

CBC and high trigger rate issues

some simulations of trigger rate capability vs. data frame readout rate and buffer depth

prompted by curiosity, but relevant to new architecture choices

some early thoughts on system architectures (FE module only) for coping with high trigger rate

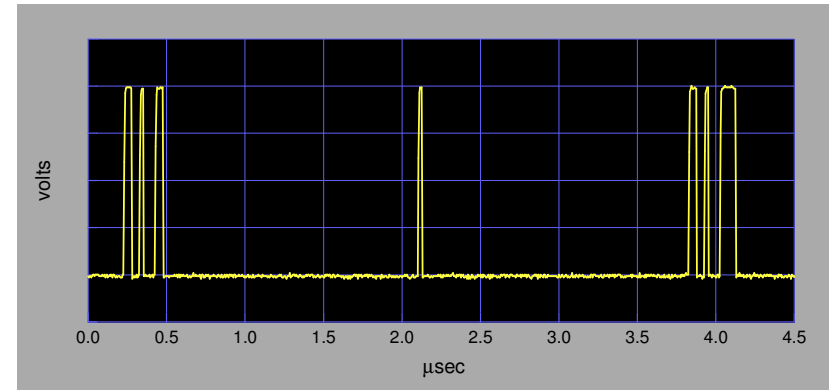
reminder of the readout mechanism and issues

- 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

- 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)

header includes triggered pipeline address + error bits



- CBC1 readout time 3.6 usec (128 + 12 + 4 bits, 40 Mbps), buffer depth 32
can cope comfortably with L1 trigger rate > 200 kHz



4 bit gap between frames for CBC

- CBC2 readout time 6.75 usec (254 + 12 + 4 bits, 40 Mbps), buffer depth 32
similar to APV25

simulating trigger acceptance efficiency(unsparisified 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}

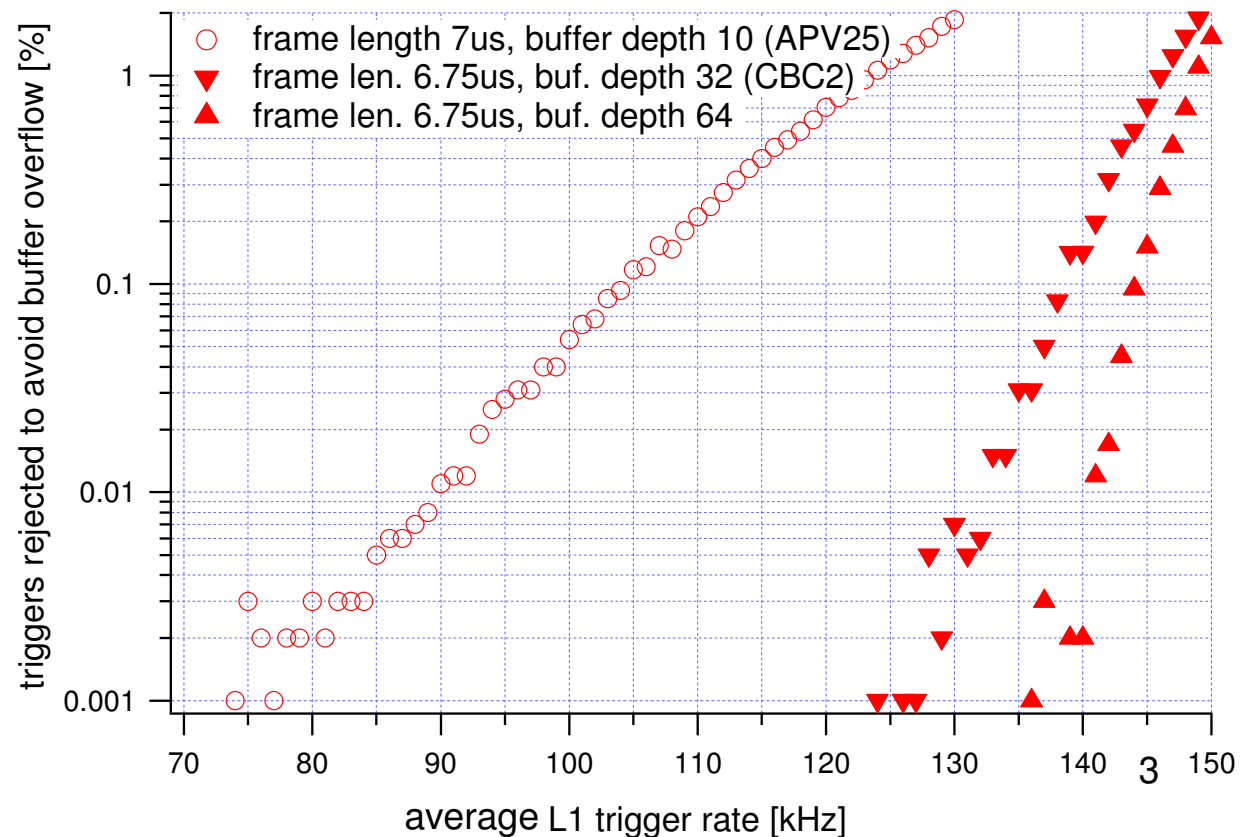
N_{VETO}/N_{TOT} [%] →

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

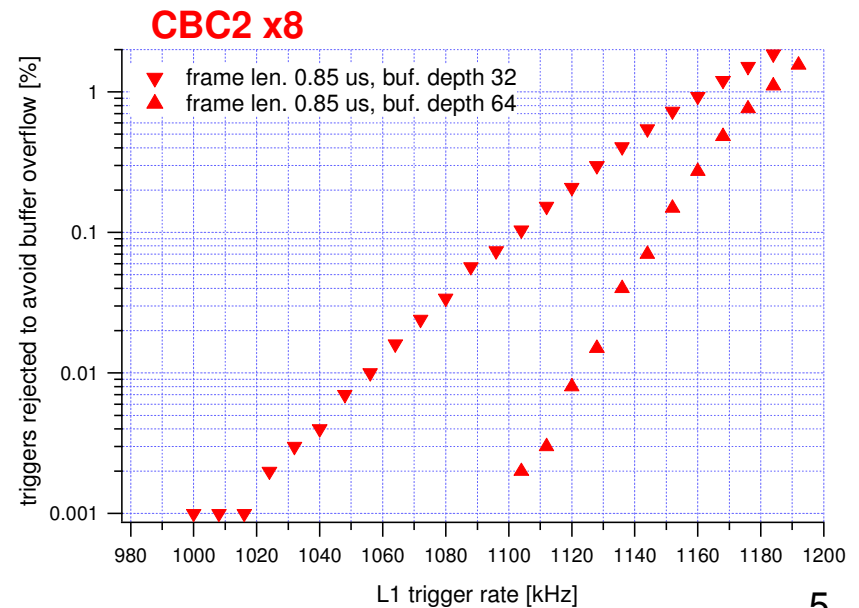
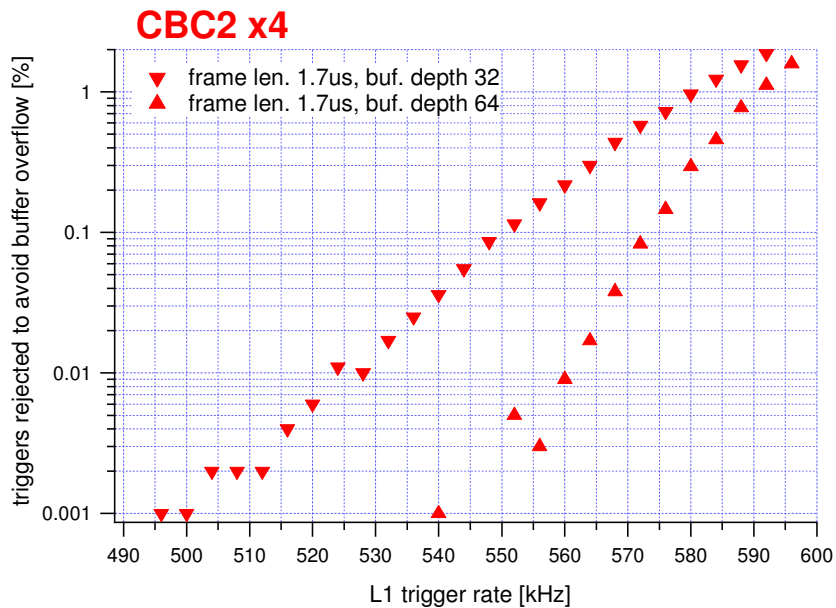
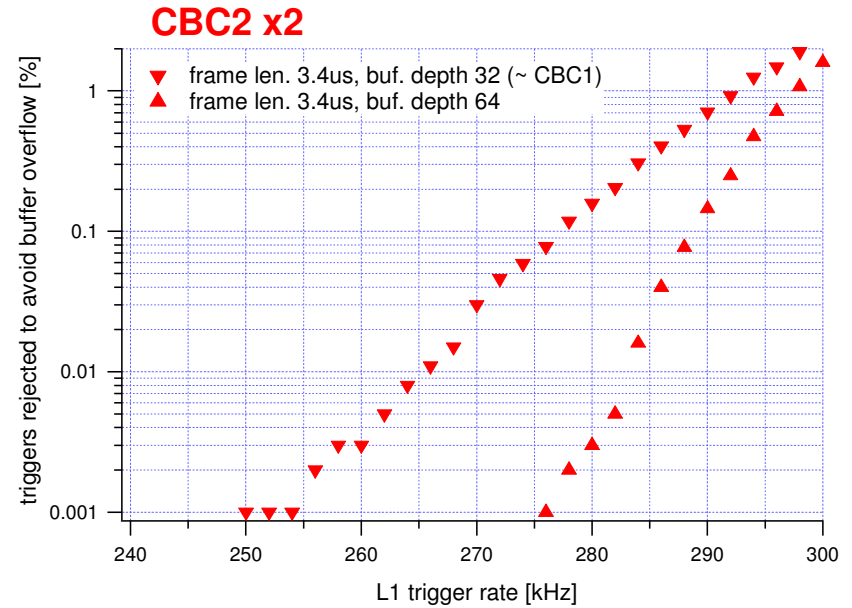
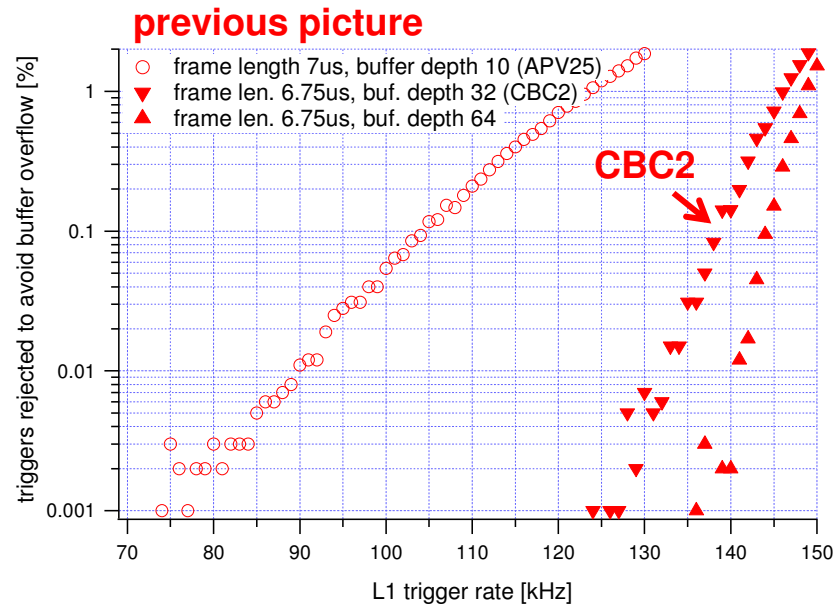
take CBC2 as starting point and apply factor (CBC2 frame length 6.75 us @ 40 Mbps)

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



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

data volumes

current situation

CBC -> concentrator: up to 3 stubs (13 bits / stub) plus 1 triggered data bit = 40 bits / 25 nsec
⇒ 10 lines per chip at 160 Mbps (no room for any more data)

concentrator -> GBT

unsparsified L1 triggered data passed on to LP-GBT by concentrator

LP-GBT data payload 80 bits / 25 nsec (3.2 Gbps)

⇒ 16 CBC chips currently take $16 / 80 = 20\%$ of GBT data bandwidth

for increased L1 trigger rate?

unsparsified L1 triggered data volume will increase

e.g. 2, 4, 8,.. bits per 25 nsec BX period

CBC -> concentrator: current bandwidth already full at 1 bit / BX

⇒ sparsify L1 triggered data on CBC, reduce stub information,
or add another line (or 2) from CBC to concentrator

concentrator -> GBT

unsparsified L1 triggered data takes more and more of data bandwidth as trigger rate increases

40% @ 2 bits / BX per CBC

80% @ 4 bits / BX per CBC

not much room left for stub data

160% @ 8 bits / BX per CBC

have to sparsify L1 triggered data somewhere

CBC or concentrator?

I think my preferred choice would be concentrator

advantages to sparsifying at the concentrator level

... a few that occurred to me so far

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

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

if digital functionality turns out to be excessively power hungry can consider 65 nm for concentrator (share large volume production with LP-GT? – maybe prototyping also?)

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 + unsparified L1 triggered data - the way it has been envisaged up to now
- 2) unsparified L1 data only (L1 trigger rate up to ~570 kHz)
- 3) stub data + sparsified L1 triggered data
- 4) mixed mode- i.e. higher (e.g. ~300 kHz) L1 rate with unsparified 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?