Common readout and timing distribution

Mikihiko Nakao (KEK IPNS)
July 4, 2008
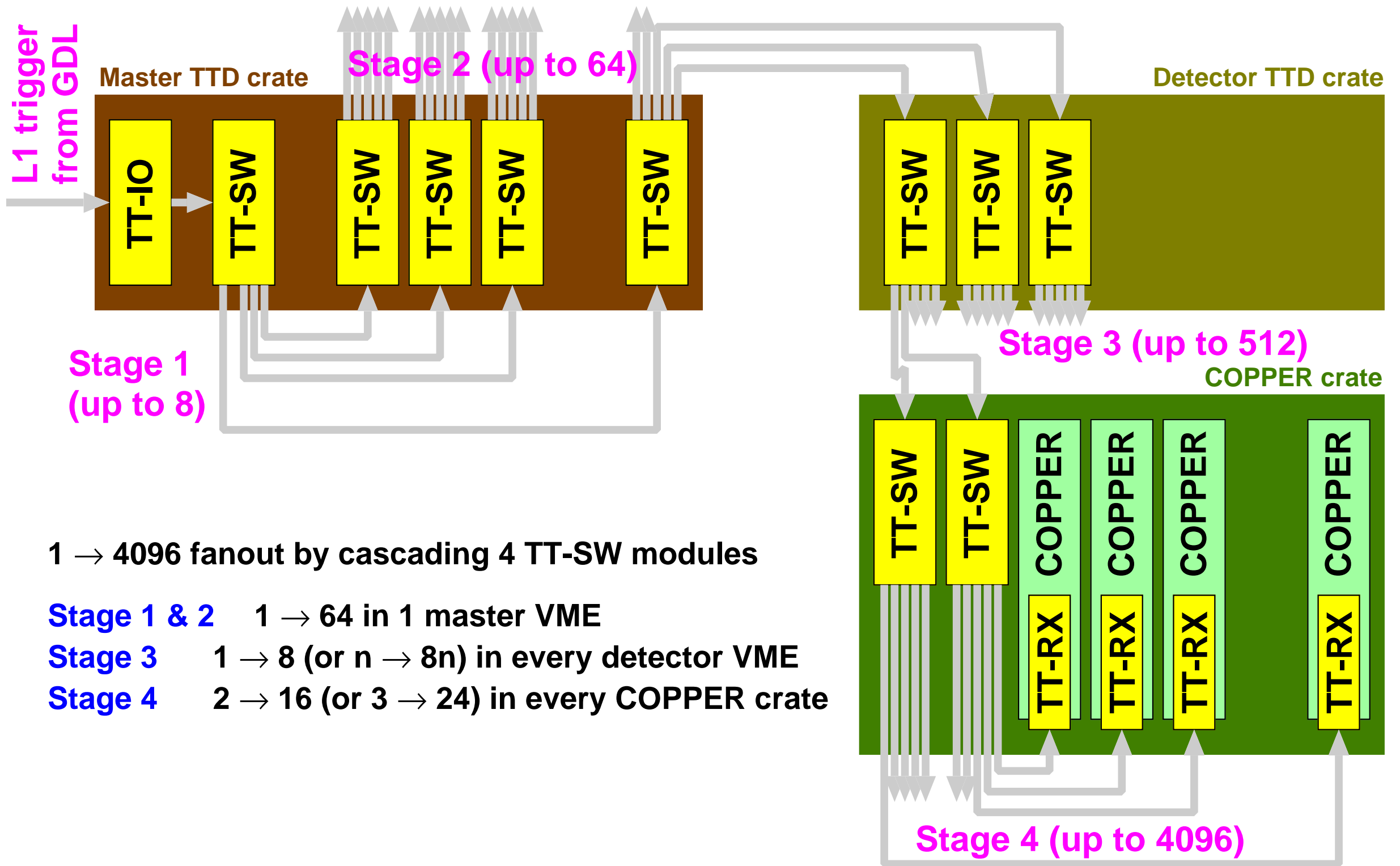2nd Open Meeting for proto SuperBelle Collaboration
mikihiko.nakao@kek.jp

Baseline design

- Capable at 30 kHz trigger rate (nominal maximum)
- Event-by-event trigger-busy handshake,
  $\sim 1\mu s$ deadtime is okay, but the current busy collection scheme already requires $\sim 1\mu s$

  ➡ No extra time to hold busy by the readout system, so the readout system should have a large enough buffer in order to avoid a long busy time

- Pipelined digitization, timing w.r.t. the system clock
  - Most detectors do not care a clock jitter of O(100ps)
  - TOP detector requires O(10ps) time resolution

Timing to COPPER

L1 trigger from GDL

Master TTD crate

Stage 2 (up to 64)

Detector TTD crate

TT-IO TT-SW TT-SW TT-SW TT-SW TT-SW

TT-SW TT-SW TT-SW

Stage 1 (up to 8)

Stage 3 (up to 512)

COPPER crate

TT-SW TT-SW COPPER COPPER COPPER COPPER

TT-RX TT-RX TT-RX TT-RX

1 → 4096 fanout by cascading 4 TT-SW modules

Stage 1 & 2   **1 → 64 in 1 master VME**

Stage 3   **1 → 8 (or n → 8n) in every detector VME**

Stage 4   **2 → 16 (or 3 → 24) in every COPPER crate**

Stage 4 (up to 4096)

Common Readout

Q-to-T + FASTBUS TDC was an excellent technology in 1995

- Problems: deadtime, old and decaying, and availability
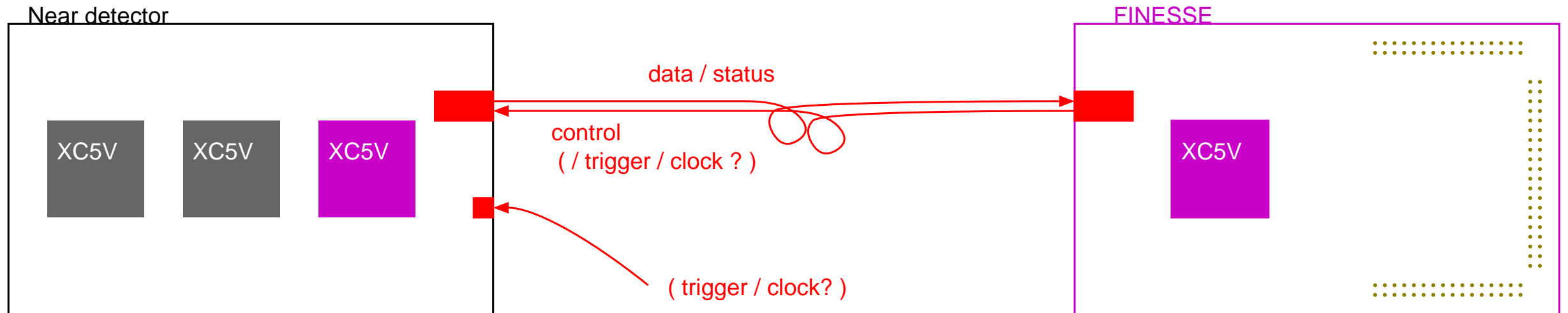- Solution: COPPER?
  Q-to-T modules were different for each detector, so could be the FINESSE module?

It turns out that many of the detector people are willing to digitize data near the detector — this makes COPPER just a digital-to-network converter.
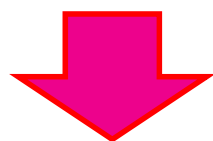
- New questions
  - Could the digital transfer link (and FINESSE) be common?
  - How to deliver timing signals to near-detector sites

Remote FINESSE

Near detector

FINESSE

XC5V   XC5V   XC5V

data / status

control
(/ trigger / clock ?)

(trigger / clock?)

XC5V

- Extend the FINESSE function over an optical link + timing line

- Need a very simple interface
  similar to FINESSE/COPPER interface at the remote card

- Colored part could be standardized

- O(25ch) (or O(50ch)) per such a card?

  ➡ 600 (or 300) links for a 15k ch CDC (150 or 75 COPPERs)

Technology choice

- Xilinx RocketIO at both ends

- RocketIO GTP (up to 3.75 Gbps) or GTX (up to 6.5 Gbps)? GTP is good enough (FINESSE doesn't have such a bandwidth)

- Optical tranceivers — 2.5 Gbps SFP tranceivers?
  - 2.5 Gbps $\Rightarrow$ ¥30,000.-
  - 4 Gbps $\Rightarrow$ ¥50,000.-
  - 10 Gbps $\Rightarrow$ >¥200,000.-

- FPGA choice — Virtex5 LX30T? (¥35,000 / chip)

  ➡ already >¥100,000 /link, faster options are unrealistic

Protocol

- Minimum requirements
  - Simple protocol to encapsulate a variable-length data
  - Link establishing scheme
  - Error detection
  - Recovery from error without stopping data taking
  - Predictable latency
  - Command / status embedding between the data packets
- Desirable
  - Clock recovery
- Not needed
 - Error correction and retransmission

Timing distribution to detector

- An extra set of signals to the detector
  — trigger, clock, clock-return, reduced-clock
  (4 pairs of COPPER cables, i.e., standard STP cable is suitable)

- Need only one cable from Ehut to a site
  A new timing distribution module has to be developed

- Clock issues
  - Correction for the clock phase drift (clock-return $\Rightarrow$ PLL)
  - Can't we just receive the KEKB clock at the detector?
  - No good measurement tool in hand

- Trigger issues
  - Trigger timing is better distributed through a dedicated line, unless SERDES latency is tolerable
  - Need another path to send the trigger id

PLL

`http://indico.cern.ch/conferenceDisplay.py?confId=28233`
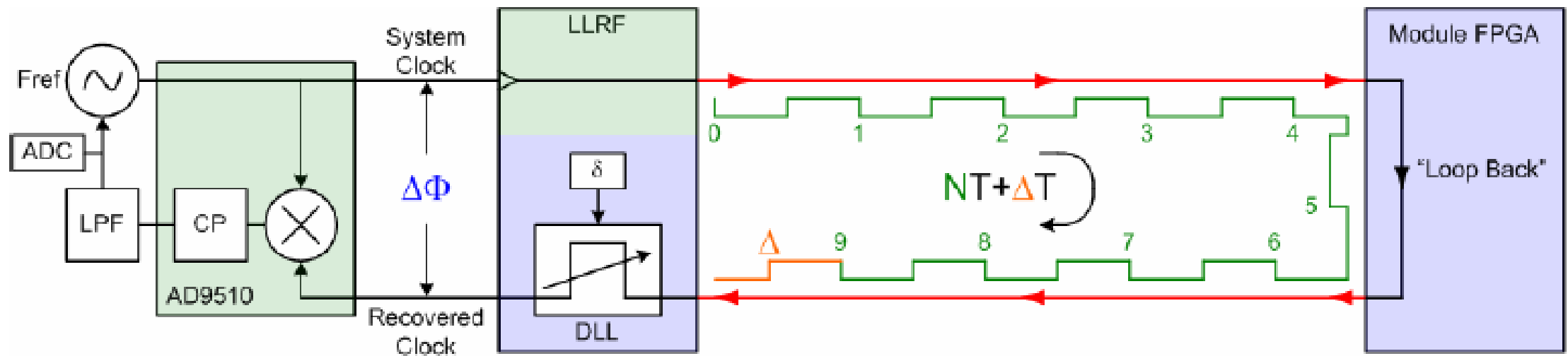
Requirements at Receiver:
• FPGA with high speed serial I/O
• External PLL with good VCO

**Requirements at Master:**
• FPGA with multiple high speed serial I/O's

What you get:
• Frequency Recovery (at noise quality of VCO)
• Line length drift compensation
• Multiple FPGA clocks aligned to 1/256 of a cycle (16 ps @ 240 MHz)
• Time accurate to 1/256 of a cycle (for stamping and event scheduling)

● It is possible to recover a 240MHz clock from a 2.4Gbps rocketIO (no need of external link?)

● Maybe we can consider running RocketIO at KEKB RFx4 (2.032 Gbps)

Intermediate merger?

- We may want an event building at the RocketIO level, to reduce number of fibres from the detector to COPPER

- A-RICH would need such a system

- No detail has been discussed

central
TTD

Signals

local TTD VME

TT-SW → TT-IO → clk fanout
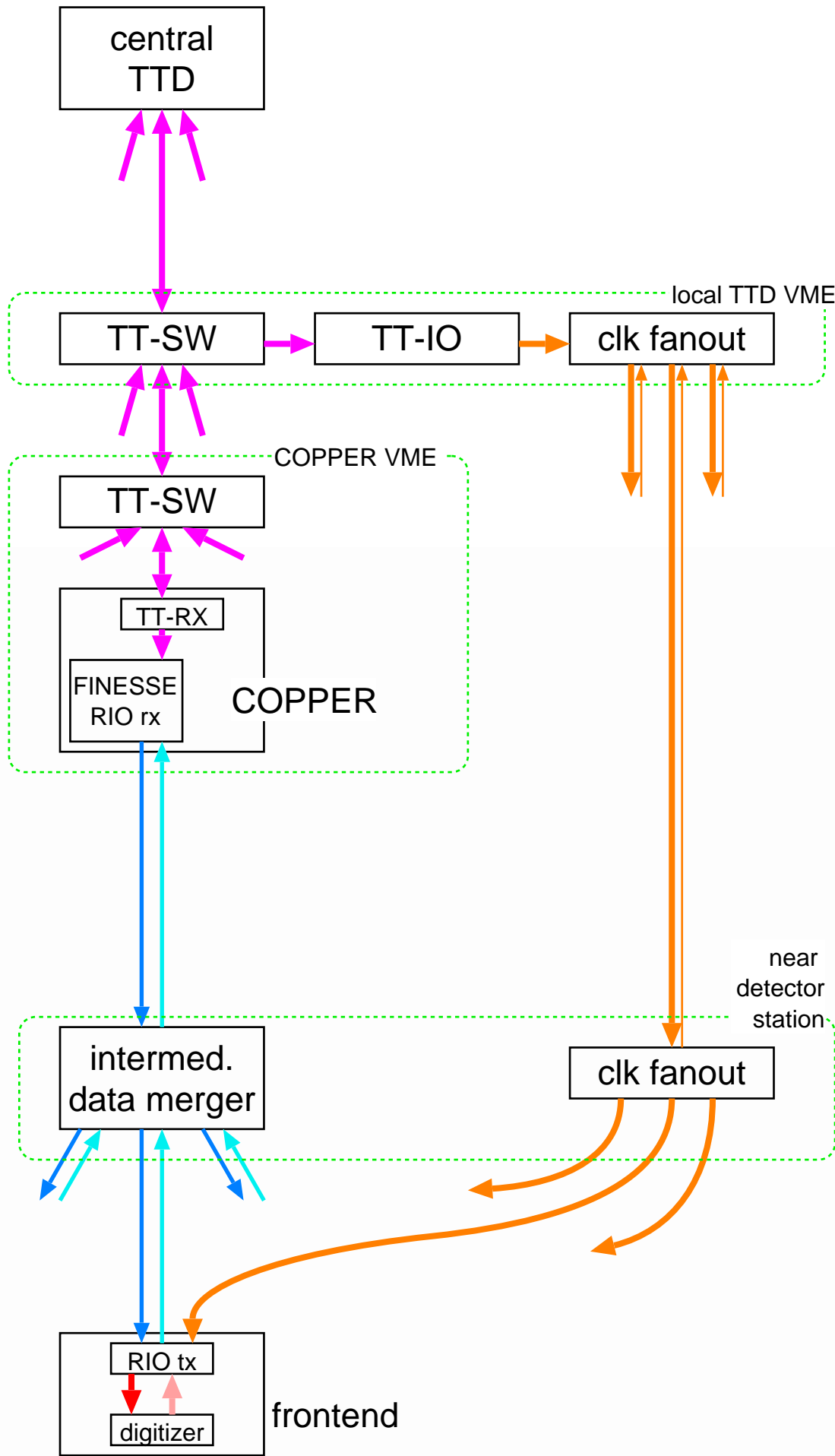
TTD signals (4-pair STP)
in CAT5 4-pair STP
(sclk, trig, serial-bus in/out)

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

COPPER VME

TT-SW

TT-RX

FINESSE
RIO rx   COPPER

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)

near
detector
station

intermed.
data merger   clk fanout

RIO tx

digitizer   frontend

Common readout and timing distribution — M. Nakao — p.11

Plans

- RocketIO tests
  - A Virtex2Pro test board is in hand (with a test firmware)
  - TT-IO module has a SFP port for RocketIO
    so far, not successfully tested $\Rightarrow$ a good example to
    further understand RocketIO
  - Buy a Virtex5 test board if needed
  - Built a Virtex5 test board
- Need to collect detector requirements

  - Number of channels, typical data size, latency
  - Digitization clock
  - Required time jitter
- Need to decide the buffer size for deadtime free data link