Timing Distribution

Mikihiko Nakao (KEK IPNS)

July 7 (Tanabata), 2009

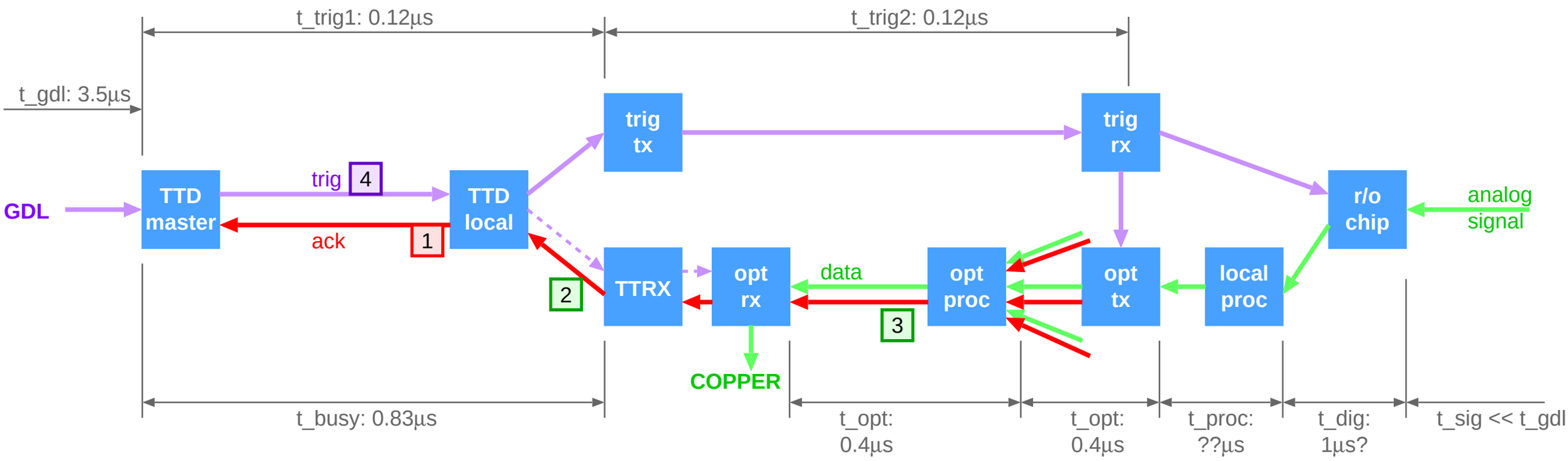
BelleII General Meeting

Requirements

- Trigger rate: max 30 kHz
- Deadtime fraction $\sim$ 3% at 30 kHz (already slightly violated)
- Key parameters
  - Readout clock frequency (f_{RCLK})
  - Depth of the ring buffer (N_{buf})
  - Minimum time interval between two triggers (t_{in})
  - Time for data transfer to the next stage (t_{out})
    (Example (SVD): $f_{\mathrm{RCLK}} = 32$ MHz, $N_{\mathrm{buf}} = 5$, $t_{\mathrm{in}} = 210$ ns, $t_{\mathrm{out}} = 30$ μs)
- Task
  - Distribute the readout clock to the frontend/FINESSE
  - Distribute the trigger timing
  - Synchronize the readout clock
  - Primitive control of the readout board
  - Primitive status collection from the readout board

Deadtimeless Flow Control (no update today)

- Readout status from frontend through COPPER to TTD
  - RocketIO + serialbus latency $\sim$ 1.5 μs
  - Status can be embedded in the RocketIO datalink using the "K character" of 8b10b encoding
- Pipelined trigger handshake scheme
  - Data integrity (no data-driven FIFO full handling)
  - TTD can issue N_{buf} (=5) triggers with at least t_{in} ($\sim$ 200 ns) interval before seeing the response



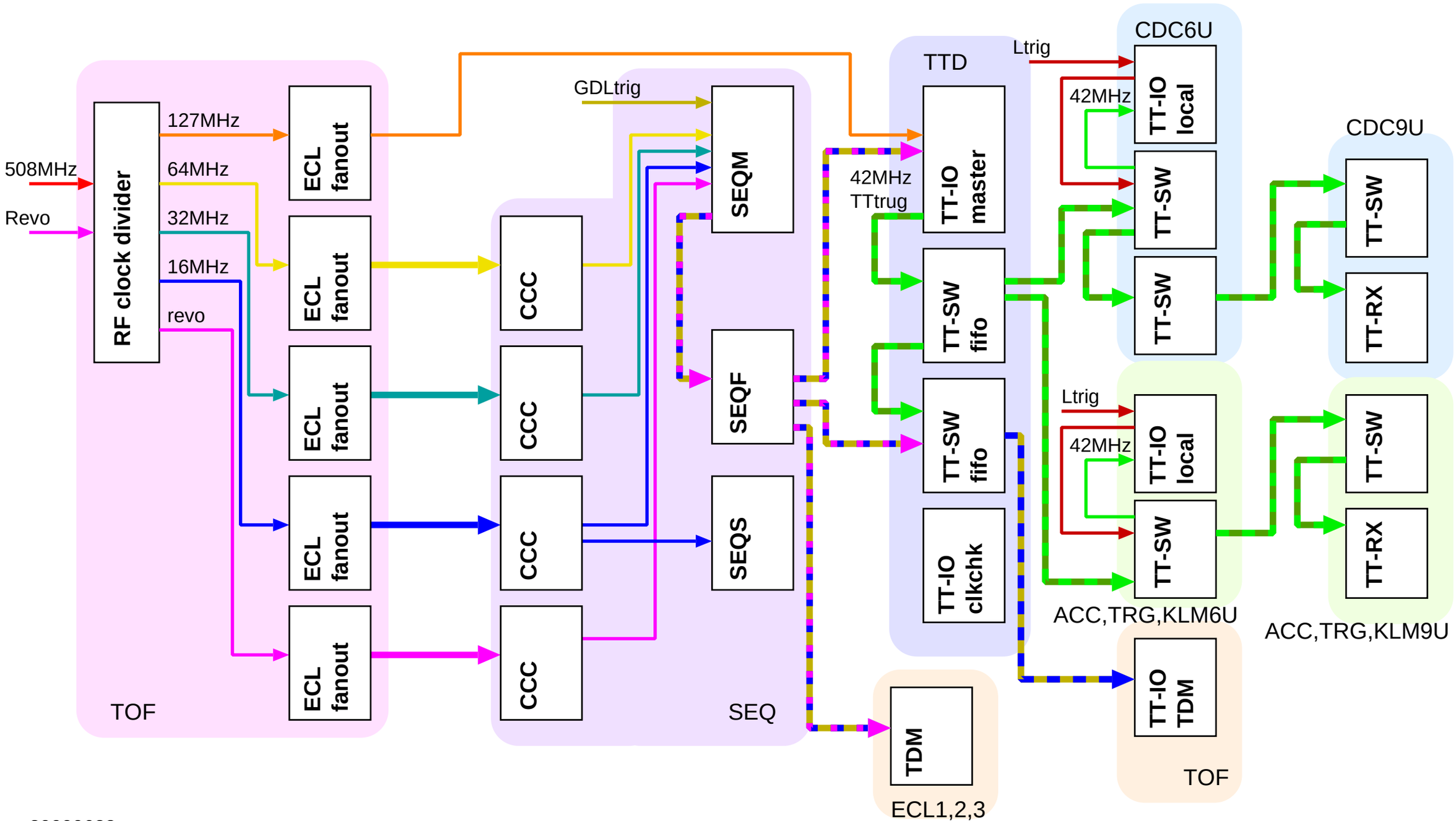
More on Clock Requirements

- Clock frequencies
  - 508 MHz — RF clock, too fast for most of electronics
  - 127 MHz — RF/4, TOP readout and GDL L1 timing
  - 64 MHz —
  - 32 MHz — SVD (APV25), CDC (proposal)
  - 42 MHz — COPPER serial link
- Jitter
  - Only TOP frontend requires ultra-small jitter O(10ps)
  - The Master Clock station naturally locates near TOP
- Clock distribution
  - 64 MHz (or 127 MHz) for clock distribution
  - Other frequencies will be locally generated

More on Trigger Requirements

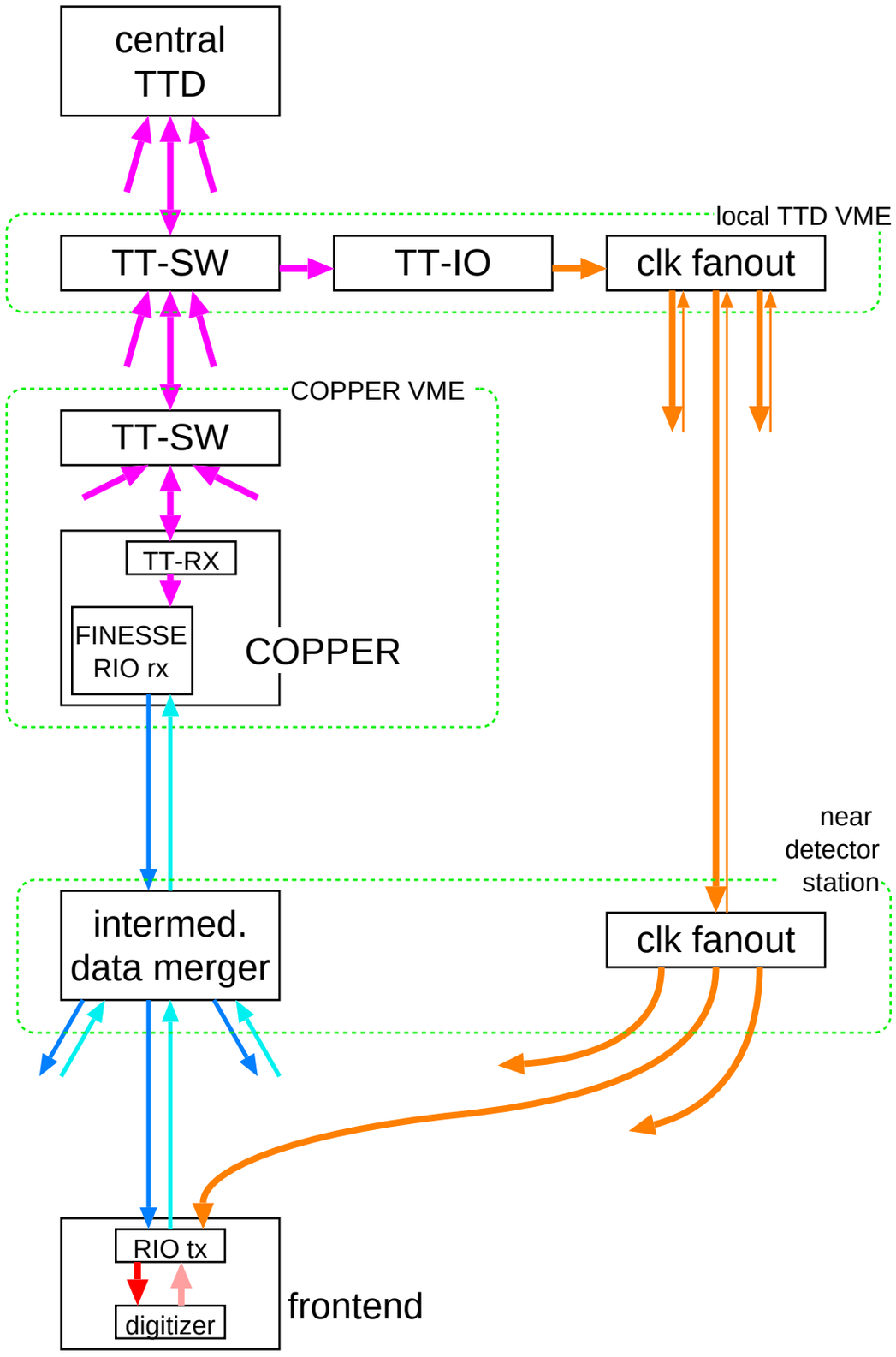
- For 32 MHz readout clock system
  - Trigger is a 32 ns pulse latched at the readout clock
  - No next trigger in next 5 clocks
- For 2 MHz readout clock system
  - Trigger is a 32 ns pulse latched at the readout clock
  - Next trigger can be within the same readout clock cycle

TTD clock distribution now



20090622

Timing Distribution Design (2008 version)



Signals

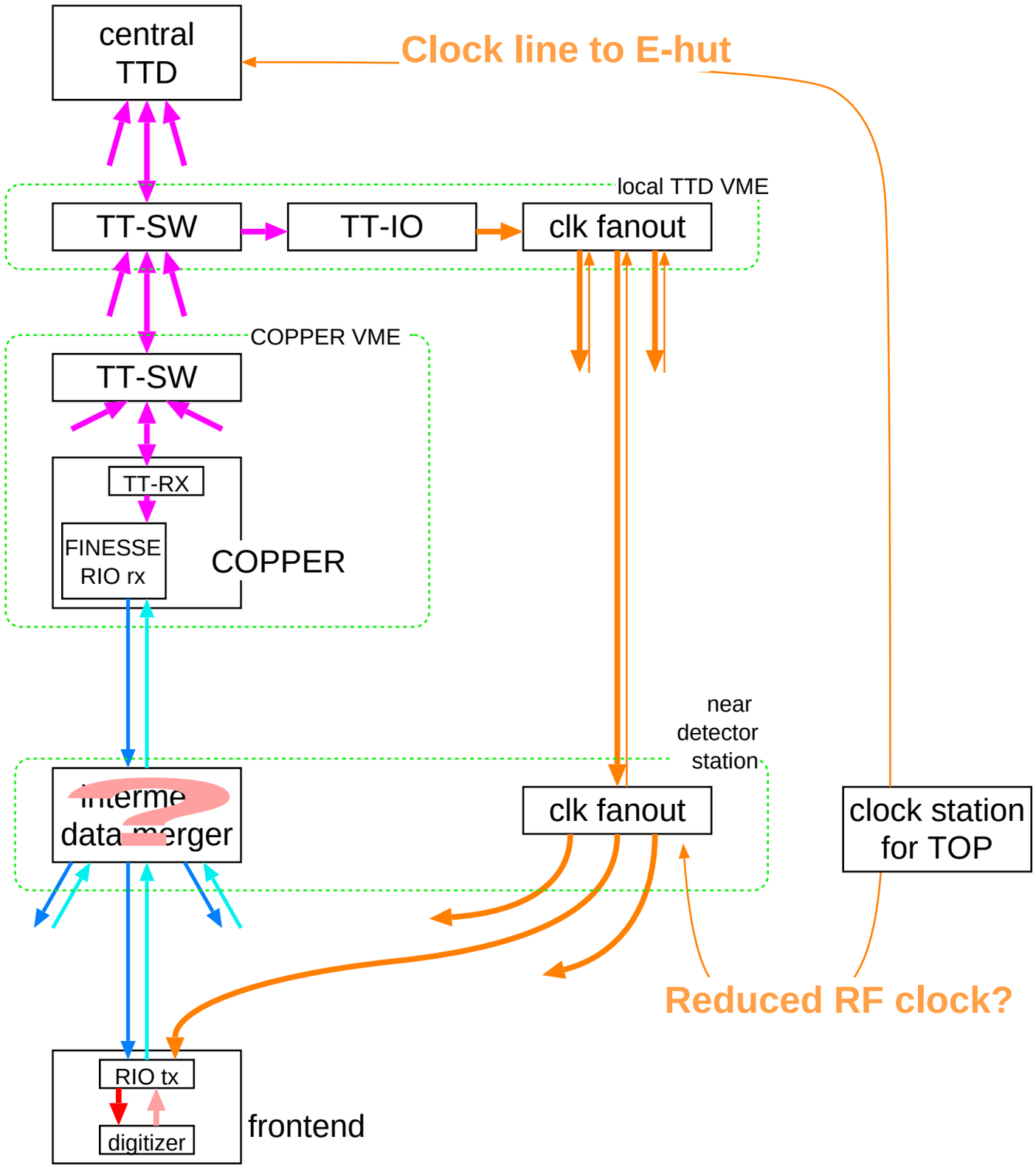
TTD signals (4-pair STP)
in CAT5 4-pair STP
(sclk, trig, serial-bus in/out)

CLK signals (4-pair STP)
sclk, trig, rclk (3 pairs)
and sclk-return (1 pair)
for phase monitor/adjust
between Ehut and detector

RIO signals (1 pair optical)
triginfo/cmd (COPPER → frontend)
data/status (frontend → COPPER)

Front end interface
(onboard or FPGA-internal signals)
rclk, trig, ctrl (RIO tx → digitizer)
data, status (digitizer → RIO tx)

Timing Distribution Design (2008 version)



Signals

TTD signals (4-pair STP)
in CAT5 4-pair STP
(sclk, trig, serial-bus in/out)

CLK signals (4-pair STP)
sclk, trig, rclk (3 pairs)
and sclk-return (1 pair)
for phase monitor/adjust
between Ehut and detector

RIO signals (1 pair optical)
triginfo/cmd (COPPER → frontend)
data/status (frontend → COPPER)

Front end interface
(onboard or FPGA-internal signals)
rclk, trig, ctrl (RIO tx → digitizer)
data, status (digitizer → RIO tx)

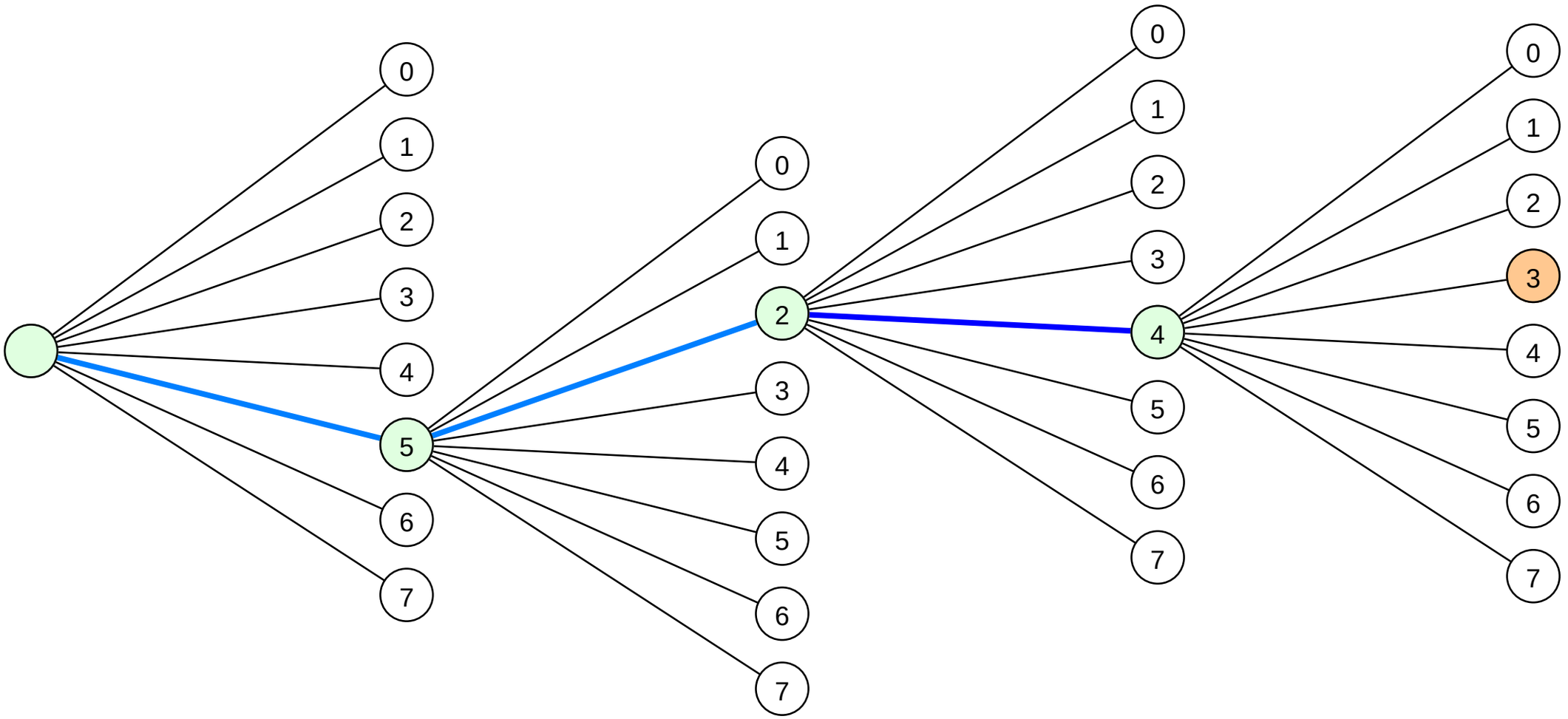
Timing Distribution Constraints

- Clock for TOP is generated near detector
  Can/should we distribute it near the detector without feeding back to E-hut?
- How many near-detector stations?
  How many frontend cards?
  Clock fanout has to be a cascadable TT-SW-like module?
- Trigger has to be at least segmented per detector basis for local run
- TT-RX will be unchanged
  (new version was developed, but existing ones will be also used)

Lessons from current TTD

- Clock generation (42 MHz) in FPGA
  - Mixing 42 MHz and 64 MHz clocks is possible, but painful
  - Would like to avoid 42 MHz for the readout clock
  - 42 MHz is still locally generated in COPPER/TT-RX
- Serial-link on CAT5 cable
  - Useful to make a compact system
  - Latency, length limit (10m) for 508 Mbps
  - Can be slower (O(100) Mbps) if no need for busy collection
- Flexible signal routing through FPGA
  - Less controllable timing
  - Cause of the timing shift problem?
    (plan to investigate during this summer shutdown)

Serial-link based point-to-point control



- Current TTD has a functionality to do point-to-point control
- Application: to reset a single COPPER, collect error status
- Not utilized yet, plan to implement during summer

Technologies for new TTD

- Ultra low-jitter clock fanout chips (e.g. from Micrel)
- Xilinx Virtex5LX(T) FPGA
  - PLL in addition to DCM (digital clock manager)
  - IODELAY
  - DCI (digitally controlled impedance)
  - 10-bit ISERDES / OSERDES (serial-link besides RocketIO)
  - Expensive...
- Xilinx Spartan-6LX(T) FPGA...
  - Cheaper alternative, all of functions are available
  - exception: 8-bit ISERDES / OSERDES
  - Datasheet just became available 10 days ago, chips will be available by the end of this (fiscal) year

Test with Virtex5

- DCM, PLL, IODELAY, DCI, ISERDES, RocketIO(GTP) can be tested
- RocketIO through RJ-45 is also available, slow serial-link performance can be compared between ISERDES and RocketIO

Summary / Near-term Schedule

- Refinement of the current TTD in Ehut
  - Firmware update / unification (CDC is now using an old version)
  - Timing measurement at various points
  - Timing monitor (CDC/TOF trigger timing) at TRG-COPPER
- Test with Virtex5
  - New feature will be tested with MGT-FINESSE-PROTOTYPE
- Study of deadtimeless flow control
  - Can be tested with existing TTD modules
- Module Development
  - Probably start with a new TT-IO prototype with Virtex5
  - Hopefully new version of TT-SW, TT-IO and clock distributor with Spartan6