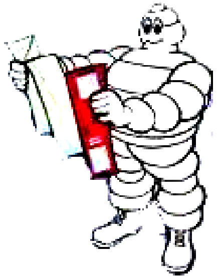
L2 Trigger Bits:
> 128 bits?

James T. Linnemann

Michigan State University

Paris Workshop

March 12, 1999



Guide Verte:
Un Peu d'Histoire

	Lum/10E30	Xsect(mb)	Phys kHz	Xing kHz	<N>	<Ncal>
Run I	15	47	705	286	2.5	2.5
Run II 396	80	55	4400	2525	1.7	2.2
Run II 132	200	55	11000	7576	1.5	5.5
Run II / Run I:						
Ratio 396	5.3	1.2	6.2	8.8	0.7	0.9
Ratio 132	13.3	1.2	15.6	26.5	0.6	2.2

Encore d'Histoire

	Lum/10E30	L1 Hz Out	L1 rej	L2 Hz Out	L2 Rej	L3 Hz Out	L3 Rej
Run I	15	800	881	150	5.3	3.5	43
Run II 396	80	6000	733	1000	6.0	20	50
Run II 132	200	6000	1833	1000	6.0	20	50
Run II / Run I:							
Ratio 396	5.3	7.5	0.8	6.7	1.1	5.7	1.2
Ratio 132	13.3	7.5	2.1	6.7	1.1	5.7	1.2

- Rejections required ≈ Run I
- But, moved rejection upstream (better L1)

Use for more physics channels, lower thresholds

this (and geometrical parallelism) ⟹ more L1 bits

L1trigger bits

- Collaboration accepted 64 bit design (1997)
  - run I was 32 bits, 28 used
    - some needed for “system” functions
- in 1997, moved 128 bit L2 hardware to L1 FW
  - allowed implementation of 128b L2 as 1:1 with L1
    - side effect: no prescale in L2 needed

New L1 tools

- Central Tracking trigger
- Central and Forward Preshower trigger

  request: quadrant matching with calorimeter

  motivation: specific physics channel

L1 bit OR’ing

- L1 will not get additional OR-ing capability as a general L1FW feature until start of run
  - some available in mu, cft, ps trigger managers
  - specs frozen for initial delivery
  - *further changes will make L1CAL late*
- OR’ing eventually possible:
  - localized by trigger terms
  - aimed at geographical combinations (quadrants)
  - NOT mu.OR.e
    - lose information to guide L2 processing

L2: Run I

- 1 to 1 with L1bits
- only 16 bits (not all 32)
- time budget was really deadtime budget

  generated direct deadtime waiting for decision

  - muon: ~10 μsec, 500 Hz, .5% deadtime
  - cal: ~110 μsec, 100 Hz, 1.1% deadtime
- Restricted to muons, cal

  much higher rejection in muons than cal

L3: Run I

- 128 bits

  90 actually used (started with 64 constraint)

  expansion by 3 from L2

- 200 Hz/48 nodes = 4 Hz, 250ms budget

  after queueing, more like 180ms

  for 20 MHz processor, 4M instructions

  processor had access to all detectors, full readout

  Run II: 60-120 M instructions (X15 to 30 Run I)

  – X 25 speed X 3 processors/node X (1/5) time X superscalar

L2 Trigger: Baseline design

- 10 KHz L1 out to 1 KHz L2 out
  - 128 L2 decision bits, 1:1 with L1
    - – As of July 1997 or earlier
    - – earlier, had accepted 64 bits in L1, 128 in L2
  - few % deadtime
- Global Processor selects events
  - threshold for object
  - matching objects from different detectors
  - cuts on quality (mostly in preprocessors)
  - kinematic variables (but Zv=0)
- Objects from single-detector preprocessors

L2 processing power (50 μs)

- Alphas: 25 to say 50K instructions per card
- SLIC’s: 100K instructions per card
- Collectively (ex STT): (ignoring idle cycles)

Cal	150K	~RI L1 towers
Mu	1.6M	raw data (but balancing!)
PS	150K	L1 clusters
CTT	50K	L1 tracks+PS/ISO tags
Global	50K	output of above

total: .5-2M instructions: ~ RI L3 / 4 ~ RII L3 / 100

- – but: processors don’t see all data
- – global sees only objects found, not all data

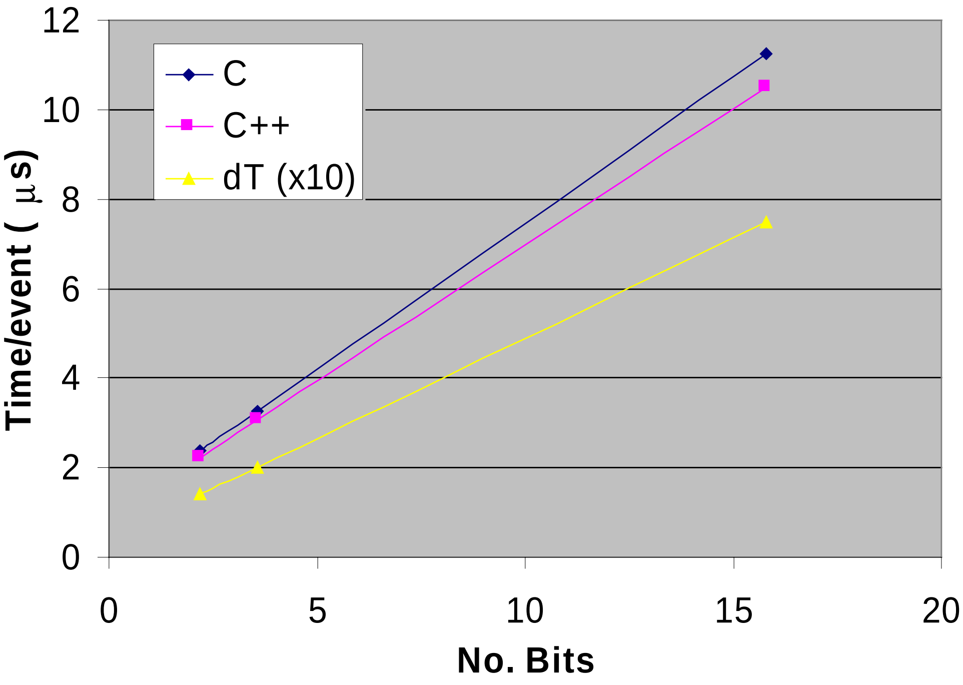
L2 Global Overhead: search 128 bits to process

- Background “barely” triggers--unless trigger list forces correlations
- typical L3 RI: 2-3 L1 bits
- ~3 L3 RI scripts/bit
- expect 5-10 L2 bits

  4-8μsec, 10% overhead

  - 1/2 search, 1/2 call

  with ~1-3 tools/script
- Bunch similar bits!
- Branching means a new layer of loops, searches
- X2 effect in L3 frame



What does that leave time for?

- Additional 2μsec for interrupt routine
- present estimate for Global: 2 μsec/tool

  ~1000 instructions

  – makes assumptions about complexity of cut/match…
- so perhaps time for that 4 such tools/script

  or X 2 scripts with same overhead

  or X 2 slower than this--need code, simulation
- RII L2: more tools/script than RI L3 <1.5>?

  lower L2 rejection $\Longrightarrow$ deeper into script before fail

  more channels $\Longrightarrow$ more complex scripts needed

  – RII “tool” = {algorithm, parameters}

Are budgets too conservative? Not with current knowledge

- Safety factor: design for 10 KHz
  - 6KHz expected: limited by SVXII deadtime
  - nominal budget is 100 μs instead of 160 μs
- Realism: queueing (16 L2 decision buffers)
  - nominal budget from 100 μs to 80-90 μs
- Long timing tails assumed (rms = 2 x mean)
  - reduces nominal to 50-60 μs from 80-90 μs
    - allows for nonlinear time tails (combinatorics)
    - price we pay in lieu of a VERY good simulation
  - buffers fill up waiting for a slow event
    - Front End buffer design: must answer in order!

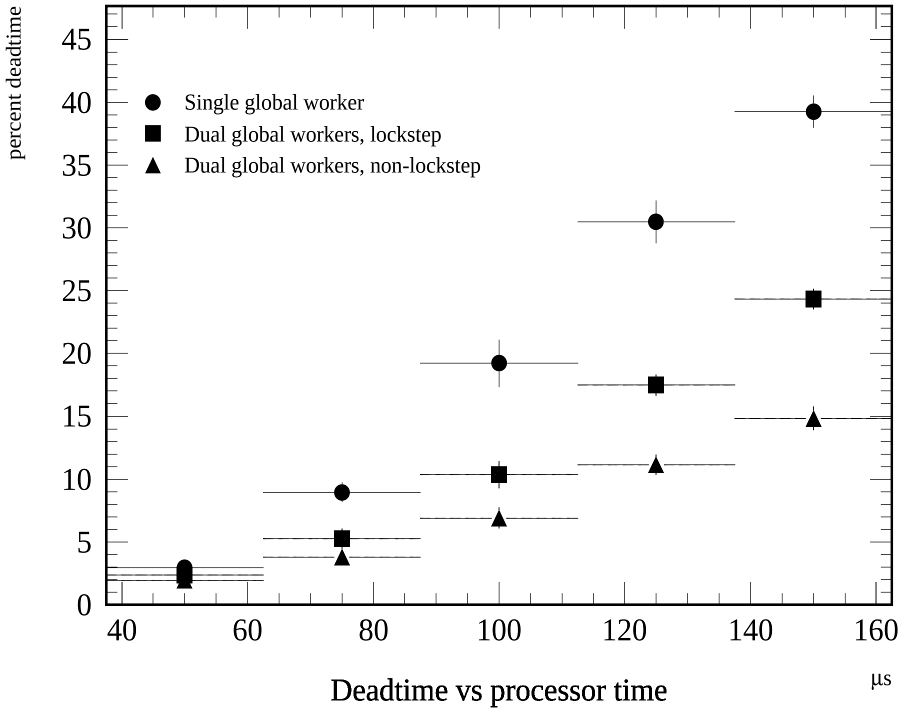
Add another processor?
Maybe, but not linear gains

- 2 processors in parallel? Hard to balance
- 2 processors, different events?

  Alternating: maybe 20%

  non-lockstep: 40%
- Inefficiency when handling slow event: tails are deadly

  answers in order!



Why use another trigger bit?

- Without separating from another channel, cannot meet output bandwidth requirements
  - new physics
  - low threshold possible only with added conditions
    - – and incompatible between physics channels
- condition requires multiple bits to express
  - geometrical parallelism (quadrant OR's)
- distinct processing needed at next level
- measure background or efficiency
  - AND need continuously monitored

What if we run out of bits?

- run sample with partial luminosity

  in lumps rather than prescaled

- combine with another bit

  - lose efficiency?

  + less channels to characterize

- find a way to express in fewer bits

  hardware upgrade needed in L1

  – geometrical parallelism

- Raise thresholds (lose efficiency)
- abandon some kind of physics

L2 view

- Q: how many bits can you manage
- No output bandwidth limits to justify branching--yet (simulation needed)
  - L3 makes better decisions than L2 (esp. Cal)
- First priority: make L1,L2 existing design work
  - code and simulation to evaluate
  - we start at low luminosity
  - trigger list less tuned at start?
- Design will allow upgrade to branching
  - more overhead, more complexity to debug
  - would likely pick new max # bits...