PSD@CBM FEE and readout (draft, for internal use)

Finogeev Dmitry, INR RAS

May 8, 2021

Actual version of the document is available at github:
`https://github.com/dfinogee/PSD-readout-manual/raw/main/PSD_readout_manual.pdf`

Contents

Part I
PSD FEE boards

1 ADC board

1.1 ADC clock scheme

2 ADC addon

3 ADC data processing

PSD_data_readout component receive data from all ADCs, process waveform and output data in GBT packets. Schematic of component is presented on fig. 1.

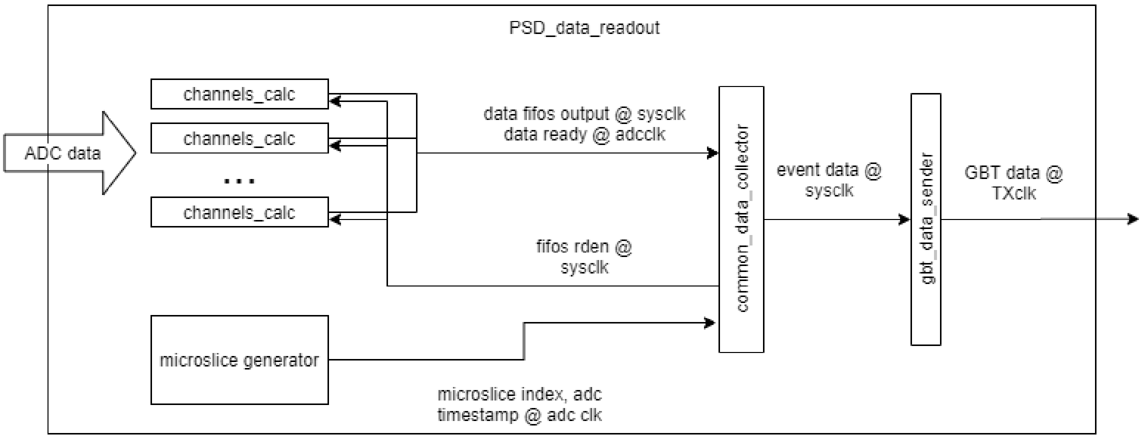

Figure 1: ADC data readout scheme

3.1 Component channels_calc

Channel_calc component scheme is presented on figure 2. ADC data inverted for negative signals, zero level and RMS are calculated and available from slow control.

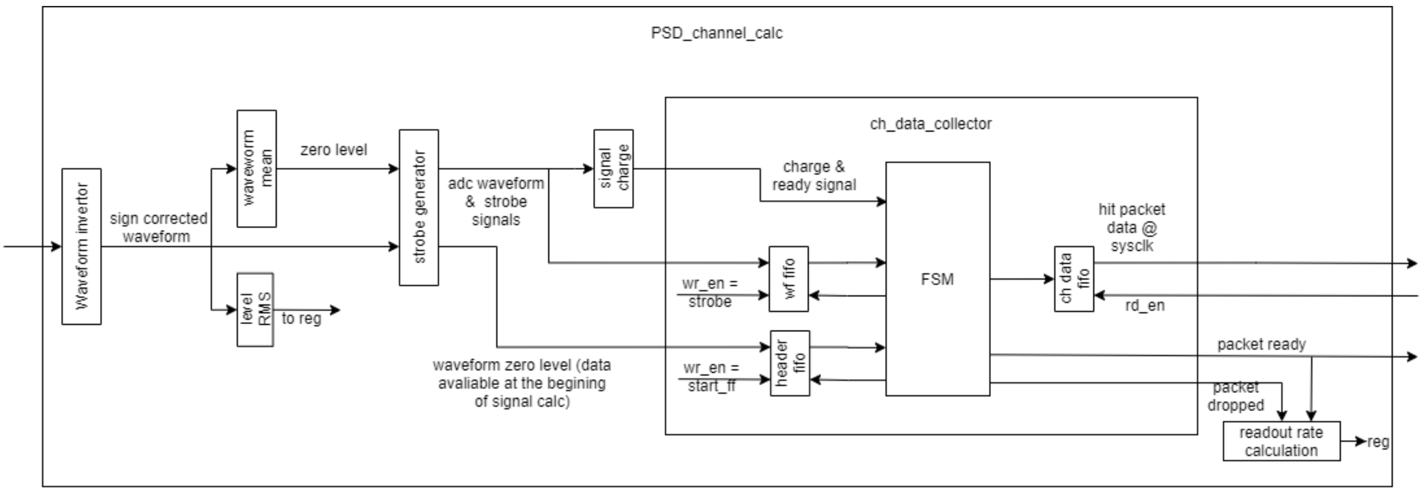

Figure 2: Channel data processing scheme

Strobe_generator component forms waveform gate, start and stop signals by threshold crossing taking waveform length and offset parameters. Waveform data that are available from the start (zero level) are latched while strobe. Signal diagram of the component is presented on figure 3 To reduce the probability of being triggered by a noise event, three neighboring points are compared with threshold. Central point is compared with the threshold value and two side points with half of threshold value.

Waveform offset parameter determine waveform position in gate, if it is 0, first point in waveform strobe is the point above threshold (the third point compared to half of threshold value). Maximum offset value is 13. Latched baseline level is value before point above threshold.

If one channel in common trigger mask parameter cross threshold, common trigger is generated. All channels in common trigger output parameter take waveform similar to they has threshold crossing together.

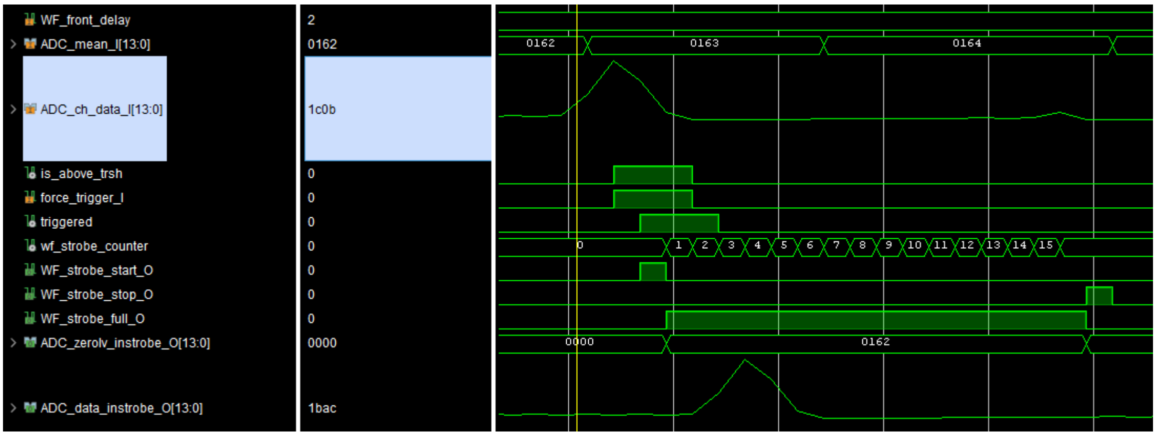

Figure 3: Signal waveform strobe (length 16, offset 3)

Ch_data_collector store waveform point in raw_fifo by strobe signal and start waveform data (zero level) by start signal. When charge ready signal raised, charge and start data from header_fifo stored in data_fifo as hit packed header. This allow to upgrade charge calculation with fitting procedure and change calculation delay. In next cycle waveform points are read from raw_fifo and (if sending wf points parameter is set on) stored as hit data in ch_data_fifo. After hit packet stored, ready signal raised or dropped signal in case fifo was full and hit packet was dropped. Ready and dropped signals are synchronous to threshold crossing and used for event ADC timestamp. Signals diagram of the component is presented on figure 4. The write size of ch_data_fifo should be equal to ceil(calculation_delay / waveform_lenght)*waveform_lenght. The size of ch_header_fifo should be ceil(calculation_delay / waveform_lenght). Write rate for mentioned fifo is equal to read rate. In case data-fifo is full while charge ready signal, hit is dropped and dropped-hits counter increased by 1. Dropped-hits counter is available in channel status and reset after each register reading.

Readout-rate component allow to measure hit rate per channel. Waveform-start signals counted with 16bit counter and 70Hz rate. Each 70 Hz cycle, count is stored in 128 shift register. Rate-mean register store the summ of values stored in shift register. Two modes: low-rate and normal are available for rate reading. In normal mode for 16 bit status register available rate-mean[22 downto 7] and result is rate/70Hz. In low-rate mode (channel-low-rate-count bit) rate-mean[15 downto 0] available for status register and result is rate/70*128Hz.

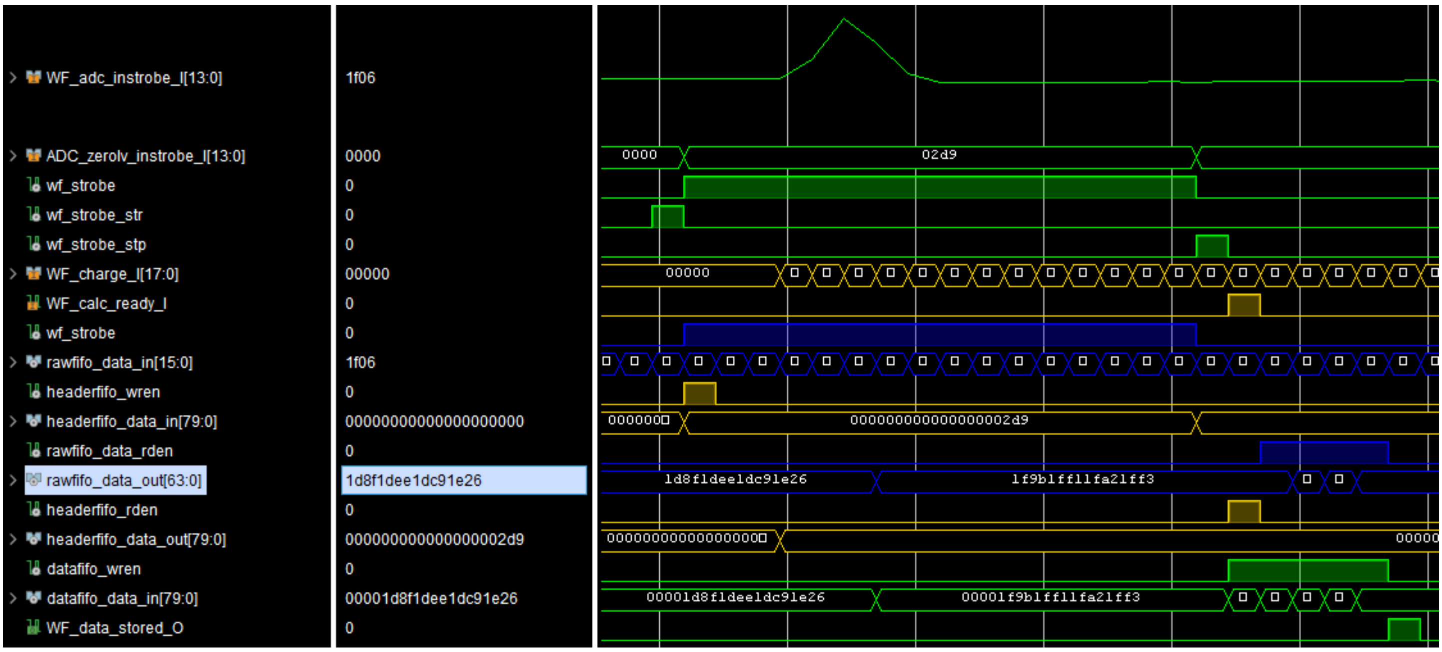

Figure 4: Channel data collecting signals

Signals could be processed one after another without dead time. If next adc point after waveform gate is higher than threshold, new signal gate is formed. Signal time is next adc cycle after first gate, not is real time of second waveform threshold crossing. Signal diagram for such case is presented on figure 5.

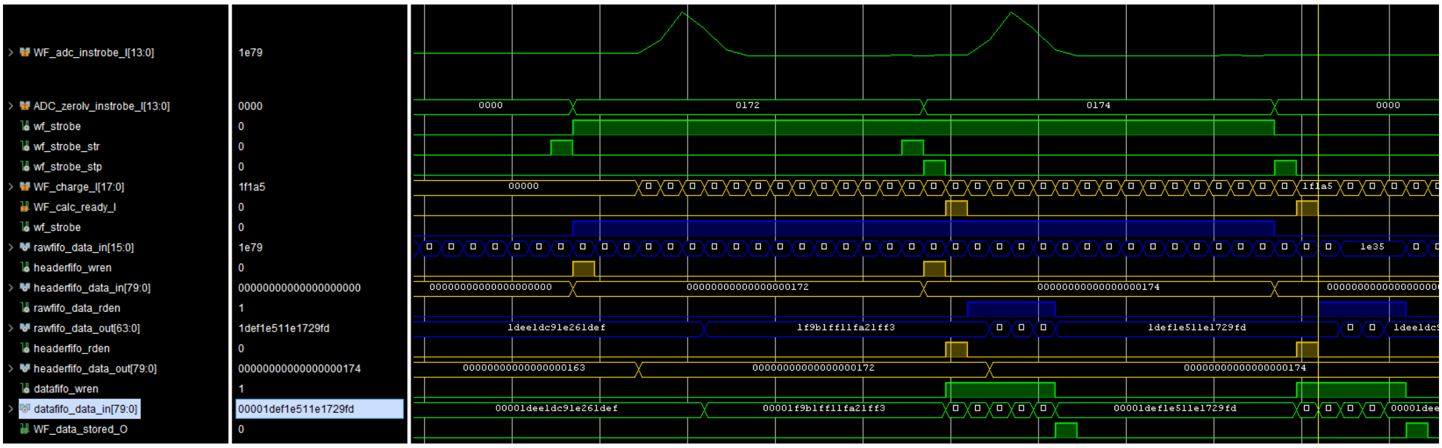

Figure 5: Channel data collecting signals

3.2 Component common_data_collector

Each channel generate single strobe with fixed latency to threshold crossing indicating waveform measurement. 32 bit strobe word is stored to data_wf_calc_fifo with mc index and ADC timestamp. FSM read stored strobes and collect data from fired channels storing outputs to common_data_fifo, each event header word with timing and data size info stored in common_header_fifo. Schematic represented on figure 3.

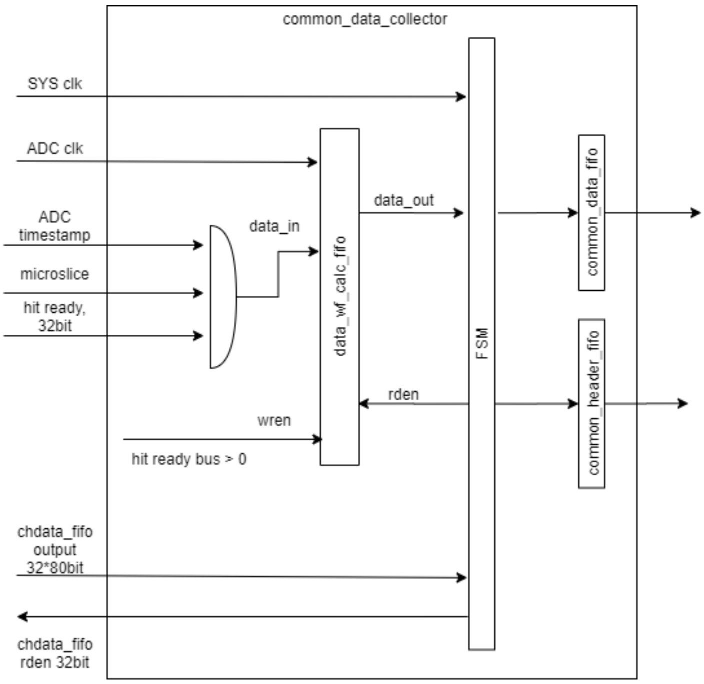

Figure 6: Data collecting scheme from all channels fifos

FSM is switched from wait to start state when data_wf_calc_fifo_isempty became '0' and fifo output is latched. Priority encoder show next fired channel from strobe and data collected from fired channel to common_data_fifo with hit_packet_iterator. Input to priory encoder is shifted to bit after fired channel when iterator reach last fired channel. Priority encoder could be equal or less than 32 bit. Simulation outputs presented on figure 4.

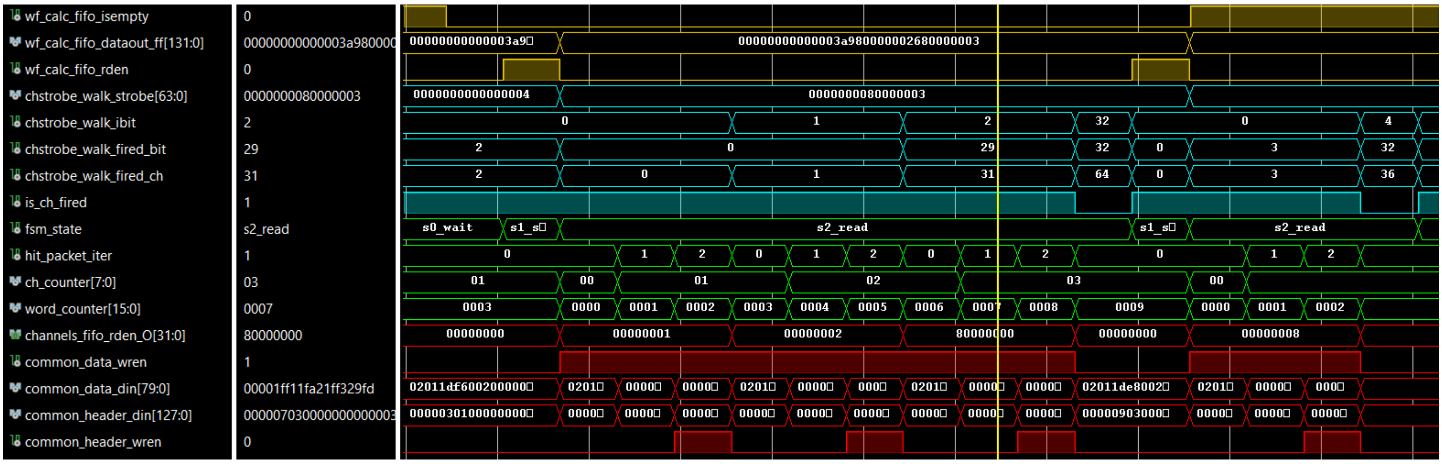

Figure 7: Data collecting signal from all channels fifos

Collecting data from all channels takes two additional FSM cycle. Mean hit rate per channel in case all channels fired is SYSCKL / total channels + 2 cycle / packet length. Test beam: 80MHz / 12 / 5 = 1.3MHz. Final setup: 120 (240) / 32 / 1 = 3.5 (7) MHz.

3.3 Component GBT_data_sender

Data stored in common_data_fifo in component common_data_collector are read by system clock with writing rate. Event and microslice headers are formed by data from common_header_fifo. Built GBT data packets are stored in gbt_data_fifo and read by GBT TX clock. Signal diagram is presented on figure 8.

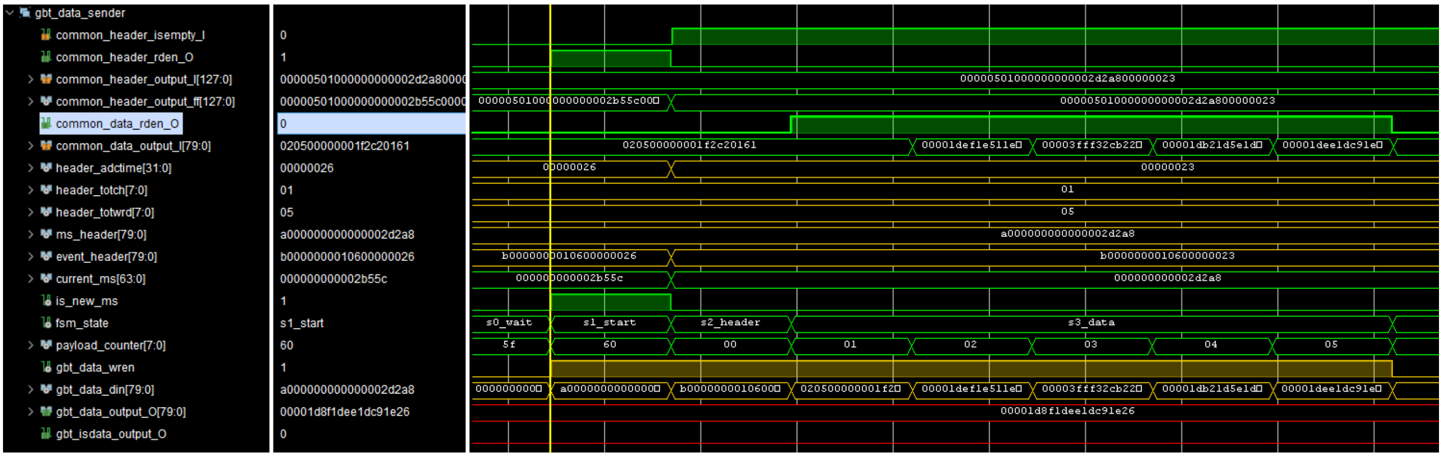

Figure 8: Channel data collecting signals

Data rate limit is 80bit X 40MHz = 0.4 GB/s(GBT). Hit rate limit per channel (without microslice word) is 40MHz / 33 (packet length) = 1,2 MHz in case all channels are fired. The rate could be increased to 2.4 MHz hits per channel in case all 32 channels are fired. If one hit data will be less than 40bit event packet will contain 17 GBT words.

GBT packet format is presented on tables: 1, 2, 3

word type	79 .. 76	75 .. 72	71 .. 64	63 .. 48	47 .. 40	39 .. 32	31 .. 16	15 .. 0
ms header	0xA	0x0		ms index				
event header	0xB	ADC idx**	0x0		n fired channels	words in packet *	adc time	
hit header	hit header (tab. 1)							
hit data	hit data (tab. 2)							
hit data	hit data (tab. 2)							
hit data	hit data (tab. 2)							
hit data	hit data (tab. 2)							
	...							
event header	0xB	ADC idx**	0x0		n fired channels	words in packet *	adc time	
	...							

Table 1: GBT data format. [* number of GBT words in event packet: event header + all hit packets] [** ADC board index]

word	79 .. 72	71 .. 64	63 .. 36	35 .. 16	15 .. 0
1	channel	words in packet *	0x0	signal charge	waveform zero level

Table 2: hit packet header. [* total GBT words in hit packet: header + data words]

word	79 .. 64	63 .. 48	47 .. 32	31 .. 16	15 .. 0
1	0x0	waveform point n	waveform point n+1	waveform point n+2	waveform point n+3

Table 3: hit packet data word.

Reserved first 8 bits in GBT data flow:

- 0x0:0x20 - hit header ch number
- 0x3 - hit data word (DOTO)
- 0xA - microslice header
- 0xB - event header
- 0xC - CRI FLIM iface mcs delimiter word
- 0xE - status packet word
- 0xF - control packet word

4 ADC control

4.1 ADC control units

Status and Control of ADC are 64 arrays each of 32 bit words. ADC control system include 4 firmware units: gbt-control-sender, gbt-control-reader, gbt-status-sender, gbt-status-reader. ADC control and monitoring strategy is describe in sec. **??**

gbt-control-sender is placed on CRI side and send control packet (129 X 16bit) via gbt to ADC. Packet could be send at any time and is not in conflict with microslice flow to ADC. gbt-control-reader receive control packet, and update registers array with raising "updated" strobe.

word	value
0	0xABBA
1	control(0)(15 .. 0)
2	control(0)(31 .. 16)
3	control(1)(15 .. 0)
4	control(1)(31 .. 16)
....	
127	control(63)(15 .. 0)
128	control(63)(31 .. 16)

Table 4: Control packet to ADC.

gbt-status-sender send status or control registers from ADC (packet 32 X 80bit). Status/control packet is prioritized to data flow, and gbt-data-fifo is not read while transaction. Status and control words starts with 0xE and 0xF accordingly to be distinguished from data flow.

Sending of control or status packets could be initiated by CRI side with 0xABBB and 0xABBC codes in MSB of RX data.

bits	79 .. 76	75 .. 64	63 .. 32	31 .. 0
word	code	addr	reg1	reg0
0	E	0	status(1)	status(0)
1	E	2	status(3)	status(2)
		....		
31	E	30	status(31)	status(30)

Table 5: Status packet from ADC.

bits	79 .. 76	75 .. 64	63 .. 32	31 .. 0
word	code	addr	reg1	reg0
0	F	0	control(1)	control(0)
1	F	2	control(3)	control(2)
		....		
31	F	30	control(31)	control(30)

Table 6: Control packet from ADC.

gbt-status-reader read each gbt word stars with 0xE or 0xF and update control or status registers. Two counters indicate the time passed from last update. Read back control register is compared with actual on CRI side.

4.2 Addon I2C control

4.3 ADC Control registers

addr	31 .. 30	29 .. 28	27 .. 24	23 .. 20	19 .. 16	15 .. 14	13 .. 12	11 .. 8	7 .. 4	3 .. 0
0	0x0	threshold ch1				0x0	threshold ch0			
1	0x0	threshold ch3				0x0	threshold ch2			
2	0x0	threshold ch5				0x0	threshold ch4			
3	0x0	threshold ch7				0x0	threshold ch6			
4	0x0	threshold ch9				0x0	threshold ch8			
5	0x0	threshold ch11				0x0	threshold ch10			
6	0x0	threshold ch13				0x0	threshold ch12			
7	0x0	threshold ch15				0x0	threshold ch14			
8	0x0	threshold ch17				0x0	threshold ch16			
9	0x0	threshold ch19				0x0	threshold ch18			
10	0x0	threshold ch21				0x0	threshold ch20			
11	0x0	threshold ch23				0x0	threshold ch22			
12	0x0	threshold ch25				0x0	threshold ch24			
13	0x0	threshold ch27				0x0	threshold ch26			
14	0x0	threshold ch29				0x0	threshold ch28			
15	0x0	threshold ch31				0x0	threshold ch30			

Table 7: ADC channels threshold control.

addr	31 .. 28	27 .. 24	23 .. 20	19 .. 16	15 .. 12	11 .. 8	7 .. 4	3 .. 0
16	0x0		status ch sel		waveform length 0..3 [(reg+1)*4]	strobe offset 0..12	control bits	
17	negative channel mask ibit = ich							
18	I2C HV bus							
19	microslice gen counter@25ns							
20	microslice period							
21	common trigger OR mask							
22	common trigger output							
23	trigger pulser rate [count @ ADC clock] (0x0 = off)							
24	status send rate (0x0 = off)				control send rate (0x0 = off)			
25	common trigger AND mask							

Table 8: ADC readout control.

bit	description
0	send waveform
1	ms gen standalone
2	readout fsm reset
3	errors reset
4	channel low rate count
5	reset channels drop counter

Table 9: Control bits

addr	31 .. 25	24 .. 24	23 .. 23	22 .. 16	15 .. 8	7 .. 0
18	0x0	start	WR	i2c dev addr	mem addr	data

Table 10: HV control via I2C.

4.4 ADC Status registers

Status registers map is presented on table 11.

addr	31 .. 30	29 .. 28	27 .. 24	23 .. 20	19 .. 16	15 .. 14	13 .. 12	11 .. 8	7 .. 4	3 .. 0
0	microslice index 31 .. 0									
1	microslice index 63 .. 32									
2	ADC time									
3	RX wrclk err cnt					RX err frclk cnt				
4	RX err detect cnt					I2C HV bus				
5	0x0								temp	
6	sel. channel baseline rms					sel. channel baseline				
7	sel. channel dropped hits					sel. channel hit rate				
8	GBT event dropped					GBT fifo count				

Table 11: ADC channels threshold control.

addr	15 .. 10	9 .. 9	8 .. 8	7 .. 0
4	0x0	error ack	busy	DATA

Table 12: HV status via I2C.

Status registers comments:

- RX err detect cnt - counter@RXclk of RX error detected bit.
- RX err frclk cnt - counter@RXclk of state when frame clock is not ready.
- RX wrclk err cnt - counter@RXclk of state when word clock is not ready.

Part II
CRI-PSD firmware

5 ADC - CRI operation

ADC - CRI communication scheme is presented on fig. 9.

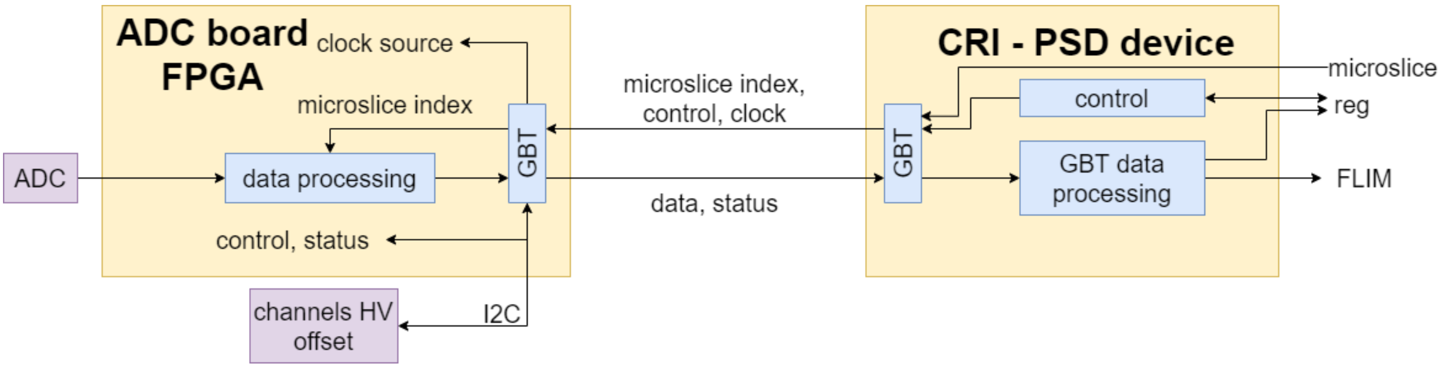

Figure 9: ADC - CRI GBT connection

GBT-FPGA on ADC board side recover RX clock (see 1.1), microslice (64 bit@40MHz each GBT word) and control packets (see 4.1). Microslice index transported to ADC clock domain (synchronous to RX clock) and each ADC cycle in microslice numerated with adc time index. Measured hits labeled with microslice + adc time indexes and sent to CRI via GBT TX. Control and monitor of data taking, MPPC HV adjustment and MPPC board temperature control is available via GBT control packets.

TODO: update with 2 FPGAs

5.1 PSD CRI operation

5.2 CRI - ADC control strategy

Implementation details are described in sec. 4.1

adc-control-sender unit placed in CRI receive mapped adc control registers (64X32bit) from AGWB "psd-adc". By software command all registers could be sent to ADC. After each transaction of control packet ADC send back control register. adc-status-reader unit read control and status packet from ADC on CRI side. Status packets received from ADC updates status on AGWB "psd-adc". Received control packets compared with actual control registers from "psd-adc" and "match" signal indicate correctness of ADC configuration. ADC reset cycle (FSM or errors) must be done in 4 writes (here and below several fields writes counted as single command):

1 write 0x1 to reset register

2 trigger sending of control packet

3 write 0x0 to reset register

4 trigger sending of control packet

Control and status packets could be sent periodically, that allow only read AGWB status registers for ADC monitoring. Also control and status packets could be requested by CRI side, in that case monitoring can be done with one write command (trigger status request) and one read command of status.

To save registers space, channel status (base level, RMS, hits dropped, hits rate, 64 bits in total) presents in status register map only for one channel. Number of monitored channel is controlled with field "mon-ch-sel". After each time the field is changed on ADC side, status packet sent to CRI automatically. That allow to monitor channel status in 3 command:

1 write numger of monitored channel to register

2 trigger sending of control packet

3 read status fields of channel status

Also status packets sent to CRI automatically after each I2C operation that allow to configure addon register in 5 commands:

1 write data for I2C transation (start = 0)

2 trigger sending of control packet

3 write start to 1

4 trigger sending of control packet

5 read I2C status

TODO: In future some of operation could be implemented on firmware level:

- Reset cycle could be automated on ADC and CRI sides for single write AGWB command
- ADC control packet could be sent automatically after changing of registers (with timeout) and ADC configuration mismatch while writing new control values could be processed on CRI side. That will allow just write new configuration on software level.
- Errors could be reset automatically after sending from ADC and stored and alarmed on CRI side
- Channel status map could be received automatically with low rate update, representing full channels status table in AGWB
- I2C operation could be automated with full addon control and status registers map in AGWB

  Also, to do not reduce data rate with status registers, packets could be sent via 4 bits of slow-control bus.

6 PSD CRI data processing

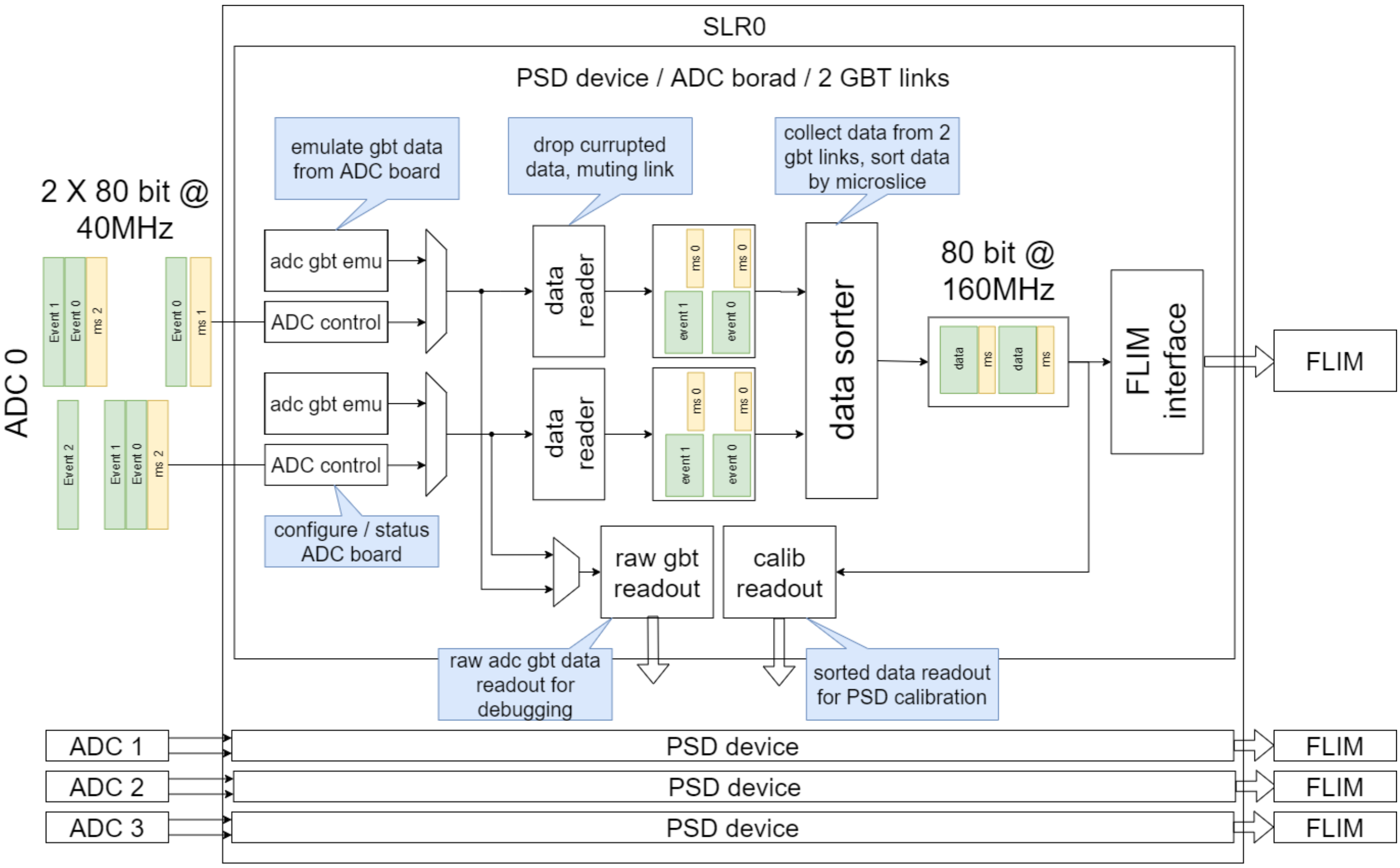

Figure 10: PSD data processing in CRI

Figure 10 present PSD data processing flow. Each ADC board is connected to psd-device with 2 GBT links. There are 4 psd-devices per SLR, 16 GBT links, 8 psd-devices, 2 SLRs in total. Each gbt link is connected to ADC-control, adc-gbt-emu and data-reader units. adc-gbt-emu mute gbt link and emulate adc gbt packets for readout tests purposes (see sec. 6.1).

adc-control translate adc control and status registers to AGWB (see sec. 5.2). data-reader unit read adc data packets, drop corrupted data and provide each packet with microslice header in separate fifo (see sec. 6.2). Also adc-reader can mute any gbt link. data-sorter unit (see sec. 6.3) reads data and header buffers from data-reader and sort data by microslice intervals. data-sorter throttle data flow in case FLIM interface is not ready to data transport. While correct operation only data-sorter should throttle data, all other units are able transport data at full 2 gbt links load. Two slow readout units are available for calibration and test: raw-gbt-readout (see sec. 6.5) and data-readout (see sec. 6.4). raw-gbt-readout allow to take data as they received from adc board including control packets. It was implemented as one of the first units of PSD-CRI, now it is rudimentary and can be used for debug. data-readout transport sorted and throttled data for tests and calibration purposes.

TODO: data throttling description

6.1 ADC GBT emulator

adc-gbt-emu generates adc data packets with variable event and hit load in frequency range 1/107Hz ... 20MHz. Hit header contains microslice index and hit data contains continuous hit counter. adc-gbt-emu is inserted before data-reader and mute gbt link if it is on.

adc-gbt-emu control:

- turn_on - mute input gbt link and output generated data if is on.
- adc_id - adc board index placed in event header
- event_rate is number of GBT clock cycles between packets (from start to start of packet). If previous packet was not sent, and is time to generate new one, new one is skipped.
- event_len - number of hits per event 1 ... 255. Emulate fired channels.
- hit_len - number of hit words, including hit header 1 ... 255. Emulate waveform data

Emulator FSM is based on three counters, simulation signals diagram is presented on figure ??; generated data format is presented on figure ??.

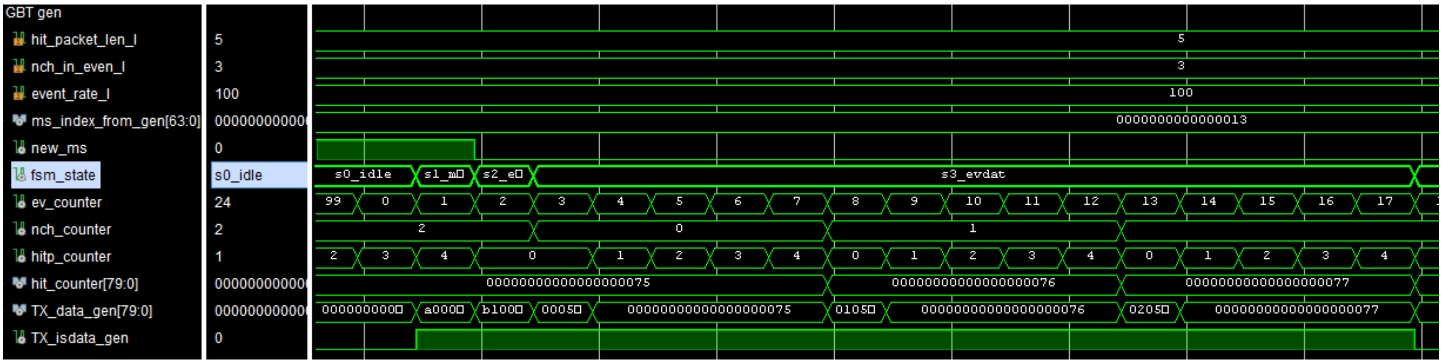

Figure 11: ADC GBT emulator signals

word type	79 .. 76	75 .. 72	71 .. 64	63 .. 48	47 .. 40	39 .. 32	31 .. 16	15 .. 0
ms header	0xA	0x0		ms index				
event header	0xB	ADC idx**	0x0		n hits	packet len *	0x0	
hit header	hit number		words in hit packet ***	ms index				
hit data	hit counter [79 ..0]							
hit data	hit counter [79 ..0]							
hit data	hit counter [79 ..0]							
hit data	hit counter [79 ..0]							
	...							
event header	0xB	ADC idx**	0x0		n hits	packet len *	0x0	
	...							

Table 13: GBT data format. [* number of GBT words in event packet: event header + all hit packets] [** ADC board index] [*** total words in hit packet, including hit header]

6.2 ADC GBT reader

ADC GBT reader reads GBT packets from one GBT link and store its to fifo event_fifo. With last packet data word header word pushed to separate fifo header_ fifo with packet length and microslice index. Event packet skipped when one of fifos is full. After reset fsm starts wait microslice header. Packets reads according to size in header and fsm wait next packet or microslice header. If next word after packet is neither ms or packet header, fsm starts wait ms header. Data drop info state is not implemented yet. Signal diagram is presented on figure 14.

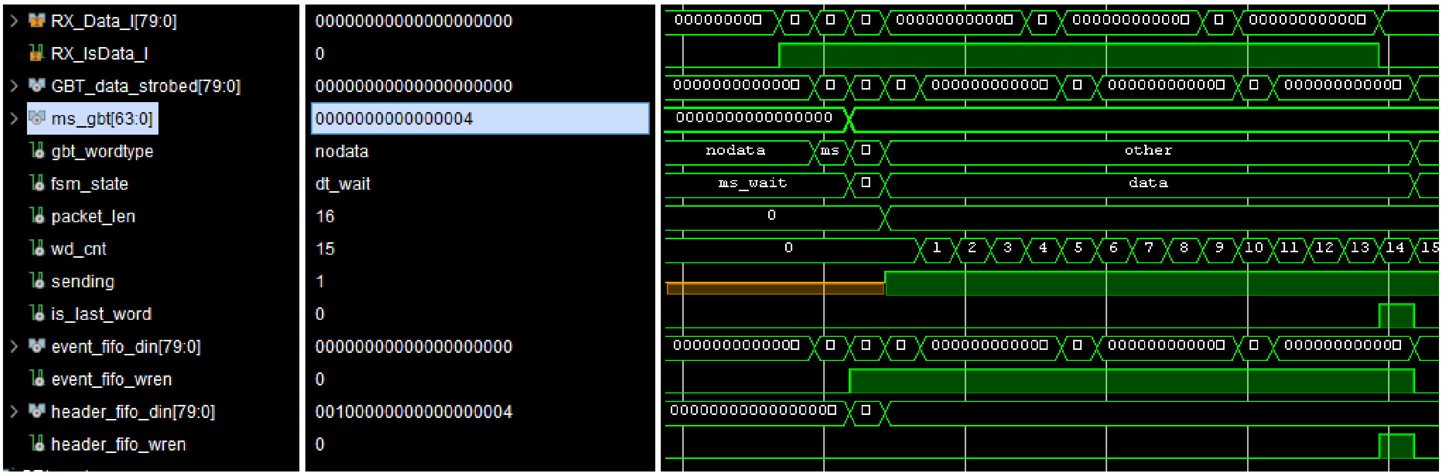

Figure 12: ADC GBT packets reader

6.3 GBT Data Sorter component

Components header_fifo and event_fifo from adc-gbt-readers for all gbt links are connected to gbt-data-sorter component. Each new microslice value collected in ms-fifo. FSM switch thought all gbt links and read all one by one links with microslice less or equal to current microslice. Data for links with equal microslice to current-ms output from the sorter, for links with less microslice data is dropped. When all links have ms higher than current ms or are empty means that all data for current microslie are read. Such condition starts counter to wait data from all links. Then counter reach value 127, FSM swithced to next-ms state. Next microslice read from fifo and header with new ms value sent to output stream. Signal diagram presented on figure 13 Output data represent combined GBT packets from all GBT link. All events from GBT links for one microslice follows one after another. Data for different microslices divided by microslice header with format 0xDAF0 + microslice (64bit).

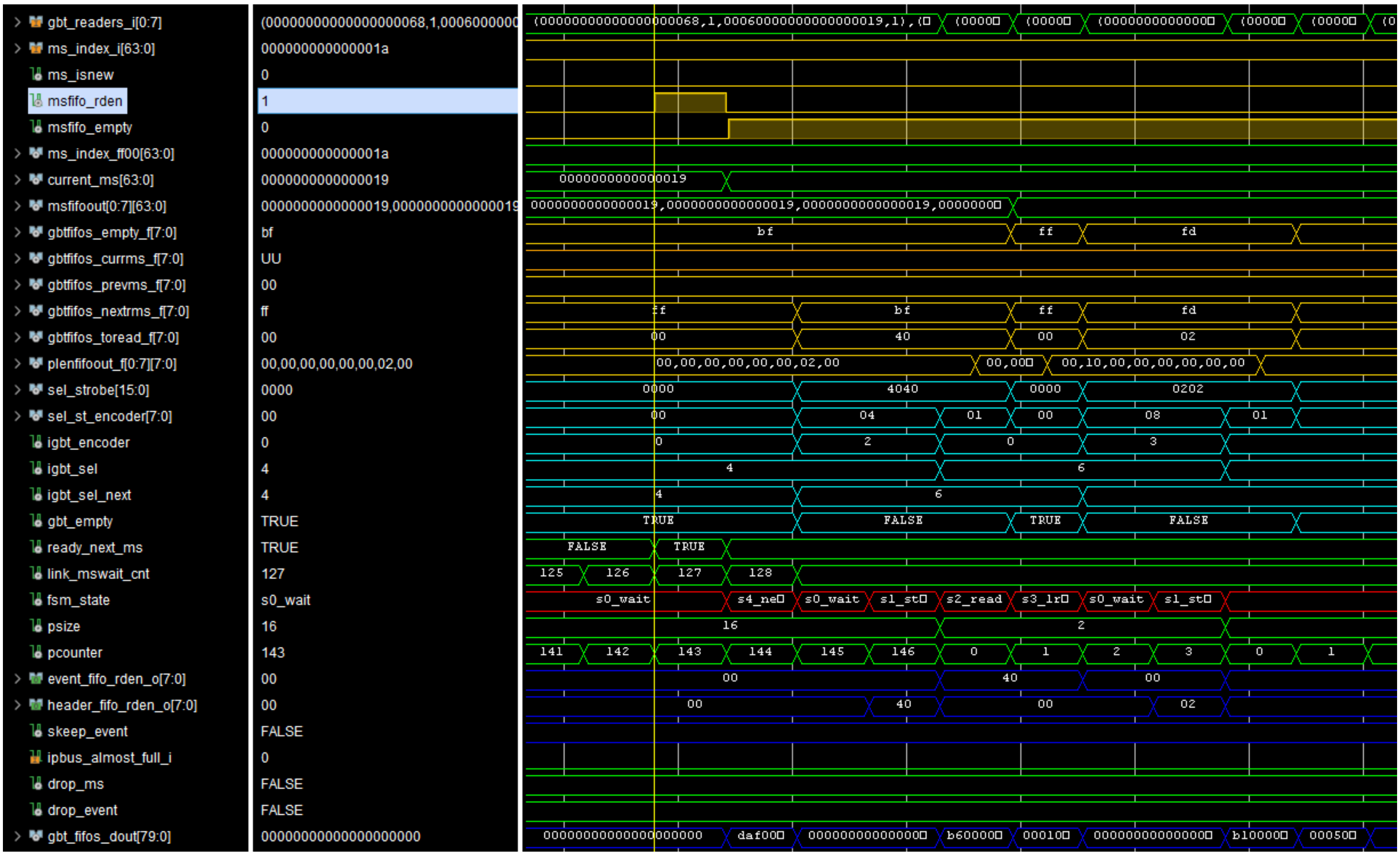

Figure 13: gbt-data-sorter signals diagram: mew ms read and event sent from one link

6.4 Calibration readout

IPbus-face-component read data stream from gbt-data-sorter and resize data to width 32 bit. Data stream from gbt-data-sorter stored in fifo-ipbus-face with 80bit write width and 160 read width. Output 160bit word divided in 5 32bit words. Each IPbus read cycle counter 0..4 increased by one, fifo-ipbus-face readed when counter equal 4 and ipbus-read signal is up. While reading empty fifo-ipbus-face all bits are '1'. Signals diagram is presented on figure 14.

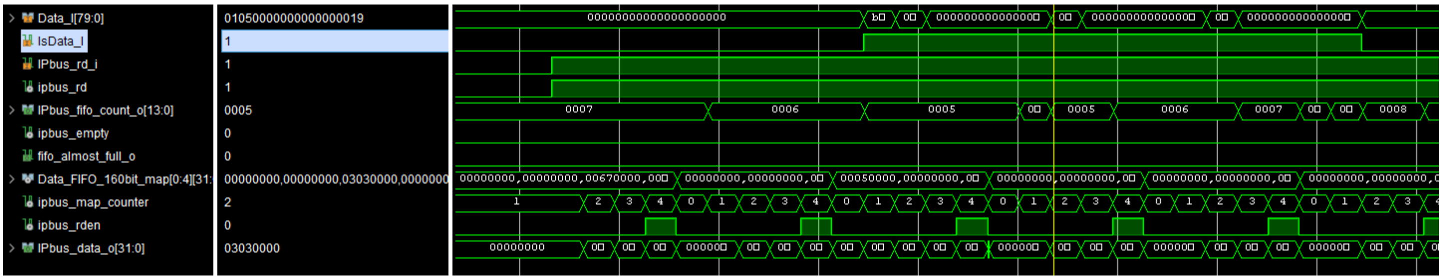

Figure 14: IPbus-face signals diagram

6.5 Raw GBT readout

6.6 PSD FLIM interface

6.7 PSD CRI FIFOs usage

Part III
PSD evaluation board

6.8 EvB control reg

range	description
0 .. 63	EvB control
64 .. 127	ADC control
128 .. 191	EvB status
192 .. 255	ADC status
256	EvB GBT readout
257	EvB readout fifo count

Table 14: EvB registers mapping

addr	31 .. 28	27 .. 24	23 .. 20	19 .. 16	15 .. 12	11 .. 8	7 .. 4	3 .. 0
0	0x0						control word	
1	microslice gen counter@25ns							
2	microslice period							

Table 15: Evaluation board control registers.

bit	description
0	data processing reset
1	error reset

Table 16: Control word bits

addr	31 .. 28	27 .. 24	23 .. 20	19 .. 16	15 .. 12	11 .. 8	7 .. 4	3 .. 0
0	0x0			control status	GBT status			
1	sorter ms dropped				sorter hit dropped			
2	gbt reader link 1 ms dropped				gbt reader link 0 ms dropped			
3	status age				control age			

Table 17: Evaluation board status registers.

bit	description
0	MGT phalin cpll lock
1	RX word clock ready
2	RX frame clock ready
3	MGT link ready
4	TX reset done
5	TX FSM reset done
6	RX ready
7	RX error detected
8	RX error latched

Table 18: GBT status bits

bit	addr	description
0	16	control readback correct

Table 19: control status bits