Selective Trigger Detector Readout Proposal

8/26/16

Jeff Landgraf

I. Configuration

The run control will specify, for each trigger, the list of trigger detectors required to be read out. Additionally it will specify a "prescale" to force the readout of all trigger detectors on some fixed percentage of events.

ZDC, VPD, TOF, MTD, BEMC, EEMC, FMS, FPS…. This needs to be a new mask, as far as I can see, as there is no current mask that separates by detectors, spans the full set of detectors, and keeps the minimum number of bits. (For example the crate index has gaps (for L0 and old crates), has multiple detectors assigned to some crates, and has multiple crates for some detectors). The mask should easily fit into 16 bits.

We will also maintain the list of detectors contributing to each DSM/QT board in the system. This will be a text file with a format along the lines of:

```
      <crate>[<board>]: det1, det2, det3, det4…
      <crate>: det1, det2, det3
```

So we might have:

```
      L1[6]: TOF, MTD, PP2PP
      L1[8]: BBC, ZDC, VPD
      L1[10]: BEMC, EEMC
      L1[11]: TOF, BEMC, DAQ10k

      FMS:  FMS
```

This would correspond to the L1 boards requiring specific, different detectors, but all boards in the FMS crate requiring the FMS detector:

II. Trigger behavior

The L0 CPU software will perform the “logical or” of the masks for each trigger fired on a given event to produce a “trigger detector mask”. It will also check the “readout all detectors” prescale, and if it is satisfied, it will override the trigger detector mask to readout all crates.

III. Communication To DSM/QT crates and L2

The current data sent from L0 to DSM/QT crates on each event consists of the following:

WORD 0	CRC	TOKEN
WORD 1	DSM ADDRESS	
WORD 2	NPRE	NPOST

I propose changing this to:

WORD 0	CRC	TOKEN			
WORD 1	DSM ADDRESS				
WORD 2	TRIGGER DETECTOR MASK	NPRE	NPOST	TrgCmd	DaqCmd

This limits npre/npost to values < 8, higher than the currently supported values.

IV. L1 Data / DAQ File Format

The DAQ file should include the “TRIGGER DETECTOR MASK”. This will be added to the Event Descriptor:

* Current Status of Event Descriptor

The current situation with the event descriptor is:

Word 0	Name				TrgDataFmtVer	
Word 1	Length					
Word 2	bunchXing_hi					
Word 3	bunchXing_lo					
Word 4	ActionWdDetectorBitMask		TrgCmd	🟨	DaqCmd	🟨
Word 5	token	🟨	AddBits (status bits for trg/daq handling)			

Word 6	DSMInput		ExternalBusy	
Word 7	InternalBusy		TrgDetMask (was TcuCtrBunch(lo))	
Word 8	TcuCtrBunch(hi)		DSMAddress	
Word 9	TCU_Mark		Npre	Crate_mask (lo 12 bits)
Word 10	Npost		Res1	Crate_mask (hi 12 bits)

The rows correspond to 32 bits, for the full 11 word event descriptor. I've taken the liberty of indicating the number of bits actually used for each field rather than actual sizes of the EvtDescData structure's elements as it currently stands, as there are many strange cases of 32 bit words from the hardware being moved, and inefficiently unpacked, due to the long historical evolution of the structure.

The tcuCtrBunch is actually a 32 bit quantity which is not aligned in the data structure. The reason I mark tcuCtrBunch(lo) as unused is because this quantity is the "fifo10" from the current TCU. The 16 bit DSMAddress is just the truncated version of tcuCtrBunch.

8 bits of the Res1 field are actually in use as the DAQ10K mask.

Also note that more bits could potentially be considered unused. There are only actually 3 addBits flags used, also the daq10k mask currently only has 6 possible bits.

I propose to remove tcuCounterBunch_lo and replace it with the trigger detector mask. Not only is the data replicated, but also the field is not used in any code checked into CVS so there should be little impact on it's reuse.

V. DSM/QT behavior

The DSM/QT crate software will parse the configuration file at startup, so will "know" which boards must be read-out for which "trigger detector mask". It will only readout necessary boards.

For boards that are not read out, the behavior depends upon the type of Crate:

DSM Crates: fill the board data with 0's

QT Crates: Set the length of the data payload for that board to 0.

In either case, the DSM/QT software will ship the event (or empty event) to L2 as normal. There should be no changes to the core L2 system.

This specific choice, can be discussed, It might be better to have some specific test pattern or flag set so that QA / Data analysis can know that the boards were not read out, rather than assume that the data values were 0 --- TBD.

VI. QA / Bit Checkers

This change will need a minor corresponding change to the QA system and any bit checkers to ensure that empty events are not treated as corrupt events.

VII. Offline Interface

The offline interface should get some function to access the "Trigger Detector Mask".

VII. Progress

1. I updated the proposal for the STP broadcast message.

2. I updated the proposal for passing the TRGDET mask to trigger/daq

I tested wheter the L2Result[] and C2Result[] arrays are currently passed from L1 through all the way to DAQ. They are not. However, I also investigated the event descriptor, and found unused space for the TRGDET mask and updated the proposal to use this space for it.

3. I Implemented the Run Control

The configuration parameters now exist in the run control.

4. TRGDET Bit Definitions:

I have updated the "rtsSystems.h" file with the following values for the TRGDET bits:

```
// Trigger Detector Bit Mask
#define TRGDET_ZDC 0
#define TRGDET_BBC 1
#define TRGDET_VPD 2
#define TRGDET_TOF 3
#define TRGDET_ETOF 4
#define TRGDET_MTD 5
#define TRGDET_BEMC 6
#define TRGDET_EEMC 7
#define TRGDET_PP2PP 8
#define TRGDET_FMS 9
#define TRGDET_FPS 10
```

5. Strawman detector to crate mapping:

I also created an initial trigger detector to crate mapping in /RTS/conf/handler/trgDetToTriggerCrate.txt. This is not in any way final, as I don’t have the full information.

6. Helper function for trigger crate software:

There is a helper function “UINT16 getTrgDetRequiredMask(char *mynodename)” which takes as an argument the name of the trigger crate and returns the required TRGDET mask. The function is provided by RC_Config.h (through rtsCfgLib).

VIII. Still To Do:

1. Implement the L0 code assigning the trgdet masks

2. Actually change the STP broadcast message. (This is the one task that must be done at the same time for all the software.)

3. Implement the changes in the trigger crates, by skipping readout if the detector mask is not satisfied.