

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

Finogeev Dmitry, INR RAS

March 19, 2021

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

Contents

| | | |
|----------|---|----------|
| 1 | ADC data processing | 2 |
| 1.1 | Component channels_calc | 2 |
| 1.2 | Component common_data_collector | 4 |
| 1.3 | Component GBT_data_sender | 5 |
| 2 | ADC control | 6 |
| 2.1 | ADC control units | 6 |
| 2.2 | ADC Control registers | 7 |
| 2.3 | ADC Status registers | 8 |
| 2.4 | EvB Contol | 8 |
| 3 | CRI modules | 9 |
| 3.1 | PSD CRI data sorting | 9 |
| 3.2 | ADC GBT emulator | 9 |
| 3.3 | ADC GBT reader | 10 |
| 3.4 | GBT Data Sorter component | 11 |
| 3.5 | IPbus face component | 11 |
| 3.6 | Evaluation Board for readout | 12 |

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.

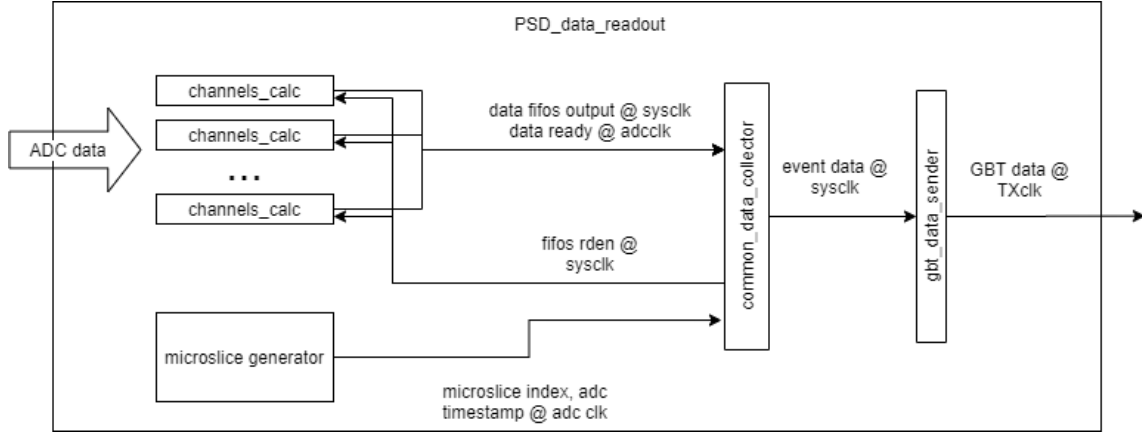


Figure 1: ADC data readout scheme

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.

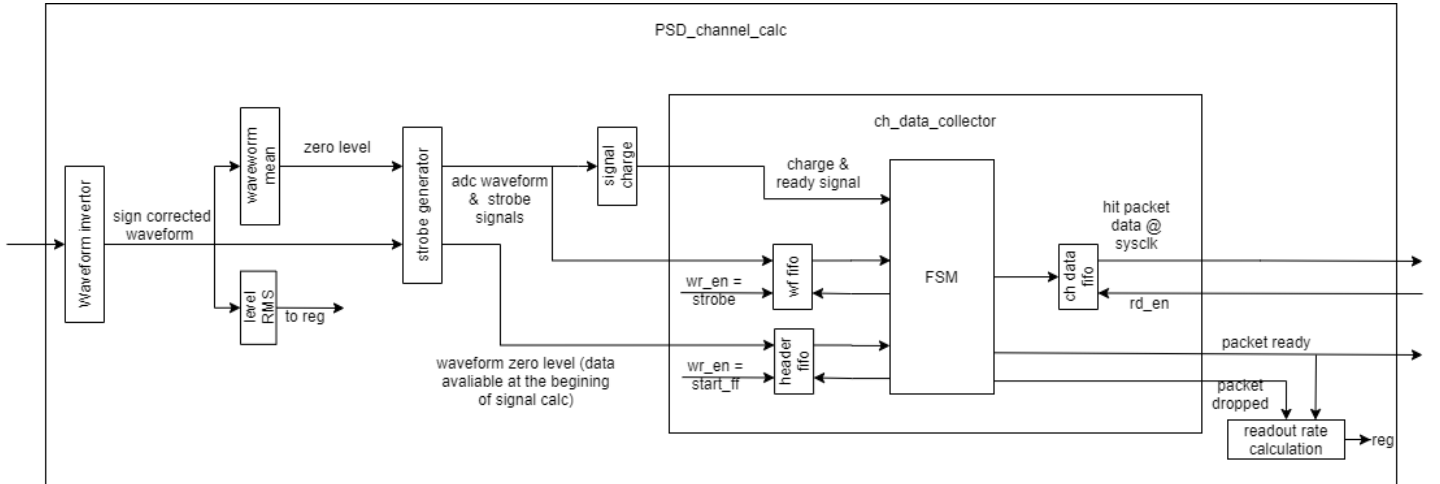


Figure 2: Channel data processing scheme

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.

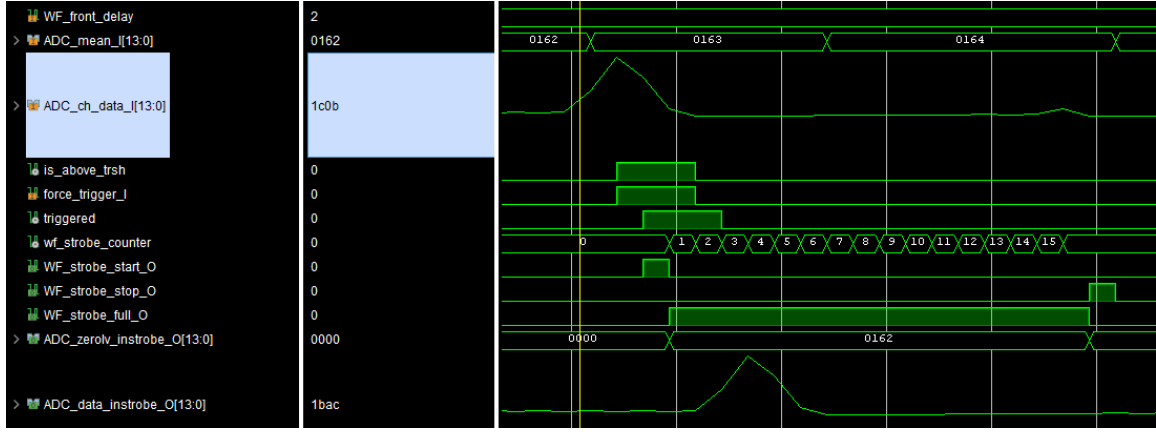


Figure 3: Signal waveform strobe (length 16, offset 3)

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 diagramm of the component is presented on figure 4. The write size of ch_data_fifo should be equal to $\text{ceil}(\text{calculation_delay} / \text{waveform_length}) * \text{waveform_length}$. The size of ch_header_fifo should be $\text{ceil}(\text{calculation_delay} / \text{waveform_length})$. 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.

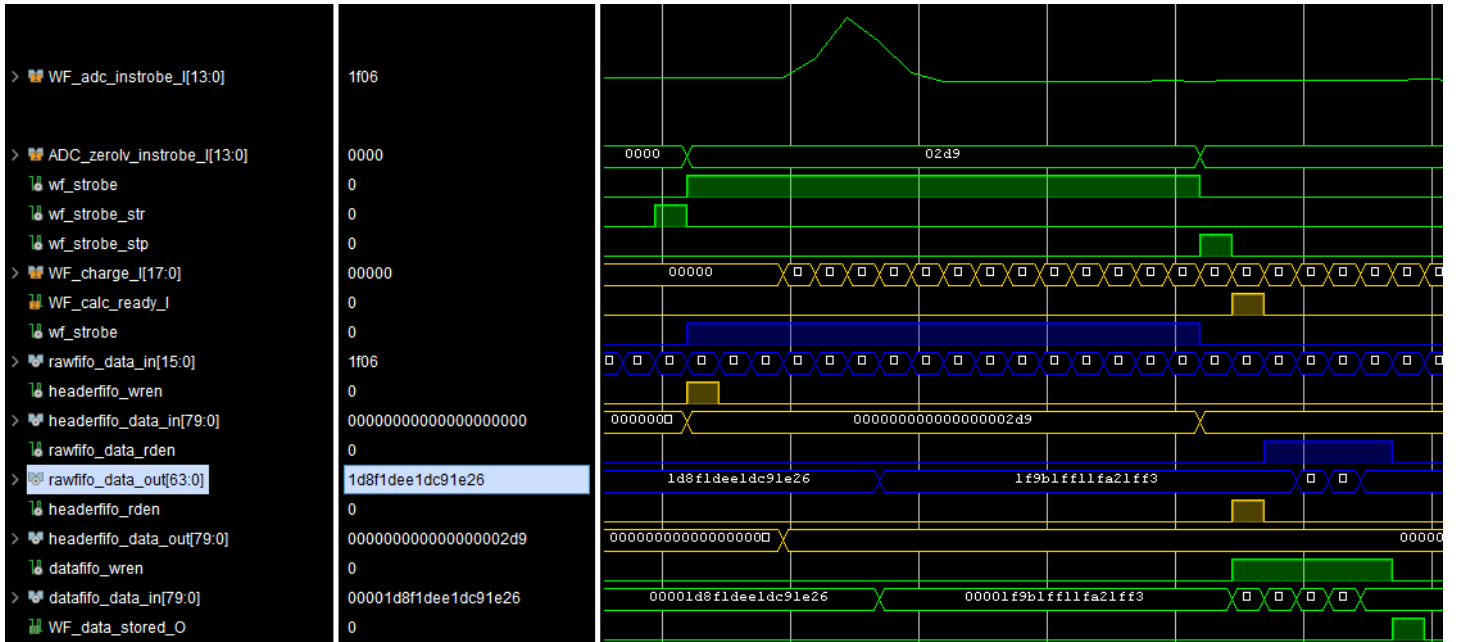


Figure 4: Channel data collecting signals

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.

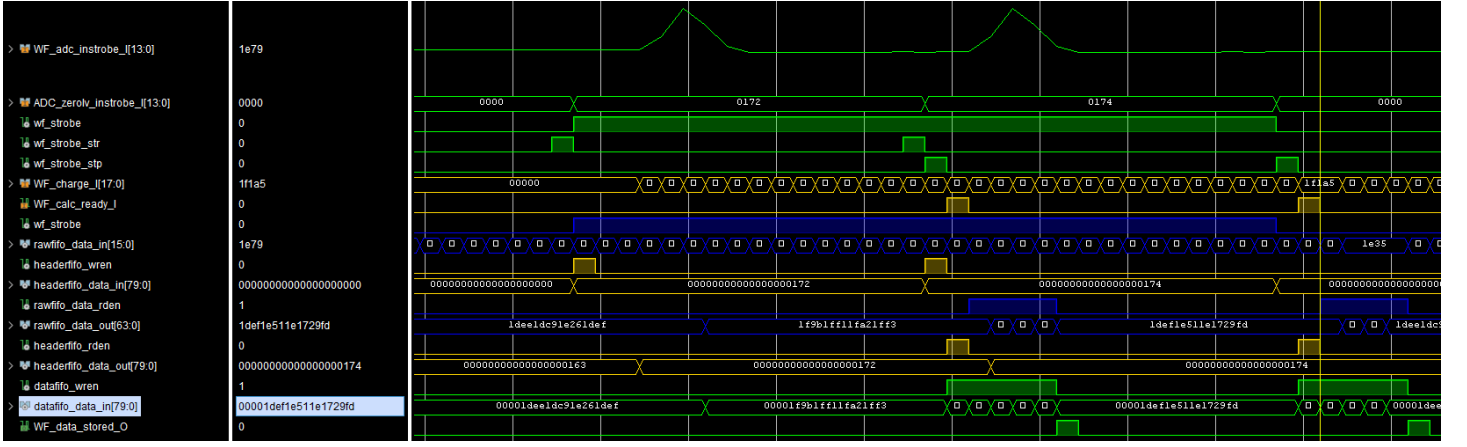


Figure 5: Channel data collecting signals

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.

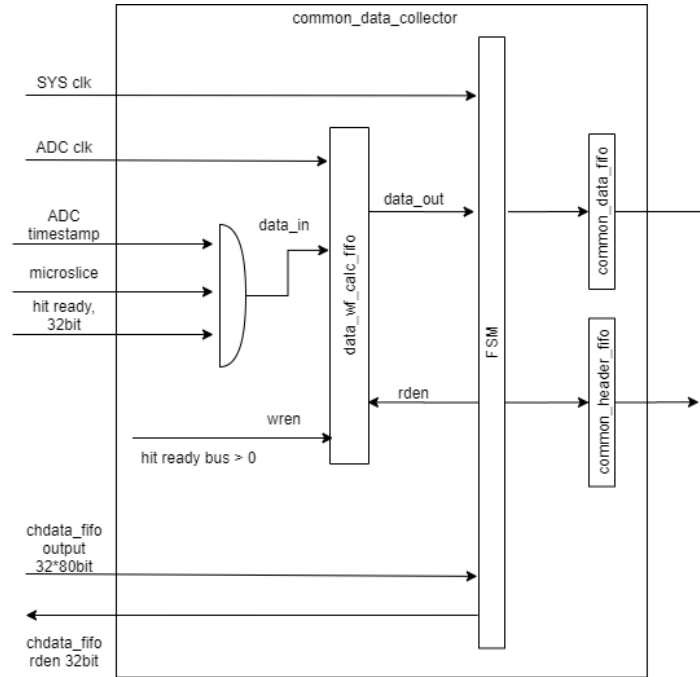


Figure 6: Data collecting scheme from all channels fifos

FSM is switched from wait to start state when data_wf_calc_fifo_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.

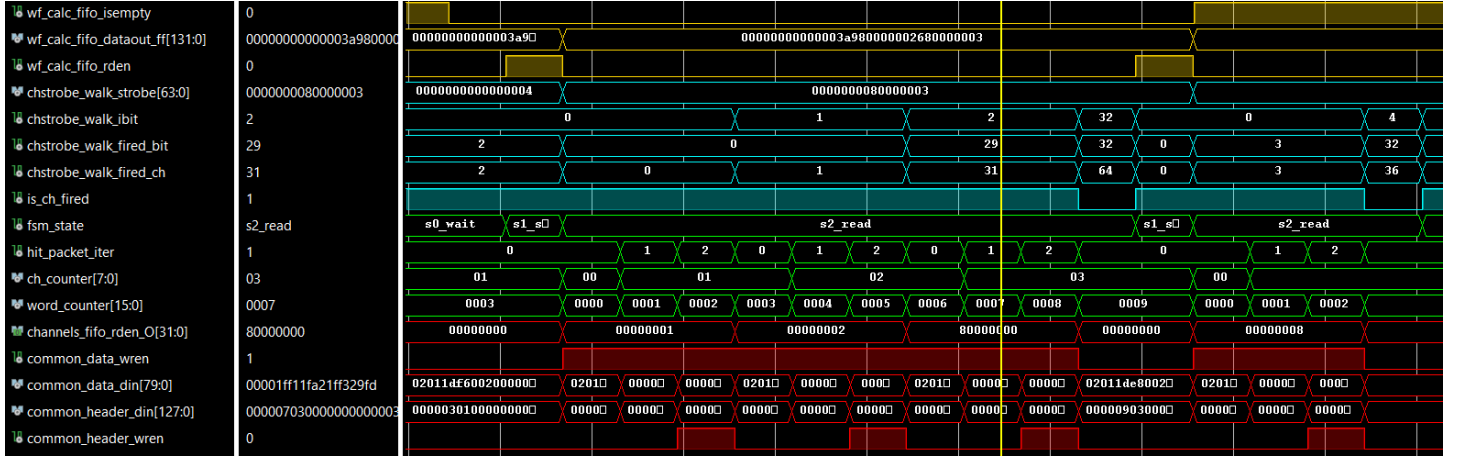


Figure 7: Data collecting signal from all channels fifos

Collecting data from all channels takes two additional FSM cycle. Mean hit rate per channel in case all channels fired is $\text{SYSCKL} / \text{total channels} + 2 \text{ cycle} / \text{packet length}$. 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.

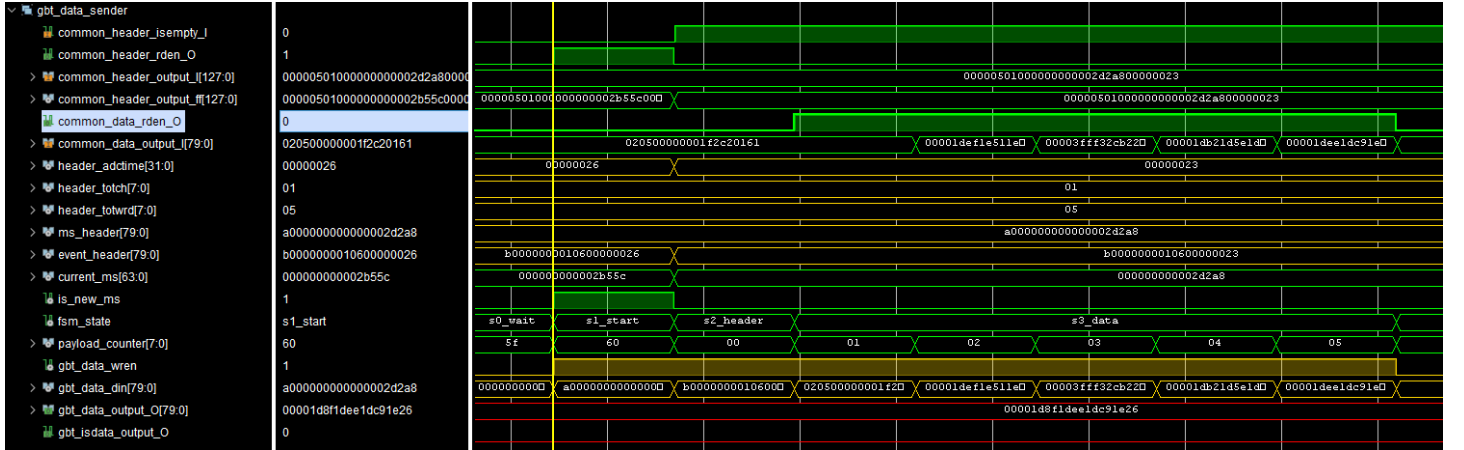


Figure 8: Channel data collecting signals

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 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 ADC control units

Status and Control of ADC and EvB are 64 arrays each of 32 bit words. ADC control system include 4 firmware units: gbt-control-sendet, gbt-control-reader, gbt-status-sender, gbt-status-reader.

gbt-control-sender send control packet (129 X 16bit) via gbt to ADC by start command initiated by software. 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 received 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) |
| | |
| 128 | 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). The transaction is initiated periodically, also control registers is sent back after each new configuration. Status/control packet is prioritized to data flow, and gbt-data-fifo is not readed while transaction. To separate data flow status word starts with 0xE, control with 0xF.

| | | | | |
|------|----------|----------|------------|------------|
| 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 starts 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 one.

2.2 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 mask * | | | | | | | |
| 22 | common trigger output ** | | | | | | | |
| 23 | trigger pulser rate (0x0 = off) | | | | | | | |
| 24 | status send rate (0x0 = off) | | | | control send rate (0x0 = off) | | | |

Table 8: 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 |
| 4 | channel low rate count |
| 5 | reset channels drop counter |

Table 9: Control bits

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

Table 10: HV control via I2C.

2.3 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 | | | | |

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.

2.4 EvB Control

| 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 13: 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 14: Evaluation board control registers.

| bit | description |
|-----|-----------------------|
| 0 | data processing reset |
| 1 | error reset |

Table 15: 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 16: 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 17: GBT status bits

| bit | addr | description |
|-----|------|--------------------------|
| 0 | 16 | control readback correct |

Table 18: control status bits

3 CRI modules

3.1 PSD CRI data sorting

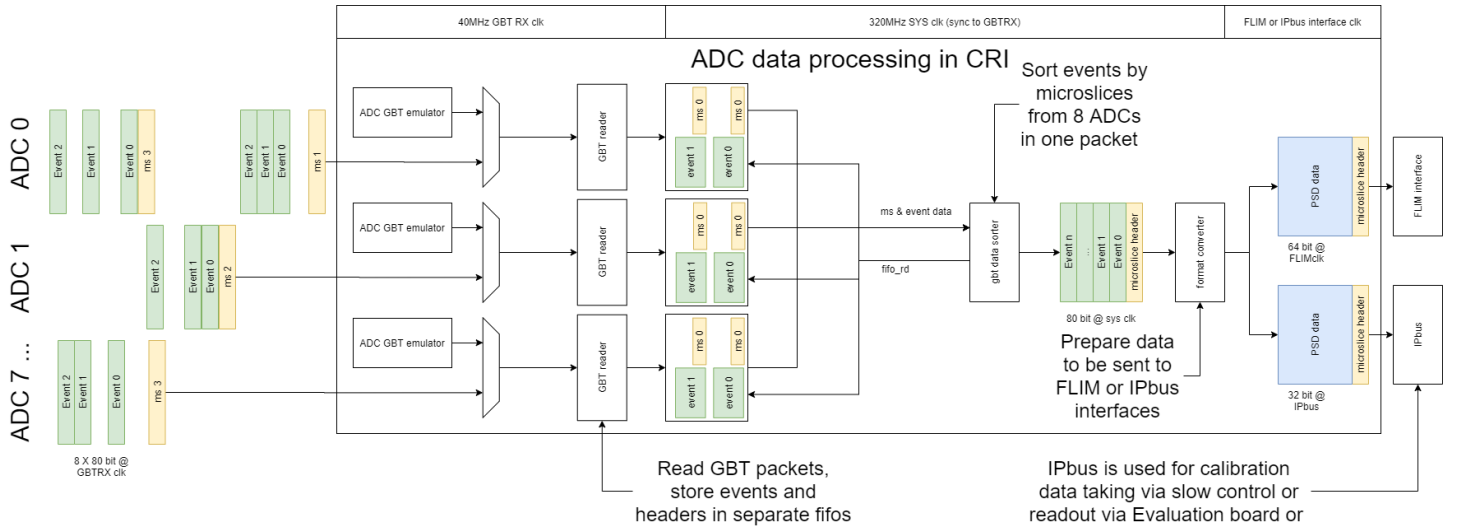


Figure 9: PSD data processing in CRI

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 11.

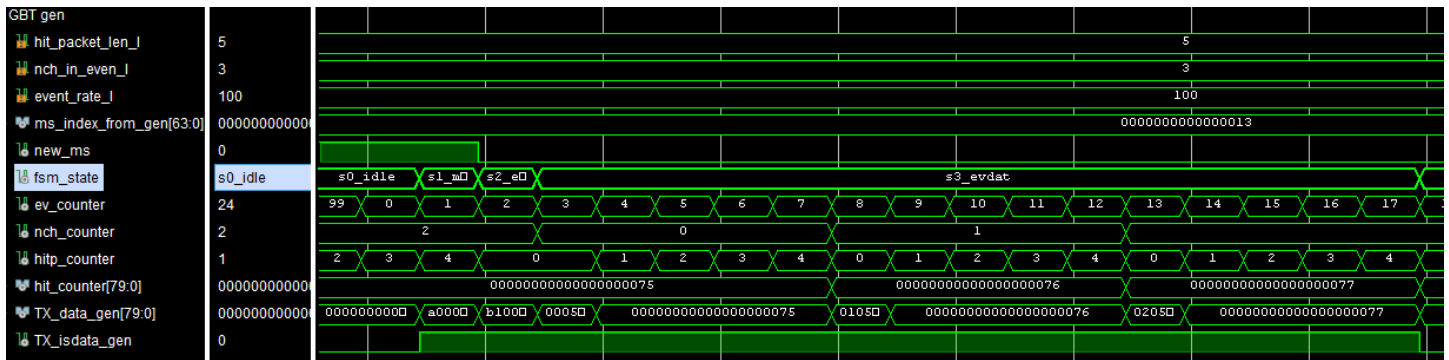


Figure 10: ADC GBT emulator signals

| | | | | | | | | |
|--------------|-----------------------|-----------|-------------------------|----------|----------|--------------|----------|---------|
| 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 *** | | 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 19: 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 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 13.

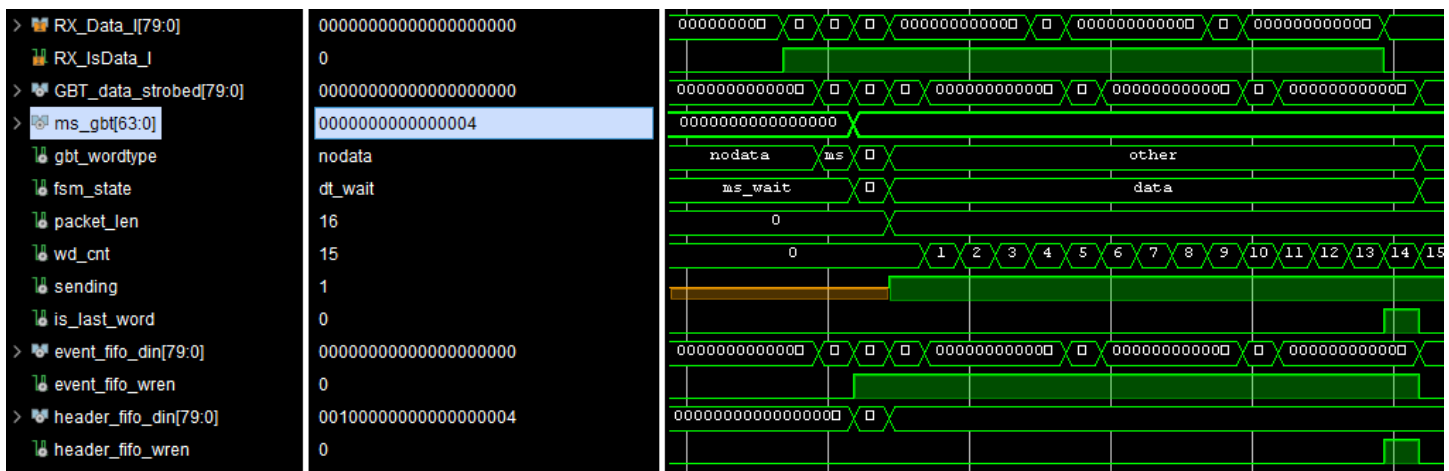


Figure 11: ADC GBT packets reader

3.4 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 microslice are read. Such condition starts counter to wait data from all links. Then counter reach value 127, FSM switched to next-ms state. Next microslice readed from fifo and header with new ms value sent to output stream. Signal diagram presented on figure 12 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).

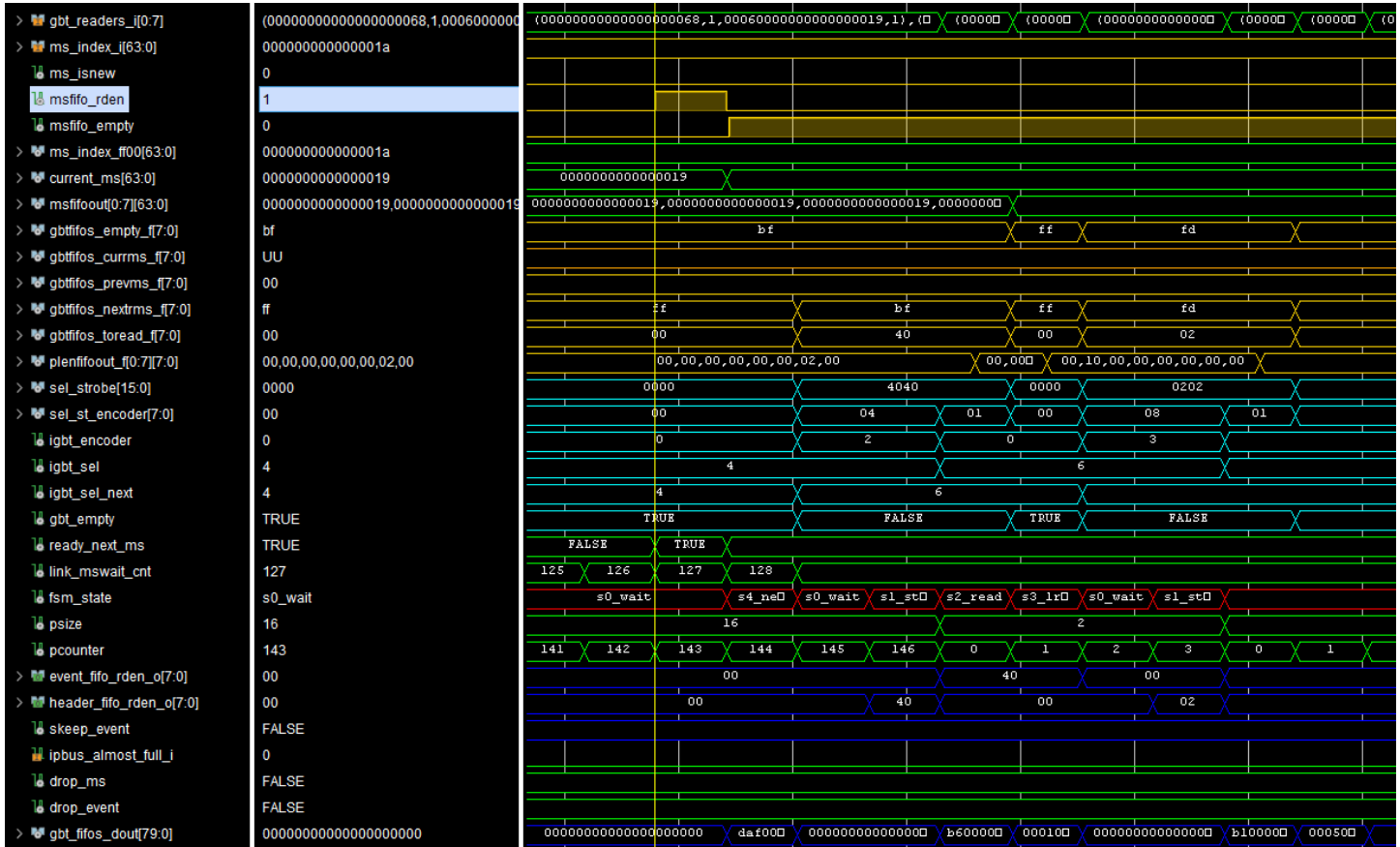


Figure 12: gbt-data-sorter signals diagram: mew ms read and event sent from one link

3.5 IPbus face component

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 13.

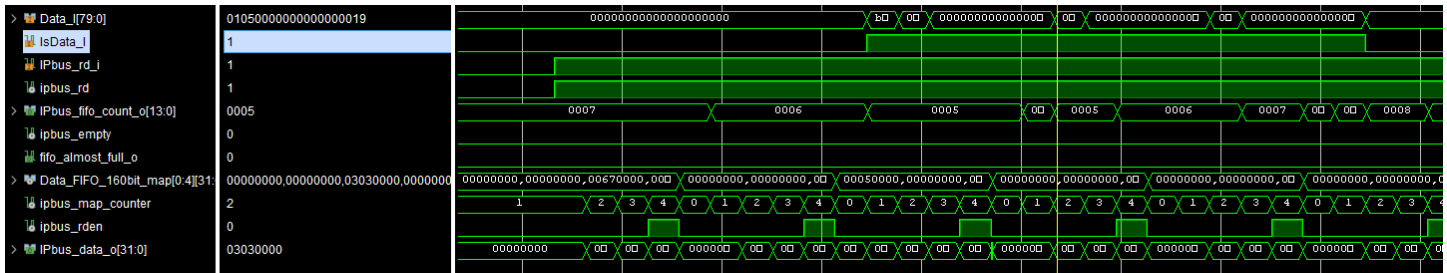


Figure 13: IPbus-face signals diagram

3.6 Evaluation Board for readout

Kintex Evaluation board include CRI data processing module and use IPbus-face module for readout. Real GBT link is connected to 0 link, other links can receive data from emulators. Simple microslice generator is used to provide ms to gbt-data-sorter and to ADC board. Tables 14 and 16 present control and status registers map. GBT slow control and IPbus readout ported via dedicated addresses prioritized to status and control registers (could replace status and control registers), shown in table ??