sparsification on CBC

previously discussed in Dec. 2012 systems meeting

https://indico.cern.ch/getFile.py/access?contribId=9&sessionId=2&resId=1&materialId=slides&confId=219588

will re-state material here

+ a few thoughts on CBC3 output configurability

data from CBC

S1	S1	S1	S1	S1	S1	S1	S1	B1	B1
B1	B1	B1	S2	S2	S2	S2	S2	S2	S2
S2	B2	B2	B2	B2	B2	S3	S3	S3	S3
S3	S3	S3	S3	B3	B3	B3	B3	B3	**R**

one of Francois' proposed schemes
40 bits / 25 nsec
10 lines @ 160 Mbps

1 readout bit per 25 nsec (readout bit = triggered data)

1 unsparsified readout bit / 25 ns limits L1 trigger rate to ~**140 kHz** (see Dec. 2012 slides)
(CBC output frame ~ 6.8 usec)

3.2 Gbps LP-GBT bandwidth => 80 bits / 25 nsec

16 CBC chips / module -> 16 unsparsified bits

readout data take **20%** of LP-GBT bandwidth

more readout data from CBC

S1	S1	S1	S1	S1	S1	S1	S1	B1	B1	**R1**
B1	B1	B1	S2	S2	S2	S2	S2	S2	S2	**R2**
S2	B2	B2	B2	B2	B2	S3	S3	S3	S3	**R3**
S3	S3	S3	S3	B3	B3	B3	B3	B3	**-**	**R4**

add an extra 160 Mbps output line

11 lines @ 160 Mbps

4 readout bits per 25 nsec

5 readout bits raises tolerable L1 trigger rate to ~**540 kHz**

16 CBC chips / module -> 5 x 16 = 80 unsparsified bits

takes **80%** of LP-GBT bandwidth
not much room for stub data

even more readout data from CBC

S1	S1	S1	S1	S1	S1	S1	S1	B1	B1	**R1**	**R5**
B1	B1	B1	S2	S2	S2	S2	S2	S2	S2	**R2**	**R6**
S2	B2	B2	B2	B2	B2	S3	S3	S3	S3	**R3**	**R7**
S3	S3	S3	S3	B3	B3	B3	B3	B3	**-**	**R4**	**R8**

add one more 160 Mbps output line

12 lines @ 160 Mbps

8 readout bits per 25 nsec

8 readout bits raises tolerable L1 trigger rate to ~**1.1 MHz**

16 CBC chips / module -> 5 x 16 = 80 unsparsified bits

takes **160%** of LP-GBT bandwidth
no option but to sparsify readout data

but where?

advantages to sparsifying at the concentrator level

concentrator functionality can be finalised a bit later on

allows to accommodate shifting specifications as other CMS sub-detectors (e.g. ECAL) and systems (e.g. trigger) reach a clearer picture of what is achievable, and/or what data is really required from the tracker

should be 8x

some functions (e.g. time-stamping) are performed once (not ~~16~~x) -> power savings?

digital functionality associated with sparsification less power hungry in 65 nm concentrator technology

concentrator can do some simple checking of CBC data

e.g. all headers the same? error bits set? – strong check on correct FE functionality
can flag "chips in error" to higher DAQ levels

can implement various modes of operation

1) stub data + unsparsified L1 triggered data - the way it has been envisaged up to now
2) unsparsified L1 data only (L1 trigger rate up to ~~~57~~0 kHz)
3) stub data + sparsified L1 triggered data

actually ~700 kHz if use 100% GBT BW

4) mixed mode- i.e. higher (e.g. ~300 kHz) L1 rate with unsparsified data, but reduced stub data volume (e.g. less or no bend info, or less stubs)
5) ….

clearly more complicated, but an "interesting" chip to design and optimise?

a few thoughts on output data configurability

S1	S1	S1	S1	S1	S1	S1	S1	B1	B1	**R1**	**R5**
B1	B1	B1	S2	S2	S2	S2	S2	S2	S2	**R2**	**R6**
S2	B2	B2	B2	B2	B2	S3	S3	S3	S3	**R3**	**R7**
S3	S3	S3	S3	B3	B3	B3	B3	B3	**-**	**R4**	**R8**

S1	S1	B1	S2	S2	B2	S3	S3	B3	B1	**R**	**R**
S1	S1	B1	S2	S2	B2	S3	S3	B3	B2	**R**	**R**
S1	S1	B1	S2	S2	B2	S3	S3	B3	B3	**R**	**R**
S1	S1	B1	S2	S2	B2	S3	S3	B3	**R**	**R**	**R**

4 MSBs of bend info

LSBs of bend info + readout

arranging output data like this could allow CBC to adapt output data as requirements become clearer

example 1

perhaps turns out that 4 MSBs of bend info sufficient
and max. trigger rate < 540 kHz

S1	S1	B1	S2	S2	B2	S3	S3	B3	B1	R	R
S1	S1	B1	S2	S2	B2	S3	S3	B3	B2	R	R
S1	S1	B1	S2	S2	B2	S3	S3	B3	B3	R	R
S1	S1	B1	S2	S2	B2	S3	S3	B3	R	R	R

back to 10 output lines required (instead of 12)

=> simplification of hybrid

example 2

perhaps turns out that bend info not required at all
and max. trigger rate < 540 kHz

S1	S1	B1	S2	S2	B2	S3	S3	B3	B1	R	R
S1	S1	B1	S2	S2	B2	S3	S3	B3	B2	R	R
S1	S1	B1	S2	S2	B2	S3	S3	B3	B3	R	R
S1	S1	B1	S2	S2	B2	S3	S3	B3	R	R	R

only 7 output lines required

=> even more simplification of hybrid

other examples can be imagined
e.g. less stubs / CBC acceptable
lower or higher trigger rate requirements

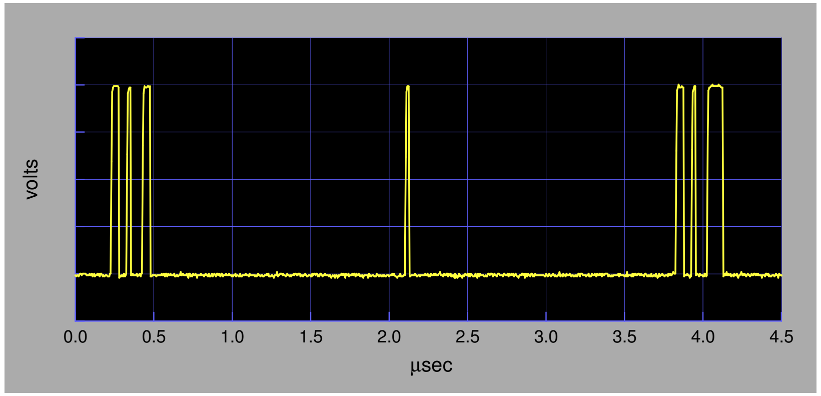
extra

reminder of the readout mechanism and issues

- L1 trigger transfers timeslice of data from pipeline memory to readout buffer memory (FIFO)

  readout commences promptly if no readout currently in progress
  else data remains in FIFO till its turn comes to be read out

CBC1 output for 2 consecutive triggers



- if burst of triggers then FIFO fills up

  subsequent triggers must be vetoed till space freed up

- 2 factors affect trigger rate capability

  readout time – i.e. data frame length

  must be less than average time between triggers for good efficiency
  (good efficiency here means need to veto triggers only occurs rarely)

  buffer depth

  deeper buffer allows to cope with prolonged burst of closely spaced triggers

- some examples

  header includes triggered pipeline address + error bits

  - APV25 readout time 7 usec (128 + 12 samples, 20 Msps)

    for average trigger rate 100 kHz => average trigger separation 10 usec
    buffer depth 10 (in deconvolution)

    4 bit gap between frames for CBC

  - CBC1 readout time 3.6 usec (128 + 12 + 4 bits, 40 Mbps), buffer depth 32

    can cope comfortably with L1 trigger rate > 200 kHz

  - CBC2 readout time 6.75 usec (254 + 12 + 4 bits, 40 Mbps), buffer depth 32

    similar to APV25

simulating trigger acceptance efficiency(unsparsified case)

simple simulation of trigger acceptance efficiency shows effect of buffer depth and average trigger rate

generate random trigger distribution (in time) corresponding to average trigger rate

=> total no. of triggers for time interval simulated -> N_{TOT}

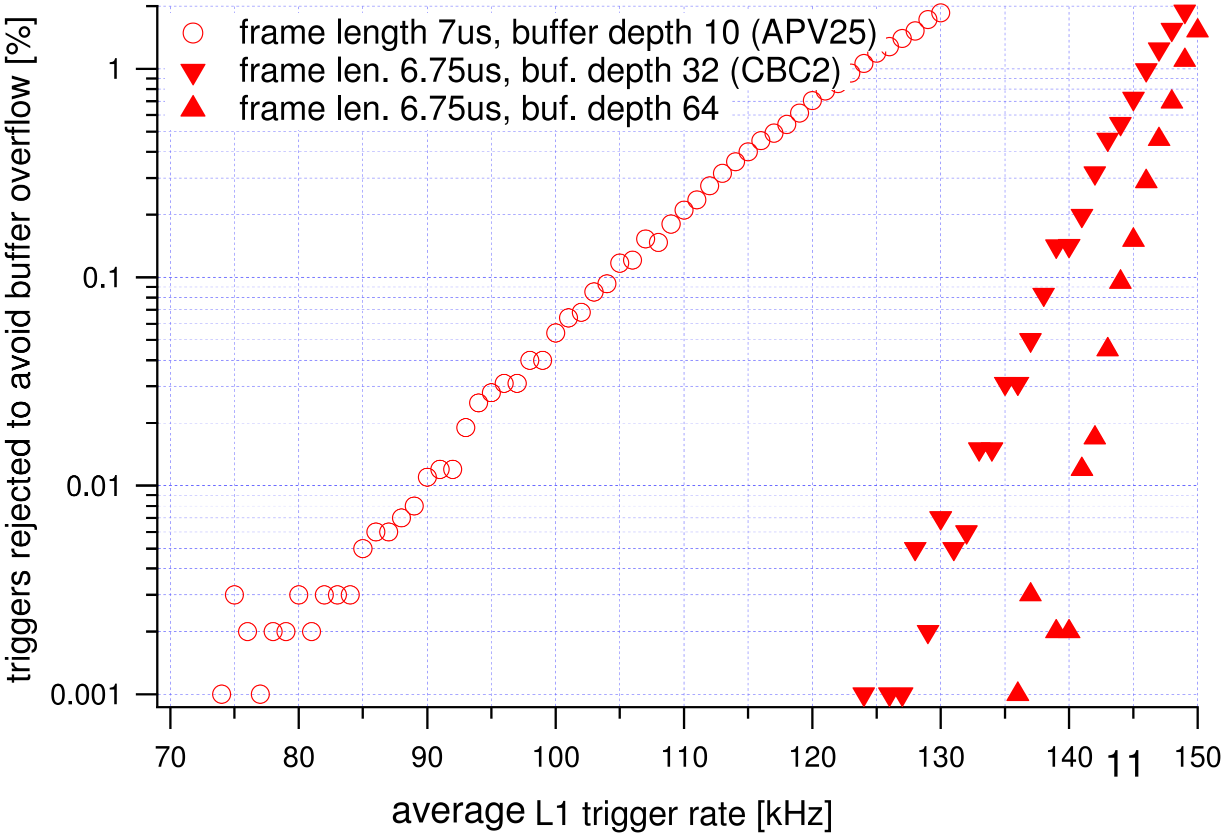
no. of events in FIFO incremented every time trigger occurs

decremented when complete frame has been read out

trigger rejected (or vetoed) if would cause FIFO to overflow

keep count of no. of triggers vetoed -> N_{VETO}

N_{VETO}/N_{TOT} [%]



simulation shows:

@ 100 kHz APVE will veto ~ 0.05% of triggers

at same veto level CBC2 can happily cope with 135 kHz ave. trigger rate

increases to > 140 kHz for twice the buffer depth

interesting to simulate a few other cases

to cope with higher trigger rate, need to reduce output frame length

=> higher output bit rate

take CBC2 as starting point and apply factor (CBC2 frame length 6.75 us @ 40 Mbps)

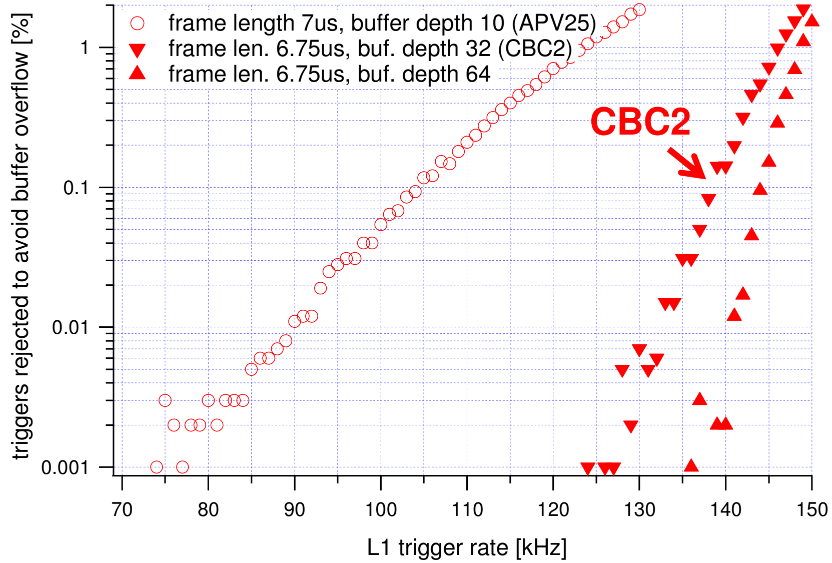
x2 => frame length 3.4 usec, output bit-rate 80 Mbps

x4 => frame length 1.7 usec, output bit-rate 160 Mbps

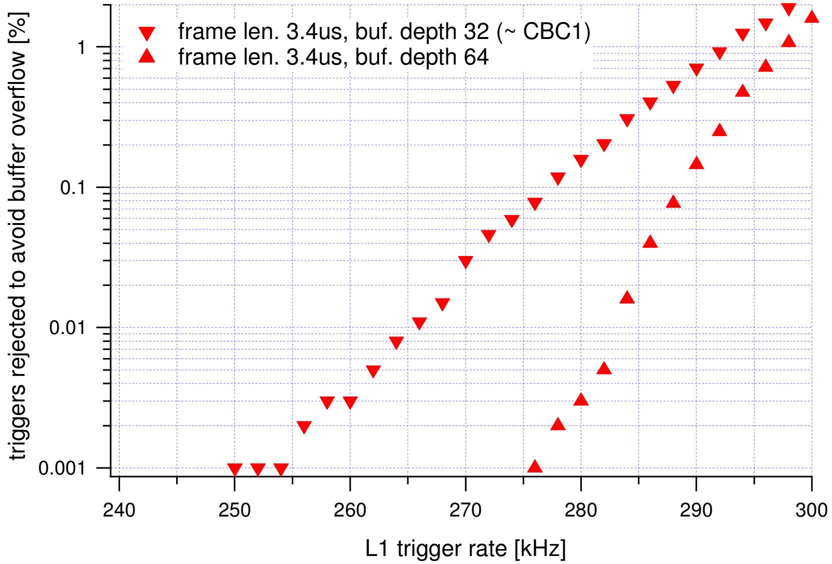
x8 => frame length 0.85 usec, output bit-rate 320 Mbps (e.g. 2 lines @ 160 Mbps)

simulation results

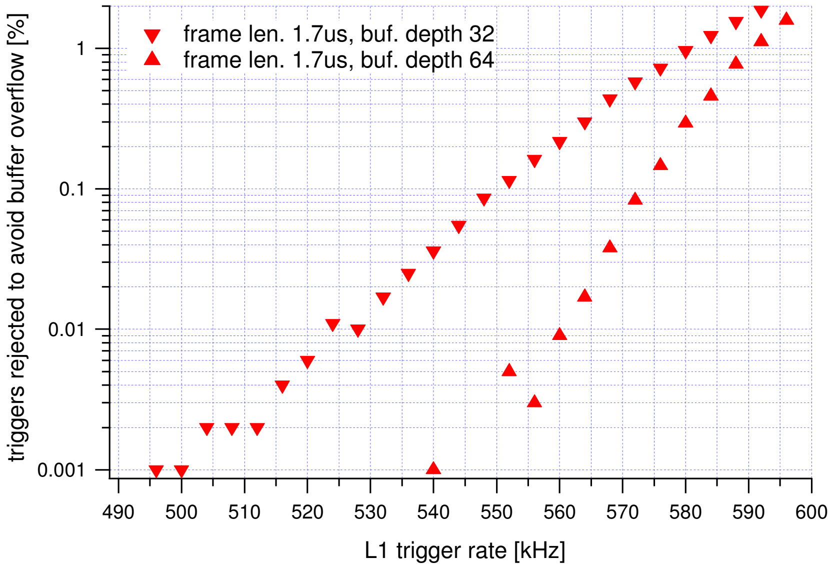
previous picture



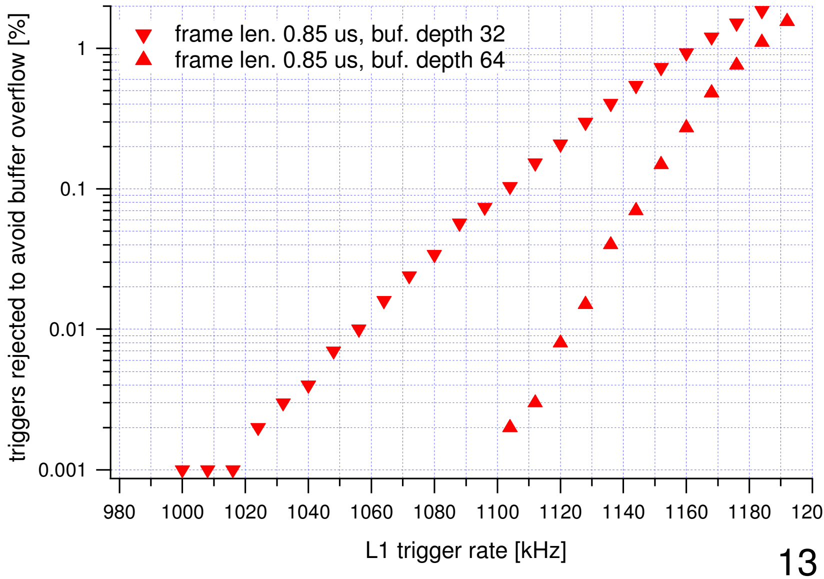
CBC2 x2



CBC2 x4



CBC2 x8



simulation results summary

at 0.05% efficiency level (5 triggers vetoed out of every 10,000)

	frame length	tolerable trigger rate for 32 deep buffer	tolerable trigger rate for 64 deep buffer
CBC2	6.75 us	135 kHz	143 kHz
CBC2 x2	3.4 us	270 kHz	285 kHz
CBC2 x4	1.7 us	540 kHz	570 kHz
CBC2 x8	0.85 us	1.08 MHz	1.14 MHz

observations

no big advantage from increased buffer depth – 32 probably enough already

Atlas proposed trigger rate capability (500 kHz) achieved for CBC2 x4