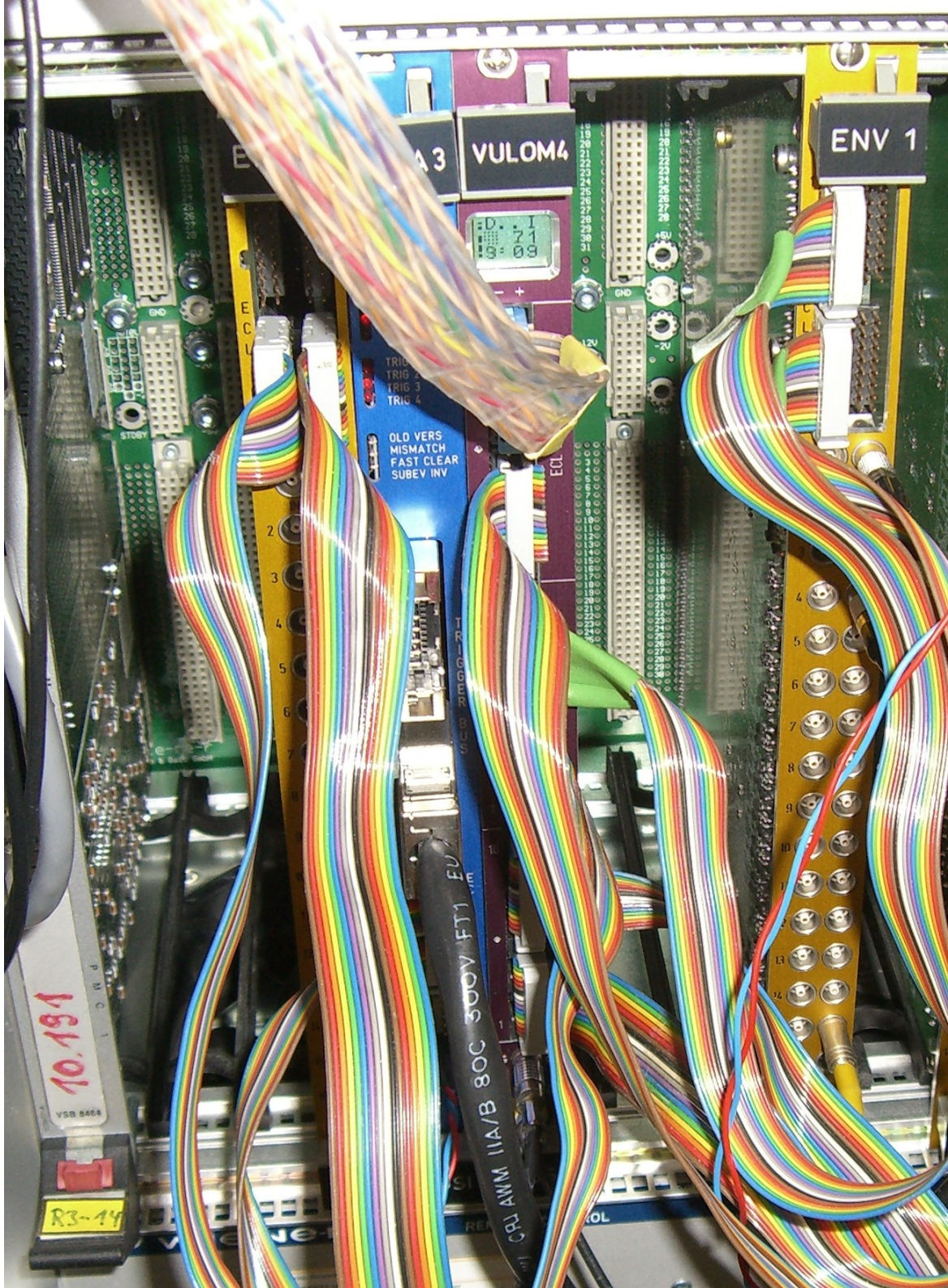
TRLO II – friendly FPGA trigger control

```
COMMAND(wiring)
{
  DEADTIME_IN(1) = ECL_IO_IN(4);

  ECL_IO_OUT(1) = ENCODED_TRIG(1);
  ECL_IO_OUT(2) = ENCODED_TRIG(2);
  ECL_IO_OUT(3) = ENCODED_TRIG(3);
  ECL_IO_OUT(4) = ENCODED_TRIG(4);
```

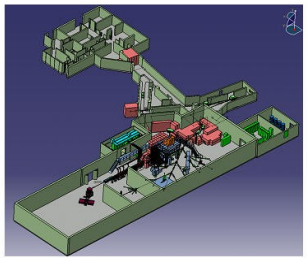
Håkan T. Johansson,
Chalmers, Göteborg

Königstein, July 2012

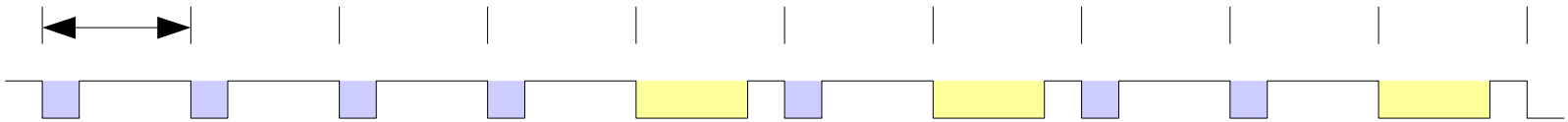


Outline

TRLO II for “rolling” ISOLDE VME DAQ



Serial timestamps Cave C - (FRS) - S2



New `trloctrl`, with parsed setup files

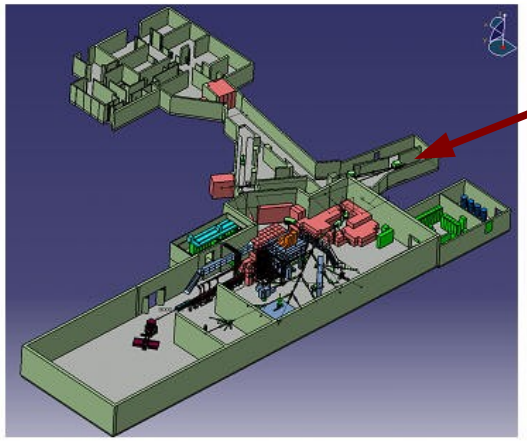
```
COMMAND(wiring)
{
  DEADTIME_IN(1) = ECL_IO_IN(4);

  ECL_IO_OUT(1) = ENCODED_TRIG
  ECL_IO_OUT(2) = ENCODED_TRIG
```

S245 experiment motto:

http://www-land.gsi.de/a_new_land/__public/experiments/s245/

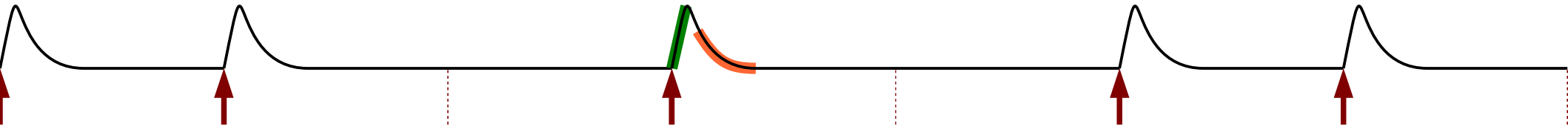
ISOLDE time structure



Protons from PS-booster every n*1.2 s

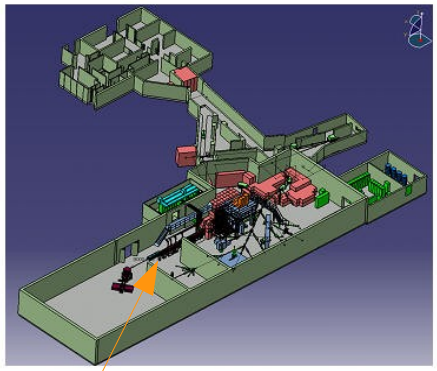
Beam time-profile:

- Release from target
- Decay of isotopes



=> Rather short “spills” of data.
(depending on halflife)

But it gets worse...



REX

REX-ISOLDE time structure

REX linac continously running

Ion bunch from REX-EBIS:

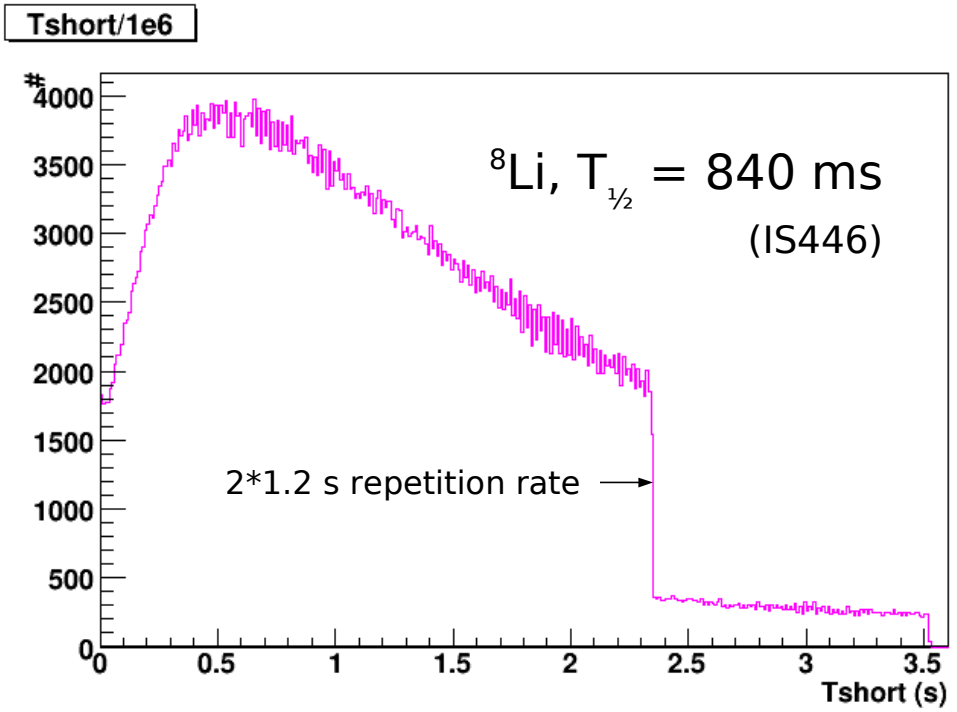
- Repetition: < 50 Hz
- Pulse length: 50 - 400 µs

=> Duty cycle: ~ 1 %

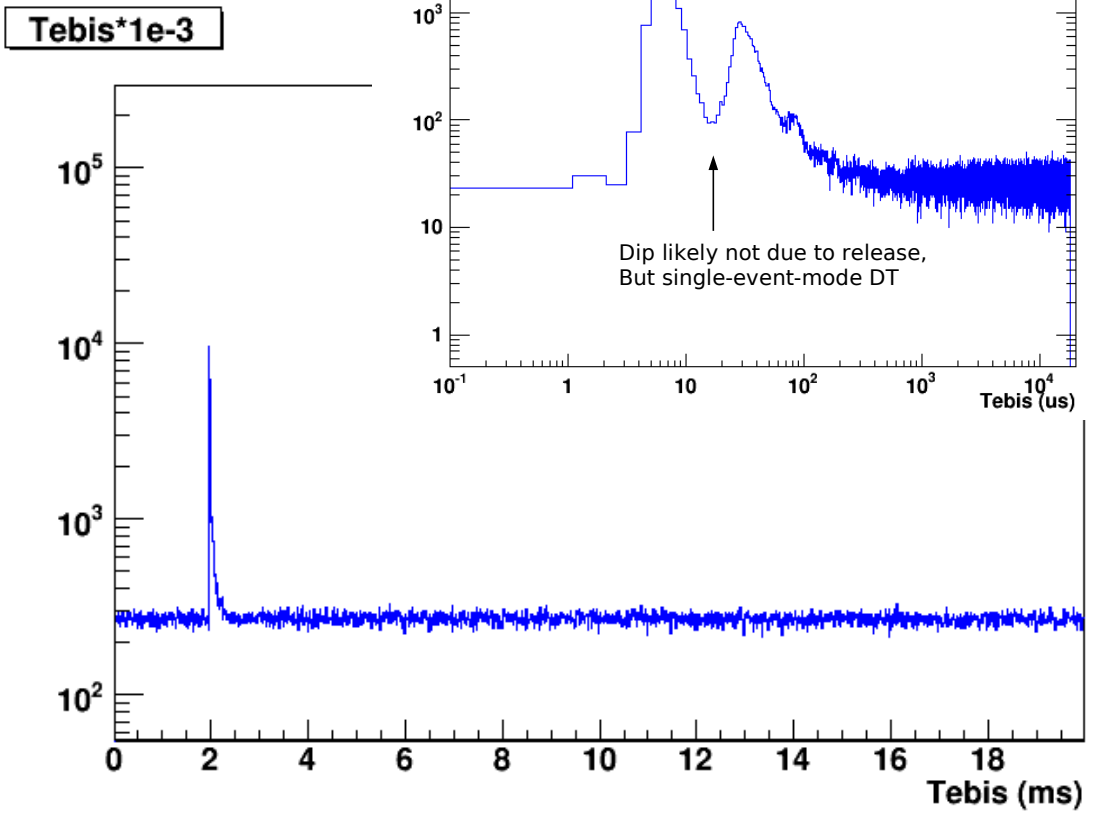
=> DAQ better be ready

Tshort (T1) & Tebis

T1: signals proton impact on production target



EBIS: ion release from charge breeder (REX-EBIS)



Vital for analysis cuts:

Tshort = Tevent – last T1

Tebis = Tevent – last EBIS

Multi-Event Operation

Background | "Physics" | Background

Limit deadtime to (busy):

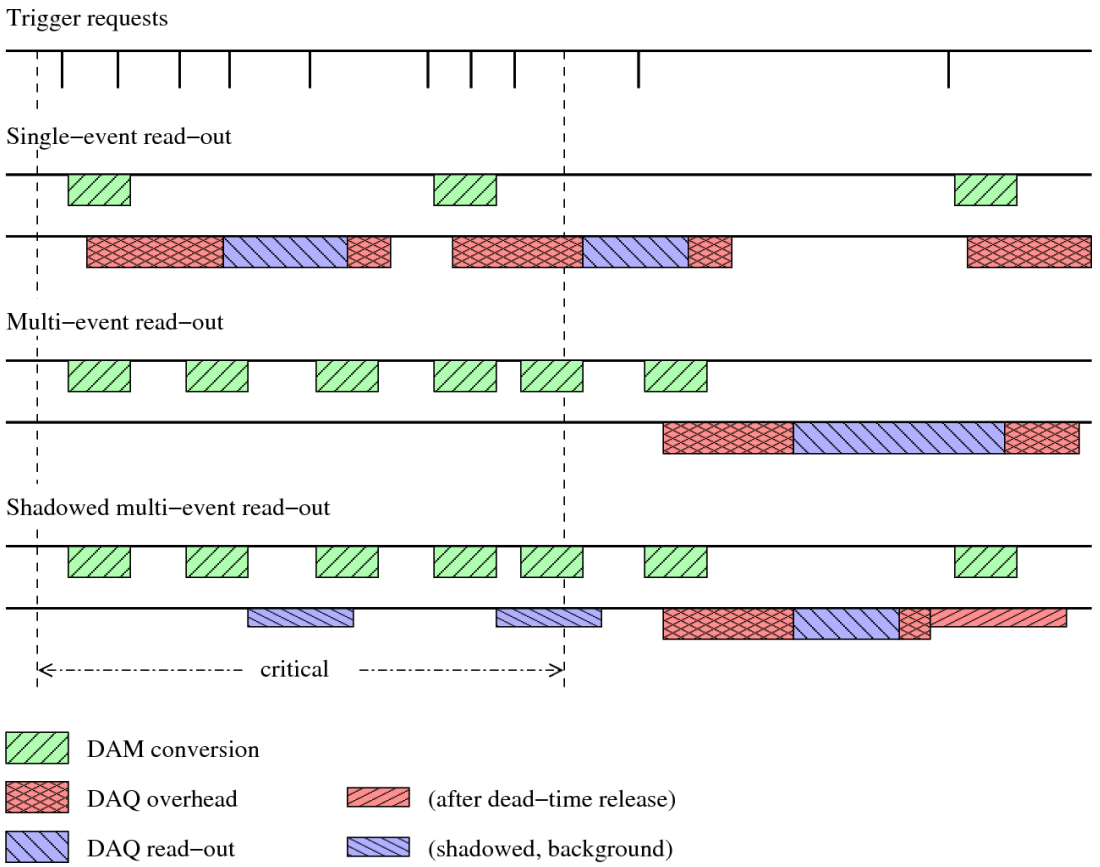
- Gate length +
- Conversion time

=> Multi-Event readout:

- Do readout when time non-critical
- Several physics events stored in one LMD (packaging) event

Also supported by TRLO II:

- Counting
- TPATs, time-stamps...

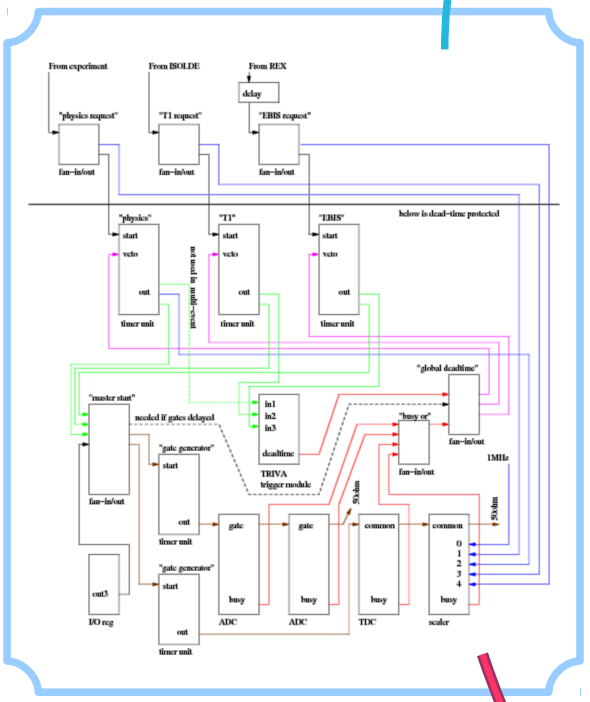


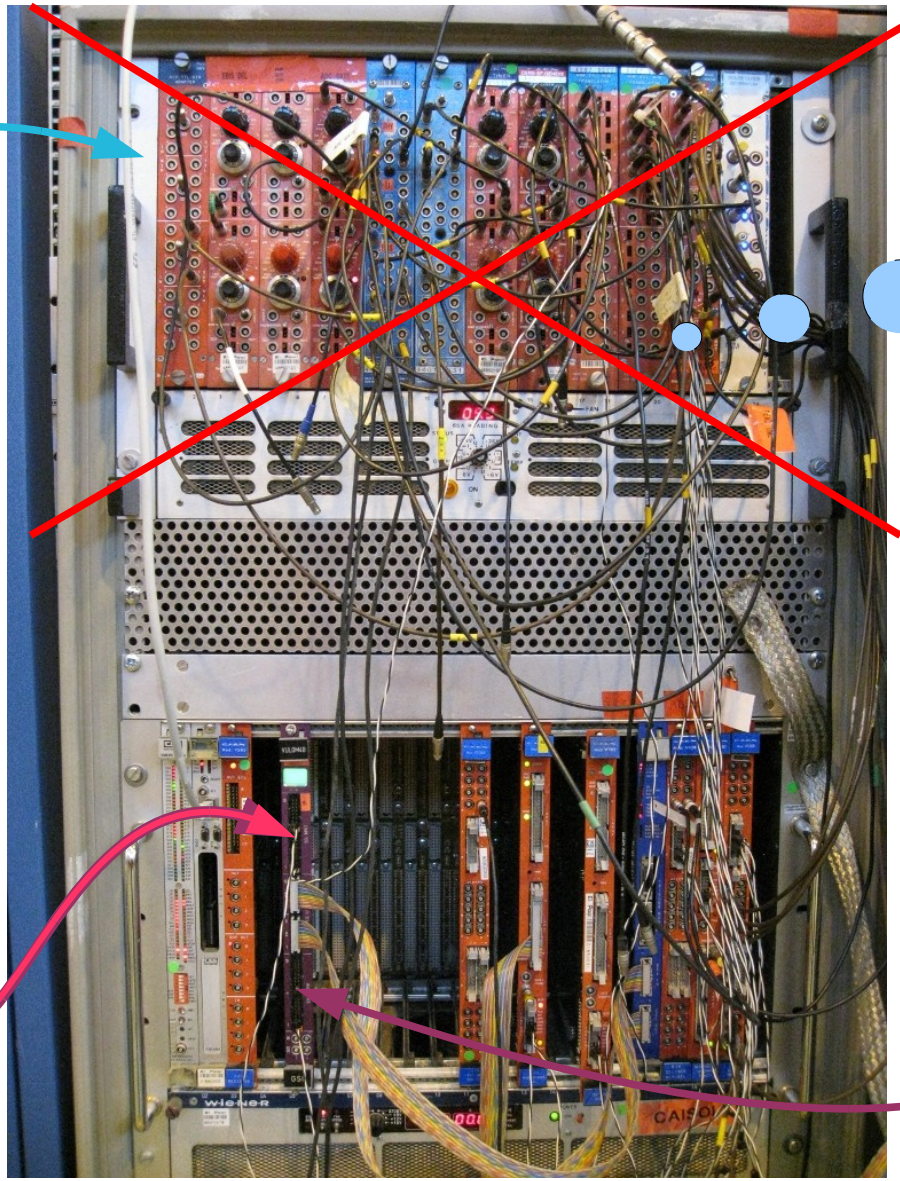
E.g. gate+conv = 10 µs → 100 kHz momentary rate

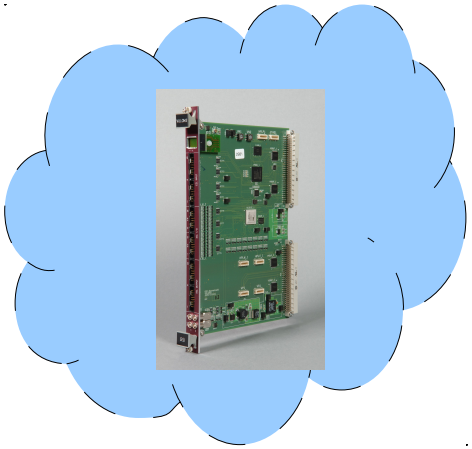
Next: TRLO II for ISOLDE

ISOLDE “std” VME DAQ gets TRLO II

Trigger logic filled
1 NIM crate







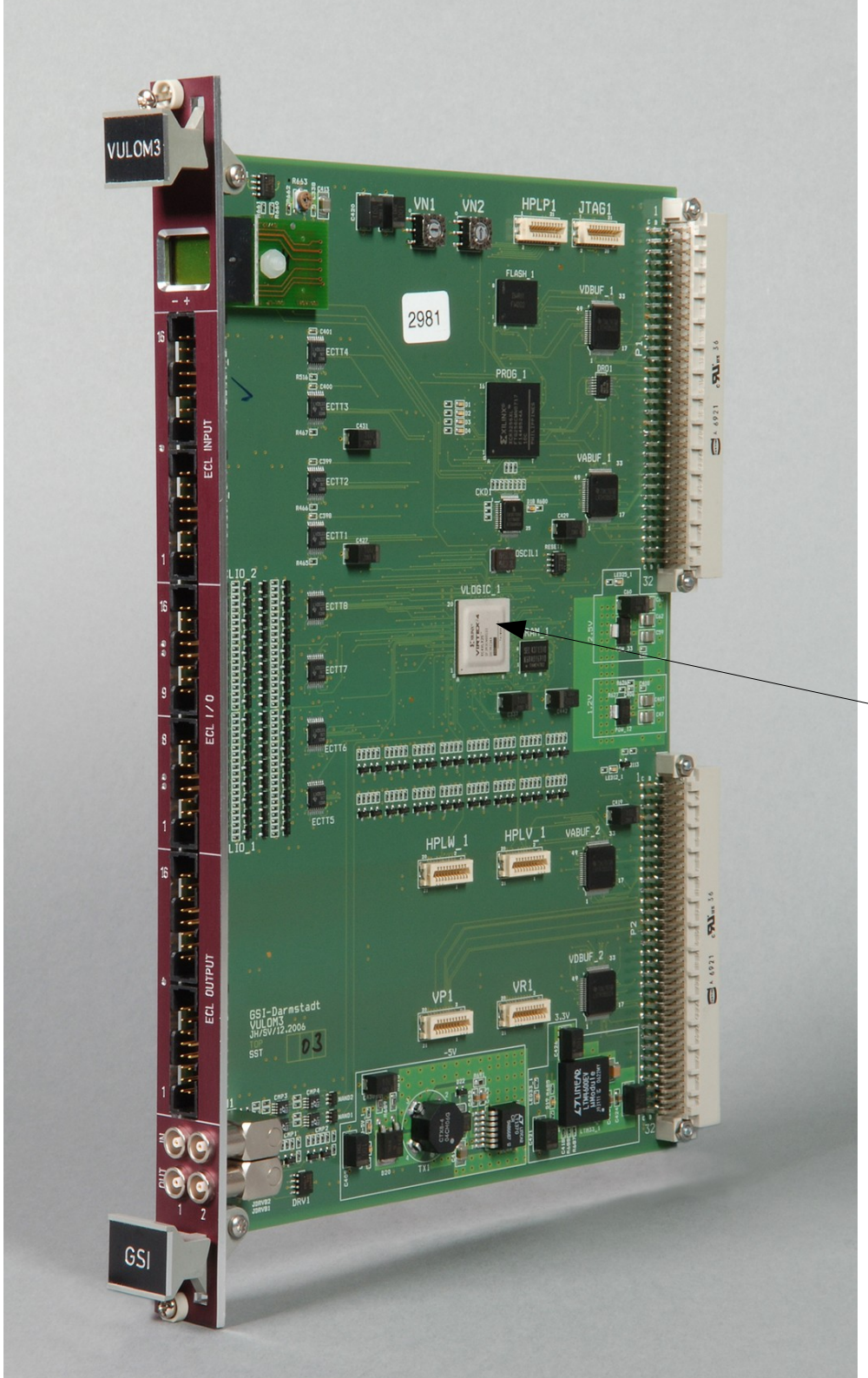
VULOM
(VME universal logic module)

by J. Hoffmann, GSI

Now by TRLO II
in VULOM4B

Inputs

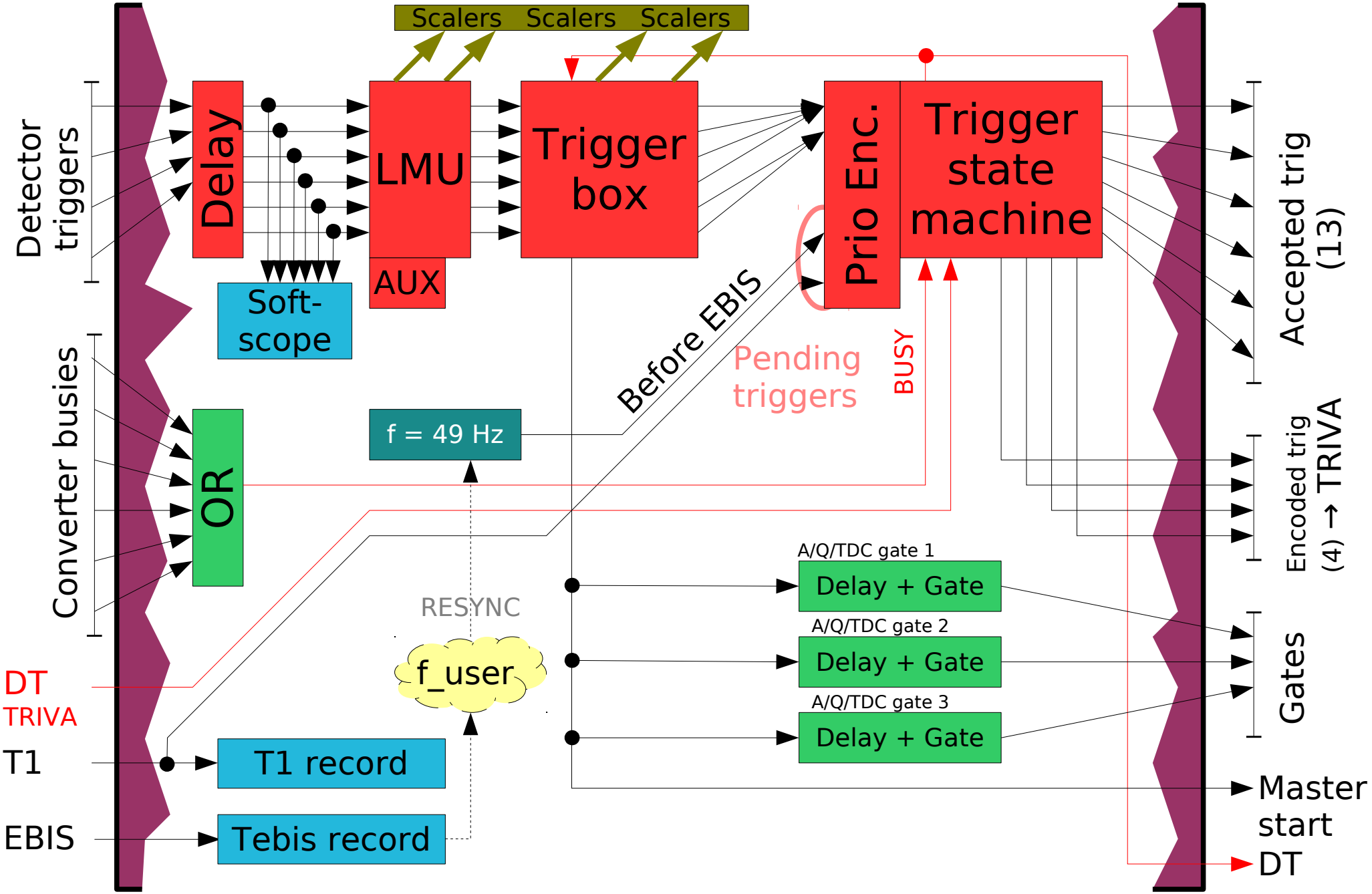
Outputs



VME

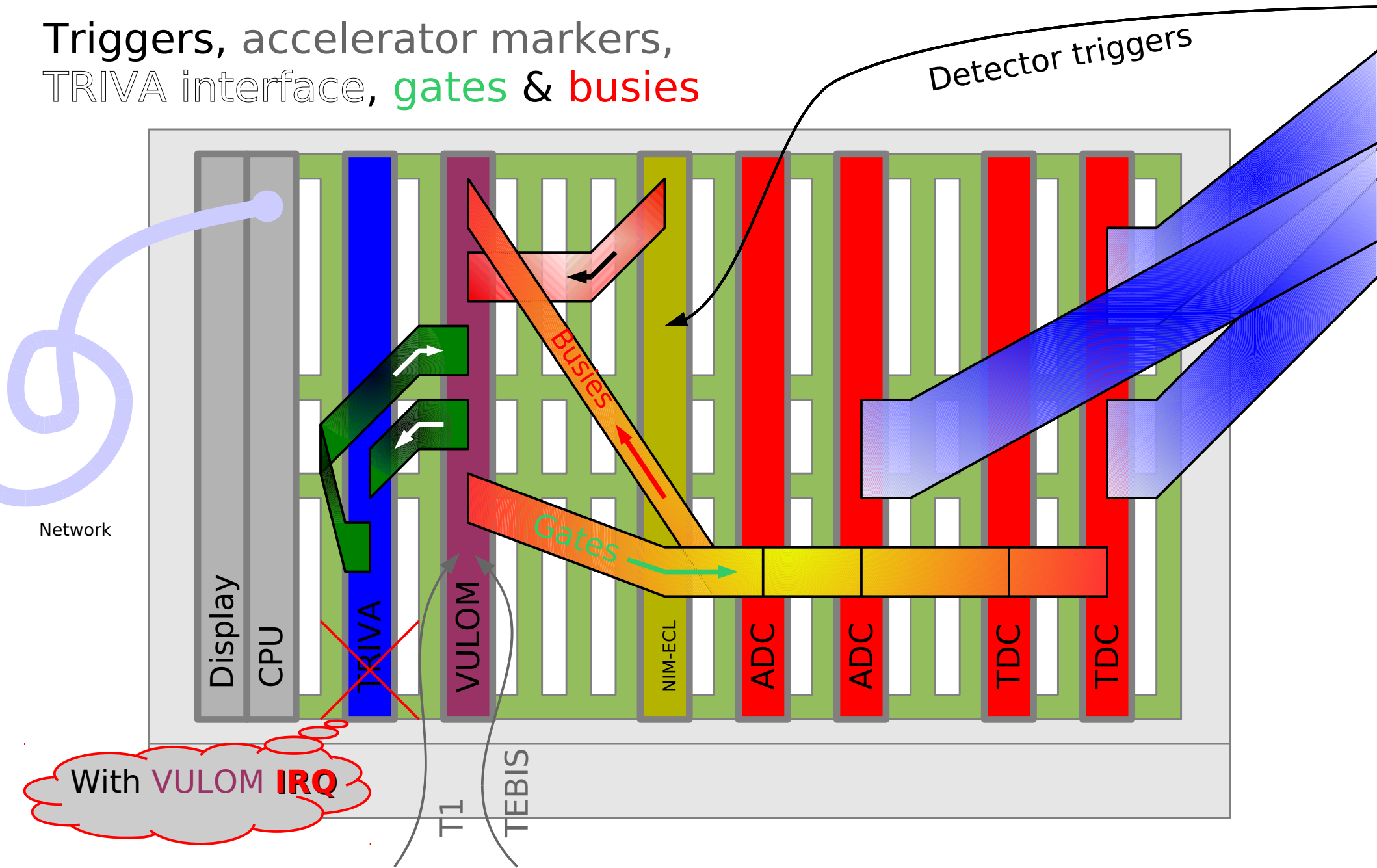
FPGA

TRLO II @ ISOLDE "std" VME



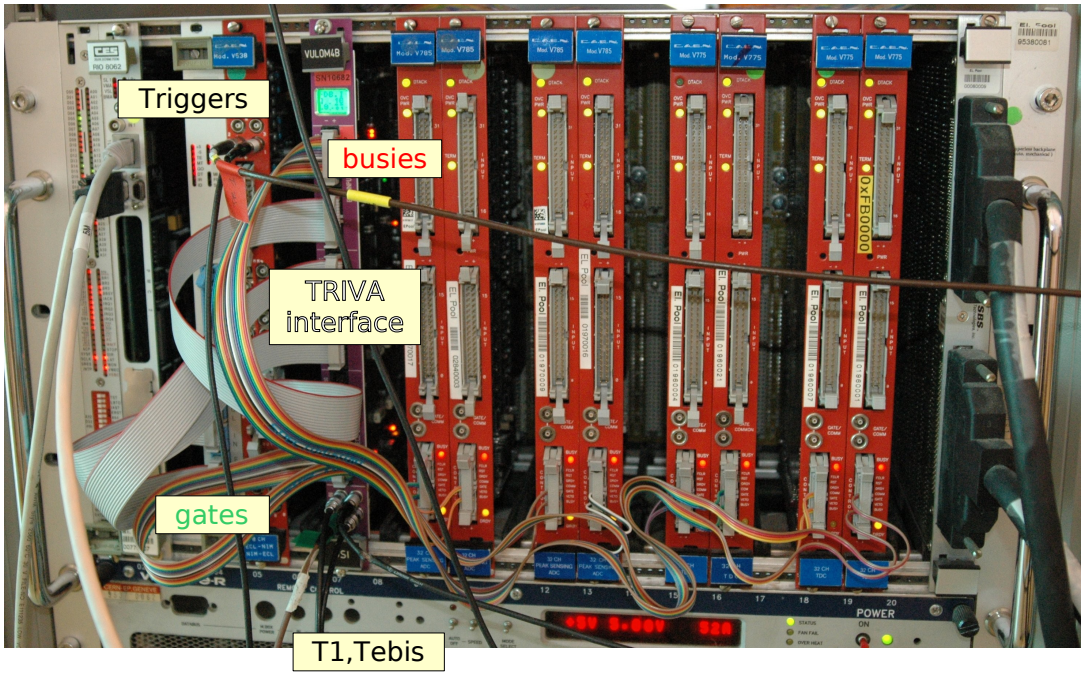
Only a few cables left (theory)

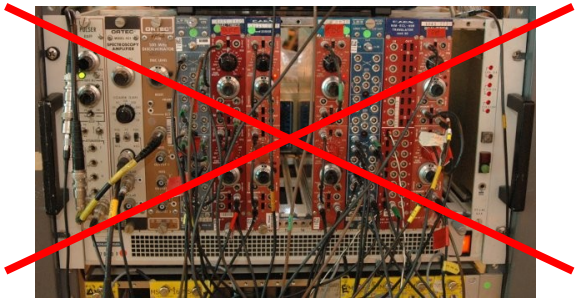
Triggers, accelerator markers,
TRIVA interface, gates & busies



Only a few cables left (practice)

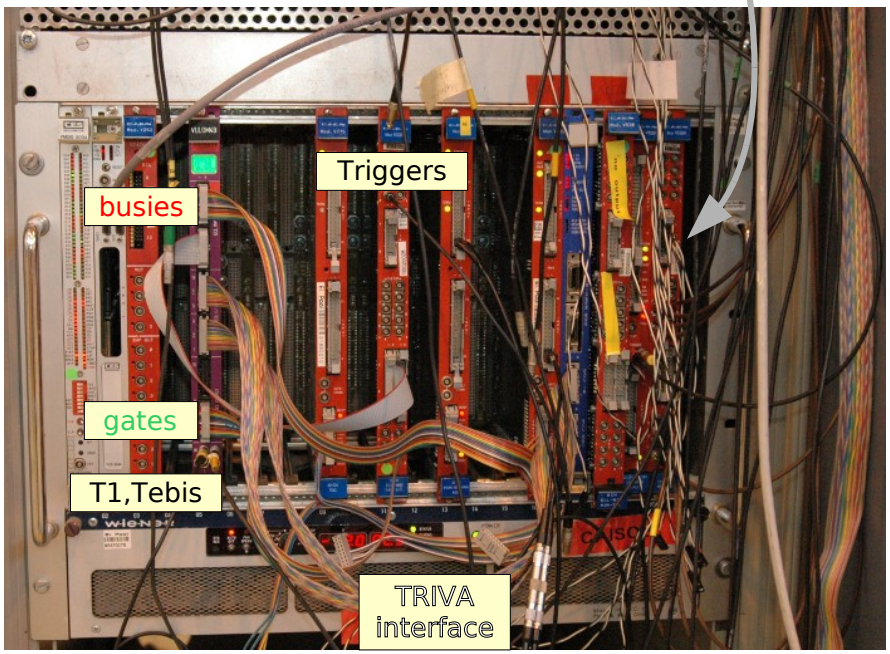
Triggers, accelerator markers,
TRIVA interface, gates & busies





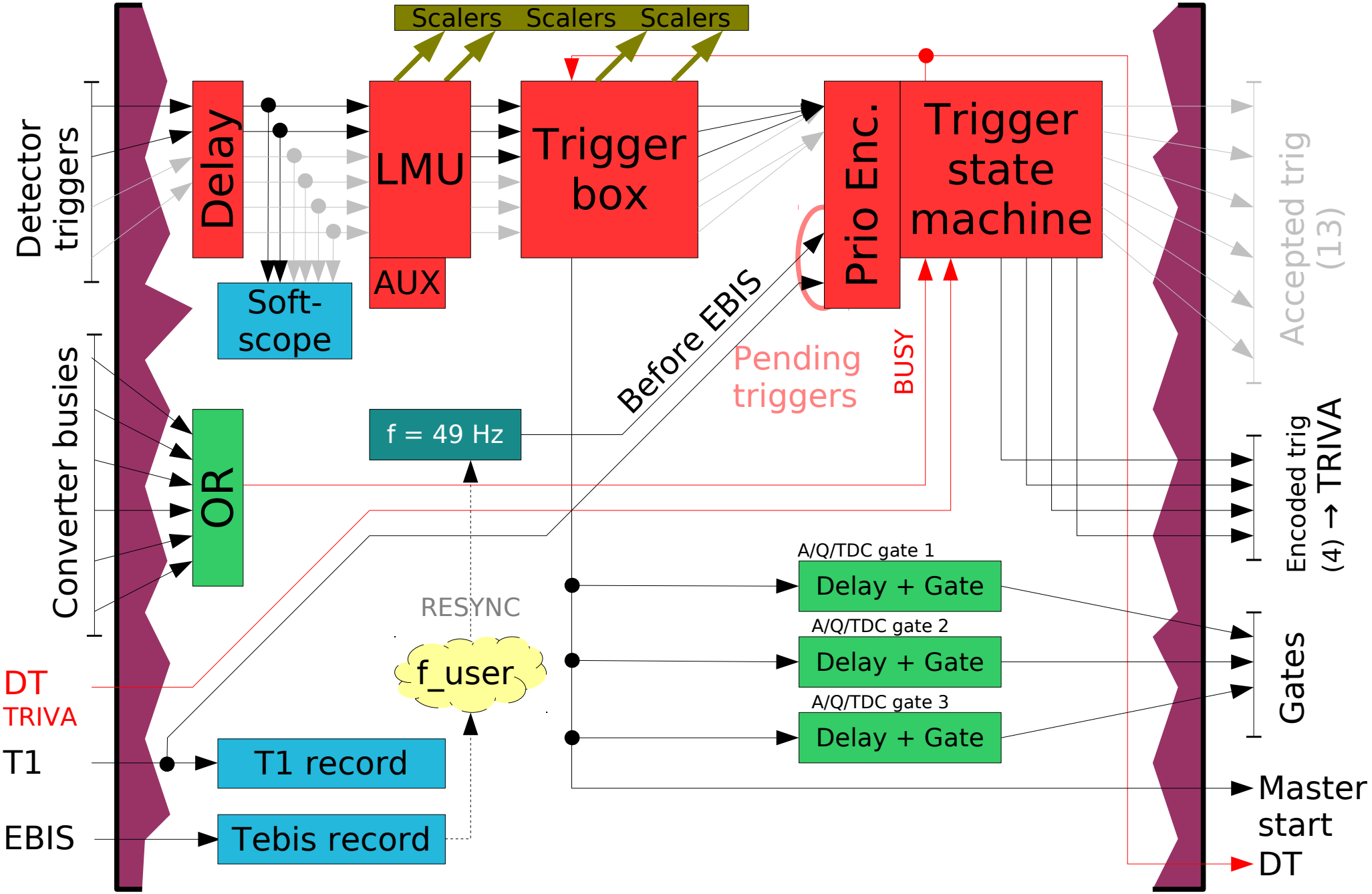
No more leaking deadtime!

Leftovers



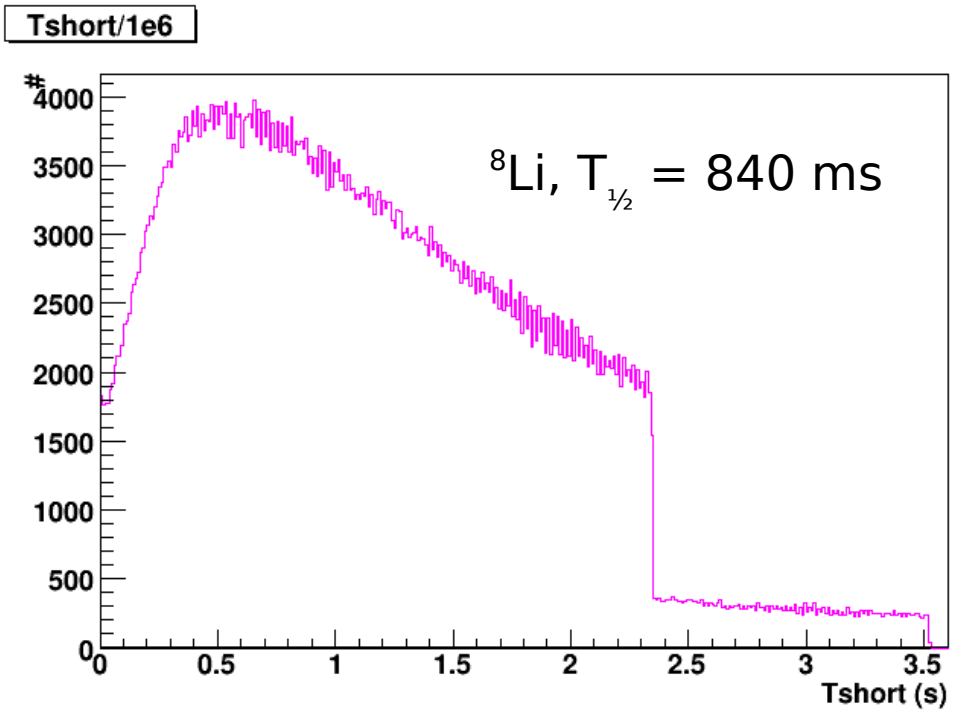
Some signals too?

Again: TRLO II @ ISOLDE “std” VME

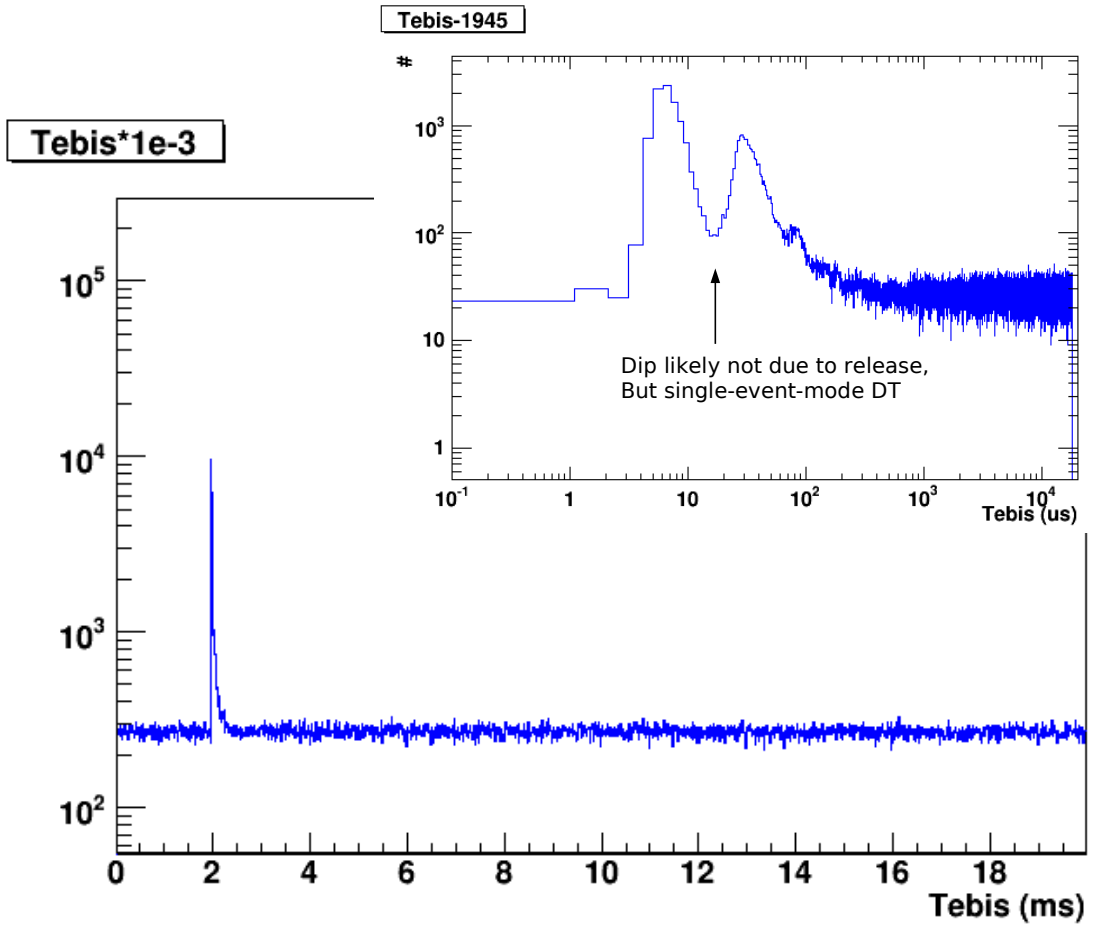


Tshort (T1) & Tebis from stamps

T1: signals proton impact on production target



EBIS: ion release from charge breeder (REX-EBIS)



T1 record

Tebis record

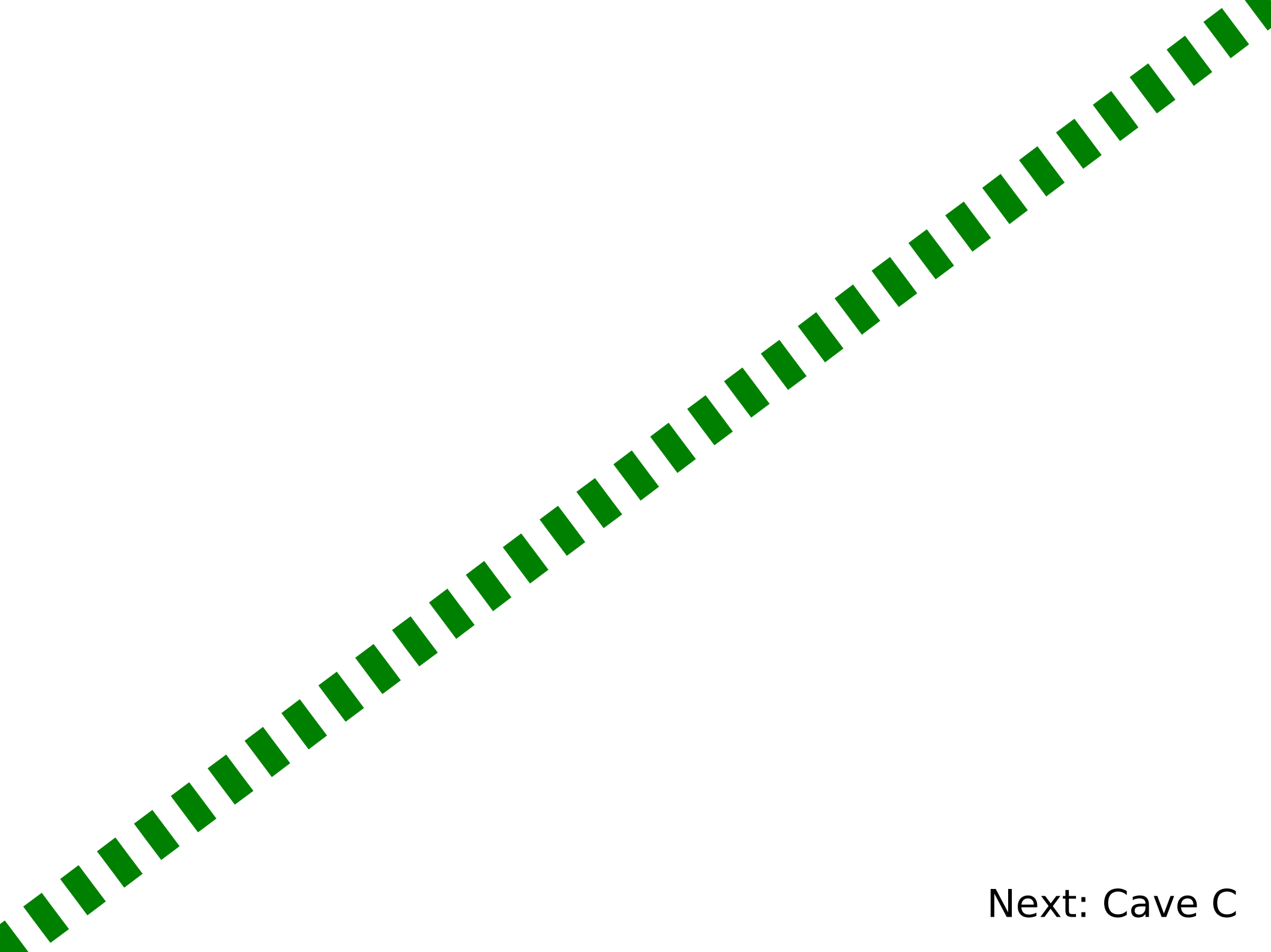
\+

All events TPAT + time recording

:

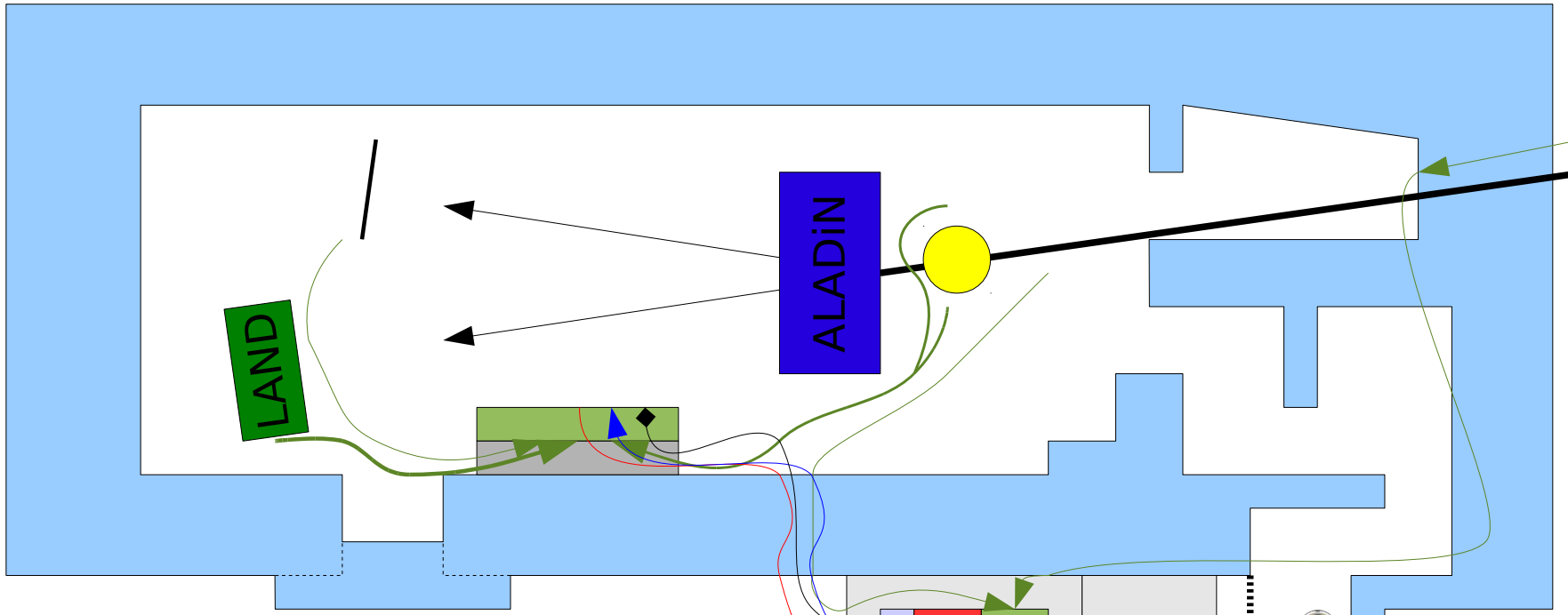
Tshort = Tevent – last T1 t-stamp

Tebis = Tevent – last EBIS t-stamp


Next: Cave C

Cave C trigger (2004-)

(Analog to LAND@ Cave B -2004)



Detector signals

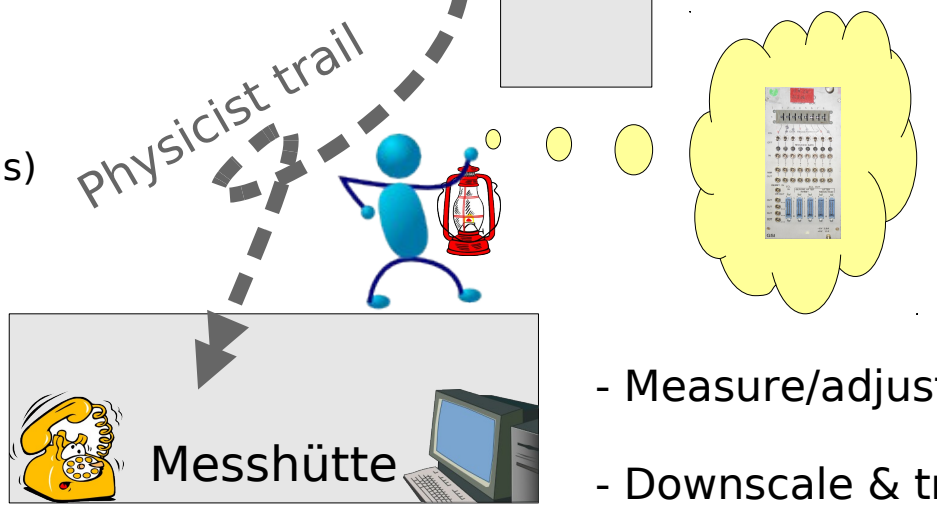
Split, QDC, TDC, CFD
Delays (500 ns)

Detector triggers (<100 ns)

Trigger decision (~50 ns)

Master start (<100 ns)

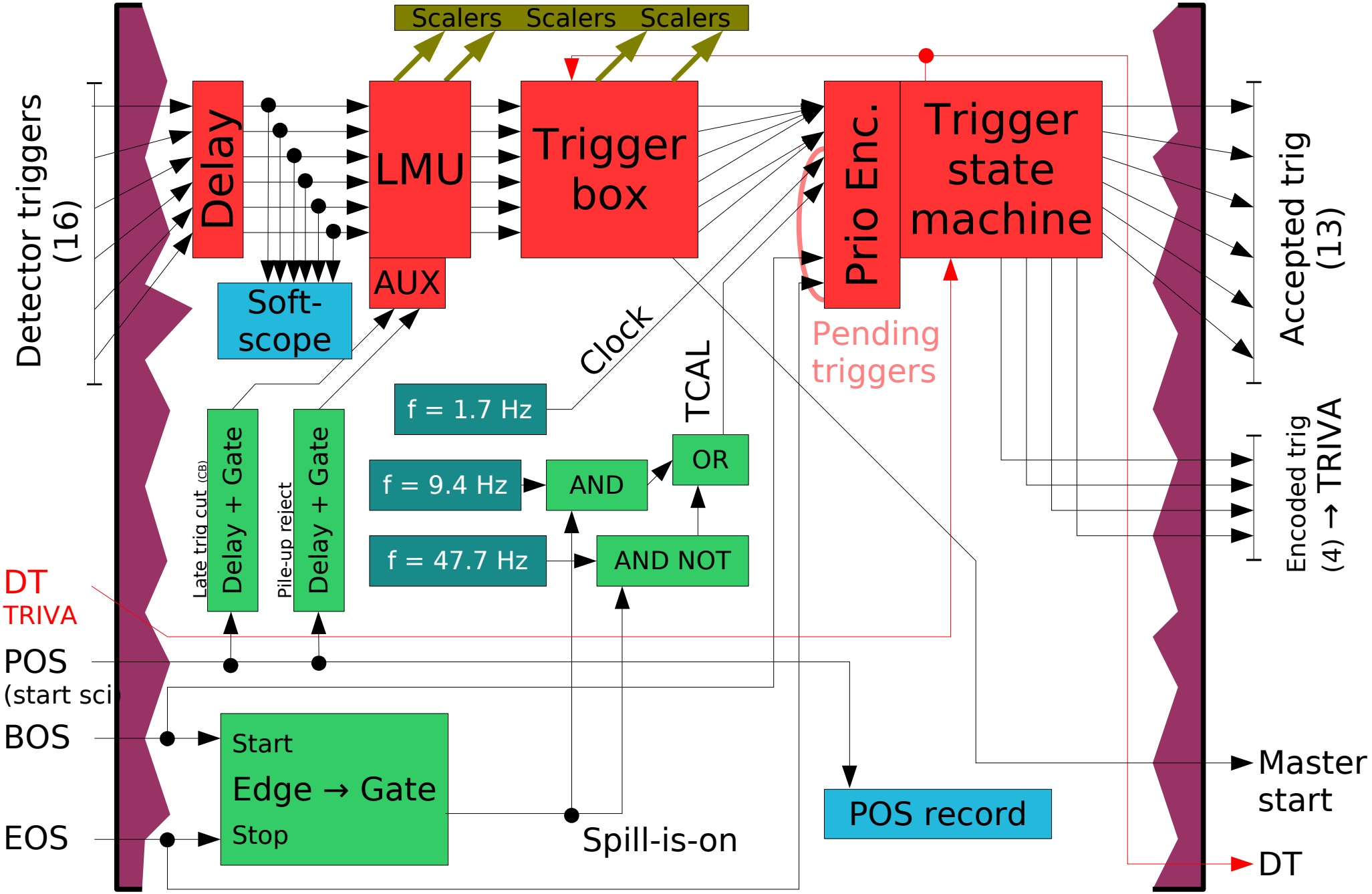
Trigger bus



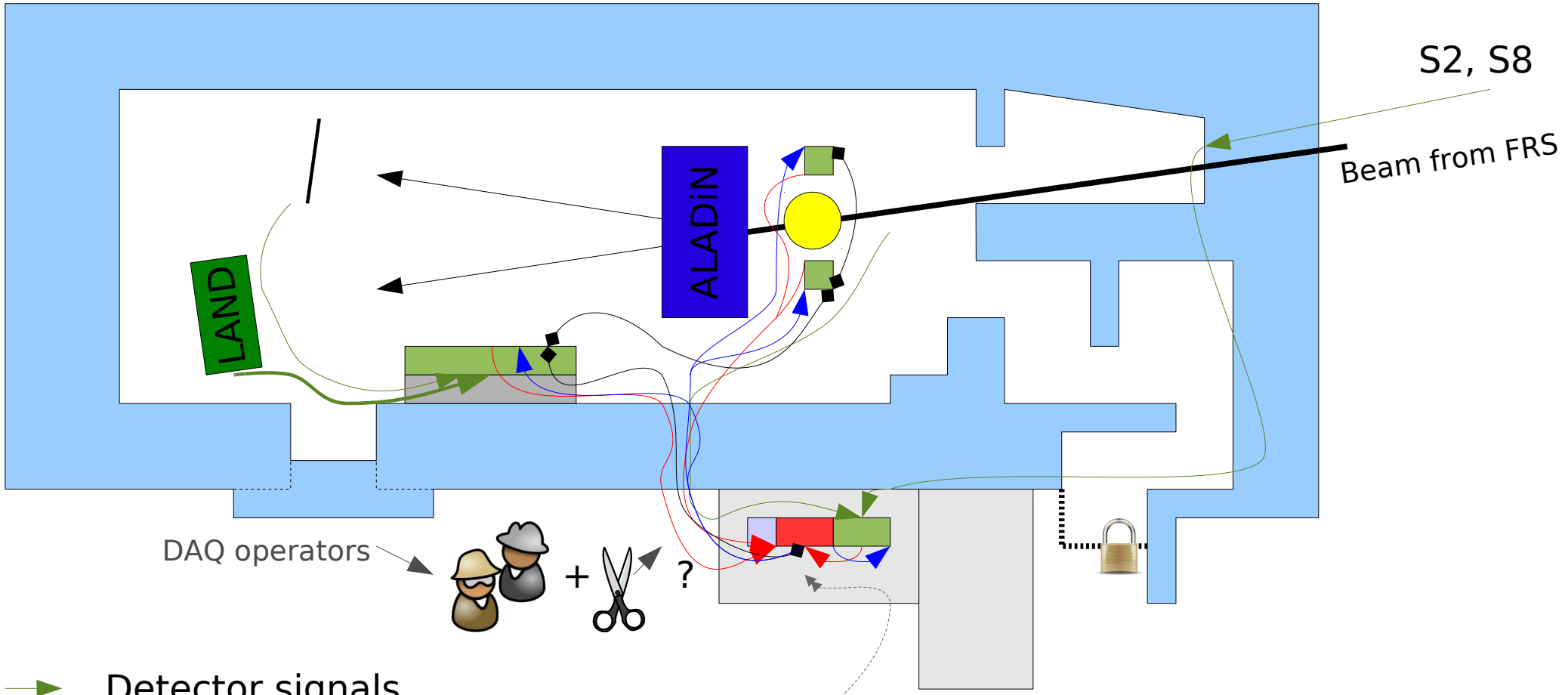
Access required:

- Measure/adjust trigger alignment
- Downscale & trigger on/off

TRLO II @ Cave C / LAND-setup



Cave C trigger (2010-)



Detector signals

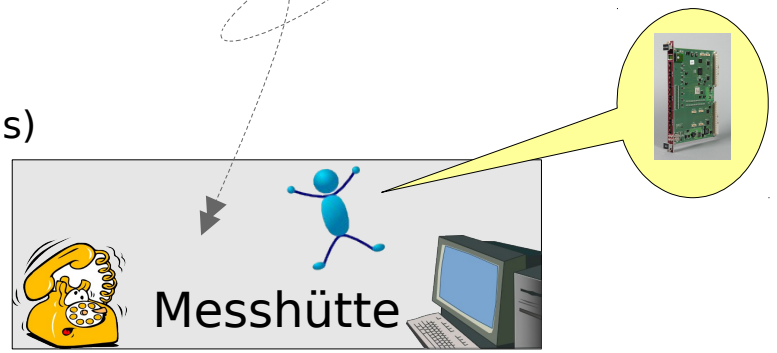
Split, QDC, TDC, CFD
Delays (500 ns)

Detector triggers (<100 ns)

Trigger decision (~50 ns)

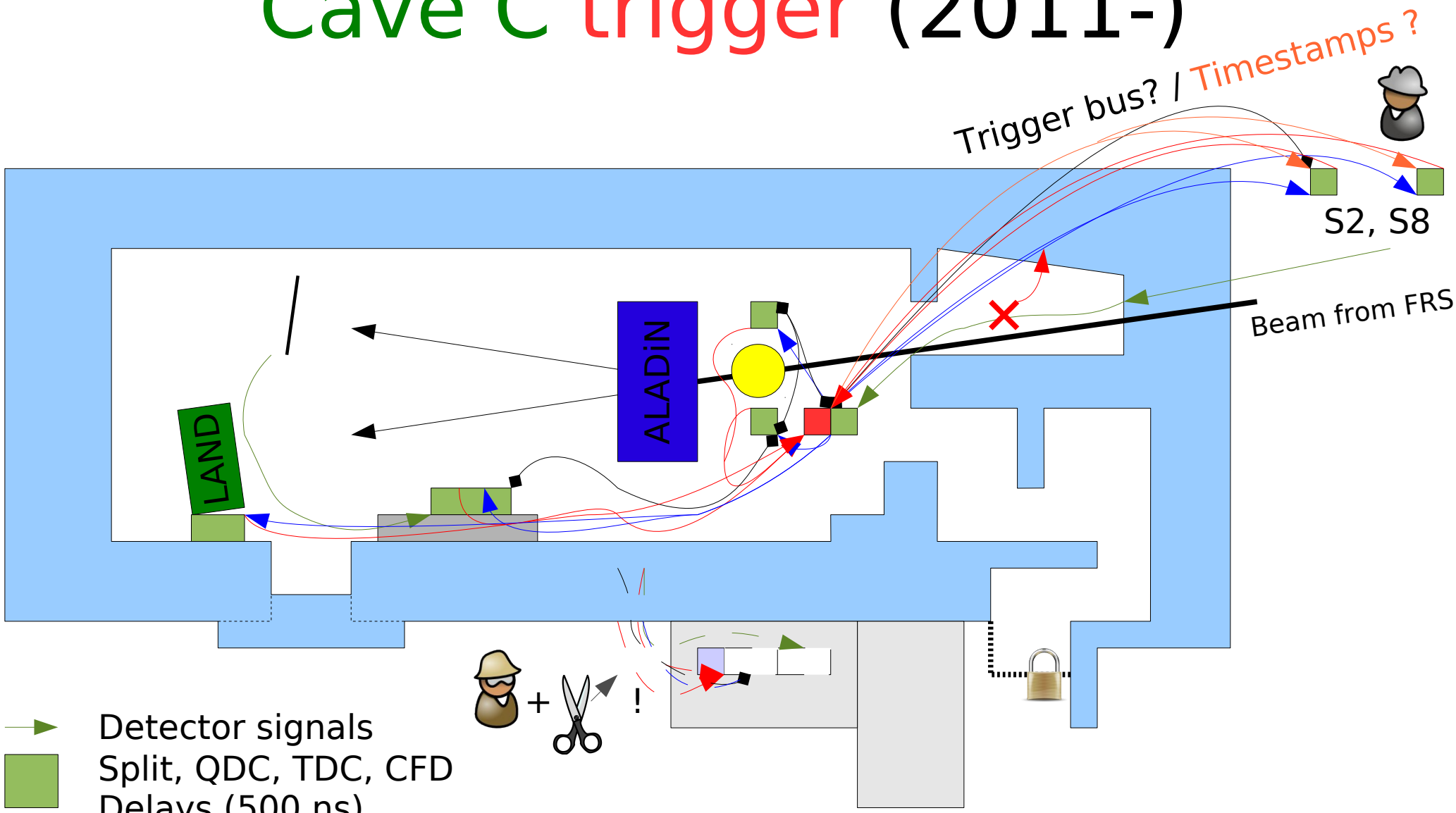
Master start (<100 ns)

Trigger bus



Access not required -
trigger completely
computer-controlled

Cave C trigger (2011-)



Detector signals

Split, QDC, TDC, CFD
Delays (500 ns)

Detector triggers (<100 ns)

Trigger decision (~50 ns)

Master start (<100 ns)

Trigger bus

Trigger fully
computer-controlled
→ moved into Cave!

Trigger bus Cave C → S2 ?

Unexpected problem:

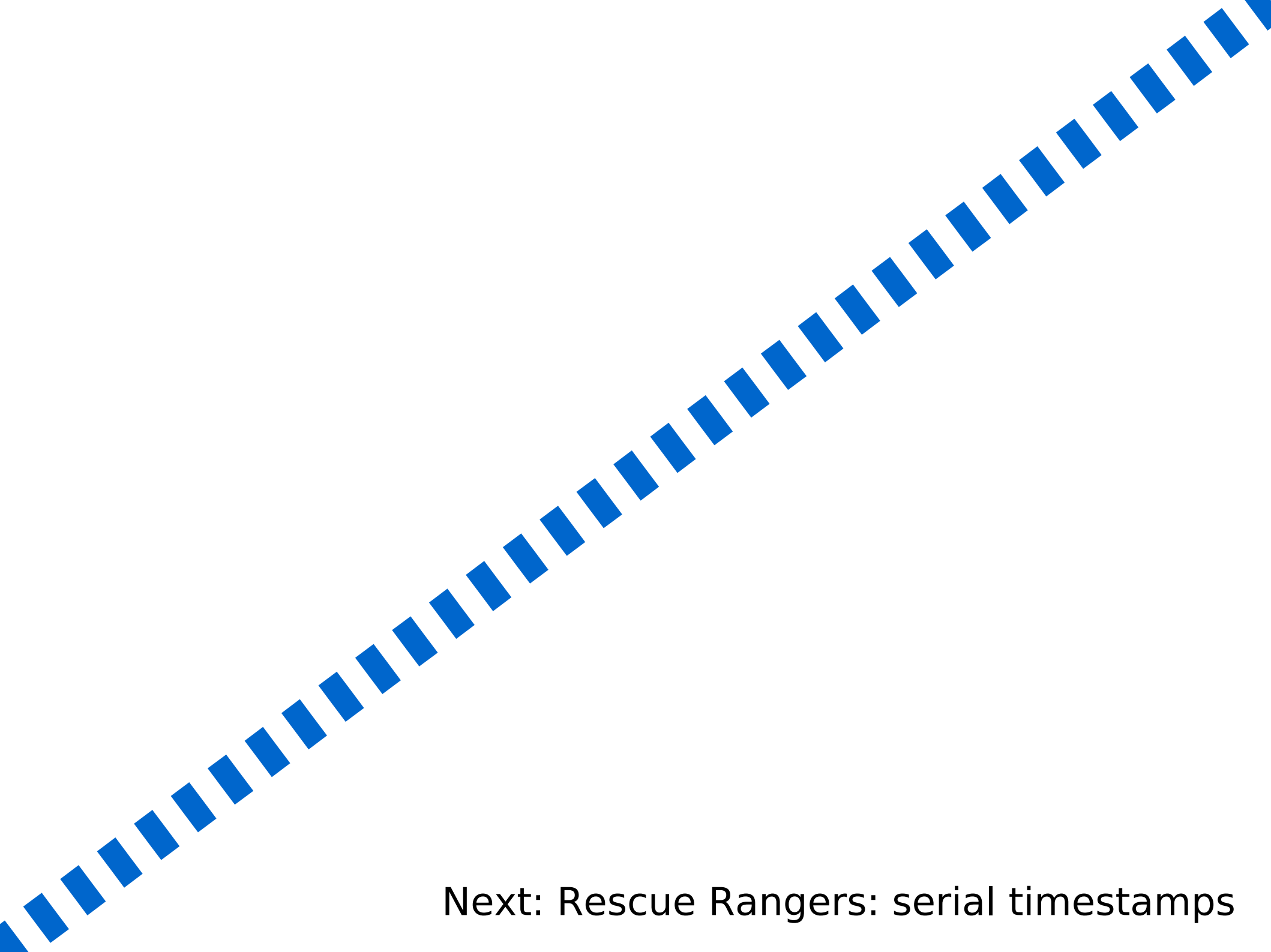
Trigger bus lost signal integrity Cave C → S2

=> Common dead-time domain
with TRIVA @ S2 not possible.

Timestamps Cave C → S2 ?

=> Time-stamp distribution with TITRIS modules not possible.
(uses same type bus cable)

Time-stamp distribution with two spare TRIDI1 modules and TRLO II. Using 'any' cable.



Next: Rescue Rangers: serial timestamps

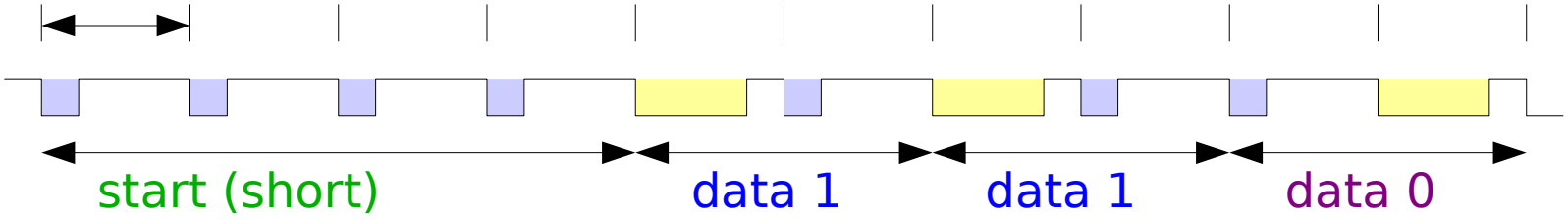
Serial timestamp protocol

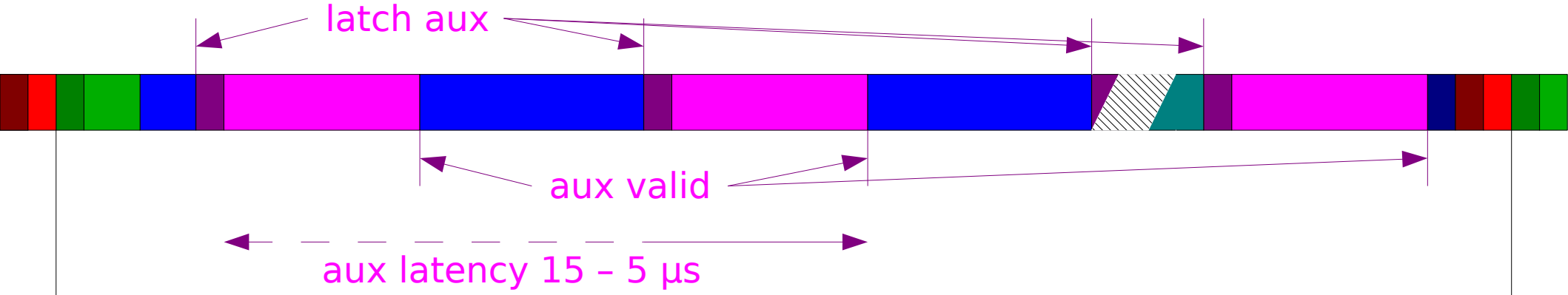
Pulses short (¼) or long (¾)

Each payload bit sent as two pulses, second inverted

Pattern of 4 short pulses for start-synchronisation

320 ns = 32 clock cycles

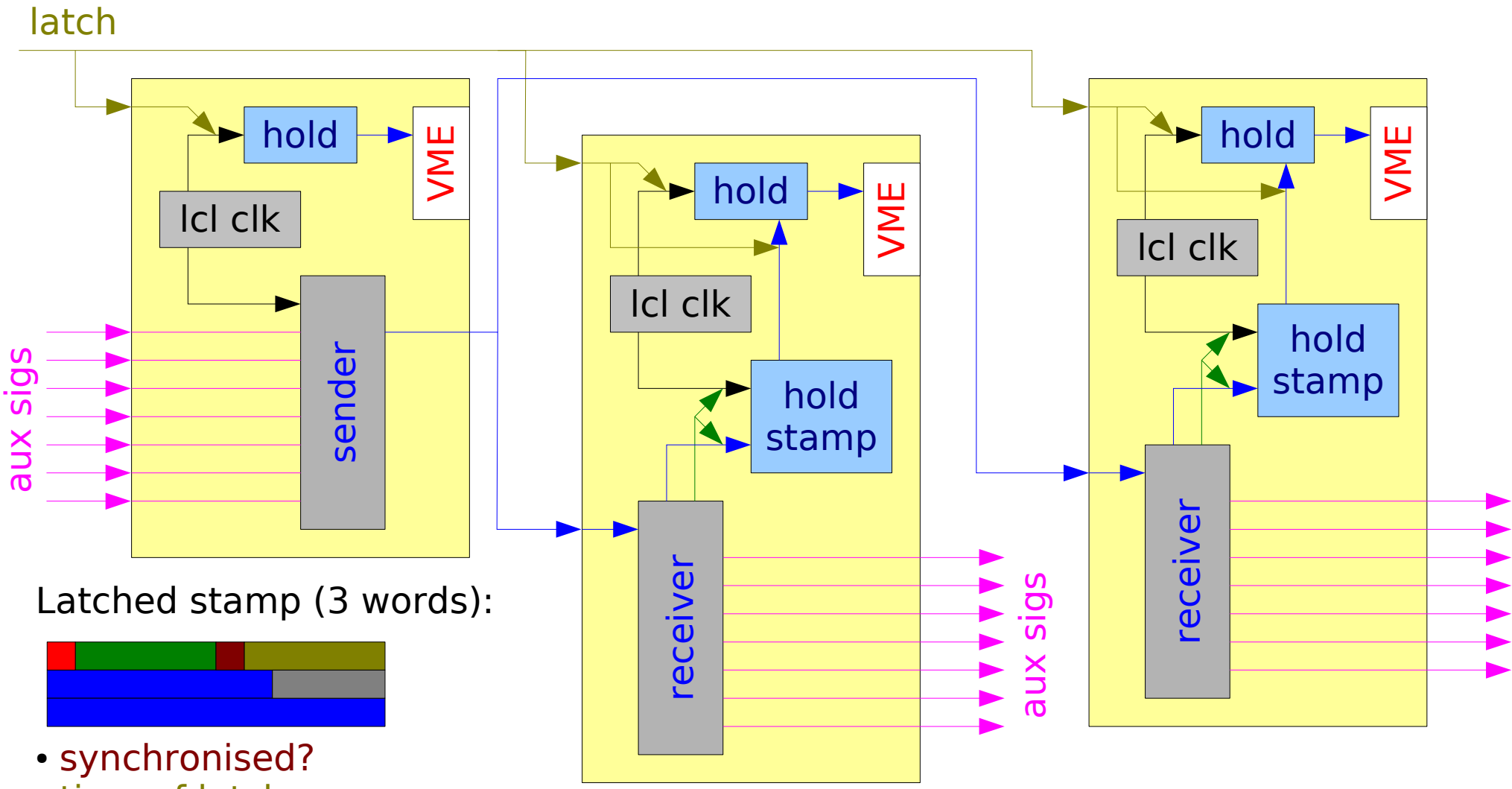




(2x128 pulses: start, 51 time bits, 7 aux bits, 8*(7+1) aux sigs, parity, end)

1 message: 82 µs

Serial timestamps with mux



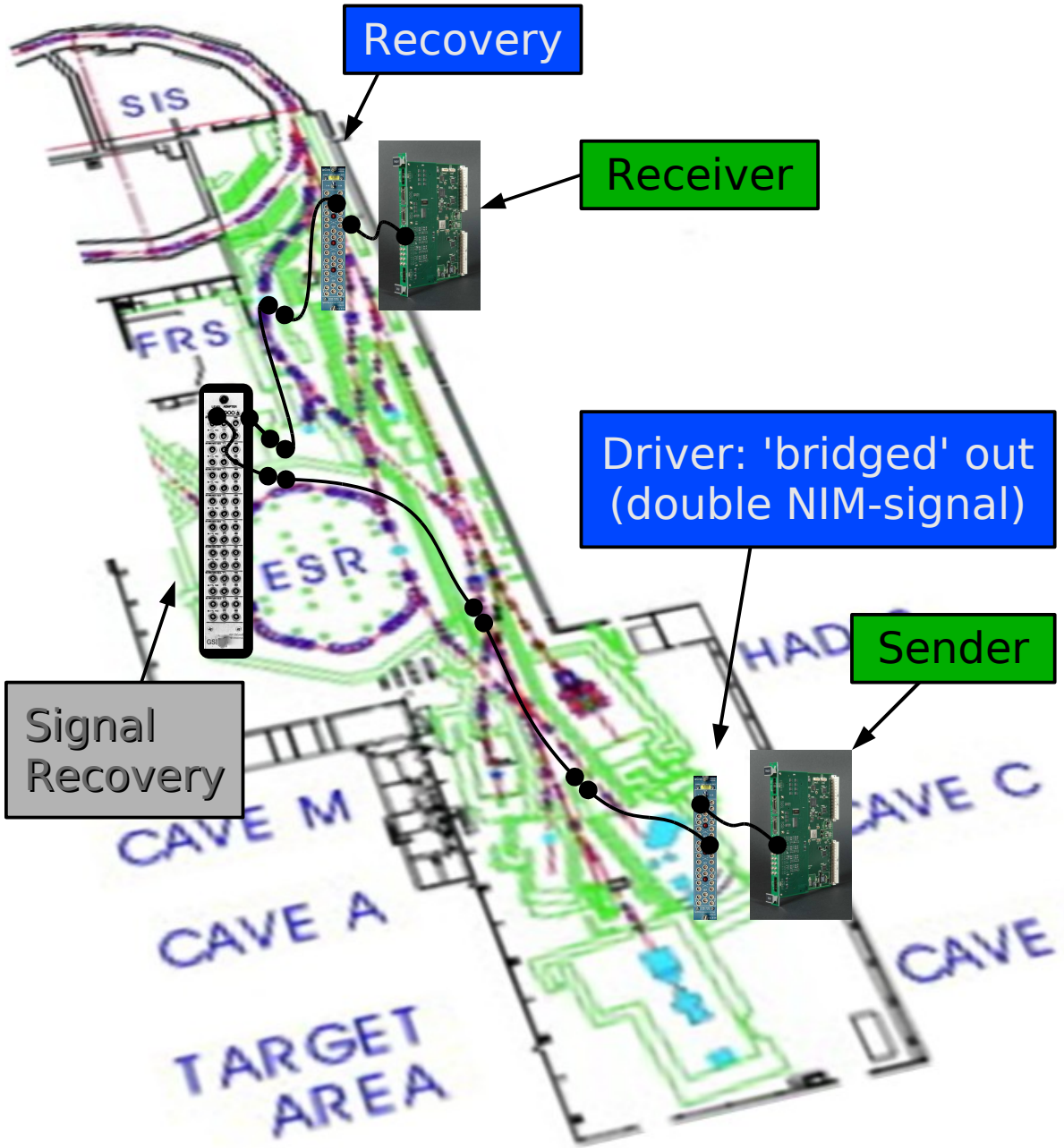
Latched stamp (3 words):

- synchronised?
- time of latch
- time of last reception
- last received time
- current drift-correction

$t = t_{stamp} + (t_{latch} - t_{reception}) + t_{driftcorr}$

Difference between local and remote clock frequency.

Serial timestamps Cave C - (FRS) - S2



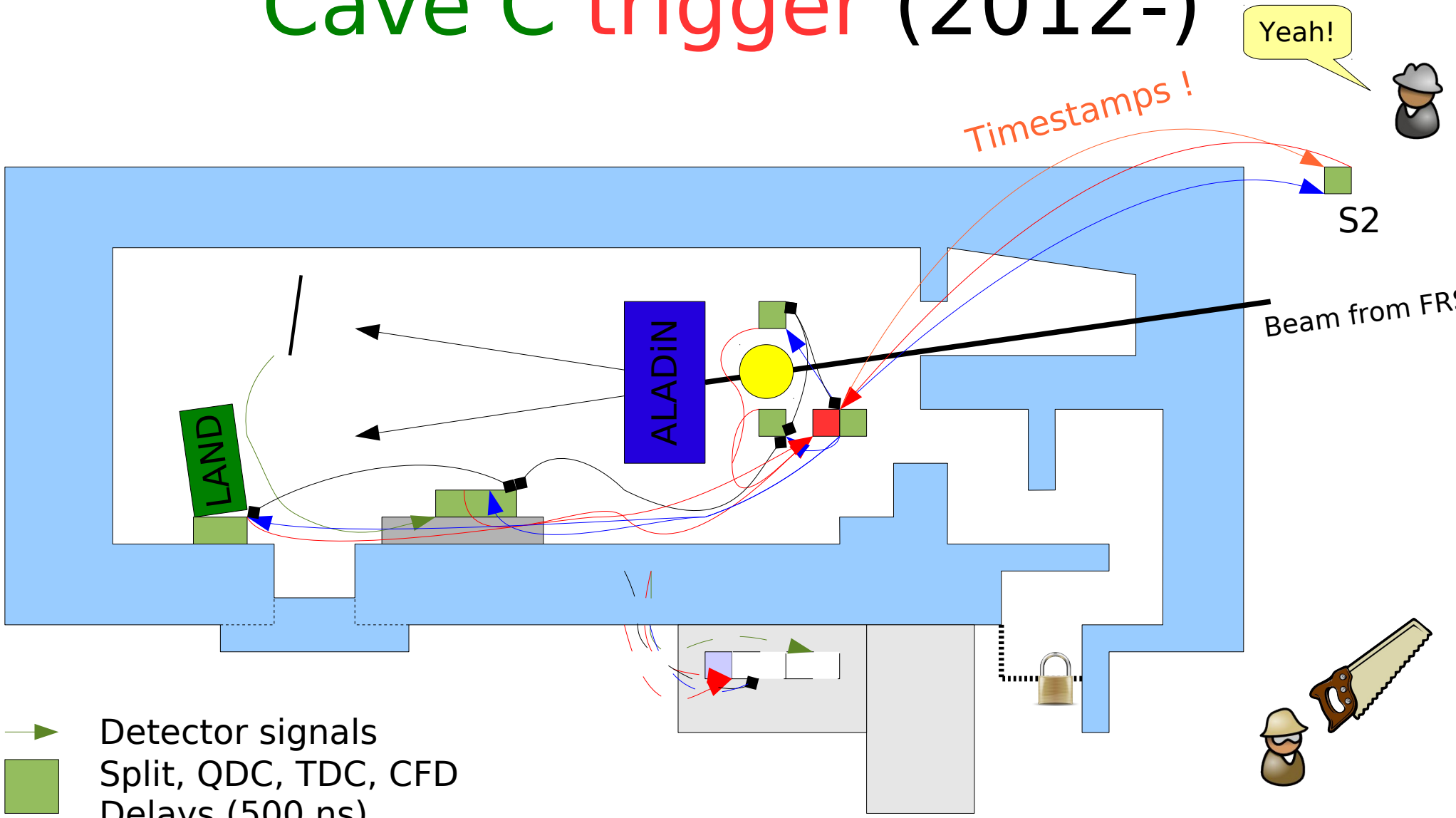
Features for easy deployment:

- Uni-directional protocol
- 1 cable of 'any' kind
- (a trigger also needed) (2^{nd} cable)
- No 'initialisation' - easy setup:

  1. Start sender,
  2. Follow signal (scope),
  3. Receiver auto-sync

- Loss of ≤ 3 protocol cycles → still in sync

Cave C trigger (2012-)



Detector signals

Split, QDC, TDC, CFD
Delays (500 ns)

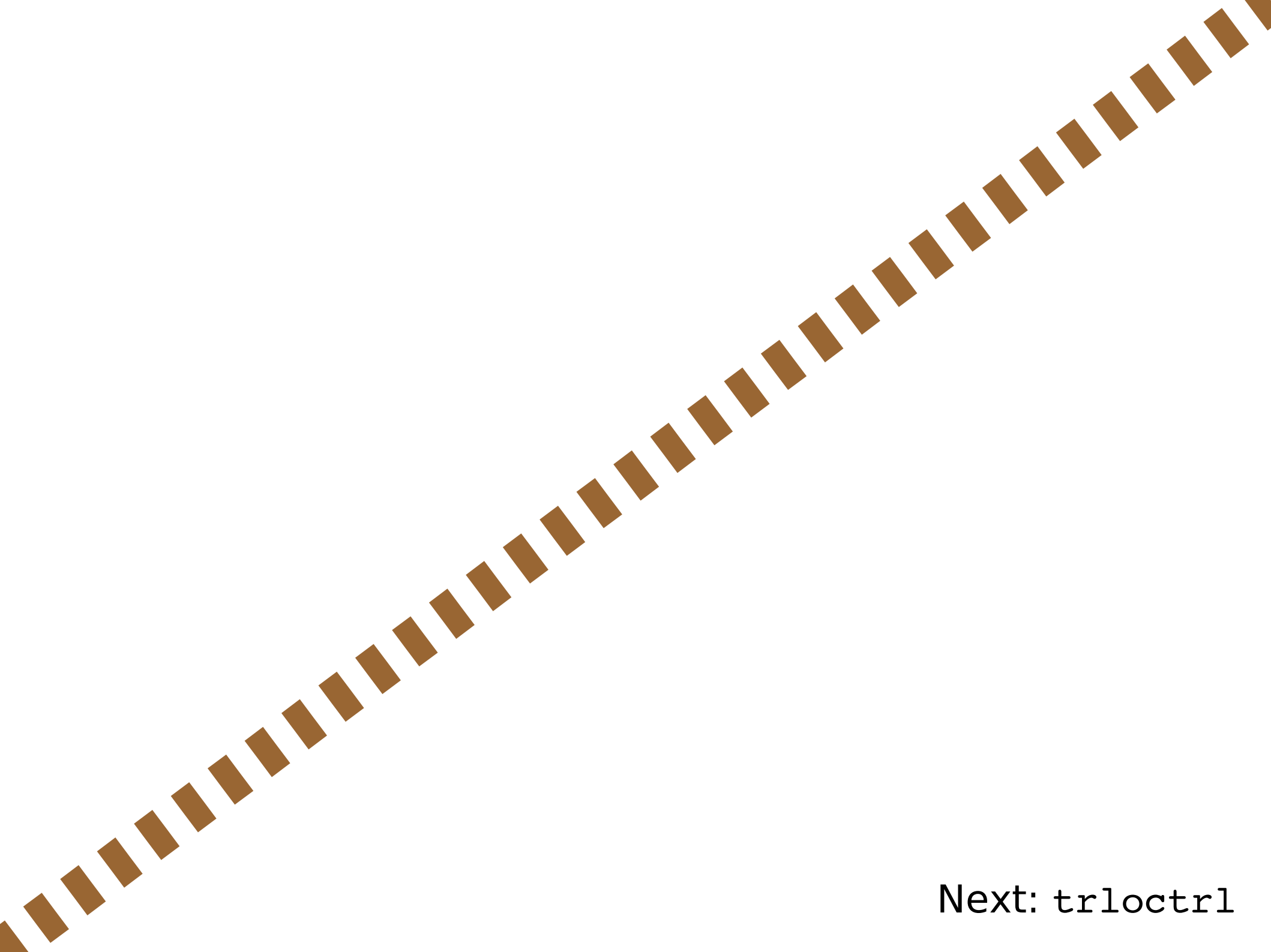
Detector triggers (<100 ns)

Trigger decision (~50 ns)

Master start (<100 ns)

Trigger bus

Separate DAQ @ S2, merge using time-stamps.



Next: `trloctrl`

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)
```

trloctrl – TRLO II command-line control

VME interface by named C structure elements is nice, but still cumbersome...

`trloctrl` already for LAND/R^{3}B setup, but setup specific

New tool with easy generic command line access:

```
trlo_ctrl "LEMO_OUT(2)=GATE_DELAY(1)"
```

```
COMMAND(wiring)
{
  DEADTIME_IN(1) = ECL_IO_IN(4);

  ECL_IO_OUT(1) = ENCODED_TRIG(1);
  ECL_IO_OUT(2) = ENCODED_TRIG(2);
  ECL_IO_OUT(3) = ENCODED_TRIG(3);
  ECL_IO_OUT(4) = ENCODED_TRIG(4);
```

Recurring operations (e.g. setup) in parsed configuration file:

```
trlo_ctrl --addr=1 --config=trlo_setup.trlo wiring
```

trloctrl - Command line

File with commands.

Execute cmd.

```
trlo_ctrl --addr=1 --config=trlo_setup.trlo wiring
```

Use: `alias trloctrl="trlo_ctrl --addr=1 --config=trlo_setup.trlo"`

```
trloctrl wiring
```

```
trloctrl --print-config
```

Dump TRLO II status.

```
trloctrl --clear-setup
```

Clear TRLO II.

```
trloctrl --mux-src-scalers
```

Debug scalers.

```
trloctrl "ECL_OUT(1)=ACCEPT_TRIG[1]"
```

Direct (without file)

```
trloctrl --show
```

Show all setup names.

```
trloctrl --ramtest
```

Test VME interface.

Example .trlo

(abridged)

```
/* Declare some variables. */
user_gate1 = ECL_OUT(1), ECL_OUT(2), ECL_OUT(4);
user_delay1   = 100 ns;
user_stretch1 = 1000 ns;

force_trig_no = 9;

/* Main setup command. */
COMMAND(wiring)
{
  DEADTIME_IN(1) = ECL_IO_IN(4);     /* TRIVA connections. */

  ECL_IO_OUT(1) = ENCODED_TRIG(1);
  ECL_IO_OUT(2) = ENCODED_TRIG(2);
  ECL_IO_OUT(3) = ENCODED_TRIG(3);
  ECL_IO_OUT(4) = ENCODED_TRIG(4);

  fast_busy_len = 1 us;      /* Busy before TRIVA deadtime. */

  GATE_DELAY(1) = MASTER_START;
  delay(1)   = user_delay1;
  stretch(1) = user_stretch1;
  user_gate1 = GATE_DELAY(1);
}

/* Utility command. */
COMMAND(mapgate1) { LEMO_OUT(2) = GATE_DELAY(1); }
```

MUX_DEST = MUX_SRC

setup = value | time (with unit)

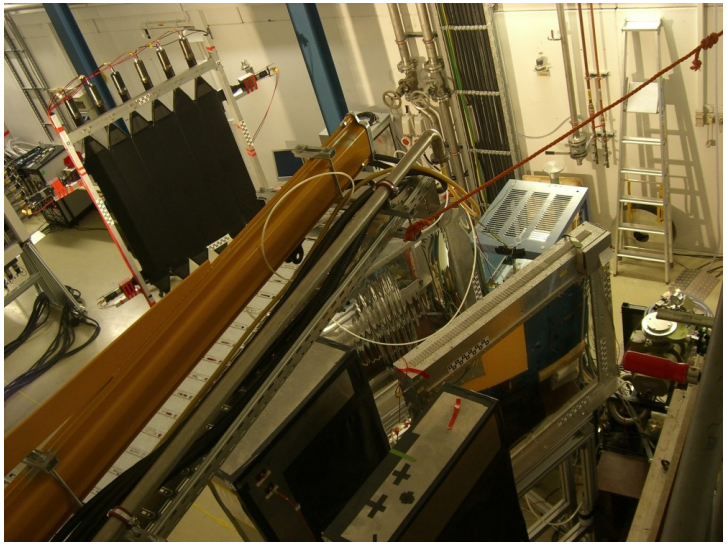
/* Prepared gates (delayed & stretched).
   Using variable values. */

TRLO II in operation:

S393 S306b S389 (2010)

S408, S405 (Oct 2011)

S412 (Apr – Jun 2012) (LAND/R^3B)



To come:

I83
IS512
S406
...

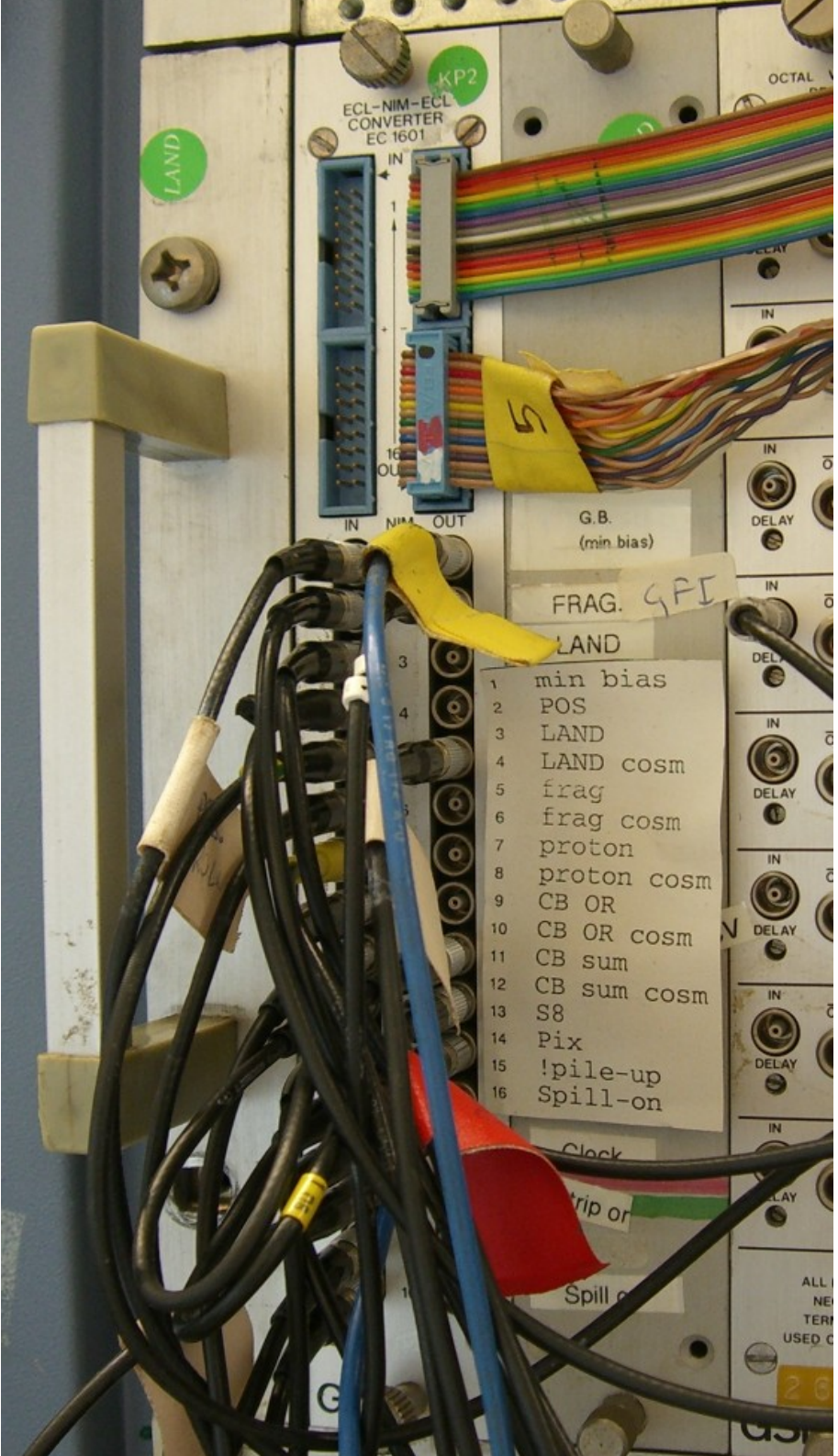
Preceded by intense *code* inspection (~300 kB) →

0 critical bugs found

3 minor bugs found in the wild
(none by users :-))

S371 (FIRST)
(V. Monaco, R. Sacchi) (Jul – Aug 2011)

S424 (AGATA-PRESPEC)
(D. Ralet, S. Pietri) (Apr – Jun 2012)



Finale!

Thank you!

FPGAs are FUN!

http://fy.chalmers.se/~f96hajo/trloii/

Live by the compiler timing messages!