical 1 in a data word. The control word contains the ChipID of the chip and the BCO number (time stamp). The data word contains the row address, column address and magnitude (ADC value) of a particular hit pixel. In the table 2 we show all data and its corresponding meaning.

Pad name	Pads number	Data Available Control Word	Data Available Data Word
Data 16/Data 16*	74/75	Logical1	Logical1
Data 15/Data15*	73/72	Logical 0	Logical 1
Data 14/Data 14*	71/70	Logical 0	ADC [1]
Data 13/Data 13*	69/68	Logical 0	ADC [0]
Data 12/Data 12*	67/66	Logical 0	Col Addr [4]
Data 11/Data 11*	65/64	ChipID [0]	Col Addr [3]
Data 10/Data 10*	63/62	ChipID [0]	Col Addr [2]
Data 9/Data 9*	61/60	ChipID [0]	Col Addr [1]
Data 8/Data 8*	59/58	ChipID [0]	Col Addr [0]
Data 7/Data 7*	57/56	Logical 0	Row Addr [7]
Data 6/Data 6*	55/54	Logical 0	Row Addr [6]
Data 5/Data 5*	49/48	BCO [5]	Row Addr [5]
Data 4/Data 4*	47/46	BCO [4]	Row Addr [4]
Data 3/Data 3*	45/44	BCO [3]	Row Addr [3]
Data 2/Data 2*	43/42	BCO [2]	Row Addr [2]
Data 1/Data 1*	41/40	BCO [1]	Row Addr [1]
Data 0/Data 0*	39/38	BCO [0]	Row Addr [0]

Table 2. Data outputs from the FPIX1 chip.

2.1.3 Scan paths and Readout

There are eight different control sequences by which we can download information to the FPIX1 chip. These are called scan paths and are formed by shift registers with different depths in the FPIX1. Shift registers are a type of sequential logic circuit, mainly used for the storage of digital data. They are a group of flip-flops connected in a chain so that the output from one flip-flop becomes the input of the next flip-flop. Most registers used in this way possess no characteristic internal sequence of states. All the flip-flops are driven by a common clock and all are set or reset simultaneously. A basic four-bit shift register can be constructed using four D flip-flops, as shown in Fig.2.8. The operation of the circuit is as follows. The register is first cleared, forcing all four outputs to zero. The input data is then applied sequentially to the D input of the first flip-flop on the left (FF0). During each clock pulse, one bit is transmitted from left to right. Assume a data word to be 1001. The least significant bit of the data has to be shifted through the register from FF0 to FF3.



FF0	FF1	FF2	FF3
1	0	0	1

Figure 2.8. A basic four-bit shift register

To choose a particular scan path [7] we select one of the eight different codes using the control signals named **Reg0**, **Reg1**, and **Reg2**. For example to configure the chip, you must select the code 101. The codes are given in Table 3. To load all the scan paths into the chip, we are use the same **ShiftIn** pad, the same **ShiftClk** pad and the same **ShiftOut** pad. There is necessary time a relationship between **ShiftIn** and **ShiftClk** signals. **ShiftIn** must be delayed by more than 5ns from **ShiftClk**.

	Reg2, Reg1, Reg0	Path
0	0 0 0	Idle – No path
1	0 0 1	Trigger path
2	0 1 0	Not used
3	0 1 1	Programmable reset delay
4	1 0 0	Not used
5	1 0 1	Configure (Mode, ChipID and Lag)
6	1 1 0	Kill or Inject
7	1 1 1	Kill or Inject

Table 3. Scan paths to program the FPIX1 chip.

A brief description the register codes for selection of each of the scan paths is as follows: The idle path should be the default path, if no action is taken **Reg2**, **Reg1** and **Reg0** should be set to 000. The paths selected by 010 and 100 instruct the chip to take no action as for the idle path. The trigger path code is 001 and you select this if you want to use the chip in external trigger mode. This path is six bits deep, and the most significant bit is scanned in first. When all six bits have been scanned in, a pulse on the **TrigAcc** pin (pad 29) will produce a new trigger request to the end of column logic cells.

The code to select programmable reset delay path is 011. Each EOC logic circuit has a mask register that is six bits deep. When all the bits in the mask register are equal to zero the EOC logic will reset itself in 64 BCO clock cycles. When all bits in the mask register are set to one, the column is effectively shut down. This scan path is 102 bits long because each column has a 6 bits mask register in its EOC logic circuit, this means 6x17=102 bits. Then, e.g., you send 102 bits of zeros to do a reset delay by means of the **ShiftIn** pad. The most significant bit of column 17 is scanned in first; the least significant bit of column 1 is scanned in last.

The register code for the configure path is 101. This path is used to program the mode of operation of the chip, the identification number (ChipID) and the BCO lag number. There are two modes of operation, triggered mode or continuous mode. If you enter a logical 1 (3.3V) into the pattern you select triggered mode, while if you enter a logical 0 (0V) you select a continuous mode. The ChipID number is the identification number of the chip. If you are working with one chip, you can give it any identification number, e.g., 0000. If you have more than one chip, as on the MCM, you need to identify the chip from which you are receiving information. The BCO lag is the minimum distance between the current BCO number and the requested BCO number in continuous mode. Typically, this number is set to two. The configure path contains 11 bits, 1 bit for the mode of operation, 4 bits for the ChipID number and 6 bits for the BCO lag. Then, by means of the **ShiftIn** pad you need to send 11 bits with the configuration that you want. The most significant bit

of the BCO lag is scanned in first, followed by the rest of the BCO lag. Then the most significant bit of the ChipID is scanned in. Finally, the mode of operation is scanned in.

The codes for the kill/inject paths are 110 or 111. These paths select the pixel cells that you want to kill (disable) and the pixel cells which will be enabled for charge injection. In the FPIX1 chip we have 160 rows by 17 columns, for a total of 2880 pixel cells. The number of data bits we need to program the kill/inject path is 5760. The first 2880 bits form the kill mask, while the second 2880 bits represents the inject mask. During the loading of the kill mask the **LoadKill** signal (pad 50) should be held high, and for the inject mask this should be held low. The scan path starts at the lowest number pixel of the column 17th and goes up to the highest pixel in that column. From there the scan path continues to the highest pixel of column 16th and goes down to the lowest number pixel in that column. In general for odd columns the scan path starts in the lowest number pixel and goes up to the highest number pixel, and for even column the scan path starts in the highest number pixel and goes down to the lowest number pixel. This scan scheme is shown in Fig. 2.9. A logical 1 (3.3V) in the kill pattern means that the pixel is killed; a logical 0 (0V) in the inject pattern means that the pixel will be injected.

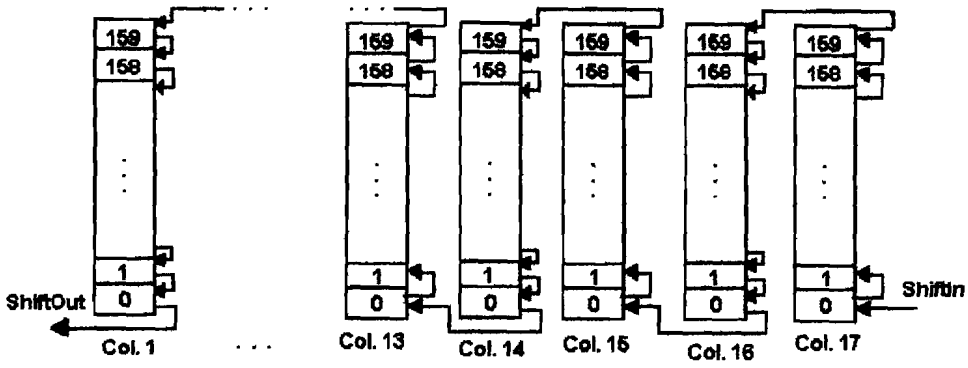


Figure 2.9. Kill/Inject Scan Path.

After you have selected a pixel cell to be enable (not kill) and to be injected with charge, you inject charge to the chip and then you must send a set of signals to the chip to read out the information that you have generated by means of the injection. These signals are: **DataRes**, **BCOClk**, **ReadClk**, **TokenIn** and **TrigAcc**. All these signals except **DataRes** are LVDS levels. You can work with these signals as CMOS levels by supplying voltage to the terminal Vdd/2, which allow us to work in single-ended mode rather than differential. Following injection, if there is data on the bus, the Data Valid bit (Data 16) will be 1. The data will change at the falling edge of the **ReadClk**, and it will be stable at the next rising edge of the **ReadClk**. Therefore, the data lines [0:15] should be latched by the data acquisition system at the rising edge of the **ReadClk** when Data Valid is logical level 1. You need to send a **DataRes** to the chip signal before injecting it in order to clear stale information. Because sending **DataRes** resets the BCO counters, the End of column registers are reset to empty and the chip stop outputting data.

As all the cells on the FPIX1 chip use the same data bus, we control the bus access using the **TokenIn** signal. If you have, for example, N FPIX1 chips connected in daisy chain, the chip with the token can access the bus and transmit its own data. Then the **TokenOut** signal of chip N-1 is connected to the **TokenIn** signal of chip N. If the chip is to be used in single chip mode, its

TokenIn can be set to a logical 1. Finally, the chip will accept hits if the **TrigAcc** signal is logical level 1, and will reject hits if this signal is logical level 0.

2.1.4 Design of patterns to control the FPIX1 chip

Now that we are familiar with the FPIX1 chip and its functions, we need to design the patterns we will use to download the information to control the chip and to read out the data generated by charge injection. The first set of information that we download to the chip is in an array of commands called initialization. This array contains all the commands to perform a reset and to set up the configuration. To do a reset we need to send the scan path code 011 (**Reg2**=0, **Reg1**=1, **Reg0**=1) and 102 bits of zero logic by means of the **ShiftIn** path. We also send the clock signal by the **ShiftClk** pad. This completes the reset. After a short delay we send the commands to configure the chip. For this, we select the scan path 101 (**Reg2**=1, **Reg1**=0, **Reg0**=1). In this mask we send the lag number equal to two, in the 6 BCO bits of the mask, the chip ID number in the 4 ChipID bits, and, the mode of operation in the 1 bit mode. This means that the mask is 11 bits long. A sketch of the FPIX1 chip initialization as seen in the screen of the data generator is shown in fig.2.10.

The next step is to send the kill/inject mask; selected by means of the path code 111 (**Reg2**=1, **Reg1**=1, **Reg0**=1) or 110. This mask activates the pixel cell or cells that we want to characterize. Sending a logic zero to a specific cell in the kill mask means it is enabled to receive charge on it, while sending a logic one means that it is disabled. In the inject mask a logic one represent that the pixel cell will not be injected and a logic zero means that it will. The kill mask is 2880 bits long, the inject mask is 2880 bits long. These bits correspond to the 2880 pixel cells that the FPIX1 contains. The shift-in is delayed by one pulse of clock from the shift-clock. The frequency of the shift-clock is 12.5MHz. The loadKill needs to be enabled during the kill mask and to be disabled for the inject mask. A sketch of the kill/inject mask as seen in the screen of the data generator is shown as Fig.2.11.

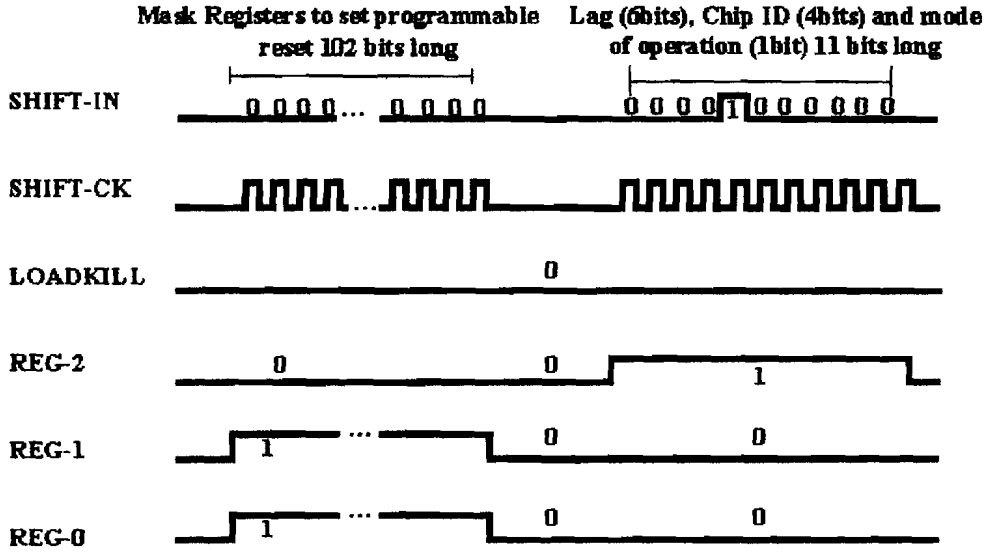


Figure 2.10. Initialization mask

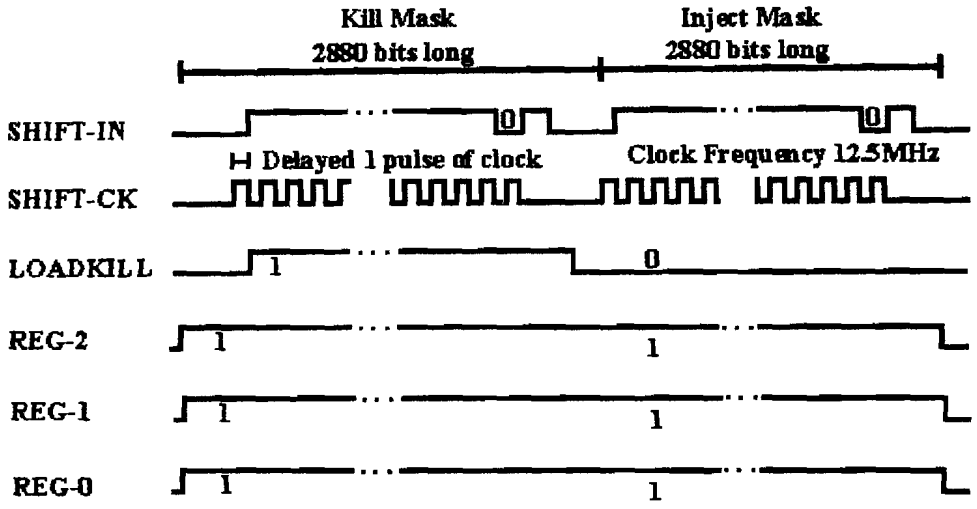


Figure 2.11. Kill/Inject mask.

Following initialization send a set of signals that inject charge and then performs the readout of the information that the chip delivers in response to the charge injection. These signals are the charge inject (**InjectIn**), the data reset (**DataRes**), the BCO clock (**BCOClk**), the readout clock (**ReadClk**), the token in (**TokenIn**) and the trigger accept (**TrigAcc**). All of these are included in a single mask in the correct time sequence. Before injecting charge you need to send a **DataRes** signal in order to clear stale information, e.g., from previous charge injection. **DataRes** causes the BCO counters to be reset, the End of column registers to be reset to empty and the chip to stop outputting data. We send the charge inject signal $10\mu s$ after send the DataRes, by means of the pin marked **InjecIn**. At the same time we send two signals, Clockpod1 and Clockpod2, with which we synchronize the logic analyzer to accept all the data that the chip delivers. A sketch of the readout is shown in Fig. 2.12.

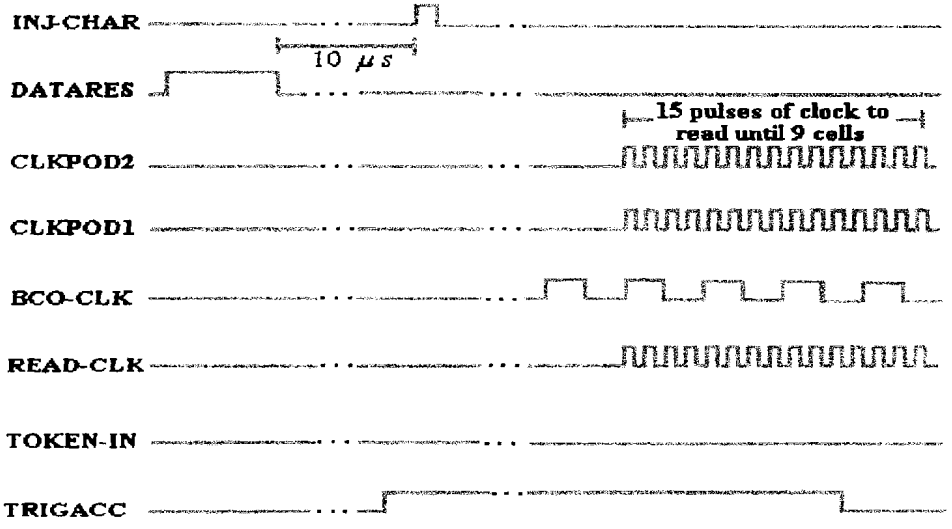


Figure 2.12. Readout mask.

2.2 THE MULTI CHIP MODULE (MCM)

The innermost detector for the BTeV experiment will be a pixel detector. This detector will be composed by 62 pixel planes, and each plane contains several pixel chips bump bonded to the sensors. A very important constraint associated with these planes is their mass. They should be as light as possible to decrease particle interactions with the material. Particle interactions cause scattering and thus increase the error in the reconstruction of the trajectories of the particles. Other important constraints related to the pixel detector are that they must work in a high radiation environment and in vacuum. The detector will suffer a substantial radiation dose, which means that it has to be built with material and glues that are radiation hard. Also they cannot outgas or evaporate in vacuum. Since the detector is inside vacuum, most of the heat has to be conducted from the chips to liquid cooling channels placed in the support material. In this case carbon fiber structures are good candidates because they provide lower mass associated with good thermal conductivity and are none the less structurally sound. For all these reasons the experiment decided to glue high density interconnect (HDI) circuits on the top of a carbon fiber plate that holds the detector and associated electronics. The cooling channels are embedded in the plate and are also made from carbon that has been fused into glassy tubes. The readout and control chips will be wire bonded to the HDI.

2.2.1 High Density Interconnect Circuits (HDI)

In the new world of electronics the HDI has gained a very high importance. The trend toward miniaturization and toward higher and higher speed devices has increased the demand for smaller and smaller semiconductor components with an even increasing density of input-output signals as well as a demand for a higher number of connection points on the printed circuit boards (PCB) on which the chips are mounted. Multi-layer flex circuits will provide the interconnection densities needed to meet the demands for the BTeV experiment.

The flex circuit contains four layers, two layers for signal interconnects, one layer for power and other signals and one layer for the ground plane [4]. There are two ways to proceed in order to produce the flex circuit. One is to use a four circuit layer flex circuit, and the other is to use two flex circuits, each with two layers and one assembled on the top of the other. Other approaches can also be considered, like one with two flex circuits with differing densities, one assembled on the top of the other. This idea is based on the fact that just portion of the circuit requires high density interconnect. The first circuit would be two layer flex circuit layout using "standard" design rules (with 100μm minimum traces and spaces widths, 400μm via cover pads and 200μm via through holes). The second flex, which would be built using more "aggressive" design rules (like 25μm traces widths and 68μm via cover pads), is used for the signal interconnects. In our application, this two rule flex circuit could be employed in the following way: the "standard" rule circuit would be used for the power and ground planes and some other low density signals, while the "aggressive" rule circuit would be used for the signal interconnects. This approach also requires that there not be many interconnections between the two different rule circuits, which is probably the case for BTeV. We interconnect efficiently from circuit to circuit, using wire bonds. Furthermore, this approach, using two different design rule circuits, lends itself to an important performance improvement, namely mass reduction. Also it may be possible to find some vendor willing to

manufacture "standard" rule flex circuit with aluminum conductors, since aluminum is a lower Z material than copper and therefore causes much less scattering.

2.2.2 Operation of the MCM

The Multi Chip Module (MCM) is the first attempt to characterize the performance of more than one chip bump bonded to a pixel detector. The MCM contains 5 FPIX1 chips connected in daisy chain. These are mounted on the top of a printed circuit board (PCB) designed for these test. A very high density circuit is required for signals interconnects in multiple FPIX1 chip designs. Limits on manufacturing prevent the use of conventional printed circuit boards (PCBs) to achieve the required high density. The FPIX1 chip bonding pads and signal traces on the prototype are on a multiple layer flex circuit manufactured by Fujitsu Computer Packaging Technologies Inc. Both the FPIX1's and the flex circuit are attached to the top of the PCB with the wire bonds going from the FPIX1 chips to the bonding pads on the flex cable. Figure 2.13 shows a sketch of the MCM assembled on the top of the PCB.

Several decoupling capacitors are mounted on the flex circuit. Power and ground are connected through heavy low resistance and inductance connections (i.e. posts interconnecting the flex circuit with the PCB). The power supply decoupling capacitors need to be located close to the pixel chips. Most of the signals are bussed in parallel to all chips with the exception of **ShiftIn**, **ShiftOut**, **TokenIn** and **TokenOut** that are daisy chained. Two bias current needs to be provided to the FPIX1 chip **Ivbbp** and **Ivfb**. Other bias currents are used for decoupling and optimization.

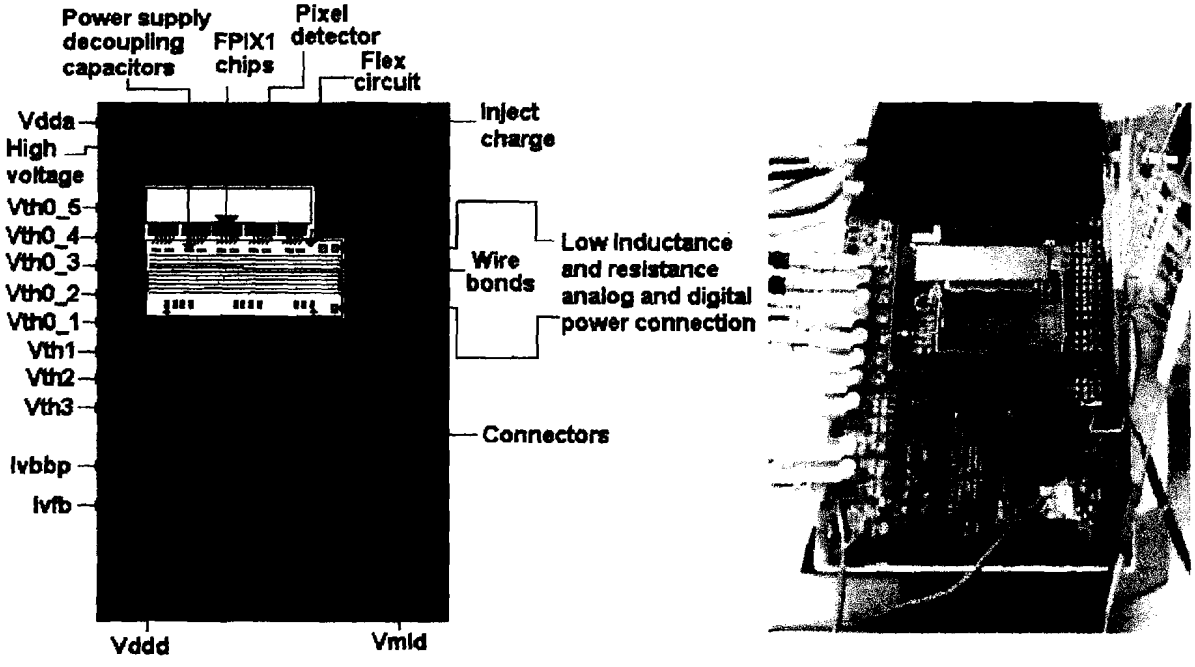


Figure 2.13. Sketch of the Fermilab Pixel Module.

All the bias currents are of the same type and are tied together (one trace feeding all FPIX1 chips for each bias current). They are accessible to the rigid board through the wire bonds. The rigid board of the MCM contains resistors to allow the setting of all bias currents. Each of the four

threshold voltages, **Vth0**, **Vth1**, **Vth2**, and **Vth3**, are also be tied together through the connector. The thresholds need also to be decoupled by capacitors mounted on the top of the flex circuit. There will be 324 microns between the centers of the outside pads of two adjoining FPIX1 chips. Figure 2.14 shows a block diagram of the flex circuit and a photograph.

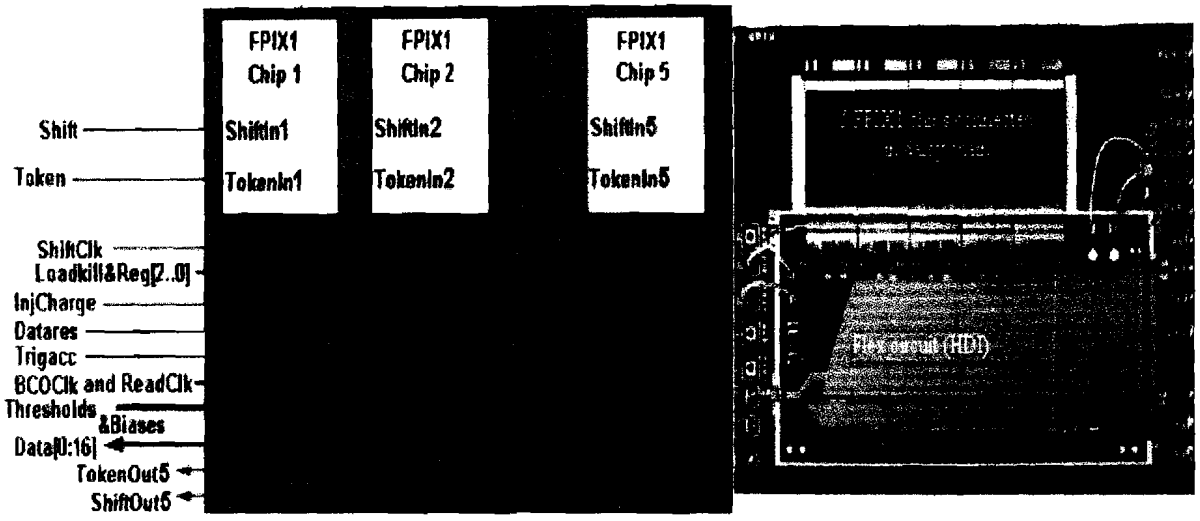


Figure 2.14. Block diagram of the flex circuit and a photograph.

Fujitsu Computer Packaging Technologies (FCPT) fabricated the flex circuit [8]. Because there are only 79.2 microns between pads a high density routing design is required in order to be able to route a trace between the outer row of pads of the FPIX1 chip. Minimum trace widths are 20 microns with a minimum clearance of 20 microns. Vias are also be very small. In table 4 we are shown Fujitsu's design rules.

Via Hole Diameter	25 μm
Via Cover Pad Diameter	108 μm
Via Center Spacing	208 μm
Via Cover Pad to Line Clearance	20 μm
Line Width	20 μm
Line to Line Clearance	20 μm
Line Center Spacing	40 μm

Table 4 FCTP's design rules.

The two rows of pads on the FPIX1 chip require a very high density design. The vias must be very small in order to fit between the traces. The traces need to be very small in order to permit the vias to be placed. A bus of top layer traces runs horizontally from the bonding pads that connect the flex circuit to the PCB. This bus of traces connects all of the common signals to the five FPIX1 chips. The second layer has traces that connect the FPIX1 chip signals to the bus. The trace widths are 20 microns with a 20 μm clearance. Vias will be 108 microns with a 25 micron hole size. Layer 3 is a power plane for Vddd and Vdda. Layer 4 is the ground plane.

The MCM operates like a single FPIX1, with the only difference that in the single FPIX1 we only had 2880 pixel cells, and with the MCM we have 14400 pixel cells. In the single FPIX1 we found that the **ShiftOut** signal is inverted in relation of **ShiftIn**. This presented a problem when we connected them in a daisy chain. To solve the problem we designed the mask in a special way to download information to the chips both for initialization and to select the cells to be characterized. The readout operates more or less like a single FPIX1. The only difference is in the **TokenIn** signal. You need to be careful setting this signal because if you don't keep this signal at logic high for a sufficient time to readout all five chips, you can lose all the information.

2.2.3 Design of patterns to control the MCM

We need to design the patterns in order to download the information to control the chip and to read out the information generated by an injection on a specific pixel cell to characterize it. Since in the MCM we have 5 chips connected in daisy chain, and the ShiftOut signal is inverted respect to ShiftIn signal we need to send the mask in a specific format to avoid this problem. When you have only one chip on the module there is no problem, but when you add another chip, then you will have this problem, and to avoid it we decide to invert the logic of the commands, i.e. where you were sending a logic one to the first chip you need to send a logic zero to the second and vice verse. So we noticed that if we program an even number of chips we need to send the first mask in negative logic, and if we program an odd number of chips we need to send the first mask in positive logic. We synthesize this explanation in Table 3.

Number of FPIX1 chips	Logic of the mask
1	+
2	- +
3	+ - +
4	- + - +
5	+ - + - +

Table 3. Format to send the masks in the MCM

The first set of information that we download to the chip is in an array of commands called initialization. This array contains all the commands to do a reset and the configuration. To do a reset we need to send the scan path code 011 (**Reg2**=0, **Reg1**=1, **Reg0**=1) and 102 bits of zero logic by means of the **ShiftIn** pad, sending also the clock signal by the pad **ShiftClk**. Afterwards, we configure the FPIX1 chip. So, we select the scan path code 101 (**Reg2**=1, **Reg1**=0, **Reg0**=1). In this mask we send the lag number equal to two, to fill the 6 BCO bits, the chip ID number using the 4 ChipID bits, and, the mode of operation that is 1 bit long. This means that the mask is 11X5 bits long. A sketch of the FPIX1 chip initialization is shown in fig.2.15.

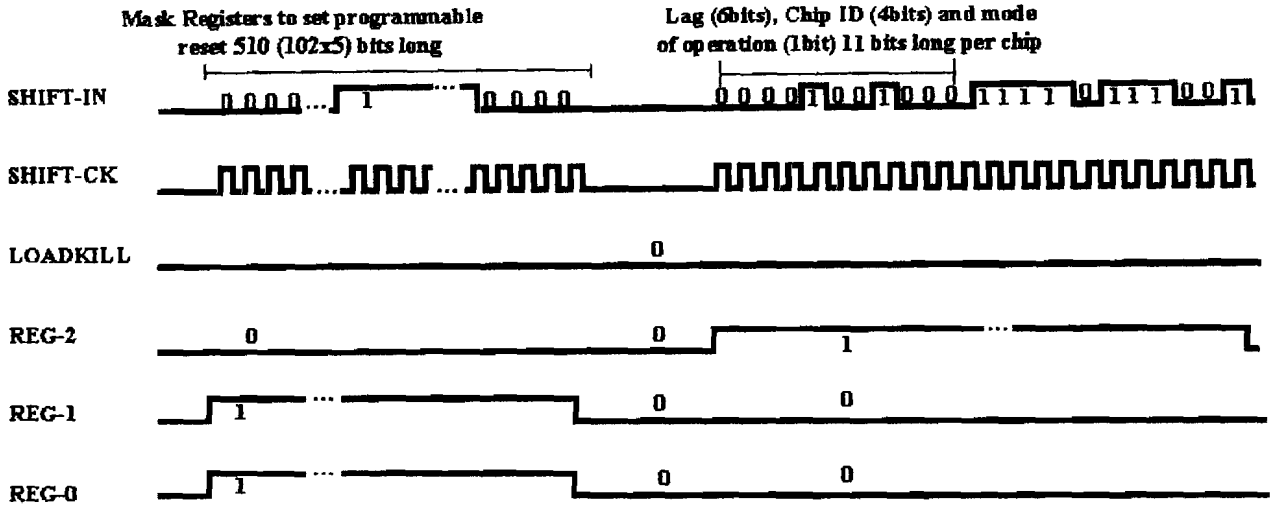


Figure 2.15 Initialization mask for the MCM

Later on we need to send the kill/inject mask selected by means of the path 111 (**Reg2**=1, **Reg1**=1, **Reg0**=1) or 110, which one we are selecting the pixel or pixel cells to be characterize. The kill mask is 14400 (2880x5) bits long, the inject mask is 14400 bits long. These bits correspond the 14400 pixel cells that the MCM contains. The format that we utilize to send these bits is as follows: the first 2880 bits are in positive logic, i.e., we send a zero logic in the kill mask to indicate that the specific pixel cell should be enabled to receive charge on it. Meanwhile we send logic one to indicate that a pixel cell should be disable. In the inject mask a logic one indicates that the pixel cell will not be injected and a logic zero that it will be. The second 2880 bits are in negative logic, that is the inverse logic of the first 2880 bits; the third are in positive logic; the fourth are in negative logic; and the fifth are in positive logic. The **ShiftIn** is delayed one pulse of clock from the **ShiftClk**. The frequency of the shift-clock is 12.5MHz. The **loadKill** needs to be enabled during the kill mask and disabled for the inject mask. In figure 2.16 we show the kill/inject mask for the MCM.

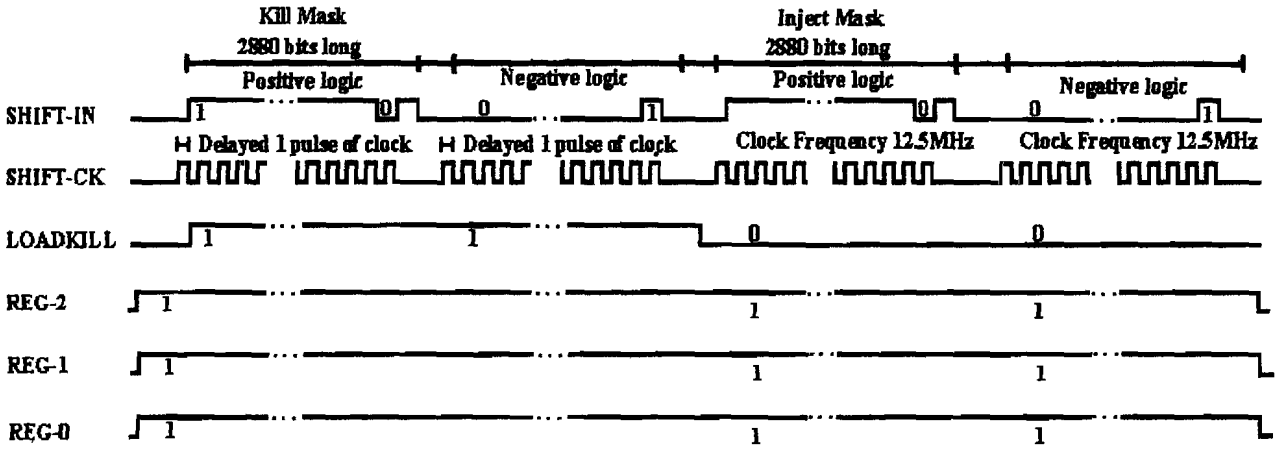


Figure 2.16 Kill/inject mask for the MCM.

After the selection of the pixel cells by means of the kill/inject mask, we need to provide to the MCM all the commands to do the readout to get the information that the module delivers in response to the charge injected. These signals are the charge inject (**InjectIn**), the data reset

(**DataRes**), the BCO clock (**BCOClk**), the readout clock (**ReadClk**), the token in (**TokenIn**) and the trigger accept (**TrigAcc**). All are contained in a special mask in a specific time sequence. Before sending the injection you need to send a **DataRes** signal in order to eliminate stale information. By sending a **DataRes** you perform a reset on the BCO counters, the End of column registers are reset to empty, and the chip stops outputting data. We send the charge inject signal $10\mu s$ after we send the **DataRes**, by means of the pin marked **InjecIn**. With these we are also sending two signals more, Clockpod1 and Clockpod2, with which we synchronize the logic analyzer to accept all the data that the chip delivers. A sketch of the readout is shown in Fig 2.17.

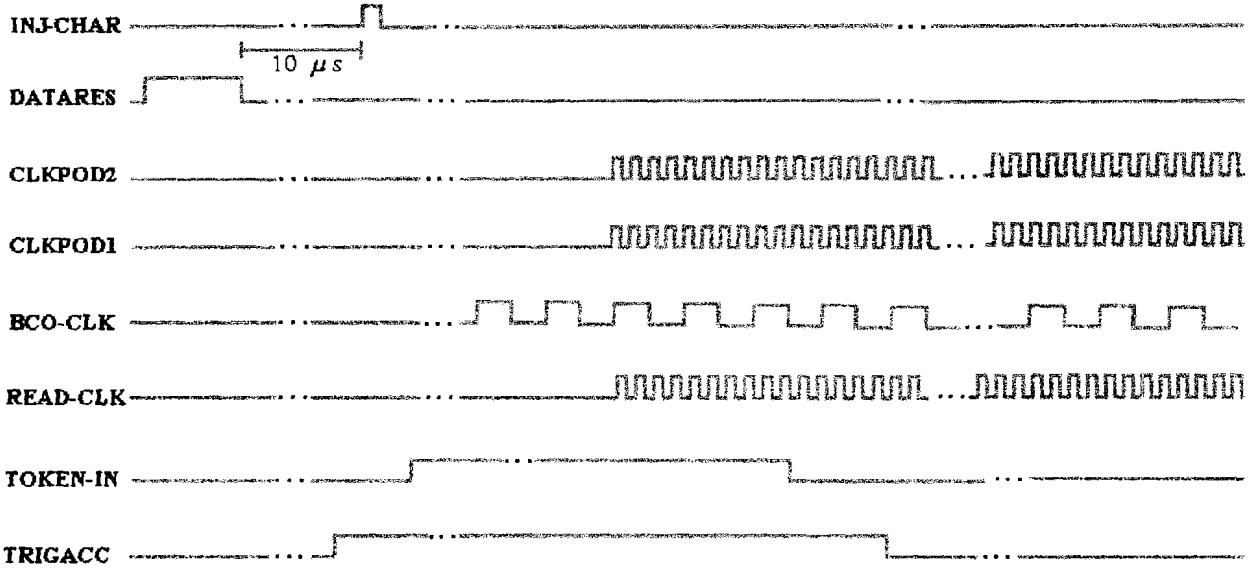


Figure 2.17 Readout mask for the MCM.

2.3 CONCLUSIONS

We described the operation of the FPIX1 chip and the MCM. Now we know how to handle these devices, and with this knowledge we are be able to perform tests on these prototype devices. In the following we will explain how to characterize the FPIX1 chip and the MCM.

CHAPTER THREE

CHARACTERIZATION

We have developed a test stand for use in characterizing the FPIX1 front-end electronics chips that are one version in the FPIX sequence of VLSI chips being developed at Fermilab [5] for readout of the BTeV [2] pixel detector. This detector will provide high-resolution space points near the interaction region for use in reconstructing tracks and vertices. The information the detector provides will be used in the first level trigger to select events that have a high probability to contain secondary decay vertices. This means that all of the hit information from every beam crossing must be made available to the trigger processors. A beam crossing occurs every 132 ns.

In order to achieve the required resolution, $\leq 9\mu m$, the pixel unit cells must not only be very small, $50\mu m$ by $400\mu m$, but the charge deposited in each must also be digitized and read out. The FPIX chips contain front-end electronics cells with the same dimensions as the pixels on the sensors. The number of FPIX chips which will be bump-bonded to each sensor will depend on the number of cells on each FPIX chip. The FPIX1 version of the readout chips contains 2880 cells. Future iterations are expected to contain an even larger number.

If the readout is to be accomplished in the short time between crossings, the information must be sparsified so that only valid hit data is presented to the trigger processors. Which cells are read out is determined by a discriminator in each cell. If a signal above threshold is detected in the cell, then it is read out. The threshold setting for all cells on a single FPIX chip is the same. On FPIX1 it is set by a voltage input, called Vth0, using an external supply. On the next iteration, the threshold will be set by means of a digital to analog converter. The test stand was built for the purpose of testing that the measured threshold (charge required to record a hit) and noise levels for the 2880

discriminators on the FPIX1 chips were uniform enough that a single Vth0 for all of the cells would suffice. If there are significant cell to cell differences, then, since Vth0 must be set to a value well above the highest noise level measured, the charge information needed for accurate position measurements might be below the resulting charge threshold for some cells and would not be read out.

To gain experience on the technical issues involved in system integration, Fermilab has made a pixel-readout-chip sub-assembly containing five FPIX1 chips called the Fermilab pixel multi-chip module (MCM). Five FPIX1 chips have been integrated into a single module connected together by a flexible circuit, or High-Density Interconnect (HDI), made by Fujitsu Computer Packaging Technologies [8]. The tests described here were performed to characterize the 14,400 pixel cells contained on the five FPIX1 chips in the MCM. The tests were performed to characterize the chips alone, i.e., without bump-bonding them to a pixel sensor and also with sensors on it.

3.1 BAD PIXEL MAP TEST

This is the first test that we carried out on the FPIX1 chip to know the status of each pixel cell in the entire chip. The bad pixel map test is based on injecting charge into all cells on the chip to see what cells are working. First we program the FPIX1 chip to kill all cells except seventeen. By testing seventeen at a time we are able to save time. Then the chip must be programmed to allow charge to be injected into these seventeen cells. Afterwards we inject these cells with a charge level well above the discriminator threshold so that they are expected to register a hit each time that we inject them. This guarantees a response of 100% for correctly working cells. We scan the entire FPIX1 chip in this way, seventeen cells at a time to determine which cells are working correctly and which are not.

Once we know which of the pixel cells are working we can continue with further test to characterize the cells that work correctly. The tests that we perform on the single FPIX1 chip and also on the MCM are: Threshold uniformity test with and without sensor bump-bonded to the chip or chips, reproducibility of the test, hits studies, inject studies, and temperature studies. In the sections to follow we will describe each of the tests.

3.2 THRESHOLD UNIFORMITY TEST WITHOUT AND WITH SENSORS BUMP-BONDED TO THE CHIP

The noise and threshold tests are performed together because they use the same data acquisition software. The noise and threshold uniformity tests are based on programming the FPIX1 chip to kill all the pixel cells except one. Then, as above, we need to program the chip to allow charge to be injected into the active cell. After the chip is programmed, we send an injected charge pulse 1000 times and count the number of times we record a hit in the cell. This procedure needs to be repeated cell by cell to get the information for all the

pixel cells in the chip. We do this for a range of charges above and below the expected threshold. For each level of charge we need to scan all the pixel cells. The levels of charge must be chosen to bracket the voltage threshold. To do this we make a preliminary scan over the chip to determine the optimal range for the test. For each test we fix Vth0 at a particular value. We have carried out these threshold uniformity and noise tests in four different regions of operation. One of these regions is 2000-3000 electrons, which corresponds to approximately 10% of the 25000 electrons signal expected for a minimum ionizing particle.

To save time, instead of injecting only one pixel cell at a time we inject seventeen pixel cells. We leave twenty cells between them to avoid all possibility of crosstalk. After acquire the information from the chip we need to process this information to get the final values of interest. The process is as follows: we need to count the times each one of the cells in the specific group of seventeen pixel cells records a hit when injected, and we need to record the level of voltage used to provide the charge for each injection. Later on we use this data to obtain the efficiency. The efficiency is obtained by dividing the number of times that the pixel cell recorded a hit by the total number of injections that we provided it. Then, for each cell, we need to graph the efficiency versus the level of voltage at which the efficiency was obtained. From the graph we can determine the threshold value, Vth_{exp}, which is chosen to be the first value of the voltage that has greater than 50% hit efficiency. For the noise value, we need to find the first voltage for which the hit efficiency is greater than 81.5% and subtracts the value for 50% hit efficiency, which yields σ_{exp}. This procedure needs to be repeated for each of the seventeen pixel cells and then, seventeen at a time, for all the pixel cells that FPIX1 contains.

After determining Vth_{exp} and σ_{exp} from our experimental data, we improve on these values by fitting the experimental points to a functional form for the efficiency versus the voltage. The experimental values will be the initial guess coefficients in the technique that we will use called the nonlinear Levenberg-Marquardt (Lev-Mar) method. This method determine a nonlinear set of coefficients which minimize the chi-square quantity. After obtaining all the fit values for all the pixel cells analyzed we need to obtain the Gaussian curves that best fit the histograms of these values for all the pixel cells characterized. From these, we obtain the mean threshold, the sigma threshold (threshold dispersion), the mean noise and also the noise dispersion, all in units of volts and electrons for all the cells on the FPIX1 chip. These results represent the characterization of the chips.

The threshold uniformity test with sensors bump-bonded to the chip is the same as without the sensors. The only difference is that now we have the sensors on the chip. With that we can study the effect that the sensors have on the chips. Thus we will compare the mean threshold, the sigma threshold (threshold dispersion), the mean noise and the noise dispersion with and without sensors on the chip or chips characterized to see how much if any effect the sensors cause.

3.3 REPRODUCIBILITY

The main goal of the reproducibility test is to demonstrate how reproducible the tests are. This is based on repeating the test that you want to verify several times and comparing all the results. If there are no significant fluctuations, we can say that the test is reproducible. If the results show us that the fluctuations are larger than the expected variation then the test is not reproducible.

In our case we performed the reproducibility test for the threshold uniformity test without sensor. We repeated the threshold uniformity ten times on the same pixel cells. Later we graphed the results to see how reproducible they were. We will see if the results changed a lot changes, if seen, could be the effect of noise in the room in which we are developing the test which might not be negligible.

3.4 HIT STUDIES

When you are doing the threshold uniformity test you need to provide to a specific group of pixel cells 1000 injections of charge by means of a voltage applied to the terminal InjectIn in the chip. But we would like to see what is the effect of the choice of the number of injections, i., e. if this number could be smaller without causing significant differences on the final results. This would mean a big reduction in time because of the large number of cells.

For this we need to perform the threshold uniformity test with different numbers of injections. The numbers of injections are 50, 100, 250, 500, 750 and 1000. We need to graph the results and the number of injections to verify whether there are significant variations on the results.

3.5 TEST SETUP

To realize the different tests described until now on the FPIX1 chip we need to arrange a set of instruments that permit us to do the tests. We need to provide power, both voltages and current. We need to send masks to select the specific pixel cells to be characterized. We need to provide to those pixel cells charge injections, and we need an instrument that can save the information that the FPIX1 chip delivers. Since we want to do the characterization automatically, we will use two computers with GPIB interface boards in order to communicate between the computer and the instruments.

To provide power to the chip we are using a power supply PS2520G which has the following characteristics [13]. The power supply has triple output, one of the outputs providing a maximum of 6 V and 3 A and the other two providing a maximum of 36 V and

1.5 A. The supplies can be operated in independent mode in which the output voltage and current of each supply are controlled independently. The PS2520G is GPIB programmable. The GPIB command sets are SCPI compatible. Each power supply features 100 memories to store voltage and current limits for instant recall via the front panel or GPIB. Both have a dual 4-digit LED display. We are using a current supply made at CPPM, Marseilles, to provide the feedback and Master Bias currents to the FPIX1 chip.

To download information to the chip we are using a digital data generator DG2020A made by SONY/Tektronix, designed for high frequencies and easy to use [14]. The instrument provides the high performance tools needed to evaluate advanced digital semiconductors and logic circuits and, has multiple outputs channels up to 36. It also has variable output delays (2.5ns to 20ns) and variable output level (-3V to 7V) by means of the SONY/Tektronix P3420 programmable output. The DG2020A is able to create complex data patterns in a sophisticated sequence, including looping, jump on event, and tri-state output control. It has a memory word size of word from 64 bits up to 64000 bits with a maximum range of frequency of 200MHz. The DG2020A has excellent signal fidelity and flexible sequence control with jump, event, and nested loops as well as a large display for easy-to-use data editing. It also has a GPIB/RS-232C interface for communication with the computer.

To provide the charge injection we use two different instruments to see if the results of the characterization changes when you provide two different waveforms with two different instruments. The first one is a function generator made by Stanford Research Systems DS345 [15]. It has the following characteristics: frequency output from 1mHz to 30.2MHz; 16,300 point arbitrary waveforms sine, square, or ramp. It has both RS-232 and GPIB interfaces. The second one is a pulse generator HP8110A, and has the following characteristics: generates all standard pulses, digital patterns and multi-levels waveforms needed to test CMOS and other digital designs up to 150MHz; has a GPIB communication; has high pulse integrity with 10ps timing resolution and down 20ps RMS-jitter. To check the pulse that we inject we use an oscilloscope.

To acquire the information that the FPIX1 delivers in response to a charge injected in a pixel cell, we use a logic analyzer system HP16500C [16]. This instrument is the preferred tool for debugging physical prototypes of complex digital systems, and to debug and verify physical prototypes of complex, high-speed digital products. Problems discovered in this stage of a project may have roots in architecture, logic design, timing or analog behavior. The HP 16500C incorporates a lot of capabilities and makes the local area network (LAN) connection a standard feature. Available measurement modules include general-purpose state and timing analyzers with memory depth up to 2 M samples; timing analyzers up to 2G-samples/s; oscilloscope modules with bandwidth up to 500MHz; a pattern generator with 200 M vectors/s; and the HP MultiProbe system, which allows the user to simultaneously attach analog scope probes to hundreds of pins on fine-pitch surface-mount integrated circuits.

Connecting the logic analyzer to a LAN makes it possible for team members located in different cities to work together efficiently to solve problems. The LAN also can be used to import data into a personal computer or workstation for convenient post-capture analysis,

for documentation and for archiving. The HP 16500C can be operated remotely over the LAN by using an X Windows interface. The HP16500C has a target control port, allows users to remotely reset a target system, activate interrupt lines or set counters, and also has a centronics parallel printer interface that makes it possible to use inexpensive, readily available printers with the HP 16500C.

To save time we decided to use two computers one for the data acquisition and the other for the processing and analysis of the information. The computer that acquares the information is a PC PentiumI, 133MHz, with the WindowsNT operating system. The computer that process and analyze the information is a PC PentiumII, 200MHz, with the operational system WindowsNT.

To communicate between the data acquisition computer and the instruments in order to do the characterization, we use a General Purpose Interface Bus (GPIB). The GPIB is an interface design which allows the simultaneous interconnection of instruments on a common communication line, including a computer controller. Groups of instruments may be provided with a sequence of commands so as to set up the complete control of a measurement of a product under test.

The GPIB uses a set of commands under a standard know as IEEE-488 and IEEE-488.2. Bus devices are classified on a general basis, according to their requirements and facilities, e.g., if they are listeners with respect to the bus, which can receive data from other sources. Such devices would include programmable power supplies display units and signal sources. Talkers are essentially measuring instruments for example frequency counters, analyzers, etc, which can send data to other devices such a computers or a printers. The controller controls the flow of data information on the bus and is able to signal which device or devices are to listen and which are to talk. In addition, it can issue special commands required for bus management. Usually the controller function is incorporated in a computer [17].

The basic GPIB standard provides for the interconnection of up to fifteen instruments or devices, this total including the computer/controller. The GPIB is potentially quite fast up to 1M byte transfers per second are possible. Instruments with differing data rates may generally be interconnected without problems.

A picture of the setup is shown in the figure 3.1, we can see all the instruments that will allow us to characterize the FPIX1 chip and the multi chip module.

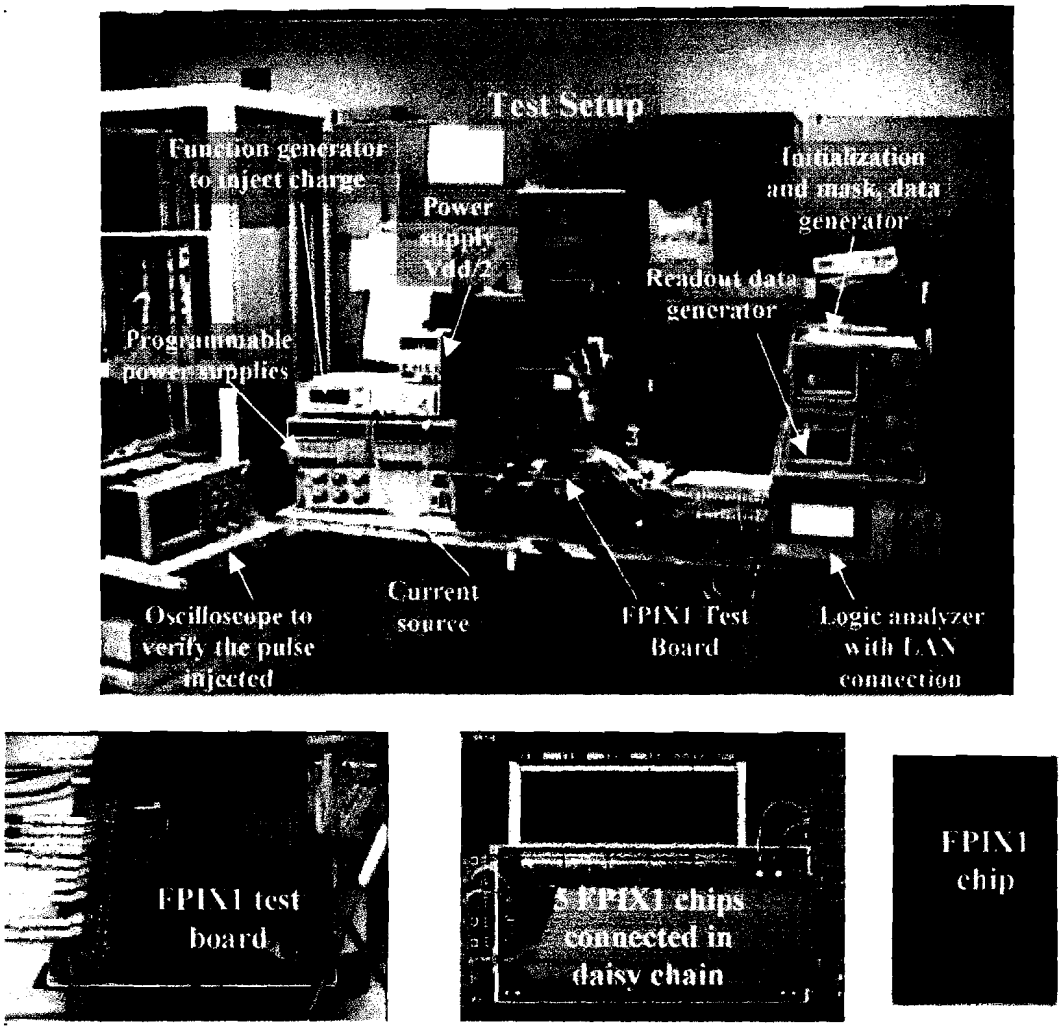


Figure 3.1 Test Setup to characterize the FPIX1 chip and the MCM.

3.6 CONCLUSIONS

We have described the test that we need to do in order to do the characterization on the MCM and the FPIX1 chip. With this we now have a clear idea of the problem that we need to solve. In the following chapter we will describe the procedures and the design of the software to acquire the data and the software to process and analyze the information acquired. The two programs together represents the characterization.

CHAPTER FOUR

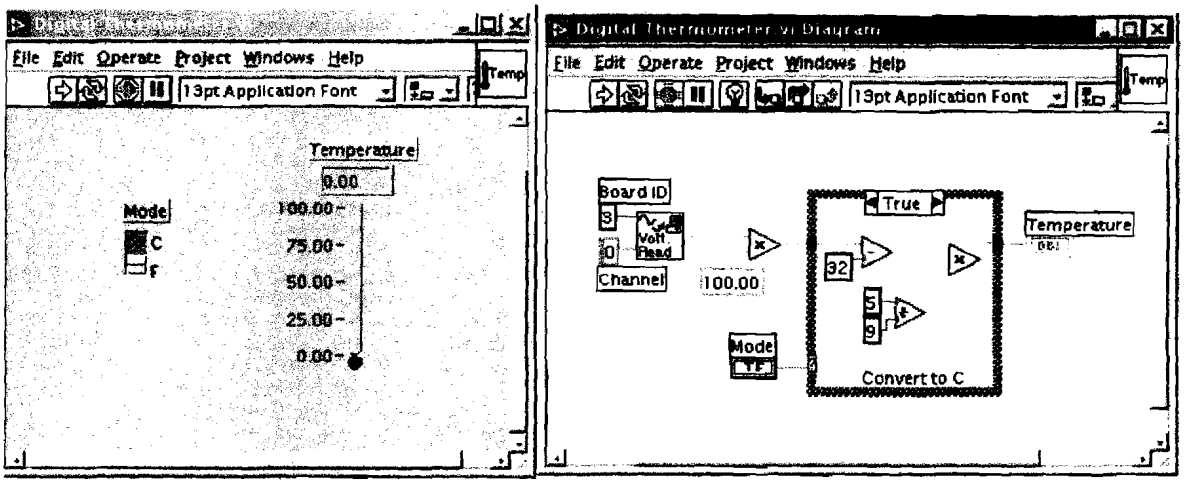
SOFTWARE

The entire BTeV pixel detector will contain nearly 30 million pixel unit cells each bump-bonded to its associated front-end electronics cell. Since the total number of components is so large, they will not be tested at a single facility, but will be tested at three or four different collaborating institutions. For this reason, the test stand has been developed using test equipment that is commercially available and in common use. Also, the software has been developed utilizing the Laboratory Virtual Instrument Engineering Workbench (LabVIEW) [4], which is widely used among the collaboration and which is familiar to both physicists and engineers. LabVIEW is a development environment based on the graphical programming language G. It is integrated fully for communication with hardware such as GPIB, VXI, PXI, RS-232, RS-485, and plug-in data acquisition boards by means of PCI interfaces. It also provides an extensive set of commands for file manipulation, including Ethernet transfers. This programming language is both very easy to understand and flexible. It also incorporates a number of sophisticated analysis techniques. We were thus able to develop the complex software needed for these tests entirely within this framework. The software has been designed to be very versatile, i.e., utilizing this software you can characterize the entire MCM or you can characterize a single FPIX1 chip. In the following sections we will describe the design of the software.

4.1 THE LabVIEW FRAMEWORK

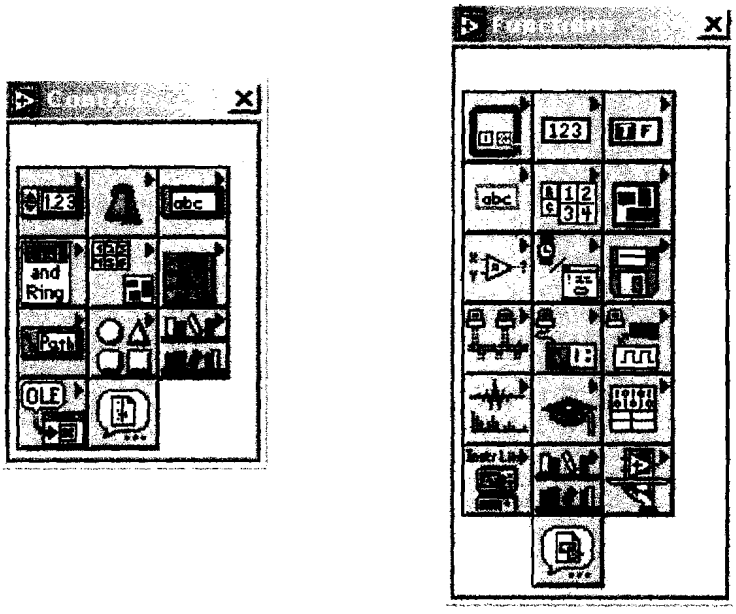
Thousands of successful engineers, scientists, and technicians use LabVIEW to create solutions for their demanding application needs. LabVIEW is a revolutionary graphical programming development environment based on the G programming language for data acquisition and control, data analysis, and data presentation. LabVIEW gives you the flexibility of a powerful programming language without the associated difficulty and complexity of competing text-based frameworks because its graphical programming methodology is inherently intuitive to scientists and engineers [18].

With LabVIEW, you build VIs instead of writing programs. You quickly create front panel user interfaces, as shown in figure 4.1a, giving you the interactive control of your software system. To specify the functionality, you intuitively assemble block diagrams (figure 4.1b) that provide a natural design notation for engineers and scientists. On the front panel of your VI, you place the controls and data displays for your system by choosing objects from a special tools database called Controls palette, showing in figure 4.2a, including numeric displays, meters, gauges, thermometers, tanks, LEDs, charts, graphs, and more. When your VI is complete, you can use the front panel to control your system while the VI is running by clicking a switch, moving a slide, zooming in on a graph, or entering a value from the keyboard. To program the VI, you construct the block diagram without worrying about the many syntactical details of conventional programming. You select objects (icons) from the Functions palette shown in figure 4.2b, and connect them with wires to pass data from one block to the next. These blocks range from simple arithmetic functions, to advanced acquisition and analysis routines, to network and file I/O operations.



a) b)

Figure 4.1 User interface and block diagram in an digital thermometer written in LabVIEW framework.



a) b)

Figure 4.2 Controls and functions in LabVIEW.

LabVIEW uses a patented dataflow programming model, called G, that frees you from the linear architecture of text-based languages. Because the execution order in LabVIEW is determined by the flow of data between blocks, and not by sequential lines of text, you can create diagrams that have simultaneous operations. LabVIEW is a multitasking and multithreaded system, running multiple execution threads and multiple VIs. LabVIEW VIs are modular in design, so any VI can run by itself or be used as part of another VI. You can create an icon for your own VIs, so you can design a hierarchy of VIs and subVIs that you can modify, interchange, and combine with other VIs to meet your changing application needs. In many applications, execution speed is critical. LabVIEW is the only graphical programming system with a compiler that generates optimized code with execution speeds comparable to compiled C programs. With the built-in Profiler, you can analyze and optimize time-critical sections of code. Consequently, you can increase your productivity with graphical programming without sacrificing execution speed [18].

The LabVIEW VISA, GPIB, VXI, and Serial VI Libraries use National Instruments industry-standard device driver software for complete control of your instrumentation. You can control any GPIB instrument connected to a National Instruments IEEE 488.2 interface board. Your VXI instruments are easily programmed with VISA (Virtual Instrument Software Architecture), the interface-independent software interface endorsed by the VXIplug&play Systems Alliance. You can communicate with your instruments from embedded VXI controllers, PXI modular systems, or computers with an MXI or GPIB-VXI interface. With the LabVIEW Instrument Wizard, you can immediately detect any instrument connected to your computer including GPIB, VXI, serial, and computer-based instruments. The wizard installs appropriate instrument drivers and helps you communicate with your instruments in minutes. LabVIEW has more than 600 instrument drivers available free on CD or from the Instrument Driver Network. LabVIEW features powerful,

comprehensive analysis libraries that rival those of dedicated analysis packages. These libraries are complete with statistics, evaluations, regressions, linear algebra, signal generation algorithms, time and frequency-domain algorithms, windowing routines, and digital filters.

4.2 THE DATA ACQUISITION SYSTEM

The first setup used to characterize the FPIX1 chip we used a PC with an AT/TNT-GPIB (General Purpose Interface Bus) interface to control the following instruments: Tektronix PS2520G programmable power supply, Hewlett Packard HP16500B logic analyzer system, Stanford Research Systems DS345 30MHz function generator, SONY/Tektronix DG2020A data generator and SONY/Tektronix P3420. The software developed to control all these instruments utilizes the LabVIEW[3] (Laboratory Virtual Instrument Engineering Workbench) framework [3]. LabVIEW is a development environment based on the graphical programming language G. It is integrated fully for communication with hardware via GPIB, VXI, PXI, RS-232, RS-485, and plug in data acquisition boards.

The first test stand that we are using to characterize the FPIX1 chip is shown as Fig. 4.3. The programmable power supply has three outputs. One of them provides power to the digital part of the FPIX1 chip. The second output provides power to the analog part. The last one provides power to the Vth0 and represents one of the inputs to the discriminators in the front ends. We are using two data generators with programmable outputs to send the information to program the chip. One of them provides data for the Kill/Inject path. The other sends a sequence of commands to the readout stage to acquire the information that the FPIX1 chip delivers. The logic analyzer system receives the information that the chip delivers and stores this information in its memory. The function generator provides the charge to be injected to the selected cells of the FPIX1 chip. We also have two other instruments that it is not necessary to drive by means of the GPIB card. These are a regulated power supply, LEADER 718-SD, which provides power to the vdd/2 input of the FPIX1 to enable single-ended readout, and a current supply made at CPPM, Marseilles, to provide the feedback and Master Bias currents through the inputs marked Ivfb and Ivbbp.

The noise and threshold tests were performed together because they use the same data acquisition software. The pixel cells characterized are a sample of 170 cells selected from a coarse grid which covers the first 100 rows on FPIX1. These noise and threshold uniformity tests are based on programming the FPIX1 chip to kill all the pixel cells except five (to save time), where the five selected are separated from each other by a sufficient distance to remove all possibility of crosstalk between them. Then, as above, we need to program the chip to allow charge to be injected into these five activated cells. This is one of the inputs of the discriminator. The other input is the Vth0 (Voltage threshold). This voltage stays constant. If the signal is higher than the Vth0, we will have a hit in the output of the discriminator. This signal asserts the wire or'ed HfastOR. After that, the information is stored in the cell until EOC set issues an output command. When this occurs the

command interpreter issues a bus request and asserts the wire or'ed RfastOR signal and then we will have data on the bus.

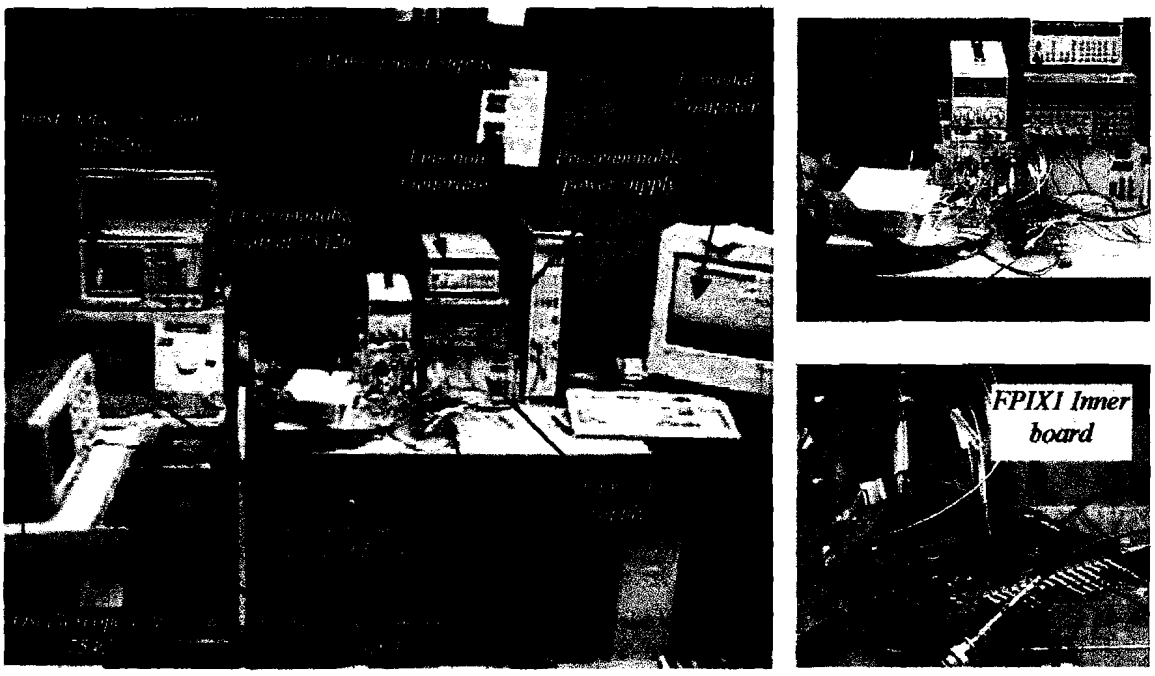


Figure 4.3 First test stand designed to characterize a single FPIX1 chip.

We scan the FPIX1 chip as follows. The FPIX1 chip is an array of pixel cells arranged in 18 columns by 160 rows. In this specific chip above row 100 the pixel cells are not connected so, we only have 100 rows working, also column 0 is a special column, for that we are not using; this means that we have 17 columns that responds to our computing control signals. To eliminate the possibility of crosstalk between cells we leave 20 cells between each of the cells to which charge is injected. For example the first five cells injected are (1,0), (2,20), (3,40), (4,60), (5,80) and the next five will be (1,1), (2,21), (3,41), (4,61), (5,81) and so on, until (1,19), (2,39), (3,59), (4,79), (5,99). This procedure is the same for all the groups of five cells, but we have also two corners with 4,3,2 and 1 pixel cell at a time. To scan these corners we utilize the same procedure as for the five pixel cells at a time. A sketch of the scan over the full chip is shown in figure 4.4.

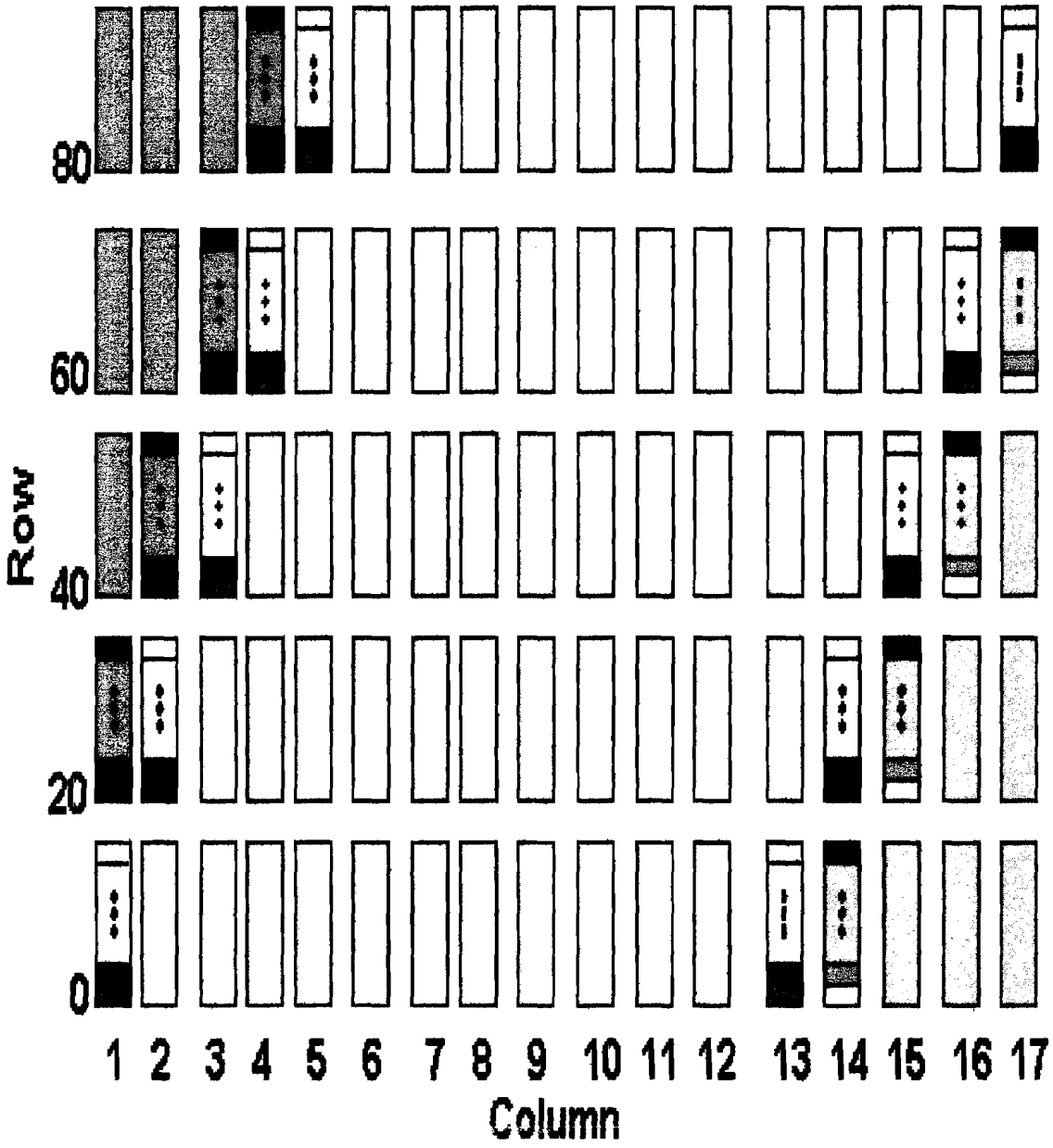


Figure 4.4 Pixel cell scan.

The general flow diagram is shown as Fig. 4.5. First we are reading the number of intervals of charge inject on which we want to do the test, we need to type the initial value, the final value and the step. This means that if for example you want to do the test in one interval that it is from 10mV to 100mV in steps to 10mV by 10mV, you will have 10 levels of voltage. In this program you don't need to type all the 10 levels for this case, you only need to type the interval number, the initial value, the final value and the step and the program calculates automatically all these levels of voltage.

The program has the particularity of accept more than one interval of voltages. With this we are obtained the number of levels on which the program will do the test, later on the program loads the initial kill/inject mask and readout patterns. Send to program the function generator with the current voltage. Send to clear the kill/inject patterns, this means that when you select the first five pixel cells and do the process, when you will select the next five pixel cells you need to disable the previous cells. With this clear we are disable these previous cells to enable the next five cells. With this we are getting the new group of five cells and we are sending to program the first data generator with the new group of cells. Send to put the logic analyzer in standby, initialize a counter. Send to program the second function generator to send the trigger to the function generator to tell him that it is time to send the inject charge, and also send the commands to do the readout. Save the data that the chip delivers in the internal memory of the logic analyzer. Ask if the counter is equal to 1000 if not come back and to again the full procedure, if yes the PC gets all the data from the logic analyzer.

The PC counts how many times is answering each cell of the group, save the result in a file, increment other counter. Ask if the counter is equal to 42, this means the 21 groups of cells, but how we are scanning the first two rows, this represent a total of 42 groups of cells. If the answer is no, come back and do the process until the answer is yes. Then ask if all the levels of voltage it was running if the answer is not, come back and do all the process with a new level of voltage. If the answer is yes, process all the data to get the errf(Vpp) for each pixel cell. Analyzes the data resulting to get the histogram and the gaussian for the voltage threshold and noise and ends.

The first subroutine is the process data to get the errf(Vpp) for each cell. The data processing software calculates the efficiency as a function of the voltage peak-peak by counting the number of hits (of 1000 possible) received. We then perform a least squares fit using the nonlinear Levenberg-Marquardt (Lev-Mar) fitting technique on the data for each cell to obtain the best threshold and noise value.

The second subroutine is the analysis to fill the histogram and perform the gaussian fit for the distributions for the 170 pixel cells, the program then performs a statistical analysis using the values from the 170 cells to get the means and standard deviations of the threshold and noise in units of electrons. At the end the program displays histograms of the noise and thresholds for the 170 cells along with the corresponding gaussian curve.

Both subroutines are shown in figures 4.5 and 4.6.

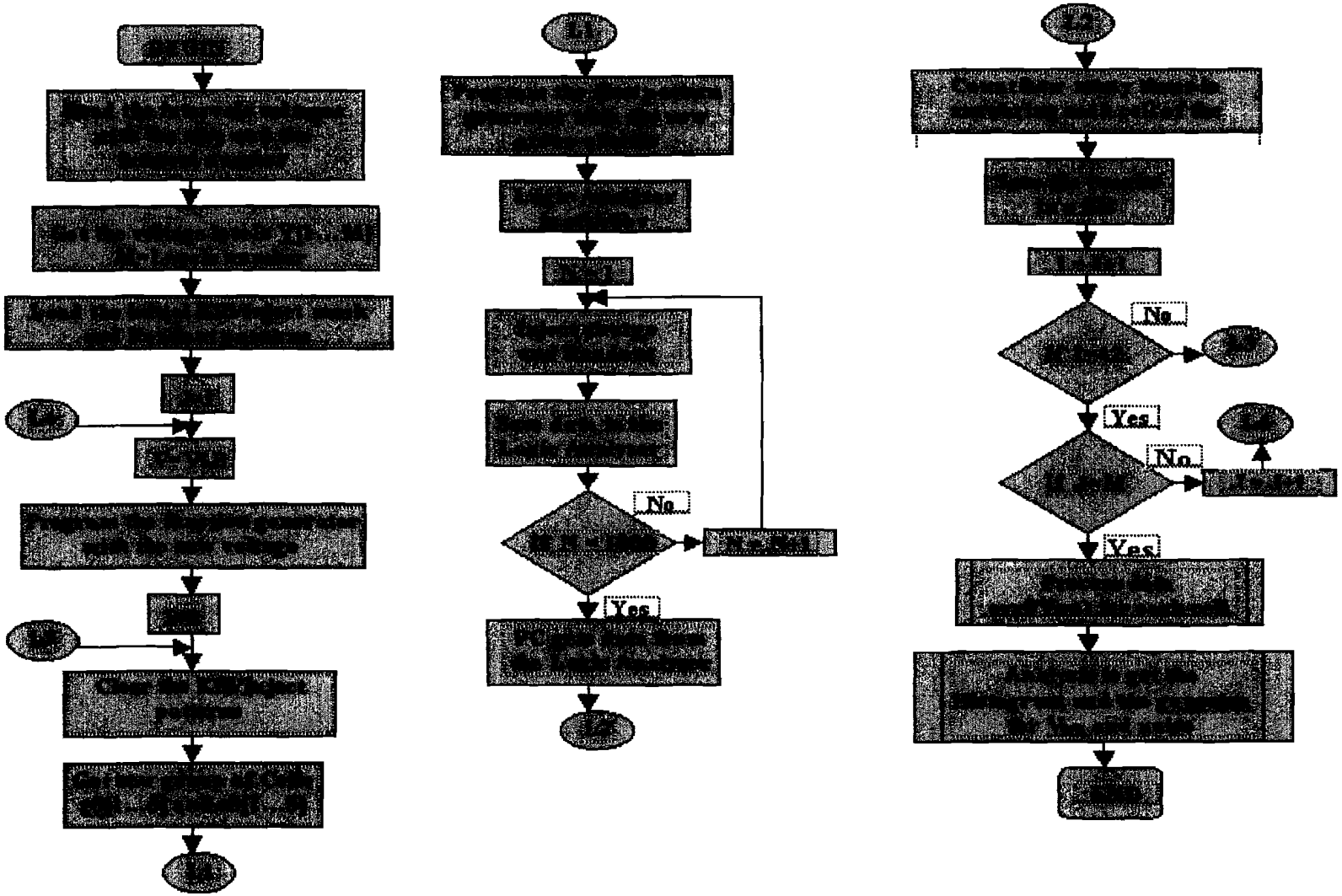


Figure 4.5 General flow diagram for the first software designed.

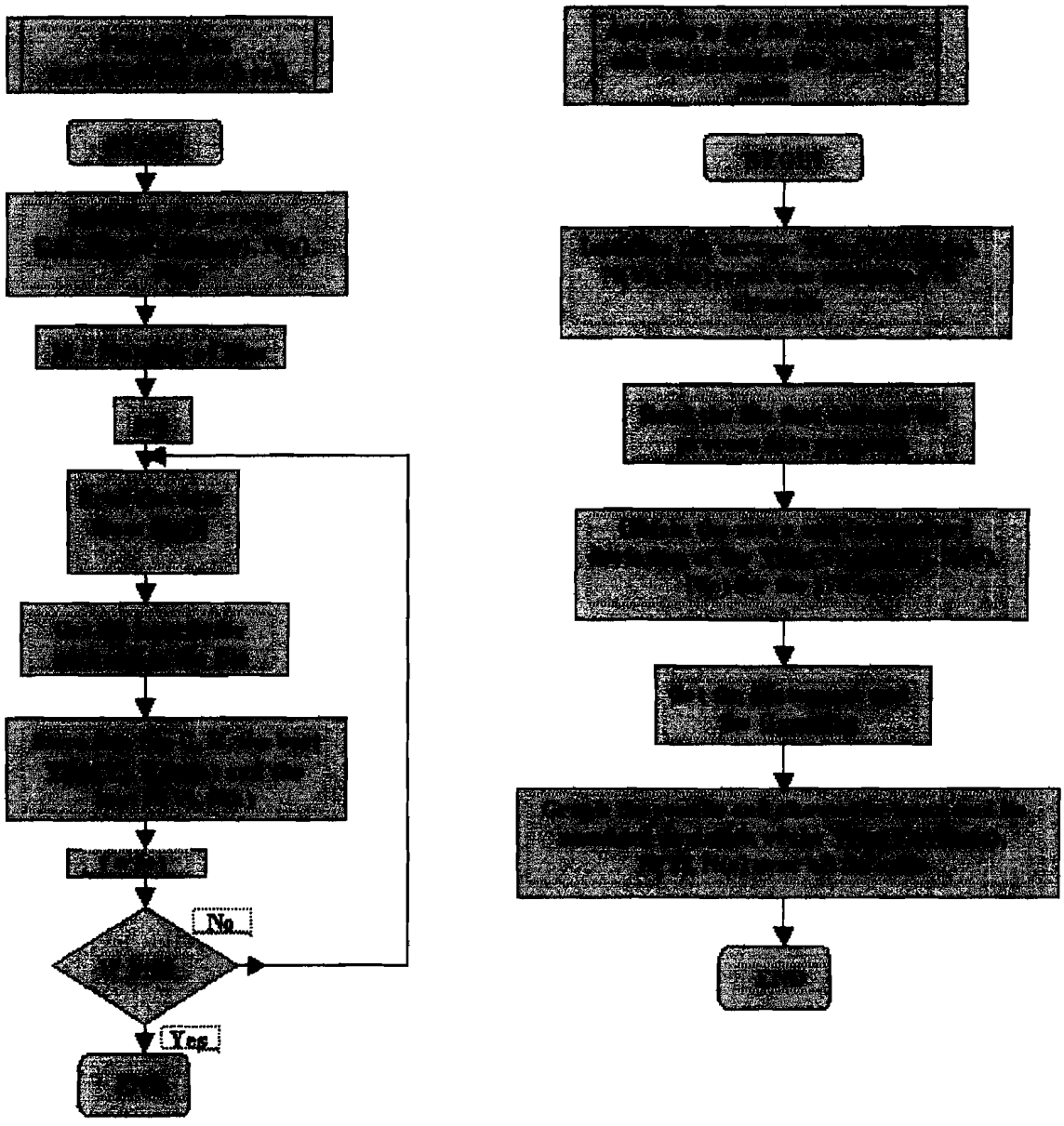


Figure 4.6 Principal subroutines of the first software designed.

To do the design the software we need to keep in mind the time response of the instruments to the commands sent to them by means of the GPIB interface. This times response are summarized in table 4.1

Instrument	Operation	Response (seg)
Data generator	Initialization kill/Inject masks	60
	Readout	60
Logic analyzer	Acquisition	20
	Transfer of data form the logic analyzer	60
Computer	Count	30
	Process	25200

Table 4.1 Time response of the instruments

Then, the noise and threshold uniformity test was performed in approximately 85.4 hours, for that we need to do modifications on the first software designed to be able to reduce the time of running of the software.

The new software we have developed takes advantage of the very flexible environment provided by LabVIEW as well as the advanced data analysis features it provides. While the use of LabVIEW and the relatively slow General Purpose Interface Bus (GPIB) for data acquisition meant that the tests were potentially time-consuming, several time-saving measures were taken that increased testing speed considerably. The test described in detail here is for the five chips on the MCM. The single chip test is the same, except for the masks.

First, two personal computers are employed, operating in parallel, both running LabVIEW programs under the Windows NT operating system. The first, PC1, is programmed for data acquisition. The second, PC2, performs the data analysis. The use of a logic analyzer with a Local Area Network (LAN) connection to receive the raw data and pass it on to PC2 makes it possible for the two computers to work simultaneously during the tests.

Second, two data generators are used to send commands to the chips and to the various instruments. This means that only small changes are needed on each cycle and thus a small amount of new information needs to be downloaded per data read cycle. Because of the many cycles involved, this results in a very large reduction in time. The data acquisition program running in PC1 synchronizes the action of the two.

We have demonstrated that an adequate characterization of each chip, by which we mean measurements of discriminator threshold uniformity and electronic noise dispersion, can be achieved by charge injecting 10% of the cells in a grid that covers the entire chip

uniformly. The threshold and noise results vary randomly within the errors over the chip and do not appear to depend on the location of the cell being tested. We performed an initial test using a very low threshold on all cells to see which cells were working. This test revealed that the last 60 rows in each column did not record hits. This is a feature of the layout that will be corrected on the next submission. Furthermore, the first column on these prototype chips was made to have outputs that could be visualized on an oscilloscope and could not be read out by the EOC logic. Thus there are actually only 1700 cells to be characterized on each FPIX1 chip (17 columns times 100 rows) in the MCM. Ten percent of this is a total of 170 cells to be charge injected on each of the five chips. Again, with time constraints in mind, we inject these 17 at a time, making sure that the ones we are injecting at the same time are separated by sufficient distance (20 cells) to avoid crosstalk. The scan of the seventeen pixel cells is shown as fig. 4.7

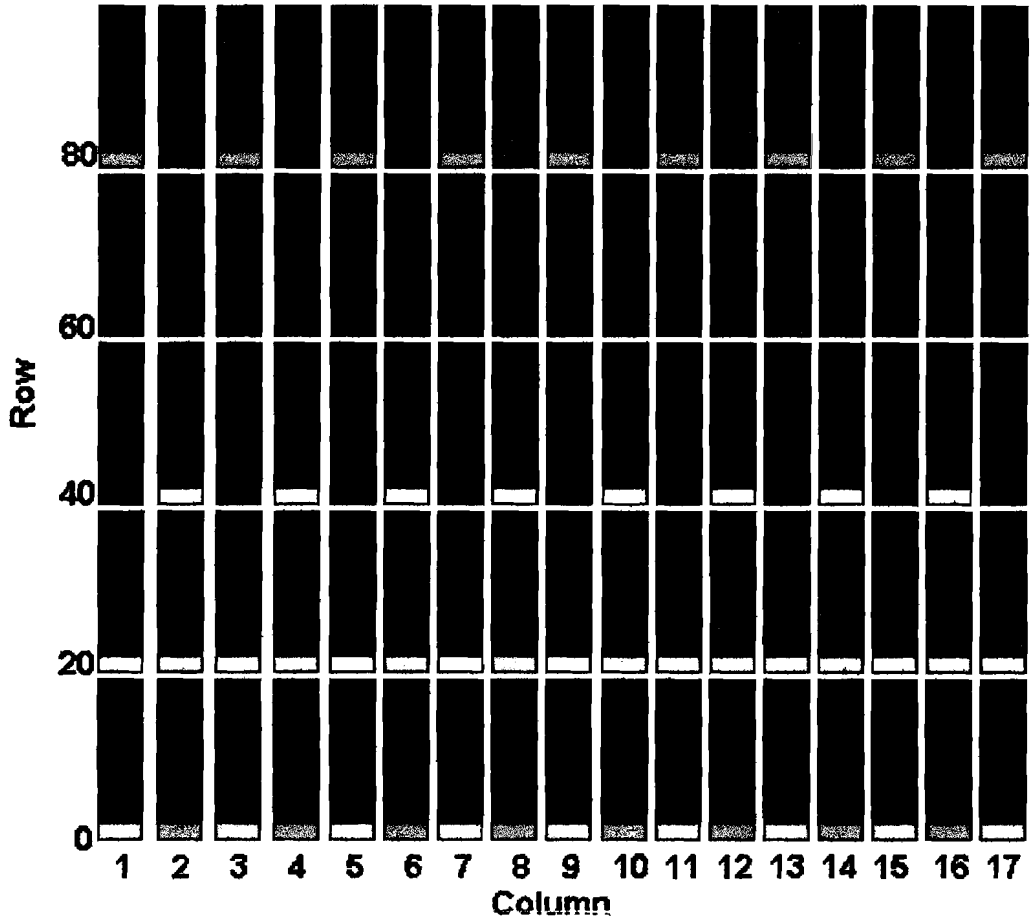


Figure 4.6 Pixel cell scan each seventeen pixel cells

PC1 contains an AT/TNT-GPIB interface to control the following instruments: Two Tektronix PS2520G programmable power supplies, a Hewlett Packard HP16500C logic analyzer system, a Stanford Research Systems DS345 function generator, two SONY/Tektronix DG2020A data generators and two SONY/Tektronix P3420 programmable outputs.

The test proceeds as follows: The program running in PC1 first initializes all the instruments used in the test and sets them to their nominal values. The two programmable power supplies provide power to the digital part of the FPIX1 chips, the analog part of the FPIX1 chips, and to all five Vth0 terminals of the chips on the module. These represent one of the inputs to the discriminators in the front ends. There are also two other instruments that are set manually. These are a regulated power supply, LEADER 718-SD, which provides power to the vdd/2 input of the FPIX1 to enable single-ended readout, and a current supply made at CPPM, Marseilles, to provide the feedback and Master Bias currents.

PC1 then initializes the five FPIX1 chips using the first of the two data generators. First, it configures the chips in continuous mode, and, since we have several chips connected in daisy chain, it gives each of the chips a unique ID number (ChipID). The FPIX1 chips deliver their information in two words, the data word and the control word. The data word contains the ADC value, the column and the row; the control word contains the ChipID number and the BCO number, which for these tests was always zero. PC1 then programs the five FPIX1s to kill (disable) all the pixels cells except 17 by means of the first data generator. The data generator sends the information to the chip in serial format. To make this easier to do we decide to generate the input parameter from the number of the pixel cell in terms of column and row. To do that we deduced a couple of equations that can do the translation of column and row into a serial number. As we discussed in chapter two there is a special scan path, for odd columns, the scan path starts in the lowest pixel and goes up to the highest pixel, and for even columns, the scan path starts in the highest pixel and goes down to the lowest pixel. Then the equation for an odd column is:

$$PC = (OddCol * 160) + NumRow \tag{4.1}$$

And for an even column the equation is:

$$PC = [(EvenCol + 1) * 160] - 1 - NumRow \tag{4.2}$$

Where: PC is the number of the pixel cell in the serial stream to be selected in the data generator,
$OddCol$ is the number of the odd column,
$EvenCol$ is the number of the even column,
$NumRow$ is the number of the row.

The same data generator is then used to enable charge injection to the 85 cells (17 times 5) it has selected. Next PC1 sends commands to the logic analyzer to initialize it and to tell it to wait for a trigger that is based on bits in the data format that the chips deliver. As we saw in chapter two there is a signal that the chip delivers called Data Valid. When you have a hit in the discriminator output this signal will be logic one (3.3V), and with the signal called Data 15 you can identify if the information acquired is a control word or a data word. Then, the trigger that the logic analyzer is waiting for when at the same time a logic one is present in the signal Data Valid and a logic one is also present in the signal Data 15.

Following this, PC1 uses the second data generator to send a trigger to the function generator DS345, which injects charge to each of the enabled cells. Any cell in which this charge produces a signal at the output of the second amplification stage that is greater than Vth0 will have a hit in the output of its discriminator. The cell then asserts the wire or'ed HfastOR and stores the hit information until the EOC set issues an output command. When this occurs the command interpreter issues a bus request, asserts the wire or'ed RfastOR signal, and presents its data on the bus.

The second data generator has also been programmed to send the commands to perform the read out of the FPIX1 chips a fixed time after the charge is injected. Upon receipt of these commands, the FPIX1s send the data generated in response to the injected charge pulses to the logic analyzer in the order in which the token passes through them. The logic analyzer acquires the information in binary format and transmits it by means of a LAN connection to the other computer (PC2), which processes the data. For each set of 85 pixel cells PC1 repeats the injection of the same charge the times of charge injections that you want, e.g., 50 times or 500 times and does this for all the charge levels contained in a specific interval values of the charge it injects, for example in a specific test we had 60 different values of the charge. When all the levels of charge have been scanned, the program changes the selection of the pixel cells and repeats the whole procedure until all 170 pixel cells on each of the five FPIX1 chips have been scanned.

The change of the pixel cells is done in an original and special way. First you select the pixel cells to which you will send a logic zero (0V) choosing the right location in the serial stream by means of equations 4.1 and 4.2, both in the kill mask in order to activate the pixel cells and after this in the inject mask in order to inject the pixel cells selected. Then when the program finishes scanning all the levels of voltage for the pixel cells selected you need to change the selection of pixel cells. To do that we send a command to put a logic one in only the places in the kill/inject mask that we selected before to be logic zero. After that we send the new group of pixel cells to be characterized. By doing in this way we save a lot of time because the other way to do it is to load a complete file to disable all the pixel cells and later on to send to program the new group of pixel cells. To load a file the data generator takes more or less 60 second to do it, and this represents a lot of time when repeated at every change in the choice of cells. The process we designed means that we do not need to load a file. We just send commands by means of the GPIB interface to disable the previous group of pixel cells and the data generator takes only milliseconds to do that.

The levels of charge must be chosen to bracket the voltage threshold. To do this we make a preliminary scan over the chip to determine the optimal range for the test. For each test we fix Vth0 at a particular value. The step size over which we are doing the scan is 1mV. We have carried out these threshold uniformity and noise tests in four different regions of operation. For example for one of the regions we apply to the Vth0 terminal 2.05V and the levels of voltage will be 51 with a initial voltage of 130mV, and a final voltage of 180mV in steps on 1mV. When you applied 130mV to the pixel cells you will not see any hit, and with 180 mV you will have 500 hits in each pixel cell selected (assuming you inject 500 times).

A general flow diagram of the data acquisition system is shown as figure 4.7. And only to show how looks like one of the subroutines in figure 4.8.

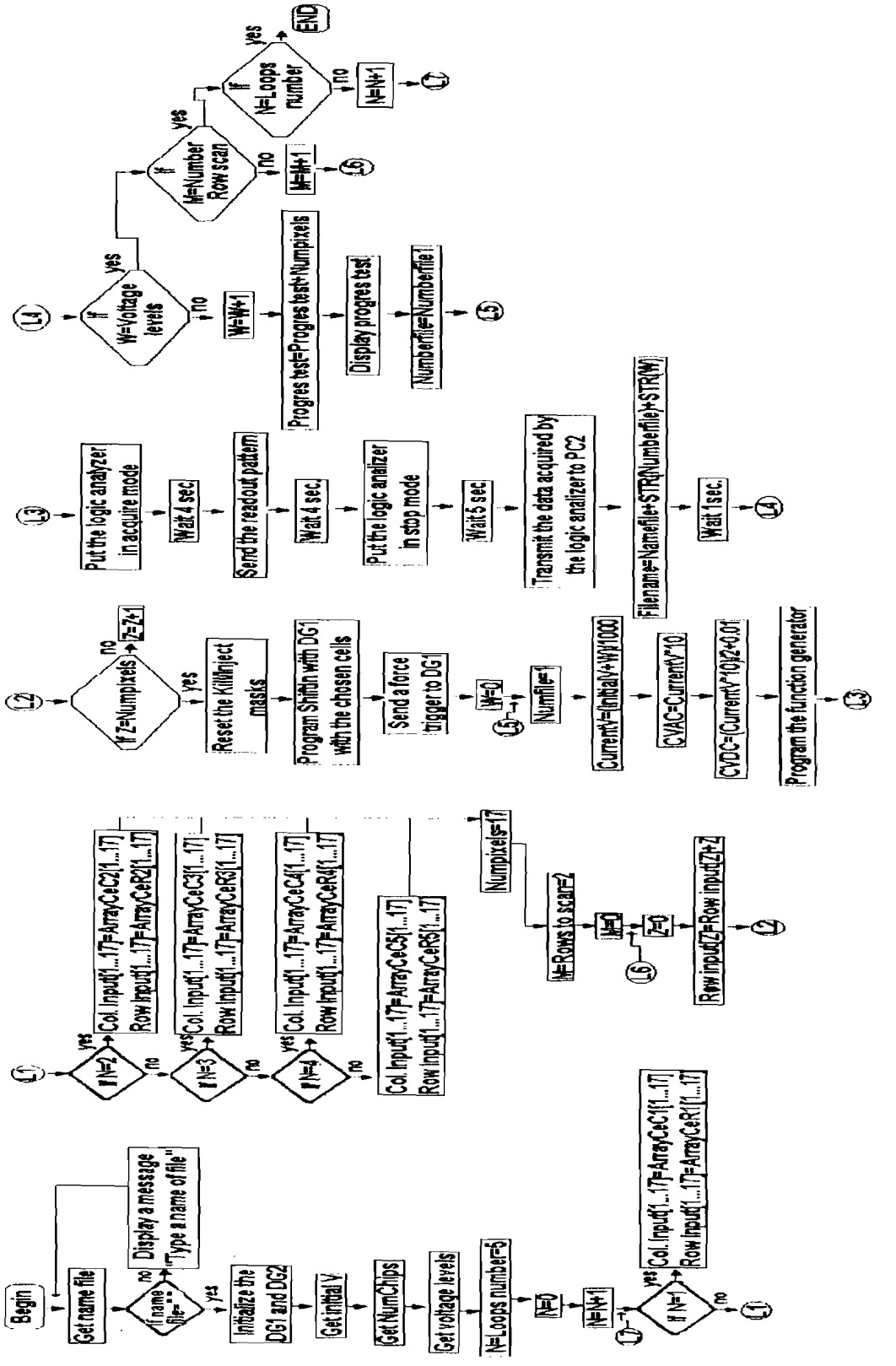


Figure 4.7 General flow diagram for the DAQ system.

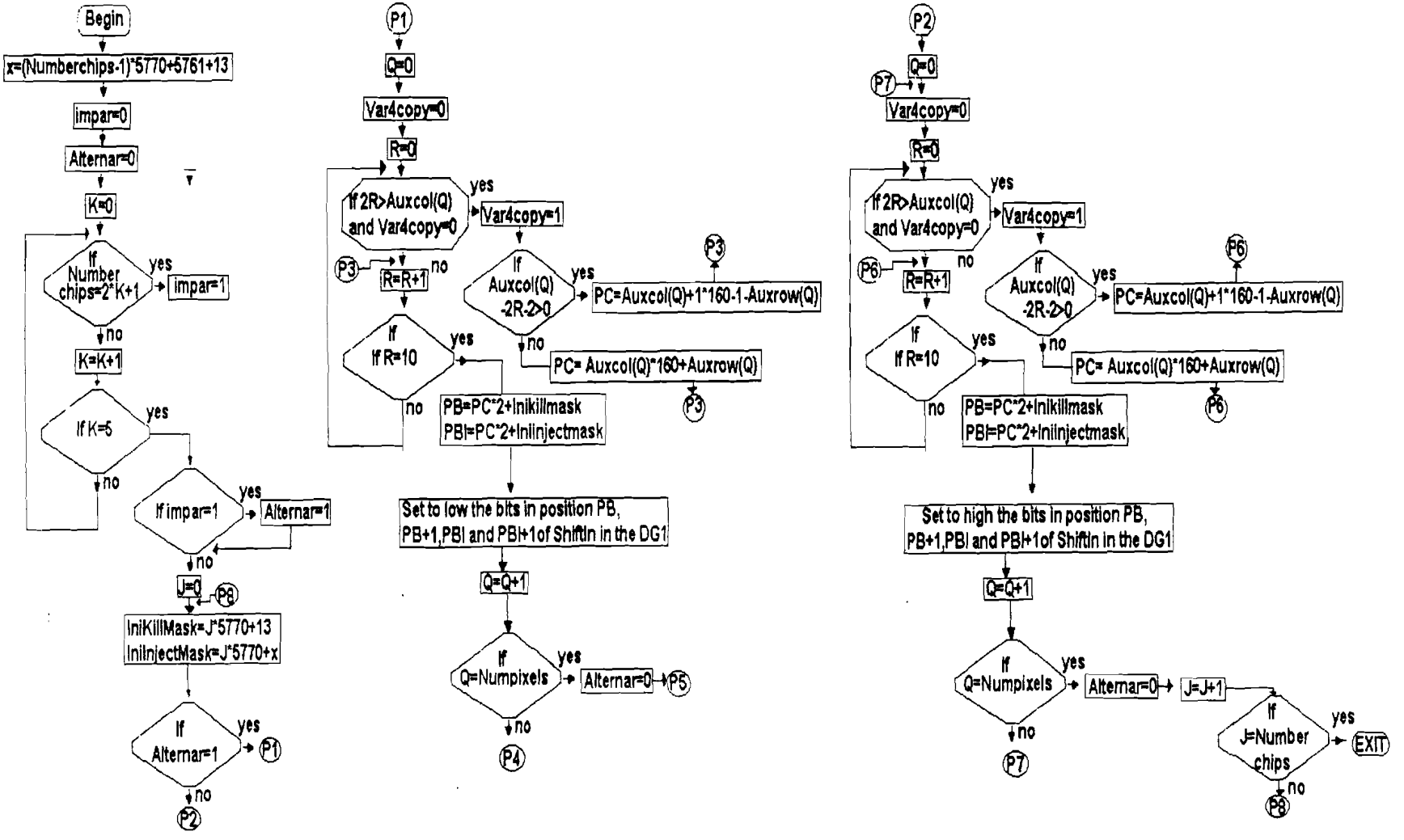


Figure 4.8 Flow diagram of the subroutine called 2FPIX1 conrtol reverse.

A picture of the user interface of the data acquisition system is shown as figure 4.9, in this we show the input data that the program needs to initiate data acquisition. These data are the following: The name of the group files that it will generate with the information that the MCM delivers, the number of chips in the module because with the software designed we can do test on the MCM by adding the chips. Then in our case we performed the test adding the FPIX1 chips one by one, the voltage levels for the test that you want to run, and the initial voltage. But the same software can also do characterizations with a single FPIX1 chip board. In that case you only need to put one in the input parameter called number of chips. An example of the code in G of the data acquisition system is shown as figure 4.10.

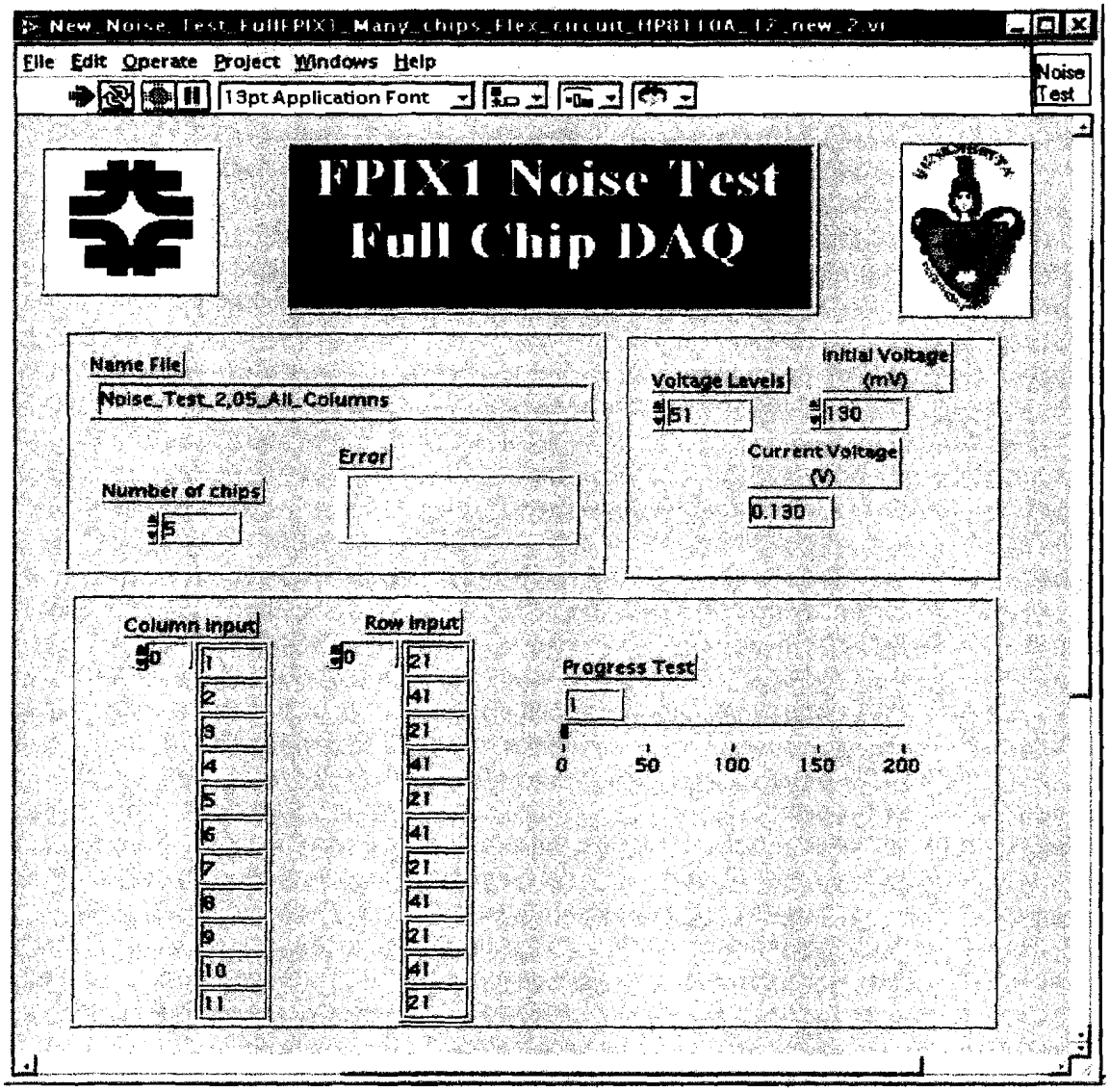


Figure 4.9 Picture of the user interface of the data acquisition system.

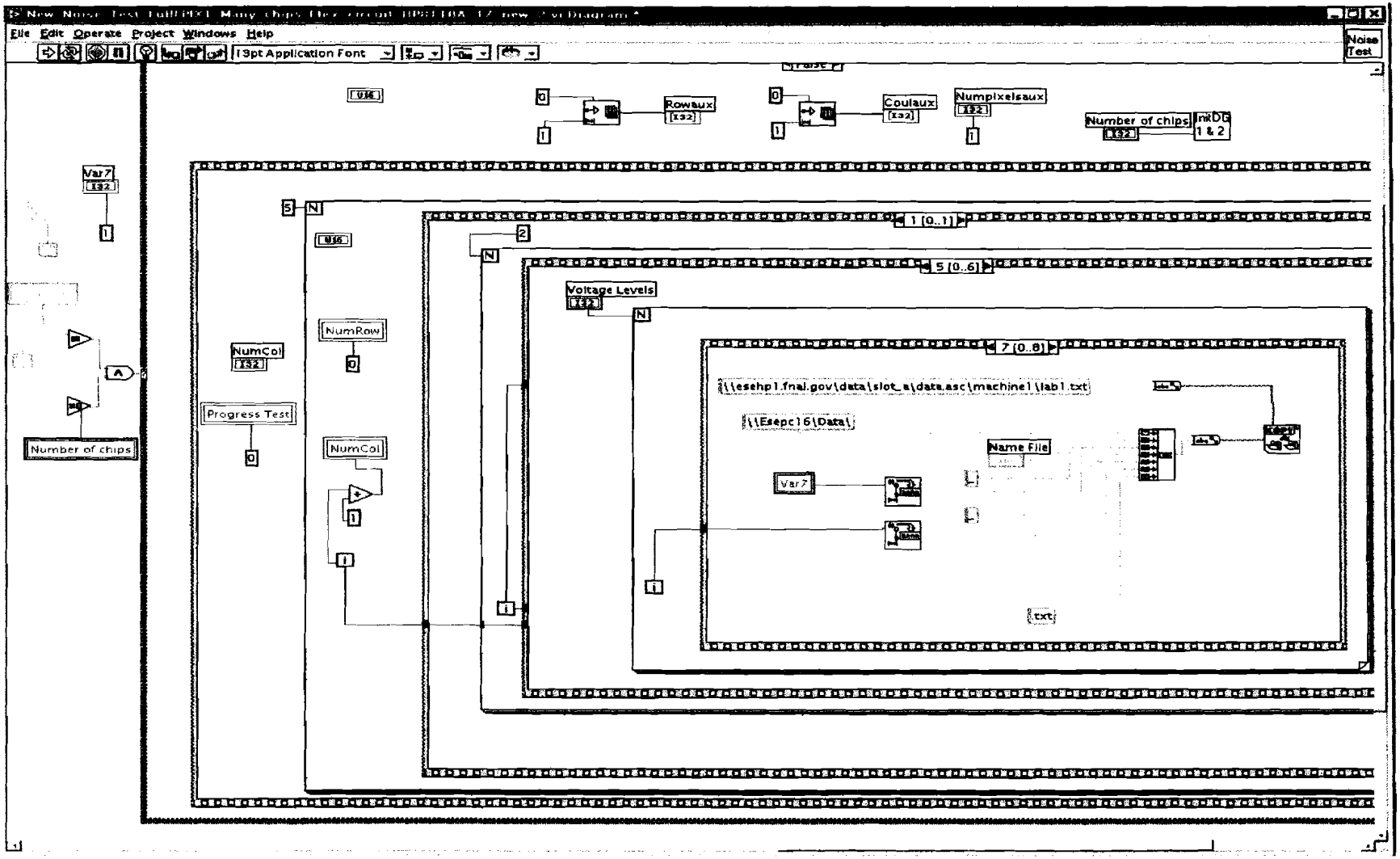


Figure 4.10 G code for the data acquisition system.

4.3 DATA ANALYSIS SYSTEM

PC2 processes the information collected to obtain the mean threshold with its corresponding standard deviation (threshold dispersion) and the mean noise with its corresponding standard deviation for each FPIX1 chip on the MCM. When PC1 begins to acquire data, PC2 waits until the transfer of information from the first 85 pixel cells finishes, which takes approximately 30 minutes. After this, both programs run at the same time. The data acquisition wait time is 35 minutes (adding an extra 5 minutes more to avoid any potential conflicts). The data processing program needs to wait for this time only once, because after the first transfer of data both programs can run simultaneously. The information processing takes exactly 30 minutes, so that the times are well matched. Thus, neither PC sits idle for an appreciable time during the test. While PC1 is acquiring the data from the current 85 pixel cells, PC2 is processing the information from the previous 85 pixel cells.

Information processing proceeds as follows: First, PC2 translates the binary code that the data acquisition system has acquired into a file that contains the ChipID number, the column number, the row number, the count of hits recorded above threshold and the voltage at each step, as well as the count of total hits (charge injections). Then it calculates efficiency for each pixel cell by dividing the number of counts recorded over the total number of hits. The discriminator output in the front-end of the FPIX1 chip is characterized by measuring its response as a function of the charge injected. Although only a single Vth0 is input to all 2880 cells on a chip, the value of the charge that triggers the discriminator may vary from cell to cell. This represents the threshold dispersion. The electrical charge injected can be calculated using the relationship between voltage and charge produced on the input capacitor. Since we have a capacitive load, C, of 6fF, the charge, Q_{inject}, is

$$Q_{inject} = VppC = Vpp \times 6 \times 10^{-15}\, coulombs, \tag{4.3}$$

where *Vpp* represents the voltage peak-peak delivered by the SRS DS345 function generator. The number of electrons that correspond to 1mV can be obtained by means of the following relationship:

$$(Q_{inject})e = \left(\frac{VppC}{e^-}\right)e = \left(\frac{Vpp \times 6 \times 10^{-15}}{1.6 \times 10^{-19}}\right)e, \tag{4.4}$$

with the result that

$$1mV \rightarrow 37.5e. \tag{4.5}$$

Because of the inherent noise of the electronics, the efficiency as a function of *Vpp* has the properties of an error function. For each pixel cell, a fit is performed to obtain the error function that best describes the experimental points. LabVIEW has a library of

functions including one that performs a fit using the nonlinear Levenberg-Marquardt (Lev-Mar) method to determine a nonlinear set of coefficients which minimize chi-square [19]. First, PC2 plots the measured efficiency versus voltage *Vpp* (charge injected). The initial values of the threshold and noise are obtained using the data. The threshold value, Vth_{exp}, is chosen to be the first value of the voltage that has greater than 50% hit efficiency. For the noise value, PC2 finds the first voltage for which the hit efficiency is greater than 81.5% and subtracts the value for 50% hit efficiency, which yields σ_{exp}. These experimental values are used as the initial guess coefficients for the nonlinear Lev-Mar Fit technique, since the fits will converge more quickly the closer the initial values are to the solutions. The nonlinear function needs to be specified, i.e., the relationship that describes the error function for each cell as a function of *Vpp*. For each point, *Vpp*, along the curve:

$$errf = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{Vpp} e^{-\left[\frac{(V-Vth)^2}{2\sigma^2}\right]} dV \tag{4.6}$$

This technique provides two ways to calculate the Jacobian; i.e., the partial derivatives with respect to the coefficients, needed in the algorithm. These methods are:
Numerical calculation: Uses a numerical approximation to compute the Jacobian.
Formula calculation: Uses a formula to compute the Jacobian. For this one needs to specify the Jacobian functions, which are the partial derivatives of the error function with respect to σ and with respect to *Vth*. These functions are:

$$\frac{\partial errf}{\partial \sigma} = \left(\frac{(V-Vth)^2 - \sigma^2}{\sqrt{2\pi}\sigma^4} \right) \int_{-\infty}^{Vpp} e^{-\left[\frac{(V-Vth)^2}{2\sigma^2}\right]} dV \tag{4.7}$$

$$\frac{\partial errf}{\partial Vth} = \frac{(V-Vth)}{\sqrt{2\pi}\sigma^3} \int_{-\infty}^{Vpp} e^{-\left[\frac{(V-Vth)^2}{2\sigma^2}\right]} dV \tag{4.8}$$

The program repeats the above procedure for all of the pixel cells being tested. When the information for all the pixel cells has been analyzed, the results are output in five files, each of which contains the final information for all cells on one chip in the MCM, i.e., the column number, the row number, the best fit *Vth* in units of volts and electrons, and the best fit σ (noise) in units of volts and electrons. We then run a second program to histogram these quantities for the 170 cells on each of the five chips and to obtain the Gaussian curves that best fit these histograms for each file. We thus obtain the mean threshold, the sigma threshold (threshold dispersion), the mean noise and also the noise dispersion, all in units of volts and electrons for the 170 cells on each FPIX1 chip. These results represent the characterization of the chips on the module.

A general flow diagram of the data processing system is shown as figure 4.11

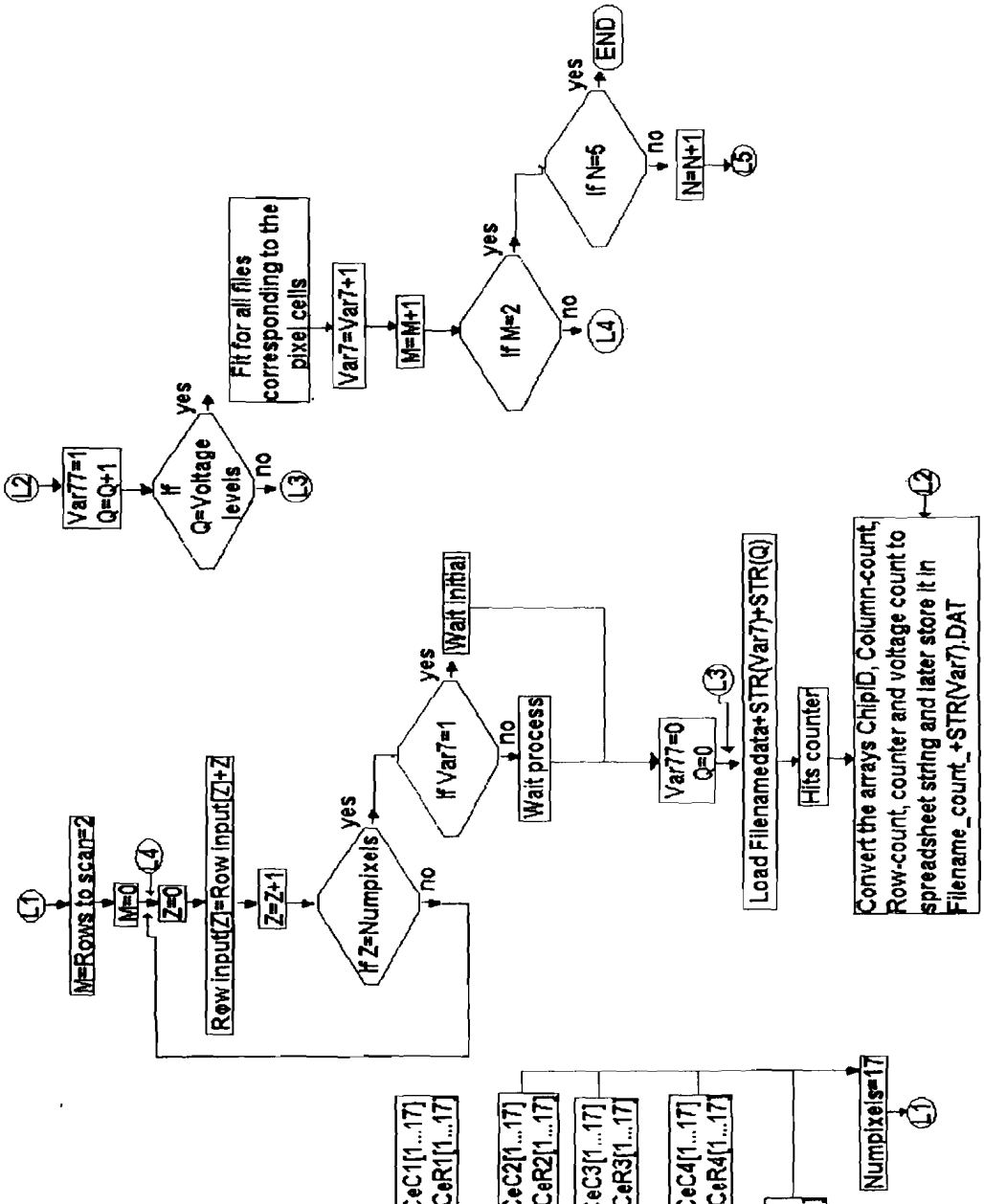


Figure 4.7 General flow diagram for processing data system.

A picture of the user interface of the data analysis system is shown as figure 4.12, in this we can see how the user interface looks and the information that we need to run the program. This includes the name of the files that the data acquisition system is acquiring, the name of the file that will contain the results on each chip on the MCM, the time that the program needs to wait while PC1 is acquiring the data, the time that the program needs to wait to begin to process the information acquired. Both these times need to be given in minutes. Also you need to provide to the program the voltage levels of the test that you are running, the initial voltage value in mV and the number of hits or injections with which you are doing the test. The group of seventeen pixel cells for which you are processing the information acquired by PC1 appears in the window. As figure 4.13, we show the G code for the data processing system.

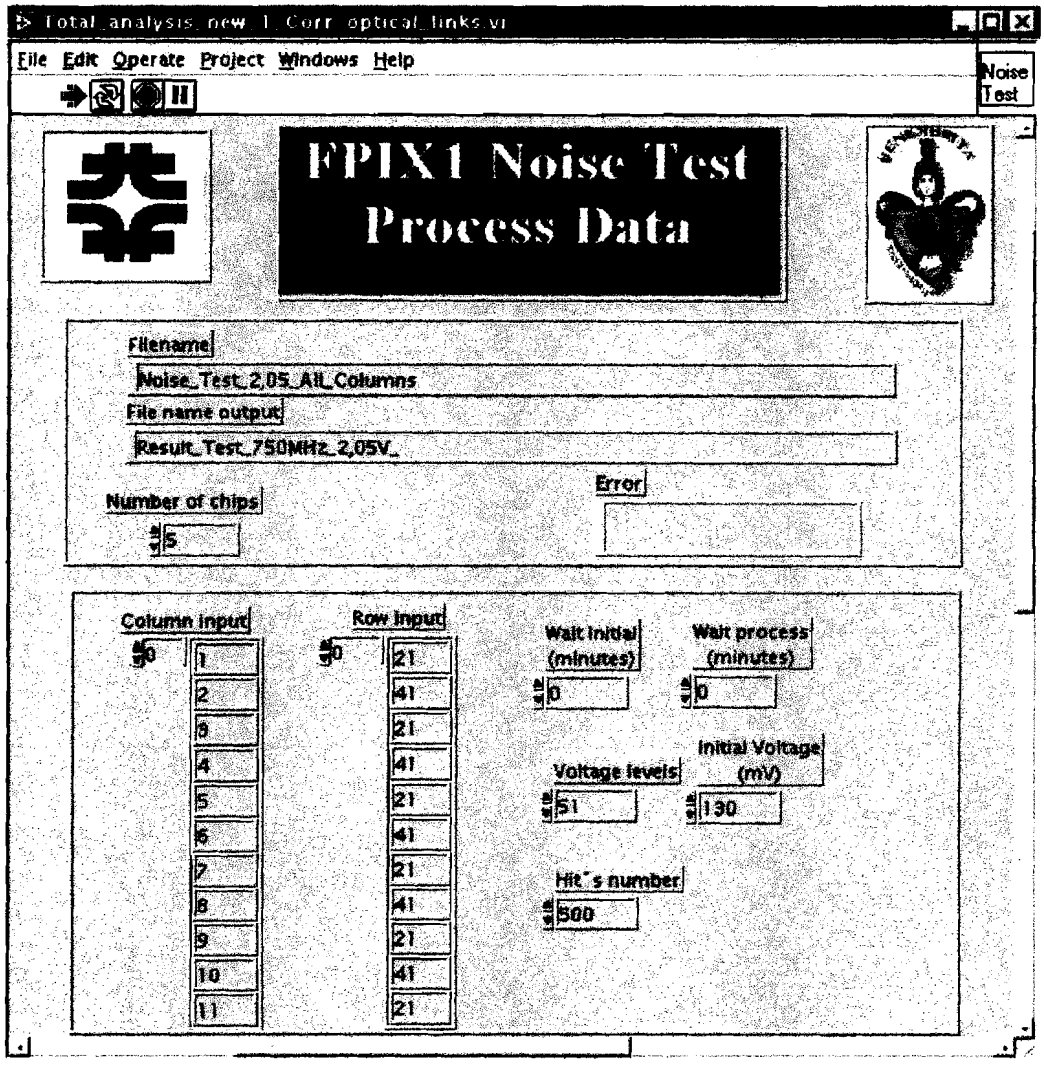


Figure 4.12 Picture of the user interface for the data analysis system.

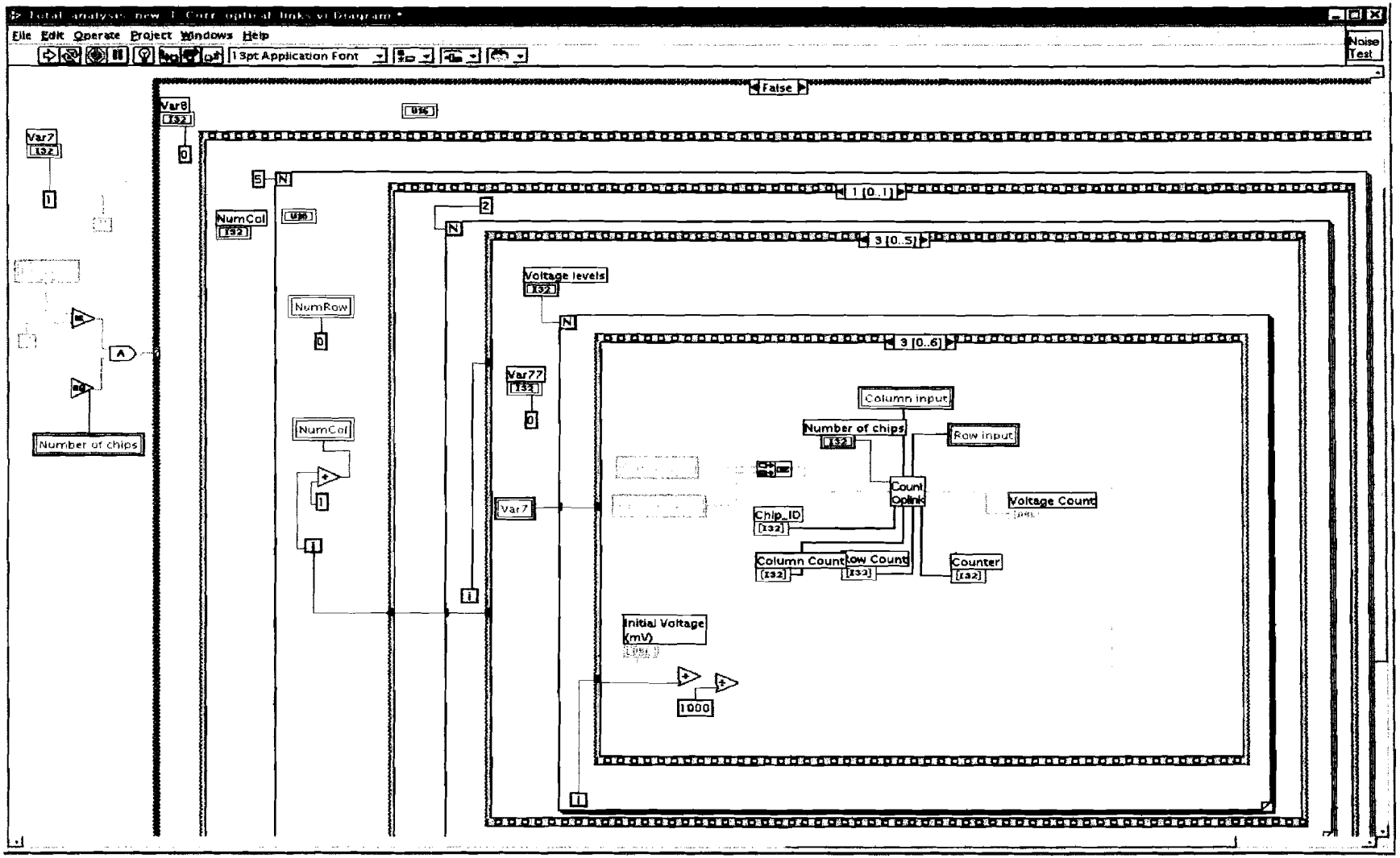


Figure 4.13 G code for the data processing system.

We optimize the software described before, with all the modifications, we were able to reduce the running time of the system. We summarize all the time reductions in the following table 4.2

Test Setup	Instruments	Acquisition time and data process (Hours)
1	1 data generator, loading two files 1 logic analyzer (Transfer of the data by means of the GPIB interface) 1 computer	157.2 (One chip)
2	2 data generators. One load the initialization and the kill/Inject masks and the other load the readout 1 logic analyzer (Transfer of the data by means of the GPIB interface) 1 computer	118 (One chip)
3	2 data generators. . One load the initialization and the kill/Inject masks and the other load the readout 1 logic analyzer (Transfer of the data by means of the GPIB interface) 1 computer	85.4 (One chip)
4	2 data generators. . One load the initialization and the kill/Inject masks and the other load the readout 1 logic analyzer (Transfer of the data by means of a LAN) 1 computer	13.2 (Five chips)
5	2 data generators. . One load the initialization and the kill/Inject masks and the other load the readout 1 logic analyzer (Transfer of the data by means of a LAN) 2 computers	6.2 (Five chips)

Table 4.2 Test stand time optimization.

Figure 4.14 shows a plot of the data contained in table 4.2.

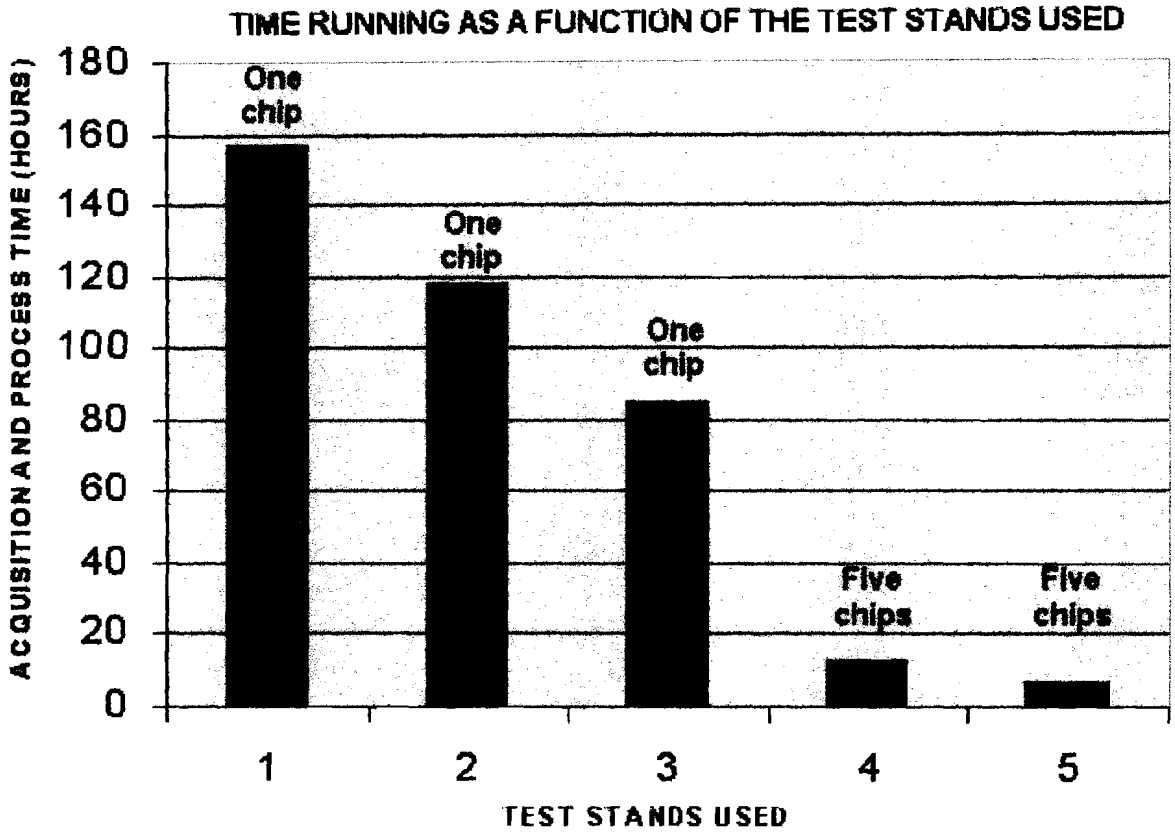


Figure 4.14 Optimization of the test stands

4.4 GENERAL FLOW DIAGRAM

A flow diagram of both programs, i.e., the data acquisition program and the data processing program is shown as Fig. 4.15. In this we show the logic of the complete system, the action of the two computers, and their role in the complete system. This is to make the understanding of the software more clear and easy.

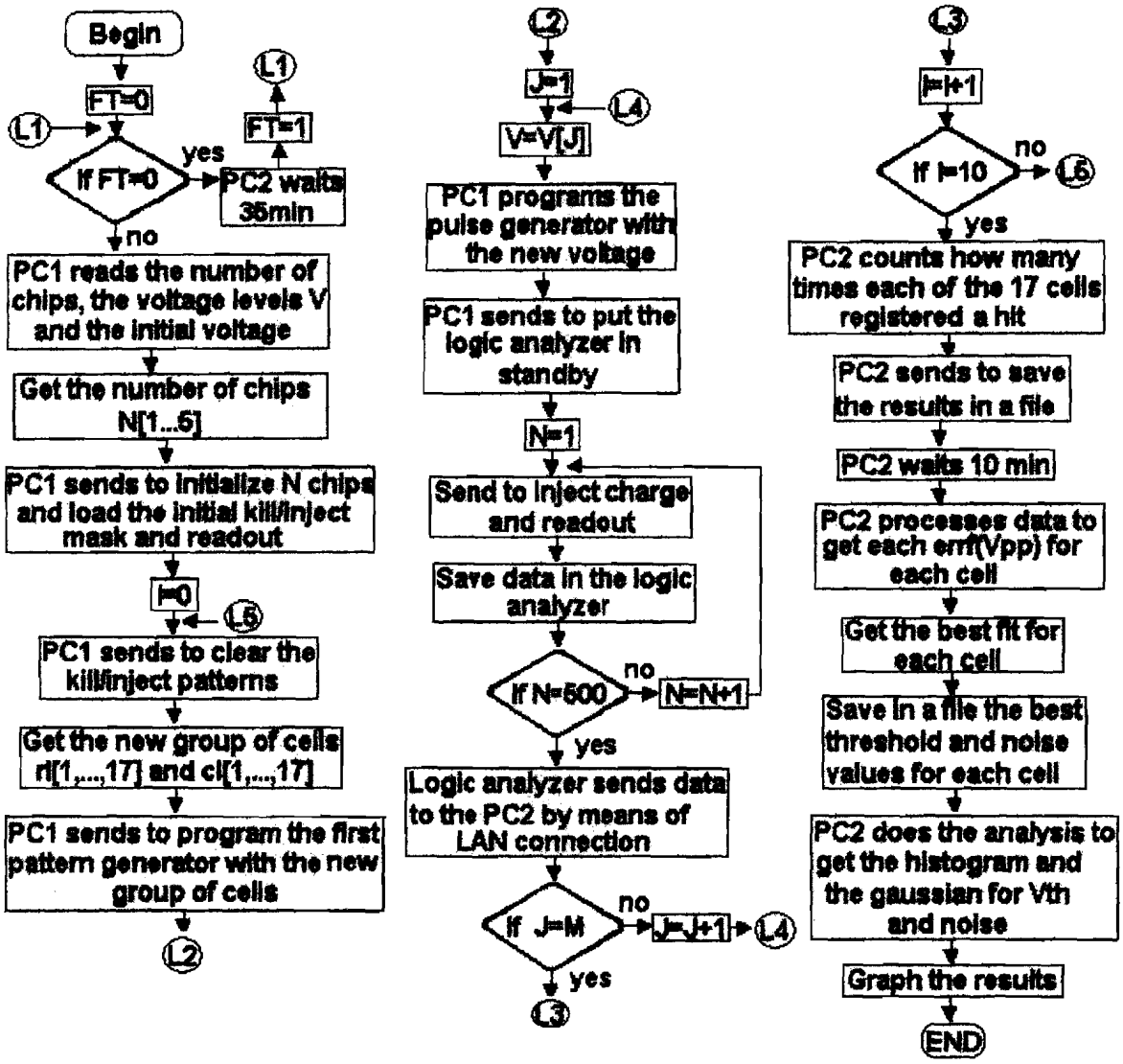


Figure 4.15, General flow diagram of the software designed.

4.5 CONCLUSIONS

In this chapter we described the software designed to characterize the FPIX1 chip, and the multi chip module, called MCM. This software is very versatile because with it you can perform all the tests needed to have a complete characterization of a single FPIX1 chip or a MCM. In the following chapter we will show the results of each of the tests that we developed to do the characterization.

CHAPTER FIVE

RESULTS OF THE CHARACTERIZATION

In this chapter we will show all the results obtained from the characterization of the single FPIX1 chip and for the MCM. We will see the results on the bad pixel map test, which we performed first for the single FPIX1 chip to gain experience handling this type of chips. We will show the first results of the characterization on a single FPIX1 chip and the setup that we used for that challenge. Once the multi chip module arrived at the laboratory to be characterized, we will describe how we added FPIX1 chips one by one to it and the results of each chip. These test were performed without a sensor. We also describe the result of the MCM test with a sensor. Later on we will show the result of the reproducibility test and the hit test. These results represent the characterization of the FPIX1 chip and the MCM.

5.1 BAD PIXEL MAP

We performed an initial test called the bad pixel map test, using a very low threshold on all cells to see which cells were working. This test revealed that the last 60 rows in each column did not record hits. This is a feature of the layout that will be corrected on the next submission. Furthermore, the first column on these prototype chips was made to have outputs that could be visualized on an oscilloscope and could not be read out by the EOC logic. A sketch of this result is shown in Fig. 5.1.

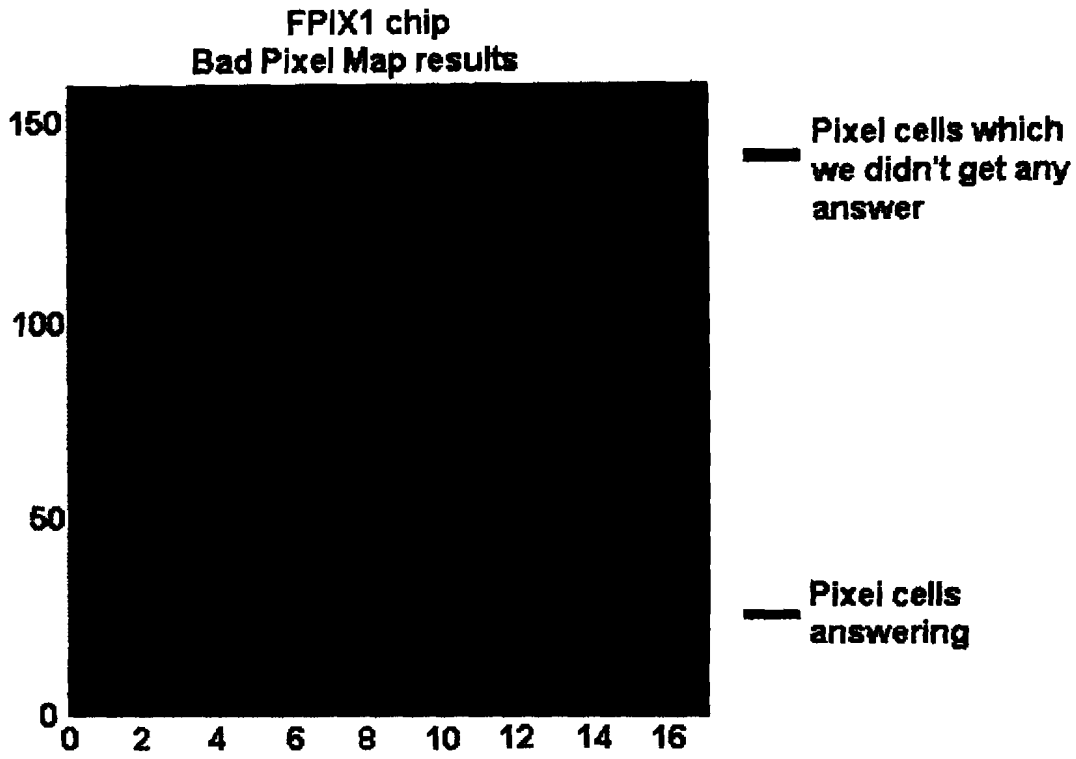


Figure 5.1 Result of the bad pixel map test.

5.2 RESULTS OF THE SINGLE FPIX1 CHIP

In response of a pulse of injected charge, the FPIX1 chip delivers its information in the following way: Several frames with extraneous data while the logic analyzer waits for the trigger. This trigger fires when the logic analyzer senses a logic one in the data valid output (**Data16**) and at the same time a logic one in the control/data output (**Data 15**).

When the trigger is present the logic analyzer begins to acquire useful information. After all valid information ends we again see 10 frames with extraneous information, then another frame with useful information and so on. At the beginning of the valid data the frame contains a control word with a chip ID number, and a BCO number. The following 5 frames are data words that contain the ADC number, the column number and the row number. An illustration of this information is shown in fig. 5.2.

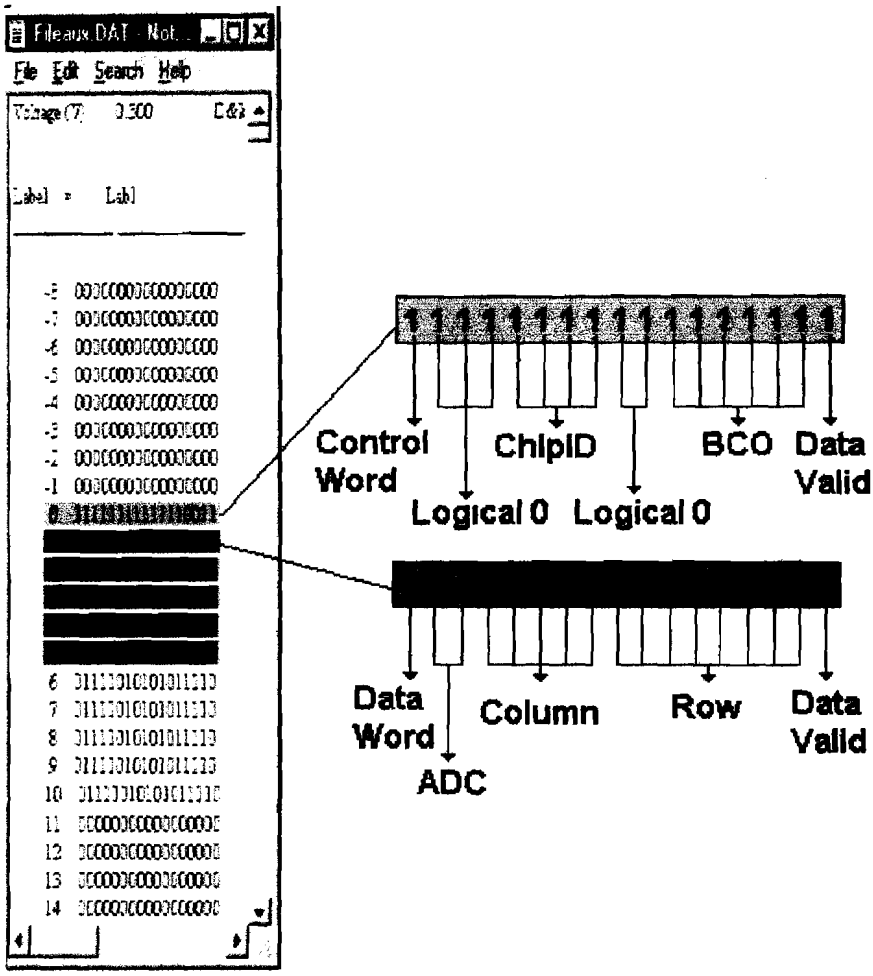


Figure 5.2 Data delivered from the single FPIX1 chip.

With all the information delivered by the FPIX1 chip, we performed the threshold and noise test over two regions of operation. The first one was about 9000e, which corresponds in this specifically chip to Vth0 of 1.95V. The second one about 2000e-3000e, corresponds to Vth0 of 2.25V. We did the first one to help us to understand the chip, and the second one because it is the region that the BTeV experiment will use. This corresponds to approximately 10% of the 25000e signal expected for a minimum ionizing particle. To do the test in these two regions we change the range of voltages. For the first region we worked over three intervals, which are, 80mV to 200mV in steps of 10mV, 200mV to 300mV in steps of 1mV and 300mV to 400mV in steps of 10mV. This means a total of 123 levels of voltage. For the second region we worked over one interval, it was from 30mV to 90mV in steps of 0.33mV representing 181 levels of voltage. All these tests were performed providing a pulse delivered by the function generator SRS DS345. The shape of this pulse is shown as figure 5.3 the characteristics of this pulse needs simulate as close as possible the response of the sensor to a minimum ionizing particle. We need to try to collect only electrons with the pulse, the rise time of this pulse needs to be very large and the fall time very fast. Our pulse has the following characteristics: it has a rise time of approximately $5\mu s$ and a fall time of 40ns.

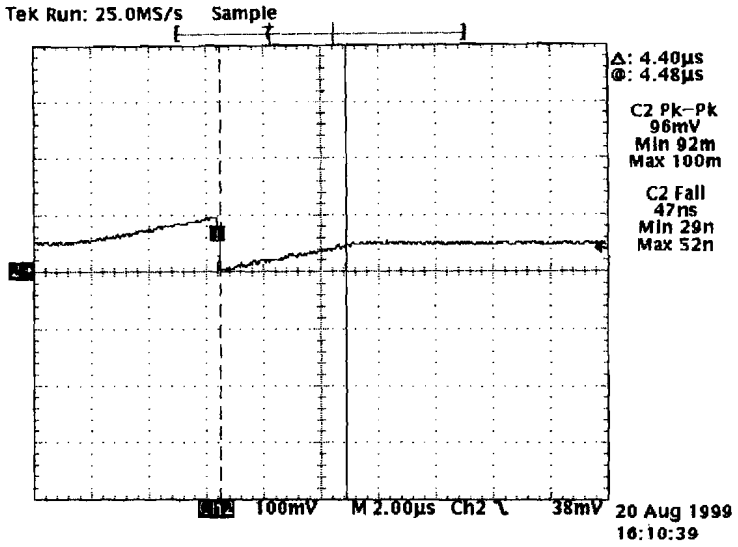


Figure 5.3 Shape of the pulse delivered by the function generator SRS DS345.

In figure 5.4 we show the efficiency versus *Vpp* for a single pixel cell on the chip and the error function that results from the fit to the data.

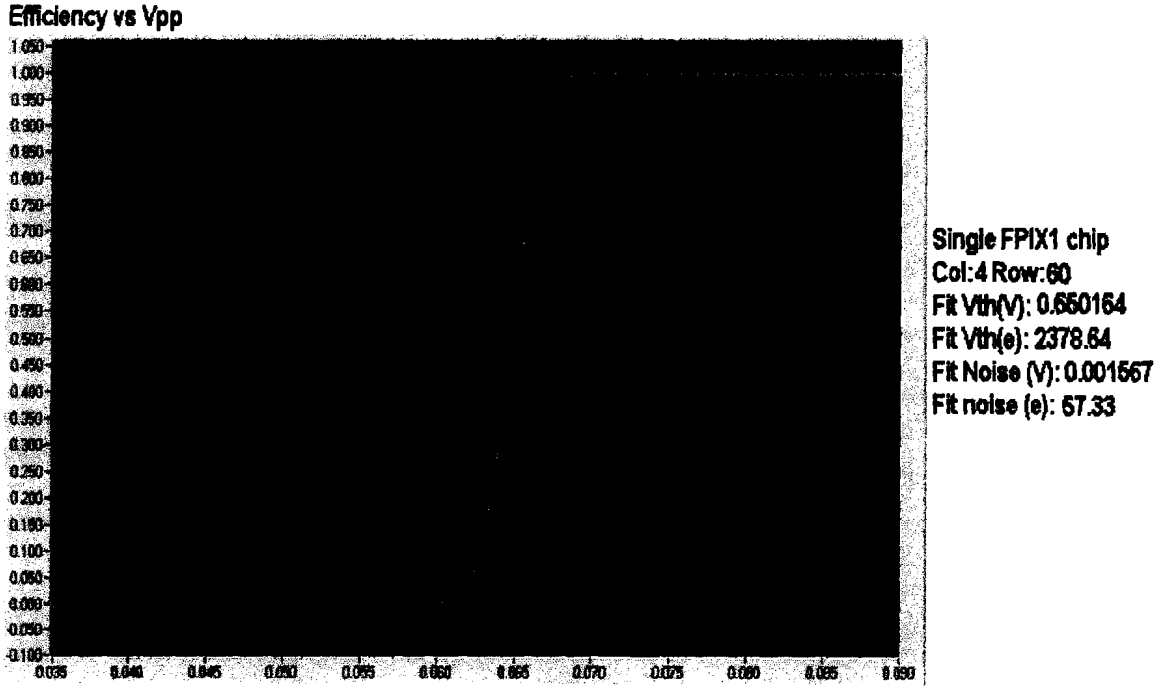


Figure 5.4 Error Function of the pixel cell (4,60) of the single FPIX1 chip.

The final results after processing all the information from the 170 pixel cells are summarized in table 5.1, and the histogram and the corresponding gaussian curve on one of the regions of operation is shown as figure 5.5 and 5.6.

Single FPIX1 chip ChipID = 0	Vth0 = 1.95V	Vth0 = 2.3V
Mean threshold (e-)	9132.97	2355.22
Sigma threshold (e-)	291.61	329.63
Mean noise (e-)	58.61	53.89
Sigma noise (e-)	7.07	6.54

Table 5.1 Final result for the single FPIX1 chip.

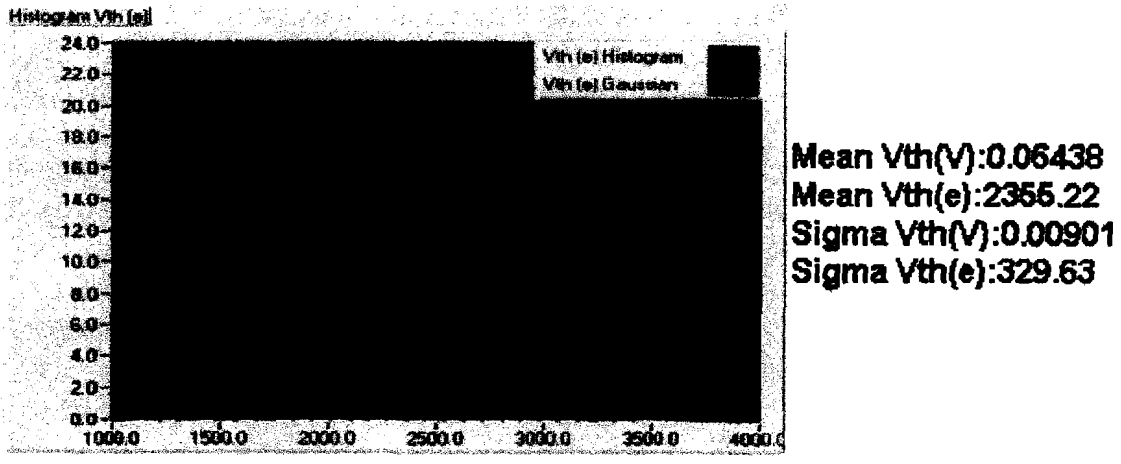


Figure 5.5 Final results for voltage threshold single FPIX1 chip.

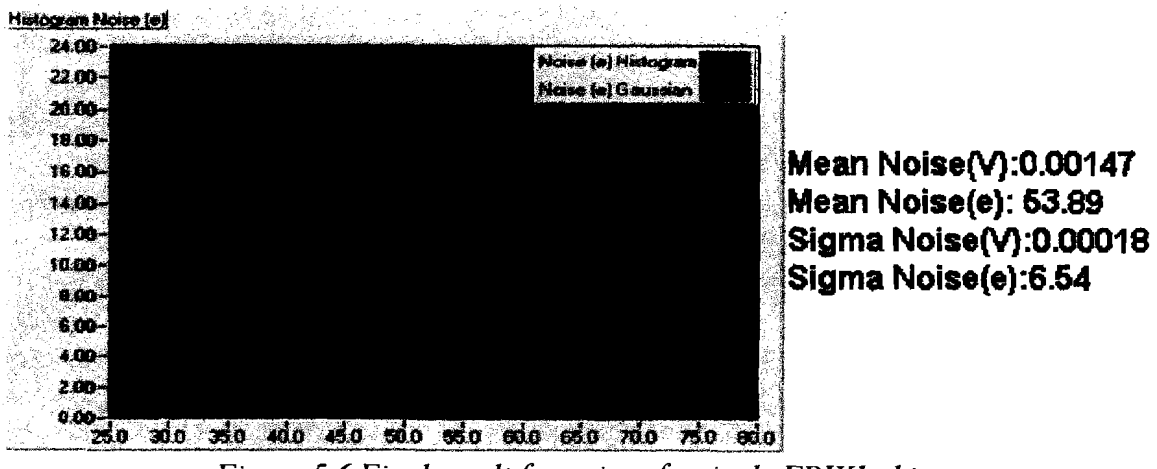
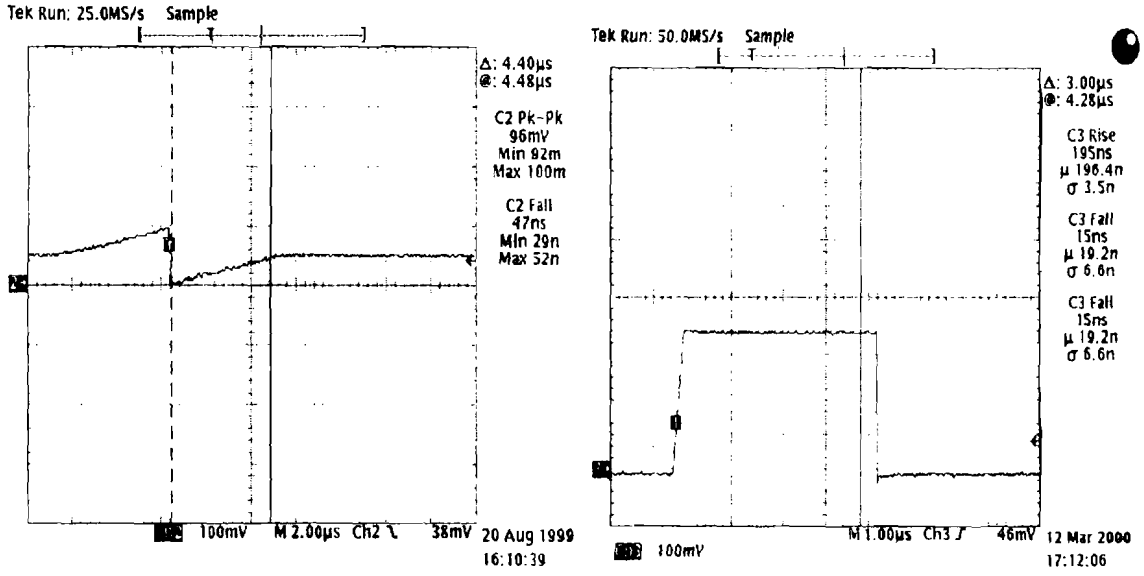


Figure 5.6 Final result for noise of a single FPIX1 chip.

5.3 MULTI CHIP MODULE (MCM) WITHOUT SENSOR, THRESHOLD UNIFORMITY AND NOISE TEST

This section reports all the results of the characterization on the MCM for each of the FPIX1 chips on it. We did the characterization for two regions of operation about 7000e and about 1000e-2000e, setting the Vth0 terminal to two different values. These values are 1.95V and 2.25V.

Because this test was developed without mounted the chips to a sensor, we need to provide to the chip a signal that models the action of the sensor. We used two different pulse generator for the test in order to try to do this as well as we could. One of them was the function generator SRS DS345. The pulse generated by this instrument starts with a ramp with a rise time of approximately $5\mu s$ and ends with a fall time of approximately 40ns. The second one was a pulse generator HP8110A, which generated a square pulse with a rise time of approximately 200ns, a $5\ \mu s$ flat top, and a fall time of 15ns. We wanted to verify if the shape of the pulses has a influence on the results of the characterization i.e., if the results change if you use an SRS DS345 form when you use an HP8110A. A picture of each of these pulses are shown in figures 5.7a and 5.7b.



a) b)

Figure 5.7a): Waveform of the pulse delivered by the function generator SRS DS345
Figure 5.7b): Waveform of the pulse delivered by the pulse generator HP8110A.

The setup that we used to get all the results for the MCM is shown in figure 5.8, we show all the instruments that we use to do the characterization and a photograph of the computers used to acquire and to process and analyze all the information delivered by the FPIX1 chips. Also we show a photograph of the flex circuit containing the five FPIX1 chips connected in daisy-chain and their order on the board.

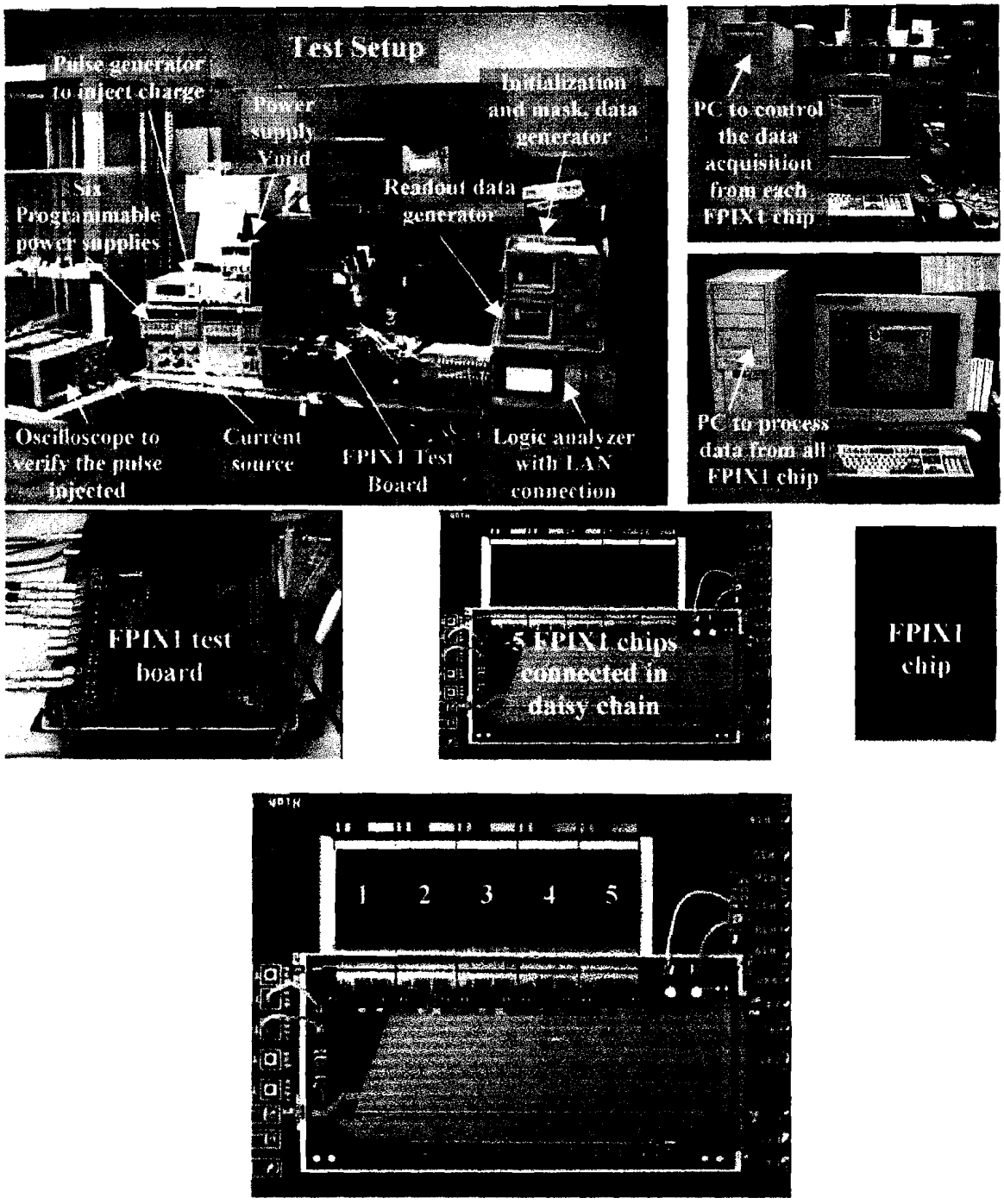


Figure 5.8 test setup used to characterize the MCM. We are showing a photo of the PC's used in our experimental arrange, also showing the order of the FPIX1 chips on the flex circuit.

First when we had only one chip (chip number 5) in the MCM, we did the test over 170 pixel cells charge injecting each cell 1000 times. These results were obtained with the function generator SRS DS345. After that we put another chip in the module now called chip number 4 and chip number 5. We again performed the threshold and noise test over 170 pixel cells, the same pixel cells as when we had only one chip. We charge injected each pixel cell 750 times in both chips. We did the tests with two different pulse generators. The first one was the function generator SRS DS345, and the second was a pulse generator HP8110A. Then we added another chip to have three chips in the module. This time we only performed the threshold and noise uniformity test with the pulse generator HP8110A. We performed the threshold and noise uniformity test over the same 170 pixel cells that we tested when we had one and two chips in the module. We charge injected each cell 750 times in the three chips.

We then added another chip connected in daisy chain on the module, for a total of four chips on the module. We again performed the threshold and noise uniformity tests only with the pulse generator HP8110A. I found that there were missing data in the readout of the daisy-chain. For example when we had only chip number 5 on the MCM all the columns output hit data. When we put the second chip (chip number 4) on the MCM, the data from column one on chip 5 was missing, although all data from chip number 4 was received. And with three chips I found that chip number 3 output all the expected data, but chip number 4 was missing the first column and chip number 5 was missing the first 2 columns. Then, when we had four chips, chip number 5 was missing the first 3 columns, chip number 4 was missing the first 2 columns, chip number 3 was missing the first column and chip number 2 readout all columns. We can see this effect in the data output shown below in figure 6.9.

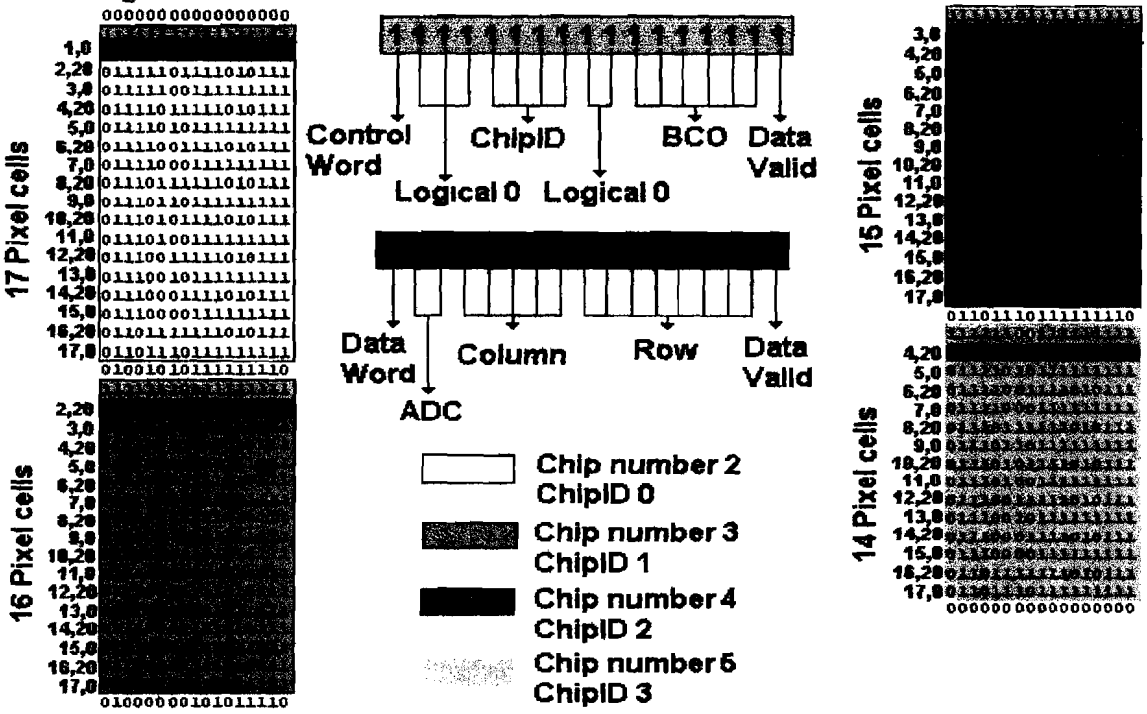


Figure 6.9. Data acquired with the logic analyzer from 4 chips in the module shown the missing column problem.

After we acquired data from 4 FPIX1's connected in daisy chain, we put the last chip on this module, the chip number 1. The missing column problem was still present. In Table 5.2 we show the complete results that we obtained adding chips one by one.

Chip's number	Chip	Mean Threshold (e)		Sigma Threshold (e)		Mean Noise (e)		Sigma Noise (e)	
		VTh0=1.95V	VTh0=2.25V	VTh0=1.95V	VTh0=2.25V	VTh0=1.95V	VTh0=2.25V	VTh0=1.95V	VTh0=2.25V
1	5	7376	1980	288	367	48	56	10	8
3	3	6741	1931	135	278	31	56	5	13
	4	6793	2049	128	255	27	46	5	8
	5	6830	1972	139	269	29	51	5	11
4	2	5767	681	367	258	28	39	7	6
	3	5657	555	237	241	29	40	7	10
	4	5706	662	236	237	25	35	5	6
	5	5856	626	228	277	26	38	5	8
5	1	6102	1127	280	296	26	40	6	9
	2	6235	1346	232	232	24	34	4	7
	3	6161	1300	226	229	25	34	5	8
	4	6143	1342	224	195	23	30	7	6
	5	6245	1238	209	248	23	37	5	10

Table 5.2. Results of the characterization of the MCM when we were adding one by one chip.

The software designed to characterize the MCM yields the threshold uniformity and noise dispersion for the five FPIX1 chips on the module in 5 hours of running with 500 hits (charge injections) per cell at each of 60 values of *Vpp*. The decision to limit the number of charge injections to 500 was based on the hits studies on the MCM to be described in section 5.5. The 5 hours of running could be reduced by as much as a factor of five by using PCs with faster processing speeds that are already available in the marketplace to process the information. For example if you used a PC PentiumIII with 750MHz processor speed, 128MB in RAM memory, the time would decrease considerably from 5 hours to 1 hour and 45 minutes to get all the results of the characterization. The time required for data acquisition and analysis has been made much shorter by taking advantage of the flexibility of the LabVIEW programming environment and by using the LAN connection to transfer the data from the logic analyzer to a second computer for analysis.

PC2 processes a very large amount of information. The raw data for 850 pixel cells (170 cells times five chips) is organized into 10 files, each containing the information for a group of 85 pixel cells. An individual file contains 931 KB of data. In a particular characterization we have to scan 60 levels of voltage in order to cover the full range. The total information acquired is obtained by multiplying the size of each file by the total number of files acquired, which in this case is 600 files. Thus, the program processes a total of approximately 550 MB of information to carry out the characterization in each range of operation.

Fig. 5.10 shows the efficiency versus *Vpp* for a single pixel cell and the error function that results from the fit to the data. Figures 5.11 and 5.12 show the histograms of the threshold and noise dispersion and curves corresponding to the Gaussian fits for one chip mounted in the MCM.

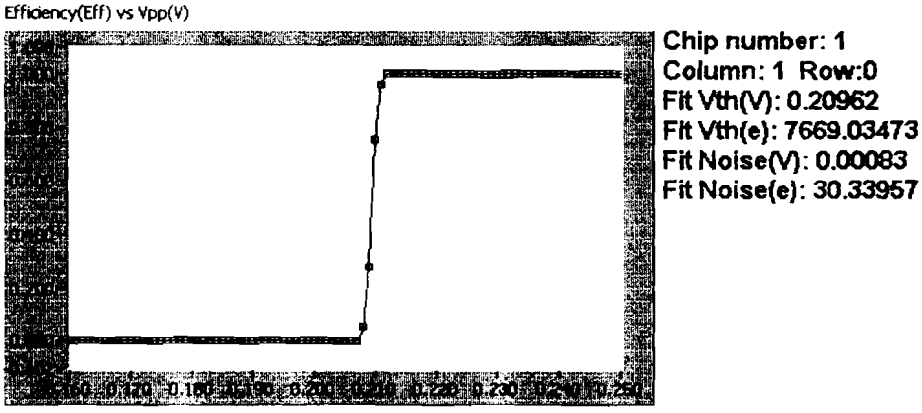


Figure 5.10. Error function from one pixel cell on one FPIX1 chip.

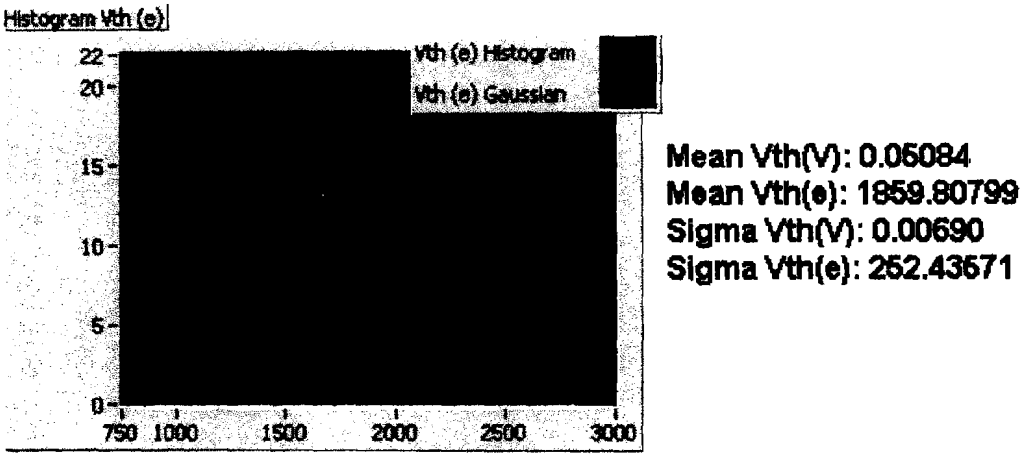


Figure 5.11. Voltage Threshold distribution in the chip number 1.

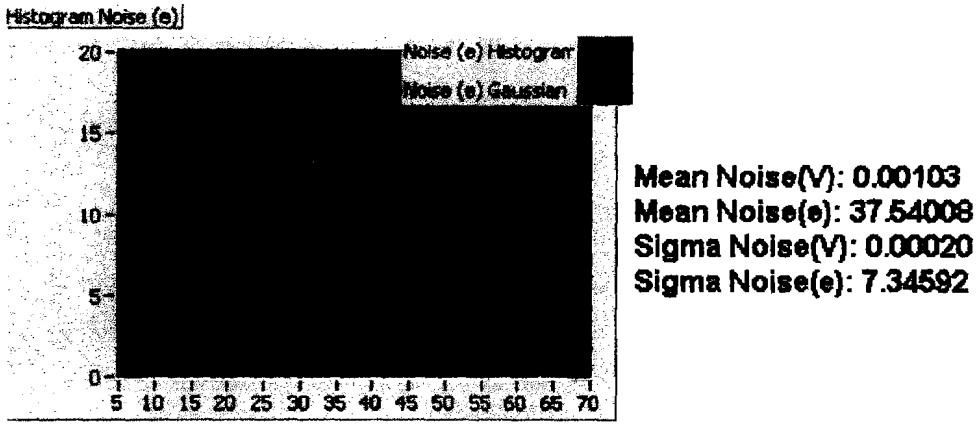


Figure 5.12. Noise distribution in the chip number 1.

We can graph the results obtained to see the differences in mean threshold, sigma threshold, noise, and sigma noise from chip to chip. We can see the variation of these values as we add chips to the flex circuit one by one we can also see the influence on the results of using two different pulse shapes for the charge injection in the following figures. The mean threshold variation can be seen in figure 5.13, and the changes in sigma threshold, noise and sigma noise in the figures 5.14, 5.15, and 5.16. Each figure shows data for one of the five chips. Each curve on the figure shows the result for a different total number of FPIX1 chips on the MCM.

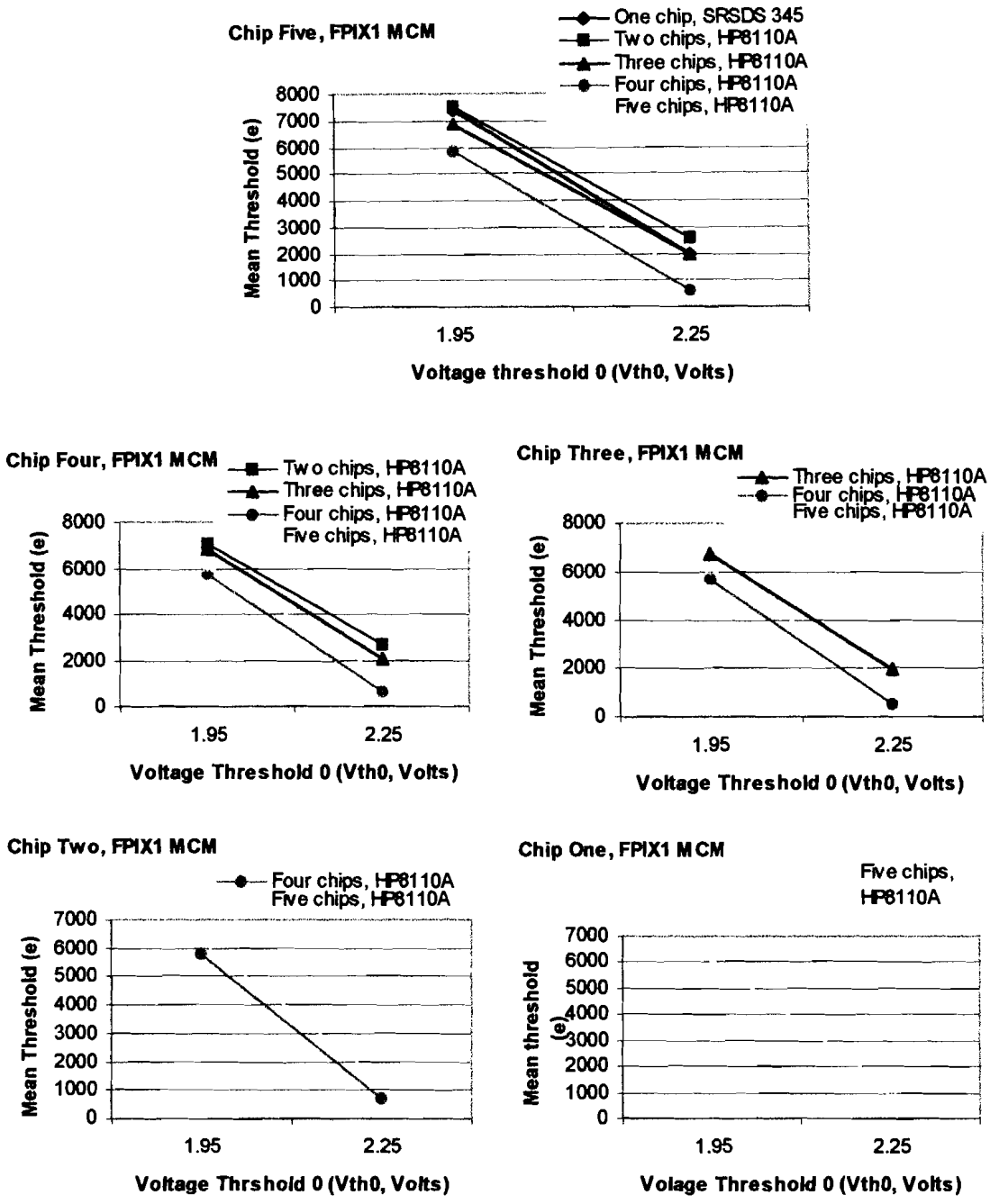


Figure 5.13. Mean threshold comparison when we were adding one by one FPIX1 chip on the MCM.

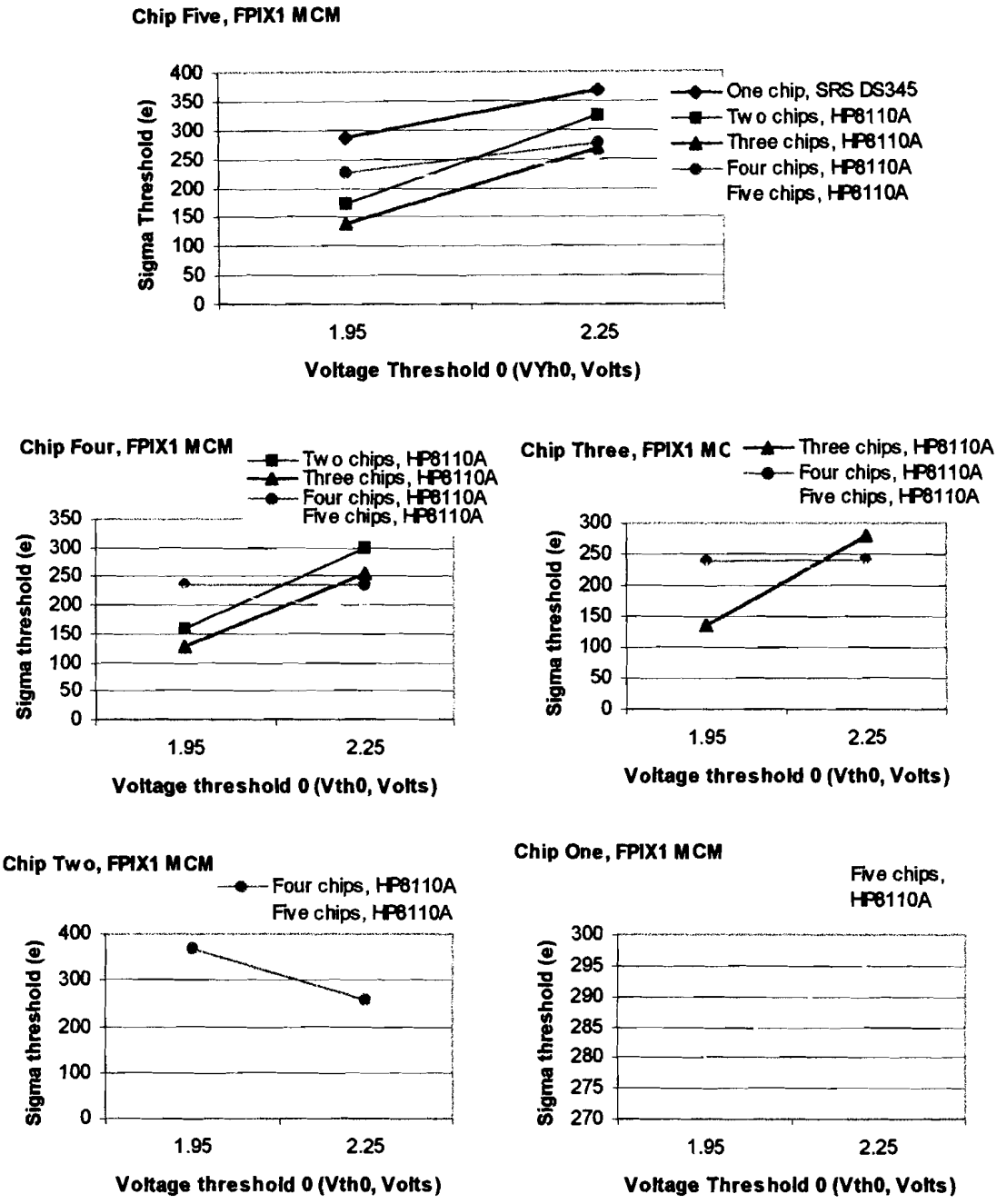


Figure 5.14. Mean sigma threshold comparison when we were adding one by one FPIX1 chip on the MCM.

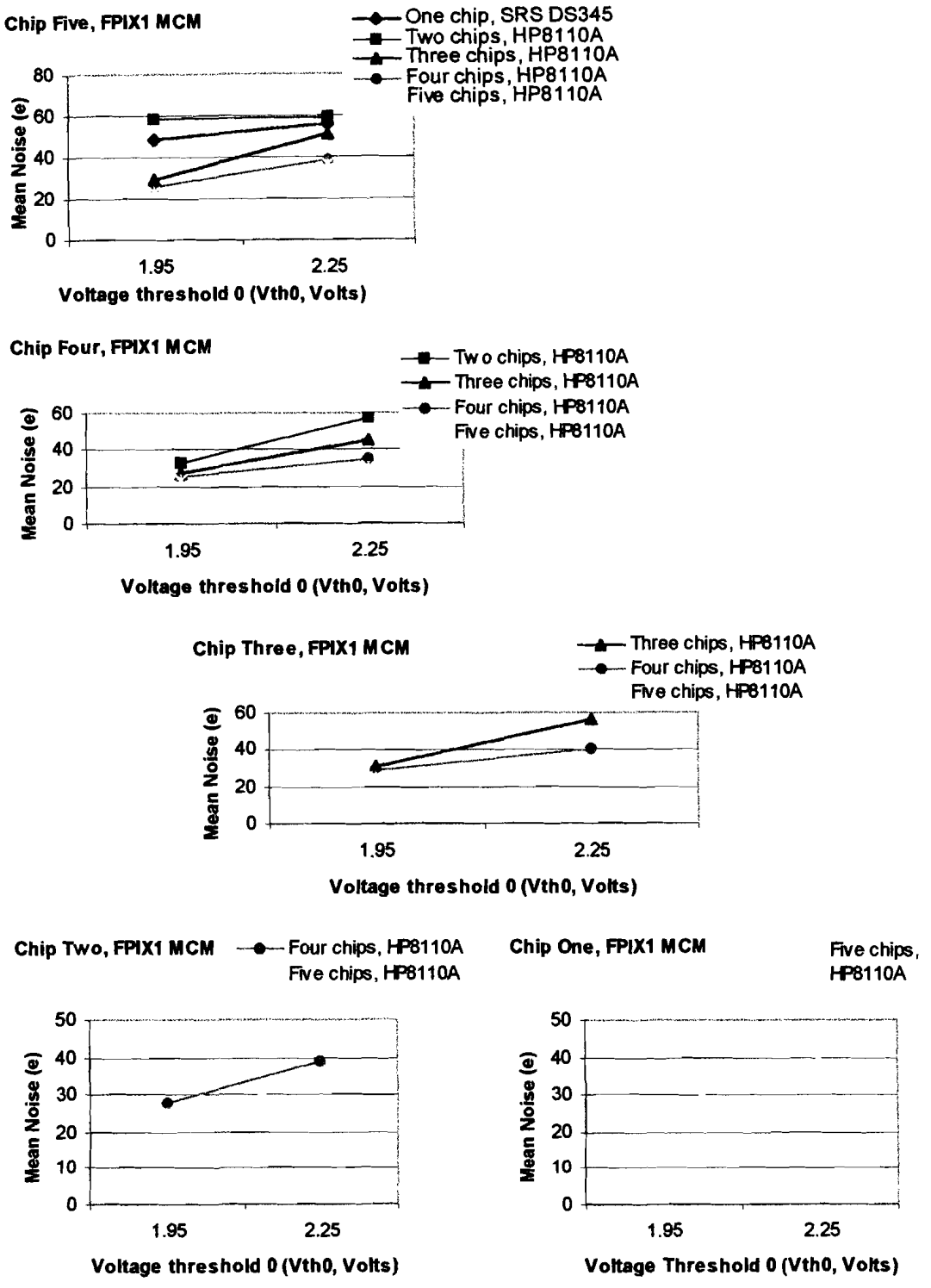


Figure 5.15. Mean noise comparison when we were adding one by one FPIX1 chip on the MCM.

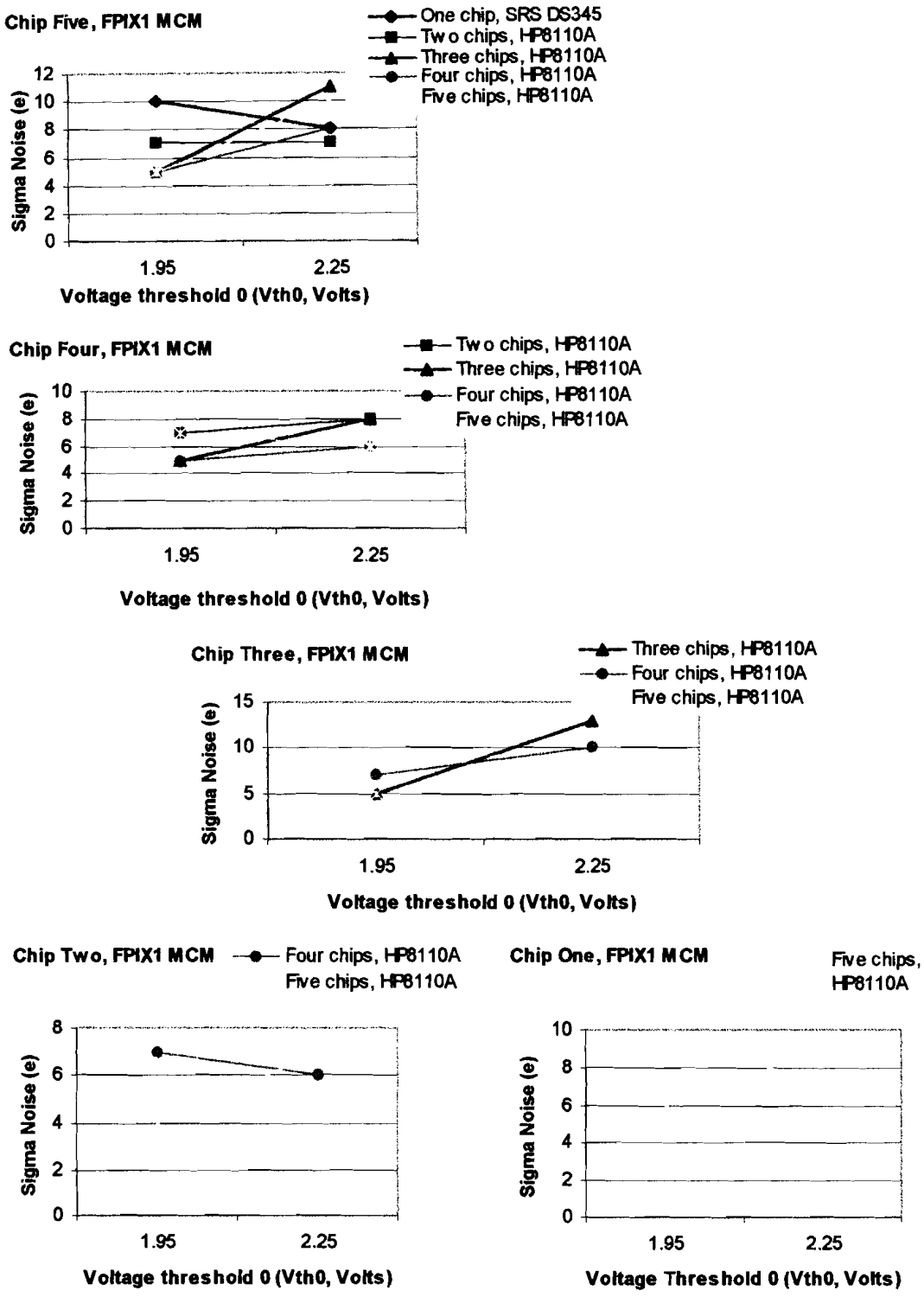


Figure 5.16. Sigma noise comparison when we were adding one by one FPIX1 chip on the MCM.

We can see from previous figures that if we use different pulse shapes for the charge injection the results are somewhat different. For example if you inject charge in the chip number 5 with the function generator SRS DS345 you obtain a mean threshold of 1980e for Vth0 set to 2.25V. If you inject charge with the pulse generator HP8110A at the same value of Vth0 you obtain a mean threshold of 2575e. This is caused by the effect of the holes that has the square wave that delivers the HP8110A because has a rise time of 195ns, with this time you are injecting some holes instead of electrons for which have been designed the charge sensitive amplifier of the FPIX1 chip. Then to have better results you need to use the function generator SRS DS345.

We did several tests to find the solution for the missing column problem. When we initialize each chip, we perform a reset and then configure the chip, i.e. send the ChipID number, the BCO lag, and indicate that it is to operate in continuous mode. You need to send a reset for each chip. We designed this mask in a special format to take into account the feature that Shift-Out is inverted respect to Shift-In. In the beginning, we left ten clock cycles between masks. The space between masks caused the missing columns because, when the bits in the reset mask are set to one, the column is effectively shut down. We then remove these ten spaces, and got back the missing column. We learned that we cannot leave any spaces between masks when we perform a reset of the chips. We can see what the data looks like when all the columns output data in the figure 5.17.

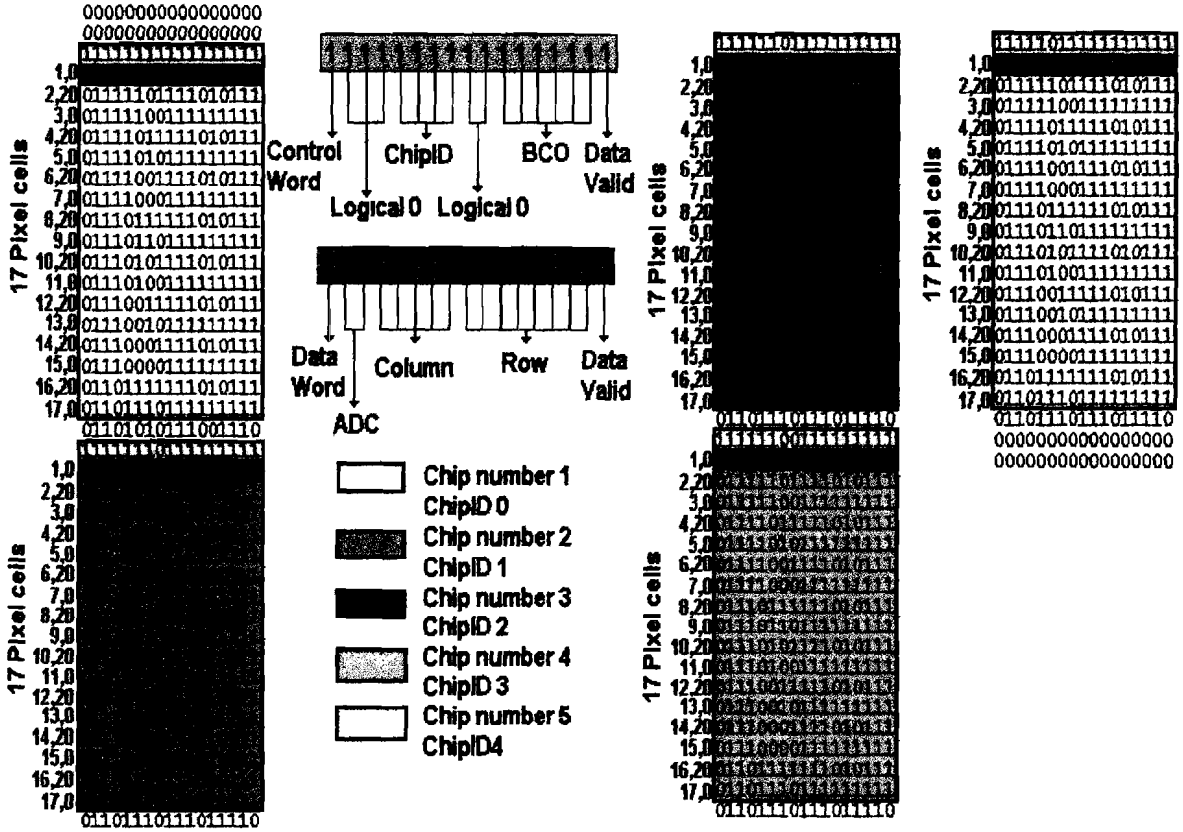


Figure 5.17. Data acquired with the logic analyzer from 5 chips in the module shown the correction of the columns missing problem.

After finding the cause for the missing columns and correcting it, we repeat the characterization, but now for four regions of operations, i.e., setting Vth0 to 1.95V, 2.05V, 2.15V and 2.25V. The results are shown in tables 5.3 and 5.4. The relationship between Vth0 and mean threshold, sigma threshold, mean noise and sigma noise are shown in the figures 5.18, 5.19, 5.20 and 5.21. It is apparent from these tables and figures that the results are much more consistent from chip to chip when all columns output data. This is because the cells which did not report hits were not charge injected. Thus, load resistance was different depending on the number of chips in the MCM. These are thus the final correct results.

	Mean threshold (e-)				Sigma threshold (e-)			
Vth0 (V)	**1.95**	**2.05**	**2.15**	**2.25**	**1.95**	**2.05**	**2.15**	**2.25**
Chip 1	7321.1	5437.2	3660.4	1859.8	221.7	240.2	247.2	252.4
Chip 2	7400.3	5506.4	3720.7	1919.5	189.9	208.8	213.6	224.9
Chip 3	7289.9	5402.2	3623.3	1830.8	209.1	219.1	227.0	238.4
Chip 4	7355.3	5472.7	3691.4	1895.2	187.5	180.9	186.3	197.1
Chip 5	7354.3	5430.9	3616.7	1770.7	217.6	235.1	246.6	264.3

Table 5.3. Final results of the characterization for threshold uniformity on the MCM for four regions of operation.

	Mean Noise (e-)				Sigma Noise (e-)			
Vth0 (V)	**1.95**	**2.05**	**2.15**	**2.25**	**1.95**	**2.05**	**2.15**	**2.25**
Chip 1	39.3	37.8	37.9	37.5	8.7	7.8	7.4	7.3
Chip 2	36.5	36.3	36.7	36.6	8.8	7.7	7.9	7.2
Chip 3	38.0	38.2	37.7	37.7	9.8	9.1	8.9	9.2
Chip 4	32.2	32.0	31.6	31.1	8.9	7.9	7.9	7.2
Chip 5	35.3	35.6	36.0	37.6	10.2	9.0	9.8	9.9

Table 5.4. Final results of the characterization for noise on the MCM for four regions of operation.

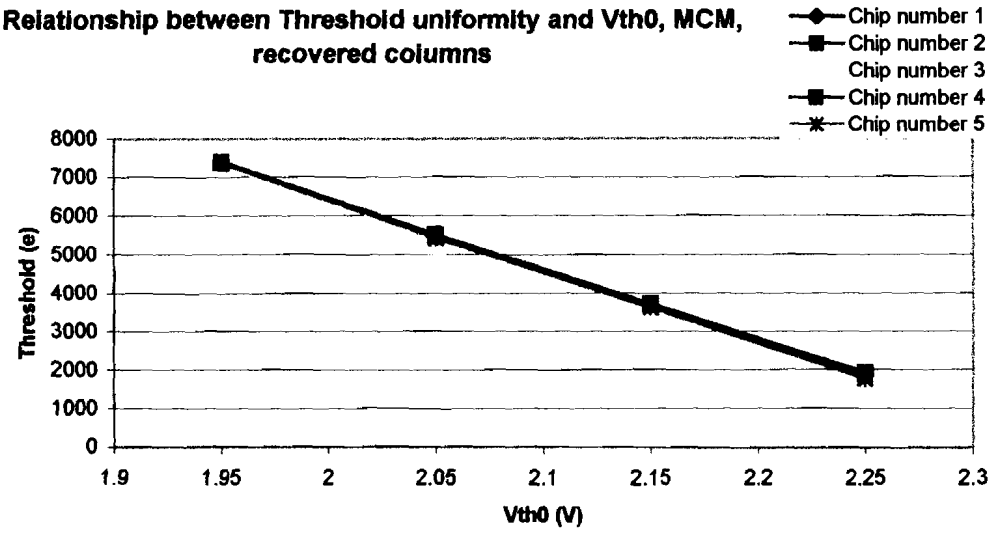


Figure 5.18. Relationship between mean threshold and Vth0 for each chip in the module.

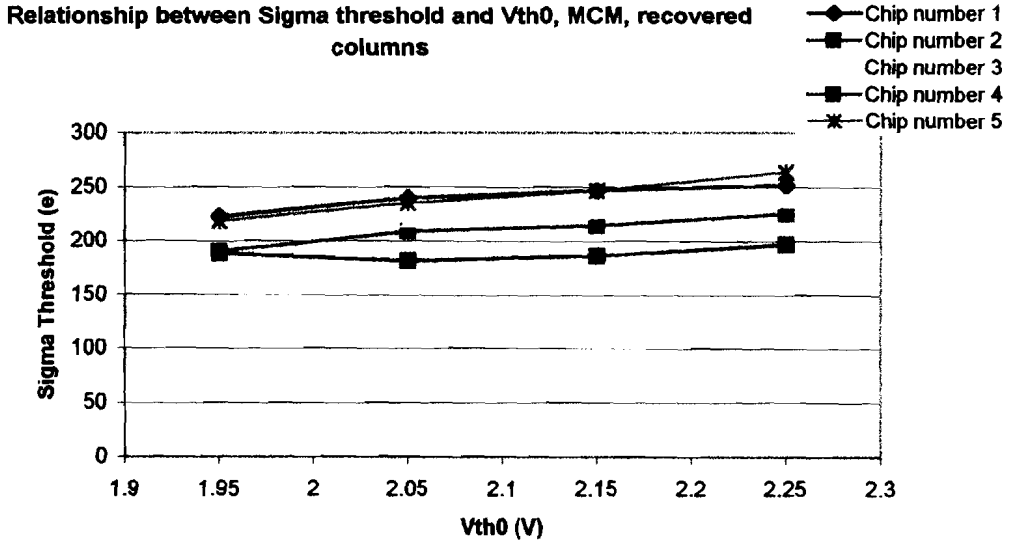


Figure 5.19. Relationship between sigma *threshold and Vth0 for each chip in the module.*

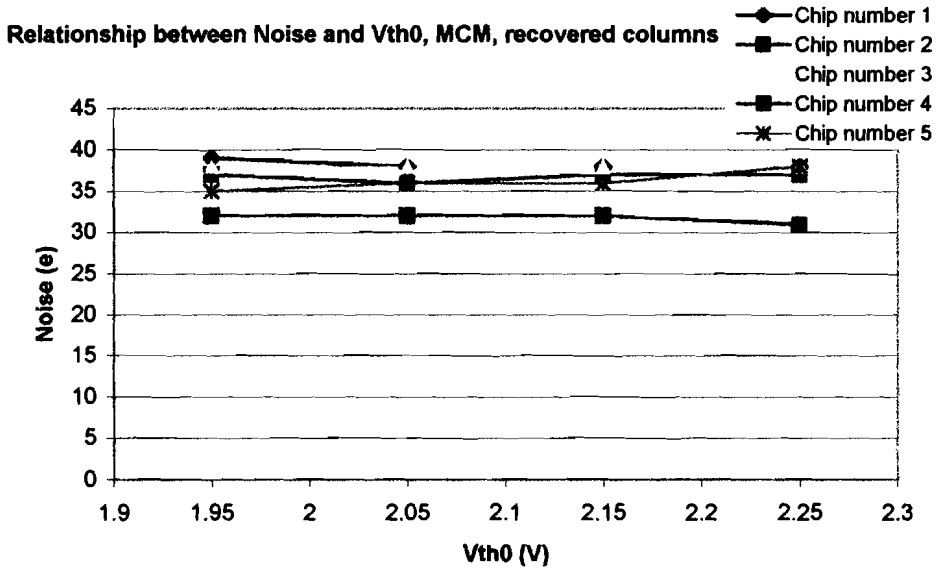


Figure 5.20. Relationship between noise and Vth0 for each chip in the module.

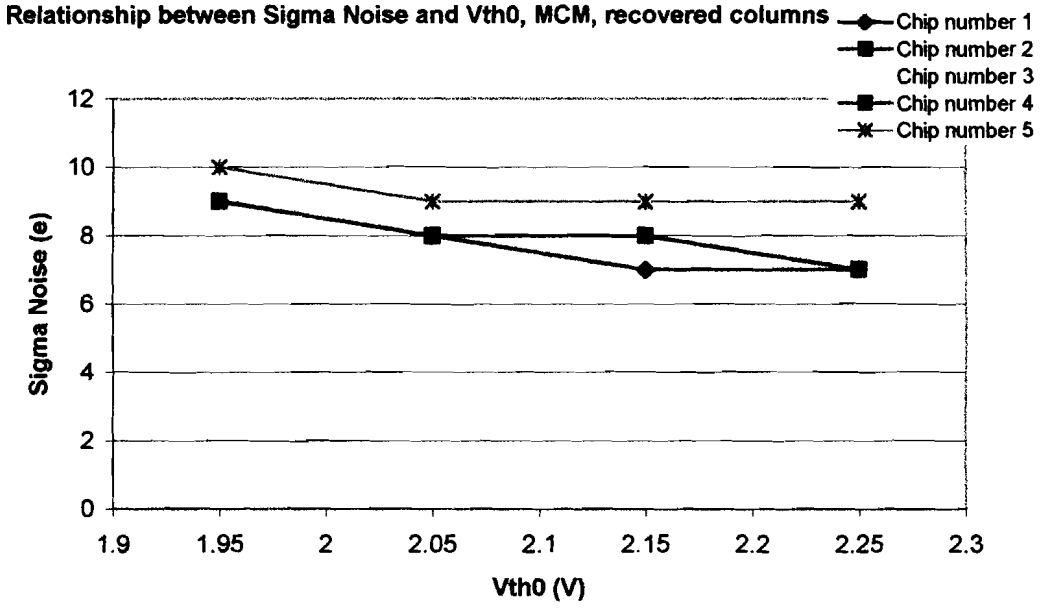


Figure 5.21. Relationship between sigma noise and Vth0 for each chip in the module.

5.4 MULTI CHIP MODULE (MCM) WITH SENSOR, THRESHOLD UNIFORMITY AND NOISE TEST

We did the characterization of the MCM mounted on a sensor. The sensor was the Tile1 P-STOP designed by the ATLAS experiment at CERN. We repeated the whole procedure with the only difference being that the FPIX1's were bump-bonded to the sensors. We want to see the effect of the sensors by looking for differences from the results that we got without sensors on the readout chips. Bench tests had shown that chip number 4 on this MCM was in short circuit. Thus in this specific MCM we only had four chips to be characterized. In table 5.5 we summarize all the results obtained for the characterization with sensor, setting the Vth0 terminal of the chips at 1.95V. We also show as figures 5.22 and 5.23 the histograms and corresponding gaussian curves of the threshold and noise distribution on the chip number 1 bump-bonded to the sensor. As could be expected, the noise, and thus also the threshold dispersion increases when the pixel cells were bump-bonded to sensors. The mean threshold is also somewhat higher. The very large difference in the mean threshold for chip number 1 and the large threshold dispersion for chip number 3 are not understood. Unfortunately, there was no time to carry out more tests.

Vth0 = 1.95V	Chip number 1 ChipID 0	Chip number 2 ChipID 1	Chip number 3 ChipID 2	Chip number 4	Chip number 5 ChipID 3
μ Threshold (e-)	9175.04	7885.93	7720.17		7823.57
σ Threshold (e-)	283.61	307.89	498.08		340.24
μ Noise (e-)	50.6	58.79	61.23		64.86
σ Noise (e-)	14.7	13.91	13.59		10.51

Table 5.5. Threshold and Noise test results over 170 pixel cells, with sensors on it.

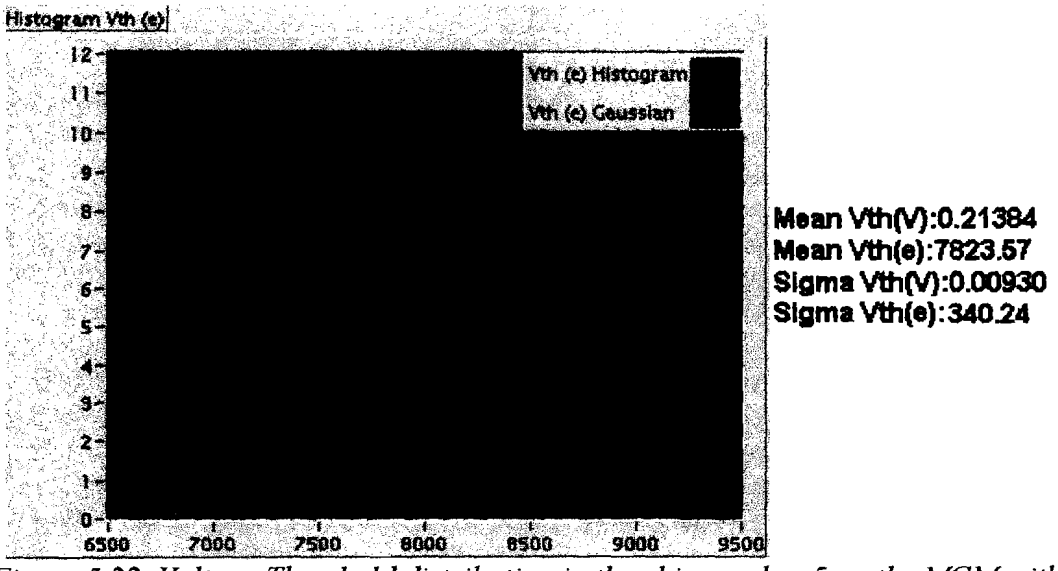


Figure 5.22. Voltage Threshold distribution in the chip number 5 on the MCM with sensors.

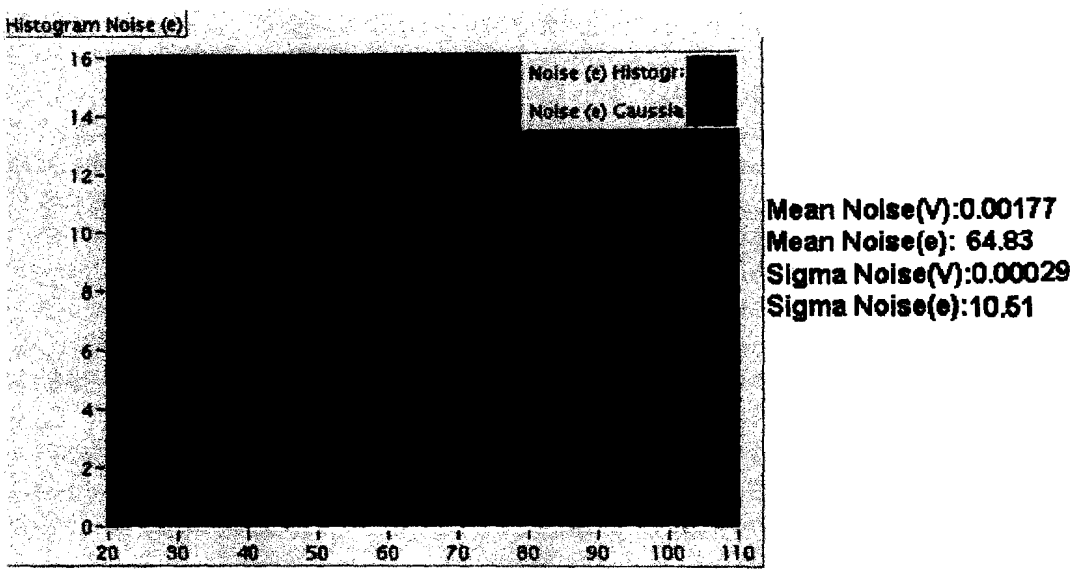


Figure 5.23. Noise distribution in the chip number 5 of the MCM with sensors.

5.5 REPRODUCIBILITY TEST

To know how reproducible the threshold uniformity and noise test is we repeated the characterization ten times to see the variation between the ten different results, In the following graphs we shows as figure 5.24 the variation of the mean threshold, as figure 5.25 the variation of the sigma threshold dispersion, as figure 5.26 the variation of the noise distribution, and as figure 5.27 the variation of the sigma noise dispersion. Reviewing all these results we can see that there is not a significant variation between the ten results of each of the quantities characterized. We can therefore say that the characterization is reproducible. All these results were obtained with the MCM without sensors on it.

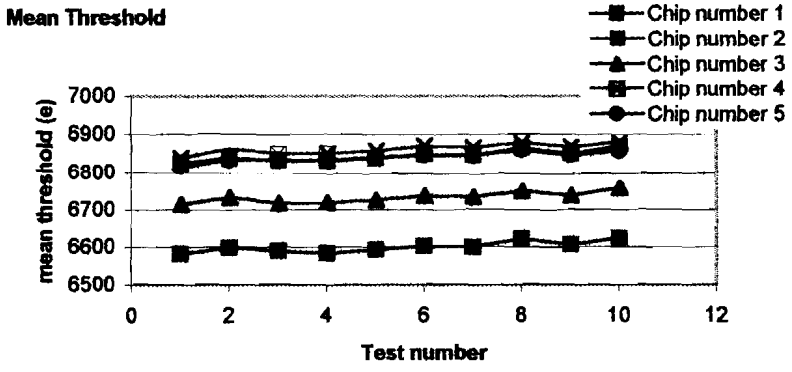


Figure 5.24. Mean threshold variation

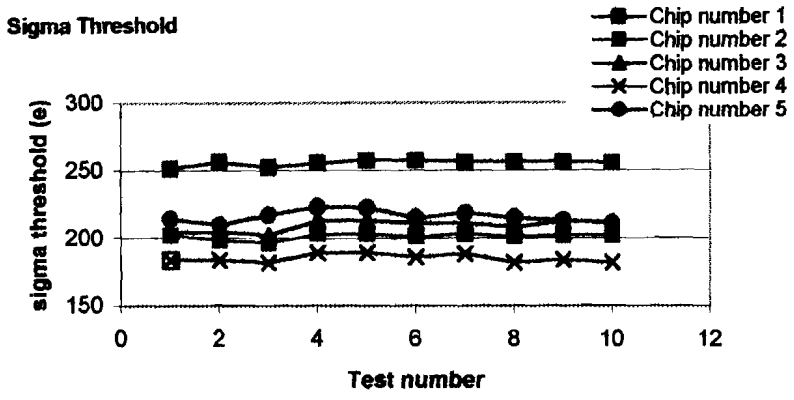


Figure 5.25 Sigma threshold variation

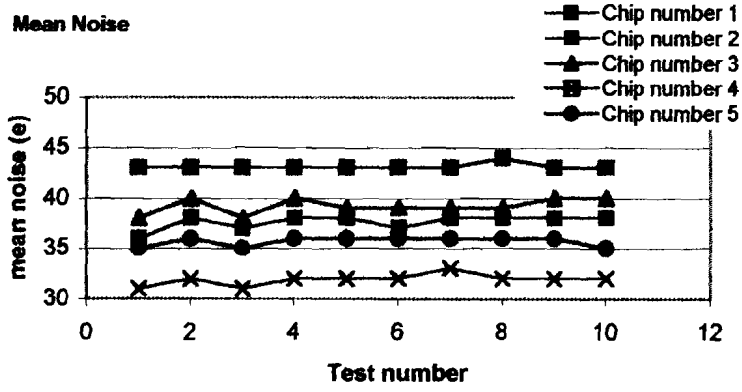


Figure 5.26. Mean noise variation

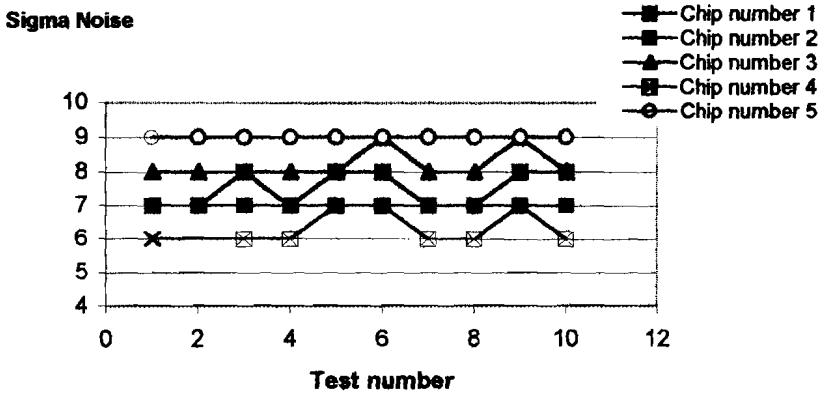


Figure 5.27. Sigma noise variation

5.6 HITS STUDIES

This test has been developed to know the influence on the results of the number of injections that you are input to each pixel cell in order to obtain the characterization. We did the characterization with 50, 100, 250, 500, 750 and 100 hits (injections). In the following graphs we show as figure 5.28 the results of the fluctuations on the results of the mean threshold distribution, as figure 5.29 the fluctuations of sigma threshold, as figure 5.30 the fluctuations for the noise distribution, and as figure 5.31 the fluctuations for the sigma noise. We can then demonstrate that there is no significant fluctuation in the results above 500 hits. Then in the following characterizations we used only 500 hits. This test was performed with the MCM without sensors on it.

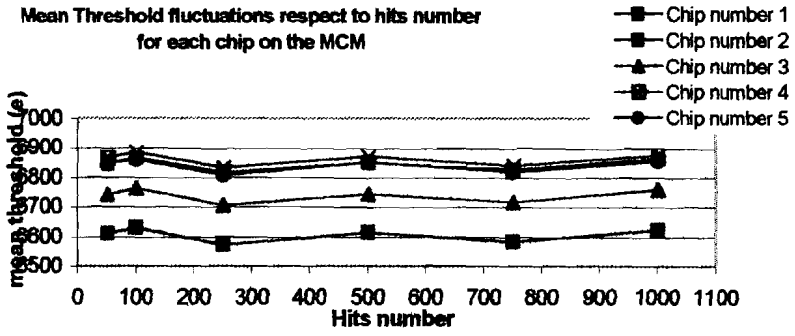


Figure 5.28. Mean threshold fluctuations respect to hits number for each chip on the MCM

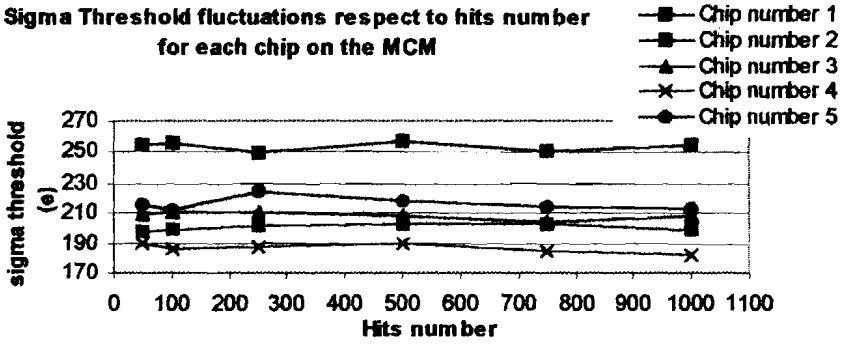


Figure 5.29. Sigma threshold fluctuations respect to hits number for each chip on the MCM

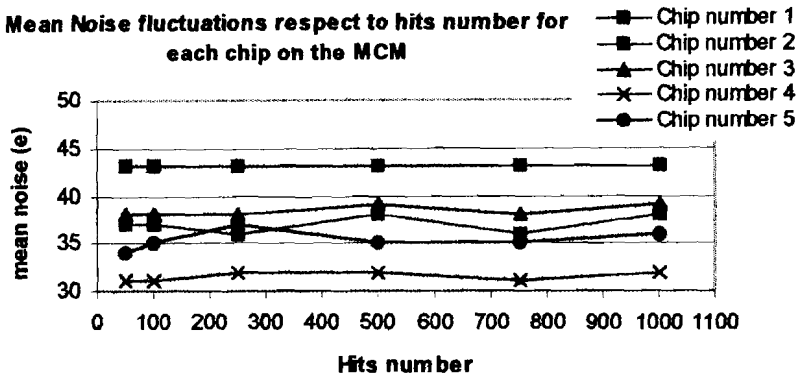


Figure 5.30. Mean noise fluctuation respect to hits number for ach chip on the MCM

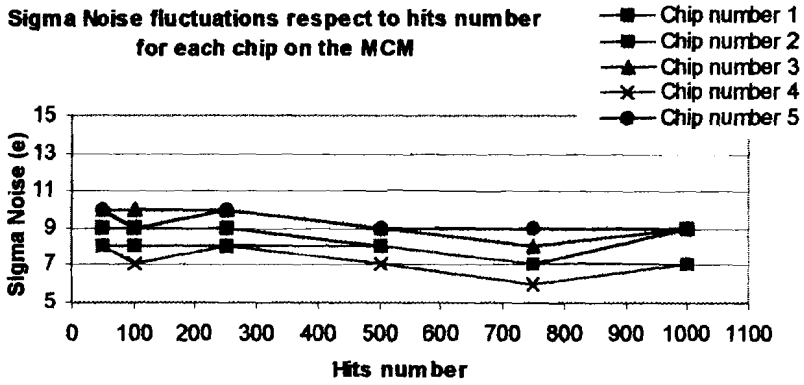


Figure 5.31. Sigma noise fluctuations respect to hits number for each chip on the MCM

5.7 OTHER STUDIES ON THE MCM

To understand whether we had a hit during the readout we did a scan each 2ns during the full readout clock cycles with the function generator SRS DS345. The threshold and noise test was done over the pixel cell 8,60 of the chip number 5 on the MCM. A sketch of the readout when we were using the function generator SRS DS345 is shown as figure 5.32.

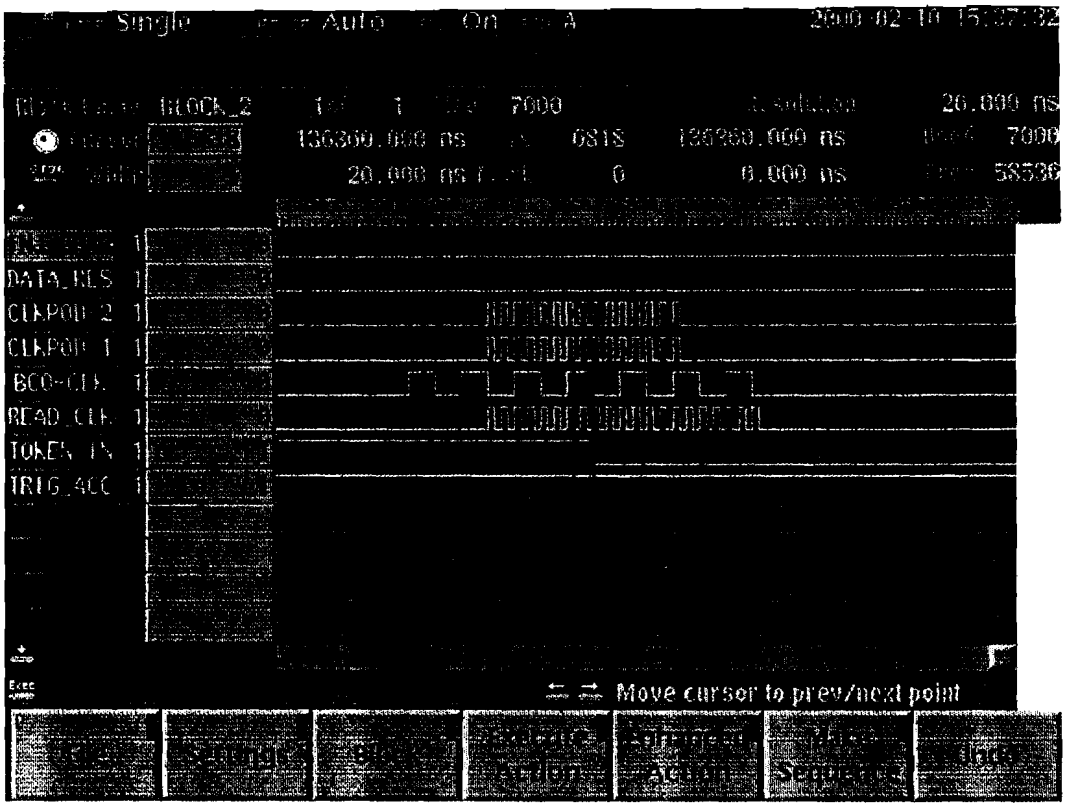


Figure 5.32. Readout with the function generator SRS DS345

To do this test we performed an upgrade of the general software we had designed for the characterization. We included an extra subroutine that controls the data generator that allows it to do a scan of only one cycle of clock pulse each 2ns. The data generator DG2020A has a function called delay. With this function you can delay the pulse that serves as a trigger for the function generator. You can cycle though delaying each time by an additional 2ns with start and end times selected by the user. In this case we did the test for full readout cycles. We did this test for only one pixel cell of one chip on the MCM. The pixel cell was column 8, row 60, and the chip was chip number 5. The results are shown as follows in figure 6.33 for the threshold fluctuations and 6.34 for the noise fluctuations for the pixel cell tested in the chip tested.

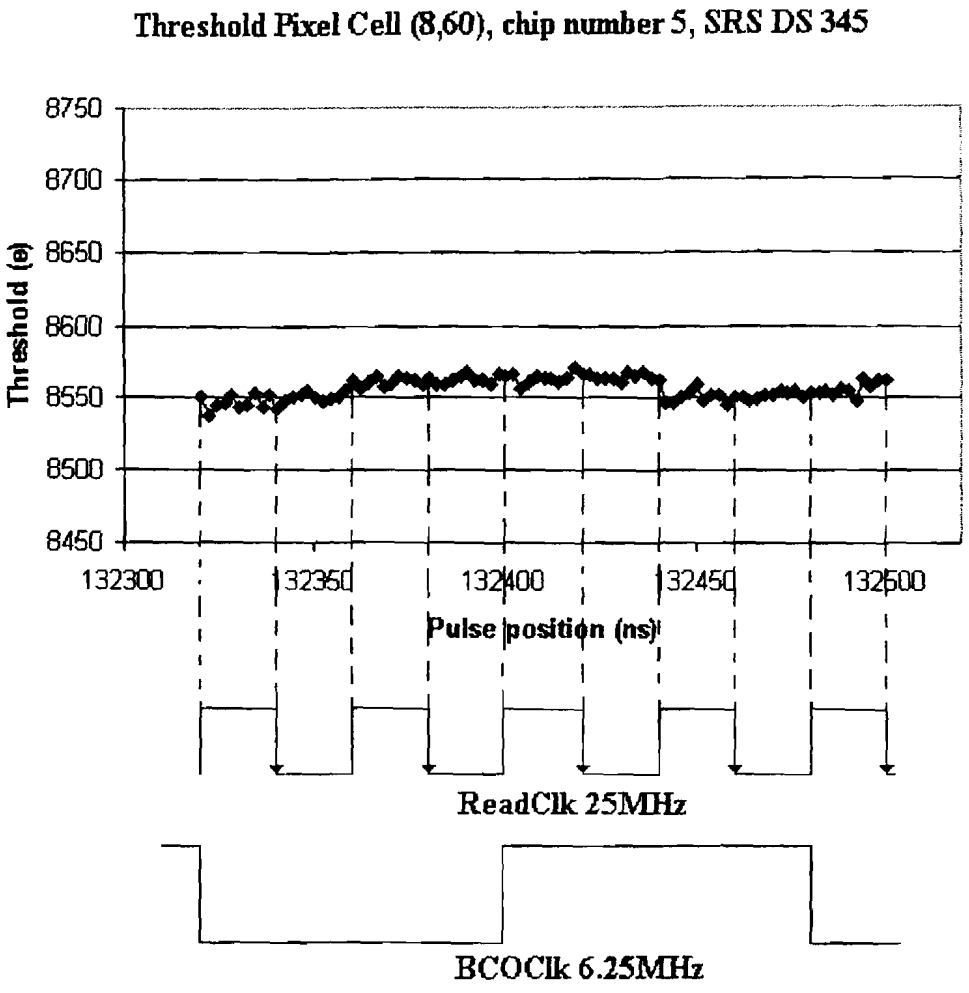


Figure 5.33. Threshold variation with an extra injection during the readout for the pixel cell (8,60) of the chip number 5 on the MCM

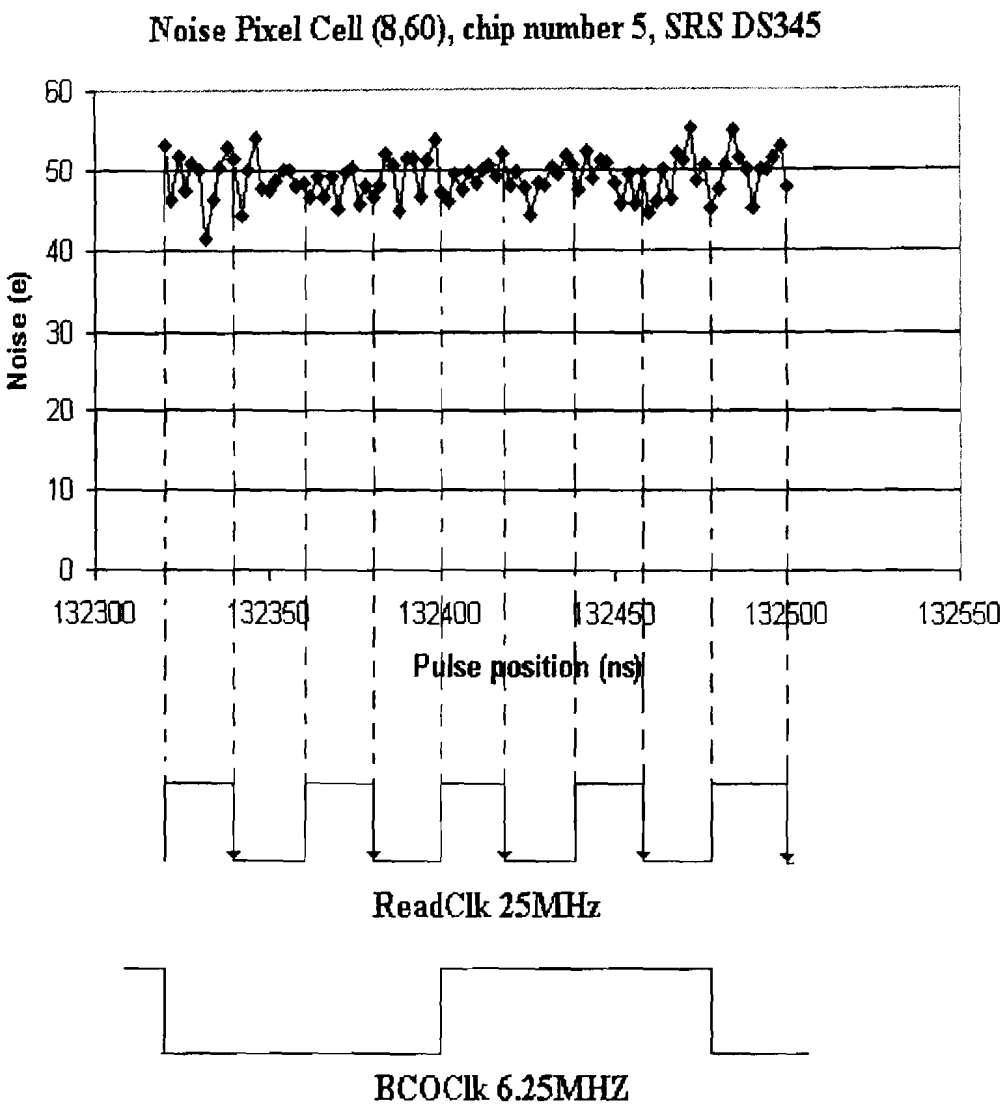


Figure 5.34. Noise variation with an extra injection during the readout for the pixel cell (8,60) of the chip number 5 on the MCM

We can appreciate by looking at the figures that there is not a significant variation if an extra hit is present during the readout stage. We also did temperature studies doing this test, we used a digital multimeter that has the capability of sensing temperature by means of a thermopar. This multimeter has special software with which you can connect it directly to the computer and save in a file the temperature each 2 minutes during the test. This allowed us to see any temperature effects on the results. As we show in figure 5.35 the mean threshold does increase when the temperature is raised.

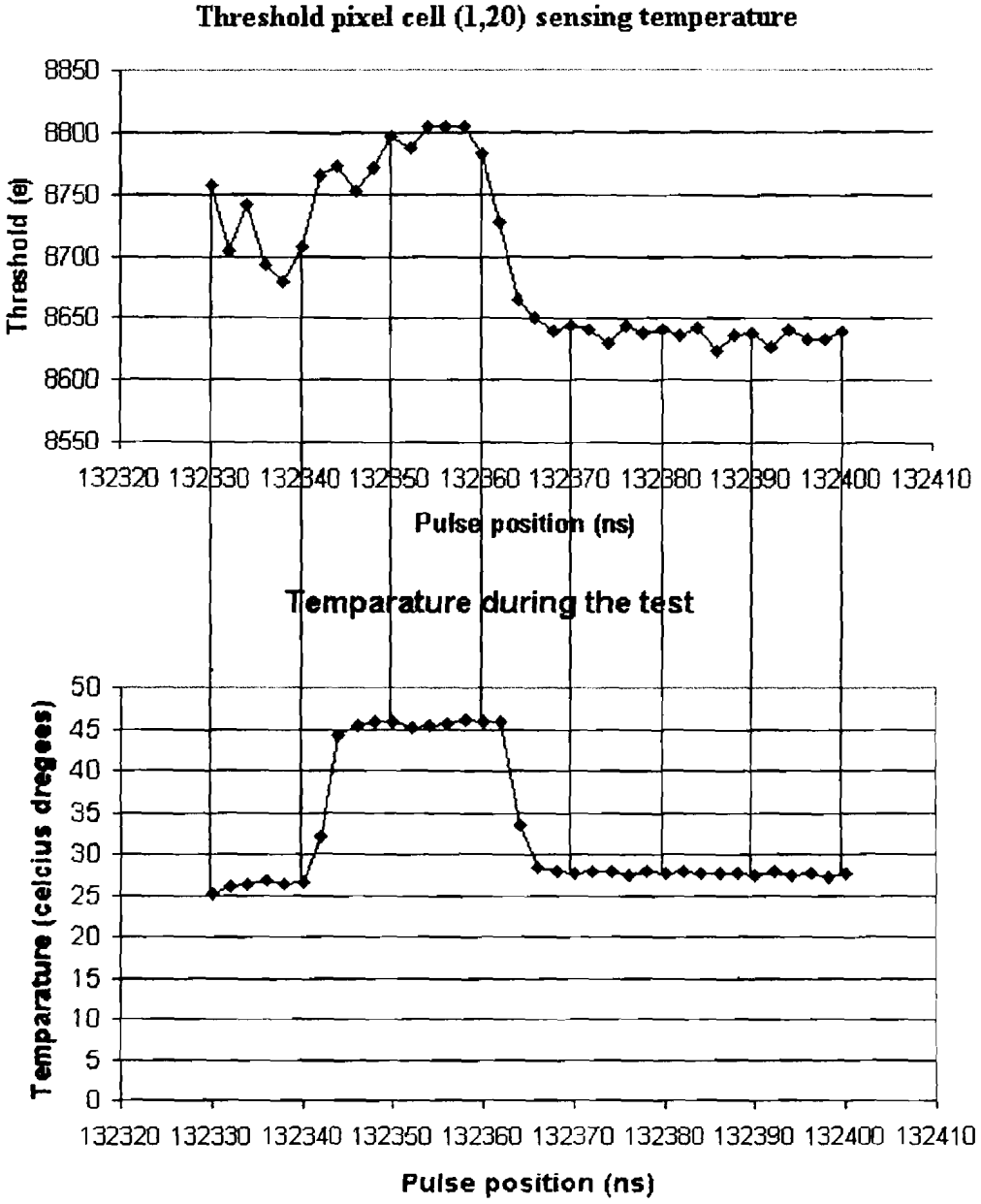


Figure 5.35. Threshold variation sensing temperature on the pixel cell (1,20) of the chip number 5 on the MCM

5.8 CHARACTERIZING THE MCM WITH ITS READOUT AND CONTROL OPTICAL INTERFACE

To verify that the ORCI [20] provides the correct command and control signals to the MCM and performs the readout without introducing errors, we took advantage of our existing test stand, including software. We repeated the noise and threshold uniformity tests that had been carried out using the test stand without the ORCI, and thus were able to make a direct comparison between the two performances. The LabVIEW software developed for the test runs the threshold dispersion and noise measurements to characterize the FPIX1 chips on the MCM with the ORCI included in the control and readout paths or without it.

We made the following modifications to the test stand to include the ORCI in the system. For the control path, instead of connecting the two data generators directly to the inputs of the MCM, we connected them to the inputs of the 104 Mbps optical link transmitter and connected the outputs of the 104 Mbps optical link receiver to the command and control lines of the MCM. For the readout path, instead of connecting the 17 data output lines of the MCM directly to the logic analyzer, we connected them to the inputs of the 1.04 Gbps optical link transmitter and the outputs of the 1.04 optical link receiver to the logic analyzer. No changes were needed to the software used to characterize the FPIX1 readout chips on the MCM. A picture of the setup is shown as Figure 5.36.

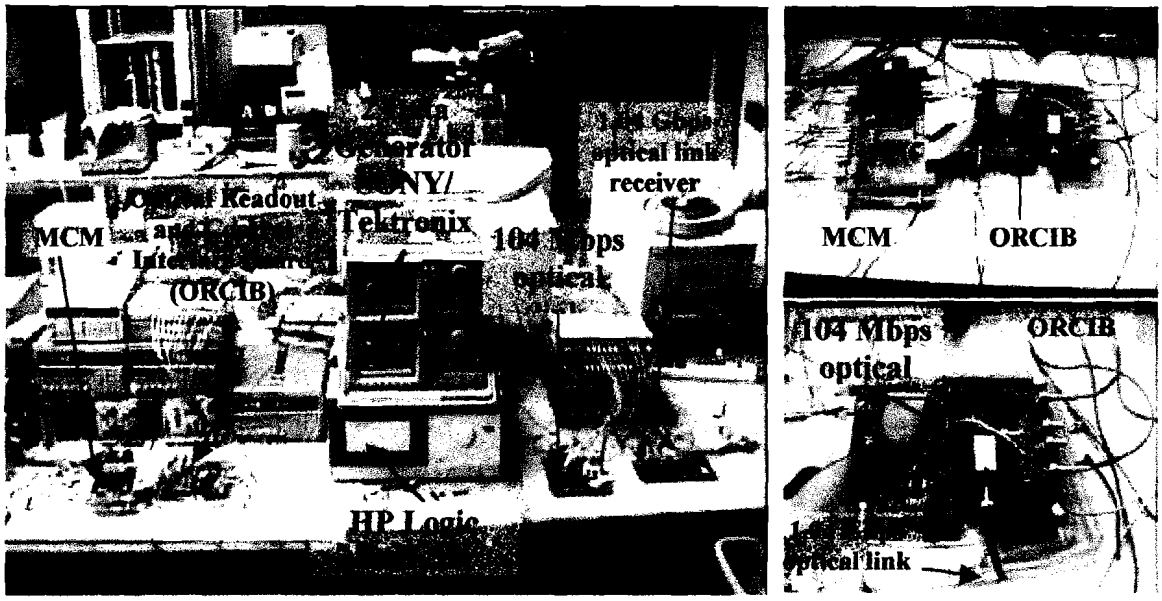


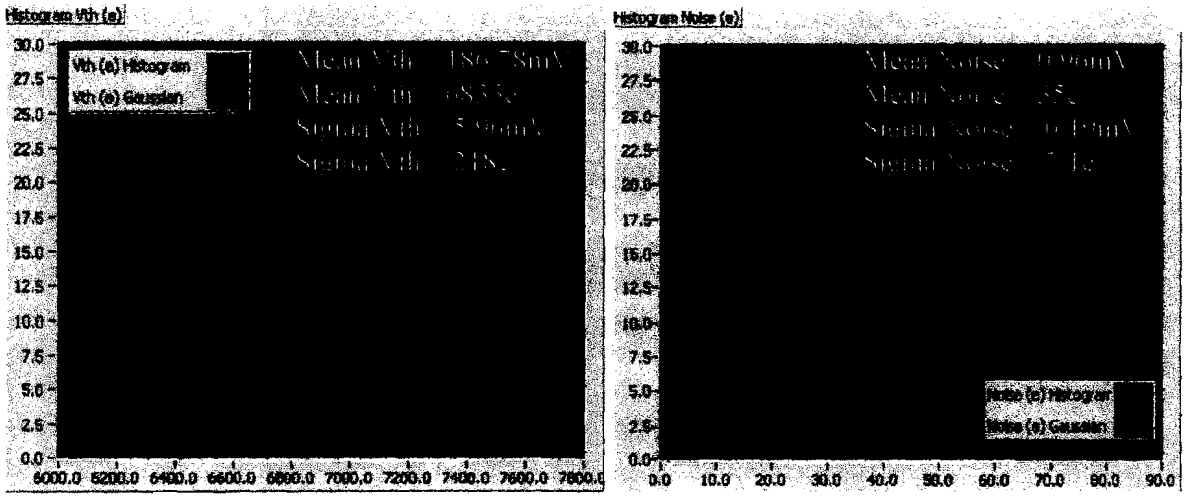
Figure 5.36. Setup used to characterize the MCM using the optical links to program, control and read out the MCM.

Table 5.6 shows the results of the threshold and noise uniformity tests of the MCM, obtained with and without the ORCI in the setup. Figure 5.37 shows the threshold and noise distribution of the first of the five FPIX1 chips on the MCM without using the ORCI. Figure 5.38 shows the threshold and noise distribution of the same FPIX1 chip using the

ORCI. The results are seen to be the same within errors. The noise is slightly higher for the case with the ORCI, most likely because of the unshielded flat cables used to interface it to the MCM.

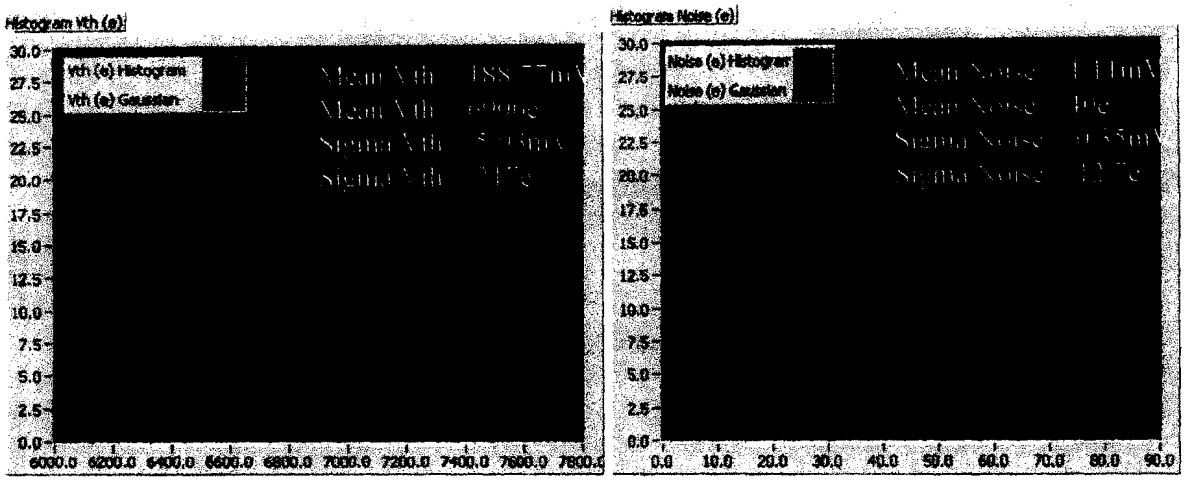
Vth0 = 1.95V	Results of the characterization of the MCM when tested without the ORCI				
	Chip 1	**Chip 2**	**Chip 3**	**Chip 4**	**Chip 5**
Mean threshold (e-)	6833	6657	6581	6792	6956
Threshold dispersion (e-)	218	239	217	168	146
Mean noise (e-)	35	37	37	31	31
Noise dispersion (e-)	7.1	8.1	9.0	5.9	6.2
Vth0 = 1.95V	Results of the characterization of the MCM when tested with the ORCI				
	Chip 1	**Chip 2**	**Chip 3**	**Chip 4**	**Chip 5**
Mean threshold (e-)	6906	6886	6820	6910	7051
Threshold dispersion (e-)	217	184	179	142	161
Mean noise (e-)	40	37	38	32	30
Noise dispersion (e-)	12.7	13.5	12.9	9.6	9.6

Table 5.6: Results of the threshold and noise uniformity of the MCM when tested with and without the ORCI.



(a) *(b)*

Figure 5.37: Distribution of the first of the five FPIX1 chips on the MCM when tested without the ORCI. a) Threshold distribution and b) Noise distribution.



(a) *(b)*

Figure 5.38: Distribution of the first of the five FPIX1 chips on the MCM when tested with the ORCI. a) Threshold distribution and b) Noise distribution.

5.9 CONCLUSIONS

Fermilab has designed and assembled a pixel-readout-chip sub-assembly containing five FPIX1 chips with flexible cable interconnections with which to address the technical issues involved in system integration for the proposed BTeV pixel detector. The module contains a total of 14400 pixel cells that need to be characterized in order to test the entire module. Software has been developed within the LabVIEW framework to control a set of instruments to perform threshold and noise tests on all five readout chips. These tests take only a few hours to run. The software designed to characterize the FPIX1 chip for the BTeV experiment is very versatile and very easy to use. The development of this work was very challenge because of the necessity to learn a lot of concepts in instrumentation, LabVIEW programming, GPIB bus protocols, and the procedures to characterize the FPIX1 chip, and later the multi chip module.

The results obtained show that the FPIX1 chip's performance is very satisfactory. The only serious problem found was that the last 60 rows in each column did not record hits. This is a feature of the layout that will be corrected on the next submission. The software we developed for the tests is really versatile. At the end of this work we had the chance of using it to perform the characterization including for first time the Optical Readout and Control Interface (ORCI) board designed to control and readout the MCM. We demonstrated that with the ORCI included in the control and readout paths there is no significant variation in the final results of the characterization.

GENERAL CONCLUSIONS

- The BTeV experiment will be one of the first experiments in using bidimensional arrays of silicon pixel detectors, because they provide high precision space points and they are quite radiation hard. Radiation hardness is necessary because the detector elements will need to be placed very close to the beam approximately 6mm. The measurements of space points, with very little noise, will provide superior pattern recognition, allowing the experiment to reconstruct tracks and vertices in real time. The BTeV pixel detector will contain about thirty million pixels. The pixel sensors will be arranged in bidimensional arrays and each pixel sensor should be read out by a dedicated electronics cell.

- In order to build the BTeV pixel detector, Fermilab has developed VLSI pixel readout chips containing front-end electronics for every pixel sensor. The pixel chip development involves a succession of steps and submissions toward a chip that meets the BTeV requirements. The chips resulting from these steps are called Fermilab Pixel Readout Chips (FPIX chips) and they have been dubbed FPIX0, FPIX1, and so on. The FPIX1 represents the first step towards the final pixel readout architecture. The FPIX1 chip is a column-based pixel chip with $50\mu m \times 400\mu m$ pixel cells arranged in an array of 160 rows by 18 columns for a total of 2880 pixel cells. Current FPIX1 prototypes have sensitive areas of up to 59.2 mm^2.

- The principal integrated circuit is one of the important parts of the pixel detector, the FPIX1 chip. We studied the chip itself, describing the way to download and acquire information from it. We have designed all the masks to program it. And, also, we have designed all the command patterns to read out all the data contained in the chip.

- Fermilab has designed and we have assembled a pixel-readout-chip sub-assembly containing five FPIX1 chips connected in daisy chain with flexible cable interconnections with which to address the technical issues involved in system integration for the proposed BTeV pixel detector. The module contains a total of 14400 pixel cells that need to be characterized in order to test the entire module and represent the first step to have a complete detector. The sub-assembly containing the 5 FPIX1 chips connected in daisy chain is called Multi-Chip Module (MCM).

- To characterize the FPIX1 chips contained in the MCM we needed to do the bad pixel map test to know if all the pixels cells are working. Another test that we need to do is the threshold and noise uniformity test, which will give us the mean threshold and the threshold dispersion of the MCM, this test needs to be done for different threshold levels. Also with this test, we can obtain the noise and noise dispersion corresponding to the MCM. As we need to know the reproducibility of this test then we need to repeat many times the threshold and noise uniformity test to know how reproducible it is. In the threshold and noise uniformity test we need to simulate the charge injection produced by the particle crossing (hit), therefore we need to inject charge to simulate

these hits, we need to obtain the relationship between the number of hits and the results of the test to see if they affect the results of test.

- We have developed a test stand including the test setup and software to characterize the FPIX1 chips contained into the MCM. The test stand characterizes automatically the FPIX1 chips connected in daisy chain into the MCM. The chain can contain from one until N FPIX1 chips. The idea of using the test setup containing general propose instruments will give us the opportunity of using the same setup with a software upgrade to characterize future versions of the FPIX chips.

- The first test stand used to characterize the FPIX1 chips took about 157 hours (6.5 days) of running to characterize only one FPIX1 chip, which means 786 hours (32 days) of running to characterize 5 FPIX1 chips connected in daisy chain. The final version after an optimization of the test setup and software takes only 6 hours of running to characterize an MCM with 5 FPIX1 chips connected in daisy chain. And it could be possible decrease the running time injecting more than 17 cells at a time, because the most recently test using a computer with a 750 MHz processor speed shows that the process time is of only about 2 hours of running to characterize a MCM with 5 FPIX1 chips connected in daisy chain.

- During the test we found that the FPIX1 chips present a inversion in Shit_out respect Shift_in, this feature was affecting the daisy chain producing a bad functionality of the chips. The problem was solved by software, but the designers need to keep in main for future version of the FPIX chips. Another lost of information were the missing columns, which was produced by spaces in the initialization mask. After removes these spaces the FPIX1 chips were programmed correctly.

- The bad pixel map test showed that the last 60 rows of all the FPIX1 chips did not record hits. This is a feature of the layout that will be corrected on the next submission. The threshold and noise uniformity test showed that the MCM could detect charge injection from 1000e (Vth0 of about 2.25V) up to 9000e (Vth0 of about 1.95V). The mean threshold and the threshold dispersion are consistent from chip to chip. The relationship between the charge injection and Vth0 was linear for all of the FPIX1 chips contained in the MCM. The result of the reproducibility test showed that there is not a significant variation between the results of each of the quantities characterized. We can therefore say that the characterization is reproducible. The hits studies showed that there is no significant fluctuation in the results above 500 hits. Then we suggest that in all the future characterizations will be used only 500 hits. All the above tests were performed with the MCM without sensors on it. As could be expected, the noise, and thus also the threshold dispersion increases when the pixel cells were bump-bonded to sensors. The mean threshold is also somewhat higher. The very large difference in the mean threshold for chip number 1 and the large threshold dispersion for chip number 3 are not understood. Unfortunately, there was no time to carry out more tests.

- The results obtained shows that the FPIX1 chip's performance is very satisfactory. At the end of this work we had the chance of using it to perform the characterization including for first time the Optical Readout and Control Interface (ORCI) board

designed to control and readout the MCM. We demonstrated that with the ORCI included in the control and readout paths there is no significant variation in the final results of the characterization.

- With the design of the test stand to characterize automatically the MCMs of the BTeV pixel detector we are doing a contribution to help that the BTeV experiment comes true.

REFERENCES

1. Fermilab's webpage, http://www.fnal.gov/pub/tour.html.
2. Butler J., Stone S., *"Proposal for an Experiment to Measure Mixing, CP Violation and Rare Decays in Charm and Beauty Decays at the Fermilab Collider, BteV"*, Fermilab Internal document, May 1999.
3. Santoro, A., et al., *"BTeV: An Expression of Interest for a Heavy Quark Program at C0"*, Fermilab Internal document, May 18, 1997.
4. Ed Barsotti, Gustavo Cancelo, Bob Downing and Sergio Zimmermann, *"Flex Circuit for the Detector Plane Interconnect"*, Internal Document # ESE-PIX-9981014, October 14, 1998.
5. G. Cancelo, D. Christian, J. Hoff, A. Mekkaoui, R. Yarema, S. Zimmermann, *"High readout speed pixel chip development at Fermilab"*, Fourth Workshop on Electronics for LHC Experiments, 528-532, 1998.
6. D. Christian *"A rad-hard pixel readout chip for the BTeV"* , Vertex 2000.
7. J. Hoff, *FPIX1 pad description*, Internal Document, November 20th 1998.
8. B. Chou, et al., *"Ultra-High-Density Interconnect Flex Substrates"*, HDI Magazine, December 1998.
9. Glenn F. Knoll, *"Radiation Detection and Measurement"*, John Wiley and Sons, New York, 2nd Edition, c.1989, Chapter 11, pp 337-386.
10. R.M. Barnett, et all., *"Particle physics booklet"*, American institute of physics, July 1996.
11. Butler J., Stone S., *"Proposal for an Experiment to Measure Mixing, CP Violation and Rare Decays in Charm and Beauty Decays at the Fermilab Collider, BteV, Appendix A"*, Fermilab Internal document, May 1999.
12. The ATLAS pixel sensor group, *"ATLAS pixel detector, technical design report"*, ATLAS experiment at CERN, May 1998.
13. Tektronix, Inc. *Reference Manual of the PS2520G.*

14. SONY/Tektronix, *Reference Manual of the GD2020A.*

15. Standfor Research Systems, *Reference Manual.*

16. Hewlett-Packard, *Reference Manual.*

17. M. Colloms., "*General Purpose Interface Bus (GPIB)*", The anchor Press Ltd, 1983.

18. National Instruments Corporation, LabVIEW *Users Manual*, 1998.

19. National Instruments Corporation, *LabVIEW Function Reference Manual*, 1998.

20. S. Vergara, et all., "*Optical Readout and Control Interface for the BTeV Pixel Vertex Detector.*" Submitted to NIMA, September 2000.

21. Richardson, J., "The ATLAS pixel on-detector electronics," 5th Workshop on Elect. LHC Exp., Snowmass, pp.83-87, Sept. 1999.

22. Skubi. P., *et.al.*, "Flex circuits for the ATLAS pixel detector," Proc. Pixel98, pp.519-534, Batavia, IL, May 1998.

23. Christian, D.C., *et.al.*, "Development of a pixel readout chip for BTeV," NIM A 435, pp.144-152, 1999.

24. Mekkaoui, A., *et.al.*, "FPIX2: an advanced pixel readout chip," 5th Workshop on Elect. LHC Exp., Snowmass, pp. 98-102, Sept. 1999.

24. Koopman, N. *et.al.*, "Fluxless No-Clean Assembly of Solder Bumped Flip Chips," 46th ECTC, Orlando, Florida, May 28-31, 1996.

25. Cancelo, G.E., *et.al.*, "Fiber Optic based readout for BTeV's Pixel Detector," to be presented 6th Workshop on Elect. LHC Exp., Krakow, Sept. 2000.

26. ATLAS Inner Detector technical design report, CERN/LHC, pp.16-17, 1997.

27. G. Tonelli., *et. al.*, "The R&D program for silicon detectors in CMS" Nuclear Instruments and Methods A, Num. 435, pp. 109, 1999.

28. Artusso, M. *et al.*, "Some further studies on factors affecting the pixel resolution" BTeV internal note, July 1998.

29. Mekkaoui, M. *et al.*, "Results from the FPIX0 chip bump-bonded to the ATLAS pixel detector", Proc. 4th. Workshop on Electronics for LHC experiment, Rome, Sep. 21-25, 1998.

30. Wrigth, J., et al., "A pixel unit-cell targeting 16ns" LBL-32912, Berkeley, 1992.

31. Mekkaoui, M. *et al.*, "FPIX0: A prototype pixel FE chip at Fermilab" Proc. 3rd. International meeting front end electronics for high resolution tracking detectors, Taos, NM, Nov. 1997

32. Blanquart, L., et al., "Pixel analog cells prototypes for ATLAS in DMILL technology", Nuclear Instruments and Methods A, 435, pp. 144-152, 1999.

33. Mekkaoui, M. *et al.*, "FPIX1: An advanced Pixel readout chip", FERMILAB Conf-99/280, published in the Proc. of the fifth workshop on electronics for LHC experiments, Colorado, September 20-24, 1999.

34. L.Adams, et al., "2nd RD49 Status report: Study of the radiation tolerance of Ics for LHC", CERN/LHCC 99-8 March 1999.

35. Mekkaoui, M. *et al.*, "Radiation tolerant FE in the TSMC 0.25μ process" to be published in the proceedings of Pixel 2000, International workshop on Semiconductor pixel detectors for Particles and X-rays, June 5-8, 2000.

36. J. Hoff, et al., "PreFPIX2: Core architecture and results" FERMILAB conference 00/260E, to be published in the proceedings of the IEEE 2000 Nuclear Science Symposium.

37. A. Kulyavtsev, et al., "FPIX0 test stand studies of chip characteristics" Internal Doc. BTeV coll., June 14, 1999.

38. S. Vergara, et al., "Optical readout and control interface for the BteV pixel vertex detector", Fermilab preprint FERMILAB-PUB-00-246E, pp. 13, November 2000.

Publications

1. M.A. Vargas, M. Sheaff, S. Vergara, "The development of software to characterize the fermilab pixel readout chip for the BTeV experiment", Preceedings of the SOMI XIV CONGRESO DE INSTRUMENTACION, pp. 608-612, Tonanzintla, Puebla, MEXICO 1999.

2. M.A. Vargas, M. Sheaff, S. Vergara, "The development of software to characterize the Fermilab pixel readout chip, FPIX1, for the BTeV experiment", Accepted to NIMA, November 2000. Fermilab preprint FERMILAB-PUB-00-244E, November 2000.

3. S. Zimmermann, S. Kwan, G. Cancelo, G. Cardoso, S. Cihangir, D. Christian, R. Downing, M. A. Vargas, et al., "Development of high data readout rate pixel module and detector hybridization at Fermilab", Preceedings of the PIXEL 2000, Genoa, June 2000.

4. G. Cancelo, S. Vergara, M.A. Vargas, et al., "Fiber optic based readout for the BteV's pixel detector", Preceedings of the LEB Sixth workshop on electronics for the LHC experiment, Cracow, Poland, October 2000.

THE DEVELOPMENT OF SOFTWARE TO CHARACTERIZE THE FERMILAB PIXEL READOUT CHIP FOR THE BTeV EXPERIMENT

M. A. Vargas[1], M. Sheaff[2], S. Vergara[1]
FCFM-BUAP, Apdo. Postal 62-570, Av. San Claudio y 14 sur, Puebla Pue, México
Tel. (5222) 233-2533. Fax (5222) 233-2403. Email: mavargas@fcfm.buap.mx

RESUMEN
Un detector silicon–pixel ha sido propuesto para ser usado en el primer nivel de disparo del experimento BTeV en Fermilab, USA. El principal objetivo del BTeV es llevar a cabo estudios de precisión de violación CP y raros decaimientos de quarks *b* y *c*. Las dimensiones de cada pixel deben ser muy pequeñas aproximadamente $50\ m \times 400\ m$ con la finalidad de obtener la resolución requerida por los dispositivos con tecnología VLSI. Celdas pixel con las mismas dimensiones han sido diseñadas y construidas en sensores, conteniendo cada sensor 2880 canales, la tarea de caracterizarlos es un reto al potencial consumo de tiempo. El software ha sido desarrollado en LabVIEW y controla un grupo de instrumentos para efectuar las pruebas de umbral y ruido sobre todo el chip en menos de una semana de funcionamiento.

Abstract
A silicon pixel detector has been proposed to be used in the first level trigger of the BTeV experiment at Fermilab. The BTeV main goals are to measure mixing, CP violation and rare decays in the b and c systems. The size of each pixel must be very small, approximately $50\ m \times 400\ m$ *to achieve the required resolution. VLSI chips with pixel cells of the same dimensions have been designed and built to instrument the sensors. Since each chip contains 2880 channels, the task of characterizing them is challenging and potentially time consuming. Software has been developed within the LabVIEW framework to control a set of instruments to perform threshold and noise tests on a full chip in less than one week of running.*

1. INTRODUCTION

The BTeV experiment has been proposed for the C-Zero interaction region of the Tevatron at Fermilab. A sketch of the apparatus is shown in Fig. 1a. The magnet that we will use, called SM3, exists at Fermilab. The other important parts of the experiment include the vertex detector, the RICH detectors, the EM calorimeters and the muon system [1]. The vertex detector is a multi-plane pixel device that sits inside the beam pipe The pixel detector (Fig. 1b) is composed of 93 pixel planes of 100x100mm each, divided into 31 triple-stations perpendicular to the colliding beam and installed a few millimeters from the beam. VLSI pixel readout chips containing front -end electronics for every pixel sensor will be bump-bunded to the detector. Since this detector will be employed for on-line track finding for the lowest level trigger system, the pixel chip will have to read out all detected hits.

The pixel chip development involves a succession of steps and submissions toward a chip that meets the BTeV requirements. The chips resulting from these steps have been dubbed FPIX0, FPIX1, and so on. The FPIX1 represents the first step towards the final pixel readout architecture. The FPIX1 is a column-based pixel chip with $50\ m \times 400\ m$ pixel cells arranged in an array of 160 rows by 18 columns for a total of 2880 pixel cells. The chip can be divided into three mutually dependent pieces, the pixel cell, the End of Column (EOC) Logic and the Chip Control Logic. A block diagram of the FPIX1 is shown in Fig.2a [2].

---

[1] *Scholarship holder of CONACyT, MEXICO/Work developed in a stay at Fermilab, PO Box 500, Batavia, ILL, 60510, USA*
[2] *University of Wisconsin, Madison WI 53706, USA/CINVESTAV, Mexico D.F., MEXICO*

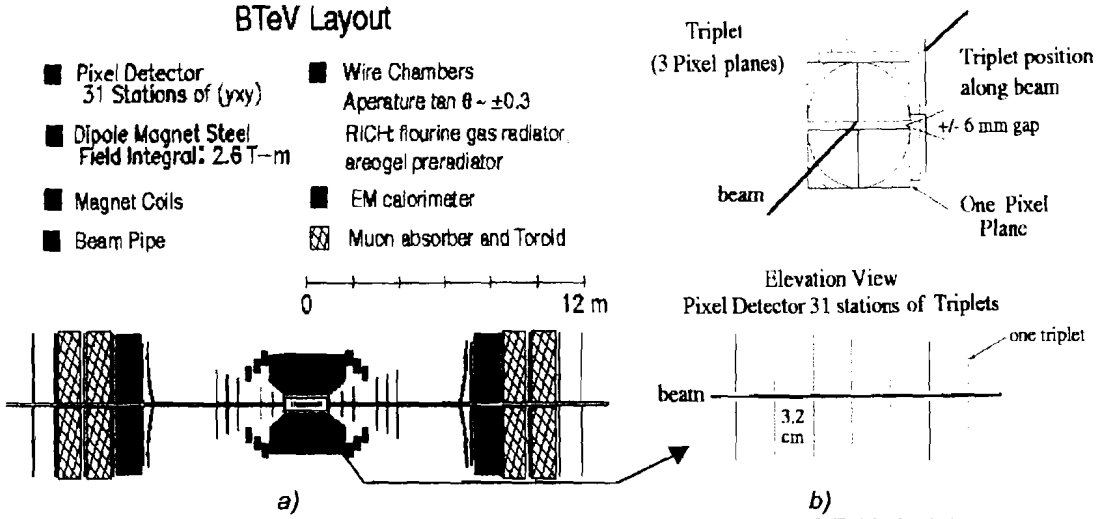


Figure 1. a) Sketch of the BTeV spectometer and b) Layout of the BTeV pixel detector.

The pixel cells hold the front-end (Fig. 2b) electronics and the digital interface to the EOC logic (Fig. 2c). The front end contains a charge-sensitive amplifier and a second amplification stage; the output of the second stage connects to a flash ADC and a discriminator. The discriminator output is asserted when the signal at the input of the discriminator is higher than the threshold (Thr). The pixel cell contains a digital interface with two major components, the command interpreter and the pixel token and bus controller. The command interpreter has four inputs corresponding to the four EOC command sets. These commands are presented by the EOC logic simultaneously to all pixel cell interpreters in a column. When an interpreter is executing the input command and the hit output from the discriminator is asserted, the interpreter associates itself with the particular EOC set and simultaneously it alerts the EOC logic to the presence of a hit via the wire-or'ed HfastOR signal. After that, the information is stored in the cell until EOC set issues an output or reset command. When this command is an output command, the interpreter issues a bus request and asserts the wire-or'ed RfastOR signal. Then the balance of the readout proceeds synchronously with the MClk. The EOC logic provides a column token, the token quickly passes pixel cells with no information until it reaches a cell that is requesting the bus. The data is composed of the ADC count bits [3:1] and the row address radd [7:0]. As the hit pixel is read out it automatically resets itself and withdraws its assertion of the RfastOR. This signal returns to its inactivated state while the last of the hit pixels is being read out [2].

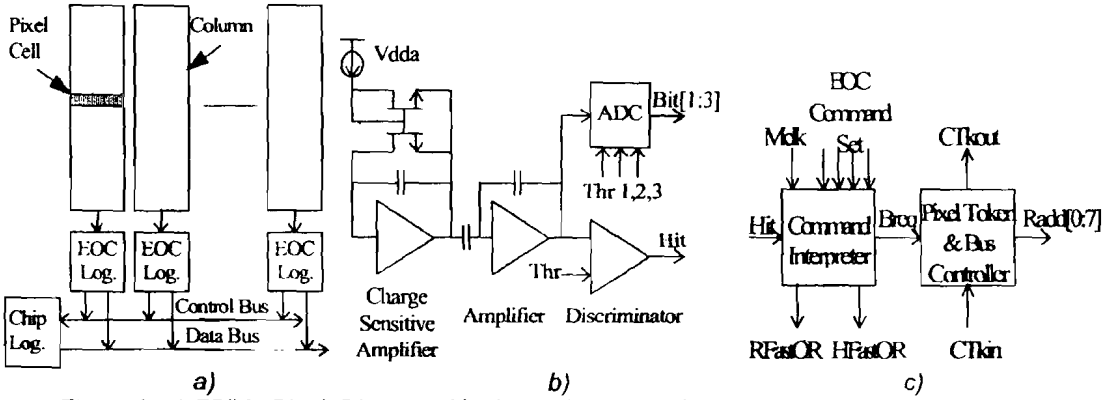


Figure 2. a) FPIX1 Block Diagram, b) Pixel cell squematic and c) Pixel cell digital interface.

The FPIX1 chip has 87 pads for external control and data readout. There are eight internal scan paths formed by shift registers with different depths, which we need to fill with information in order to program the chip. Two scan paths are the mask register path that is used to set the programmable reset delay and the mode path which contains the chip mode (triggered mode or continuous mode), the chip ID and the BCO Time Stamp. The last mode path is the Kill/Inject path, which selects the pixel cell or cells to be characterized.

2. CHARACTERIZATION

To characterize the FPIX1 chip we are using a PC with an AT/TNT-GPIB (General Purpose Interface Bus) interface to control the following instruments: Tektronix PS2520G programmable power supply, Hewlett Packard HP16500B logic analyzer system, Stanford Research Systems DS345 30MHz function generator, SONY/Tektronix DG2020A data generator and SONY/Tektronix P3420. The software developed to control all these instruments utilizes the LabVIEW[3] (Laboratory Virtual Instrument Engineering Workbench) framework. LabVIEW is a development environment based on the graphical programming language G. It is integrated fully for communication with hardware such as GPIB, VXI, PXI, RS-232, RS-485, and plug in data acquisition boards.

The test stand that we are using to characterize the FPIX1 chip is shown as Fig. 3. The programmable power supply has three outputs. One of them provides power to the digital part of the FPIX1 chip. The second output provides power to the analog part. The last one provides power to the Vth0 and represents one of the inputs to the discriminators in the front ends. We are using two data generators with programmable outputs to send the information to program the chip. One of them provides data for the Kill/Inject path. The other sends a sequence of commands to the readout stage to acquire the information that the FPIX1 chip delivers. The logic analyzer system receives the information that the chip delivers and stores this information in its memory. The function generator provides the charge to be injected to the selected cells of the FPIX1 chip. We also have two other instruments that it is not necessary to drive by means of the GPIB card. These are a regulated power supply, LEADER 718-SD, which provides power to the vdd/2 input of the FPIX1 to enable single-ended readout, and a current supply made at CPPM, Marseilles, to provide the feedback and Master Bias currents through the inputs marked Ivfb and Ivbbp.

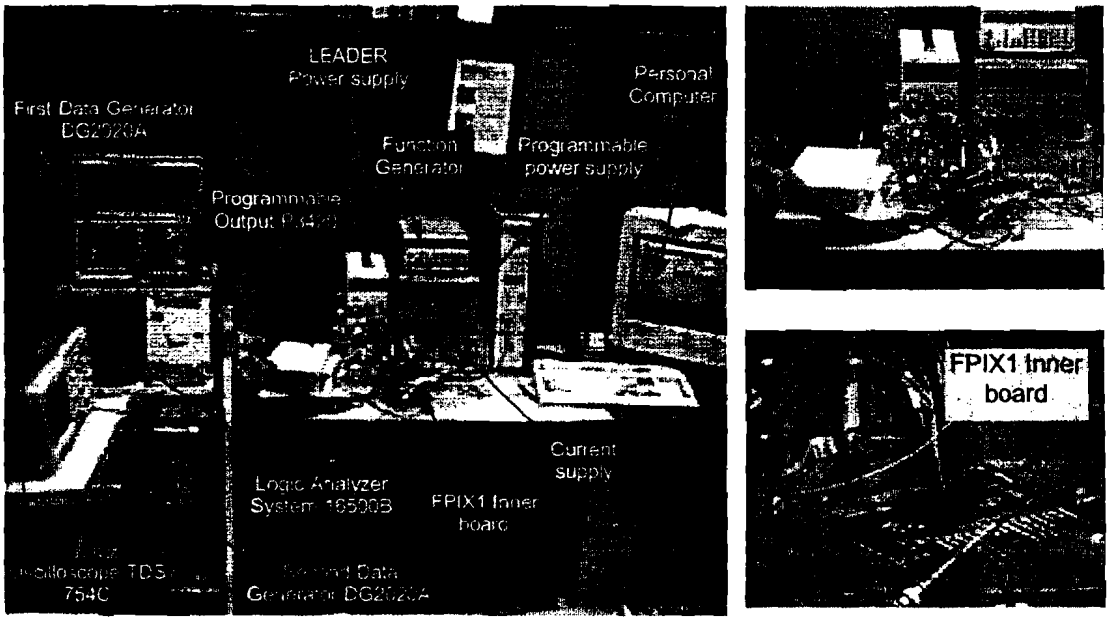


Figure 3. Test setup to get the threshold uniformity and noise from the FPIX1 chip.

We performed the following tests on the FPIX1 chip: bad pixel map, noise, and threshold uniformity. The bad pixel map test is based on injecting charge into all cells on the chip to see what cells are working. First we program the FPIX1 chip to kill all cells except five. By testing five at a time we are able to save time. Then the chip must be programmed to allow charge to be injected into these five cells. Afterwards we inject these cells with a charge level well above the discriminator threshold so that they are expected to register a hit each time that we inject them. This guarantees a response of 100% for correctly working cells. We scan the entire FPIX1 chip in this way five cells at a time to determine which cells are working correctly and which are not.

The noise and threshold tests are performed together because they use the same data acquisition software. The results shown in this article are from a sample of 170 cells selected from a coarse grid which covers the first 100 rows on FPIX1. Further tests on the full chip are planned. These noise and threshold uniformity tests are based on programming the FPIX1 chip to kill all the pixel cells except five (again to save time), where the five selected are separated from each other by a sufficient distance to remove all possibility of crosstalk between them. Then, as above, we need to program the chip to allow charge to be injected into these five activated cells.After the chip is programmed, we send an injected charge pulse 1000 times and count the number of times we record a hit in each of the five cells. This procedure needs to be repeated in steps of five cells at a time to get the information for all the 170 cells that we choose. We do this for a range of charges above and below the expected threshold. For each level of charge we need to scan all the 170 cells.The levels of charge must be chosen to bracket the voltage threshold. To do this we make a preliminary scan over the chip to determine the optimal range for the test. In our case we fix the Vth0 at 1.95V and we perform the test over three intervals. For the test reported here, the first one was from 80mV to 200mV in steps of 10mV, the second one was from 200mV to 300mV in steps of 1mV, and the third one was from 300mV to 400mV in steps of 10mV. This means a total of 123 levels of voltage, where the voltages represent the voltage peak-peak of the pulse delivered by the function generator. We can translate these voltages into electrical charge as a function of load. In our case we have a capacitive load of 6fF. We can calculate the electrical charge injected (Q_{inject}) in each case by means of Eq. 1.

$$Q_{inject} = VppC = Vpp \times 6 \times 10^{-15} \quad (1)$$

After we acquire the data for the 170 pixel cells, we process the information to get the mean and the standard deviation of both the threshold and the noise. Thus, the software developed to do these tests not only acquires the data but also performs the analysis using the data from the FPIX1 chip.

3. SOFTWARE

The software is a very important part of the tests, because it is by means of the software that we can control all the phases in the test. The software to do the bad pixel map works as follows: Initialize all the instruments used in the test. Program the FPIX1 to kill all the pixels cells except five, by means of the first data generator. After that start the logic analyzer and tell it to wait for a trigger. Then tell the second data generator first to inject the charge and then a set time later to send all the commands to do the read out of the FPIX1. Upon receipt of these commands, the FPIX1 begins to send the data generated in response to the injected charge pulses to the logic analyzer. The program waits until this transfer is completed, then reads all the data from the logic analyzer and starts to translate the binary code it provides into a list containing the column and row number from the cells that they are indicating hits. The program repeats this whole procedure over all the 2880 pixels cells in step of five. At the end it generates a list of all pixels that are working, from which we can tell which cells are working and which cells are not.

The software written to perform the noise and threshold uniformity tests for the selected 170 cells on the FPIX1 is divided into two large parts. The first one has the responsibility to acquire all the data from the FPIX1 and the second one performs the analysis of all the data generated to get the mean noise and threshold (defined as the voltage at which we are 50% efficient) with their

corresponding standard deviations. The data acquisition software is almost the same as for the bad pixel map test. The differences between them are that in the noise and the threshold test we need to inject a pulse of charge 1000 times and read all the data generated by the FPIX1 by means of the logic analyzer. These data are then processed to count the number of times that each cell registers a hit. The software scans the 170 cells in steps of five and it repeats the same procedure for each level of voltage selected. The total number of levels for the tests reported here was 123. The data acquired is stored in 42 files that the user specifies with a group name only at the beginning of the program, because the program itself generates an ascending number file, incrementing by one for each new set of five cells. All these files contains the column and row number of each tested cell with its total hit count for each level of voltage delivered from the function generator. The time that take to acquire the data from the 170 cells is 2 days. Note that it would take longer if we did not do five cells at a time and if we did not use two data generators to avoid having to repeat downloading all of the information for each new test. The data processing software calculates the efficiency as a function of the voltage peak-peak by counting the number of hits (of 1000 possible) received. We then perform a least squares fit using the nonlinear Levenberg-Marquardt (Lev-Mar) fitting technique on the data for each cell to obtain the best threshold and noise value. The program then performs a statistical analysis using the values from the 170 cells to get the means and standard deviations of the threshold and noise in units of electrons. At the end the program delivers histograms of the noise and thresholds for the 170 cells along with the corresponding gaussian curve.

4. TESTS RESULT ON THE FPIX1 CHIP

The bad pixel map test showed that all the cells above row 100 for all the columns are not working. This problem is understood and can be corrected. The threshold uniformity test showed a mean threshold of 9133e with a standard deviation of 292e. The noise test showed a mean noise of 58e with a standard deviation of 7e. Histograms of the data for the 170 cells along with the corresponding gaussian curves are shown as Fig. 4.

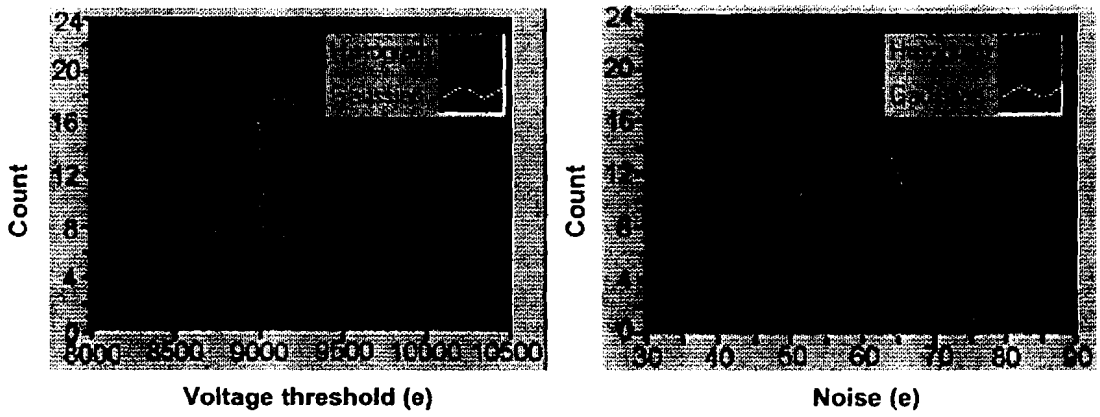


Figure 4. Test setup to get the threshold uniformity and noise from the FPIX1 chip.

5. REFERENCES

1. S. Stone., *The BTeV program at Fermilab*, BTeV pub.note., Fermilab, July 1998.
2. G. Cancelo., D. Christian., J. Hoff., A. Mekkaoui., R. Yarema., S. Zimmermann., *High readout speed pixel chip development at Fermilab*, Fourth Workshop on Electronics for LHC Experiments, 528-532, (1998).
3. National Instruments Corporation, *LabVIEW Users Manual*, (1998)

Date: Mon, 13 Nov 2000 16:50:47 -0500
From: Pat Moore <pmoore@engin.umich.edu>
To: mavargas@fnal.gov
Subject: Acceptance of your (KNO-00-617)

13 November 2000

M. C. Aurora Vargas-Trevino
Fermilab
MS 368
POB 500
Batavia IL 60510

Re: TITLE: The development of software to characterize the Fermilab pixel readout chip, FPIX1, for the BTeV experiment
AUTHOR: M. A. Vargas, et al.
REG. No.: KNO-00-617 R2

Dear Dr. Vargas:

Thank you for returning your revised manuscript to us and for addressing the reviewer comments. Your manuscript has now been accepted and we are sending this final version to the publisher in Amsterdam for future publication in NUCLEAR INSTRUMENTS & METHODS IN PHYSICS RESEARCH (Section A).

Except for your acknowledgement of this acceptance message, further communication about this paper should be addressed to:

Elsevier Science BV
NIMA Issue Management
POB 2759
1000 CT Amsterdam
THE NETHERLANDS
Fax: 31 20 485 2704
E-mails: authorsupport@elsevier.nl, nima-j@elsevier.nl

Please acknowledge receipt of this acceptance via return E-mail.

Best regards,
Glenn F. Knoll
Editor, NUCLEAR INSTRUMENTS & METHODS IN PHYSICS RESEARCH (Section A)
University of Michigan
1906 Cooley Bldg 2104
2355 Bonisteel Blvd
Ann Arbor MI 48109-2104
USA

The Development of Software to Characterize the Fermilab Pixel Readout Chip, FPIX1, for the BTeV Experiment.**

M. A. Vargas [a,b,c,*], M. Sheaff [d], S. Vergara [a,b,c,*]

[a] *Fermi National Accelerator Laboratory, PO Box 500, Batavia, IL, 60510, USA*
[b] *Faculty of mathematical and physical sciences-BUAP, Apdo. Postal 62-570, Av. San Claudio y 14 Sur, 72570, Puebla Pue, Mexico*
[d] *Physics Department, University of Wisconsin, Madison, WI, 53706, USA*

Abstract

Fermilab has designed and assembled a pixel-readout-chip sub-assembly containing five FPIX1 chips with flexible cable interconnections with which to address the technical issues involved in system integration for the proposed BTeV pixel detector. The module contains a total of 14400 pixel cells that need to be characterized in order to test the entire module. Software has been developed within the LabVIEW framework to control a set of instruments to perform threshold and noise tests on all five readout chips. These tests take only a few hours to run.
PACS: 07.05.-t; 07.05.Hd; 07.05.Kf

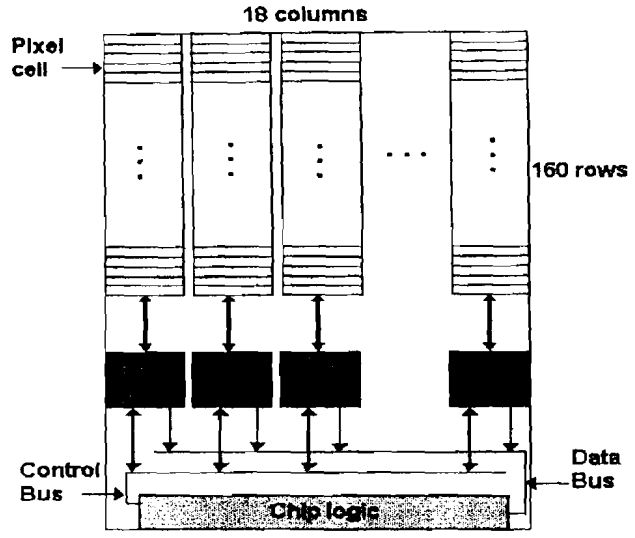
1. Introduction

We have developed a test stand for use in characterizing the FPIX1 front-end electronics chips that are one version in the FPIX sequence of VLSI chips being developed at Fermilab [1] for readout of the BTeV [2] pixel detector. This detector will provide high-resolution space points near the interaction region for use in reconstructing tracks and vertices. The information the detector provides will be used in the first level trigger to select events that have a high probability to contain secondary decay vertices. This means that all of the hit information from every beam crossing must be made available to the trigger processors. A beam crossing occurs every 132 ns.

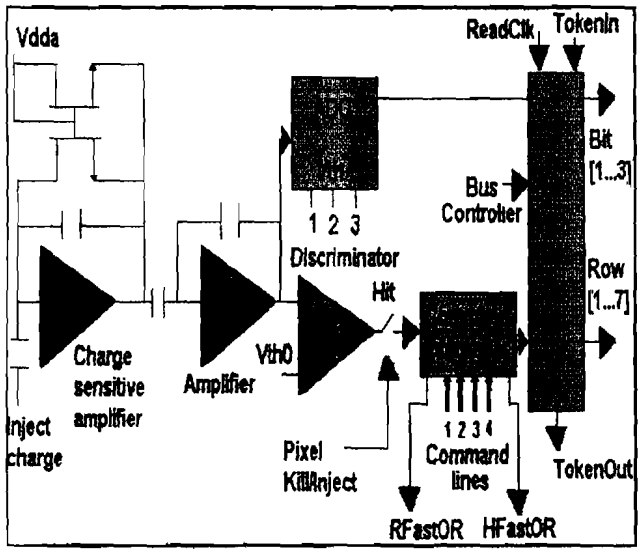
In order to achieve the required resolution, $\leq 9\mu m$, the pixel unit cells must not only be very small, $50\mu m$ by $400\mu m$, but the charge deposited in each must also be digitized and read out. The FPIX chips contain front-end electronics cells with the same dimensions as the pixels on the sensors. The number of FPIX chips which will be bump-bonded to each sensor will depend on the number of cells on each FPIX chip. The FPIX1 version of the readout chips contains 2880 cells. Future iterations are expected to contain an even larger number.

If the readout is to be accomplished in the short time between crossings, the information must be sparsified so that only valid hit data is presented to the trigger processors. Which cells are read out is determined by a discriminator in each cell. If a signal above threshold is detected in the cell, then it is read out. The threshold setting for all cells on a single FPIX chip is the same. On FPIX1 it is set by a voltage input, called

** *Work supported by the U.S. Department of Energy under contract No. DE-AC02-76CH03000.*
[c] *Scholarship holder of CONACyT, MEXICO.*
* *Corresponding author. Tel.: (630) 840 2261; fax: (630) 840 8208. E-mail address: mavargas@fnal.gov.*



a)



b)

Figure 1. a) FPIX1 chip; b) Front-End of the FPIX1 chip.

The digital interface has two major components, the command interpreter and the pixel token and bus controller. The command interpreter has four inputs corresponding to the four EOC command sets that are part of the chip control logic. Four independent command sets are provided for de-randomization. These commands allow the front ends to record hits from the current beam crossing (BCO) while hits from previous beam crossings are being read out. Commands are presented by the EOC logic simultaneously to all pixel cell interpreters in a column. When an interpreter is executing the input command and the hit output from the discriminator is asserted, the interpreter associates itself with the particular EOC set, and simultaneously it alerts the EOC logic to the presence of a hit

via the wire-or'ed Hit Fast Or (HfastOR) signal. After that, the information is stored in the cell until the same EOC set issues an output or reset command. When the command is an output command, the interpreter issues a bus request and asserts the wire-or'ed Read Fast Or (RfastOR) signal.

The readout of hits then proceeds synchronously with the Read Clock (ReadClk). The EOC logic provides a column token; the token quickly passes pixel cells with no hit information until it reaches a cell that is requesting the bus. The cell is then read out. The data is composed of the ADC count bits [3:1] and the row address radd [7:0]. As the hit pixel is read out it automatically resets itself and withdraws its assertion of the RfastOR. This signal returns to its inactivated state while the remaining hit pixels are being read out [1].

The FPIX1 chip has 87 pads for external control and data readout. There are eight internal scan paths formed by shift registers with different depths, which we need to fill with information in order to program the chip. Two scan paths are the mask register path, which is used to set the programmable reset, and the mode path which contains the chip mode (triggered or continuous), the chip ID and the BCO Time Stamp. The last two scan paths are the Kill/Inject paths, which select the pixel cell or cells to be characterized. These last two paths go through all pixel cells in the module and are filled by 2880 times 5 shifts of data.

3. Test Procedures

a) Overview

The software we have developed takes advantage of the very flexible environment provided by LabVIEW as well as the advanced data analysis features it provides. While the use of LabVIEW and the relatively slow General Purpose Interface Bus (GPIB) for data acquisition meant that the tests were potentially time-consuming, several time-saving measures were taken that increased testing speed considerably.

First, two personal computers are employed, operating in parallel, both running LabVIEW programs under the Windows NT operating system. The first, PC1, is programmed for data acquisition. The second, PC2, performs the data analysis. The use of a logic analyzer with a Local Area Network (LAN) connection to receive the raw data and pass it on to PC2 makes it possible for the two computers to work simultaneously during the tests.

Second, two data generators are used to send commands to the chips and to the various instruments. This means that only small changes are needed on each cycle and thus a small amount of new information needs to be downloaded per data read cycle. Because of the many cycles involved, this results in a very large reduction in time. The data acquisition program running in PC1 synchronizes the action of the two.

We have demonstrated that an adequate characterization of each chip, by which we mean measurements of discriminator threshold uniformity and electronic noise dispersion, can be achieved by charge injecting 10% of the cells in a grid that covers the entire chip uniformly. The threshold and noise results vary randomly within the errors over the chip and do not appear to depend on the location of the cell being tested. We performed an initial test using a very low threshold on all cells to see which cells were working. This test revealed that the last 60 rows in each column did not record hits. This is a feature of the layout that will be corrected on the next submission. Furthermore, the first column on these prototype chips was made to have outputs that could be visualized on an oscilloscope and could not be read out by the EOC logic. Thus there are actually only 1700 cells to be characterized on each FPIX1 chip (17 columns times 100 rows) in the MCM. Ten percent of this is a total of 170 cells to be charge injected on each of the

five chips. Again, with time constraints in mind, we inject these 17 at a time, making sure that the ones we are injecting at the same time are separated by sufficient distance (20 cells) to avoid crosstalk.

Figure 2 contains photographs of the components of the FPIX1 test stand. Clockwise, starting at the upper left, we show the arrangement of the various instruments, an individual FPIX1 chip, the module with five chips mounted on it, and the final module assembly mounted on the test board that was designed to allow signals to pass to and from the module for communication with the test instruments.

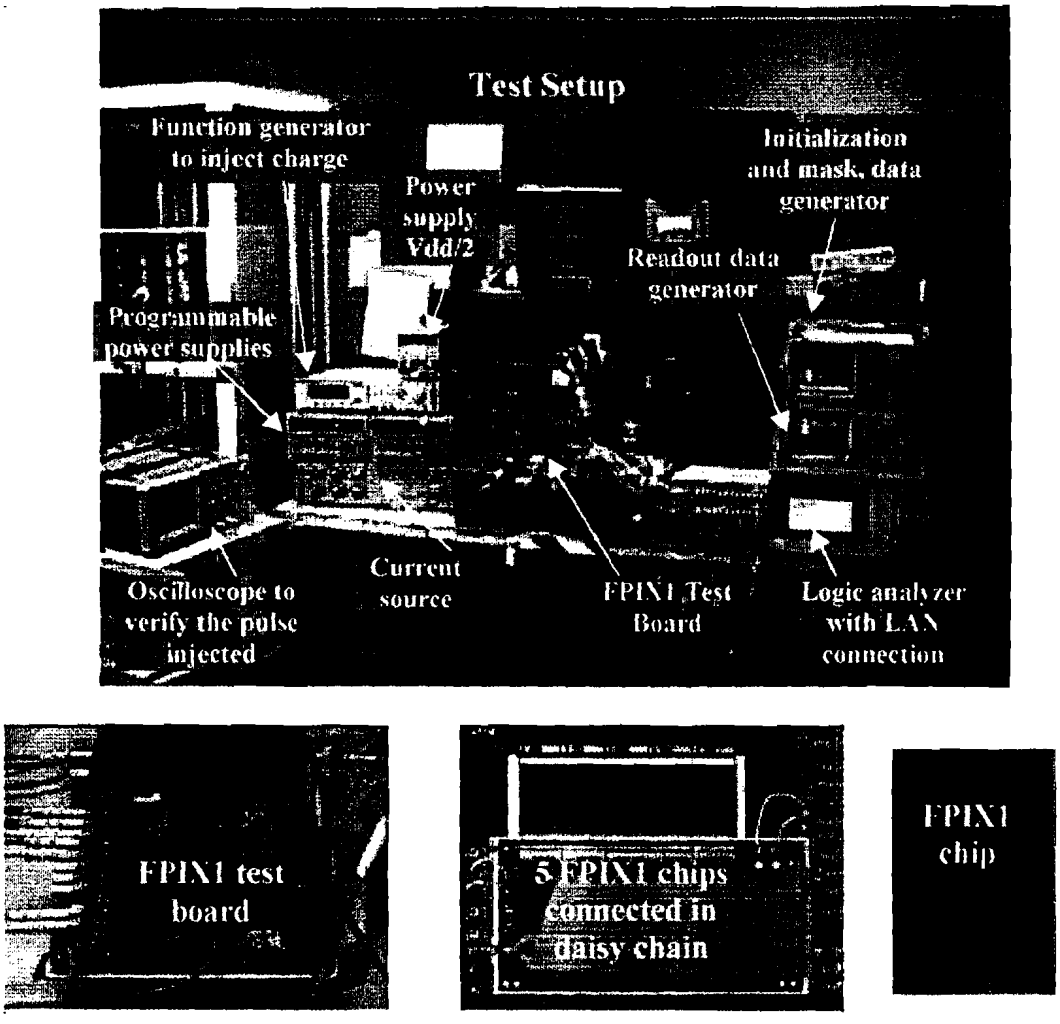


Figure 2. Test stand used to characterize the Fermilab pixel module.

b) Data Acquisition

PC1 contains an AT/TNT-GPIB interface to control the following instruments: Two Tektronix PS2520G programmable power supplies, a Hewlett Packard HP16500C logic analyzer system, a Stanford Research Systems DS345 function generator, two SONY/Tektronix DG2020A data generators and two SONY/Tektronix P3420 programmable outputs.

The test proceeds as follows: The program running in PC1 first initializes all the instruments used in the test and sets them to their nominal values. The two programmable power supplies provide power to the digital part of the FPIX1 chips, the analog part of the FPIX1 chips, and to all five Vth0 terminals of the chips on the module. These represent one of the inputs to the discriminators in the front ends. There are also two other instruments that are set manually. These are a regulated power supply, LEADER 718-SD, which provides power to the vdd/2 input of the FPIX1 to enable single-ended readout, and a current supply made at CPPM, Marseilles, to provide the feedback and Master Bias currents.

PC1 then initializes the five FPIX1 chips using the first of the two data generators. First, it configures the chips in continuous mode, and, since we have several chips connected in daisy chain, it gives each of the chips a unique ID number (ChipID). The FPIX1 chips deliver their information in two words, the data word and the control word. The data word contains the ADC value, the column and the row; the control word contains the ChipID number and the BCO number, which for these tests was always zero. PC1 then programs the five FPIX1s to kill (disable) all the pixels cells except 17 by means of the first data generator. The same data generator is then used to enable charge injection to the 85 cells (17 times 5) it has selected. Next PC1 sends commands to the logic analyzer to initialize it and to tell it to wait for a trigger that is based on bits in the data format that the chips deliver.

Following this, PC1 uses the second data generator to send a trigger to the function generator DS345, which injects charge to each of the enabled cells. Any cell in which this charge produces a signal at the output of the second amplification stage that is greater than Vth0 will have a hit in the output of its discriminator. The cell then asserts the wire or'ed HfastOR and stores the hit information until the EOC set issues an output command. When this occurs the command interpreter issues a bus request, asserts the wire or'ed RfastOR signal, and presents its data on the bus.

The second data generator has also been programmed to send the commands to perform the read out of the FPIX1 chips a fixed time after the charge is injected. Upon receipt of these commands, the FPIX1s send the data generated in response to the injected charge pulses to the logic analyzer in the order in which the token passes through them. The logic analyzer acquires the information in binary format and transmits it by means of a LAN connection to the other computer (PC2), which processes the data. For each set of 85 pixel cells PC1 repeats the injection of the same charge 500 times and does this for 60 different values of the charge it injects. When all the levels of charge have been scanned, the program changes the selection of the pixel cells and repeats the whole procedure until all 170 pixel cells on each of the five FPIX1 chips have been scanned.

The levels of charge must be chosen to bracket the voltage threshold. To do this we make a preliminary scan over the chip to determine the optimal range for the test. For each test we fix Vth0 at a particular value. We have carried out these threshold uniformity and noise tests in four different regions of operation.

c) Data Analysis

PC2 processes the information collected to obtain the mean threshold with its corresponding standard deviation (threshold dispersion) and the mean noise with its corresponding standard deviation for each FPIX1 chip on the MCM. When PC1 begins to acquire data, PC2 waits until the transfer of information from the first 85 pixel cells finishes, which takes approximately 30 minutes. After this, both programs run at the same time. The data acquisition wait time is 35 minutes (adding an extra 5 minutes more to avoid any potential conflicts). The data processing program needs to wait for this time

only once, because after the first transfer of data both programs can run simultaneously. The information processing takes exactly 30 minutes, so that the times are well matched. Thus, neither PC sits idle for an appreciable time during the test. While PC1 is acquiring the data from the current 85 pixel cells, PC2 is processing the information from the previous 85 pixel cells.

Information processing proceeds as follows: First, PC2 translates the binary code that the data acquisition system has acquired into a file that contains the ChipID number, the column number, the row number, the count of hits recorded above threshold and the voltage at each step, as well as the count of total hits (charge injections). Then it calculates efficiency for each pixel cell by dividing the number of counts recorded over the total number of hits. The discriminator output in the front-end of the FPIX1 chip is characterized by measuring its response as a function of the charge injected. Although only a single Vth0 is input to all 2880 cells on a chip, the value of the charge that triggers the discriminator may vary from cell to cell. This represents the threshold dispersion. The electrical charge injected can be calculated using the relationship between voltage and charge produced on the input capacitor. Since we have a capacitive load, C, of 6fF, the charge, Q_{inject}, is

$$Q_{inject} = VppC = Vpp \times 6 \times 10^{-15} coulombs, \tag{1}$$

where *Vpp* represents the voltage peak-peak delivered by the SRS DS345 function generator. The number of electrons that correspond to 1mV can be obtained by means of the following relationship:

$$(Q_{inject})e = \left(\frac{VppC}{e^-}\right)e = \left(\frac{Vpp \times 6 \times 10^{-15}}{1.6 \times 10^{-19}}\right)e, \tag{2}$$

with the result that

$$1mV \rightarrow 37.5e. \tag{3}$$

Because of the inherent noise of the electronics, the efficiency as a function of *Vpp* has the properties of an error function. For each pixel cell, a fit is performed to obtain the error function that best describes the experimental points. LabVIEW has a library of functions including one that performs a fit using the nonlinear Levenberg-Marquardt (Lev-Mar) method to determine a nonlinear set of coefficients which minimize chi-square [5]. First, PC2 plots the measured efficiency versus voltage *Vpp* (charge injected). The initial values of the threshold and noise are obtained using the data. The threshold value, Vth_{exp}, is chosen to be the first value of the voltage that has greater than 50% hit efficiency. For the noise value, PC2 finds the first voltage for which the hit efficiency is greater than 81.5% and subtracts the value for 50% hit efficiency, which yields σ_{exp}. These experimental values are used as the initial guess coefficients for the nonlinear Lev-Mar Fit technique, since the fits will converge more quickly the closer the initial values are to the solutions.

The nonlinear function needs to be specified, i.e., the relationship that describes the error function for each cell as a function of *Vpp*. For each point, *Vpp*, along the curve:

$$errf = \frac{1}{\sqrt{2\pi\sigma^2}} \int_{-\infty}^{Vpp} e^{-\left[\frac{(V-Vth)^2}{2\sigma^2}\right]} dV \tag{4}$$

This technique provides two ways to calculate the Jacobian; i.e., the partial derivatives with respect to the coefficients, needed in the algorithm. These methods are:

Numerical calculation: Uses a numerical approximation to compute the Jacobian.

Formula calculation: Uses a formula to compute the Jacobian. For this one needs to specify the Jacobian functions, which are the partial derivatives of the error function with respect to σ and with respect to *Vth*. These functions are:

$$\frac{\partial errf}{\partial \sigma} = \left(\frac{(V-Vth)^2 - \sigma^2}{\sqrt{2\pi}\sigma^4} \right) \int_{-\infty}^{Vpp} e^{-\left[\frac{(V-Vth)^2}{2\sigma^2}\right]} dV \tag{5}$$

$$\frac{\partial errf}{\partial Vth} = \frac{(V-Vth)}{\sqrt{2\pi}\sigma^3} \int_{-\infty}^{Vpp} e^{-\left[\frac{(V-Vth)^2}{2\sigma^2}\right]} dV \tag{6}$$

The program repeats the above procedure for all of the pixel cells being tested. When the information for all the pixel cells has been analyzed, the results are output in five files, each of which contains the final information for all cells on one chip in the MCM, i.e., the column number, the row number, the best fit *Vth* in units of volts and electrons, and the best fit σ (noise) in units of volts and electrons. We then run a second program to histogram these quantities for the 170 cells on each of the five chips and to obtain the Gaussian curves that best fit these histograms for each file. We thus obtain the mean threshold, the sigma threshold (threshold dispersion), the mean noise and also the noise dispersion, all in units of volts and electrons for the 170 cells on each FPIX1 chip. These results represent the characterization of the chips on the module. A flow diagram of both programs, i.e., the data acquisition program and the data processing program is shown as Fig. 3

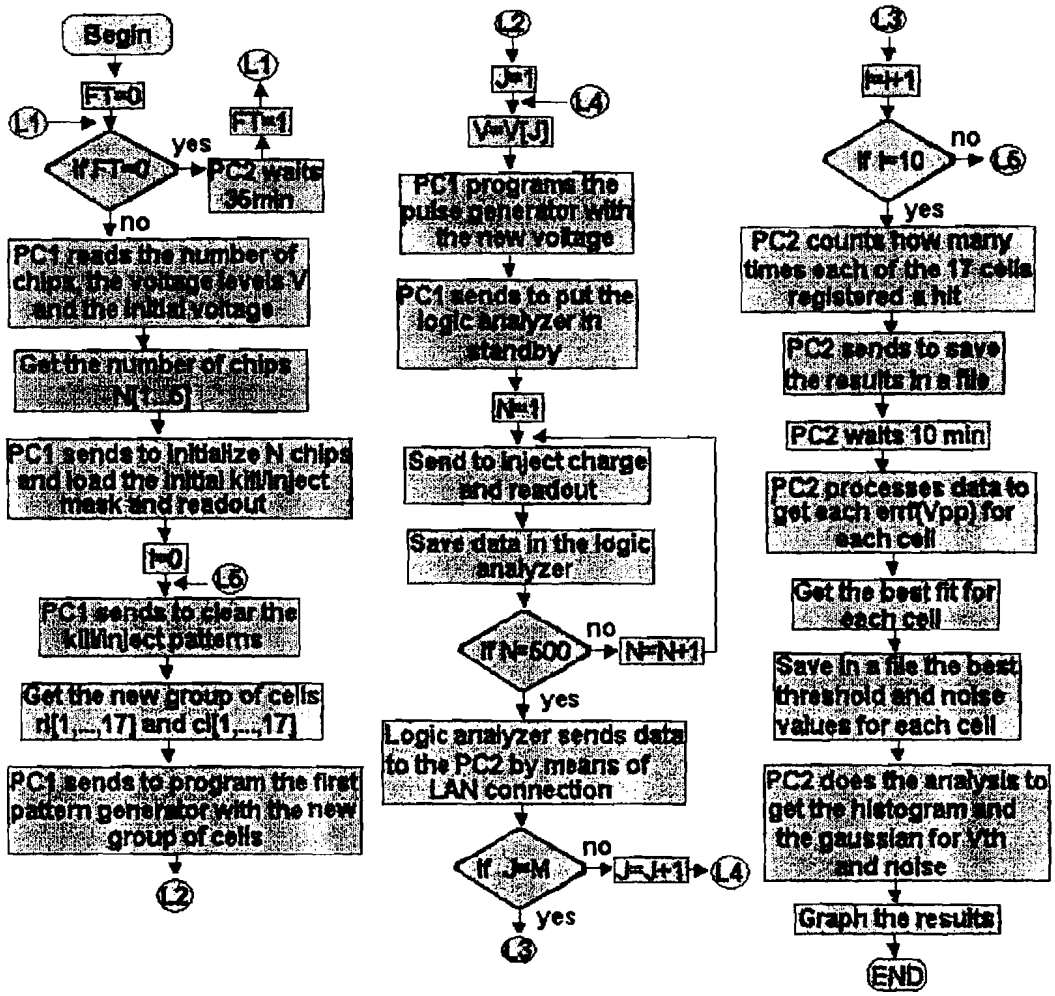


Figure 3. Flow diagram of the software designed.

3. Results

The software designed to characterize the MCM yields the threshold uniformity and noise dispersion for the five FPIX1 chips on the module in 5 hours of running with 500 hits (charge injections) per cell at each of 60 values of *Vpp*. This time could be reduced by as much as a factor of five by using PCs with faster processing speeds that are already available in the marketplace. The time required for data acquisition and analysis has been considerably decreased by taking advantage of the flexibility of the LabVIEW programming environment and by using the LAN connection to transfer the data from the logic analyzer to a second computer for analysis.

PC2 processes a very large amount of information. The raw data for 850 pixel cells (170 cells times five chips) is organized into 10 files, each containing the information for a group of 85 pixel cells. An individual file contains 931 KB of data. In a particular characterization we have to scan 60 levels of voltage in order to cover the full range. The total information acquired is obtained by multiplying the size of each file by the total number of files acquired, which in this case is 600 files. Thus, the program processes a

total of approximately 550 MB of information to carry out the characterization in each range of operation.

Fig. 4 shows the efficiency versus *Vpp* for a single pixel cell and the error function that results from the fit to the data. Figures 5 and 6 show the histograms of the threshold and noise dispersion and curves corresponding to the Gaussian fits for one chip mounted in the MCM. Table 1 and Table 2 contain the results of the characterization for the five FPIX1 chips for four regions of operation.

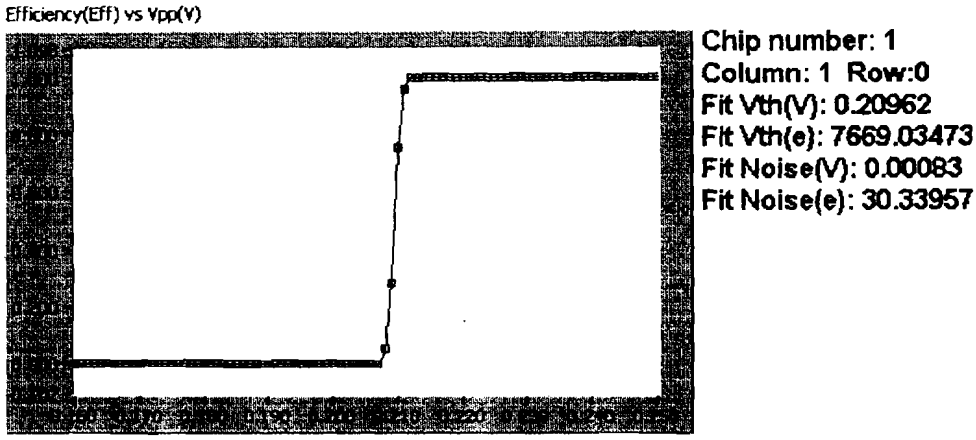


Figure 4. Error function from one pixel cell on one FPIX1 chip.

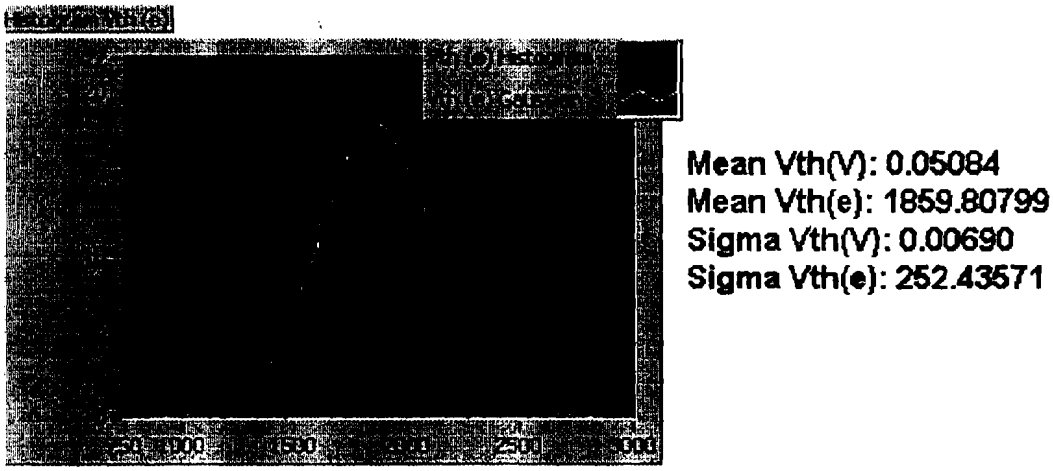


Figure 5. Voltage Threshold distribution in the chip number 1.

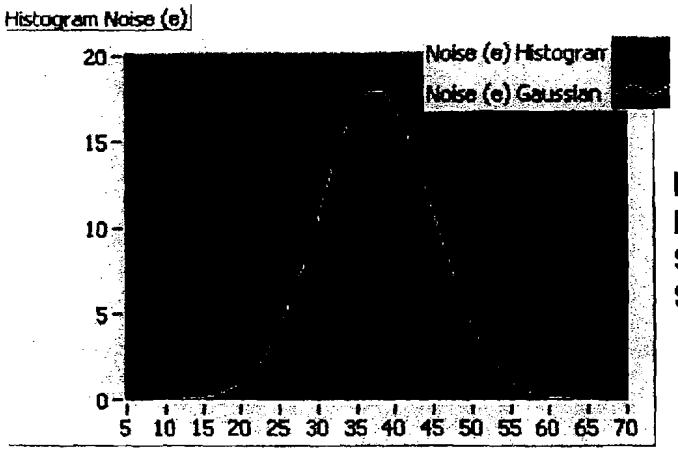


Figure 6. Noise distribution in the chip number 1.

	Mean threshold (e-)				Sigma threshold (e-)			
Vth0 (V)	**1.95**	**2.05**	**2.15**	**2.25**	**1.95**	**2.05**	**2.15**	**2.25**
Chip 1	7321.1	5437.2	3660.4	1859.8	221.7	240.2	247.2	252.4
Chip 2	7400.3	5506.4	3720.7	1919.5	189.9	208.8	213.6	224.9
Chip 3	7289.9	5402.2	3623.3	1830.8	209.1	219.1	227.0	238.4
Chip 4	7355.3	5472.7	3691.4	1895.2	187.5	180.9	186.3	197.1
Chip 5	7354.3	5430.9	3616.7	1770.7	217.6	235.1	246.6	264.3

Table 1. Final results of the characterization for threshold uniformity on the MCM for four regions of operation.

	Mean Noise (e-)				Sigma Noise (e-)			
Vth0 (V)	**1.95**	**2.05**	**2.15**	**2.25**	**1.95**	**2.05**	**2.15**	**2.25**
Chip 1	39.3	37.8	37.9	37.5	8.7	7.8	7.4	7.3
Chip 2	36.5	36.3	36.7	36.6	8.8	7.7	7.9	7.2
Chip 3	38.0	38.2	37.7	37.7	9.8	9.1	8.9	9.2
Chip 4	32.2	32.0	31.6	31.1	8.9	7.9	7.9	7.2
Chip 5	35.3	35.6	36.0	37.6	10.2	9.0	9.8	9.9

Table 2. Final results of the characterization for noise on the MCM for four regions of operation.

Acknowledgements

The authors wish to thank S. Zimmermann for valuable advice during the course of this project and G. Chiodini and G. Cardoso for technical assistance. We also want to express our appreciation to the Fermilab staff and management, with special thanks to J. Appel, S. Kwan, and E. Barsotti. This work would not have been possible without the support and encouragement of A. Cordero and A. Fernandez of FCFM/BUAP. The support of the U.S. DOE and CONACyT, Mexico, is also gratefully acknowledged.

References

[1] G. Cancelo, D. Christian, J. Hoff, A. Mekkaoui, R. Yarema, S. Zimmermann, *High readout speed pixel chip development at Fermilab*, Fourth Workshop on Electronics for LHC Experiments, 528-532, 1998.

[2] J. Butler, S. Stone, et al., *Proposal for an experiment to measure mixing, CP violation, and rare decays in Charm and Beauty particle decays at the Fermilab collider- BTeV*, submitted to the Fermilab Program Advisory Committee, May 2000. Stage I approval given June 27, 2000.

[3] B. Chou, et al., *Ultra-High-Density Interconnect Flex Substrates*, HDI Magazine, December 1998.

[4] National Instruments Corporation, LabVIEW Users Manual, 1998.

[5] National Instruments Corporation, *LabVIEW Function Reference Manual*, 1998

Development of High Data Readout Rate Pixel Module and Detector Hybridization at Fermilab

S.Zimmermann, S.Kwan, G.Cancelo, G.Cardoso, S.Cihangir, D.Christian, R.Downing, J.Hoff, P.Kasper, A.Mekkaoui, A.Vargas*+, R. Yarema

Fermi National Accelerator Laboratory, Batavia, IL 60510 (USA)
**Universidad Automona de Puebla, Puebla, PUE 72570 (Mexico)*

Abstract
At Fermilab, both pixel detector multichip module and sensor hybridization are being developed for the BTeV experiment. The base line design of the module and preliminary results of characterization tests are presented.

1. Introduction

At Fermilab, the BTeV experiment has been proposed for the C-Zero interaction region of the Tevatron [1]. The innermost detector for this experiment will be a pixel detector composed of 31×2 pixel planes of approximately 100×100 mm each, assembled perpendicularly to the colliding beam and installed a few millimeters from the beam. Each plane is formed by sets of three different lengths of pixel hybridized modules, each composed of a single active-area sensor tile and of one row of pixel readout (RDO) integrate circuits (ICs).

The pixel detector will be employed for the lowest level trigger system, hence, the pixel readout ICs will read out all detected hits. This requirement imposes a severe constraint on the RDO IC, hybridize module and data transmission rate to the data acquisition system. Several factors impact in the amount of data that each RDO IC needs to transfer: IC active area, distance from the beam, data format, etc. The final size of the RDO IC is still under consideration.

The BTeV pixel detector is based on a design relying on a hybrid approach. With this approach, the RDO IC and the sensor array are developed separately and the detector is constructed by flip-chip mating of the two together. This offers maximum flexibility in the development process, choice of fabrication technologies, and the choice of sensor material.

2. Proposed Pixel Module

Figure 1 shows a sketch of the proposed module (top and side views). The module is composed of three layers. The lowest layer is formed by

Work supported by the U.S. Department of Energy under contract No. DE-AC02-76CH03000.

+ Scholarship holder of CONACyT, Mexico.

the RDO ICs. The back of the ICs are in thermal contact with the supporting structure while the other side is bump-bonded to the pixel sensor. The clock, control and power pad interface of the RDO ICs extend beyond the edge of the sensor. The Kapton interconnect circuitry is glued on the top of this assembly and the RDO IC pad interface wired bounded to the circuit. The circuit then extends to one end of the module where the rad-hard module controller and high speed data serializers ICs and fiber optic connectors are assembled [2]. These components are located in this position so that they are outside the tracking volume. ATLAS and CMS explore similar solution [3, 4, 5].

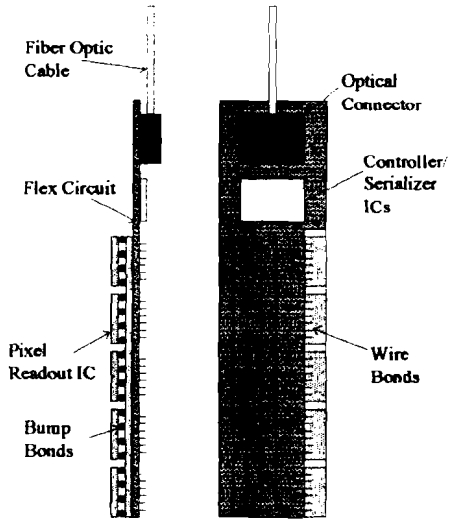


Figure 1. The Flex Kapton Module

The pixel module readout strategy is paramount to employ the pixel detector in the lowest level trigger. Our present assumptions are based on simulations that describe the track behavior inside the pixel detector. The parameters used for the simulations are: luminosity of 2×10^{32} $cm^{-2}s^{-1}$ (corresponds to an average of two interactions per bunch crossing), pixel size of 400×50 μm, threshold of 2000 e^- and magnetic field of 1.6 Tesla.

Figure 2 shows a sketch of the 40 ICs that may compose a pixel half plane. The beam passes on the place represented by the black dot. These numbers assume specific data format and IC size. The distance from the beam to the closest ICs is 6 mm.

Column with more hits

11	13	17	17	20	18	16	13	8
11	18	26	31	39	33	25	18	12
16	20	37	61	76	59	39	26	18
17	35	63	141	234	130	65	36	16
23	35	74	234					

Beam

Figure 2. Average Bit Data Rate, in MBits/sec

The column with more hits requires the biggest data bandwidth. Figure 3 shows the proposed block diagram of the circuit interconnect with the data rate for this column. The serializer IC handles 16 bit words at 60 MWords/sec. There is one dedicated 12 bit data bus connecting the pixel RDO IC with the highest data rate directly to the serializer, another 12 bit data bus that is shared by two ICs and finally a six bit data bus that is shared by the two remaining ICs. The six bit data bus is split between the two serializers. With this interconnect scheme there is an additional bandwidth to handle peak data rates and unpredicted hits between 3.1 and 4.6 times the required average bandwidth.

Clearly, a high density circuitry is necessary to interconnect the pixel RDO ICs with the controller and serializer ICs. The width of the circuit

trace area will be around 5 mm and we estimate that for clocks, controls and data we will need approximately 45 circuit traces running in parallel in the densest portion of the flex circuit. Therefore, each trace, vias and clearance has to fit in less than 110 μm. We are presently working with Fujitsu Computer Packaging Technologies (FCPT, San Diego) in prototyping such circuit. FCPT capabilities include flex circuits with line traces of 20 μm in a 40 μm pitch, copper line thickness smaller than 5 μm, vias spaced by 200 μm, via cover pads of 100 μm and average via hole diameter of 26 μm. FCPT can manufacture circuits with four copper layers or more using the *Z* via technique to interconnect flex circuit copper layer pairs [6].

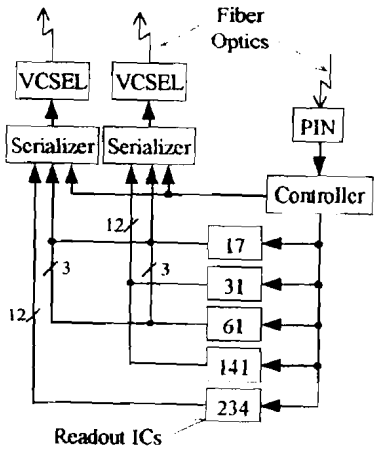


Figure 3. Pixel Module Block Diagram

Another approach to control and readout the RDO ICs is also under consideration. This option is a direct consequence of the BTeV detector layout [1]. The BTeV detector covers the forward direction, 10 to 300 mrad, with respect to both colliding beams. Hence, all volume outside this section is outside the active area and can be used to house readout and control electronics without interfering with the experiment. This other option takes advantage of this consideration. The idea is to move the serializer and controller logic 30 cm from the beam where the radiation dose will be less than 10 KRads. There we hopefully can employ components-off-the-shelve. The only IC on the pixel module would then be the RDO IC.

3. First Prototype Pixel Module

We have assembled and characterized the first prototype of the pixel module. It is composed of five FPIX1 RDO ICs [7] and a four layer high density flex circuit manufactured by FCPT. Figure 4 shows a close up of the ICs and the high density flex circuit (the sensor is not bump-bonded). In this prototype, different from the baseline design, the flex circuit is located on the side of the ICs.

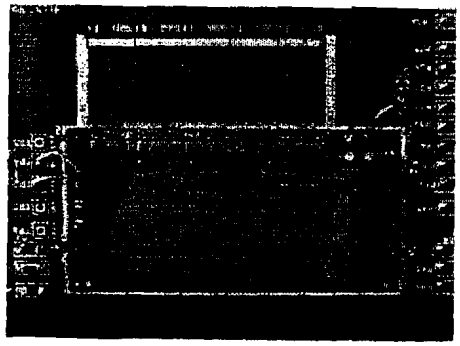


Figure 4. Prototype Pixel Module

The test stand is formed by standard instrumentation controlled by GPIB [8]. We have characterized threshold and noise of each of the four chips (one chip had failed) bump-bonded to the pixel sensor. The results for one specific threshold are summarized in Table 1. They were measured by injecting charge in the front end with a

pulse generator. The comparison of these results with the results of a single FPIX1 IC shows no noticeable degradation in performance [9]. Furthermore, tests with dead timeless mode, where the charge inject in the front end is time swept in relation to the readout clock also does not reveal any degradation in performance, strongly suggesting no crosstalk problems between the digital and analog sections of the FPIX1 and flex circuit.

Table 1. Performance of the Five Chip Module (in e^-)

Chip	1	2	3	4
Threshold	1649	1406	1589	2865
Threshold σ	254	307	222	250
Noise	62	53	49	62
Noise σ	16	13	11	11

4. Results of the hybridization to pixel sensors

As already stated, the hybridization approach pursued offers maximum flexibility. However, it requires the availability of highly reliable, reasonably low cost fine-pitch flip-chip mating technology. We have tested three bump bonding technologies: indium, fluxed solder, and fluxless solder. Real sensors and RDO ICs were indium bumped at both the single chip or wafer level by BOEING, NA. Inc (Anaheim, CA) and Advance Interconnect Technology Ltd. (Hong Kong) with satisfactory yield and performance. Figure 5 shows the hit maps of one FPIX1 detector using a radioactive source. All the channels seem to be working.

We have also conducted tests on dummy detectors to evaluate eutectic Pb/Sn solder. The vendor, MCNC (Research Triangle Park, NC), together with UNITIVE Electronics, produced the dummy parts, and then carried on with the bumping process. The detectors are composed of channels which are a number of daisy-chained bumps at 50 μm pitch connected to probe pads at an edge of the dummy detector. We characterized the bump yield by measuring the resistance of each channel, and (to check for shorts) the resistance between neighboring channels.

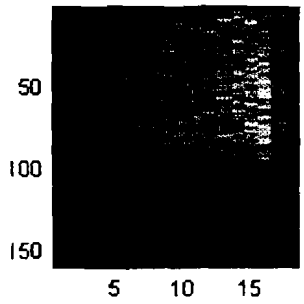


Figure 5: Hit map of one detector

Both fluxed and fluxless solder bumps have been studied. We found much better results using the fluxless process. The yield from the fluxed process is poor and the delivered parts have a lot of residue left behind from the cleaning of the flux. For the fluxless assemblies, a process called Plasma Assisted Dry Soldering (PADS) [10] is used. The bumped chip wafer (top plates of the dummies) and un-bumped substrate wafer (bottom plates of the dummies with only under-bump metallization put on) were diced and tacked together (flip-chip assembly) before being treated in the PADS process. The joins were then reflowed at 250°C. After being reflowed, the detectors were rinsed with methanol and dried in air. The

diameter of the bumps is ~40 microns, and the height is ~15 microns after mating. We estimate that the single solder bump resistance is less than 1 Ohm.

Two of the 82 detectors tested were misaligned to cause one bump shift resulting in open channels and shorted adjacent channels. Five detectors had over 50% of their channels open or at high resistance. One was sent to MCNC to be examined. It was taken apart and found to have the bumps on the top plate (chip) not touching the pads on the bottom plate (substrate). This was probably due to contamination or debris on the substrate on that particular location. All this results in an assembly yield of 91.5% (75 good out of 82). For the good detectors, a channel yield of 99.32% or 6.8×10^{-3} failed-channel/channel (106 open or high-resistance channels altogether), and with 14 or 16 bumps per channel and with the assumption that only one bump is bad, a bump yield of 99.95% or 4.5×10^{-4} failure/bump.

5. Conclusions

We have described the baseline design and a variation of the pixel module to handle the data rate required for the BTeV experiment at Fermilab. The present prototype has shown good electrical performance characteristics.

Indium bump bonding is proven to be capable of successful fabrication at 50 micron pitch on real detectors. For solder bumps at 50 micron pitch, much better results have been obtained with the fluxless PADS processed detectors. The results are adequate for our needs and our tests have validated it as a viable technology.

References

[1] Kulyavtsev, A., *et.al.*, BTeV proposal, Fermilab, May 2000.

[2] Cancelo, G.E., *et.al.*, "Fiber Optic based readout for BTeV's Pixel Detector," to be presented 6th Workshop on Elect. LHC Exp., Krakow, Sept. 2000.

[3] Richardson, J., "The ATLAS pixel on-detector electronics," 5th Workshop on Elect. LHC Exp., Snowmass, pp.83-87, Sept. 1999.

[4] Skubi. P., *et.al.*, "Flex circuits for the ATLAS pixel detector," Proc. Pixel98, pp.519-534, Batavia, IL, May 1998.

[5] Baur, R., "Readout architecture of the CMS pixel detector," see these proceedings.

[6] Chou, B., *et.al.* "Ultra-High-Density Interconnect Flex Substrates," HDI Magazine, Marietta, GA, Dec. 1998.

[7] Christian, D.C., *et.al.*, "Development of a pixel readout chip for BTeV," NIM A 435, pp.144-152, 1999.

[8] Vargas, A., "Development of software to characterize the Fermilab pixel readout chip, FPIX1, for the BTeV Experiment," to be submitted to NIM.

[9] Mekkaoui, A., *et.al.*, "FPIX2: an advanced pixel readout chip," 5th Workshop on Elect. LHC Exp., Snowmass, pp. 98-102, Sept. 1999.

[10] Koopman, N. *et.al.*, "Fluxless No-Clean Assembly of Solder Bumped Flip Chips," 46th ECTC, Orlando, Florida, May 28-31, 1996.

Fiber Optic based readout for BTeV's Pixel Detector

Gustavo I. E. Cancelo*, Sergio Vergara**, Sergio Zimmermann*,
Guilherme Cardoso*, Aurora Vargas**, Jeff Andresen*
* Fermilab, **University of Puebla, Mexico

Abstract

The current paper describes the design of Fiber Optics Links for BTeV's Pixel Detector readout. BTeV pixel detectors chips will be located as close as 6mm from the accelerator's beam into the vacuum pipe. The readout electronics will be located at about 6cm from the beam, imposing strong constrains regarding radiation, mass, power dissipation, and size. The current development is the first prototype designed to initialize, control and readout pixel detector chips using optical links. Results of link performance are shown. The current development is designed in two boards, which will become the major parts of a test stand for pixel detector bench and beam tests.

I. INTRODUCTION

BTeV's pixel detector consists of 31 double-plane stations of about 100 cm^2 of active detection area. These planes are perpendicular to the direction of the beam. The beam passes through the center of each plane formed by two halves. One of the half planes is shown in Figure 1. Since BTeV will use the pixel detector as part of the lowest level trigger system, one of the most important requirements is hit readout speed [1]. The primary goal is to achieve a data transfer rate capable of handling the hit rate generated by Fermilab's Tevatron beam with a luminosity of $2 * 10^{32}$ p/cm^2 and a bunch crossing (BCO) time of 132 ns. Furthermore, the required readout bandwidth must be achieved while keeping a small power and mass budget. In particular, mass is very critical for the Pixel Detector, the most inner part of BTeV's detector where multiple scattering must be minimized.

A fiber optic based design, as proposed in this paper, is the technology that best adapts to BTeV's requirements. Every pixel plane will generate up to 16 Gb/s of data. The pixel amplifier and discriminator chips, located underneath the pixel detectors will store that information. However, since the pixel detector is the primary component of BTeV's trigger, the data must be readout as soon as possible. A multi chip module (MCM) design is being proposed for the pixel detector electronics as shown in Figure 1. Every module is autonomous. It groups a certain number of Pixel amplifier/discriminator chips and the readout electronics to transfer the data from the pixel planes to the trigger processor and DAQ. Furthermore, every module must allow for an incoming link to receive commands to initialize and control the pixel devices and provide them with timing information (i.e. clocks). A second approach under consideration moves the Pixel Detector Fiber Optic components 25 cm away of the MCMs. The advantage here is that the optoelectronics, and specially, the serializers and decoder chips receive much less radiation. Furthermore, it decreases the amount of mass in the active region.

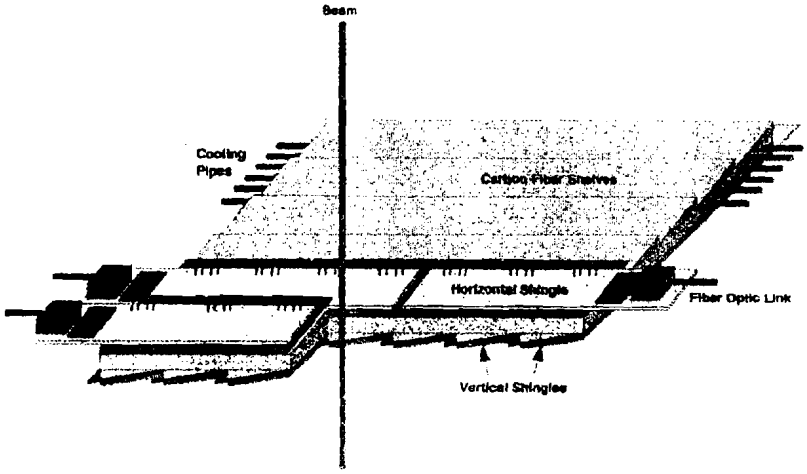


Figure 1: BTeV's Pixel Detector Plane

II. THE CONTROL AND READOUT OPTICAL LINK PROTOTYPE

A Control and Readout Optical Link prototype has been designed with the following purpose:

- Test optical transmission and reception issues such as dynamic range, noise, biasing, bandwidth, optical power, etc.
- Test the bi-phase mark encoded signal concept for the FPIX pixel chip [2] initialization and control link.
- To provide a good step toward the system integration of BteV Pixel Detector's.

The Optical Link Control and Readout prototype is organized in such a way that builds up a test stand for Pixel Detector Modules. Figure 2 shows a block diagram of the system.

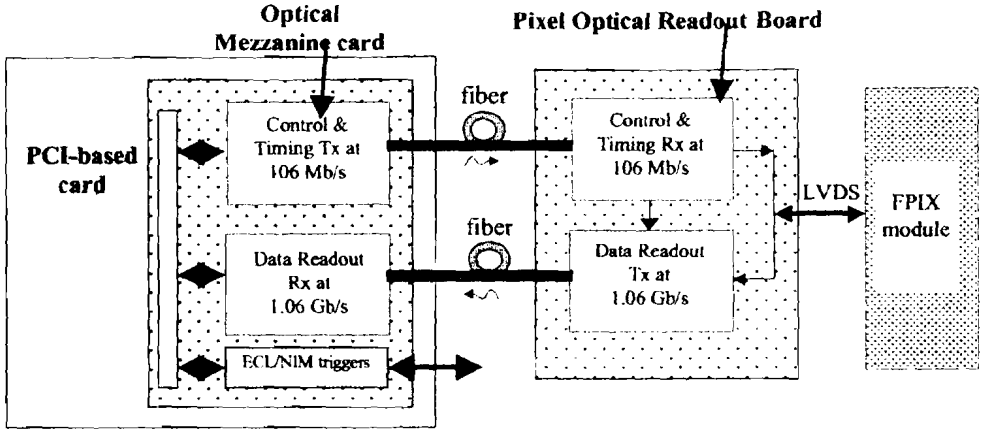


Figure 2: Two board Control and Readout Optical Link prototype

III. THE CONTROL OPTICAL LINK

The Control Optical Link carries initialization, control and timing information for the Pixel chips. The encoding used for this link is of bi-phase mark type. A 53 MHz clock modulates the initialization and control data in order to reduce the total number of fibers and provide an electrically balanced transmission. The bi-phase mark encoding guarantics at least one transition per bit reducing synchronization problems at the receiving end. As shown in Figure 3, the receiver uses a Phase Lock Loop (PLL) to recover the clock and data. The PLL must serve the double function of recovering the clock and reduceing the jitter. The Command interpreter decodes the serial information. Commands are of two different types, the ones used to initialize pixel cell and chip parameters and others to control or reset the FPIX chip in running mode. The commands are decode by the Command Interpreter and sent to the MCM as LVDS signals.

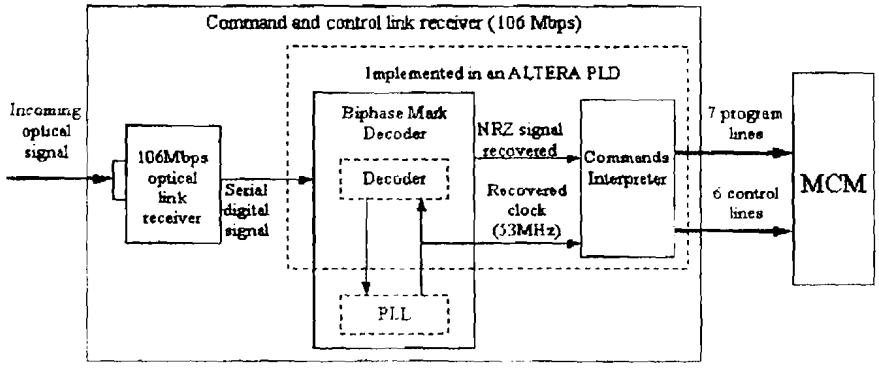


Figure 3: Control & Timing Receiver block diagram

The Control link receiver utilizes a Mitel 1A354 PIN photodiode operated in photovoltaic mode, connected to a high speed amplifier and discriminator as shown in Figure 4. The peek optical power in the fiber is 1mW. The PIN output signal is about 90mV and the output of the amplifier is 370 mV. Figure 5 shows scope images of the optical eye pattern and the bi-phase mark signal decodification at the receiving end. The amplifier used is high bandwidth allowing for a rise and fall time of about 1.5 ns. The PIN's response is very linear up to at least 1.5 mW.

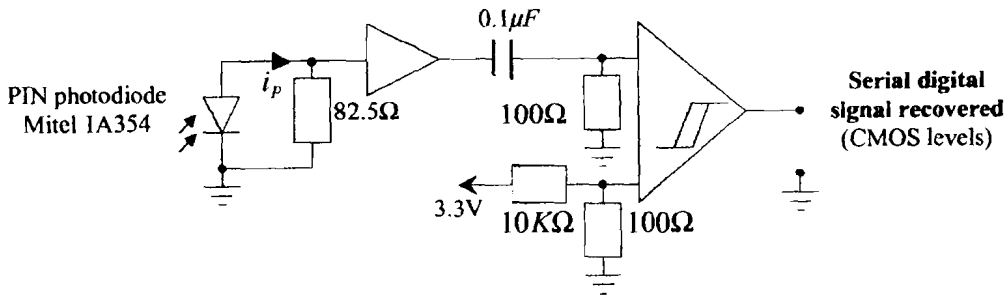


Figure 4: PIN, amplifier and discriminator

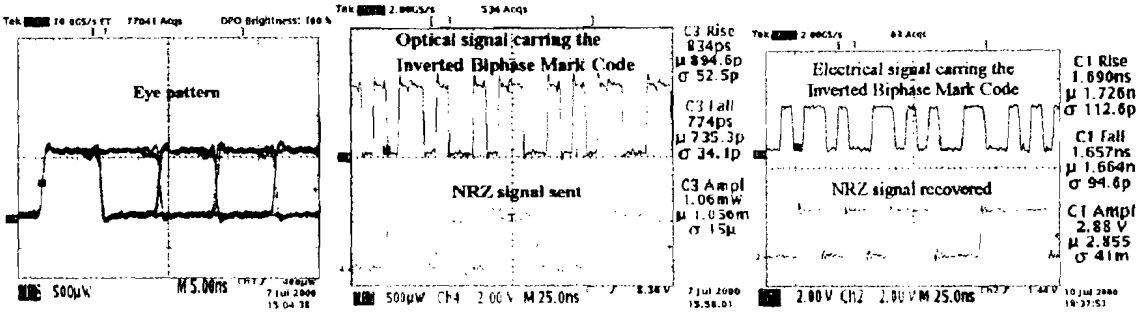


Figure 5: 53 MHz link signals: a) Eye pattern, b) NRZ-bi phase sent, c) NRZ-bi phase received

A strong constraint on the recovered clock is jitter, because the clock is used to readout pixels and to clock the Gigabit serializer. Gigabit serializers multiply the input clock frequency by 20 but they are unable to reject jitter. A small jitter may represent a big percentage of the output data's period, increasing the bit error rate in the channel. As it can be appreciated in Figure 6a, the jitter of the recovered clock in the present board is very low, about 64 ps peek-to-peek. The bit error rate of the channel has been measured to be better than 10^{-14}. Figure 6b shows the signal rise and fall time of the clock to be around 1.05ns.

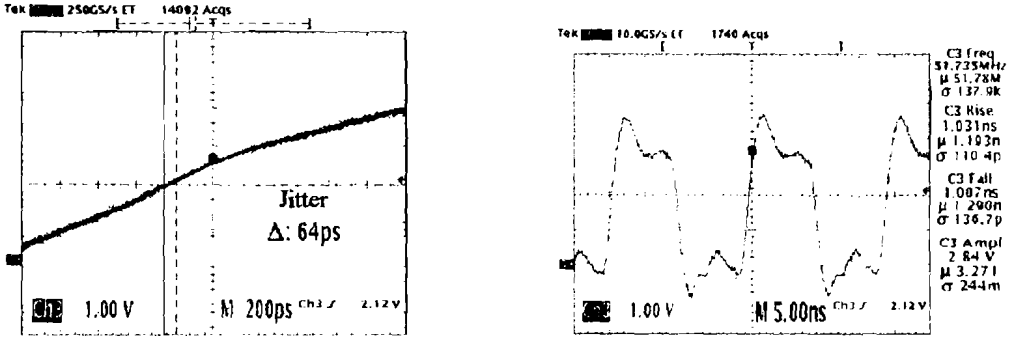


Figure 6: a) Clock's jitter, b) Clock's rise and fall time

IV. THE READOUT OPTICAL LINK

The readout electronic serializes the data from the 5-chip Pixel Module into a G-Link based serial link operating at 1.06 Gb/s. In the final design the G-Link will be replaced by a radiation hardened serializer [3]. Alternative approaches are also being analyzed to relocate the optoelectronics. Extending the LVDS signals from the FPIX module to about 25 cm will allow us to place the optoelectronics outside the high radiation area. Performance of the G-Link has already been reported in [4].

V. INTEGRATION OF THE OPTICAL LINKS TO THE READOUT OF A PIXEL MODULE

As shown in Figure 2 the optical links will be used to initialize and deliver commands and clocks to the Pixel chips as well as to readout the pixel data. BTeV Pixel Detector chip characterization has been extensively carried out at Fermilab and is reported elsewhere [5][6]. However, the optical links were integrated with the existing pixel assemblies. Several comparison tests were run using a single FPIX chip and a MCM with 5 pixel devices. Figure 7 shows the current setup.

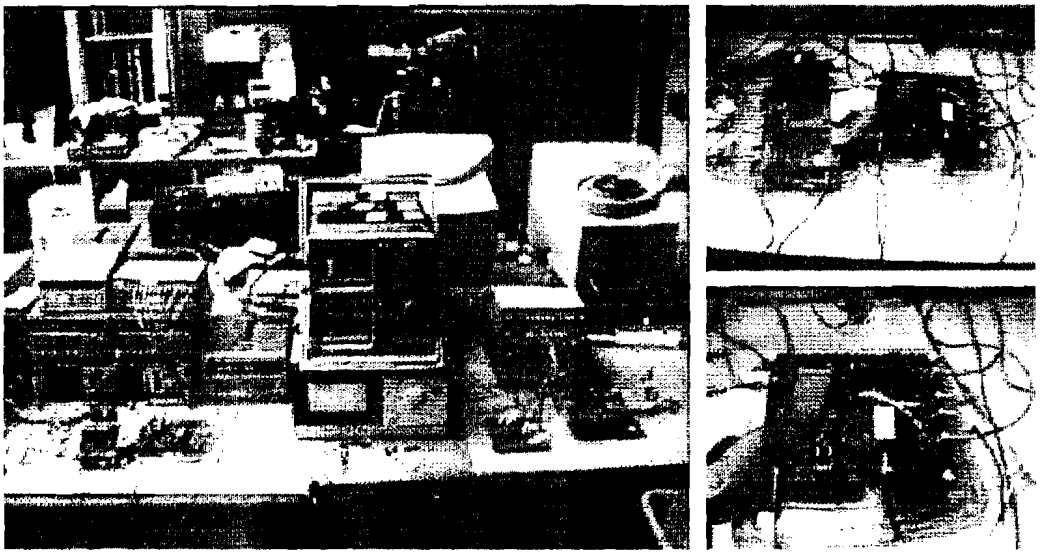
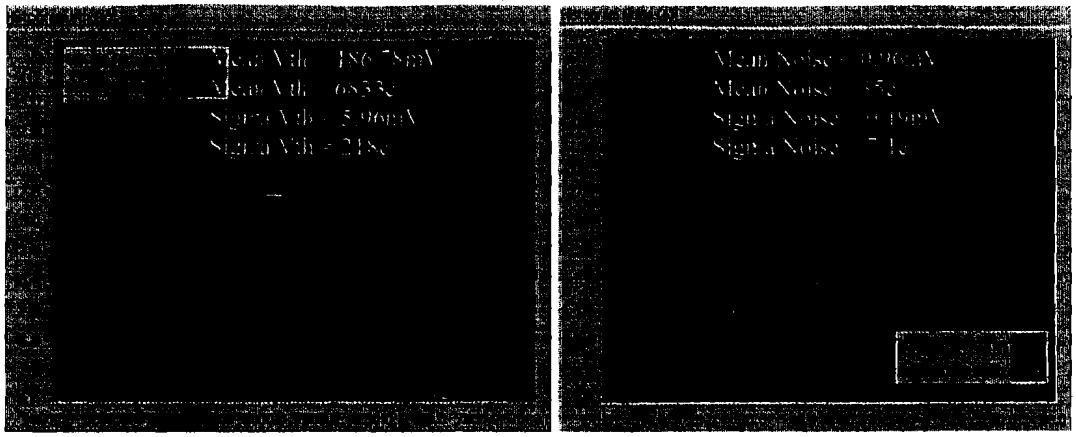


Figure 7 Control and Readout Optical Link prototype setup

Table 1 compares mean and sigma of noise and threshold of a five-chip FPIX module using two different setups. In the first experiment, the MCM was controlled by a probe station and the data was stored directly into a logic state analyzer. The FPIX initialization patterns were generated by a pattern generator, whose outputs were directly connected to the MCM inputs. In the second case, the initialization and control data is provided through the Control Optical Link at 52MHz and the data is readout through the 1.06 Gb/s optical link. Figures 8 and 9 show the comparative histograms of noise and threshold for both tests.

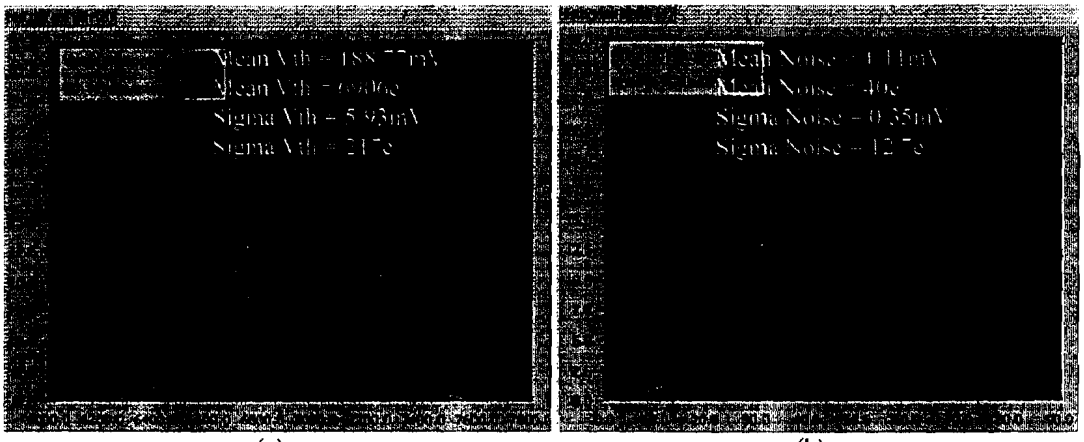
Vth0 = 1.95V	Results of the characterization of the MCM without Optical Link				
	Chip 1	**Chip 2**	**Chip 3**	**Chip 4**	**Chip 5**
Mean threshold (e-)	6833	6657	6581	6792	6956
Sigma threshold (e-)	218	239	217	168	146
Mean noise (e-)	35	37	37	31	31
Sigma noise (e-)	7.1	8.1	9.0	5.9	6.2
Vth0 = 1.95V	Results of the characterization of the MCM with Optical Link				
	Chip 1	**Chip 2**	**Chip 3**	**Chip 4**	**Chip 5**
Mean threshold (e-)	6906	6886	6820	6910	7051
Sigma threshold (e-)	217	184	179	142	161
Mean noise (e-)	40	37	38	32	30
Sigma noise (e-)	12.7	13.5	12.9	9.6	9.6

Table 1 MCM characterization comparison



(a) (b)

Figure 8: Five FPIX1 MCM without using the optical links. a) Threshold distribution and b) Noise Distribution.



(a) (b)

Figure 9: Five FPIX1 MCM using the optical links. a) Threshold distribution and b) Noise Distribution.

VI. CONCLUSIONS

A 5 chip FPIX1 Pixel Detector Multichip Module has been integrated to be controlled and readout through fiber optics. The first prototypes have proved to have an excellent performance. The readout clock has been recovered from the bi-phase signal with only 65 ps of jitter. This noise is acceptable for clocking a high-speed serializer at 1.06 Gb/s. The BER of the 106 MB/s link is better than 10^{-14} . EMI and radiation tests will define the location of BTeV's optoelectronics.

REFERENCES

[1] BTeV: An Expression of Interest for a Heavy Quark Program at C0, BTeV collaboration, Fermilab, May 18, 1997.

[2] ", G. Cancelo, D. Christian, S. Zimmermann, J. Hoff, A. Mekkaoui, R. Yarema, "Development of a pixel readout chip for BTeVNuclear Instruments & Methods in Physics Reasearch A 435 (1999) pp144-152.

[3] P. Denes, "Radiation hard gigabit systems", Proceedings of the 2nd Workshop on Optical Readout Technologies for ATLAS, Oxford, Jan. 1999.

[4] M.Pearce et al, Development of Radiation Tolerant Gb/s Optical Links for the Front-end Read-out of the ATLAS Liquid Argeon Calorimeter, Proceedings of Fourth Workshop on Electronics for LHC Experiments September 1998.

[5] G.Cancelo, G.Cardoso, S.Sihanguir, D.Christian, R.Downing, J.Hoff, A.Mekkaoui, A.Vargas, R.Yarema, S.Zimmermann. "Development of high data readout rate pixel module and detector hibriditization at Fermilab", Proceedings of PIXEL2000, Genoa, Italy, June 5-8 2000.

[6] M.A.Vargas, M.Sheaff, S.Vergara, The Development of Software to Characterize the Fermilab Pixel Readout chip FPIX1 for the BTeV experiment. Submitted to Niclear Instruments and Methodes, Aug. 2000.