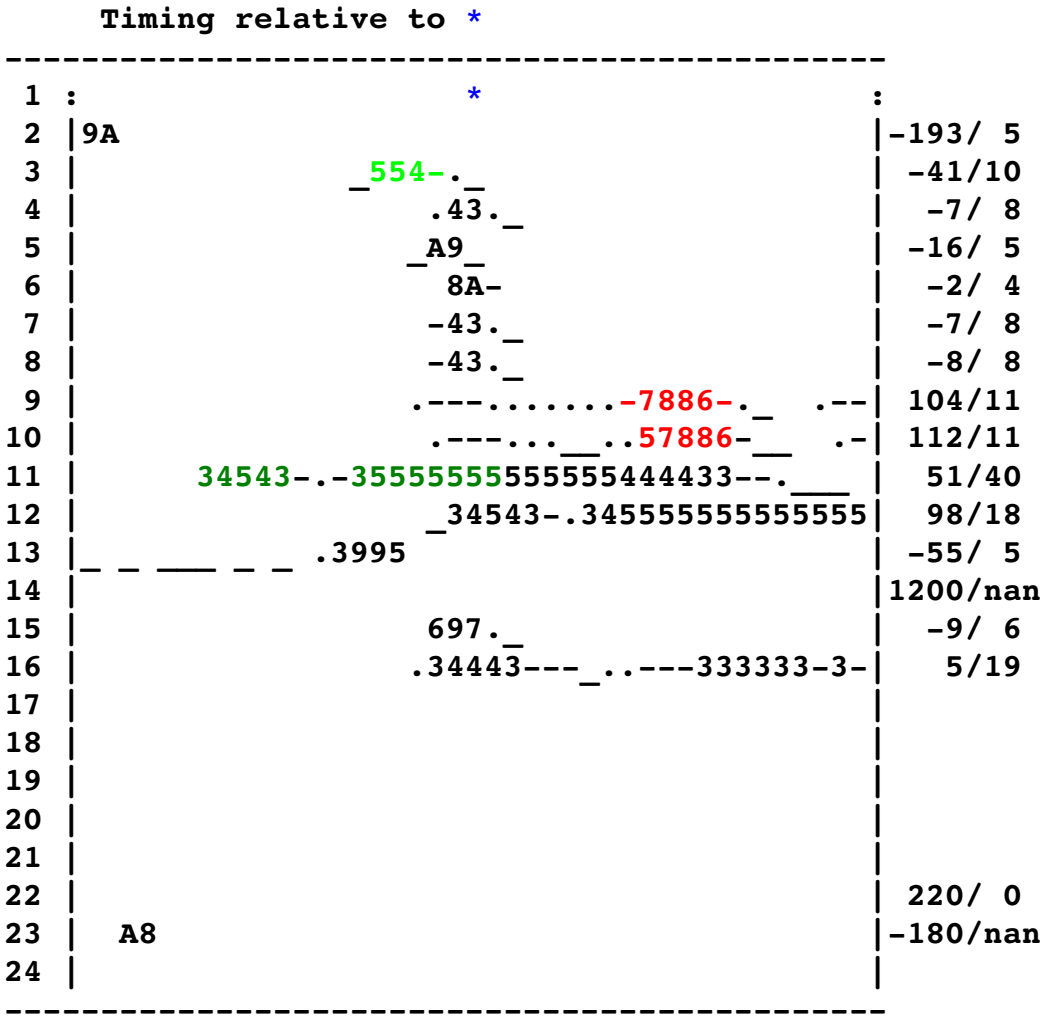
TRLO II – flexible FPGA trigger control



Håkan T. Johansson, Chalmers, Göteborg

With Ana Henriques, Lisbon

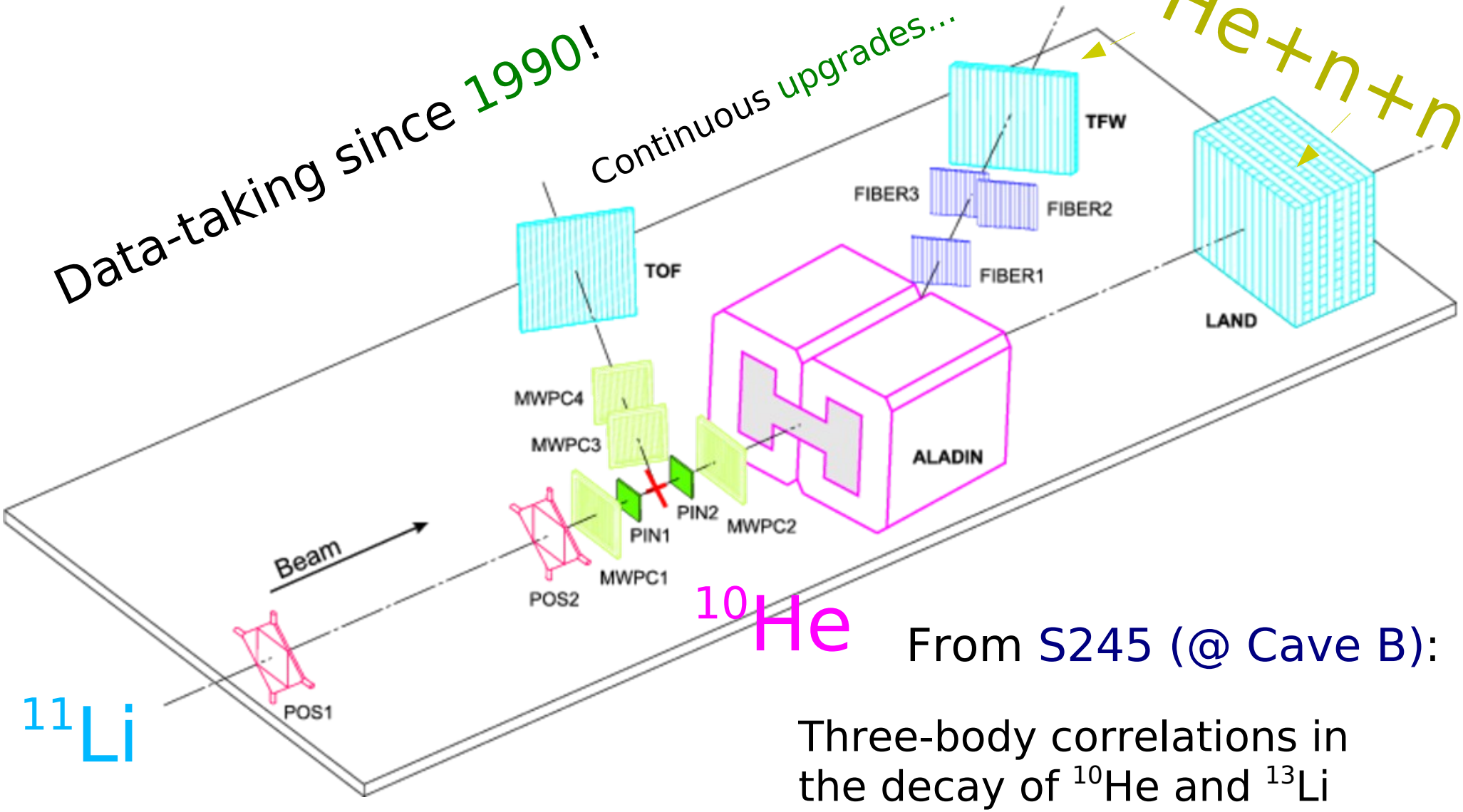
Uppsala, November 2010

ALADiN-LAND setup → R^{3}B

Data-taking since 1990!

Continuous upgrades...

^{8}He+n+n



^{10}He

^{11}Li

From S245 (@ Cave B):

Three-body correlations in the decay of ^{10}He and ^{13}Li

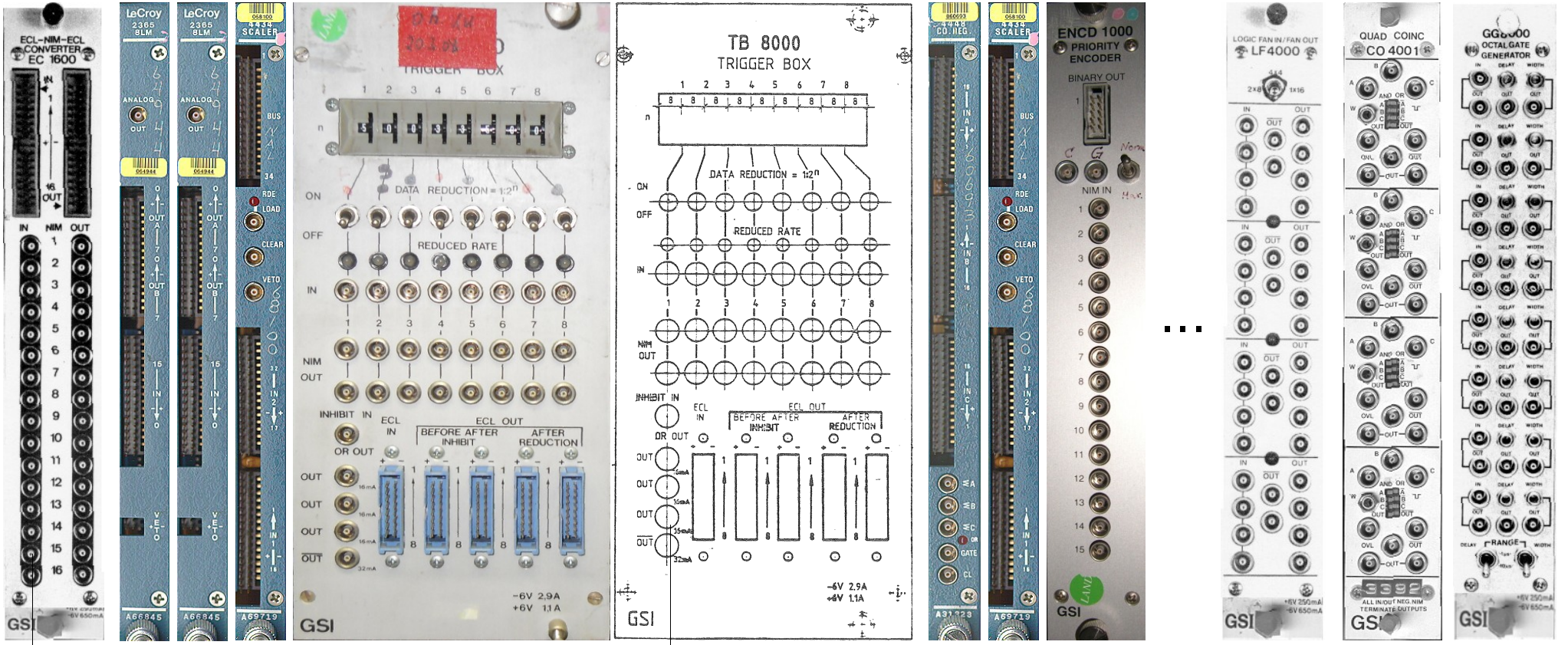
H.T. Johansson, Yu. Aksyutina, et. al.
Nuclear Physics A, Vol 847 (2010) pp. 66-88

The trigger has to be *fast*...

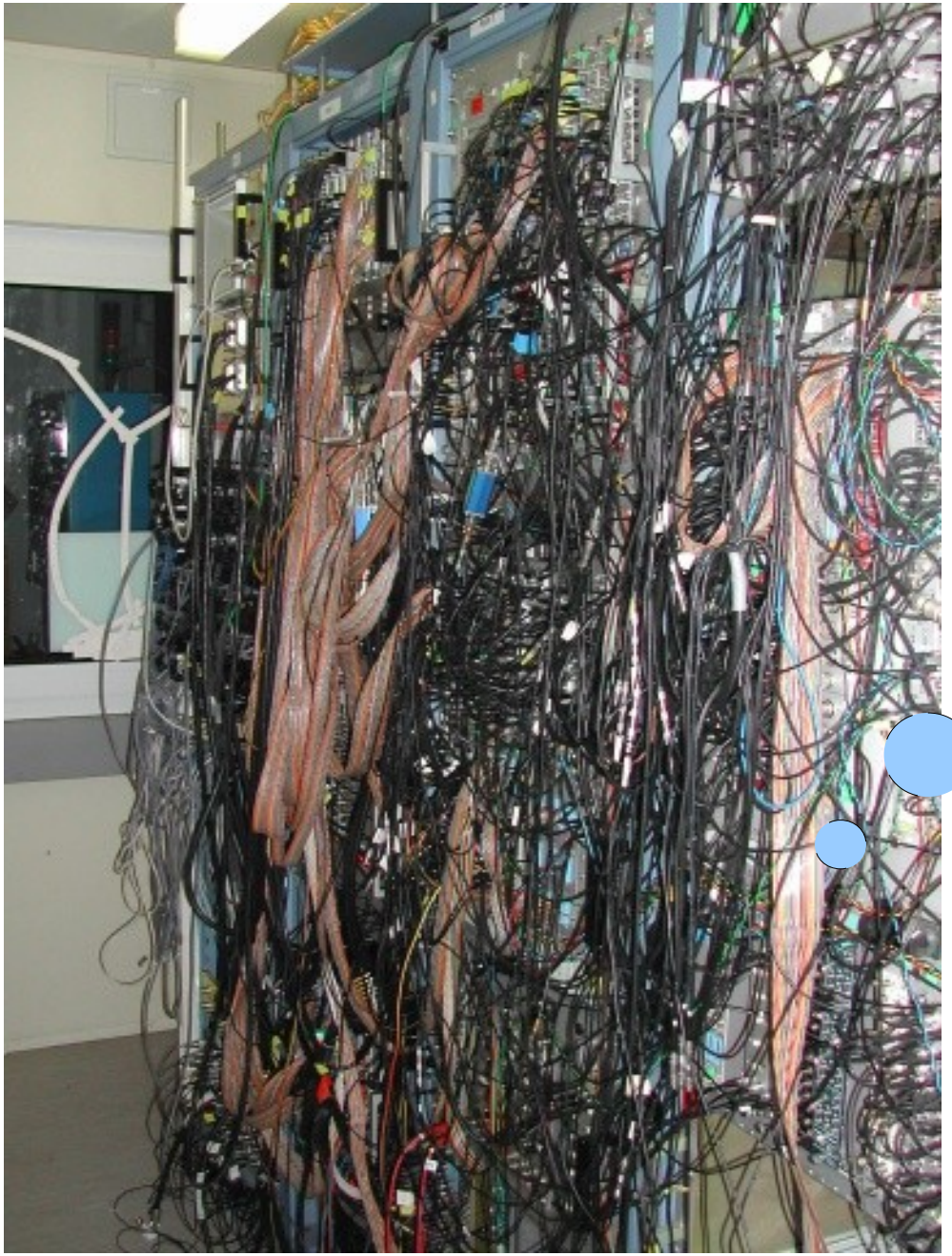
Detector signals
Logic matrix
→ coincidences
Scaler

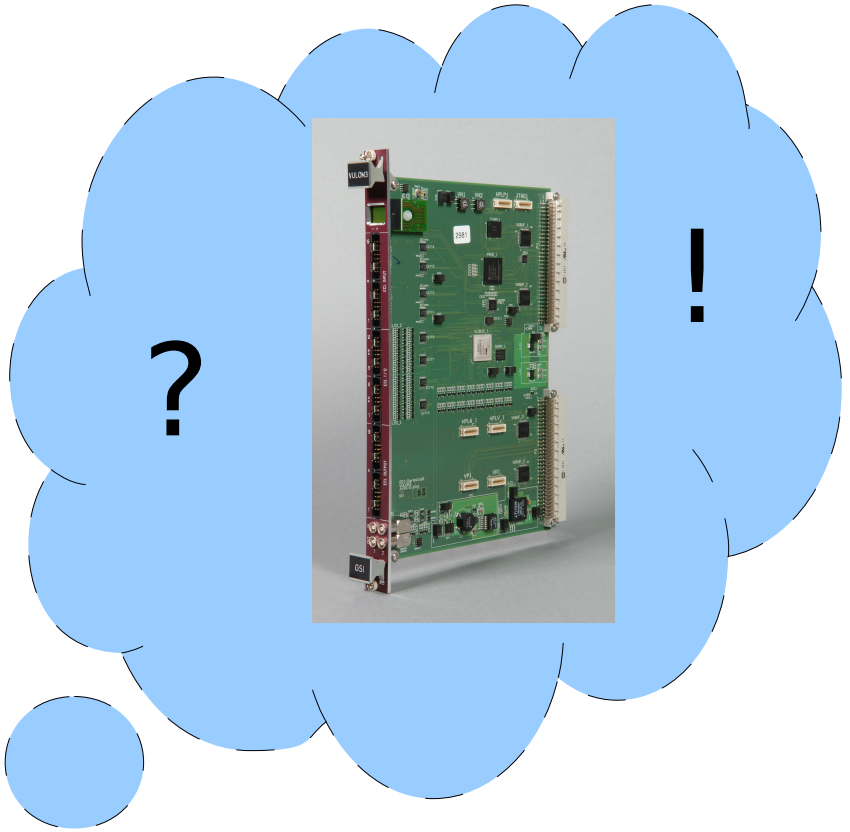
Trigger boxes:
deadtime veto
downscale

Pattern unit
Scaler
Priority encoder



Master start in 45 ns





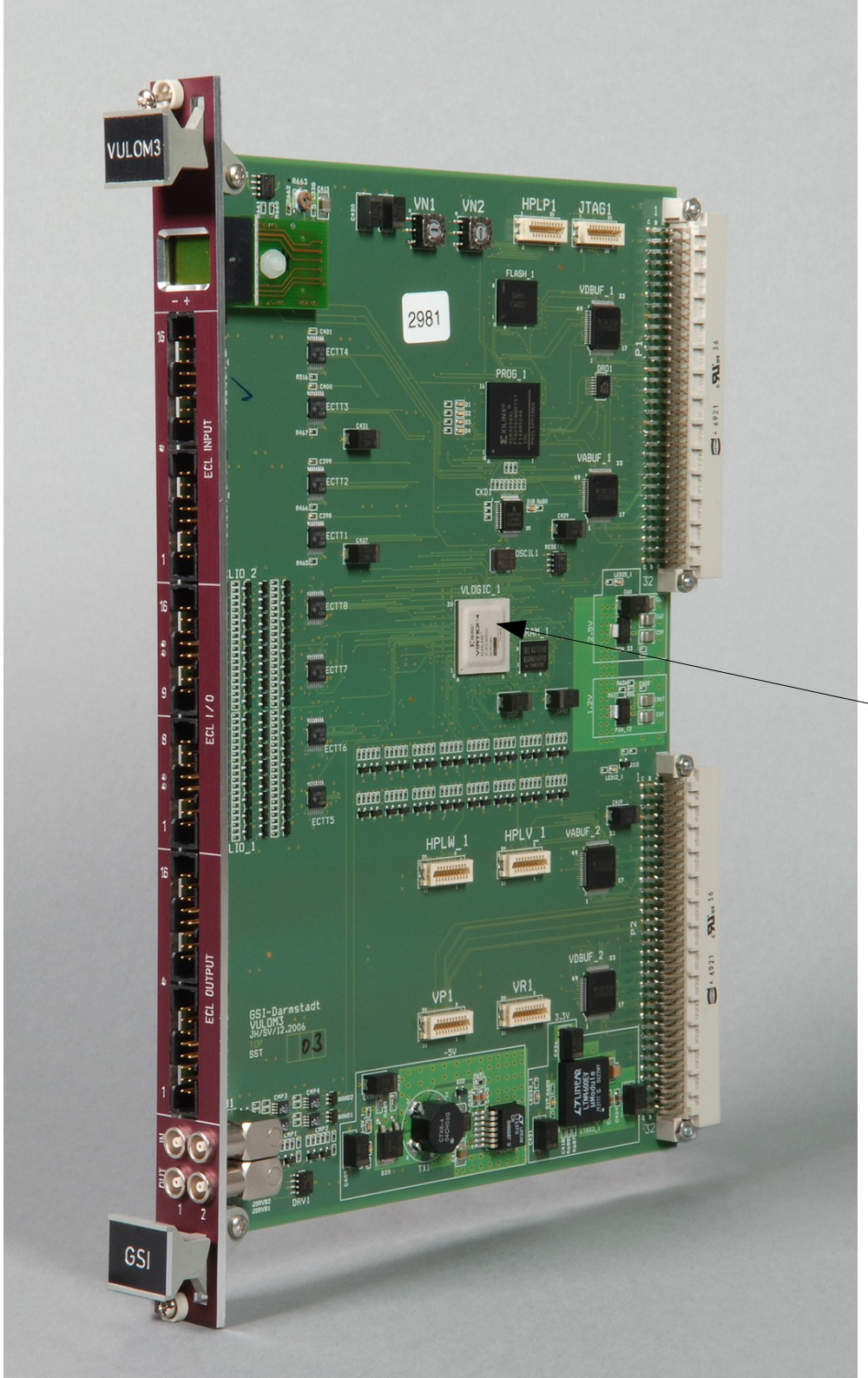
VULOM
(VME universal logic module)

by J. Hoffmann, GSI

Original TRLO firmware
by J. Frühauf, GSI

Inputs

Outputs



VME

FPGA

Fast-path: detector signals → master start

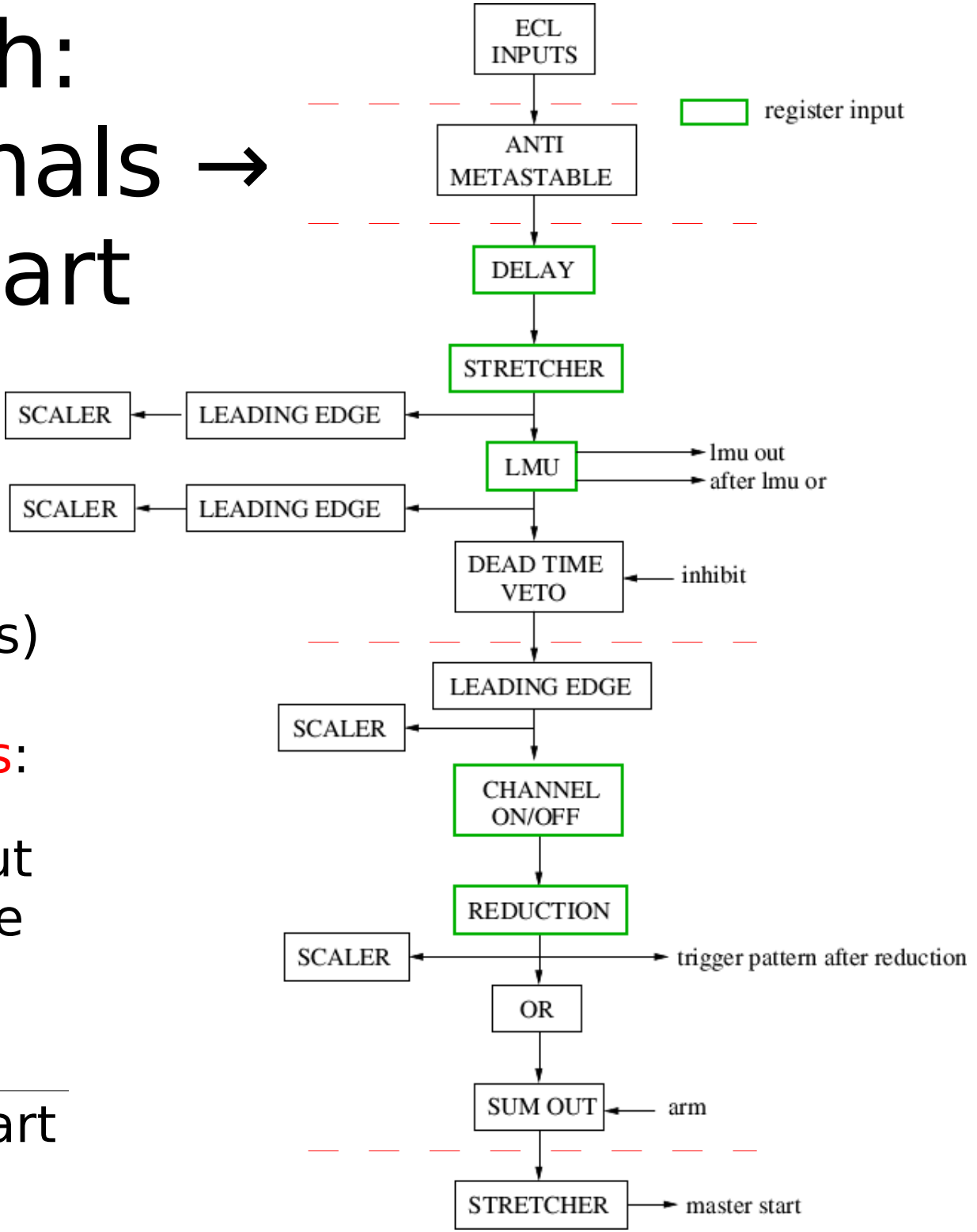
Timing-critical trigger decision

All logics: 100 MHz (10 ns)

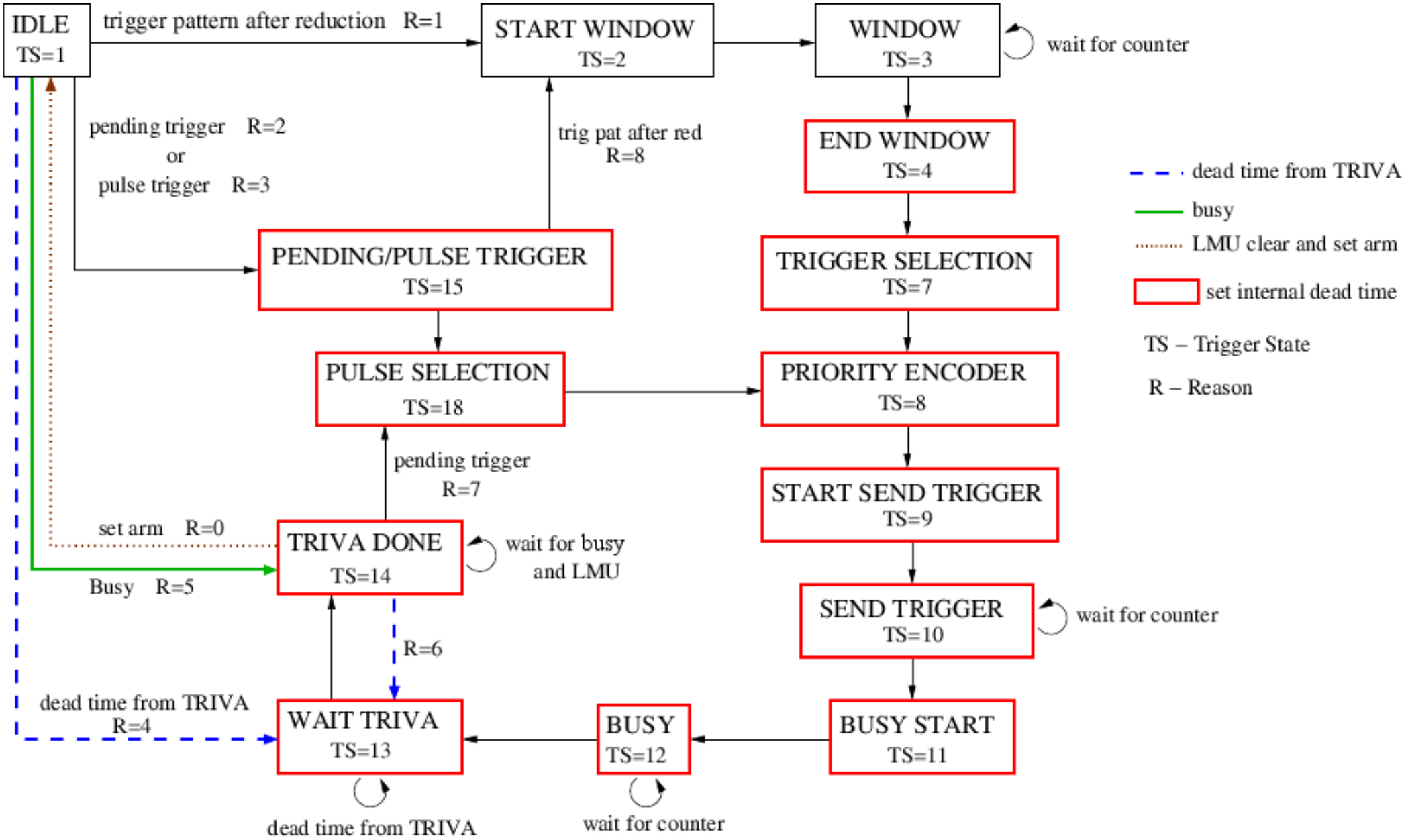
Decision in 2 clock cycles:

- 18 ns: module in + out
- 5 ns: anti-metastable
- 20 ns: decision
- 10 ns: clock jitter

45-55 ns: in → master start



Trigger state machine



Stable VME interface definition

VME registers as C structure, with named entries, generated by compilation

Version number is MD5 of full VHDL code

Named constants

Setup registers in block RAM for readback.

Aggressive checksumming of output data.

```
#define TRLO_MD5SUM_STAMP           0xccb60dee

// Constants for 'direct_mode':

#define TRLO_DIRECT_MODE_LOGIC              0x0
#define TRLO_DIRECT_MODE_DIRECT             0x1
#define TRLO_DIRECT_MODE_LOGIC_OR_DIRECT    0x2

typedef struct trlo_output_map_t
{
  /*     0 0x0000 */ uint32_t version_md5sum;
  /*     1 0x0004 */ uint32_t compile_time;
  /*     2 0x0008 */ uint32_t timing_tick;
  /*     3 0x000c */ uint32_t deadtime_tick;

typedef struct trlo_setup_map_t
{
  /*     0 0x2000 */ uint32_t mux[122];
  /*   122 0x21e8 */ uint32_t direct_mux[26];
  /*   148 0x2250 */ uint32_t direct_mode[26];
  /*   174 0x22b8 */ uint32_t direct_or[3];
  /*   177 0x22c4 */ uint32_t scaler_mode[8];

// MUX src indices:

#define TRLO_MUX_SRC_ECL_IN(i)         ( 0+(i))
#define TRLO_MUX_SRC_ECL_IO_IN(i)      (16+(i))
#define TRLO_MUX_SRC_LEMO_IN(i)        (24+(i))
#define TRLO_MUX_SRC_WIRED_ZERO        (32)
```

General logic

- Pulsers (programmable frequency).
- PRNG (pseudo-random sequence).
- LMU (not the same as in fast-path).
- Downscale.
- Delay and stretch (a.k.a. gate-and-delay).
- Edge-to-gate conversion (e.g. spill mimic).
- Fan-in (masked all-or).
- Coincidence.

Monitoring

- Scalers. Latched 32-bit with selectable input.

  Hybrid flip-flop / block RAM -> few resources. (25 ff/ch)

- Timer latches (latch global timer on a signal edge).

- Multi-entry buffer for the timer-latches.

  Block RAM fifo.

- Self-triggering soft-scope.

  Block RAM circular recording. Block RAM multi-trace fifo.

- Input pattern latch.

- Front-panel LEDs.

- Front-panel display.

Multiplex everything!

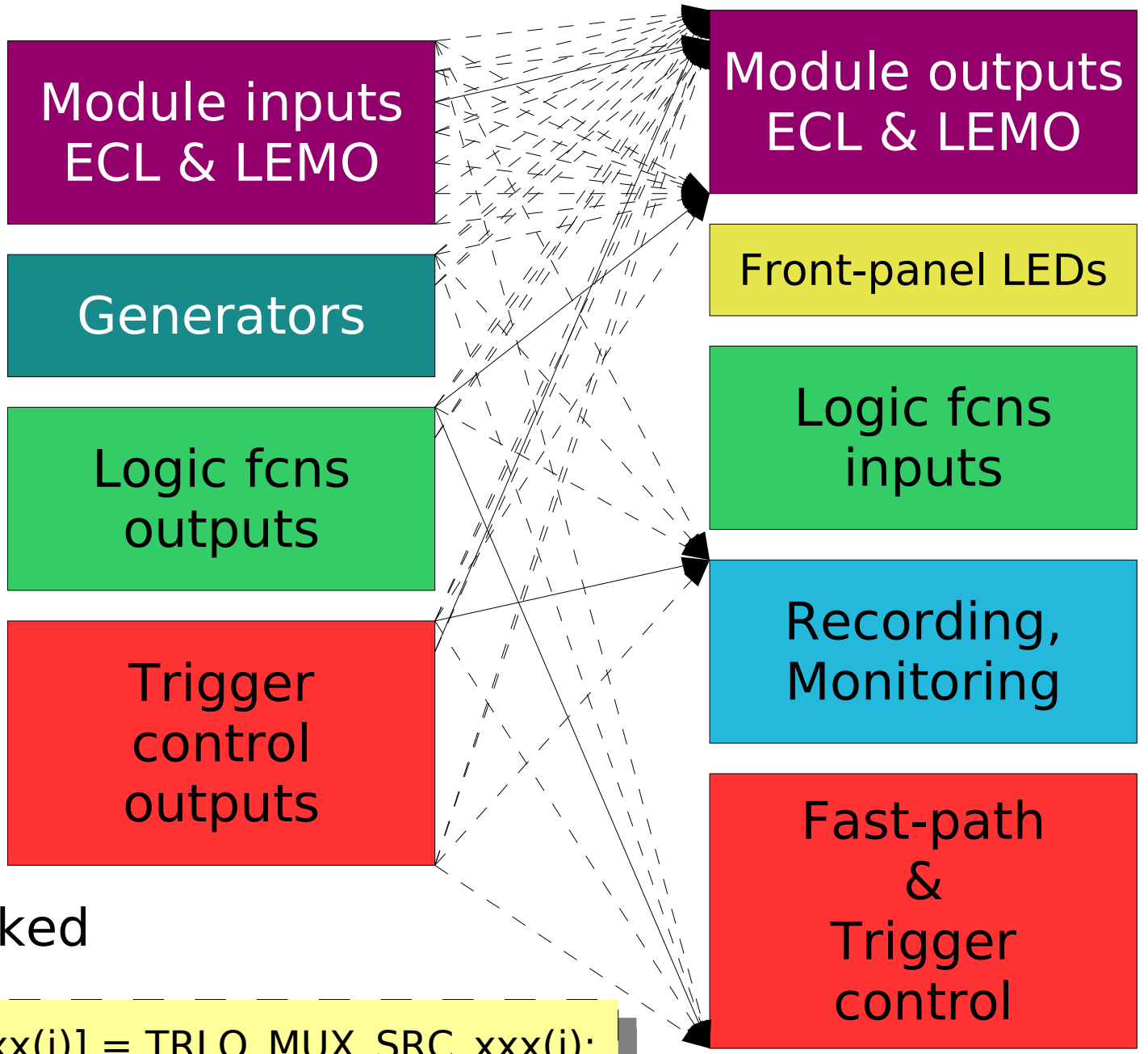
Any signal destination can use any source.

Routing cost:
2 cycles = 20 ns

Exceptions for timing-critical fast-path:

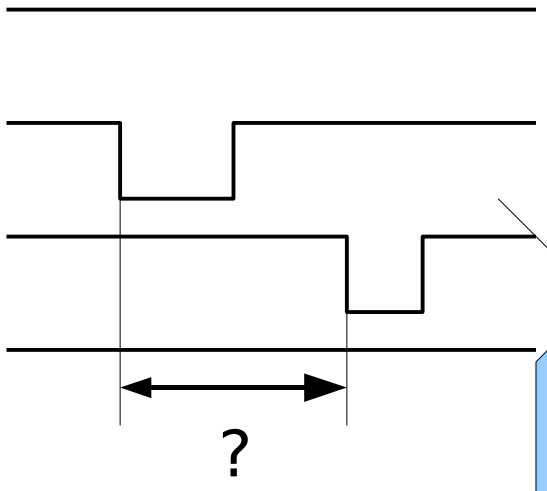
use fixed ECL in

master-start to any output, bitmasked




```
mux[TRLO_MUX_DEST_xxx(j)] = TRLO_MUX_SRC_xxx(i);
```

Trigger alignment - measurement



The triggers need to be aligned to make good coincidences

FPGA: self-triggered multi-event softscope

VME readout

```
...
Ch 22:  138-255e
Ch 23:  98-110
End
Start
Ch  9:  125-142
Ch 10:  126-143
Ch 11:  114-131
Ch 12:  127-144
Ch 21:  0-255e
End
Start
Ch  9:  125-142
Ch 10:  126-143
...
```

Analysis (histogramming)

```
     Timing relative to *
-----------------------------------------------
 1 :                     *                     :
 2 |9A                                         |-193/ 5
 3 |                 _554-._                   | -41/10
 4 |                     .43._                 |  -7/ 8
 5 |                    _A9_                   | -16/ 5
 6 |                      8A-                  |  -2/ 4
 7 |                     -43._                 |  -7/ 8
 8 |                     -43._                 |  -8/ 8
 9 |                    .---.......-7886-._  .--| 104/11
10 |                    .---...__..57886-__  .-| 112/11
11 |       34543-.-3555555555555444433--.___   |  51/40
12 |                    _34543-.345555555555555|  98/18
13 |_ _ ___ _ _ .3995                          | -55/ 5
14 |                                           |1200/nan
15 |                     697._                 |  -9/ 6
16 |                    .34443---_.---333333-3-|   5/19
17 |                                           |
18 |                                           |
19 |                                           |
20 |                                           |
21 |                                           |
22 |                                           | 220/ 0
23 |  A8                                       |-180/nan
24 |                                           |
-----------------------------------------------
```

2-log counts/bin (_ . - = 0,1,2)

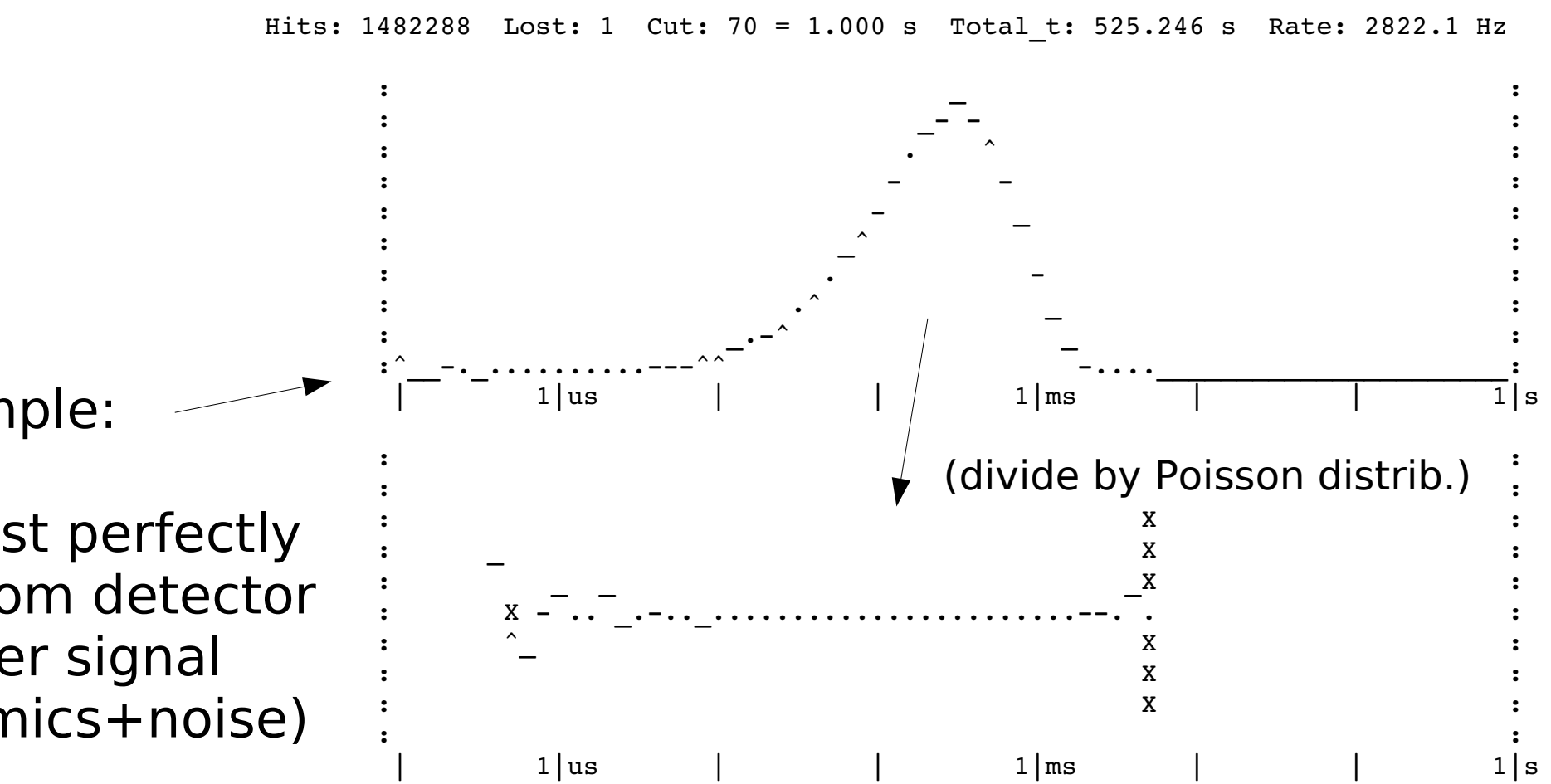
Pile-up measurements

Record inter-arrival times of signals

Every ion entering the cave → off-line pile-up rejection

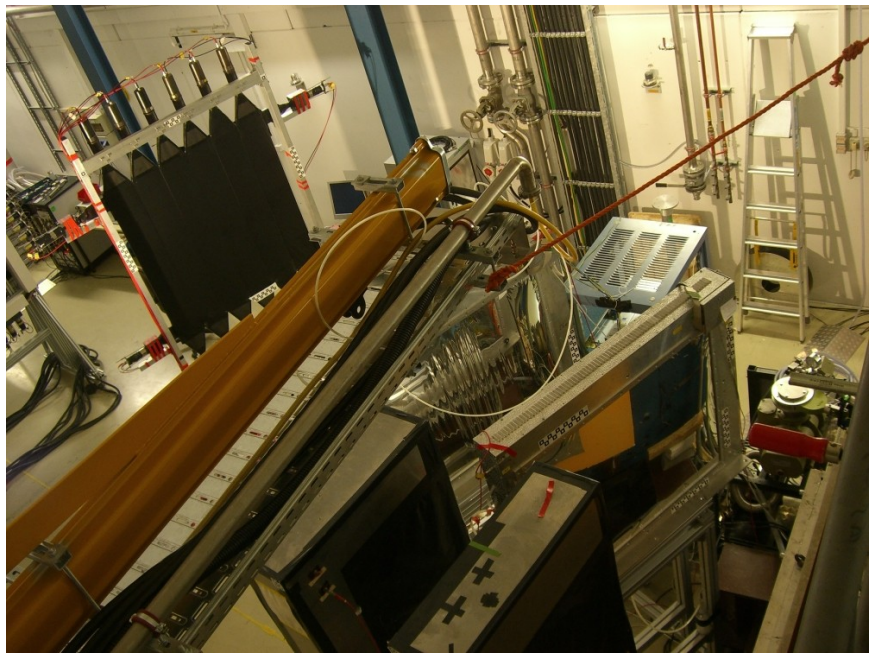
Example:

almost perfectly random detector trigger signal (cosmics+noise)



In operation:

S393 S306b S389

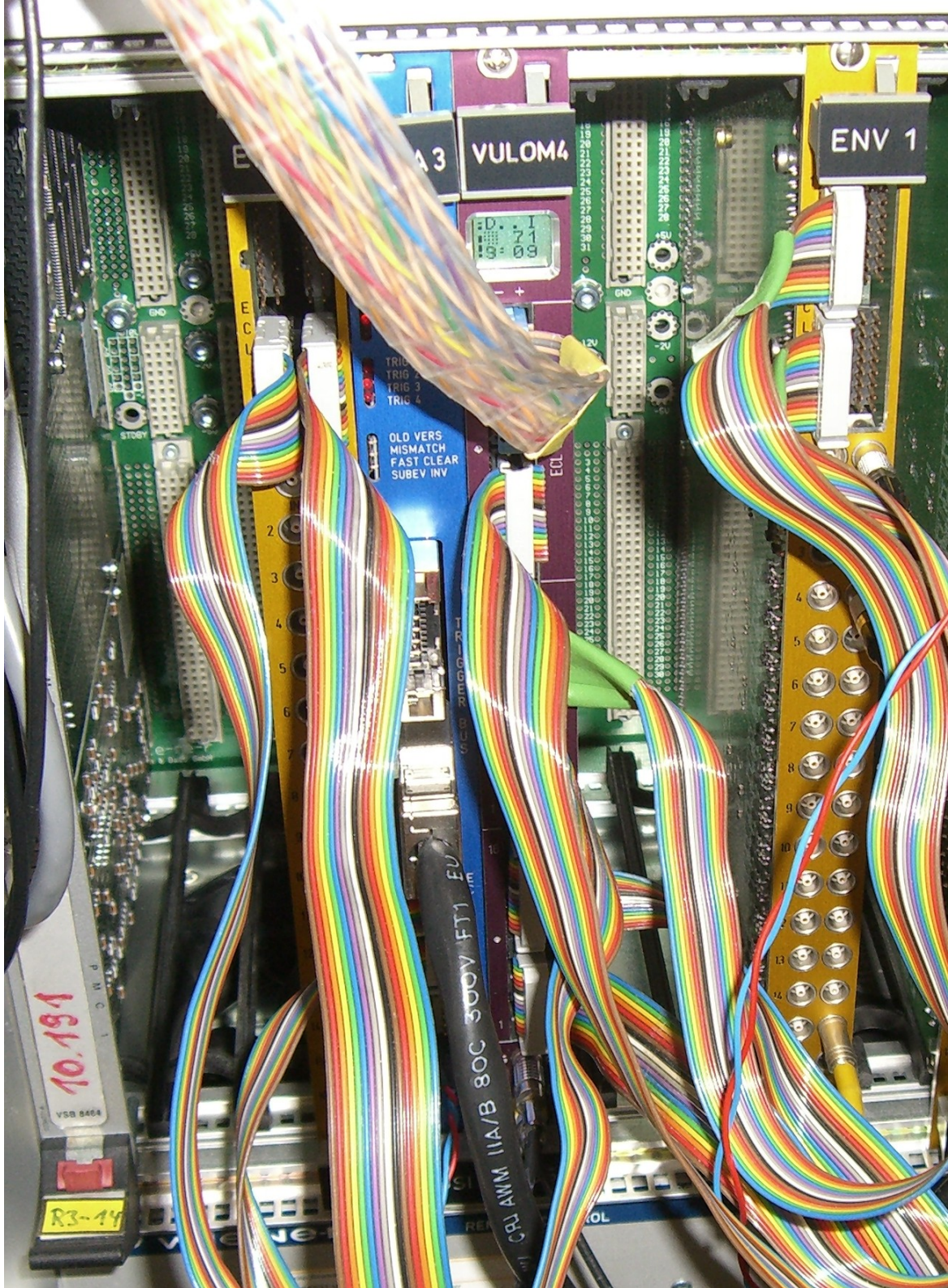


(Aug - Oct 2010)

Preceeded by intense *code* inspection (~300 kB) →

0 critical bugs found

2 minor bugs found in the wild



Work-in-progress

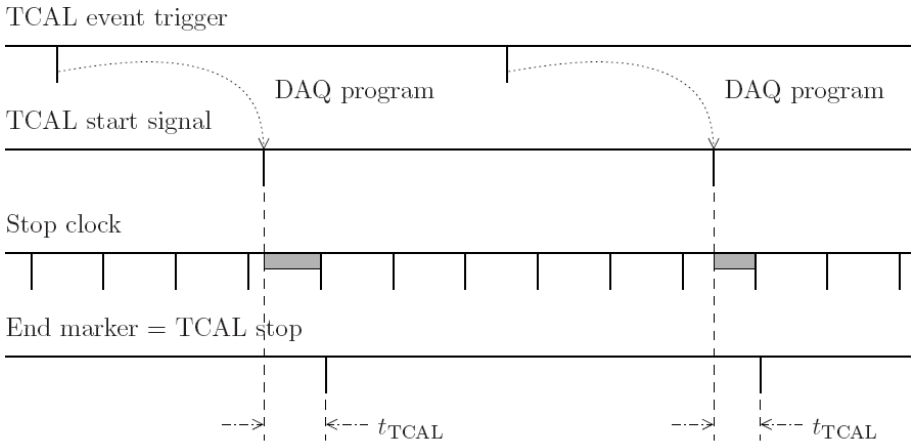
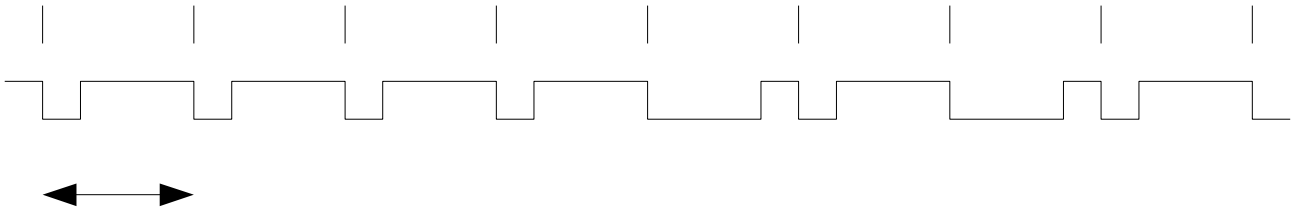


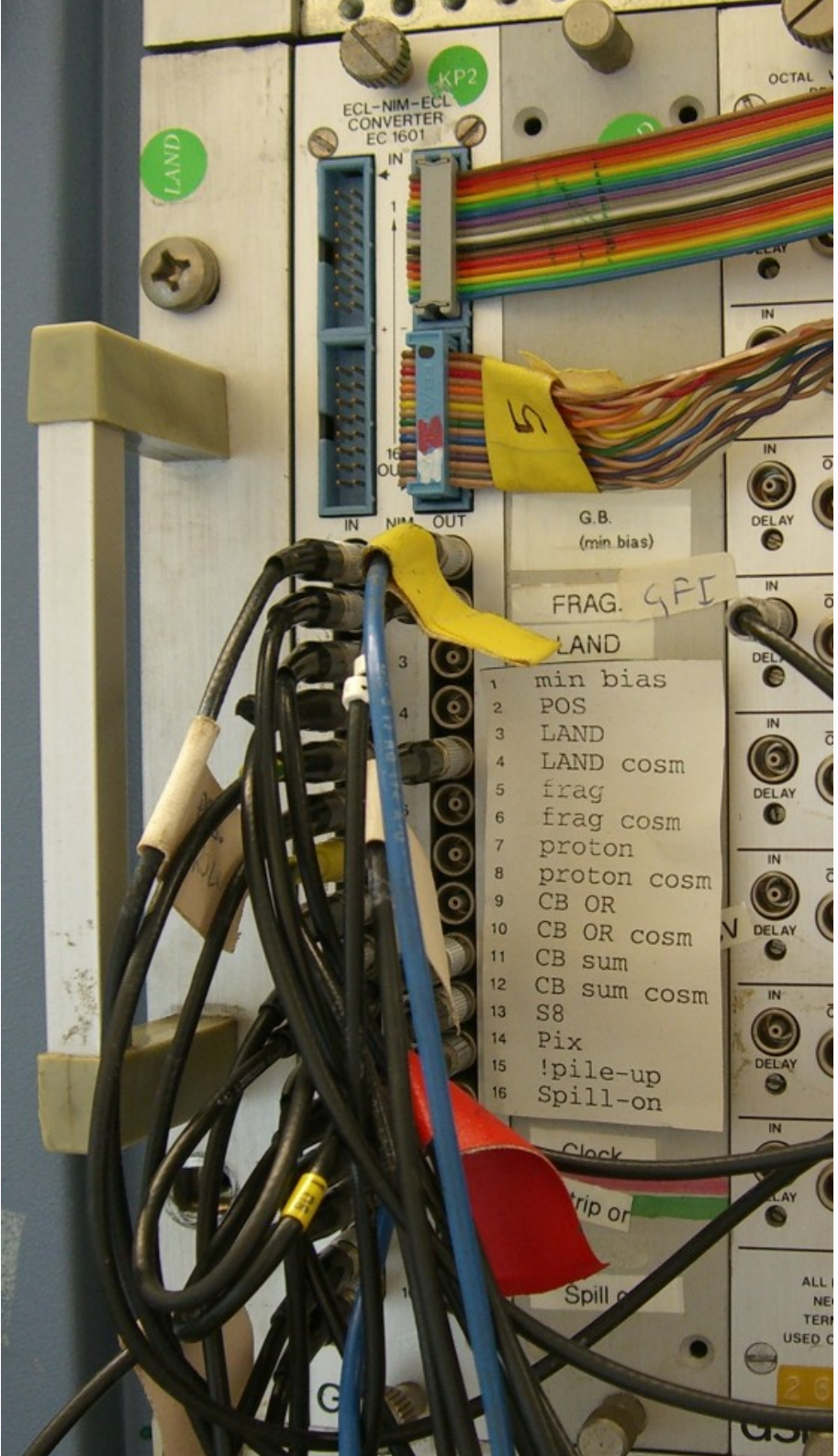
Figure 9.2: At the arrival of a TCAL start pulse, the next clock pulse is used as TCAL stop.

- Random TCAL
  (generate and measure semi-random pulses).
- 64-bit timestamps
  (10 ns resolution, 32 bits wrap after 42.9 s).
- Serial timestamps send + receive
  (for event synchronisation between systems over one signal line, 'survive' disconnections).



320 ns / 32 cycles

1 message cycle: 41 µs



Finale!

Thank you!

FPGAs are FUN!

Live by the compiler timing messages!