HIGH READOUT SPEED PIXEL CHIP DEVELOPMENT AT FERMILAB

G. Cancelo, D. Christian, J. Hoff, A. Mekkaoui, R. Yarema, S. Zimmermann
Fermi National Accelerator Laboratory, Batavia, IL 60510

Abstract

Pixel detectors are becoming a very important part of high energy physics experiments, including those at the Tevatron and the LHC. At Fermilab, a pixel detector for the BTeV experiment is proposed for installation a few millimeters from the beam. Its information will be used in on-line track finding for the lowest level trigger system. This application requires pixel chips with high readout speed. The architecture of the pixel chips being designed at Fermilab will be presented, and future proposed developments and simulations will be summarized.

1. INTRODUCTION

At Fermilab, the BTeV experiment has been proposed for the C-Zero interaction region of the Tevatron [1, 2]. The innermost detector for this experiment will be a pixel detector composed of 93 pixel planes of 100×100 mm each, divided in 31 triple-stations perpendicular to the colliding beam and installed a few millimeters from the beam. This detector will be employed for on-line track finding for the lowest level trigger system [3] and, therefore, the pixel chips will have to read out all detected hits. Simulations have shown that, given a luminosity of 2 $\times 10^{32}$ $cm^{-2}s^{-1}$ (which corresponds to two interactions per crossing), a pixel chip of 8×7.2 mm active area placed 6 mm from the beam (the innermost chip), will be hit by one or more tracks in approximately one 132 ns bunch crossing (BCO) out of four [4]. At this luminosity, it is estimated that an average of approximately five 50×400 μm pixels will be hit in the innermost chip in those crossings with any data [5]. Therefore, this pixel chip has to sustain an average readout rate of 1.25 pixels per BCO. To account for statistical fluctuations and other effects (for example, the same simulations have shown that more than 20 pixels can be hit in just one BCO) the chip has to be capable of even higher data transfer performance.

Another very important factor that impacts the required data transfer rate is the need for analog to digital conversion (ADC) of the detected pulse height. Simulations have shown that a 50×400 μm pixel with a two or three bit ADC may be enough to achieve the necessary resolution [6]. Experiments that are now being arranged for a test beam should help to confirm the final chip requirements. Though there is no final agreement about the pixel size within the BTeV collaboration, there is a reasonable consensus that the experiment will require analog readout.

The pixel chip will be installed very close to the beam and therefore will have to be implemented in a radiation hard technology. We intend to use the new 0.5 μm process from Honeywell to accommodate this very severe constraint. Prototype chips will be made using the Hewlett Packard 0.5 μm CMOS process.

The pixel chip development described here is a succession of steps and submissions toward a chip that meets the BTeV requirements, each achieving specific engineering goals. The chips resulting from these steps have been dubbed FPIX0, FPIX1, and so on. In the next sections we will describe FPIX1, which represents the first step towards the final pixel readout architecture necessary for the BTeV experiment. Its primary purpose is to determine how fast the chip can process information internally and therefore, allow accurate extrapolation to the ultimate possible readout speed. Previous steps included FPIX0 [7, 8] and Pre-FPIX1, which were designed to test different front-end configurations and cross-talk management ideas. We anticipate that enhancements to the FPIX1 architecture will be necessary to achieve the readout speed required by BTeV. Simulations based on our experience with FPIX1 will guide the design of these enhancements.

2. FERMILAB PIXEL CHIP 1 (FPIX1)

The FPIX1 is a column based pixel chip with 50×400 μm pixel cells arranged in an array of 160 rows by 18 columns. Following an idea presented by Wright, et al. [9], FPIX1 stores hit information awaiting readout in the pixel unit cells, and uses an indirect addressing scheme to reference the hits to BCO numbers held in registers at the end of each column. However, rather than using pointers to accomplish the indirect addressing, FPIX1 uses a command driven design, which is described below. The chip can be divided into three mutually dependent pieces: the Pixel Cell, the End-Of-Column (EOC) Logic and the Chip Control Logic [10] (Figure 1). The responsibility of the Chip Control Logic is to control and maintain all features that are common to the chip such as the clocks, the “current” and “requested” BC0 number, and the status of off-chip communication. Each one of the eighteen EOC Logic cells controls one column. The EOC Logic responds to information from the Chip Logic, and from the 160 pixels in a column by broadcasting commands to the Pixel Cells, and by arbitrating with the other EOC

Work supported by the U.S. Department of Energy under contract No.DE-AC02-76CH03000. Fermilab conference paper No. 98/278.

Logic cells for control of the on-chip data bus. Finally, each Pixel Cell connects to one sensor pixel and responds to commands from the EOC Logic. The commands used in this architecture are the following: the “input” command instructs a Pixel Cell to accept hits from its sensor pixel and to respond to a hit by alerting the EOC. In the absence of an input command, the hit is ignored. The “output” command instructs the Pixel Cell to prepare to write its information onto the bus. The “reset” command instructs the Pixel Cell to reset its contents. Finally, the “idle” command instructs the Pixel Cell to do nothing.

Each EOC Logic cell consists of four EOC command Sets, each one capable of generating its own commands. When a Pixel Cell receives a hit, it immediately associates itself with whatever EOC Set is broadcasting the “input” command. From that point until it is reset or output, the Pixel Cell only responds to commands from its associated EOC Set. Meanwhile, the EOC Set holds the timestamp. The EOC Set can then issue the “output” or the “reset” command. The hit information is stored inside the Pixel Cell until readout.

The Chip Command Logic supports two readout modes. The first one, the “continuous” readout mode, requires no external trigger. This mode is expected to be used in the BTeV experiment. The other readout mode is the external trigger mode, in which an external system must provide the timestamp of the hits that should be read out. This mode is applicable for pixel detectors with external trigger, and for diagnostic purposes.

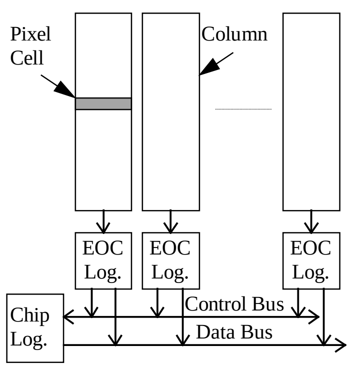

Figure 1. FPIX1 Block Diagram

2.1 Pixel Cells

The pixel cells hold the front-end electronics and the digital interface with the EOC Logic. The FPIX1 front-end is based on a design implemented in the FPIX0 and Pre-FPIX1 test chips. Reference [8] reports in detail on measurements of FPIX0, both unbonded and bonded to an ATLAS test sensor. The front-end (Figure 2) contains a charge sensitive amplifier (CSA) and a second amplification stage. The DC feedback used in the CSA is similar to the one described in [11]. The average discharge time of the CSA can be adjusted from 50 ns to 1 ms by an external current source without requiring any reset signals to be transmitted across the sensitive analog region. The output of the second stage connects to a flash ADC and a discriminator.

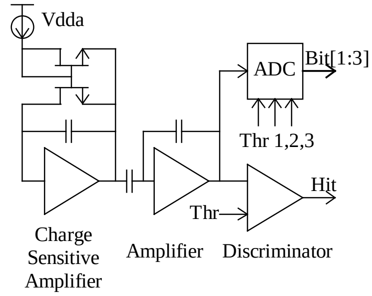

Figure 2. Front-end

The discriminator output Hit is asserted when the signal at the input of the discriminator is higher than the threshold (Thr). The flash ADC consists of three comparators directly connected to SR flip flops inside the pixel cell. The four thresholds (common for all pixel cells) are input to the chip as DC levels. During readout, tri-state buffers connect the outputs of the ADC flip flops to the EOC Logic, where they are encoded into two bits.

The digital interface of the pixel cell is depicted in Figure 3. It 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. 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 that is issuing the input command. Simultaneously, it alerts the EOC Logic to the presence of a hit via the wire-or’ed HFastOR signal. After the association to a particular EOC Set has been made, the Interpreter ignores commands from all other EOC Sets. The pixel hit information is stored in the cell until the associated EOC Set issues an output or reset command.

When the associated EOC Set issues the output command, the Interpreter issues a bus request and asserts the wire or’ed RfastOR signal. This operation is executed independent of the Master Clock (Mclk). The balance of the readout proceeds synchronous with the Mclk. The EOC Logic provides a column token on the bottom of the column as a means to regulate bus access. The token quickly passes pixel cells with no information until it reaches a cell that is requesting the bus. This propagation to a hit pixel is done in less than one clock cycle, even if the pixel is the last in the chain. At the next rising edge of the MClk, the hit pixel with the column token loads its data onto the bus and drives it to the EOC logic for one clock cycle. In parallel, the column token is transmitted to the next hit pixel, pipelining the output of the Pixel Cell with the token passing. This allows the readout of one Pixel Cell per clock cycle, without any wasted MClk cycles. 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. The RFastOR returns to its inactive state while the last of the hit pixels is being read out. This way, the EOC Logic is able to detect when the last hit pixel in the column is being output. At the next rising edge of the MClk, control of the on-chip bus is transferred to the next column with hit data.

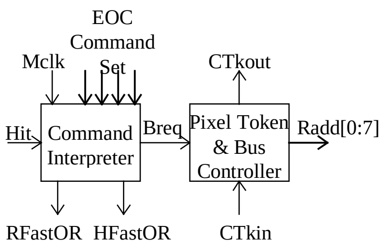

Figure 3. Pixel Cell Digital Interface

At any given time, only one EOC Set is permitted to broadcast the input command. This insures that hit pixel cells are associated with only one EOC Set. Pixel cells that have not been hit continue to monitor all four EOC Sets, waiting for a coincidence of hit and input commands.

2.2 End of Column Logic

Figure 4 shows a block diagram of the EOC Logic. It consists of a Priority Encoder and four EOC command Sets. The EOC Sets themselves consist of a timestamp register, a state machine for generating the appropriate EOC commands, and two comparators.

The Priority Encoder selects one EOC Set to issue the input command. When there is a hit somewhere in the column, the HFastOR signal is asserted, and the state machine inside the assigned EOC Set responds by latching the Current BCO (CBCO) in its EOC timestamp register and by issuing the idle command at the next rising edge of the BCO clock. This ensures that 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 issue the input command to the column (at the rising edge of the BCO clock). Since the EOC Logic has four EOC Sets, pixel cells in the column can be hit without loss of data in four different crossings before any data is read out.

A "hit" EOC Set waits for matches with its stored timestamp BCO (SBCO). If the match is between the Requested BCO (RBCO) and the SBCO, the EOC Set broadcasts the output command, and if the match is between the Current BCO (CBCO) and the SBCO, it broadcasts the reset command. Any of the bits in the comparison between CBCO and SBCO can be programmed to be ignored. This allows for a user defined reset delay. This feature is designed primarily for the externally triggered readout mode.

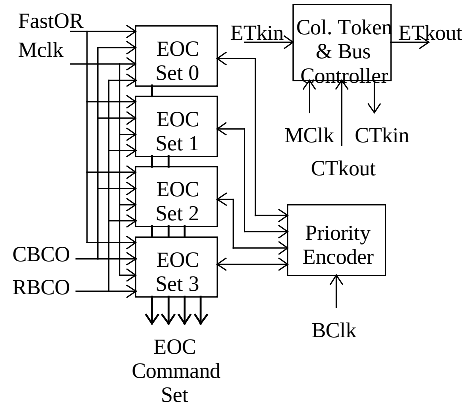

Figure 4. End Of Column Logic

A second state machine is implemented inside the Column Token and Bus Controller, to control the access to the EOC data bus. This access is arbitrated by an EOC token. As soon as 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 stays in this state until all hits in the column are read out. The Column Controller now passes the EOC token to the next EOC Logic by asserting ETkout. The early delivery of the CTkin to a column, even before the ETkin is received, allows for the pixel data to be asserted on the internal data bus as soon as the EOC token arrives at the EOC Logic. This, added with the assertion of ETkout as soon as CTkout is received by the Column controller, allows for full clock speed readout of the pixel data, even when the chip finishes the read out of one column and starts the read out of the next.

2.3 Chip Logic

The Chip Logic controls the features associated with the whole chip. It consists of the Current BCO counter (CBCO), the Readout BCO counter, a multiplexer, and a chip controller. The CBCO increments synchronously with the BClk and is delivered to the EOC logic. The multiplexer multiplexes the Readout BCO counter or the External Request BCO (in case external trigger is used). The output of the multiplexer forms the RBCO number that is delivered to the EOC logic. When the chip is operating in the "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 EOC Timestamp registers. The clock of the Readout BCO counter is not a free running clock. The Readout BCO counter lags two counts behind the CBCO, in order to avoid comparisons in the EOC logic of an event that did not yet stabilize inside the pixel cells. The Readout BCO counts in parallel to the CBCO counter (using the BClk) until the EOC Sets detect a match between the RBCO and the timestamp latched in the EOC Timestamp register. Then the clock stops and the chip controller starts the read out of the chip by issuing a ETkin to the first EOC Logic. When the read out of that specific timestamp finishes (i.e., the Chip Logic detects that ETkout of the last column was asserted), the controller quickly increments the Readout BCO Counter (using the MClk) until another match is detected or the Readout BCO counter reaches two counts behind the CBCO.

Other functions of the Chip Logic include external bus arbitration, which is done again by a token passing from chip to chip, the control of the configuration of the chip, and data "throttling". The configuration is programmed using a serial bit stream to set features like pixel cell kill (to disable noisy pixels) and pulse inject select (for enabling programmable pixel cells to accept charge inject directly into the front-end using an external voltage source). Data throttling is the following: there is an external input to the pixel chip that allows one to command the pixel cells to disregard a bunch crossing, even if it contains hit information. The DAQ can monitor the timestamp of the data as it is read out. If the timestamp gets too delayed with respect to the current BCO, creating the possibility of recorded hit losses, the DAQ can command the pixel chip to disregard future bunch crossings (creating dead time, but no data bias), until the delay drops to an acceptable value.

3. FERMILAB PIXEL CHIP 2 (FPIX2)

Before we can choose the final FPIX2 architecture, we need to decide how many bits of analog information are actually necessary for the BTeV experiment. The present plan is to use FPIX0, bump bonded to ATLAS detectors, in a test beam in order to collect information that will provide an experimental basis for this decision. As described above, FPIX1 implements a two bit flash ADC inside the pixel cell. We have confidence that a three bit ADC could fit inside a slightly larger cell, for example 50 $\times$450 μm. However, if the test beam shows that four or more bits of analog to digital conversion is required, the ADCs will have to be moved to the periphery of the chip. Analog pulse height information will have to be transmitted for digitization from the pixel cells to the column periphery.

As we have designed FPIX1, we have also considered and simulated features, which may be included in the future to increase the readout speed. We will now present some of them. Clearly, one possibility is to increase the readout clock frequency (MClk for FPIX1), but this is constrained by the internal speed of the chip and by power dissipation. Another option is to increase the width of the output data word, but this is also constrained by the complexity and mass of the multichip module interconnect and the chip power dissipation. A third option is to achieve some form of data compression inside the chip. The data alignment by timestamp implemented in FPIX1 already achieves an initial degree of data compression. During readout, the timestamp has to be transmitted just once for all the hits that occurred simultaneously. In the next FPIX generation, we hope to implement what we have named "group" reading. The concept is that instead of reading a column by individually reading each pixel cell, we will read groups of consecutive pixel cells in parallel. So, in this sense, we will not have pixel row address, but actually pixel group address, and the row position of several pixels can be uniquely identified with just one row address. Simulations described in next section have shown a data compression by a factor of 2.36 with respect to a pixel by pixel readout. Another option for data compression is the transmission of the column address just once for all the hit data in a given column.

Another method to increase the readout speed is to use two readout clocks. One, the bus readout clock, operates at higher frequency, and is associated just with the output data bus logic of the chip. The other, the internal readout clock, operates at some lower frequency compatible with the microelectronics process and some reasonable power dissipation (for example, the bus readout clock operates at twice the frequency of the internal readout clock). The problem which then must be solved is how to deliver enough hit information to the output data logic to optimize the utilization of the output data bus. If the chip has hits and has control over the data bus, it should transmit data continuously over the bus without wasting readout clock cycles. We have considered two approaches to the solution of this problem. One is to use multiple buses inside the pixel chip, which can transfer hit information from more than one column in parallel to the output data bus logic. The output logic then multiplexes at higher rate the hit data of different internal buses to the output data bus. This approach does not look attractive for the BTeV experiment, since simulations have shown that, in 60% of the BCOs with tracks, only one column is hit. Therefore, simultaneous readout of multiple columns will in general not avoid waste of readout clock cycles [5]. The second option is to use a very wide internal data bus. The chip would transmit all information associated with a group (group column and row address and four ADC conversions) in one internal readout clock cycle and in parallel to the data output bus logic. The output logic would then divide this word into narrower words, and transmit them at a higher frequency. This second approach looks very attractive, since it addresses the concern previously raised by the simulations. Furthermore, it takes advantages of features already implemented in FPIX1, specifically the set of pipeline features that allows the chip to read at full clock speed even when the chip finishes the read out of one column and starts the read out of the next.

Finally, we are also considering pipelining incrementing the Readout BCO Counter with the readout of data associated with a previous timestamp, or even some faster method to locate the next stored timestamp.

Future submissions will also contain outputs to signal internal error states, and digital to analog converters to allow the thresholds to be downloaded to the chip along with other configuration data.

4. SIMULATIONS

We have done extensive simulations of different enhanced FPIX chip architectures. We will describe here a simulation that assumes two or three bit flash ADCs inside the pixel cells. For a more complete description of these results, see Reference [12]. The objective of the simulations was to determine if the proposed FPIX2 chip architecture could achieve the necessary data rate to avoid dead time in the BteV experiment. The conditions for the simulations were as follows: The delays used for the internal logic of the chip are set by Verilog and Spice simulations of FPIX1. We simulated the performance of the pixel chip that will receive the maximum fluence in the detector. We assumed a pixel size of $50\times400\mu$m and a chip with 160 rows by 18 columns, four pixel "grouping", a wide internal bus operating at 26.5 MHz and an output bus operating at 53 MHz with a half the width of the internal bus. The hit input was provided by a Monte Carlo simulation of minimum bias events of approximately 5000 BCOs, assuming n+ on n sensors and discriminator thresholds set at 2000 e_-. The total number of interactions per BCO is a random number chosen from a Poisson distribution with mean 2. This is equivalent to a luminosity of 2×10^{32} $cm^{-2}s^{-1}$, the highest luminosity expected for the BTeV experiment. The number of pixels hit by each track is affected by the fact that the sensors are located in a 1.6 T magnetic field [5]. A total of 1240 BCOs with hits, and a total of 6152 hits, were simulated. This represents an average of approximately five pixel hits per bunch crossing with tracks. The results of the simulations show that reading groups of four consecutive pixels results in a data compression of 2.36 with respect to a pixel by pixel readout. The data rate did not approach the maximum possible rate; the output bus was only approximately 40% of the time utilized. This gives us a reasonable margin to account for other effects not included in the Monte Carlo simulations.

5. CONCLUSION

In this paper we have described the FPIX1 chip. We have also described proposed enhancements that we plan to incorporate in the future in order to increase data compression and readout speed. Finally, simulations have demonstrated that the FPIX architecture should be capable of achieving the hit readout rate required by the BTeV experiment.

6. REFERENCES

[1] Kaplan, D. M., "BTeV/CO," Proc. Int. Symp. on Near Beam Physics, pp. 66-71, Fermilab, 1998.

[2] Santoro, A., *et al.*, "An Expression of Interest for a Heavy Quark Program at C0," BTeV pub. note, Fermilab, May 1997.

[3] Husby, D., *et al.*, "Design of a secondary-vertex trigger system for a hadron collider," Nucl. Instrum. Meth. A383, pp. 193-198, 1996.

[4] Kasper, P., "Study of the pixel occupancy in the BTeV detector," BTeV int. note, Apr. 98.

[5] Kasper, P., Pixel hit data generated from Monte Carlo simulations for minimum bias and b-quarks events, Fermilab, Mar. 1998.

[6] Artuso, M. and Wang, J., "Some further studies on factors affecting the pixel resolution," BTeV int. note, July 1998.

[7] Mekkaoui, M., *et al.*, "Results from the FPIX0 chip bump-bonded to the ATLAS pixel detector," this workshop.

[8] Mekkaoui, A., "FPIX0: A prototype pixel FE chip at Fermilab," Proc. 3rd Int. Meeting Front End Elect. for High Resol. Tracking Det., Taos, NM, Nov. 1997.

[9] Wright, M., Millaud, J., and Nygren, D., "A pixel unit-cell targeting 16ns resolution and radiation hardness in a column read-out particle vertex detector," LBL-32912, Berkeley, 1992.

[10] Hoff, J., "FPIX1 architecture and operation: design and simulations," PPD/ETT/ES int. note, Fermilab, June 1998.

[11] Blamquart, *et al.*, "Pixel analog cells prototypes for ATLAS in DMILL technology," Nucl. Instrum. Meth. A395, pp. 313-317, 1997.

[12] Cancelo, G., Zimmermann, S., "Modeling and simulation of a readout architecture for pixel-based detectors," to be presented in the 1998 Nuclear Science Symp., Toronto, Canada, Nov. 1998.