

# PSD@CBM firmware description (draft, for internal use)

Finogeev Dmitry, INR RAS

January 27, 2021

Actual version of the document is available at github:  
[https://github.com/dfinogee/PSD-readout-manual/raw/main/PSD\\_readout\\_manual.pdf](https://github.com/dfinogee/PSD-readout-manual/raw/main/PSD_readout_manual.pdf)

## Contents

<b>1</b>	<b>ADC data processing</b>	<b>2</b>
1.1	Component channels_calc . . . . .	2
1.2	Component common_data_collector . . . . .	4
1.3	Component GBT_data_sender . . . . .	5
<b>2</b>	<b>ADC control</b>	<b>6</b>
2.1	Control registers . . . . .	6
2.2	Status registers . . . . .	7
<b>3</b>	<b>CRI modules</b>	<b>8</b>
3.1	PSD CRI data sorting . . . . .	8
3.2	ADC GBT emulator . . . . .	8
3.3	ADC GBT reader . . . . .	9

# 1 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.

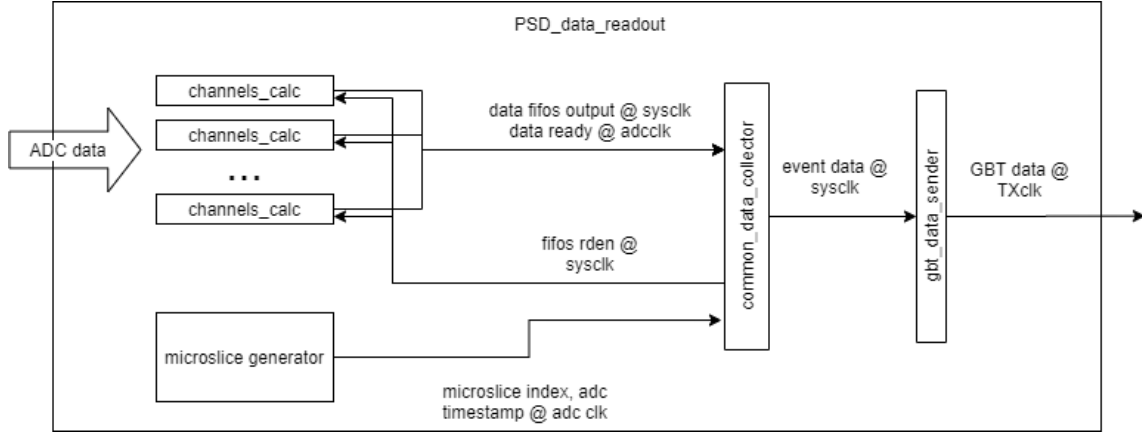


Figure 1: ADC data readout scheme

## 1.1 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.

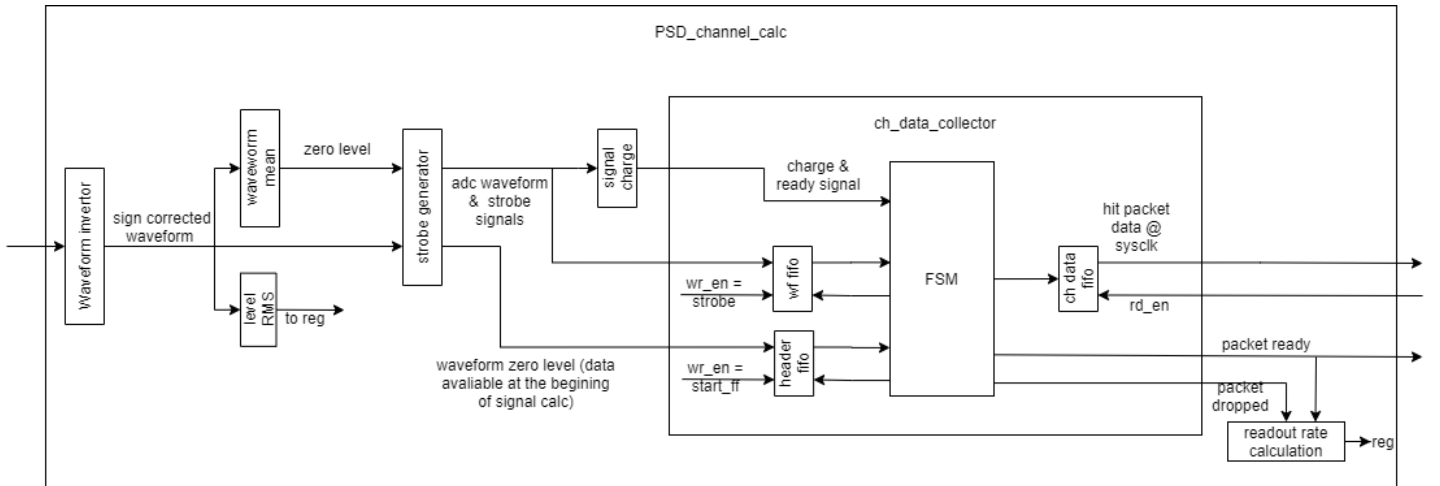


Figure 2: Channel data processing scheme

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.

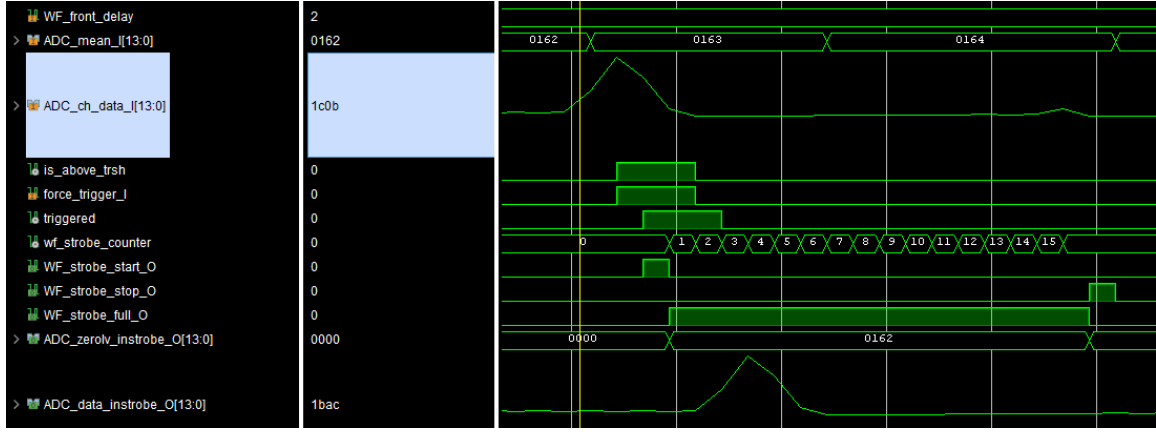


Figure 3: Signal waveform strobe (length 16, offset 3)

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.

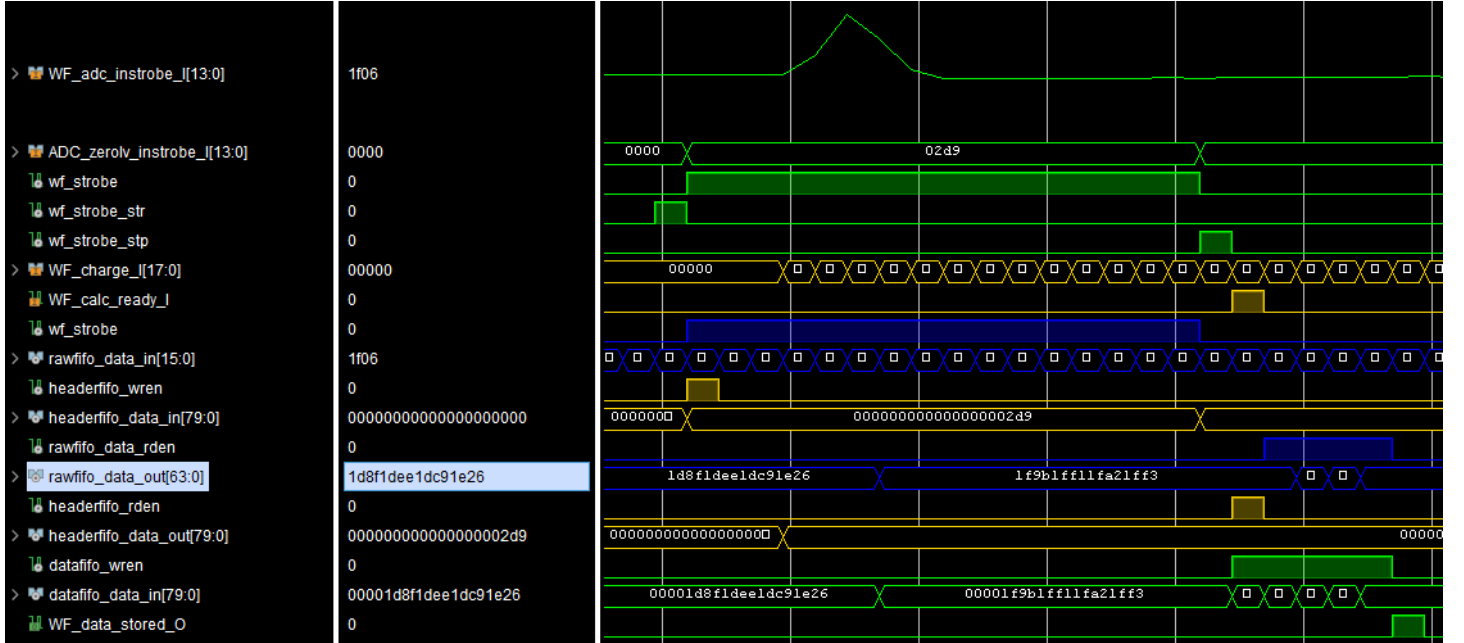


Figure 4: Channel data collecting signals

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.

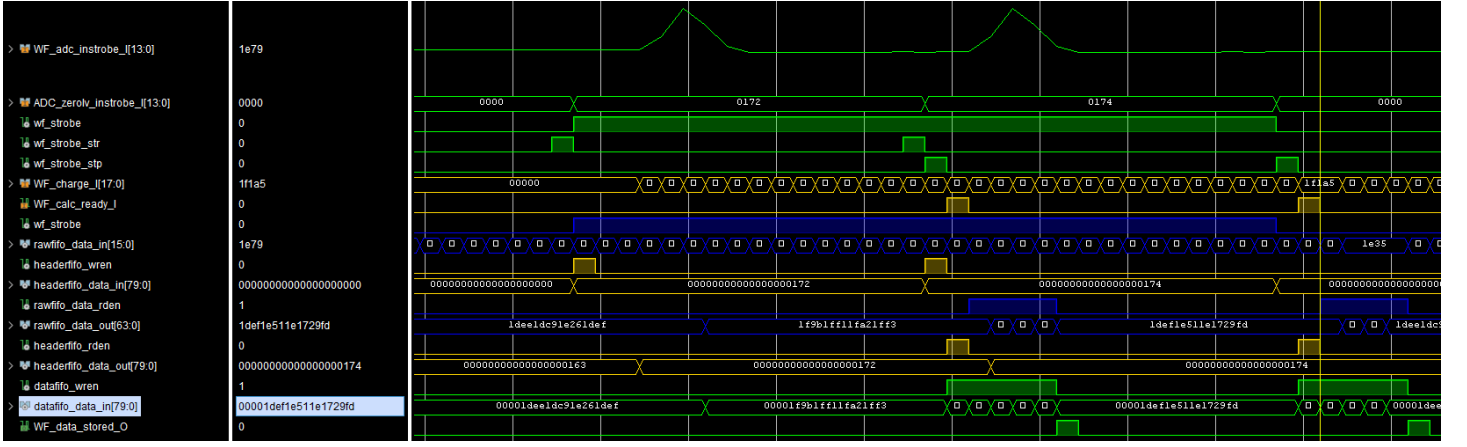


Figure 5: Channel data collecting signals

## 1.2 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.

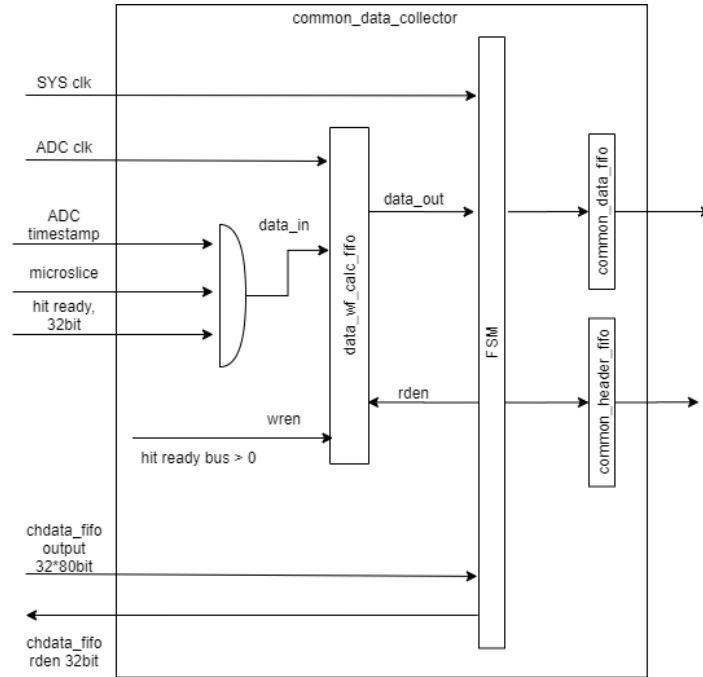


Figure 6: Data collecting scheme from all channels fifos

FSM is switched from wait to start state when data\_wf\_calc\_fifo\_iseempty 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.

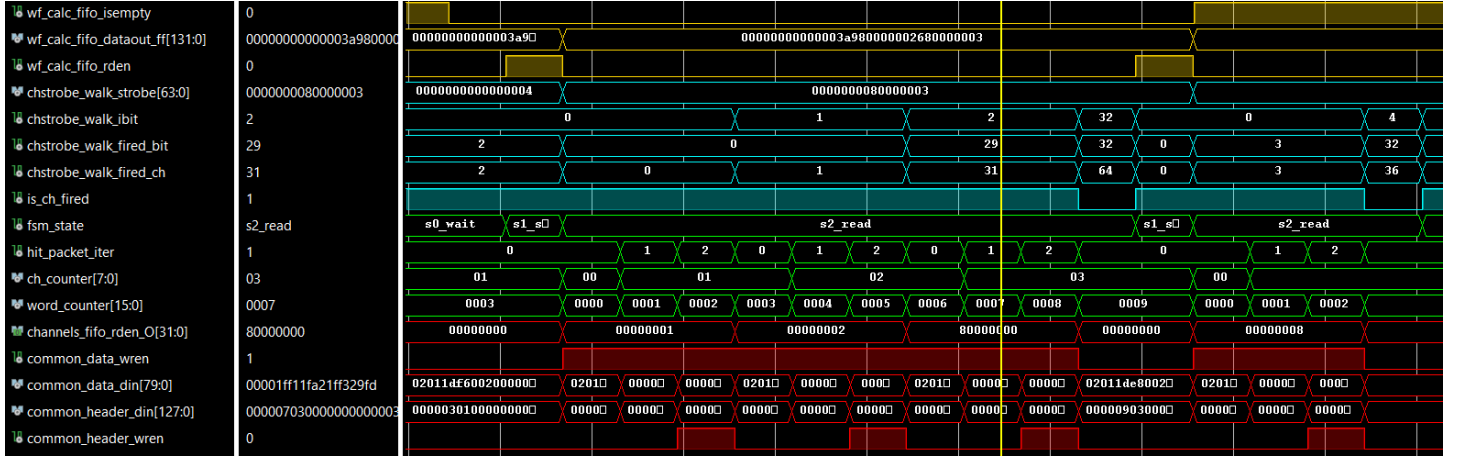


Figure 7: Data collecting signal from all channels fifos

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

### 1.3 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.

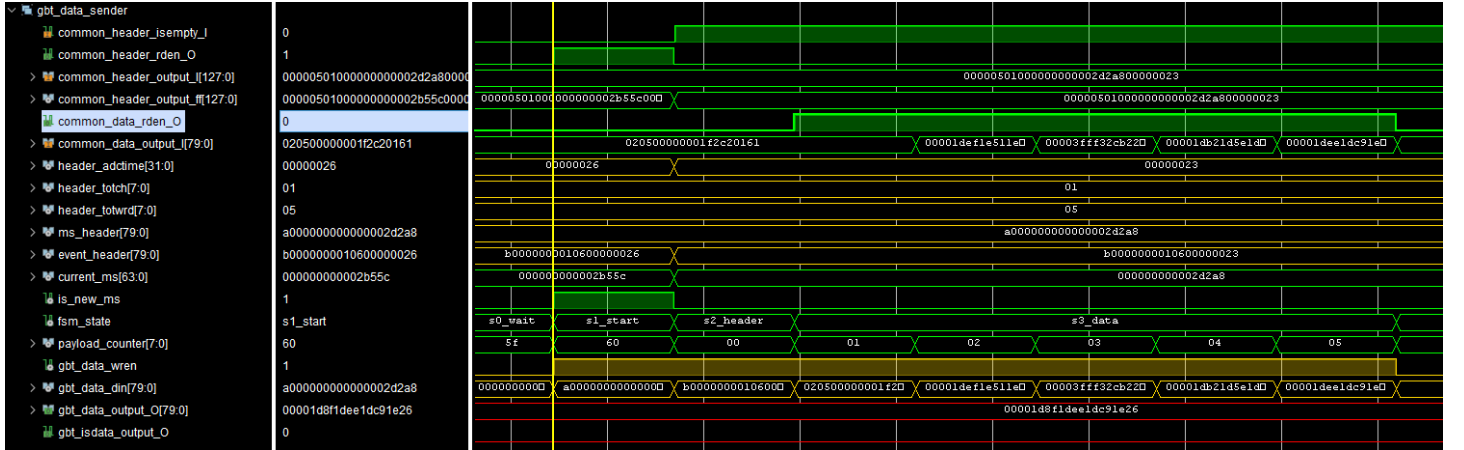


Figure 8: Channel data collecting signals

Data rate limit is  $80\text{bit} \times 40\text{MHz} = 0.4 \text{ GB/s(GBT)}$ . Hit rate limit per channel (without microslice word) is  $40\text{MHz} / 33 (\text{packet length}) = 1.2 \text{ MHz}$  in case all channels are fired. The rate could be increased to  $2.4 \text{ MHz}$  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.

## 2 ADC control

### 2.1 Control registers

To avoid configuration corruption while GBT link fail, register 31 is reserved for lock key word. Control registers are available for writing if register 31 is 0xafafafaf. Register 31 is always open for writing.

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 4: 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 mask *							
22	common trigger output **							

Table 5: ADC readout control. [\* channels set generates common trigger] [\*\* triggered channels set]

bit	description
0	send waveform
1	ms gen standalone
2	readout fsm reset
3	errors reset

Table 6: Control bits

addr	31 .. 18	17 .. 17	16 .. 16	15 .. 8	7 .. 7	6 .. 0
18	0x0	WR	ENA	DATA	0x0	ADDR

Table 7: HV control via I2C.

## 2.2 Status registers

Status registers map is presented on table 8.

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				

Table 8: ADC channels threshold control.

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

Table 9: 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.

### 3 CRI modules

#### 3.1 PSD CRI data sorting

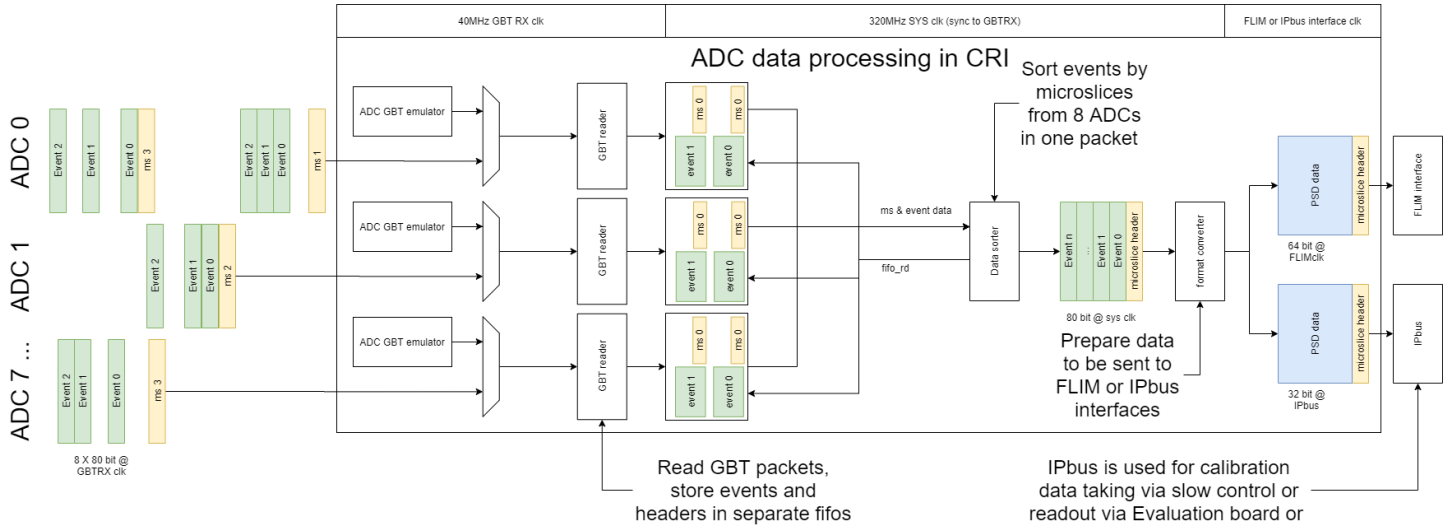


Figure 9: PSD data processing in CRI

#### 3.2 ADC GBT emulator

ADC GBT emulator generate GBT ADC packets filling hit packages with continuous hit counter. Parameters are:

- ms\_index - current microslice index @GBTclk to generate ms headers.
- event\_rate is number of GBT clock cycles between packets (from start to start). If previous packet was not sent, and is time to generate new one, new one skipped.
- nch\_in\_even - number of hits per event 1 ... 32. Emulate fired channels.
- hit\_packet\_len - number of hit packet words, including hit header 1 ... 5.

Emulator FSM is based on three counters, signals diagram is presented on figure 10; generated data format is presented on figure 8.

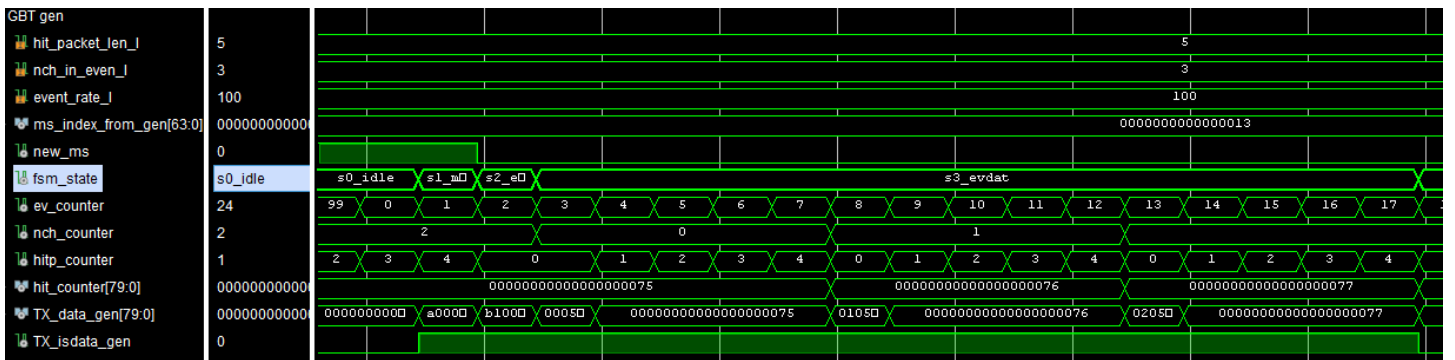


Figure 10: ADC GBT emulator signals



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 ***	hit counter[63 .. 0]				
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 10: 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]

### 3.3 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 lenght 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 11.

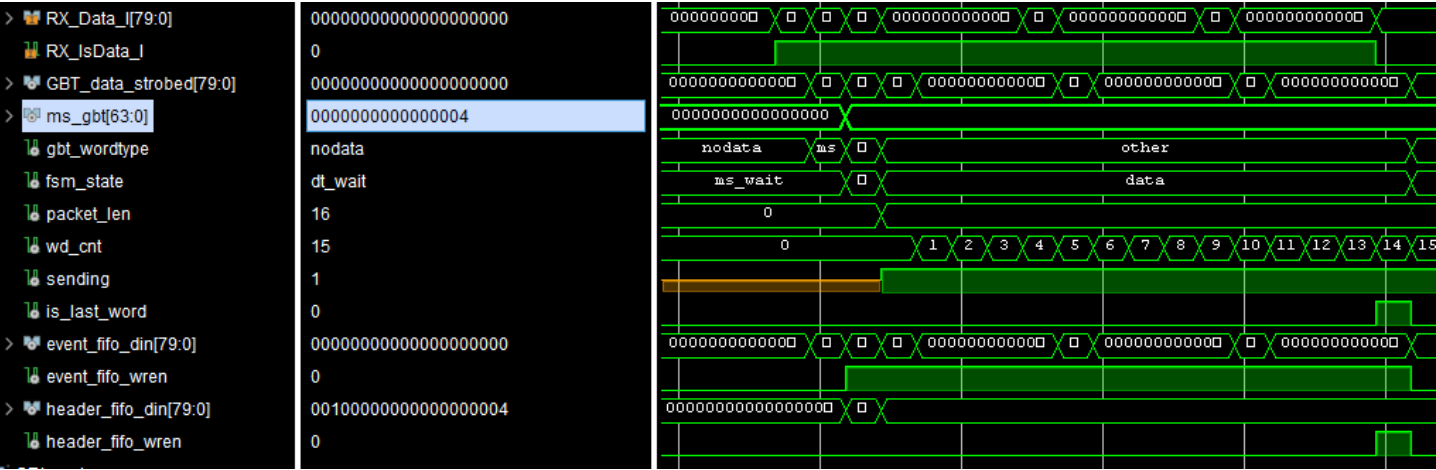


Figure 11: ADC GBT packets reader