The Fermilab Silicon Strip Readout test chip

Pier Francesco Manfredi[c], Massimo Manghisoni[b,c],
Lodovico Ratti[a,c], Valerio Re[b,c], Valeria Speziali[a,c], Jim Hoff[d],
Abderrezak Mekkaoui[d], Raymond Yarema[d],
David Christian[d], Simon Kwan[d]

[a] Università di Pavia, Dipartimento di Elettronica, Via Ferrata 1, 27100 Pavia, Italy

[b] Università di Bergamo, Dipartimento di Ingegneria, Viale Marconi 5,
24044 Dalmine (BG), Italy

[c] INFN, Sezione di Pavia, Via Bassi 6, 27100 Pavia, Italy

[d] Fermi National Accelerator Laboratory, Batavia, IL 60510-0500, USA

November 15th, 2003

Summary

A new chip is being developed for the readout of the Silicon Tracker in the BTeV experiment at Fermilab. A first prototype of the chip was submitted in July 2003. It includes the full back-end section and, for test purpose, different versions of the analog channel. This note contains an overview of the test chip and its contents.

1 Introduction

The Fermilab Silicon Strip Readout (FSSR) chip is a mixed-signal circuit occupying an area of 7.27×4.46 mm^2. It can be described as including four logical sections, as shown in Fig. 1. They are the core, the programmable registers and digital-to-analog converters, the programming interface and the data output interface.

The core consists of 128 analog readout channels, logically subdivided in 16 sets of 8 channels each, the end-of-set logic (16 blocks, one for each set of front-end channels) and the core logic, which controls the data flow from the core to the data output interface. The programming interface accepts commands and data from a serial input bus and, in response to a command, provides data on a serial output bus. The programmable registers are used to hold input values for DACs that provide currents and voltages required by the core, for instance the threshold levels for the discriminator in the analog channel. They have other functions, such as controlling data output speed and selecting the pattern for charge injection tests. The data output interface accepts data from the core, serializes the data and transmits them off chip, using a point to point protocol. Note that for test purposes (explained later in the document) the present version of the chip includes only 114 analog channels and needs no DACs for threshold

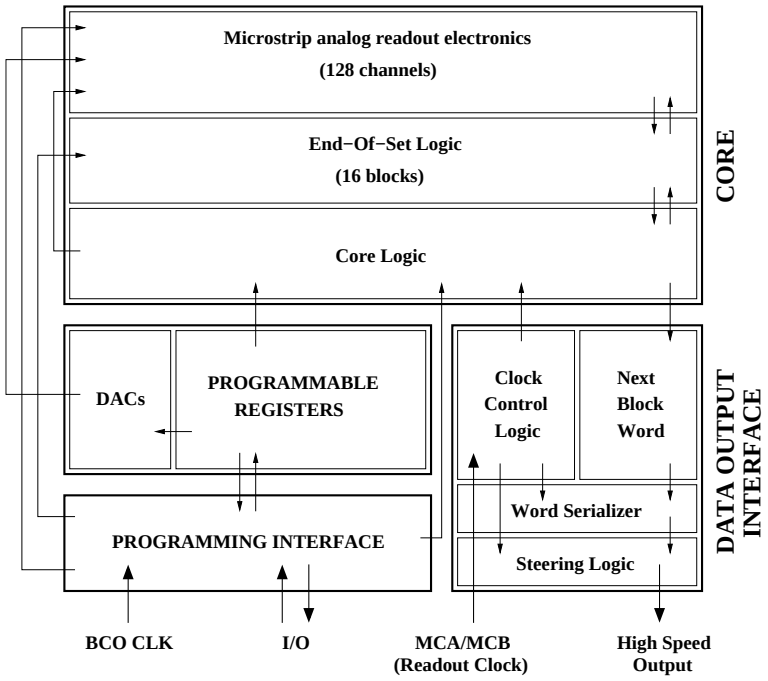

Figure 1: FSSR chip test block diagram. Arrows represent control and data flow.

programming. All I/O (except the test signal inject) is differential and is fed by means of Low Voltage Differential Signaling (LVDS), which is based on the principle illustrated in Fig. 2. Tables 1, 2 and 3 list all of the wire bonding pads (each 100×100 μm^2 in area) in the present test version of the FSSR chip. Analog bias is fed to the analog channel blocks, with the exception of the discriminator. The chip layout, including the bonding pad diagram, is shown in Fig. 3.

Other pads (five +2.5 V bias pads, five ground pads and five input pads), which are not listed in the tables, are used for setting the chip address register by internal wire bonds. Furthermore, each of the 114 channels has an input pad for connection to a microstrip. These pads are 150 μm long and 75 μm wide but are arranged in such a way that they have a 50 μm effective pitch (see Fig. 4). There are 13 more analog input pads not connected with any channel (see next sections for further details).

A charge injection test pad (InjectIn) has been placed among the analog channel input pads. It can be used to check the chip response to an injected charge when the analog input pads are connected to the detector and are no longer available for test through direct charge injection. Eight single devices, seven NMOS and one PMOS, with the same gate dimensions as the input elements of the charge preamplifiers, have been included in the chip for test purposes. Four wire bonding pads (drain, source, gate and substrate or well) are associated with each of them. The single devices, with their channel dimensions,

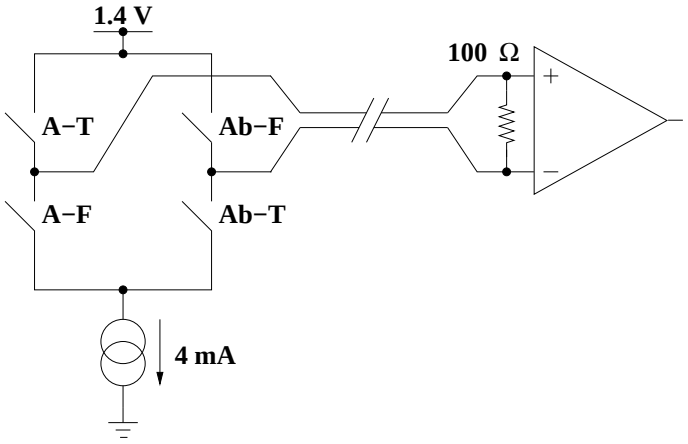

Figure 2: schematic principle of the Low Voltage Differential Signaling concept. When the signal being driven is TRUE, switches A-T and Ab-T are closed, and switches A-F and Ab-F are open. The voltage on the non-inverting amplifier input is 1.4 V and 4 mA flows through the 100 Ω resistor, making the voltage of the inverting terminal 1 V. 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 the current flow are reversed.

Pad No.	Pad Name and Description	Pad No.	Pad Name and Description
1	VSSA - Substrate	41	ShiftInb - Shift In
2	VDDA - +2.5 V Analog Bias	42	ShiftIn - Shift In
3	GNDA - Analog Ground	43	ShiftOutb - Shift Out
4	VSSA - Substrate	44	ShiftOut - Shift Out
5	VSSA - Substrate	45	ORb - Operation Reset
6	VDDA - +2.5 V Analog Bias	46	OR - Operation Reset
7	GNDA - Analog Ground	47	MClkAb - Master Clock A
8	GND - Dis Ground	48	MClkA - Master Clock A
9	VDD - +2.5 V Dis Bias	49	MClkBb - Master Clock B
10	VDDD - +2.5 V Digital Bias	50	MClkB - Master Clock
11	VSSD - Digital Ground	51	Subs(eoc) - Substrate
12	VDDD - +2.5 V Digital Bias	52	VDDD(di) - +2.5 V DOI Bias
13	VSSD - Digital Ground	53	Out6b - Data Output
14	VSSA - Substrate	54	Out6 - Data Output
15	VDDD - +2.5 V Digital Bias	55	Out5b - Data Output
16	VSSD - Digital Ground	56	Out5 - Data Output
17	VDDD - +2.5 V Digital Bias	57	VSSD(di) - DOI Ground
18	VSSD - Digital Ground	58	VDDD(di) - +2.5 V DOI Bias
19	PrmpIntVref - Vref in Prmp/Int	59	Out4b - Data Output
20	PrmpVbn2 - Vbn2 in Prmp	60	Out4 - Data Output
21	PrmpVbp2 - Vbp2 in Prmp	61	Out3b - Data Output
22	PrmpVbp3 - Vbp3 in Prmp	62	Out3 - Data Output
23	Prmpvbn1 - Vbn1 in Prmp	63	VSSD(di) - DOI Ground
24	IntVbn1 - Vbn1 in Int	64	VDDD(di) - +2.5 V DOI Bias
25	ShpBlrDisRef - Vref in Shp/Blr/Dis	65	Out2b - Data Output
26	DisVtn - Threshold Voltage in Dis	66	Out2 - Data Output
27	DisVtp - Threshold Voltage in Dis	67	VSSD(di) - DOI Ground
28	BlrVbp1 - Vbp1 in Blr	68	VDDD (di) - +2.5 V DOI Bias
29	ShpVbp2 - Vbp2 in Shp	69	Out1b - Data Output
30	ShpVbp1 - Vbp1 in Shp	70	Out1 - Data Output
31	FFRb - Firefighter Reset	71	VSSD(di) - DOI Ground
32	FFR - Firefighter Reset	72	VDDD(di) - +2.5 V DOI Bias
33	BCOClkb - BCO Clock	73	OutClkb - Output Clock
34	BCOClk - BCO Clock	74	OutClk - Output Clock
35	VSSD (c) - Core Ground	75	VSSD(di) - DOI Ground
36	VDDD (c) - +2.5 V Core Bias	76	Subs(di) - Substrate
37	VSSD (c) - Core Ground	77	ChipHasData - Diagnostic Signal
38	VDDD (c) - +2.5 V Core Bias	78	ChipHit - Diagnostic Signal
39	ShiftCtrlb - Shift Control	79	PPrmpVbp2 - Vbp2 in PPrmp
40	ShiftCtrl - Shift Control	80	PPrmpVbn2 - Vbn2 in PPrmp

Table 1: FSSR wire bonding pads.

Pad No.	Pad Name and Description	Pad No.	Pad Name and Description
81	PPrmpVbp3 - Vbp3 in PPrmp	91	VDDD - +2.5 V Digital Bias
82	PPrmpVbn1 - Vbn1 in PPrmp	92	VDD - +2.5 V Discriminator Bias
83	VSSD - Digital Ground	93	GND - Discriminator Ground
84	VDDD - +2.5 Digital Bias	94	GNDA - Analog Ground
85	VSSD - Digital Ground	95	VDDA - +2.5 V Analog Bias
86	VDDD - +2.5 V Digital Bias	96	VDDA2 - PMOS Prmp Bias
87	VSSA - Substrate	97	VSSA - Substrate
88	VSSD - Digital Ground	98	GNDA - Analog Ground
89	VDDD - +2.5 V Digital Bias	99	VDDA - +2.5 Analog Bias
90	VSSD - Digital Ground	100	VSSA - Substrate

Table 2: FSSR wire bonding pads (cont'd).

GLOSSARY

PI	Programming Interface
Prmp	Preamplifier
Int	Integrator
Shp	Shaper
Blr	Baseline Restorer
Dis	Discriminator
BCO	Beam Cross Over
DOI	Data Output Interface
PPrmp	PMOS Input Preamplifier

34 Analog Pads	31 DOI Pads	14 PI Pads	21 Miscellaneous Pads
6 Ground	5 Ground	4 Ground	6 Ground
7 Power	5 Power	4 Power	6 Power
5 Substrate	1 Substrate	2 Shift Control	3 Substrate
16 Internal Voltage and Current Biases	12 Signal Output	2 Shift In	2 BCO Clock
	2 Clock Output	2 Shift Out	2 Firefighter Reset
	2 Master Clock A		2 Operation Reset
	2 Master Clock B		
	2 Diagnostic Signals		

Table 3: FSSR test chip pads by functional group.

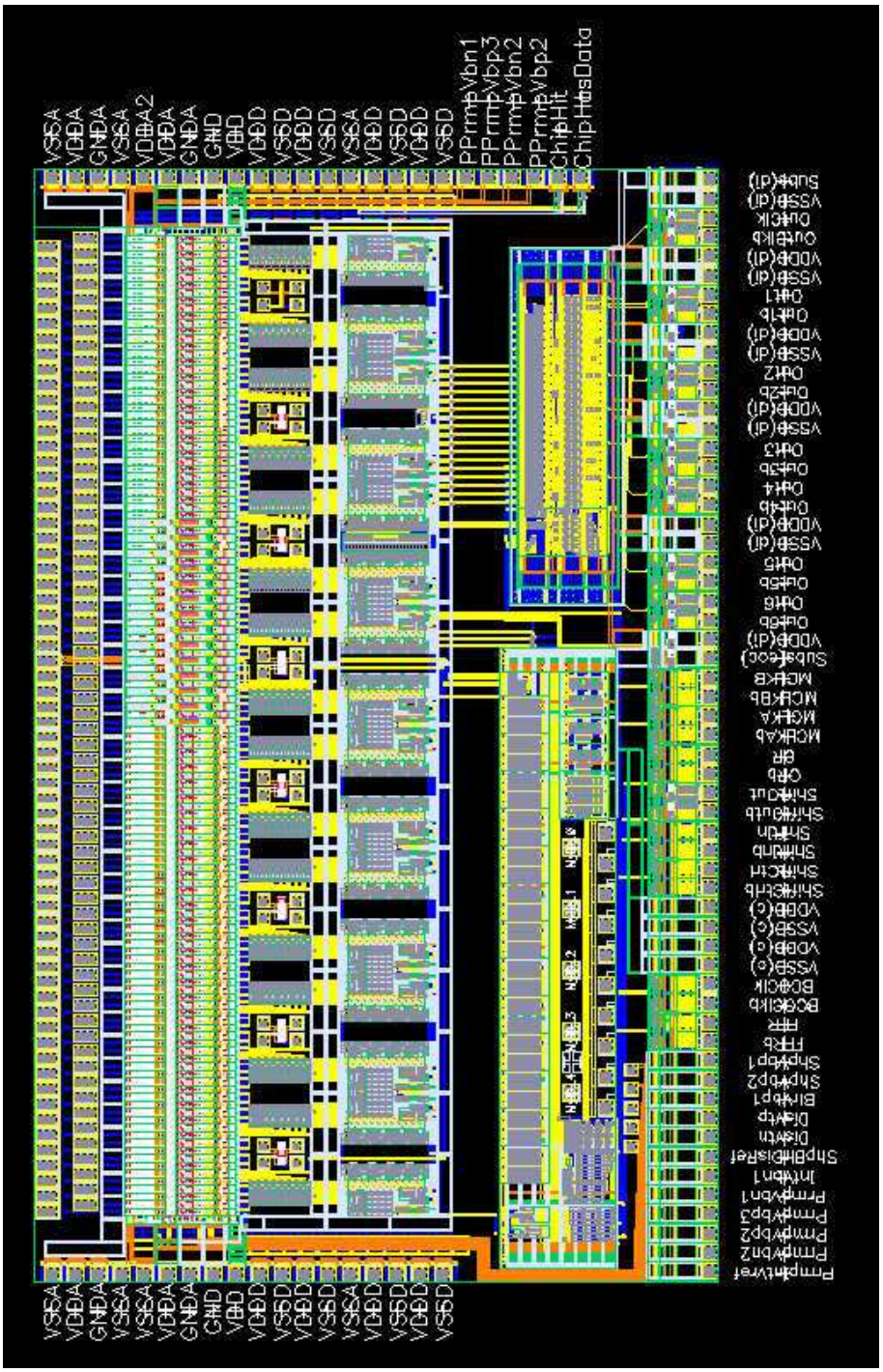

Figure 3: layout of the FSSR test chip with wire bonding pad diagram.

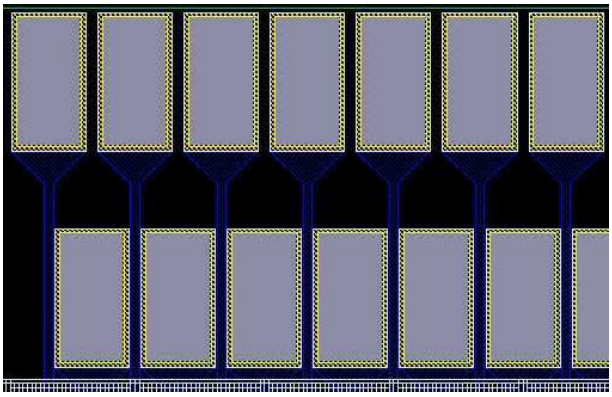

Figure 4: analog channel input pads for connection to microstrip detectors.

are listed in Table 4. Finally, other 56 probing pads (four for each of 14 individual channels) have been included in the chip layout for direct acquisition of the response at the output of the four main blocks of the analog channel.

2 FSSR Core

In this test version, the analog section of the FSSR Core consists of 114 channels (the space of the 14 missing ones is used for the probing pads of 14 individual channels), each including a charge preamplifier, an integrator, a shaper and a discriminator. A symmetric baseline restorer was added in some of the circuits in order to evaluate whether any significant advantage, compared with the increase in the circuit complexity, might be gained by baseline shift suppression. The block diagram of the analog channel is shown in Fig. 5. Detailed schematics of

Device No.	**Device Type**	**Gate Dimensions W/L** $[\mu m/\mu m]$
1	NMOS	1500/0.35
2	NMOS	1500/0.45
3	NMOS	1500/0.55
4	NMOS	1000/0.45
5	NMOS	2000/0.45
6	NMOS	1500/0.35
7	NMOS	1500/0.35
8	PMOS	1220/0.35

Table 4: Single devices included in the FSSR test chip.

the single analog stages can be found in the appendix.

Following is a list of the 114 analog channels with their main features:

- **0 to 63**: channels with NMOS input preamplifier (W/L=1500/0.45)
  - **even numbers**: without baseline restorer
  - **odd numbers**: with baseline restorer
- **64 to 91**:
  - **even numbers**: blank channels (surface used for probing pads)
  - **odd numbers**: individual channels with probing pads
    - **65**: channel with NMOS input preamplifier (W/L=1500/0.35)
    - **67**: channel with NMOS input preamplifier (W/L=1500/0.45) and baseline restorer
    - **69**: channel with NMOS input preamplifier (W/L=1500/0.45)
    - **71**: channel with NMOS input preamplifier (W/L=1500/0.55)
    - **73**: channel with NMOS input preamplifier (W/L=1000/0.35)
    - **75**: channel with NMOS input preamplifier (W/L=1000/0.45)
    - **77**: channel with NMOS input preamplifier (W/L=1000/0.55)
    - **79**: channel with NMOS input preamplifier (W/L=2000/0.35)
    - **81**: channel with NMOS input preamplifier (W/L=2000/0.45)
    - **83**: channel with NMOS input preamplifier (W/L=2000/0.55)
    - **85**: channel with PMOS input preamplifier (W/L=1220/0.35)
    - **87**: channel with PMOS input preamplifier (W/L=1220/0.35) and baseline restorer

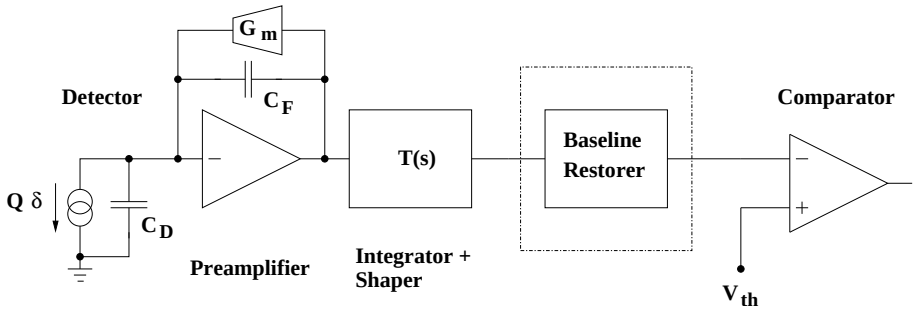

Figure 5: block diagram of the analog channel. G_m is the preamplifier transconductor, C_F is the preamplifier feedback capacitance, C_D is the detector simulating capacitor and V_{th} is the discriminator threshold voltage. For the sake of simplicity, integrator and shaper are represented by a single block, whose transfer function is T(s).

    - **89**: channel featuring a preamplifier with PMOS input (W/L=1220/0.35) and without compensation capacitance in the local loop of the folded cascode stage
    - **91**: channel with PMOS input preamplifier (W/L=1220/0.35)

- **92 to 103**: channels with PMOS input preamplifier (W/L=1220/0.35)
- **104 to 115**: channels with PMOS input preamplifier (W/L=1220/0.35) and baseline restorer
- **116 to 127**: channels featuring a preamplifier with PMOS input (W/L=1220/0.35) and without compensation capacitance in the local loop of the folded cascode stage

The layout of one of the individual channels, with the relevant probing pads, is shown in Fig. 6.

Table 5 displays the wire bonding pads referring to internal biases of the analog channels. For the circuit to work properly, some of the pads (namely pads no. 24, 26, 27, 28, 29 and 30) must be externally powered, possibly using lemo connections. The other pads can be connected to pins so that they can be probed or overridden if necessary. See the appendix for further details about the location of the relevant nodes in the single stages.

InjectIn pad is number 72 among the preamplifier input pads. The adjacent pads are not connected, which provides InjectIn with better isolation. As stated in the introduction, when the chip is already wired to microstrip detectors, InjectIn pad allows it to be tested through injection of a known amount of charge. Channels to be tested can be selected by scanning in a 128 bit word carrying information about the injection pattern, which controls the switches in the schematic of Fig. 7. Further discussion on the subject can be found in the section relevant to Programmable Registers.

The Core communicates with the other FSSR logical blocks through the Core Logic. The 128 front-end channels are subdivided in 16 sets of eight channels each. Each of the 16 blocks composing the End-Of-Set Logic deals with one of the eight channel sets.

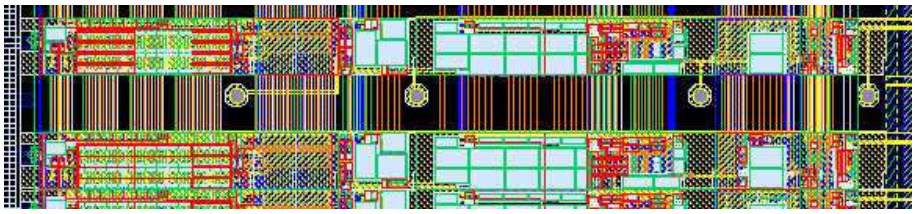

Figure 6: layout of an individual channel. The probing pads are located, from left to right, at the preamplfier, integrator, shaper and discriminator outputs.

Pad No.	Pad Name	External Bias Needed
19	PrmpIntVref	no
20	PrmpVbn2	no
21	PrmpVbp2	no
22	PrmpVbp3	no
23	PrmpVbn1	no
24	IntVbn1	yes
25	ShpBlrDisRef	no
26	DisVtn	yes
27	DisVtp	yes
28	BlrVbp1	yes
29	ShpVbp2	yes
30	ShpVbp1	yes
79	PPrmpVbp2	no
80	PPrmpVbn2	no
81	PPrmpVbp3	no
82	PPrmpVbn1	no

Table 5: Wire bonding pads referring to internal biases of the analog channel.

Operation of the FSSR Core is schematically represented in Fig. 8. The **ChipHit** and **ChipHasData** lines shown in the picture are two diagnostic signals. In particular, ChipHit goes high whenever a discriminator fires while ChipHasData goes high every time the core has data to output.

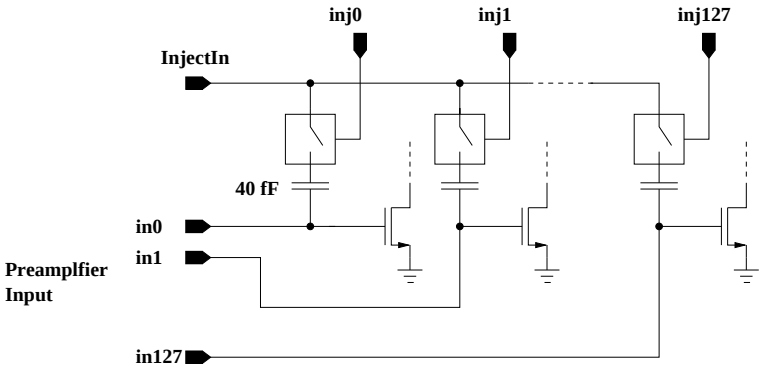

Figure 7: channel selection for test charge injection through InjectIn.

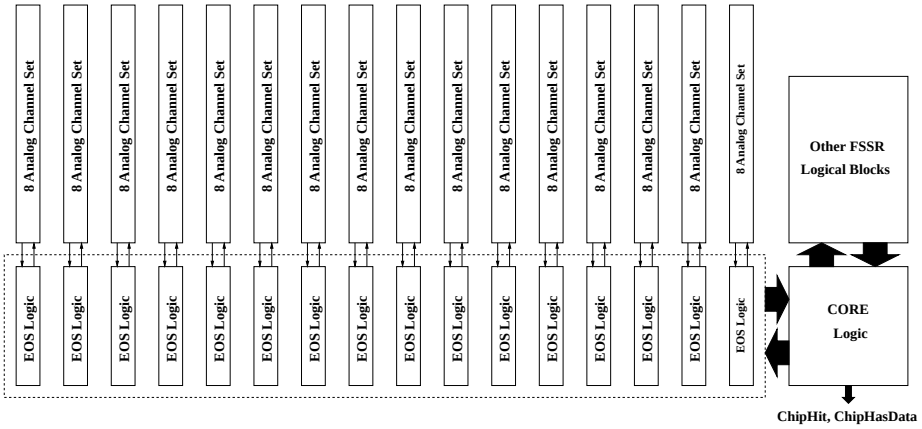

Figure 8: block diagram of the FSSR Core.

3 Programming Interface

The Programming Interface provides a means for the user to control the operation of the FSSR chip, 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 the <**Shift in**> lines. 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**. After a <**Read**> command, the contents of the requested register are output on <**Shift Out**>. <Shift Control> must also be kept high after a <Read> command while data is being output on <Shift Out>. There is a one-cycle delay before the output appears. Data is shifted out on the rising edge of the BCO clock.

The Programming Interface will respond to all broadcast commands (**wild chip address=10101**) and to all commands in which the chip address matches the contents of the **Chip Address Register** (set by internal wire bonds, as explained in the introduction). 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 FSSR command format is illustrated in Fig. 9 and the instruction codes are listed in Table 6. Programmable Registers numbering code is discussed in the next section.

Instruction	Code
Write (followed by 2, 8 or 128 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 6: Programming Interface instructions.

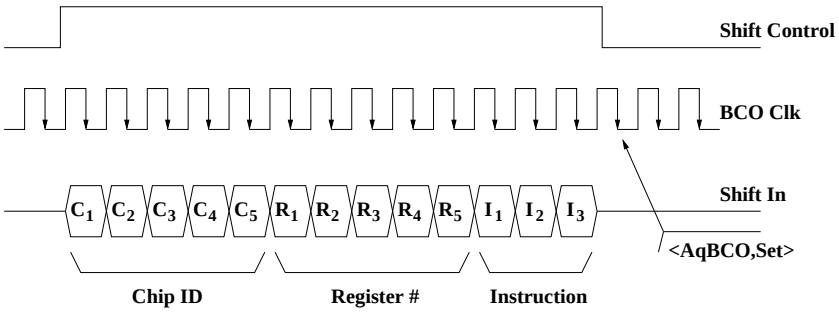

Figure 9: Programming Interface input format. The <AqBCO,Set> command is executed at the indicated clock falling edge.

4 Programmable Registers and DACs

Programmable Registers are listed in Table 7. The table also shows the number of bits for each register and how the various types of reset available in the chip (FFR=Firefighter Reset, OR=Operation Reset, SPR=Smart Programming Reset, SCR=Smart Core Reset) affect each register. Detailed description of the registers follows:

CapSel - CapSel is a 2-bit register. It is used to select the shaping time in the analog front-end channel according to Table 8

AqBCO - AqBCO is an 8-bit register. When AqBCO is set the value of the BCO counter is loaded in the AqBCO register. This register can be read at any later time to verify BCO syncronization. BCO syncronization 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 AqBCO register contents are not zero, a bit is set in the sync/status word (see section 5).

Alines - The Active Lines register, Alines, is a 2-bit register. This register, which is implemented using redundant logic designed to be immune to

Reg. Name	**Addr.**	**No. of Bits**	**FFR**	**OR**	**SPR**	**Notes**
-	00000 (0)	-	-	-	-	NU
-	00001 (1)	-	-	-	-	NU
-	00010 (2)	-	-	-	-	NU
-	00011 (3)	-	-	-	-	NU
-	00100 (4)	-	-	-	-	NU
-	00101 (5)	-	-	-	-	NU
-	00110 (6)	-	-	-	-	NU
-	00111 (7)	-	-	-	-	NU
-	01000 (8)	-	-	-	-	NU
-	01001 (9)	-	-	-	-	NU
-	01010 (10)	-	-	-	-	NU
-	01011 (11)	-	-	-	-	NU
-	01100 (12)	-	-	-	-	NU
CapSel	01101 (13)	2	00	Un	D	IW
-	01110 (14)	-	-	-	-	NU
AqBCO	01111 (15)	8	00000000	Un	D	-
Alines	10000 (16)	2	00 (0)	Un	Un	IW
Kill	10001 (17)	128	no kill(=0)	Un	Un	IW
Inject	10010 (18)	128	no inject(=0)	Un	Un	IW
SendData	10011 (19)	1	no send(=0)	Un	Un	IW
RejectHits	10100 (20)	1	reject(=1)	Un	Un	IW
WildReg	10101 (21)	-	-	-	-	-
-	10110 (22)	-	-	-	-	NU
-	10111 (23)	-	-	-	-	NU
SPR	11000 (24)	-	NA	NA	NA	IW
-	11001 (25)	-	-	-	-	NU
-	11010 (26)	-	-	-	-	NU
-	11011 (27)	-	-	-	-	NU
SCR	11100 (28)	-	-	-	-	IW
-	11101 (29)	-	-	-	-	NU
-	11110 (30)	-	-	-	-	NU
-	11111 (31)	-	-	-	-	NU

Table 7: FSSR chip Programmable Registers. NU=Not Used, Un=Unchanged, IW=Ignores WildRegister, D=Default, NA=Not Affected.

CapSel Value	Peaking Time $[ns]$
00	60
01	85
10	-
11	125

Table 8: peaking time selection in the analog channel through CapSel.

single event upset, determines the data output configuration according to Table 9.

Kill - The Kill register has one bit for each analog channel, therefore is a 128-bit register. Kill[n]=1 (n^{th} bit of the register equal to 1) opens a switch at the output of the discriminator of the n^{th} channel, effectively killing the corresponding microstrip..

Inject - Inject is a 128-bit register too. Inject[m]=1 closes a switch connecting the InjectIn test pad to the charge inject capacitor associated with the m^{th} channel (see Fig. 7).

SendData - SendData is a 1-bit, SEU tolerant register. Setting SendData=0 disables the core readout. If data is being read out when Send Data is changed from 1 to 0, up to two data words may be lost (either one or zero on the transition from 1 to 0 and either one or zero on the transition from 0 to 1).

RejectHits Like Sendata, RejectHits is a 1-bit, SEU tolerant register. Setting RejectHits=1 inhibits the core from accepting any more strip hits (data already latched is not affected).

Three of the register addresses do not correspond to physical registers:

WildRes - WildReg (10101) is used to broadcast commands that, for instance, set all 8-bit registers to their default values.

Alines Value	Number of output pairs
00	1
01	2
10	4
11	6

Table 9: active lines register code.

SCR - The Smart Core Reset is executed when the register is set. This command clears all microstrips and End-Of-Set Logic and resets the BCO counter to zero.

SPR - The Smart Programming Reset too is executed when the corresponding register is set. SPR resets the bias currents and the threshold voltages to default (safe) values, but does not affect the data output configuration, the Kill and Inject registers or SendData or RejectHits.

SCR and SPR are thoroughly discussed in section 6, which also contains a description of the two hardware resets (FFR and OR).

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 registers are loaded by shifting data in starting from the bit intended for channel 127 through the bit for channel 0. 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 (bit for channel 127 first, bit for channel 0 last). 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.

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 relationshp between these blocks is shown in Fig. 10.

The Clock Control Logic receives two clock signals from an external board managing data to and from chips belonging to the same module. These two clocks (**MCA**, Master Clock A and **MCB**, Master Clock B) are bussed to all FSSR chips in a module. MCA and MCB are both nominally 68.8 MHz clocks, with MCB lagging MCA by 90°. The Clock Control Logic block uses MCA and MCB to derive the two clocks used internally to control readout (**RCLK**, the **Readout Clock**, and **SCLK**, the **Serial Clock**), as well as the Output Clock (**OutCLK**), which is output along with the serialized data. The external board feeding the two clocks to the chips may adjust the phase relation of MCA and MCB as necessary to maximize the length of time that data received from FSSR chips is valid. Note that proper operation of the FSSR chip does not require any phase or frequency relationship between the BCO clock and the readout clocks (MCA and MCB).

The Serial Clock is derived from MCA and MCB using a simple exclusive OR. The Core Readout Clock used by the FSSR Core is derived using a counter.

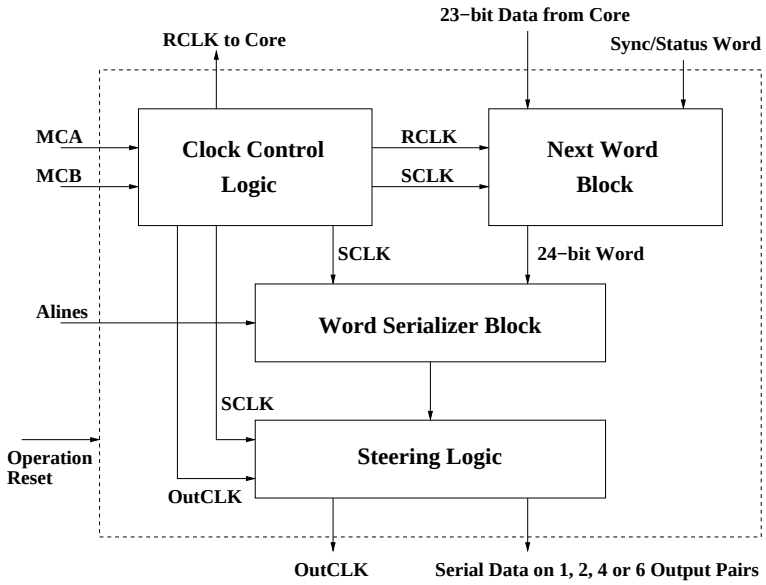

Figure 10: Data Output Interface block diagram.

SCLK is twice the frequency of MCA and MCB. The frequency of RCLK depends on the number of active data output lines (see Table 10) 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 Clock, which has the same frequency as MCA, is derived from SCLK. Output data is asserted on the positive going edges of SCLK; OutCLK transitions occur on the negative going edges of SCLK.

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 ouptput stage of the next word block. Otherwise, the **Sync**

Configuration	SCLK Frequency	RCLK Frequency
6 output pairs	MCA×2	MCA/2
4 output pairs	MCA×2	MCA/3
2 output pairs	MCA×2	MCA/6
1 output pairs	MCA×2	MCA/12

Table 10: dependence of SCLK and RCLK on data output configuration.

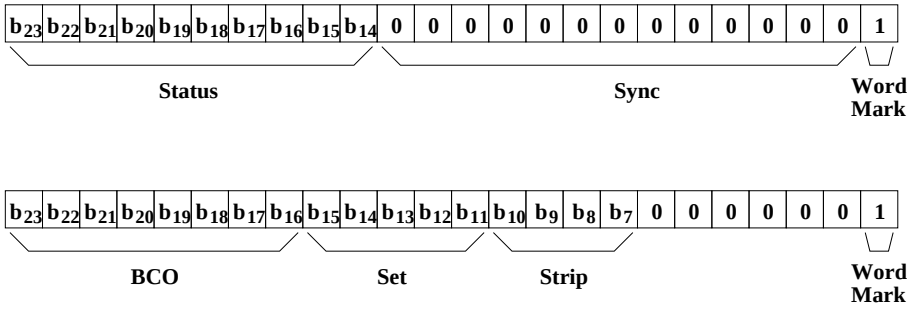

Figure 11: format of Sync Word (top) and Data Word (bottom).

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 set number 15, there is at least on 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 Word will be output at least twice every time that the FSSR readout scans through all 16 sets. As illustrated in Fig. 11, the Next Word Block adds a word mark bit, which is guaranteed to be one, to the data. The Sync Word can be used by the external board supervising the module operation to establish the location of the 24-bit word boundary. Sync Word can be distinguished from Data Word (Core Data) because the Sync Word has 13 zeros in bits 1 to 13. The Word Mark bit and the strip and set number encoding ensure 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 Word are reserved for status and error codes. So far, five of these bits have been assigned, as Table 11 shows. Set and strip numbering codes are displayed in Table 12 and Table 13 respectively. From them one can infer, for instance, that strip 0, which corresponds to the leftmost channel in the chip, and is the first strip of the first set, will have the code 010100101, while strip 127, corresponding

Bit No.	Meaning	Bit No.	Meaning
23	SendData	18	Not Assigned
22	RejectHits	17	Not Assigned
21	Alines-b1	16	Not Assigned
20	Alines-b0	15	Not Assigned
19	AqBCO≠0	14	Not Assigned

Table 11: status bit assignment in Sync Word.

Set No.	Code	Set No.	Code	Set No.	Code	Set No.	Code
1	01010	5	01100	9	10100	13	10010
2	01011	6	01101	10	10101	14	10011
3	01111	7	11101	11	10111	15	11011
4	01110	8	11100	12	10110	16	11010

Table 12: set coding.

Strip No.	Code	Strip No.	Code
1	0101	5	1010
2	0111	6	1011
3	0110	7	1001
4	1110	8	1101

Table 13: strip coding.

to the rightmost channel in the chip and being the last strip of the last set, will have the code 110101101.

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. Data format when more than one output line is active is shown in Fig. 12.

The Steering Logic drives the output data clock and the serial output data off chip. OutCLK is one half the frequency of SCLK and is phased so that its edges fall halfway between the edges of the data lines.

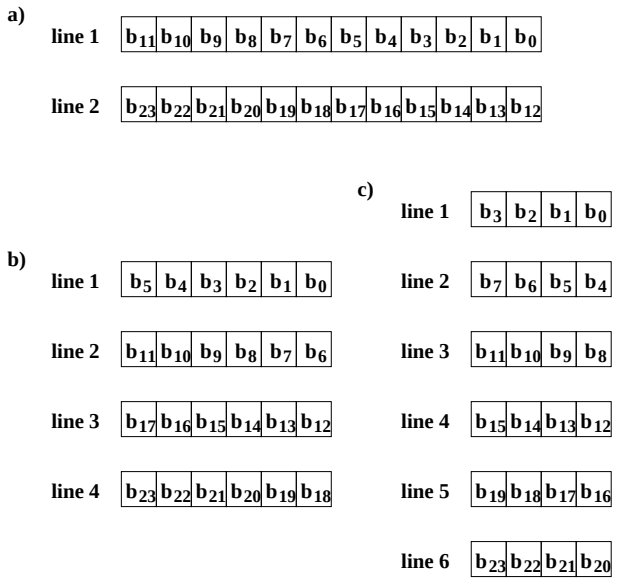

Figure 12: data format when 2 (a), 4 (b) or 6 (c) output lines are active.

6 Resets

The FSSR chip, in the present version, has two hardware resets (**Firefighter Reset** and the **Operation Reset**) and two software reset (**Smart Core Reset** and **Smart Programming Reset**).

SCR - When a Smart Core Reset is received, the BCO counter is reset to zero, all End-Of-Set Logic is cleared and all strip hits are cleared.

SPR - When a Smart Programming Reset is received, all of the registers that hold values input to DACs are reset to their default values (note anyway that in this version of the chip DACs are not used) and the AqBCO register is cleared.

OR - When an Operation Reset is received, the Data Output Interface is 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 to 1 transition of OutCLK.

FFR - When a Firefighter Reset is received, the FSSR chip is reset to a "safe" mode. The effect is the same as a simultaneous Smart Core Reset, Smart Programming Reset and Operation Reset. In addition, the Firefighter Reset resets Alines, Kill, Inject, SendData and RejectHits. This means that the number of active output pairs is reset to one, no strip is killed and

none connected to the test charge injection signal (InjectIn), data output from the Core is inhibited and strip unit cells are set to **Ignore New Hits**.

The various reset actions are summarized in Table 7.

7 Procedures

The following procedure should be followed

- when switching on the FSSR chip:
  1. Firefighter Reset: place the FSSR chip in a "safe" mode
  2. <Write,Alines,b0b1>: configure the Data Output Interface to use the proper number of output lines (if more than one pair is to be used)
  3. <Reset,RejectHits>: enable the Core
  4. <Set,SCR>: Smart Core Reset
  5. <Set,SendData>: enable data readout
- before performing any Kill or Inject operation:
  1. <Set,RejectHits>: disable the Core
  2. Scan in Kill/Inject patterns
  3. <Reset,RejectHits>: enable the Core
- when turning on the chip and starting with a Kill/Inject operation:
  1. Firefighter Reset
  2. <Write,Alines,b0b1>
  3. <Set,RejectHits>
  4. Scan in Kill/Inject patterns
  5. <Reset,RejectHits>
  6. <Set,SCR>
  7. <Set,SendData>

If an FSSR chip loses data output syncronization, then an Operation Reset will recover syncronization without requiring the Core or Programmable Registers to be reset.

If an FSSR chip loses BCO syncronization, then a Smart Core Reset will recover syncronization without requiring the Data Output Interface or Programmable Registers to be reset. As shown in Fig. 13 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.

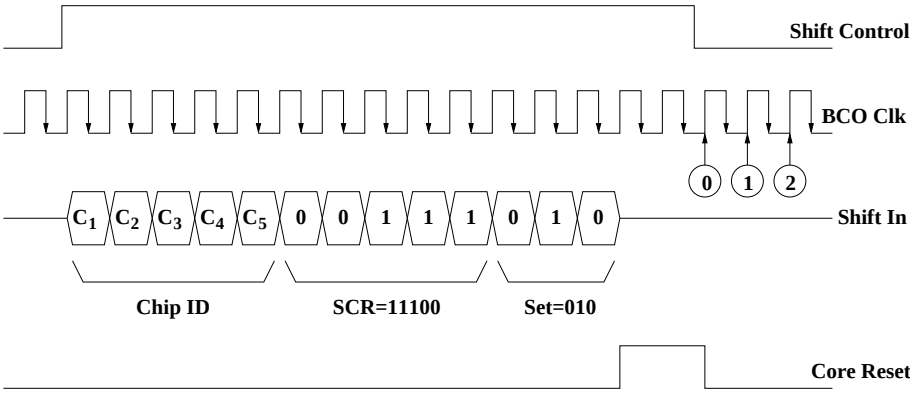

Figure 13: the Core Reset line is asserted on the rising edge of the BCO Clock following the last bit of the <SCR,Set> command. It stays active until just after the first BCO Clock rising edge after Shift Control has been lowered. The next rising edge of the BCO clock will increment the BCO counter from zero to one, so the BCO numbers will be assigned as indicated by the numbers in circles.