

FPIX2.1 Specification

Version 1.15

October 5, 2007

David Christian

1. Introduction

FPIX2.1 consists of four logical sections: the core, the programming interface, the programmable registers and digital to analog converters, and the data output interface. A photograph of the chip is shown in Figure 1 and a block diagram in Figure 2.

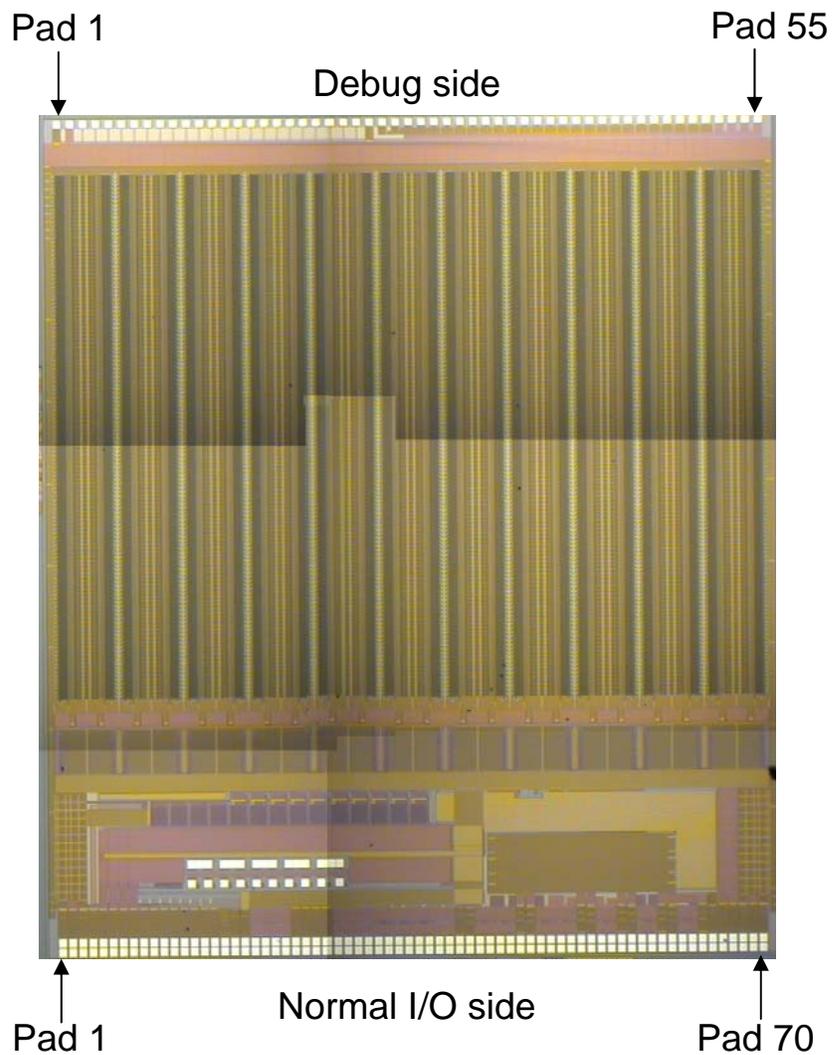


Figure 1: FPIX2.1 photograph.

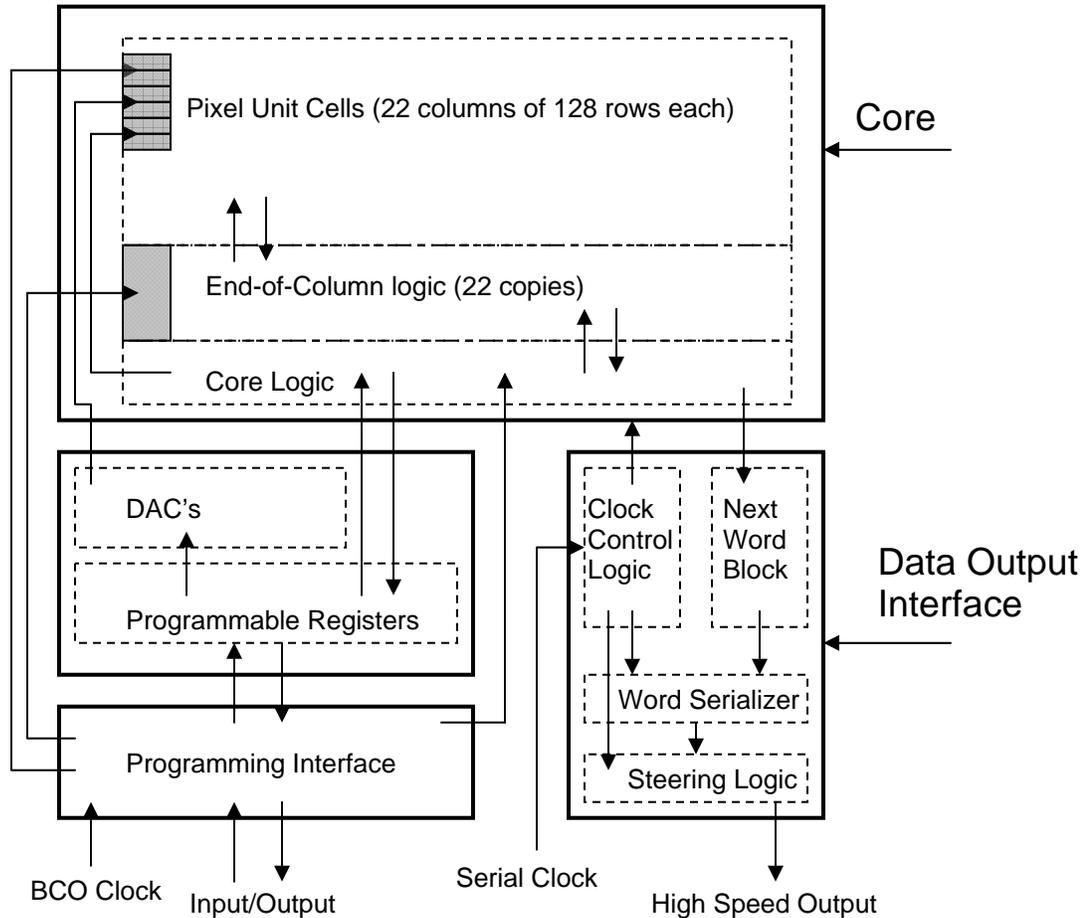


Figure 2: FPIX2.1 block diagram. Arrows represent control and data flow.

The core consists of the pixel unit cells, each of which contains an amplifier and a flash ADC, end-of-column logic associated with each column of pixels, and core logic, which controls the flow of data from the core to the data output interface. The programming interface accepts commands and data from a serial input bus, and, in response to commands, provides data on a serial output bus. The programmable registers are used to hold input values for the DAC's that provide currents and voltages required by the core, such as the discrimination threshold and the threshold levels for each of the FADC bits. The data output interface accepts data from the core, serializes the data, and transmits it off chip using a point-to-point protocol. All I/O (except the test signal inject and the analog output signal from pixel 0,0) is differential and uses Low Voltage Differential Signaling (LVDS), as illustrated below (Figure 3). See Table 1 for a listing of the FPIX2.1 wire bonding pads.

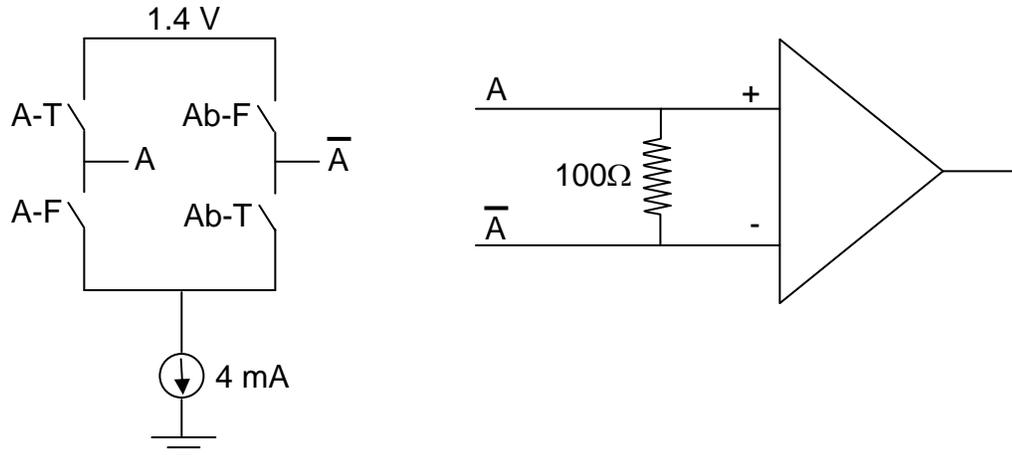


Figure 3: The drawing on the left is a simplified sketch of the FPIX2 LVDS drivers. When the signal being driven is TRUE, switches A-T and Ab-T are closed, and switches A-F and Ab-F are open. The voltage of A is 1.4V, and when the signals are terminated as shown on the right hand side of the figure, 4mA flows through the 100Ω external resistor to Ab, making its voltage 1.0V. When the signal is FALSE, switches A-F and Ab-F are closed, A-T and Ab-T are open, and the voltages and direction of current flow are reversed.

| Chip Pin | Signal Name | Class | Type | LVDS? | Comp. | Notes |
|----------|---------------|---------|------|-------|-------|---|
| 1 | Ground (vssa) | Analog | Pwr | | | Ground for Pixel Analog Frontends |
| 2 | Analog Power | Analog | Pwr | | | Power for Pixel Analog Frontends |
| 3 | Ground (vssa) | Analog | Pwr | | | Ground for Pixel Analog Frontends |
| 4 | InjectIn | Analog | Ana | | | Controls test injection (50 ohms to ground) |
| 5 | Digital Power | Digital | Pwr | | | Power for Pixel Backends and EOC Logic |
| 6 | Ground | Digital | Pwr | | | Ground for Pixel Backends and EOC Logic |
| 7 | Digital Power | Digital | Pwr | | | Power for Programming Interface |
| 8 | Ground | Digital | Pwr | | | Ground for Programming Interface |
| 9 | RefRes | Analog | Ana | | | Establishes all analog voltages in FPIX2. Must be tied to Analog Ground through 700k ohms |
| 10 | Vref | Analog | Ana | | | Analog voltage for debug purposes (bypass) |
| 11 | Vbbp | Analog | Ana | | | Analog voltage for debug purposes (bypass) |
| 12 | Vbbp1 | Analog | Ana | | | Analog voltage for debug purposes (bypass) |
| 13 | Vmaster | Analog | Ana | | | Analog voltage for debug purposes (bypass) |
| 14 | Vth0 | Analog | Ana | | | Analog voltage for debug purposes (bypass) |
| 15 | AnaOut00 | Analog | Ana | | | The analog output of the pixel at Row 0 Col 0. Must be tied to Analog Power through 600 ohms. |
| 16 | nGuard | Analog | Pwr | | | NGuard voltage for silicon detectors |
| 17 | Analog Power | Analog | Pwr | | | Power for Pixel Analog Frontends |
| 18 | Ground | Analog | Pwr | | | Ground for Pixel Analog Frontends |
| 19 | Ground | Analog | Pwr | | | End-of-column logic substrate connection |
| 20 | FFRb | Digital | In | Y | 21 | "Firefighter" reset (comp.) |
| 21 | FFR | Digital | In | Y | 20 | "Firefighter" reset |
| 22 | BCOCIk | Digital | In | Y | 23 | Beam Crossover Clock (comp.) |
| 23 | BCOCIk | Digital | In | Y | 22 | Beam Crossover Clock |
| 24 | Ground | Digital | Pwr | | | Core Ground |
| 25 | Digital Power | Digital | Pwr | | | Core Power |
| 26 | Ground | Digital | Pwr | | | Core Ground |
| 27 | Digital Power | Digital | Pwr | | | Core Power |
| 28 | ShiftCtrlb | Digital | In | Y | 29 | Shift Control (comp.) |
| 29 | ShiftCtrl | Digital | In | Y | 28 | Shift Control |
| 30 | ShiftInb | Digital | In | Y | 31 | Shift Input (comp.) |
| 31 | ShiftIn | Digital | In | Y | 30 | Shift Input |

| | | | | | | |
|------|---------------|---------|-----|---|----|--------------------------------------|
| 32 | ShiftOutb | Digital | Out | Y | 33 | Shift Output (comp.) |
| 33 | ShiftOut | Digital | Out | Y | 32 | Shift Output |
| 34 | GotHitb | Digital | Out | Y | 35 | Asynchronous Chip Hit signal (comp.) |
| 35 | GotHit | Digital | Out | Y | 34 | Asynchronous Chip Hit signal |
| 36 | ChipTalkingb | Digital | Out | Y | 37 | Chip is outputting data (comp.) |
| 37 | ChipTalking | Digital | Out | Y | 36 | Chip is outputting data |
| 38 | serialClkb | Digital | In | Y | 39 | Master Clock (comp.) |
| 39 | serialClk | Digital | In | Y | 38 | Master Clock |
| 40 | Ground | Analog | Pwr | | | End of Column Substrate |
| 41 | Digital Power | Digital | Pwr | | | Data Interface Power |
| 42 | Out6b | Digital | Out | Y | 43 | (LSB) Serial Output 6 (comp.) |
| 43 | Out6 | Digital | Out | Y | 42 | (LSB) Serial Output 6 |
| 44 | Out5b | Digital | Out | Y | 45 | Serial Output 5 (comp.) |
| 45 | Out5 | Digital | Out | Y | 44 | Serial Output 5 |
| 46 | Ground | Digital | Pwr | | | Data Interface Ground |
| 47 | Digital Power | Digital | Pwr | | | Data Interface Power |
| 48 | Out4b | Digital | Out | Y | 49 | Serial Output 4 (comp.) |
| 49 | Out4 | Digital | Out | Y | 48 | Serial Output 4 |
| 50 | Out3b | Digital | Out | Y | 51 | Serial Output 3 (comp.) |
| 51 | Out3 | Digital | Out | Y | 50 | Serial Output 3 |
| 52 | Ground | Digital | Pwr | | | Data Interface Ground |
| 53 | Digital Power | Digital | Pwr | | | Data Interface Power |
| 54 | Out2b | Digital | Out | Y | 55 | Serial Output 2 (comp.) |
| 55 | Out2 | Digital | Out | Y | 54 | Serial Output 2 |
| 56 | Ground | Digital | Pwr | | | Data Interface Ground |
| 57 | Digital Power | Digital | Pwr | | | Data Interface Power |
| 58 | Out1b | Digital | Out | Y | 59 | (MSB) Serial Output 1 (comp.) |
| 59 | Out1 | Digital | Out | Y | 58 | (MSB) Serial Output 1 |
| 60 | Ground | Digital | Pwr | | | Data Interface Ground |
| 61 | Digital Power | Digital | Pwr | | | Data Interface Power |
| 62 | OutClkb | Digital | Out | Y | 63 | Serial Output Clock (comp.) |
| 63 | OutClk | Digital | Out | Y | 62 | Serial Output Clock |
| 64 | Ground | Digital | Pwr | | | Data Interface Ground |
| 65 | Ground | Analog | Pwr | | | Data Interface Substrate |
| 66 | Ground | Digital | Pwr | | | End of column Ground |
| 67 | Digital Power | Digital | Pwr | | | End of column Power |
| 68 | Ground | Analog | Pwr | | | Ground for Pixel Analog Frontends |
| 69 | Analog Power | Analog | Pwr | | | Power for Pixel Analog Frontends |
| B_70 | Ground | Analog | Pwr | | | Ground for Pixel Analog Frontends |

Table 1a: FPIX2.1 wire-bonding pads (normal I/O side).

| Chip Pin | Signal Name | Notes |
|----------|--------------------|---|
| 1 | Ground (vssa) | Ground for Pixel Analog Frontends |
| 2 | Ground (vssa) | Ground for Pixel Analog Frontends |
| 3-24 | AOUT(0-21) | Analog out for pixel row 127 (Column 0-21) |
| 25 | Ground (vssa) | Ground for Pixel Analog Frontends |
| 26 | Ground (vssa) | ESD protection for digital outputs |
| 27 | Ground (vdda) | ESD protection for digital outputs |
| 28 | Digital Power | Vddd |
| 29 | Ground (vssd) | Digital ground |
| 30-51 | Digital Out (0-21) | Comparator output for pixel row 127 (Column 0-21) |
| 52 | Ground (vssd) | Digital ground |
| 53 | Digital Power | Vddd |
| 54 | Ground (vssd) | Digital ground |
| 55 | Digital Power | Vddd |

Table 1b: FPIX2.1 wire-bonding pads (debug side). The analog outputs must be pulled up to 2.3 volts (nominally through 604Ω). The digital outputs are single-ended CMOS signals.

With nominal settings, the rise time (to 90%) of typical signals is about 40 ns. A very large value of I_{bb} can be used to decrease the rise time to about 30 ns. As shown in Figure 5, large signals fall at a constant rate determined by I_{ff} , and small signals return to zero with an RC time constant determined by the feedback capacitor and the small-signal feedback resistance. A typical signal takes many microseconds to return to baseline. If a faster fall time is desired, I_{ff} may be increased from its nominal value.

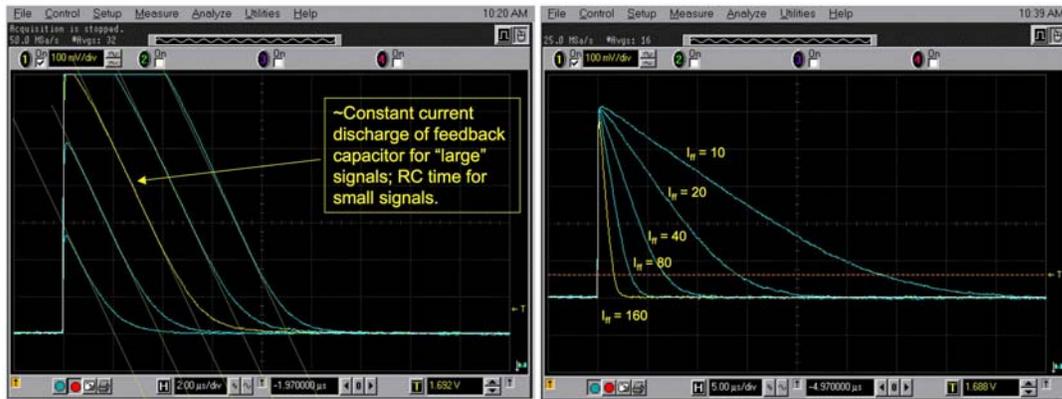


Figure 5: I_{ff} provides both leakage current compensation and return-to-zero for large pulses¹.

The second stage of the amplifier is intended as a gain stage only, with a gain of four, determined by the ratio of the feedback capacitor to the coupling capacitor. V_{ref} sets the DC operating point of the second stage (the DC voltage at the signal input to the comparators is V_{ref}). The difference between V_{ref} and V_{fb2} controls the resistance of the second stage feedback transistor. If V_{fb2} is too close to V_{ref} , the feedback resistance becomes essentially infinite and the second stage of the amplifier becomes unstable and exhibits a low-frequency oscillation. The threshold of each of the eight comparators is determined by the voltage difference between V_{ref} and $V_{th0} - V_{th7}$. A hit is registered whenever the output of the second stage (a negative-going signal) goes below V_{th0} . The seven other comparators, together with an ecoder, form a three bit flash ADC. All of the threshold DAC's cover the same range, and in each case the least significant bit corresponds to approximately 200 electrons at the input of a pixel amplifier.

¹ The pictures shown in Figure 5 were taken with an FPIX2.1 biased as designed. As is documented in EPP-doc-338, unintended feedback loops make FPIX2.1 susceptible to a number of different modes of instability. As is explained in that document, a work-around has been developed that allows stable operation. The work-around involves setting a different unit current for the DACs than anticipated in the design. This changes the numerical values required for many of the registers.

lbp1 controls the bias of the first stage of the amplifier, and lbp2 is a similar control for the second stage.

If the “inject” switch is closed (see section 4), a test pulse may be injected through a (nominally) 3 fF capacitor. Since the inject switch is a simple PMOS switch, the DC voltage of the test signal must always be greater than ~ 0.8 V (a negative voltage will close all of the inject switches). The test signal is intended to have a DC voltage of ~ 2 V. The test pulse should have a fast (< 10 ns) negative step followed by a slow ramp back to 2V. For small amplitudes, a square pulse can be used. However, if a large amplitude square pulse is used, the amplifier will saturate in response to the rising edge and take a very long time to recover. A step of 1V corresponds to an input of about 20000 electrons.

2.1.2 Digital Section

Essentially an explanation of the command interpreter... maybe also some on read out.

2.2 End-of-Column Logic

2.3 Core Logic

3. Programming Interface

The programming interface provides a means for the user to control the operation of FPIX2.1, and to load and read back the contents of any of the programmable registers. Serial commands are input to the programming interface using the “shift control” and “shift in” lines. Commands are shifted in high order bit first. When “shift control” is high, “shift in” is latched into the input register of the programming interface on the falling edge of the BCO clock. “Shift control” must also be kept high after a “read” command while data is being output on “shift out.” After a “read” command, the contents of the requested register are output on “shift out.” There is a one-cycle delay after the end of a “read” command before the output starts to appear. Data is shifted out on the rising edge of the BCO clock. “Shift control” is latched on rising edges of the BCO clock. The required timing of “shift control” and “shift in” is shown in Figure 6.

The programming interface will respond to all broadcast (wild chip address = 10101) commands, and to all commands in which the chip address matches the contents of the “chip address register”, which is set by internal wire bonds (if the wire bonds are not present, the chip address defaults to the complement of the wild chip address, 01010). Each command consists of 5 bits of chip address, followed by a 5-bit register number, and a 3-bit instruction code. For Write commands only, the instruction code is followed by data, which is written to the

specified register. With one exception, all “set,” “reset,” and “default” commands affect a register for an entire BCO clock period, starting on the rising edge immediately after the last instruction bit is latched. The exception is the “AqBCO, set” (Acquire current BCO number) command, which is executed on the first negative going BCO clock edge after “shift control” goes low. The FPIX2.1 command format is illustrated in Figure 6, and the instruction codes are listed in Table 2 below.

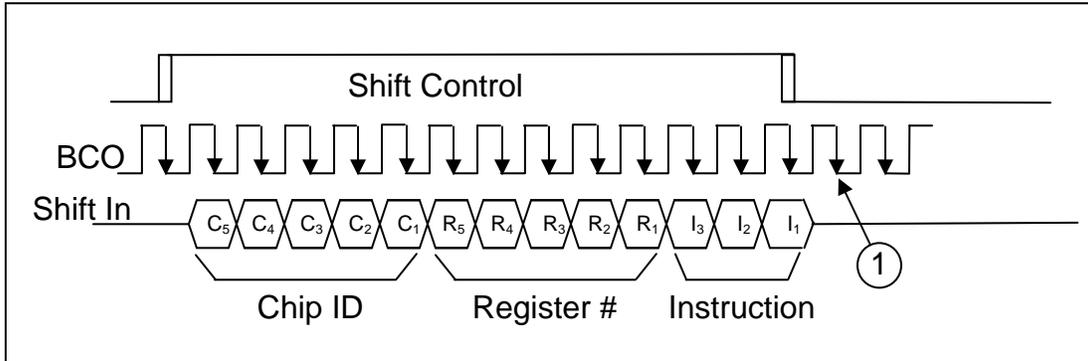


Figure 6: Programming Interface Input Format. The “Acquire BCO” set command is executed at the clock edge indicated by the number 1 in a circle.

| Instruction | Code |
|---|-------------|
| Write (followed by 2, 8, or 2816 bits of data) | 001 |
| Set (all bits in register = 1) | 010 |
| Read | 100 |
| Reset (all bits in register = 0) | 101 |
| Default (set register to default value) | 110 |

Table 2: Programming Interface Instructions.

4. Programmable Registers and DAC’s

The registers are listed in Table 3. The table also lists how the Firefighter Reset and the Smart Programming Reset affect each register.

The first 14 registers in Table 3 (Ibp – Vth7) are all 8-bit registers. These registers hold the values that are input to the digital to analog converters.

AqBCO is also an 8-bit register. When AqBCO is Set, the value of the BCO counter is loaded into the AqBCO register. This register can be read at any later time to verify BCO synchronization. BCO synchronization can also be checked without reading the AqBCO register if the AqBCO Set command is timed so that

the BCO counter value latched should equal zero. If the AqBCO register contents are not zero, a bit is set in the sync/status word (see section 5).

The “active lines” register, Alines, is a 2-bit register. This register, which is implemented using redundant logic designed to be immune to single event upset, determines the data output configuration (see section 5).

Kill and Inject are serpentine registers running up and down the pixel columns. Each register has one bit in each pixel unit cell ($22 \times 128 = 2816$ bits in each register). Kill=1 opens a switch at the output of the pixel discriminator, effectively killing the pixel. Inject=1 closes a switch connecting the Test Signal Inject line to the charge injection capacitor associated with a pixel.

SendData, RejectHits, and Mod256 are SEU tolerant single bit registers. These registers must be programmed using SET & RESET (rather than WRITE). They cannot be read using READ; their values have to be determined from the sync/status word (see Section 5 below). SET RejectHits inhibits the core from accepting any more pixel hits (data already latched is not affected). RESET SendData disables the core readout. If data is being read out when SendData is RESET, up to two data words may be lost (either one or none on the transition from 1 to 0, and either one or none on the transition back from 0 to 1). If Mod256 is SET (or after a system reset), the BCO counter is a normal 8-bit binary counter. If Mod256 is RESET, the BCO counter counts modulo 159 (0-158). This mode is appropriate for use at the Tevatron collider, which has a harmonic number of 1113. If the BCO clock ticks every 7th RF clock cycle, then one Tevatron orbit is equal to $1113/7 = 159$ clock cycles.

Three of the register addresses do not correspond to physical registers. WildReg (10101) is used in broadcast commands that, for instance, set all 8-bit registers to their default values. Two “registers” are actually commands, which are executed when the “register” is Set. SCR is the Smart Core Reset. This command clears all pixels and end-of-column logic, and resets the BCO counter to zero. SPR is the Smart Programming Reset. SPR resets the bias currents to default values and all thresholds to default values, but does not affect the data output configuration, the Kill and Inject registers, or SendData or RejectHits. SCR and SPR are discussed more fully in Section 6, which also contains a description of the hardware (“Firefighter”) reset.

All of the registers except Kill and Inject are loaded least significant bit (b0) first. All of the registers except Kill and Inject can be read non-destructively at any time. After a read command, the requested register contents are copied to a shadow register, then shifted out, most significant bit first. Note this bit order is opposite to the order used to load the register.

Kill and Inject are loaded by shifting data into the register in the order illustrated in Figure 7. The bit intended for column 21, row 0, is input first. The bit intended

for column 0, row 0, is input last. Kill and Inject cannot be read non-destructively, as no shadow register is implemented. After a read command, data from Kill or Inject appears on “shift out” in the same order that it was loaded – (21,0) first. As the data is read out, it is also shifted back through the entire shift register, so that at the end of the read operation (if “shift control” is lowered just after the last bit is shifted out), the register contents are restored.

| Register Name | Address | Firefighter Reset | Smart Prog Reset | Notes |
|----------------|-------------------------|---------------------|------------------|--|
| | 00000 ₂ (0) | | | Forbidden |
| lbp1 | 00001 ₂ (1) | Default=100 | Default | (Default values are given as base 10 numbers.) |
| lbp2 | 00010 ₂ (2) | Default=74 | Default | |
| lbb | 00011 ₂ (3) | Default=29 | Default | |
| lff | 00100 ₂ (4) | Default=13 | Default | |
| Vref | 00101 ₂ (5) | Default=202 | Default | |
| Vfb2 | 00110 ₂ (6) | Default=172 | Default | |
| Vth0 | 00111 ₂ (7) | Default=8 | Default | |
| Vth1 | 01000 ₂ (8) | Default=8 | Default | |
| Vth2 | 01001 ₂ (9) | Default=8 | Default | |
| Vth3 | 01010 ₂ (10) | Default=8 | Default | |
| Vth4 | 01011 ₂ (11) | Default=8 | Default | |
| Vth5 | 01100 ₂ (12) | Default=8 | Default | |
| Vth6 | 01101 ₂ (13) | Default=8 | Default | |
| Vth7 | 01110 ₂ (14) | Default=8 | Default | |
| AqBCO | 01111 ₂ (15) | Default=0 | Default | |
| Alines | 10000 ₂ (16) | 00 ₂ (0) | Unchanged | |
| Kill | 10001 ₂ (17) | “no kill” | Unchanged | Ignores WildReg |
| Inject | 10010 ₂ (18) | “no inj” | Unchanged | Ignores WildReg |
| SendData | 10011 ₂ (19) | 0(noSend) | Unchanged | Ignores WildReg |
| RejectHits | 10100 ₂ (20) | 1(reject) | Unchanged | Ignores WildReg |
| WildReg | 10101 ₂ (21) | | | |
| Mod256 | 10110 ₂ (22) | 1(normal 8-bit BCO) | Unchanged | Ignores WildReg |
| | 10111 ₂ (23) | | | Unused in FPIX2 |
| SPR | 11000 ₂ (24) | Unaffected | Unaffected | Ignores WildReg |
| | 11001 ₂ (25) | | | Unused in FPIX2 |
| | 11010 ₂ (26) | | | Unused in FPIX2 |
| | 11011 ₂ (27) | | | Unused in FPIX2 |
| SCR | 11100 ₂ (28) | Unaffected | Unaffected | Ignores WildReg |
| | 11101 ₂ (29) | | | Unused in FPIX2 |
| | 11110 ₂ (30) | | | Unused in FPIX2 |
| | 11111 ₂ (31) | | | Forbidden |

Table 3: FPIX2 Programmable Registers.

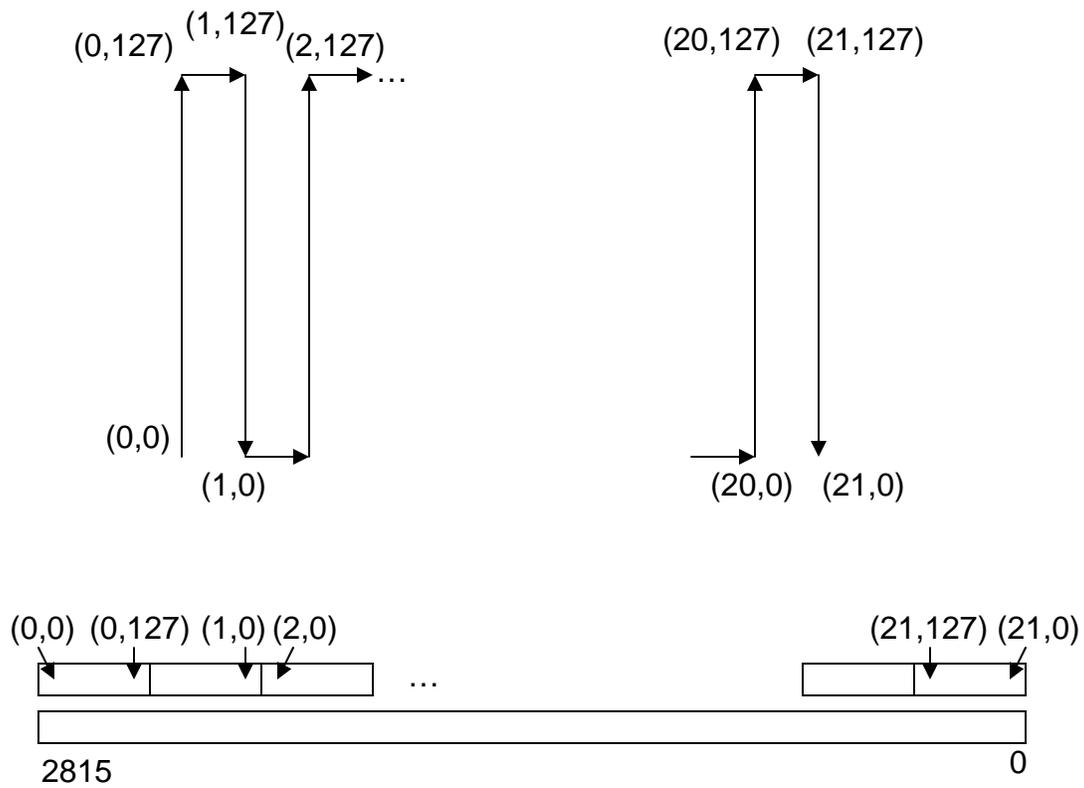


Figure 7: The Kill and Inject registers are loaded in the order indicated above. The bit intended for pixel column 21, row 0, is shifted in first. The bit intended for (0,0) is shifted in last. The registers are read back in the same order: (21,0) appears on “shift out” first, (0,0) appears last.

5. Data Output Interface

There are four functional blocks in the data output interface: the clock control logic, the next word block, the word serializer, and the steering logic. The relationship between these blocks is shown in Figure 8.

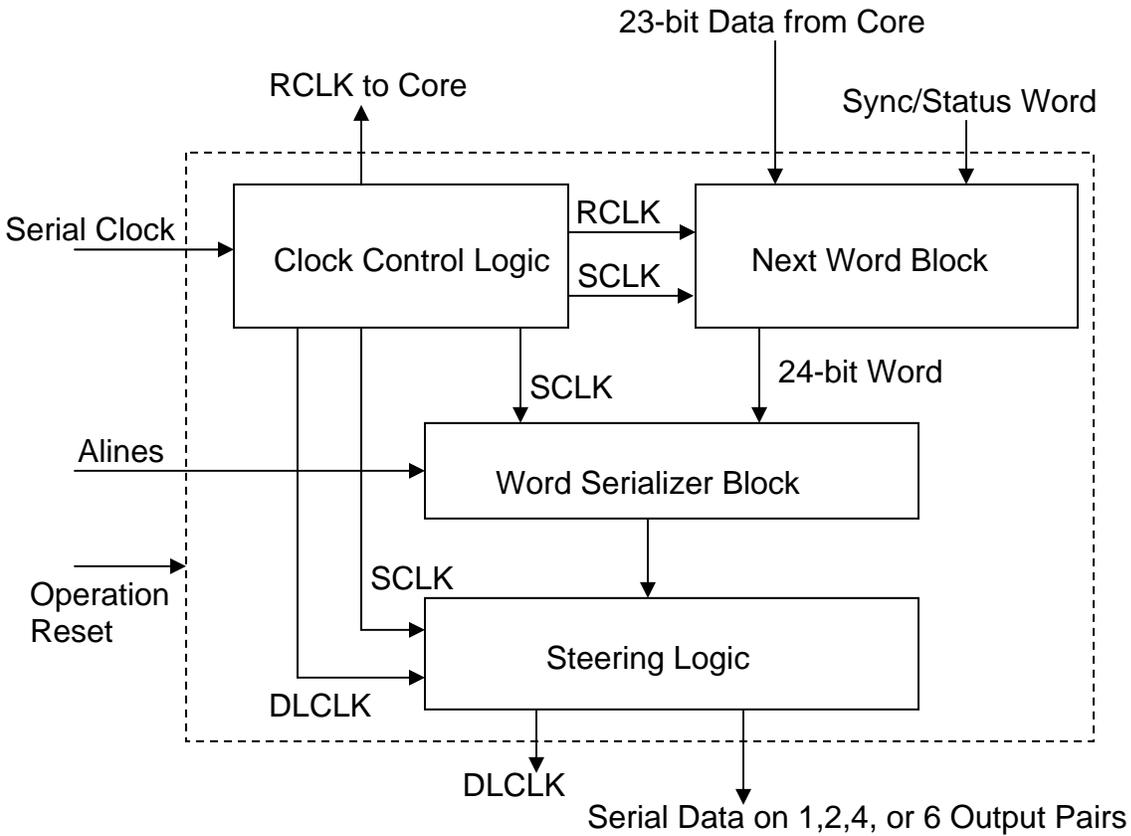


Figure 8: Data Output Interface block diagram.

The (nominally 140 MHz) Serial Clock is used for data read out and transmission. The clock control logic block uses the serial clock to derive the two clocks used internally to control read out (RCLK and SCLK), as well as the output Data Latch Clock (DLCLK), which is output along with the serialized data.

The core Readout Clock (RCLK) used by the FPIX core is derived using a counter. SCLK is a simple copy of the Serial Clock input to the chip. The frequency of RCLK depends on the number of active data output lines (see table below), and is given by dividing SCLK by the number of bits serialized on each output path. The relationship between SCLK, the number of active lines, and RCLK insures that data is output from the chip at the same rate as it is read out of the core. This means that no extra buffer memory is required in the data output interface.

The output Data Latch Clock (DLCLK) is one half the frequency of SCLK. Output data is asserted on the positive going edges of SCLK; DLCLK transitions occur on the negative going edges of SCLK.

| Configuration | RCLK Frequency |
|----------------|----------------|
| 6 output pairs | (SCLK) ÷ 4 |
| 4 output pairs | (SCLK) ÷ 6 |
| 2 output pairs | (SCLK) ÷ 12 |
| 1 output pair | (SCLK) ÷ 24 |

Table 4: Dependence of RCLK on data output configuration.

The next word block selects the source of the next word to be input to the word serializers. If “core talking” is asserted and it has been asserted for at least one RCLK cycle, then the “core data word” is selected and latched into flip flops in the output stage of the next word block. Otherwise, the “sync word” is selected and latched into the same flip-flops. This data transfer takes place on the falling edge of RCLK. When the core horizontal token reaches column number 21, there is at least one RCLK cycle in which “core talking” is not asserted. When the horizontal token is launched again (in response to “core has data”), one more RCLK cycle is required before valid data is delivered from the core to the data output interface. This guarantees that the “sync/status word” will be output at least twice every time that the FPIX2 readout scans through all 22 columns. As illustrated below, the next word block adds a word mark bit, which is guaranteed to be one, to the data. The PDCB uses the sync/status word to establish the location of the 24-bit word boundary. It can distinguish between the sync/status word and core data because the sync/status word has 13 zeros in bits 1-13. The word mark bit, and the pixel column number encoding, ensures that core data can never have 13 consecutive bits equal to zero, either within a word, or across two data words. The most significant 10 bits (b14 – b23) of the sync/status word are reserved for status and error codes. Table 5 lists the bit assignments.

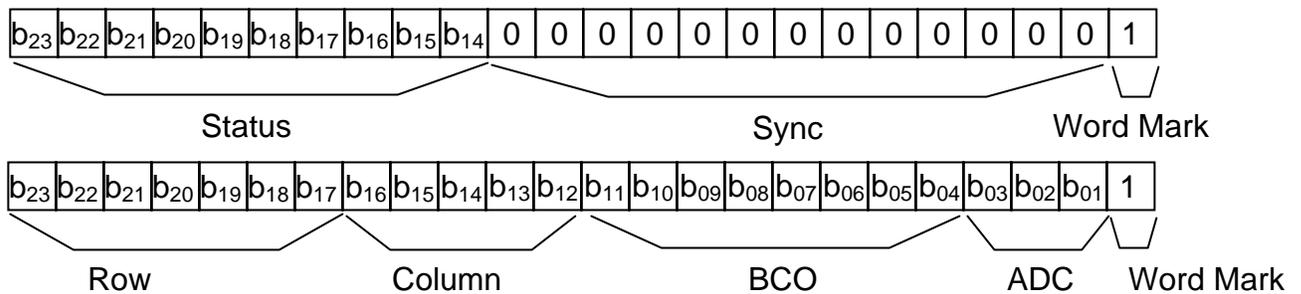


Figure 9: Format of sync/status word (top) and data word (bottom).

| Bit Number | Meaning | Bit Number | Meaning |
|------------|----------------|------------|--------------------|
| 23 | SendData | 18 | Modulo 256 |
| 22 | RejectHits | 17 | Chip Address b_4 |
| 21 | Alines-b1 | 16 | Chip Address b_2 |
| 20 | Alines-b0 | 15 | Chip Address b_1 |
| 19 | AqBCO $\neq 0$ | 14 | Chip Address b_0 |

Table 5: Status bit assignment in sync/status word.

| Col. # | Code |
|--------|-------|--------|-------|--------|-------|--------|-------|
| 0 | 11011 | 6 | 00110 | 12 | 11110 | 18 | 10010 |
| 1 | 00001 | 7 | 00111 | 13 | 01101 | 19 | 10011 |
| 2 | 00010 | 8 | 11010 | 14 | 01110 | 20 | 11101 |
| 3 | 00011 | 9 | 01001 | 15 | 01111 | 21 | 10101 |
| 4 | 11111 | 10 | 01010 | 16 | 11001 | | |
| 5 | 00101 | 11 | 01011 | 17 | 10001 | | |

Table 6: Column numbering code.

The word serializer block serializes the data for output. On the falling edge of RCLK, the word held in the flip flops at the output of the next word block is latched into the serializer flip flops. This data is output serially to the steering logic in 1, 2, 4, or 6 parallel paths depending on the status of the “active lines” register (Alines), which controls the data output configuration.

| Alines Contents | Number of Output Pairs |
|-----------------|------------------------|
| 00 | 1 |
| 01 | 2 |
| 10 | 4 |
| 11 | 6 |

Table 7: Active lines register code.

The steering logic drives the output data clock and the serial output data off chip. DLCLK is one half the frequency of SCLK and is phased so that its edges fall one half way between the edges of the data lines.

6. Resets

The FPIX2.1 has one hardware reset (“fire fighter reset”) and two software resets (“smart core reset” and “smart programming reset”). It is expected that all chips on a sensor module will share the hardware reset lines, so that all chips will be reset at once. The software reset commands may be broadcast to all chips on a module, or sent to one chip only.

When a “smart core reset” is received, the BCO counter is reset to zero, all end-of-column logic is cleared, and all pixel hits are cleared.

When a “smart programming reset” is received, all of the registers that hold values input to DAC’s are reset to their default values, and the AqBCO register is cleared. Since the default value for all of the threshold registers is zero (maximum threshold), this effectively disables all pixels even though RejectHits is unchanged.

When a “fire fighter reset” is received, the FPIX2 is reset to a “safe” mode. The BCO counter is reset to zero, all end-of-column logic is cleared, and all pixel hits are cleared. All DAC registers are reset to their default values and AqBCO is cleared. The data output interface is also reset. This consists of zeroing all of the word serializer registers, halting RCLK and SCLK, then starting RCLK and SCLK again such that when data is shifted out of the chip, the word mark bit is accompanied by a 0 → 1 transition of DLCLK. In addition, the “fire fighter reset” resets Alines, Kill, Inject, SendData, and RejectHits. This means that the number of active output pairs is reset to one, no pixel unit cell is killed and none connected to the charge injection signal, data output from the core is inhibited, and all pixel unit cells are set to ignore new hits.

The various reset actions are summarized in Table 3.

7. Procedures

It is anticipated that the following procedure will be followed when turning on an FPIX2.1:

- 1) Fire Fighter Reset – place FPIX2 in a “safe” mode.
- 2) Write, Alines, # – configure the Data Output Interface (DOI) to use the proper number of output lines (if more than one pair is used).
- 3) Wait for sync word to be received, confirming DOI configuration.
- 4) Write, Vth0 – Vth7, # – download the thresholds.
- 5) Write, ..., # – download any other register values required (pixels to be killed, bias currents, Vref, ...).
- 6) **Reset**, RejectHits – enable the core.

- 7) Set, SCR – smart core reset (see below for timing necessary to insure system wide BCO number synchronization).
- 8) Set, SendData – enable the data readout.

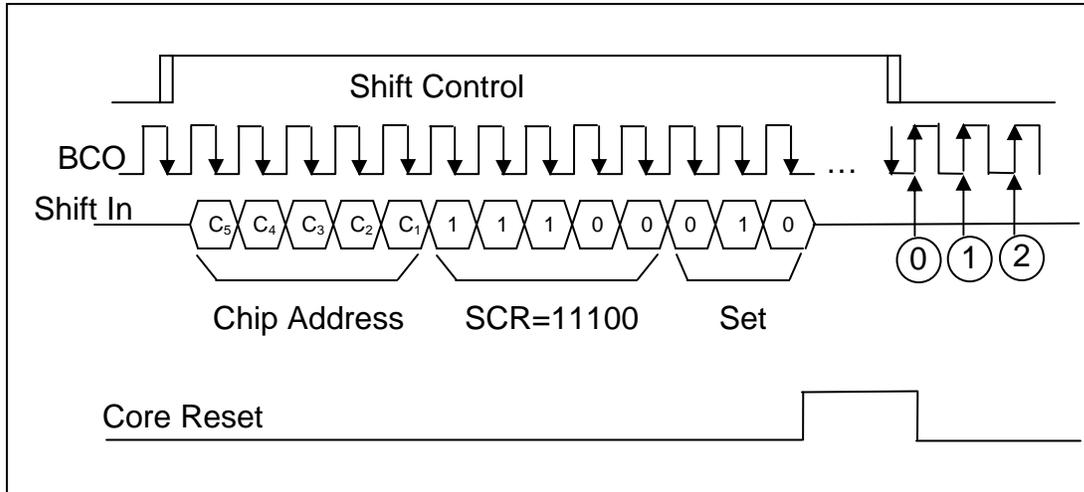


Figure 10: The “Core Reset” line is asserted on the rising edge of the BCO clock following the last bit of the SCR,Set command. It stays active until just after the first BCO clock rising edge after “shift control” has been lowered. The next rising edge of the BCO clock will increment the BCO counter from zero to one, so the BCO numbers will be assigned as indicated by the numbers in circles.

If an FPIX2 loses BCO synchronization, then a smart core reset will recover synchronization without requiring the DOI or programmable registers to be reset. As shown in Figure 10, core data will be lost until shift control is dropped at the appropriate time to restart the BCO counter at zero in sync with the rest of the system. The core will become active approximately 5 ns into BCO period 0.

Appendix A: Revision History

2-Apr-08: Various updates, including the register default values, added by T. Engelmores & D. Winter

11/1/06: Changed title to "FPIX2.1;" added diagram of pixel unit cell & description of the analog section of the unit cell; deleted table summarizing bond pads; changed reference to MCA/MCB in Figure 1. – DCC

1/22/07: Corrected Table xx (list of registers)... old table still referred to FPIX2A.

10/2/07: Corrected many references to FPIX2 features; added a description of the feature that allows BCO clock cycles to be counted mod(159).

10/5/07: Added chip photograph & table of debug-side pad assignments.