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

Finogeev Dmitry, INR RAS

January 7, 2021

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

Contents

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.

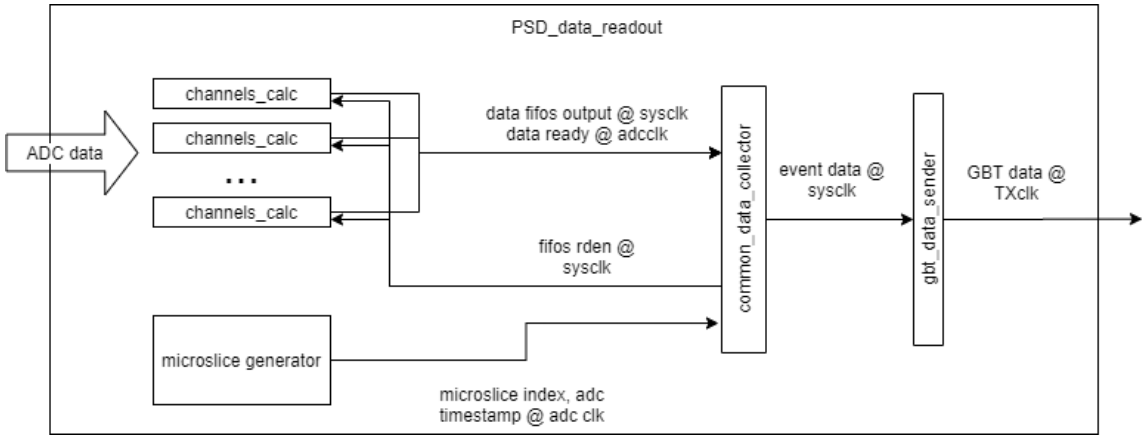

Figure 1: ADC data readout scheme

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 avaliable from slow control.

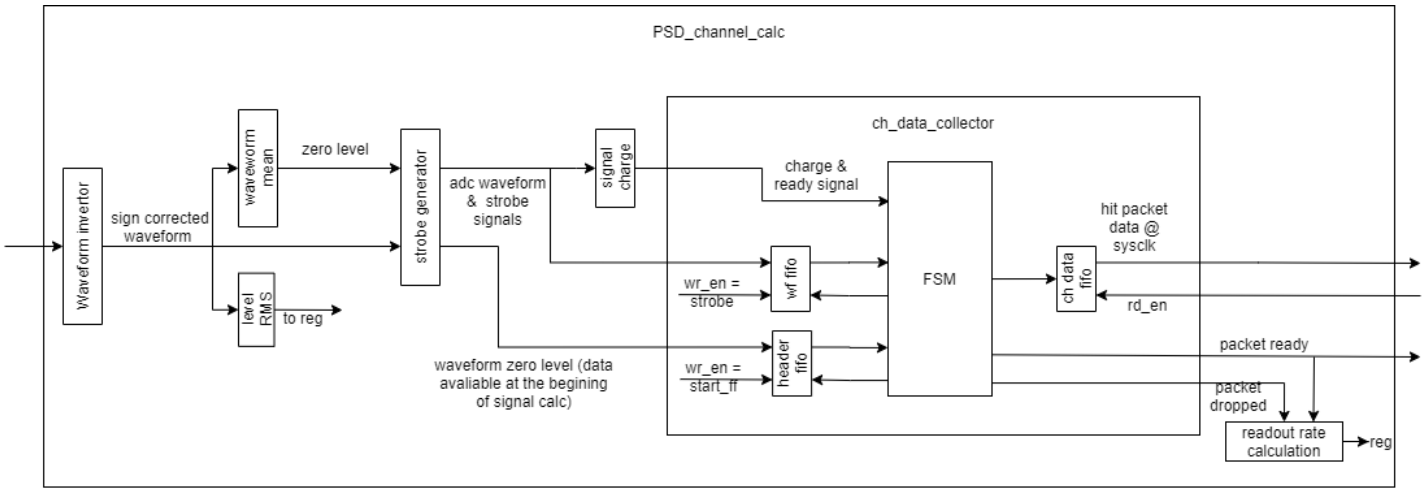

Figure 2: Channel data processing scheme

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

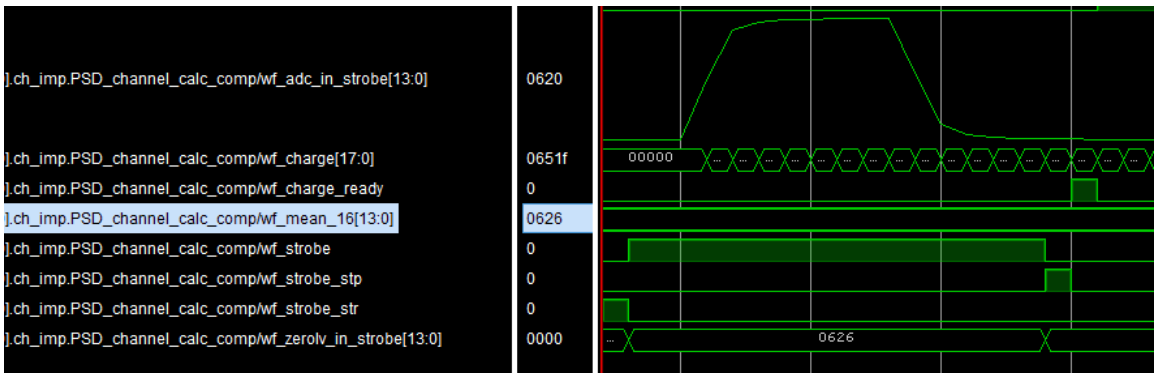

Figure 3: Signal waveform strobe (length 16, offset 3)

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 diagramm of the component is presented on figure 4.

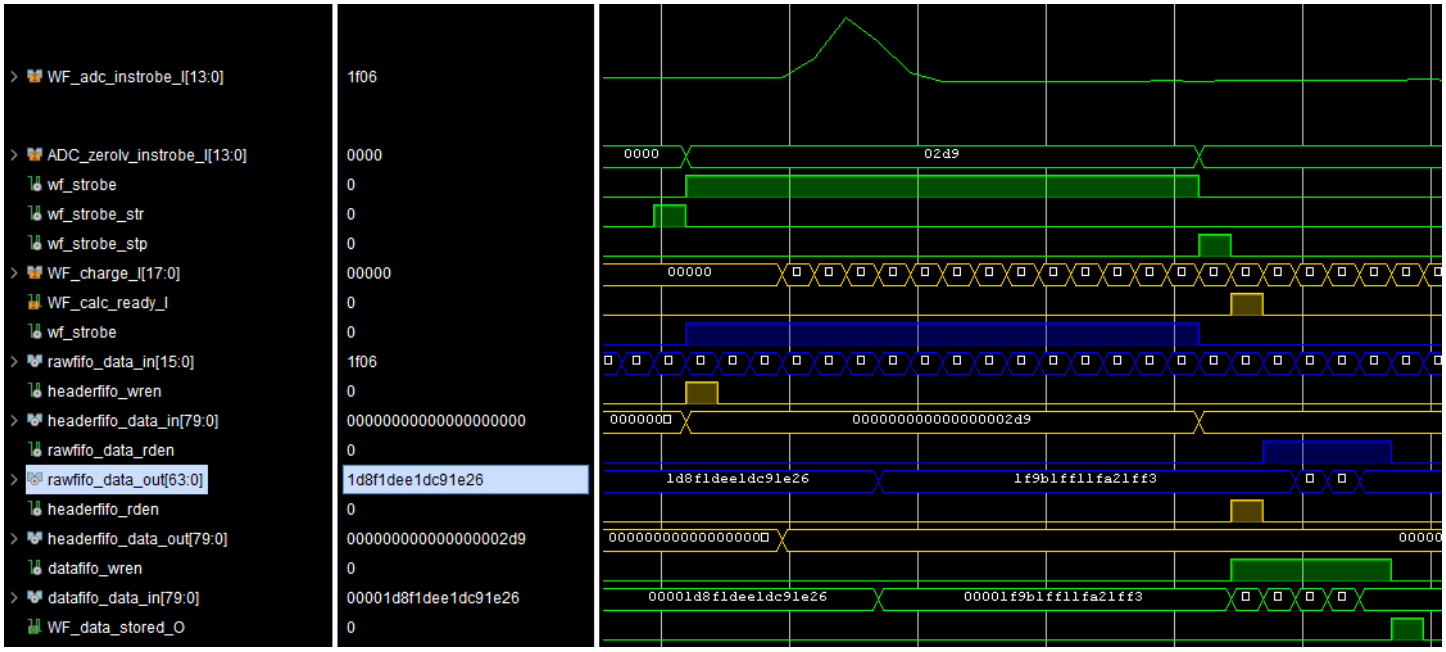

Figure 4: Channel data collecting signals

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.

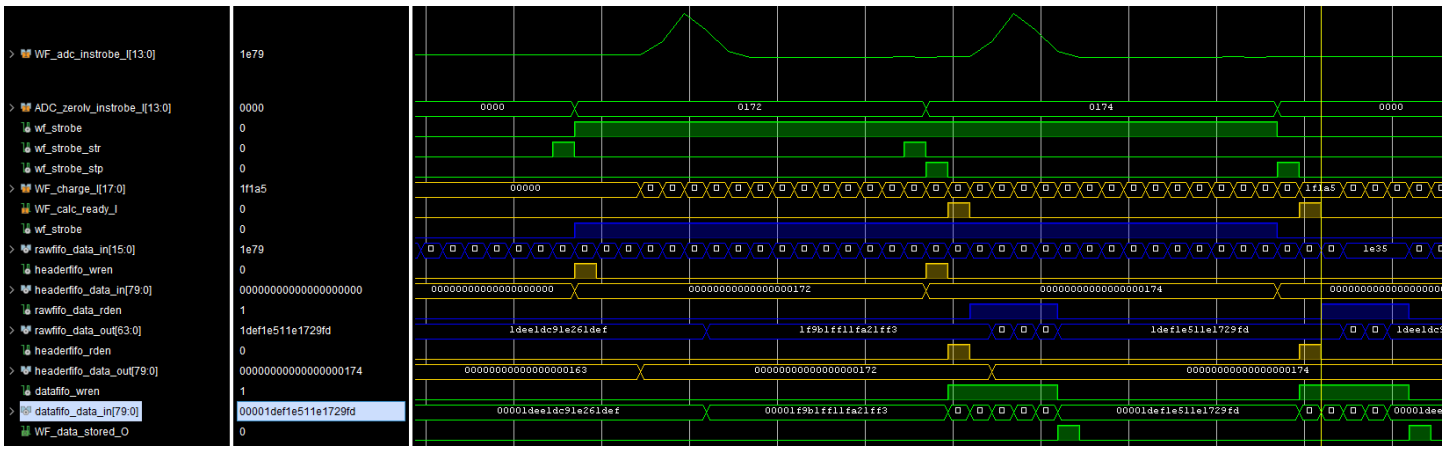

Figure 5: Channel data collecting signals

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. Shematic represented on figure 3.

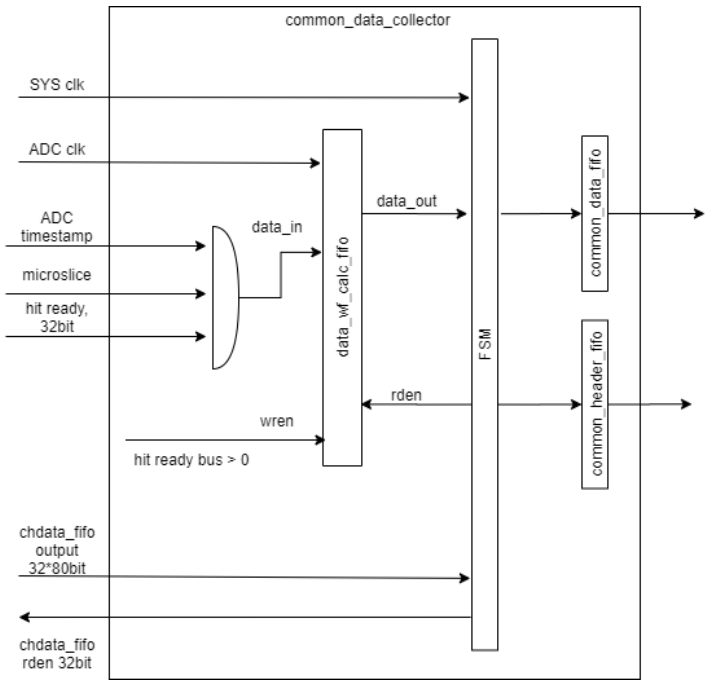

Figure 6: Data collecting scheme from all channels fifos

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.

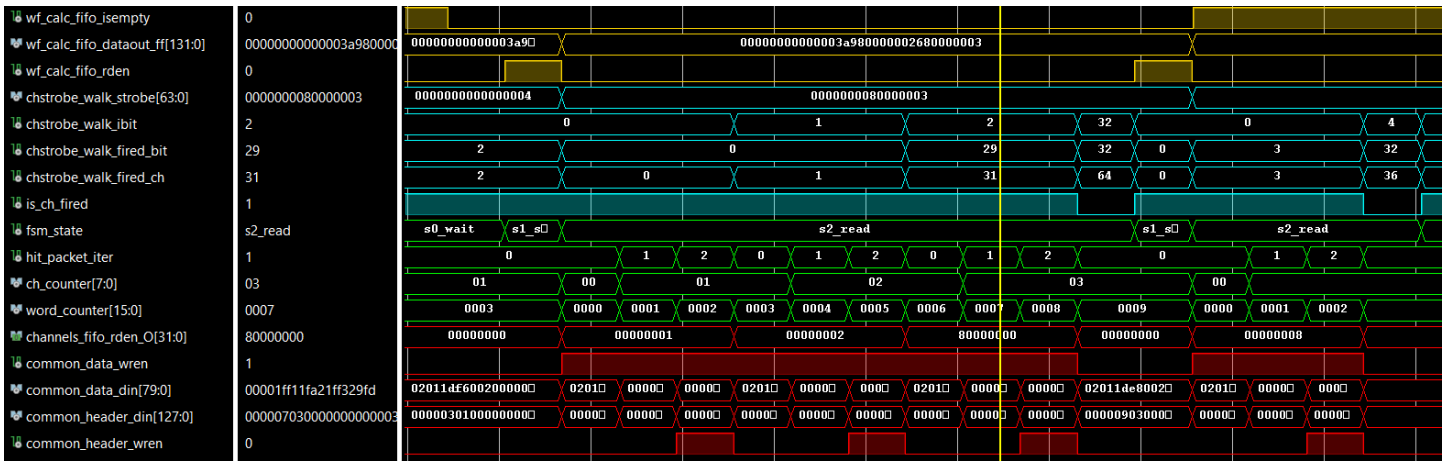

Figure 7: Data collecting signal from all channels fifos

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.

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.

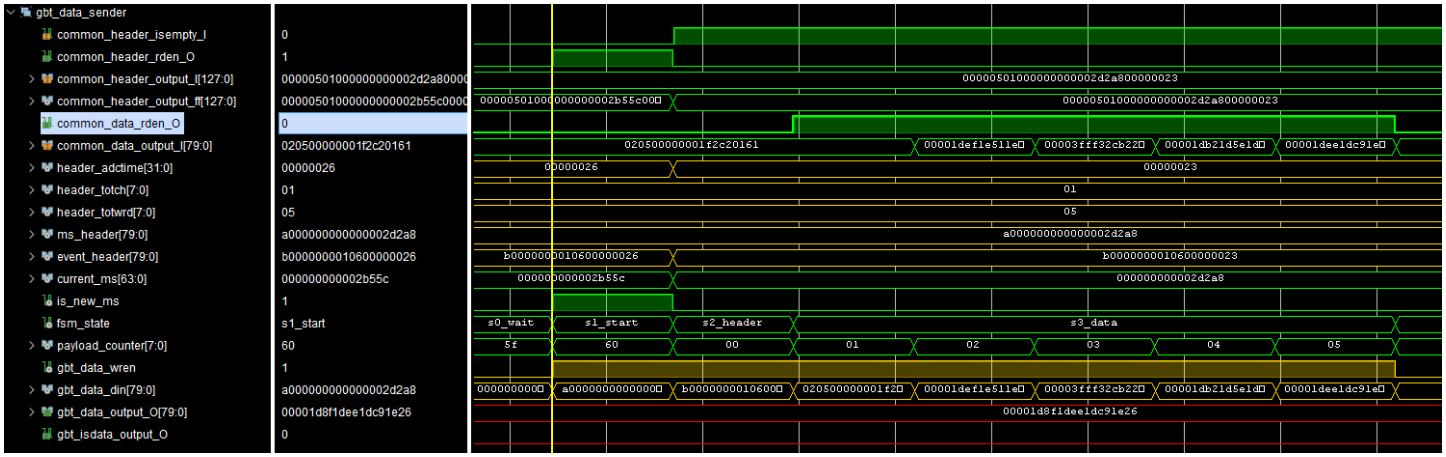

Figure 8: Channel data collecting signals

Data rate limit is 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.

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				waveform length 0..3 [(reg+1)*4]	strobe offset 0..12	control bits	
17	negative channel mask ibit = ich							

Table 5: ADC readout control.

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.

3 CRI modules

3.1 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 9; generated data format is presented on figure 8.

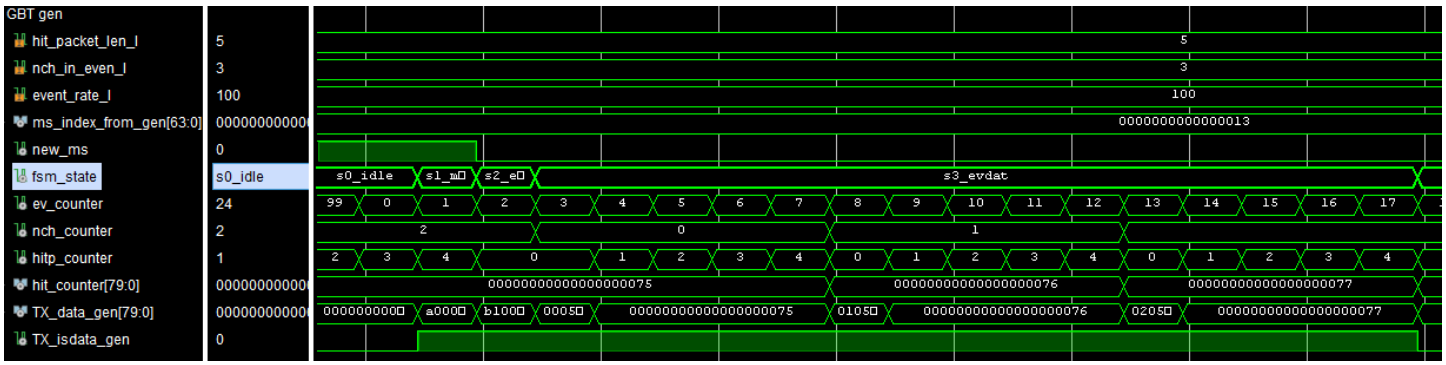

Figure 9: ADC GBT emulator signals

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 8: 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]