

Fermi National Accelerator Laboratory

PIXEL DETECTOR PROJECT

FPIX1 CHIP

Document # ESE-PIX-05241999

-PRELIMINARY-

May 24, 1999

Jim Hoff, Abder Mekkaoui and Sergio Zimmermann

TABLE OF CONTENTS

| | | |
|-----------|--|-----------|
| 1 | INTRODUCTION..... | 1 |
| 2 | DETAILED PAD DESCRIPTION | 4 |
| 3 | PROGRAMMING AND THE SCAN PATHS..... | 8 |
| 4 | DATA READOUT | 11 |
| 5 | RESETTING..... | 12 |
| 6 | TRIGACC..... | 12 |
| 7 | BIAS CURRENTS AND REFERENCE VOLTAGES..... | 13 |
| 7.1 | THRESHOLD VOLTAGE FOR THE HIT COMPARATOR | 13 |
| 8 | DIFFERENTIAL OUTPUTS..... | 14 |
| 9 | PAD LOCATION AND CHIP DIMENSIONS | 15 |
| 10 | APPENDIX I..... | 17 |

1 Introduction

This document describes the FPIX1 chip. For a general overview of the FPIX1 chip and its functionality the reader should refer to the paper “High Speed Pixel Chip Development at Fermilab” included in the Appendix I.

Table 1 describes the FPIX1 pads. FPIX1 has two pad columns, where the column with odd pad numbers are closer to the chip edge. This detail is depicted in Table 1 by the two separate columns with pad numbers. In the next sections we will describe the usage of these pads.

Table 1. FPIX1 Pads Description

| Odd Pads | Even Pads | Pad Name | Description | Type |
|----------|-----------|-----------|---|------|
| 1 | | Vssa | Substrate voltage and ground for pixel analog circuitry | GND |
| | 2 | Inject In | Analog Input for test charge | |
| 3 | | Vdda | Power for pixel analog circuitry | 3.3V |
| | 4 | Ivbbnl | Decoupling point and override for internal bias | |
| 5 | | Vss_comp | Ground for pixel A to D conversion | GND |
| | 6 | Ivfb | Feedback current (decoupled) | |
| 7 | | Vdd_comp | Power for pixel A to D conversion | 3.3V |
| | 8 | Ivbbp | Master Bias Current (decoupled) | |
| 9 | | Vssd | Ground for pixel command interpreter and drivers | GND |
| | 10 | Ivbbn | Decoupling point and override for internal bias | |
| 11 | | Vddd | Power for pixel command interpreter and drivers | 3.3V |
| | 12 | Ivbbpl | Decoupling point and override for internal bias | |
| 13 | | Vsub | Substrate voltage | GND |
| | 14 | Ivbb_comp | Decoupling point and override for internal bias | |
| 15 | | Vssd_eoc | Ground for End-of-column Logic | GND |
| | 16 | Vth3 | Most significant threshold for ADC | |
| 17 | | Vddd_eoc | Power for End-of-column Logic | 3.3V |
| | 18 | Vth2 | Threshold for ADC | |
| 19 | | Vsub | Substrate voltage | GND |
| | 20 | Vth1 | Threshold for ADC | |
| 21 | | Vssd_eoc | Ground for End-of-column Logic | GND |
| | 22 | Vth0 | Threshold for Hit Comparator | |

| | | | | |
|----|----|------------|--|-----------------------|
| 23 | | Vddd_eoc | Power for End-of-column Logic | 3.3V |
| | 24 | Vpstop | Bias voltage for the detector p-stops | |
| 25 | | Vsub | Substrate voltage | GND |
| | 26 | Vnguard | Bias for the detector n-guard | |
| 27 | | TokenIn | Input for the readout token ¹ | In, LVDS |
| | 28 | TokenIn* | | |
| 29 | | TrigAcc | Under triggered mode, the trigger signal; under non-triggered mode, the throttle (logical level high = Accept) | In, CMOS |
| | 30 | ShiftIn | Input for the scan paths | In, CMOS |
| 31 | | DataReset | Reset signal which clears all clock, counter and data registers | In, CMOS |
| | 32 | ProgramRes | Reset signal which clears all scan paths except Kill and Inject | In, CMOS |
| 33 | | ShClk | Shift Clock signal | In, CMOS |
| | 34 | ReadClk* | Readout Clock | In, LVDS ² |
| 35 | | ReadClk | | |
| | 36 | BCOClk* | Beam Cross Over clock | In, LVDS |
| 37 | | BCOClk | | |
| | 38 | Data0* | If Data[15]=1 \Rightarrow Row 0. If Data[15]=0 \Rightarrow Readout BCO 0 (RBCO 0) | Out, LVDS |
| 39 | | Data0 | | |
| | 40 | Data1* | If Data[15]=1 \Rightarrow Row 1. If Data[15]=0 \Rightarrow RBCO 1 | Out, LVDS |
| 41 | | Data1 | | |
| | 42 | Data2* | If Data[15]=1 \Rightarrow Row 2. If Data[15]=0 \Rightarrow RBCO 2 | Out, LVDS |
| 43 | | Data2 | | |
| | 44 | Data3* | If Data[15]=1 \Rightarrow Row 3. If Data[15]=0 \Rightarrow RBCO 3 | Out, LVDS |
| 45 | | Data3 | | |
| | 46 | Data4* | If Data[15]=1 \Rightarrow Row 4. If Data[15]=0 \Rightarrow RBCO 4 | Out, LVDS |
| 47 | | Data4 | | |
| | 48 | Data5* | If Data[15]=1 \Rightarrow Row 5. If Data[15]=0 \Rightarrow RBCO 5 (MSB ³ | Out, LVDS |

¹ Just one of the signals of the differential signals is described. The absence of the asterisks on the signal label mean the true signal, while the label with asterisks mean the complement of the true signal.

² The FPIX1 chip output drivers do not actually conform with the LVDS specifications, though with the correct termination it produces similar results. To understand the details of the output drivers refer to Section 7.

| | | | | |
|----|----|-------------|--|-----------|
| | | | RBCO) | |
| 49 | | Data5 | | |
| | 50 | LoadKill | On the falling edge of this signal the Kill pattern is loaded into the pixels | In, CMOS |
| 51 | | Vss_pad | Ground for the pad circuitry | GND |
| | 52 | ChipHasData | Logical level high is if chip has data | Out, CMOS |
| 53 | | Vdd_pad | Power for pad circuitry | 3.3V |
| | 54 | Data6* | If Data[15]=1 \Rightarrow Row 6 If Data[15]=0 \Rightarrow logic level 0 | Out, LVDS |
| 55 | | Data6 | | |
| | 56 | Data7* | If Data[15]=1 \Rightarrow Row 7 (MSB Row Address). If Data[15]=0 \Rightarrow logic level 0 | Out, LVDS |
| 58 | | Data7 | | |
| | 58 | Data8* | If Data[15]=1 \Rightarrow Col 0. If Data[15]=0 \Rightarrow ChipID 0 | Out, LVDS |
| 59 | | Data8 | | |
| | 60 | Data9* | If Data[15]=1 \Rightarrow Col 1. If Data[15]=0 \Rightarrow ChipID 1 | Out, LVDS |
| 61 | | Data9 | | |
| | 62 | Data10* | If Data[15]=1 \Rightarrow Col 2. If Data[15]=0 \Rightarrow ChipID 2 | Out, LVDS |
| 63 | | Data10 | | |
| | 64 | Data11* | If Data[15]=1 \Rightarrow Col 3. If Data[15]=0 \Rightarrow ChipID 3 (MSB ChipID) | Out, LVDS |
| 65 | | Data11 | | |
| | 66 | Data12* | If Data[15]=1 \Rightarrow Col 4 (MSB Col Addr). If Data[15]=0 \Rightarrow logic level 0 | Out, LVDS |
| 67 | | Data12 | | |
| | 68 | Data13* | If Data[15]=1 \Rightarrow ADC0. If Data[15]=0 \Rightarrow logical level 0 | Out, LVDS |
| 69 | | Data13 | | |
| | 70 | Data14* | If Data[15]=1 \Rightarrow ADC1 (MSB ADC). If Data[15]=0 \Rightarrow logic level 0 | Out, LVDS |
| 71 | | Data14 | | |
| | 72 | Data15* | Control bit, logic level 1 means data, while logic level 0 means header. | Out, LVDS |
| 73 | | Data15 | | |
| | 74 | Data16 | DataValid | Out, LVDS |

³ MSB: Most Significant Bit

| | | | | |
|----|----|-----------|---|-----------|
| 75 | | Data16* | | |
| | 76 | Reg2 | Scan path selection (MSB) | In, CMOS |
| 77 | | Vddd | Power for pixel command interpreter and drivers | 3.3V |
| | 78 | Reg1 | Scan path selection bit | In, CMOS |
| 79 | | Vssd | Ground for pixel command interpreter and drivers | GND |
| | 80 | Reg0 | Scan path selection bit | In, CMOS |
| 81 | | Vss_comp | Ground for pixel A to D conversion | GND |
| | 82 | Shift Out | Output from scan path | Out, CMOS |
| 83 | | Vdd_comp | Power for pixel A to D conversion | |
| | 84 | TokenOut* | Chip Token Output | Out, LVDS |
| 85 | | Vdda | Power for pixel analog circuitry | 3.3V |
| | 86 | Token Out | | |
| 87 | | Vssa | Substrate voltage and ground for pixel analog circuitry | GND |

2 Detailed Pad Description

Vssa: Analog ground. Tie directly to a low impedance, low-noise ground plane

Inject-In: This is the analog test input. Pixels are set for Injection and/or Killed by the Kill/Inject scan path. Those, which are set for Injection and not Killed, are given a charge through Inject-In. This pin is generally tied to analog ground through a 50-ohm resistor. A voltage pulse on this pin causes charge injection into the appropriate pixels. The magnitude of this injection is a function of the size of the voltage pulse. As the design is optimized for electron collection, the charge is injected on the negative edge of the pulse. If you are NOT injecting, keep this pad at an AC ground. If you are injecting, start the signal at X volts, drop it to some voltage less than X volts, and hold it at this new voltage. X is greater than 1 volt and less than 2 volts. About $40 e^-$ are injected in the input of the pre-amplifier for one millivolt of voltage change in this pad.

Vdda: Analog VDD = 3.3V. A $\pm 10\%$ variation on this voltage has been allowed for by simulation. However, users should nevertheless consider 3.3 volts a religious obligation. Board designers should design this trace to handle a sustained DC current of approximately 100 mA.

Ivbbnl: 1 nF decoupling capacitor to analog ground. This pad may be optionally used for optimization

Vss_comp: Ground for analog to digital converters in each pixel. It should be tied to the same low impedance, low-noise ground plane as Vssa (pad 1)

Ivfb: 1 nF decoupling capacitor to analog ground. 5 nA (nominal) current should be sunk from the chip.

Vdd_comp: VDD for analog to digital converters. 3.3 Volts

Ivbbp: 1 nF decoupling capacitor to analog ground. 7 μ A (nominal) current should be sunk from the chip.

Vssd: Digital Ground for digital circuitry in the pixel cells. This should be connected to a low impedance, “dirty” ground plane. It should NOT be connected to the analog ground plane.

Ivbbn: 1 nF decoupling capacitor to analog ground. This pad may be optionally used for optimization.

Vddd: Digital Power for digital circuitry in the pixel cells. 3.3 volts. It is also “dirty” and should NOT be connected to analog power. It should also be able to handle a sustained DC current of approximately 100 mA.

Ivbbpl: 1 nF decoupling capacitor to analog ground. This pad may be optionally used for optimization.

Vsub: Substrate voltage (0 V). It should be connected to the low impedance, low noise analog ground plane. DC current consumption should be 0 amps.

Ivbb_comp: 1 nF decoupling capacitor to analog ground. This pad may be optionally used for optimization.

Vssd_eoc: Digital Ground for the End-of-column circuitry. 0V. This should be connected to a low impedance, “dirty” ground plane. It should NOT be connected to the analog ground plane.

Vth3: Threshold voltage for the most significant analog to digital comparator. 2V – 0V.

Vddd_eoc: Digital Power for End-of-column circuitry. 3.3V. It is also “dirty” and should NOT be connected to analog power.

Vth2: Threshold voltage for the “middle” analog to digital comparator. 2V – 0V.

Vsub: Substrate voltage (0 V). It should be connected to the low impedance, low noise analog ground plane.

Vth1: Threshold voltage for the least significant analog to digital comparator. 2V – 0V.

Vssd_eoc: Digital Ground for the End-of-column circuitry. 0V. This should be connected to a low impedance, “dirty” ground plane. It should NOT be connected to the analog ground plane.

Vth0: Threshold voltage for the Hit comparator. 2V – 0V.

Vddd_eoc: Digital Power for End-of-column circuitry. 3.3V. It is also “dirty” and should NOT be connected to analog power.

Vpstop: Detector bias. This voltage is dependent on which detector is used

Vsub: Substrate voltage (0 V). It should be connected to the low impedance, low noise analog ground plane.

Vanguard: Detector bias. This voltage is dependent on which detector is used

Token In/Token In*: Low voltage differential signal input ($V_{hi} \cong 1.75V$; $V_{lo} \cong 1.55V$). When this 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. This is the only LVDS input that has its own internal 100-ohm load resistor. This signal can be held high for the hottest chip.

TrigAcc: TrigAcc is a dual-mode signal. When the chip is operating in Triggered mode, it is the trigger signal, and on the rising edge of this signal, the requested BCO number is replaced by the Trigger most recently scanned into the chip. When the chip is operating in Non-triggered or continuous mode, it is the Accept/Reject signal. When Accept/Reject is a logical high (3.3 volts), the chip accepts incoming data. When Accept/Reject is a logical low (0 volts), the chip rejects (ignores) incoming data. In either mode, this is a single ended CMOS signal.

ShiftIn: CMOS input for the scan paths. At the rising edge of the ShiftClk, the value at the shift in will be scanned into the path chosen by the Reg0, Reg1, and Reg2 signals.

DataReset; CMOS level reset signal. When high (3.3 volts), the BCO counters are reset to zero; the End-of-column registers are reset to Empty; the chip stops Talking.

ProgramReset: CMOS level reset signal. When high (3.3 volts), the mask registers are reset to zero (no masking); the chip is reset to continuous mode (Non-triggered); the BCO lag is set to 2; the ChipID is set to zero.

ShClk: The Shift Clock is a CMOS level clock. At the rising edge of this clock, the contents of the ShiftIn are scanned into whatever scan path is selected by Reg<0:2>.

ReadClk*/ReadClk: A free-running LVDS level differential clock ($V_{hi}=1.75V$; $V_{low}=1.55V$). The simulated frequency was 26 MHz, but the real frequency will be whatever the chip can handle and/or whatever the board can supply.

BCOClk*/BCOClk: This is an LVDS representation of the Beam Crossover signal.

DATA: These are the data outputs. They are tri-statable LVDS outputs ($V_{hi}=1.75V$; $V_{low}=1.55V$) with the exception of Data 16. Data 16 is the Data Valid signal. It is a tri-statable LVDS IO, which senses the present state of the Data Valid and reacts to that state depending on whether or not the chip has data to send. In Table 2, Data 16 is shown as being 0 if no data is available. This is because if the chip is alone, it will never tri-state its Data Valid signal. If the chip is part of a Daisy Chain, then there will always be one chip driving the Data Valid signal. Therefore, a user should never really see Data Valid in a high impedance state. When there is data available, Data Valid will be a 1. When there is data available, it will come in two forms: Control Words which tell the data acquisition system the chip ID of the chip sending the data and BCO number (time stamp) of the data; and Data Words which tell the Row Address, Column Address and Magnitude (ADC value) of a particular hit pixel. Control Words are distinguished from Data Words by bit Data 15, which is a Logical 0 for a Control Word and a Logical 1 for a Data Word.

Table 2. Control and Data Words

| | No Data Available | Data Available Control Word | Data Available Data Word |
|------------------|-------------------|-----------------------------|--------------------------|
| Data 16/Data 16* | 0 | Logical 1 | Logical 1 |
| Data 15/Data 15* | X | Logical 0 | Logical 1 |
| Data 14/Data 14* | X | Logical 0 | ADC [1] |
| Data 13/Data 13* | X | Logical 0 | ADC [0] |
| Data 12/Data 12* | X | Logical 0 | Col Addr[4] |
| Data 11/Data 11* | X | ChipID [0] | Col Addr[3] |
| Data 10/Data 10* | X | ChipID [0] | Col Addr[2] |
| Data 9/Data 9* | X | ChipID [0] | Col Addr[1] |
| Data 8/Data 8* | X | ChipID [0] | Col Addr[0] |
| Data 7/Data 7* | X | Logical 0 | Row Addr[7] |
| Data 6/Data 6* | X | Logical 0 | Row Addr[6] |
| Data 5/Data 5* | X | BCO[5] | Row Addr[5] |
| Data 4/Data 4* | X | BCO[4] | Row Addr[4] |
| Data 3/Data 3* | X | BCO[3] | Row Addr[3] |
| Data 2/Data 2* | X | BCO[2] | Row Addr[2] |
| Data 1/Data 1* | X | BCO[1] | Row Addr[1] |
| Data 0/Data 0* | X | BCO[0] | Row Addr[0] |

Load Kill: The falling edge of this signal latches the kill pattern scanned into the pixel array. It is a CMOS level signal which is kept at a Logical 1 (3.3 volts) during the Kill scan and dropped to a Logical 0 to latch the kill pattern. During normal operation, it is kept at a Logical 0.

Vss_pad: Ground for the pad circuitry. 0V. This should be connected to a low impedance, “dirty” ground plane. It should NOT be connected to the analog ground plane.

ChipHasData: This is a CMOS level diagnostic output which goes to a Logical 1 when the chip has data to output in response to the current requested BCO number.

Vdd_pad: Power for the pad circuitry. 3.3V. It is also “dirty” and should NOT be connected to analog power. The board designer should expect that this signal will carry 50 mA and it will turn on when the chip has data to output and turn off when the chip tri-states itself.

Reg<2:0>: The scan path selectors for the ShiftIn. These are CMOS inputs and they should be set to 000 during normal, continuous mode operation. They will likely be set to 001 during normal, triggered mode operation.

Vddd: Digital Power for digital circuitry in the pixel cells. 3.3 volts. It is also “dirty” and should NOT be connected to analog power. It should also be able to handle a sustained DC current of approximately 100 mA.

Vssd: Digital Ground for digital circuitry in the pixel cells. This should be connected to a low impedance, “dirty” ground plane. It should NOT be connected to the analog ground plane.

Vss_comp: Ground for analog to digital converters in each pixel. It should be tied to the same low impedance, low-noise ground plane as Vssa (pad 1)

ShiftOut: The scan output for each of the scan paths as selected by Reg<2:0>. This is a CMOS level signal, which will be driven to the ShiftIn of the next chip.

Vdd_comp: VDD for analog to digital converters. 3.3 Volts

TokenOut*/TokenOut: The LVDS output of the Chip Token signal (Vhi=1.75V; Vlow=1.55V). Unlike the Data outputs, this is not tri-statable.

Vdda: Analog VDD = 3.3V. A $\pm 10\%$ variation on this voltage has been allowed for by simulation. However, users should nevertheless consider 3.3 volts a religious obligation. Board designers should design this trace to handle a sustained DC current of approximately 100 mA.

Vssa: Analog ground. Tie directly to a low impedance, low-noise ground plane

All LVDS outputs with the exception of the TokenOut require a resistive load. TokenOut does not require this because the load is included in the TokenIn pad circuitry. The LVDS receivers for the ReadClk and the BCOClk will, likewise require a 100-ohm load impedance. The CMOS outs and ins require no special circuitry.

3 Programming and the Scan Paths

There are eight internal scan paths formed by shift registers with different depths in FPIX1. Each scan path uses the same Shift In, the same Shift Out and the same ShClk. The particular scan path chosen is determined by the bit code set onto pads 76, 78 and 80 also known as Reg[2:0]. Table 3 show the path codes.

Table 3. Scan Path codes

| Reg[2:0] | | Path |
|----------|-----|----------------------|
| 0 | 000 | Idle – no path |
| 1 | 001 | Trigger path |
| 2 | 010 | Not used |
| 3 | 011 | Mask registers |
| 4 | 100 | Not used |
| 5 | 101 | Mode, ChipID and Lag |
| 6 | 110 | Kill or Inject |
| 7 | 111 | Kill or Inject |

Figure 1 shows the relation between ShiftIn and ShClk. The bit in the ShiftIn is sampled on the low to high transition of ShClk.

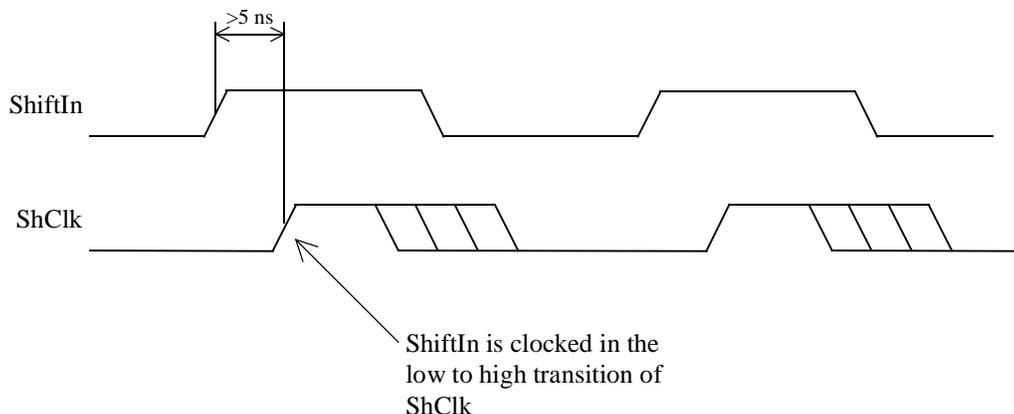


Figure 1. Relation between ShiftIn and ShClk

We will now describe the different scan paths. The Idle path should be the default path. If nothing is happening, Reg[2:0] should be set to 000. Paths 010 or 100 are, for the moment, the same as Path 000. However, in the future (FPIX2, etc) it is likely that they will be used for diagnostic scan paths or D-to-A converter scan paths.

The Trigger path (001) is used when the chip is in External Trigger Mode. It is a six bit deep path and the most significant bit should be scanned in first. When all six bits have been scanned in, a pulse on the TrigAcc pin (pad 29) will cause the new requested trigger to be presented to the End-of-column Logic cells. Using this scan path in Continuous Mode will have no effect.

The Mask register path (011) is used to set the programmable reset delay. Each end of column logic has a six bit deep Mask register. A Logic 1 located in any bit of this register causes the comparison between the specific bit of the Current BCO number and the specific bit of the Stored BCO number to be ignored. Since the End-of-column Set sends a reset command when the Current BCO number equals the Stored BCO number, the presence of the mask register allows the user to decide how long data is stored before it is reset. With all the bits in the mask register equal to zero, the End-of-column Set will reset itself in 64 BCO clock cycles. With the most significant bit ignored, the End-of-column Set will reset itself in 32 clock cycles. With all bits in the mask register set to one, the column is effectively shut down. With six bits per End-of-column cell, this scan path has $6 \times 17 = 102$ bits. The most significant bit of column 17 is scanned in first. The least significant bit of column 1 is scanned in last.

The mode path (101) contains the chip Mode (1=Triggered Mode, 0=Continuous Mode), the Chip ID, and the BCO lag. 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 so that if, for example, the Current BCO number was 5, the maximum value of the Requested BCO number would be 3. With 6 bits for the lag, 4 bits for the ChipID and 1 bit for the mode, this scan path is 11 bits long. The most significant bit of the lag is scanned in first followed by the remainder of the lag. Then the most

significant bit of the ChipID is scanned in followed by the remainder of the ChipID. Finally, the mode is scanned in.

The last mode path is the Kill/Inject path (110 or 111). It is $160 \times 18 \times 2 = 5760$ bits long. The first 2880 bits form the kill mask, while the second 2880 bits form the inject mask. During the loading of the kill mask, the LoadKill signal (pad 50) should be held high. It is dropped low before the first bit of the inject mask is scanned in. The scan path starts at the lowest pixel of the 17th column and proceeds up the column to the highest pixel. From there, the path moves to the highest pixel in the 16th column and proceeds from there to the lowest pixel. In other words, if it is an odd numbered column, the path receives its input at the lowest pixel from the column to its right and scans the data to the highest pixel of this odd numbered column where it outputs the path to the column on its left. If it is an even numbered column, the path receives its input at the highest pixel from the column to its right and scans the data to the lowest pixel of this even numbered column where it outputs the path to the column to its left. For a sketch of this path see Figure 2.

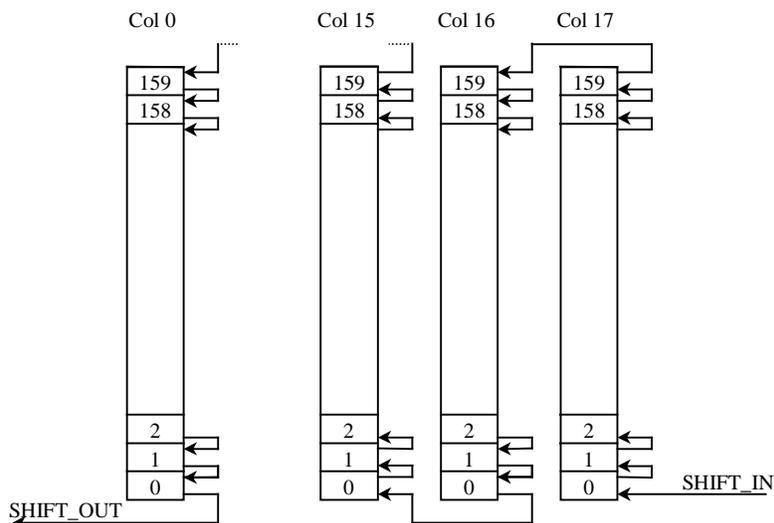


Figure 2. Kill/Inject scan path

A logical 1 in the kill pattern means that the pixel is killed. A logical 0 in the inject pattern means that the pixel will be injected.

During all scanning operations in Continuous Mode, the TrigAcc signal (pad 29) should be held to a Logical 0 (reject). This is to prevent accidental injection of hits. If possible, DataReset should be activated after programming is done and before TrigAcc is returned to a Logical 1 (accepts). In External Trigger Mode, it is impossible to use TrigAcc in this way. However, since the Trigger path is far away from the Pixels, there should be no problem with accidental hits.

All of the Scan Signals (ShClk, ShiftIn, ShiftOut, Reg[2:0] and TrigAcc) are all single-ended CMOS-level signals.

4 Data readout

The data lines (Data[0:16]) transmit different information depending on the readout cycle. The readout cycle is qualified by the Data[15] signal, which is the Control bit. The Control bit logical level “0” means header and logical level “1” means data. The header information is composed of the readout BCO (RBCO[0:5]) and the chip identification number (ChipID[0:3]). Unused data bits during header readout cycle are set to zero. During data cycle (Control = 1) the FPIX1 chip drive the row address (Row[0:7]), column address (Col[0:4]) and the analog to digital conversion count (ADC[0:1]) on the data lines (see Table 1).

Data16 behaves as Data Valid bit (see Table 1, pads 74 and 75). When Data Valid is logical 1, there is data being driven on the bus and when Data Valid is logical 0, there is no data on the bus. Data will change at the falling edge of the ReadClk and will be stable at the rising edge of the ReadClk. Therefore, the data lines Data[0:15] should be latched by the DAQ system at the rising edge of the ReadClk when Data Valid is logical level 1.

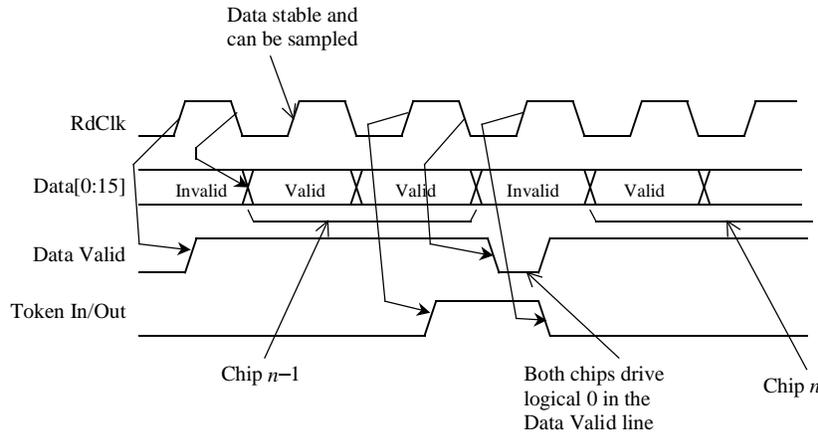


Figure 3. Data readout

FPIX1 is meant to share the same data bus. To control the bus access, a token passing scheme is used: the chip with the token can access the bus and transmit its own data. The token outputs and token inputs of the chips are daisy chained. By this reason, Token Out is placed such that it can be fed straight into the Token In of the next chip in the chain. Furthermore, to allow compact layout of the circuit interconnect, token In is the only LVDS input to have its own internal load resistor, dispensing external termination resistors.

Figure 3 assumes that two chips are connected in a daisy chain. The Token In of the next chip (identified as chip n in the figure) connects with the Token Out of the previous chip (identified as chip $n-1$ in the figure). The figure shows the behavior of the token between these two chips.

The FPIX1 rules for driving the Data Valid line are the following:

- a) If the chip $n-1$ is driving the data on the data bus, the Data Valid line is active (not in tri-state) and is logic level 1.
- b) If the chip $n-1$ is driving the token, Data Valid is active, and it can be logic level 1 or 0.
- c) When chip n receives the token, it examines the state of the Data Valid line. If Data Valid is logic level 1, it means that the previous neighbor chip (chip $n-1$) is driving Data Valid and data on the bus, and chip n can not drive Data Valid. When Data Valid drops to logical level 0, the neighbor is no longer driving data on the bus, and chip n can activate the Data Valid output and drive a logical level 0. At this moment both chips will be forcing logical level 0 on the Data Valid line.

These rules are depicted in Figure 3. Observe the note that says that both chips, chip n and chip $n-1$, are driving the Data Valid line.

If the chip is to be used in a single chip mode, its Token In can be set to a logical

1. In this case, its Token Out will be a square wave at on half the ReadClk frequency.

5 Resetting

Pulsing Program Reset will do the following:

1. The External Trigger will be reset to zero.
2. Every mask register in every column will be restored to all zeros. In other words, the End-of-column Sets will reset themselves 64 clock cycles after they have been hit.
3. The chip will be reset to Continuous Mode
4. The ChipID will be reset to 0000.
5. The lag will be reset to 2 (000010).

The Kill/Inject pattern will NOT be affected. The Current and Requested BCO Counters are NOT affected.

Pulsing Data Reset will do the following:

1. Reset the Current BCO Counter to 0.
2. Reset the Requested BCO Counter to -4 (111100)
3. Reset all hit pixels by forcing all End-of-column Command Sets to the Reset Command
4. Reset all End-of-column Priority Encoders to point to their respective Register A.
5. Reset all End-of-column Sets to "empty".

It is highly recommended that, if in Continuous Mode, that TrigAcc (pad 29) be set to logic level 0 (reject) while performing either reset. Also, it is highly recommended that all clocks be set to the logic level 0 for the duration of the reset.

6 TrigAcc

In Continuous Mode, TrigAcc is the throttle. If the system is overloaded or if some situation arises during which it would be better not to accept any hits (such as resetting), TrigAcc should be set to logical level 0 (reject). This will prevent the command interpreter in each pixel from accepting hits from the analog section of each pixel. In

other words, all hits will be ignored for the duration of the reject. Under normal operation, TrigAcc should be set to logical level 1 (accept).

In External Trigger Mode, TrigAcc is the Trigger signal. At the falling edge of this signal, a new external trigger scanned in along the Trigger path will be presented to the End-of-column Logic cells. It should be noted that the scanning of external triggers can be pipelined with data output. In other words, scanning in a new trigger while the chip is outputting data from an old trigger is the expected mode of operation, and the old trigger will not be corrupted until the TrigAcc signal is pulsed.

7 Bias Currents and Reference Voltages

FPIX1 has external bias currents and reference voltages. The external bias current pads are 4, 6, 8, 10, 12 and 14 and the voltage references are 2, 16, 18, 20, 22 and 24. Pad 26 is the bias for the detector and is not used internally by FPIX1.

Only two bias currents are necessary. (Ivbbp and Ivfb). The other currents are there for decoupling and possible optimization. The other bias currents can be used for optimization and override some internal bias. Decoupling capacitors in all bias current inputs improves the performance.

The reference voltage are the threshold voltages Vth0, Vth1, Vth2 and Vth3 and the analog voltage to inject charge. The Vth's voltages should be decoupled with small caps. We will now describe the key bias currents and voltages references.

7.1 Threshold Voltage for the Hit Comparator

One FPIX1 chip was characterized to measured the relationship between threshold voltage for the hit comparator (Vth0) and charge input (Qth). This result is depicted in Figure 4. The slope of the graph is approximately $18 e^- / mV$.

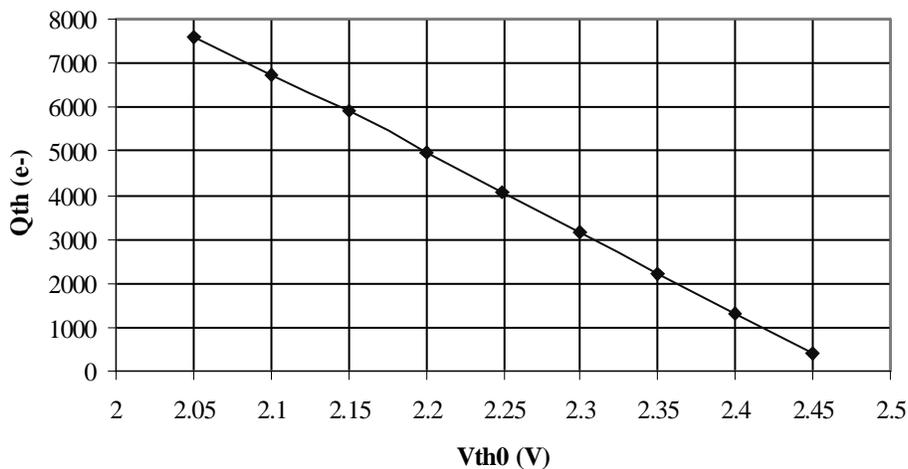


Figure 4. Qth as a function of Vth

8 Differential Outputs

The single ended outputs and inputs are CMOS outputs. The differential outputs use LVDS like drivers. The “true” version of the differential signal is the one identified without asterisk, while the “bar” version has the asterisk. For example, Figure 5 shows the case for the differential Data0 signal. When internal signal Data0In is logical 1, the voltage in Data0 is higher than Data0*. The opposite happens when the Data0In is logical 0.

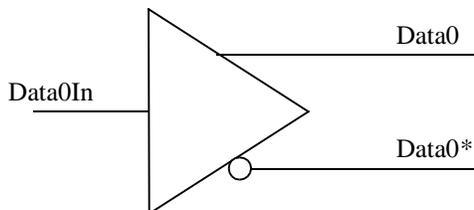


Figure 5. Differential output

The block diagram of the differential outputs is shown in Figure 6. The transistors operate like switches, sourcing or sinking current from the outputs OA and OB. The control of the transistors (signals A+ and A-) are such that the transistors behave in the following way: when the transmitter input is set to logical 1, transistors T_{A1} and T_{B2} are closed, while T_{A2} and T_{B1} are open; when the input of the transmitter is set to logical 0, we have the opposite, transistors T_{A2} and T_{B1} are closed, and T_{A1} and T_{B2} are open. The value of R is 500 Ω.

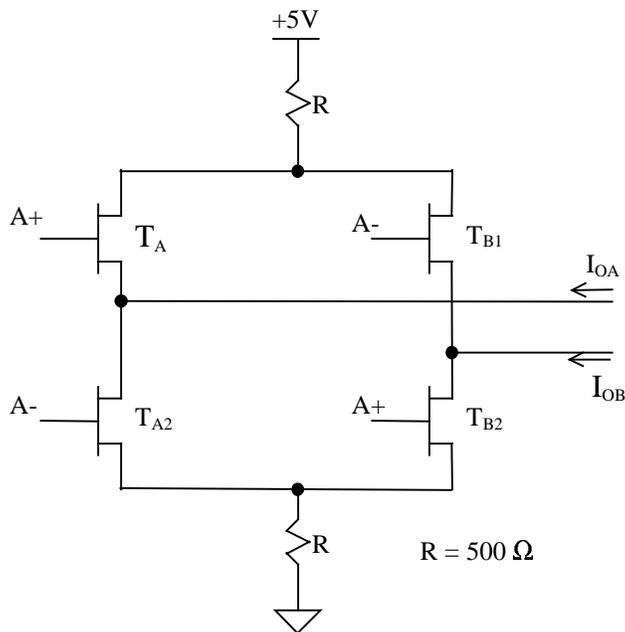


Figure 6. Block diagram of the differential drivers

9 Pad location and chip dimensions

Figure 7 shows the pad sequence on the top of the chip dye.

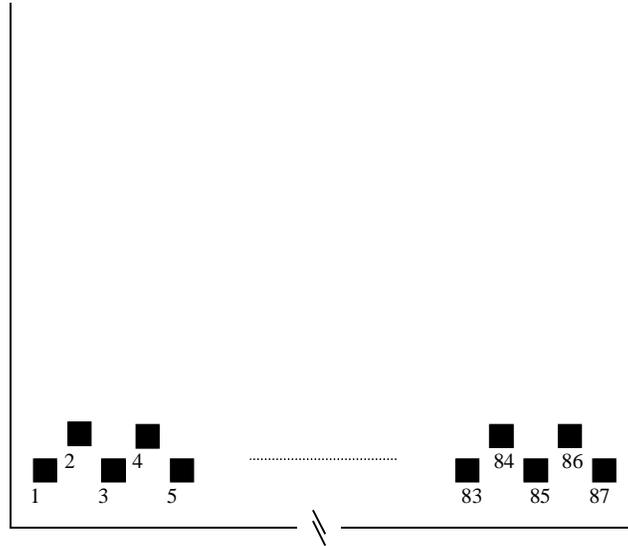


Figure 7. Pad location and dimensions

The chip has an area of $7.5 \times 12.5 \mu\text{m}$. Figure 8 shows several dimensions of the FPIX1.

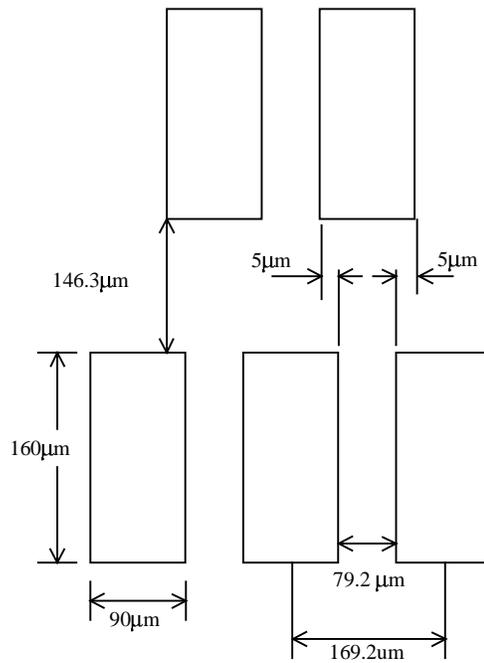
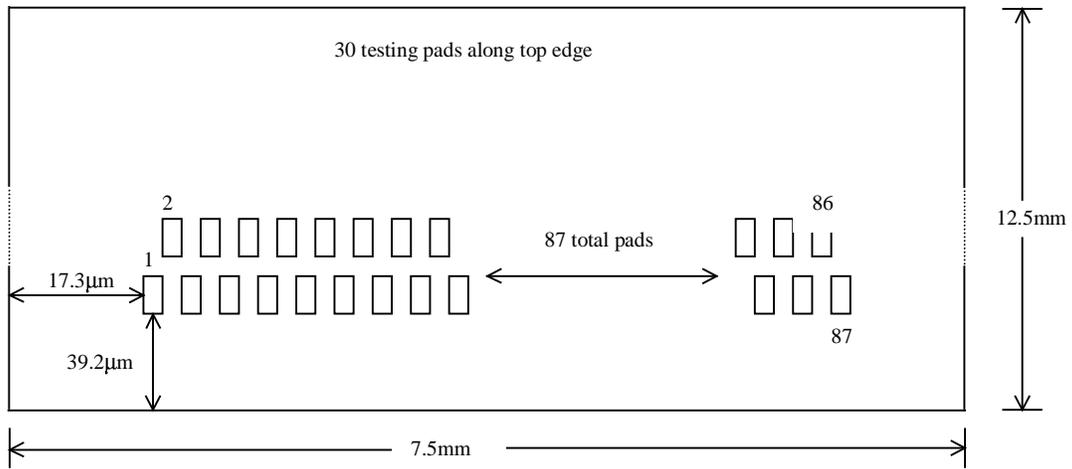


Figure 8. Chip dimensions

10 Appendix I

This appendix is a copy of the article published on the proceedings of the Fourth Workshop on Electronics for LHC Experiments, pages 528-532.

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 $2 \times 10^{32} \text{ cm}^{-2} \text{ s}^{-1}$ (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 $50 \times 400 \mu\text{m}$ 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 9). 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" BCO 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 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.

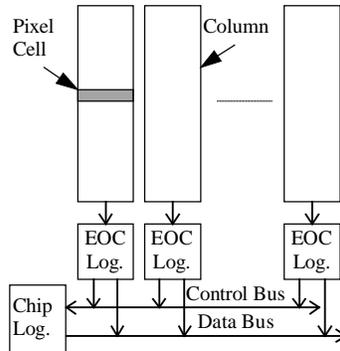


Figure 9. FPIX1 Block Diagram

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 10) 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.

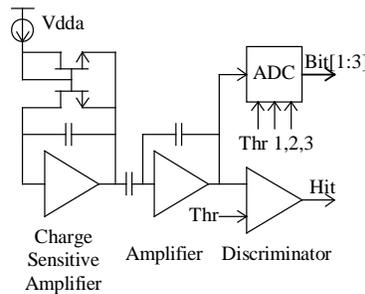


Figure 10. Front-end

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 11. 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.

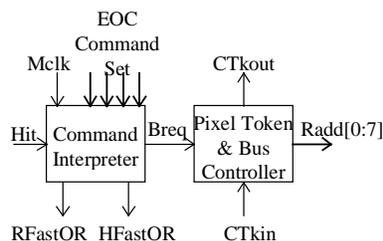


Figure 11. Pixel Cell Digital Interface

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 12 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.

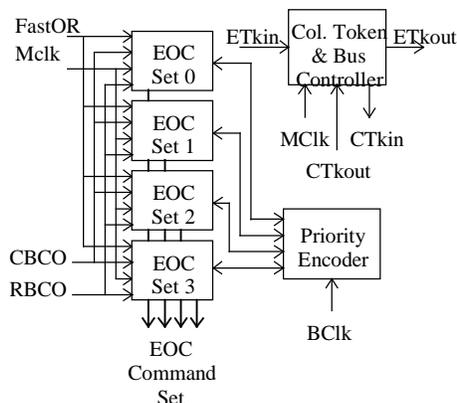


Figure 12. End Of Column Logic

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 $50 \times 450 \mu\text{m}$. 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 $50 \times 400 \mu\text{m}$ 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 $2 \times 10^{32} \text{ cm}^{-2} \text{ s}^{-1}$, 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.