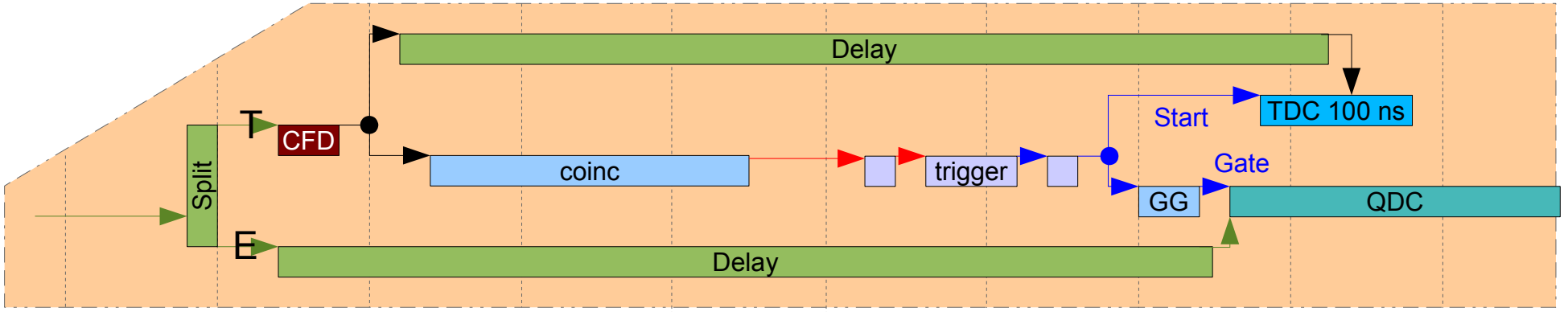
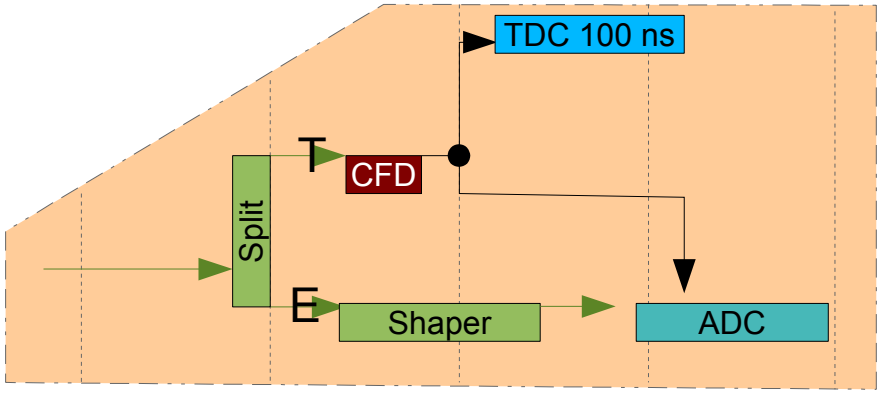
Triggered



Trigger-less = free-running = ...



Each channel self-triggers

Data timestamped vs global clock

Coincidences by
software trigger → events

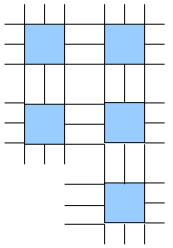
Triggered & free-running, pros, cons, coexistence...

The free-running usual complaints

Huge data-rates
(depends on necessary thresholds)

Local 'pile-up'
(real signal missed due to preceeding hit)
(issue shared with *any* triggered common-stop-system)

Massive fan-outs/fan-ins for local neighbour-firing
(any triggered system does a large fan-in too...)



Hardware trigger issues

Noisy triggers?
(global or / multiplicity of many channels)

Long trigger latency + master start distribution
(delays before QDCs run out of delay)

Read-out overhead per event?
(slowest system rules)
(multi-event readout - everywhere?)

Detectors - information needed

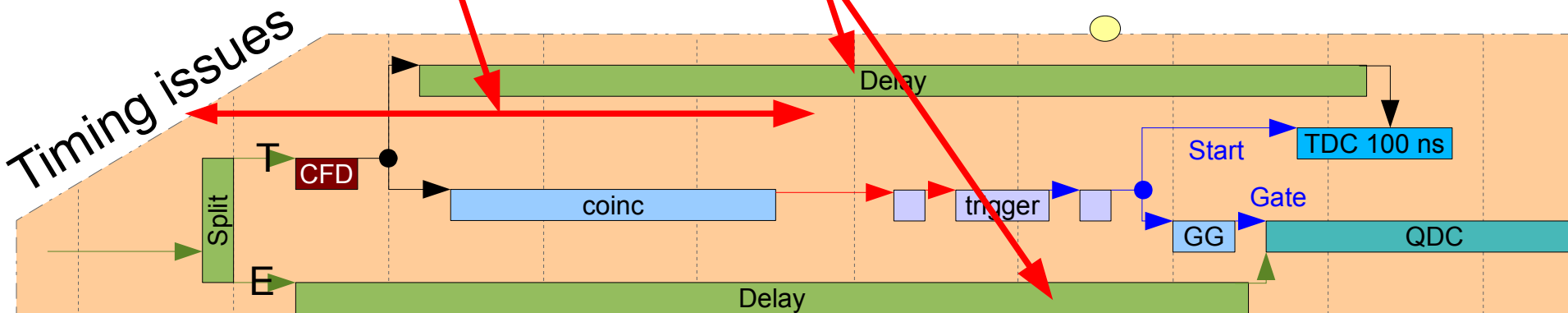
What data-rates will (can) you deliver?
(worst-case!) (free-running?)

Please, tell us!

1 us

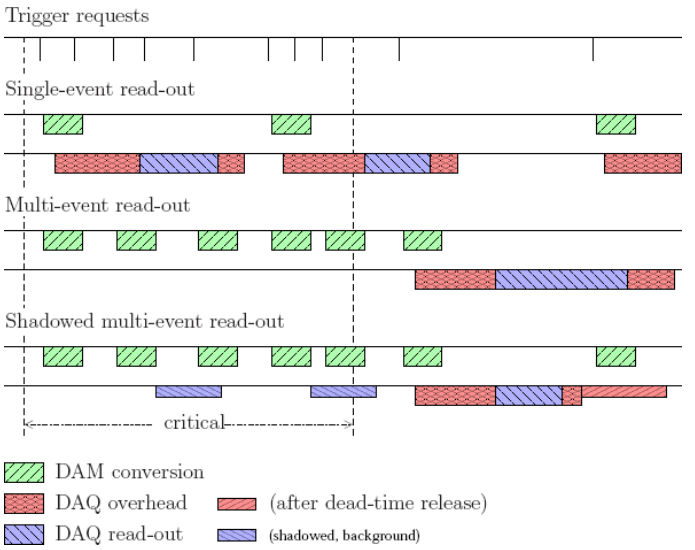
What trigger latency do you cause?

How much delay must everyone else support?



Detectors – information II

Is your triggered system multi-event capable?
(in the background)



What calibration procedures (triggers) do you need?
(CLOCK & TCAL enough?)
(Analysis experience strongly suggests continuous calibrations!)

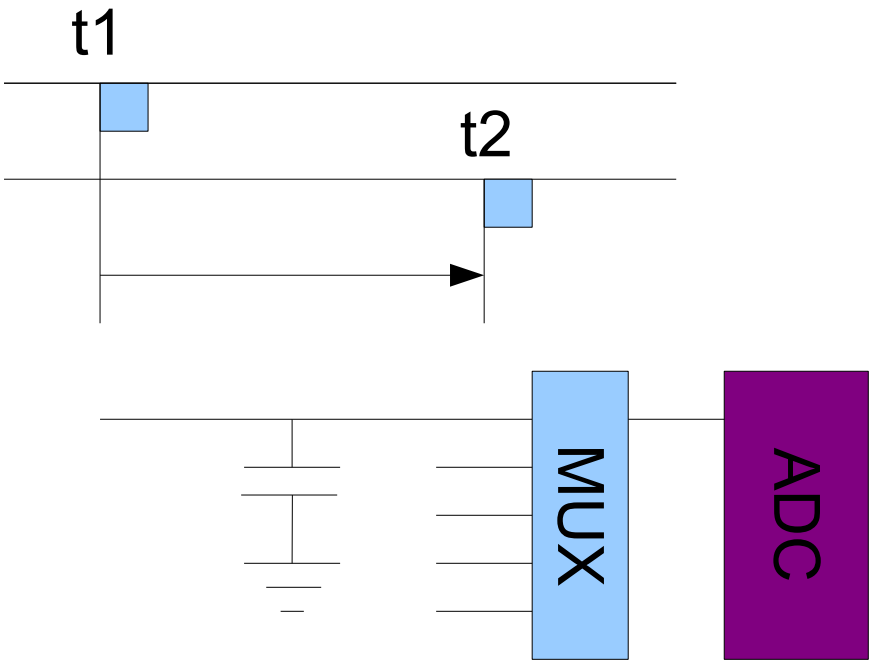
Tacquilas

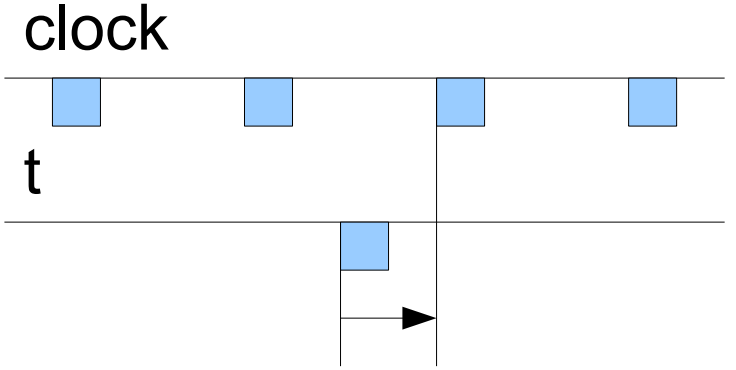
TDC = TAC + ADC

Classic:

- Charge capacitor between t1 and t2.
- Output pulse: height ~ time interval

- Multiplex → ADC: digitise pulse height





Alternative style:

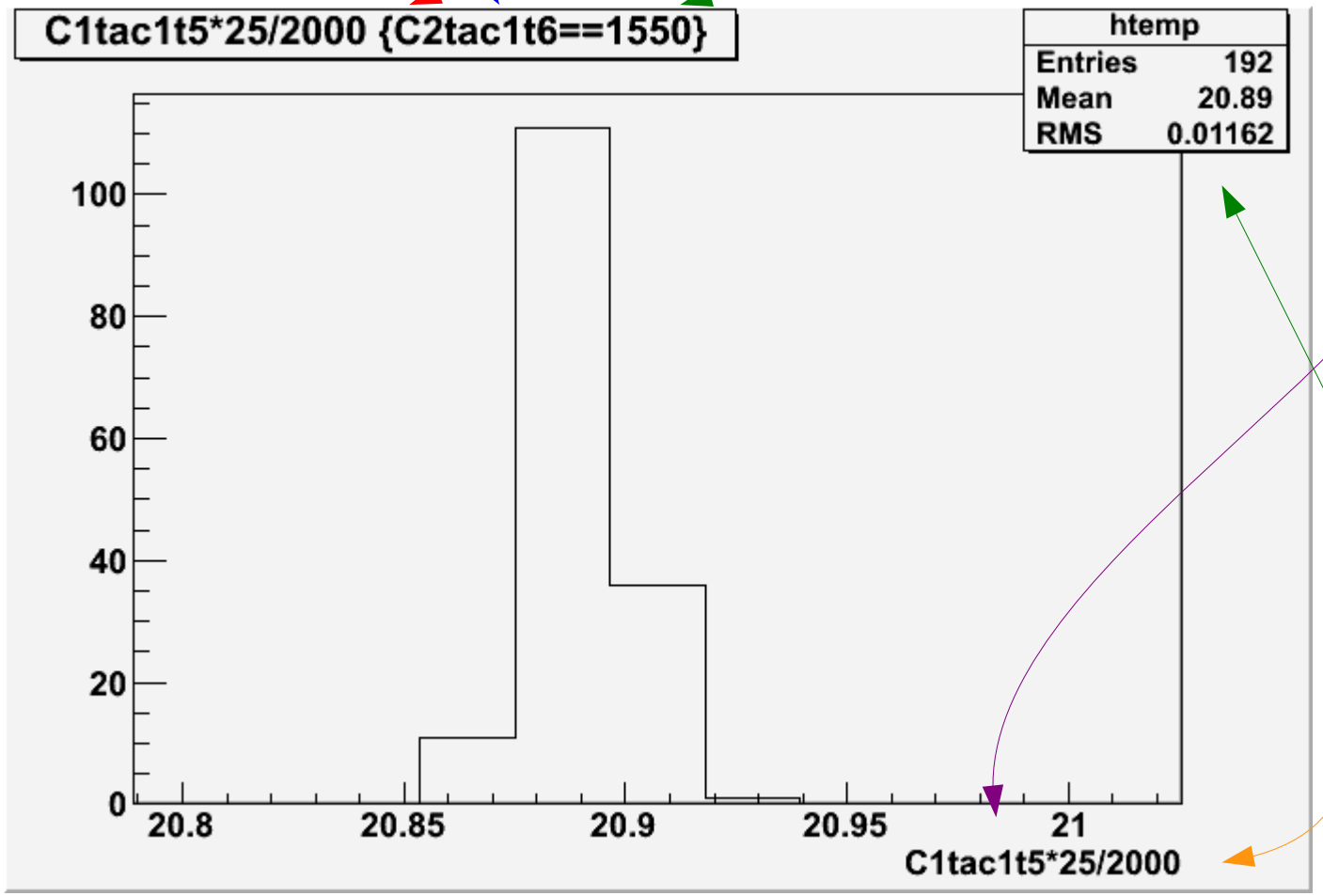
+ Measure time to next clock:

  - With classic TDC
  - With abused FPGA

+ Store clock cycle number

Tacquila – crate vs. crate (pulser)

Choosing one channel in crate 2, specific time



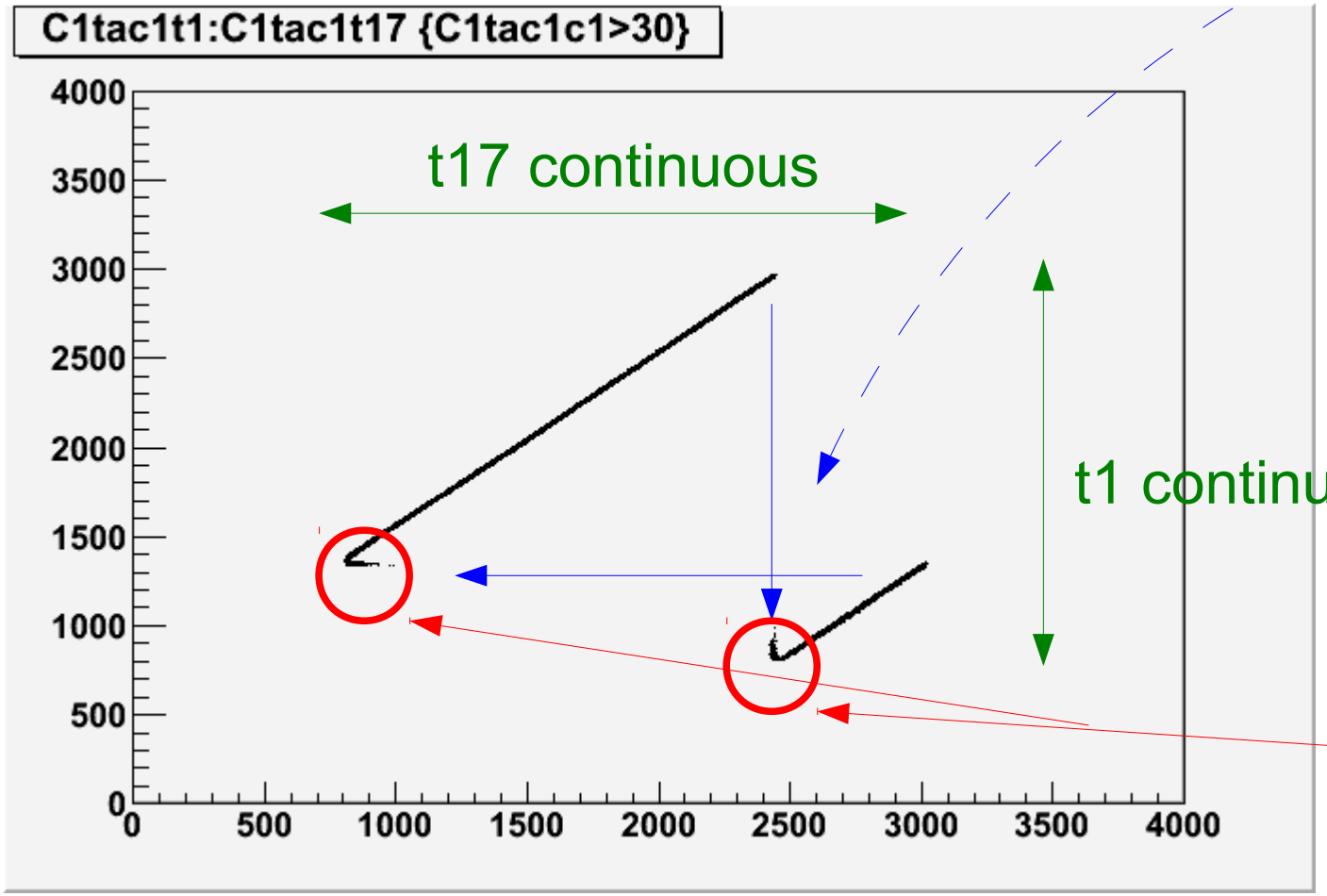
Plot channel in other crate

(in ~ ns)

Combined σ_{t5} σ_{t6}

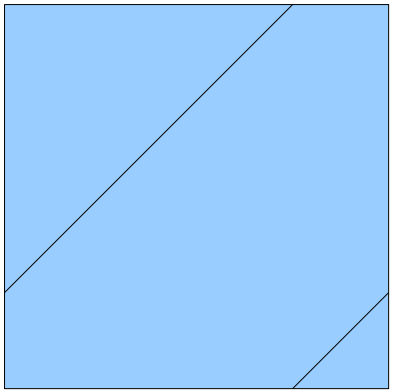
Tacquila ambiguity I

Channel TAC vs channel TAC (any will do)
(here: ch vs. 17th ('master start'))



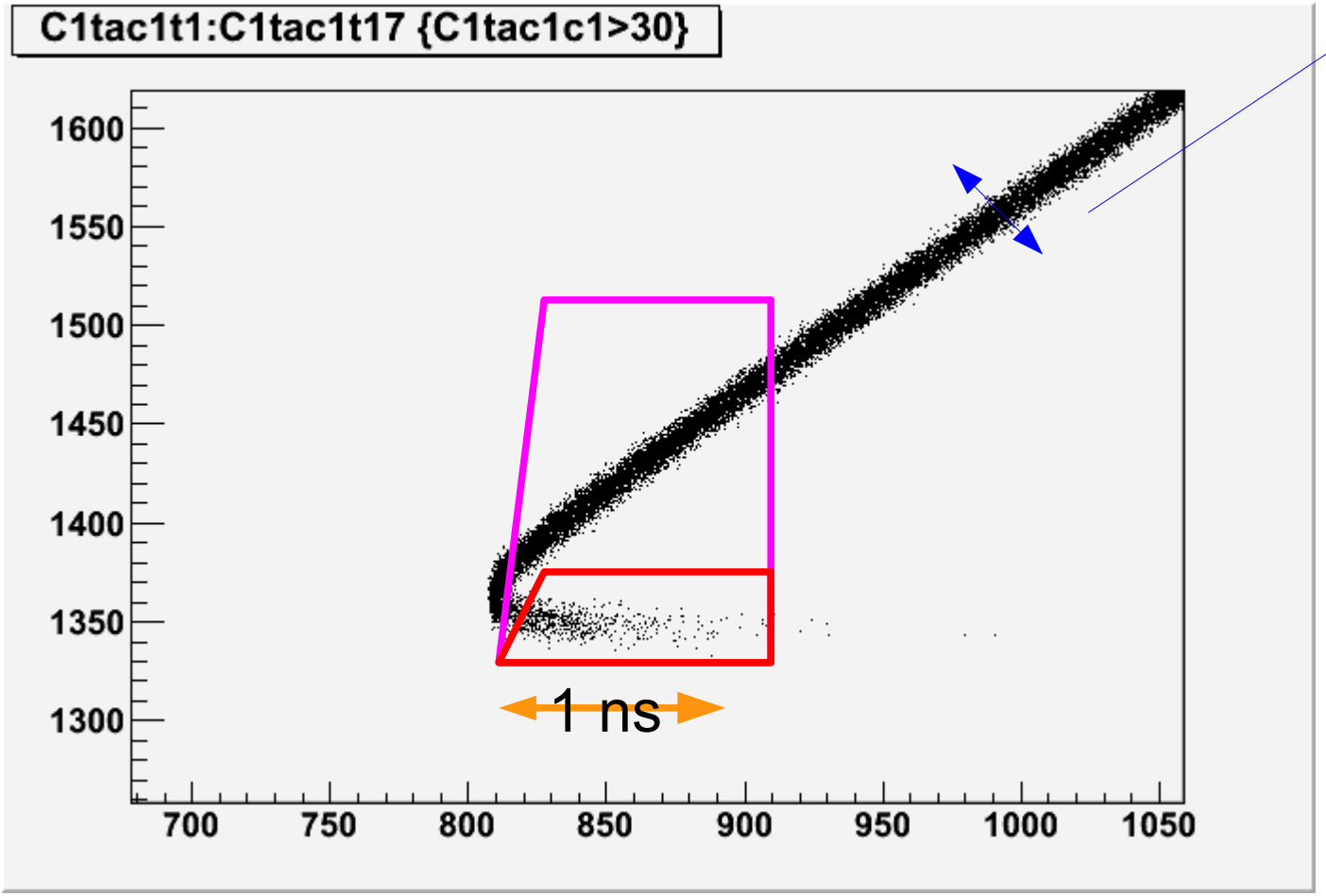
Discontinuity ok, due to clock interval wrap

Ideally:



'Hooks' = ambiguity

Tacquila ambiguity II



Width likely due to
pulser jitter vs. master start
→ do not care here

Affected range:
~ 4 % (1 ns)

Of those:
~ 10 % in 'bad' branch

→ ~4 ‰ of total
(0 – 1 ns wrong)

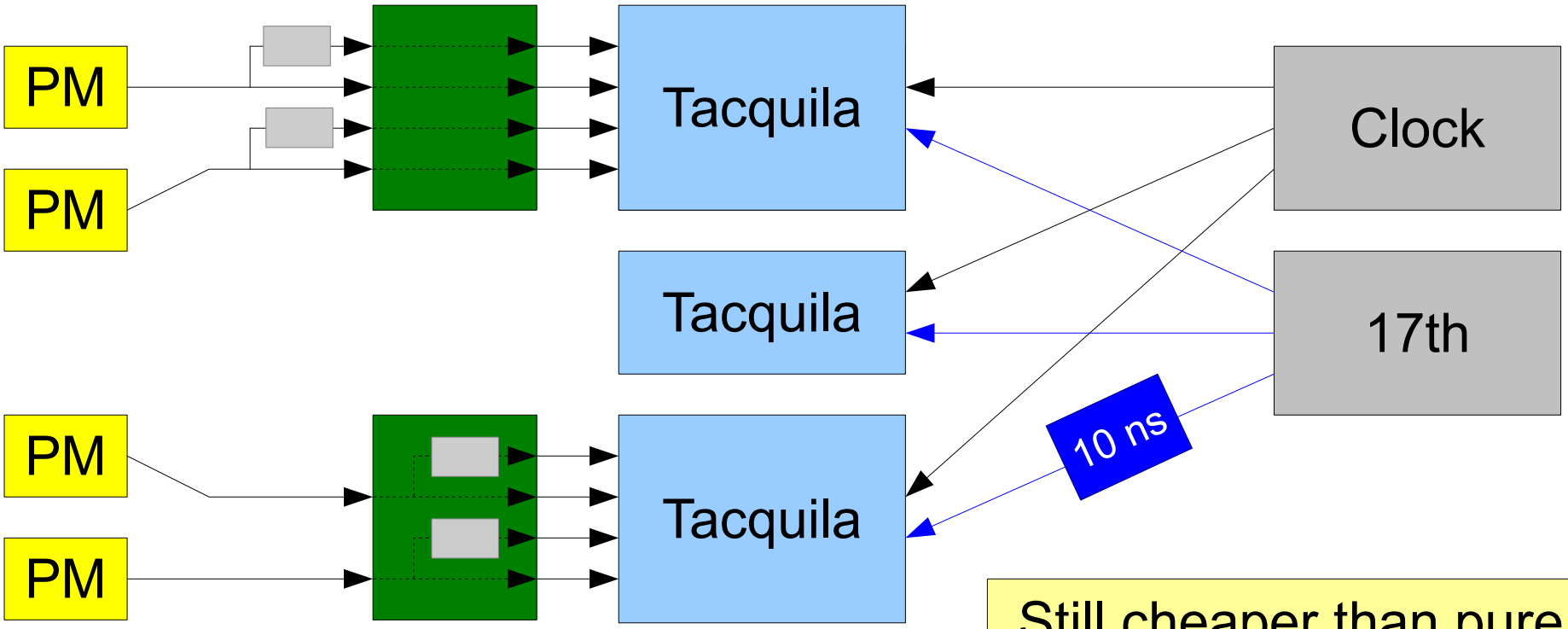
Ch + 17 needed →
~ 1 % of hits affected

Tacquila ambiguity workaround?

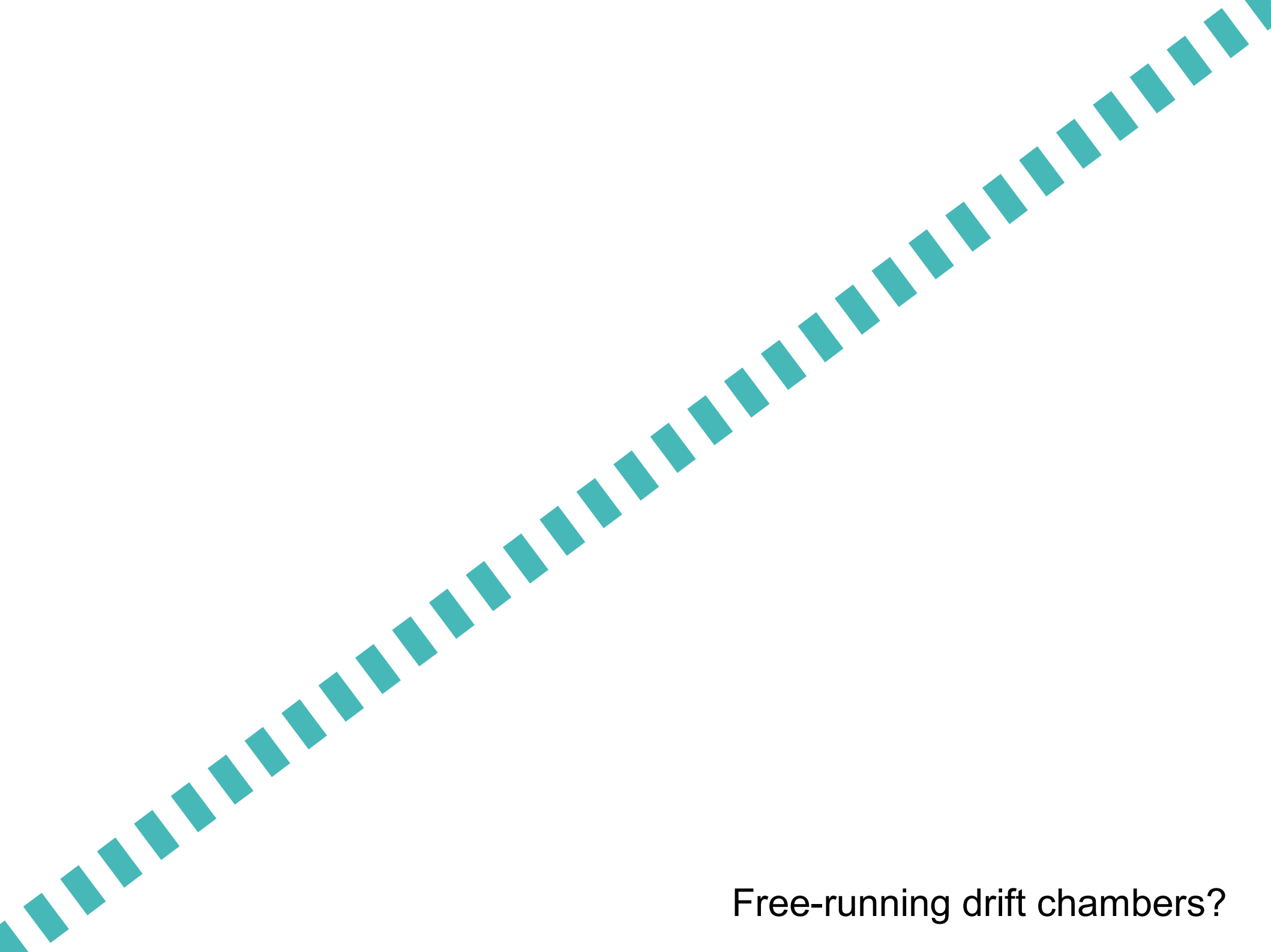
Workaround (for e.g. TOF-walls??):

Each detector channel to 2 tacquila channels, with relative delay

Channel 17 (master start) is measured on many boards, relative to same clock → fixed by using different delays (at least one)



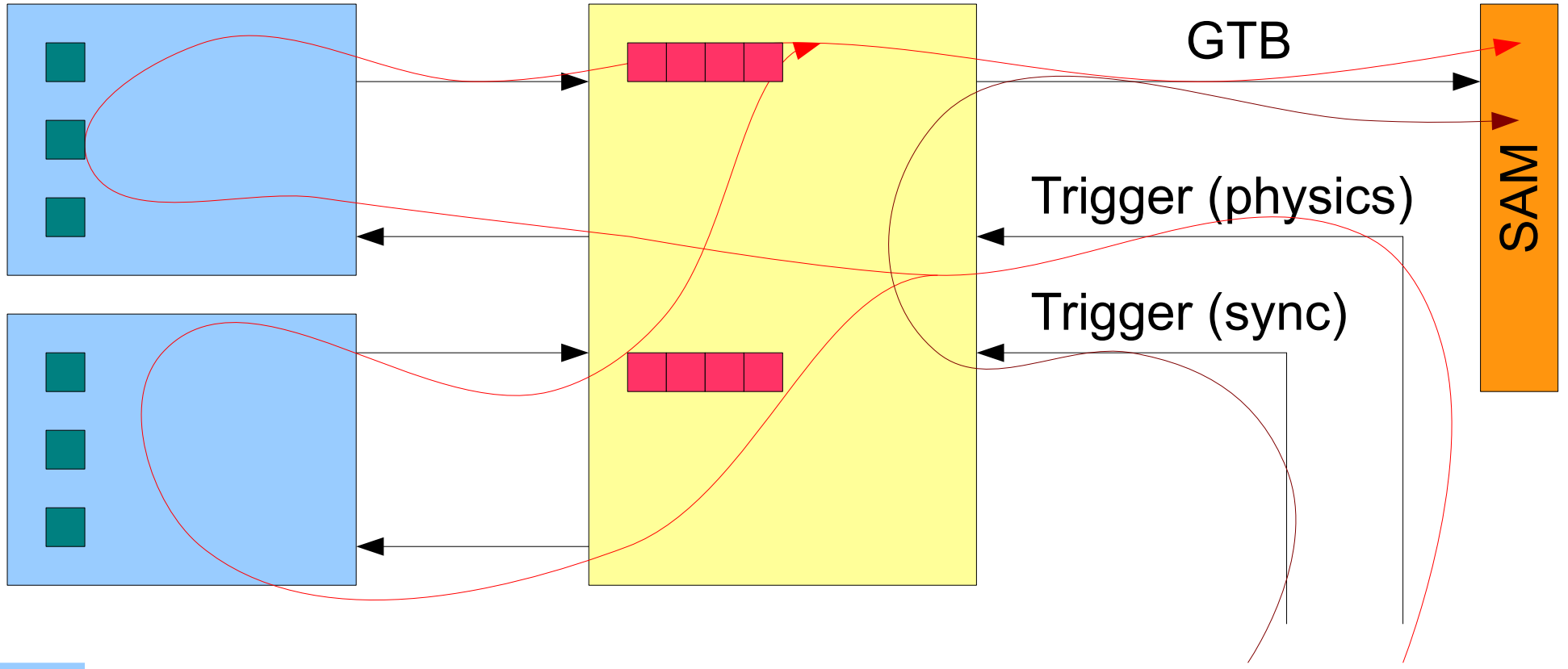
Still cheaper than pure VME



Free-running drift chambers?

Proton Drift Chambers

Currently: no trigger delivery, requires trigger



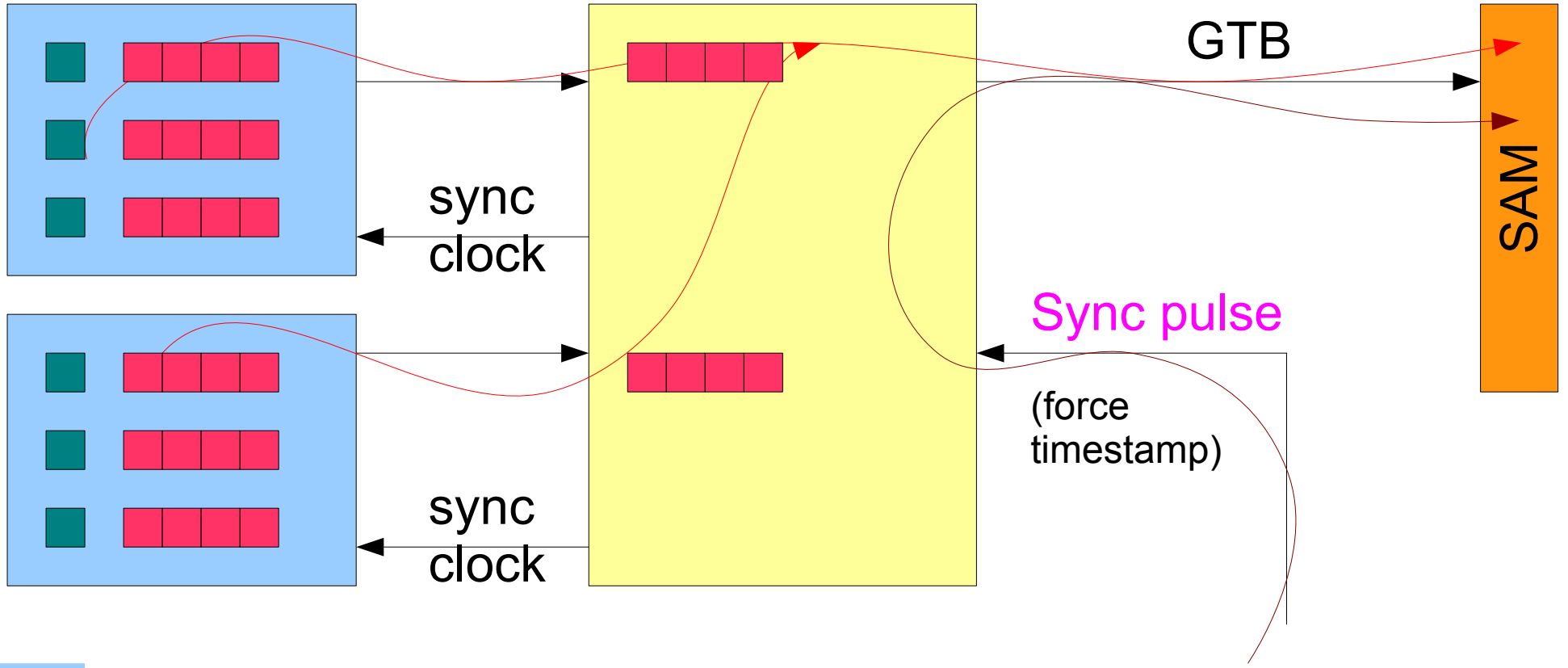
FE cards free-running into local circular buffer (256 x 2.5 ns steps)

Physics trigger causes encoding (and buffering?), send to CCB

CCB buffers and transfers data over GTB to SAM (→ VME)

Free-running Drift Chambers

With existing hardware: free-running mode should be possible

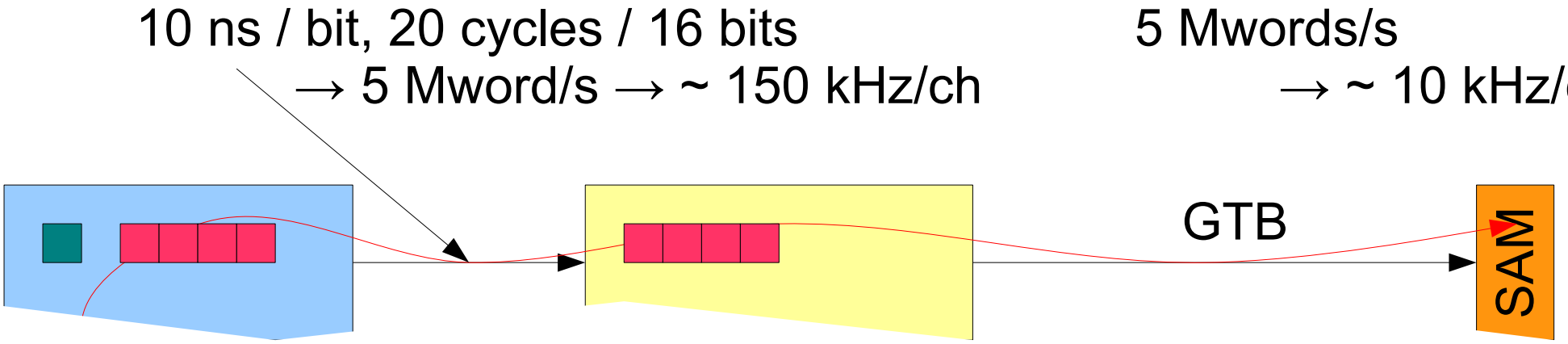


FE cards free-running, time-stamp all edges (protect against short pulses)

FE cards operate with few-bit clock, sync from CCB

Data continously flowing from FE buffers and CCB buffers over GTB to SAM (→ VME). CCB inserts high bit clock epochs and global sync.

Drift Chamber Rates?



FE delivers channel (4 bits) + edge (1 bit) + data lost (1 bit) + timestamp (16 bits) → ~ 22 bits

covers 1 entry transmission (2+5 bits) x 16 ch (4 bits) x 8 entry buffers (3 bits) + wrap (2 bits)

CCB adds FE card number (4 bits) + fill with more time bits to 32 bits

CCB inserts epoch markers (higher ~30 bit time stamps)

DAQ policy discussion

DAQ integration policy

Thorough integrity checks

- Event-wise checks, e.g.:
  - Event counters
  - Trigger type
  - Data packaging

- All mishaps reported (centrally)

(Expect DAQ integration team to add tests.)
(Expect 'evil' trigger conditions.)

Error reports = system broken

Any reported errors to be:

- Investigated
- Explained
- Repaired

(Broken sub-systems are ***removed*** from the DAQ.)

DAQ stop/start not for slow-control

No slow-control operation may require a (user) DAQ stop/start cycle.

Even if the read-out data-bus is used for communication.

(Reason: independent systems in the dead-time domain shall be unaffected.)

The DAQ always runs II

The read-out may not (on non-fatal errors):

- Crash
- Exit
- Do infinite looping, etc...

Instead, re-init until error-condition resolved (by user).

(Ex. non-fatal: missing/replugging gate cables.)

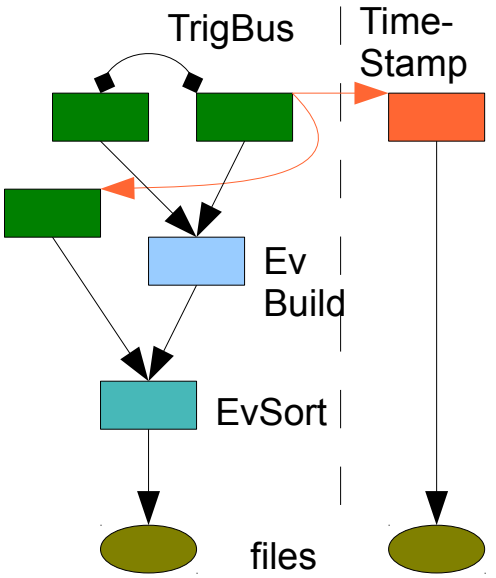
(Broken VME-bus is fatal.)

Working unpack procedures

Unpack prodedures:
Allow integrity checking
Usable online
(i.e. with network data source)

Enforcement

Non-compliant systems are ***removed***.
(from the dead-time / data-collection domain).



Compliance rule-of-thumb:

- 1 month *before* test experiment
- 2 months *before* production run

Removed systems become time-stamped,
with separate data-file-production.