Timing distribution for SuperBelle DAQ

Mikihiko Nakao (KEK, IPNS)

Dec 12, 2008

Open Collaboration Meeting

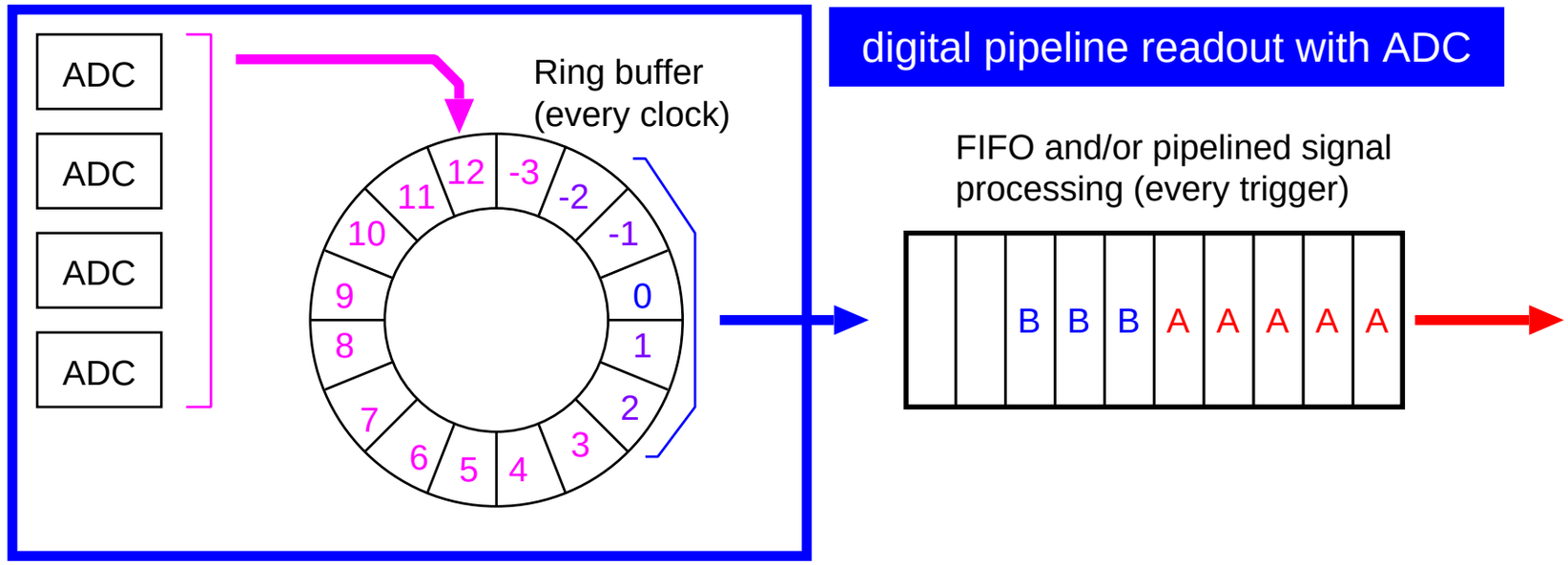
mikihiko.nakao@kek.jp

Requirements / Wishes

- Trigger rate
  - 10 kHz (nominal) and 30 kHz (maximum) was decided when the target luminosity was 1×10^{35} cm^{-2}s^{-1}
  - We now have ~400 Hz at 1.6×10^{34} cm^{-2}s^{-1}
    - ➡ 20 kHz if it scales with luminosity
  - 20 kHz (nominal) and 40 or 50 kHz (maximum) would be a more suitable requirement (???) unless the rate is guaranteed by the trigger group
- Readout deadtime
  - Hope to have "deadtimeless" readout scheme
    - ➡ "<3%" deadtime would be a target ("10%" deadtime would not be allowed)
  - Back-end should be designed to avoid additional deadtime due to back pressure (buffer size/depth)

Pipeline readout

- Concept
  - Driven by clock and trigger

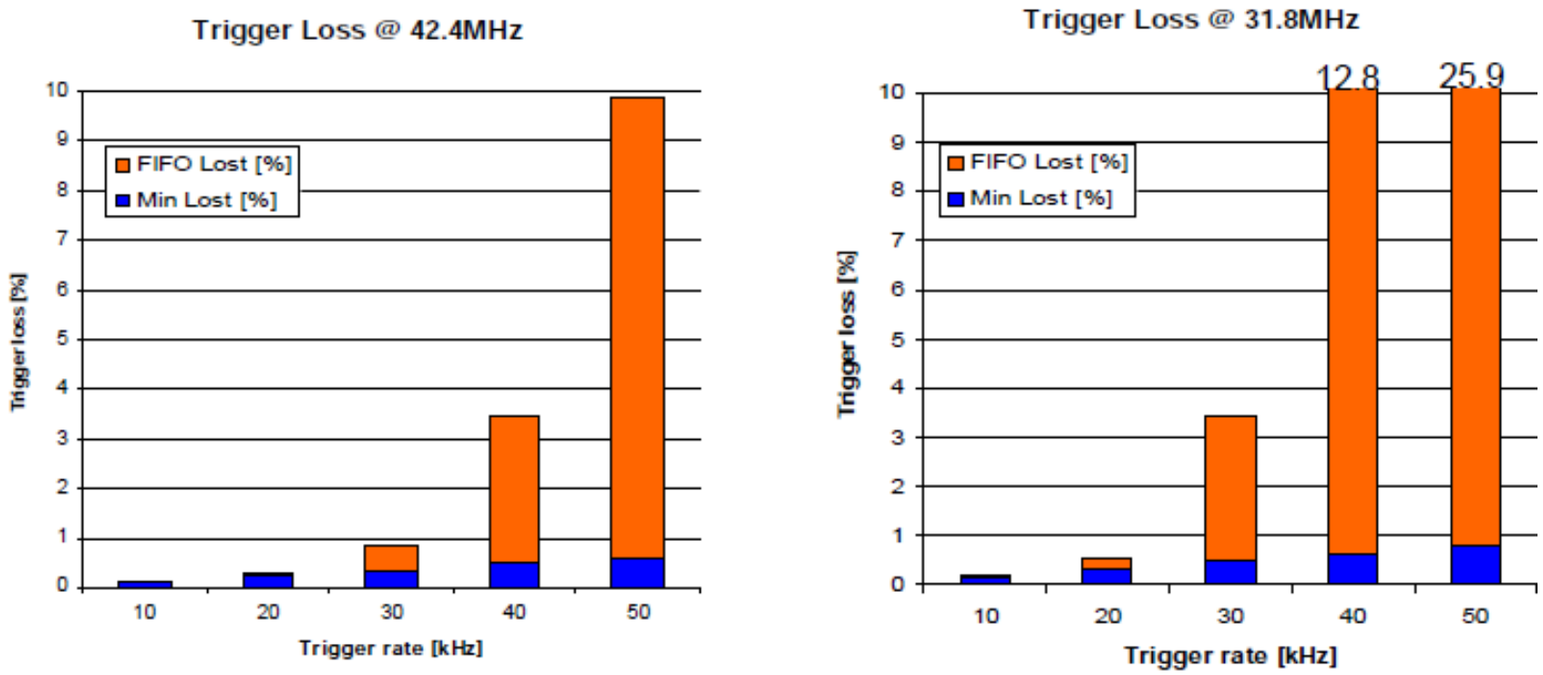


- Key parameters
  - System clock frequency (f_{SCLK})
  - Depth of the ring buffer (N_{buf})
  - Minimum time interval between two triggers (t_{in})
  - Time for data transfer to the next stage (t_{out})
- Examples
  - SVD (APV25): f_{SCLK} =42 MHz, N_{buf} =5, t_{in} =140 ns, t_{out} =20 μs
  - ECL (barrel): f_{SCLK} =2 MHz, ...

Impact on deadtime

- Four parameters (f_{SCLK}, N_{buf}, t_{in}, t_{out}) determine the deadtime characteristics
  - e.g., 31.8 MHz clock for APV25 is unpreferable
  - For 20 kHz trigger rate, 31.8 MHz clock is OK
- Outer detectors should have better deadtime characteristics (if not, please let us know immediately!)

APV Trigger Simulation (2)

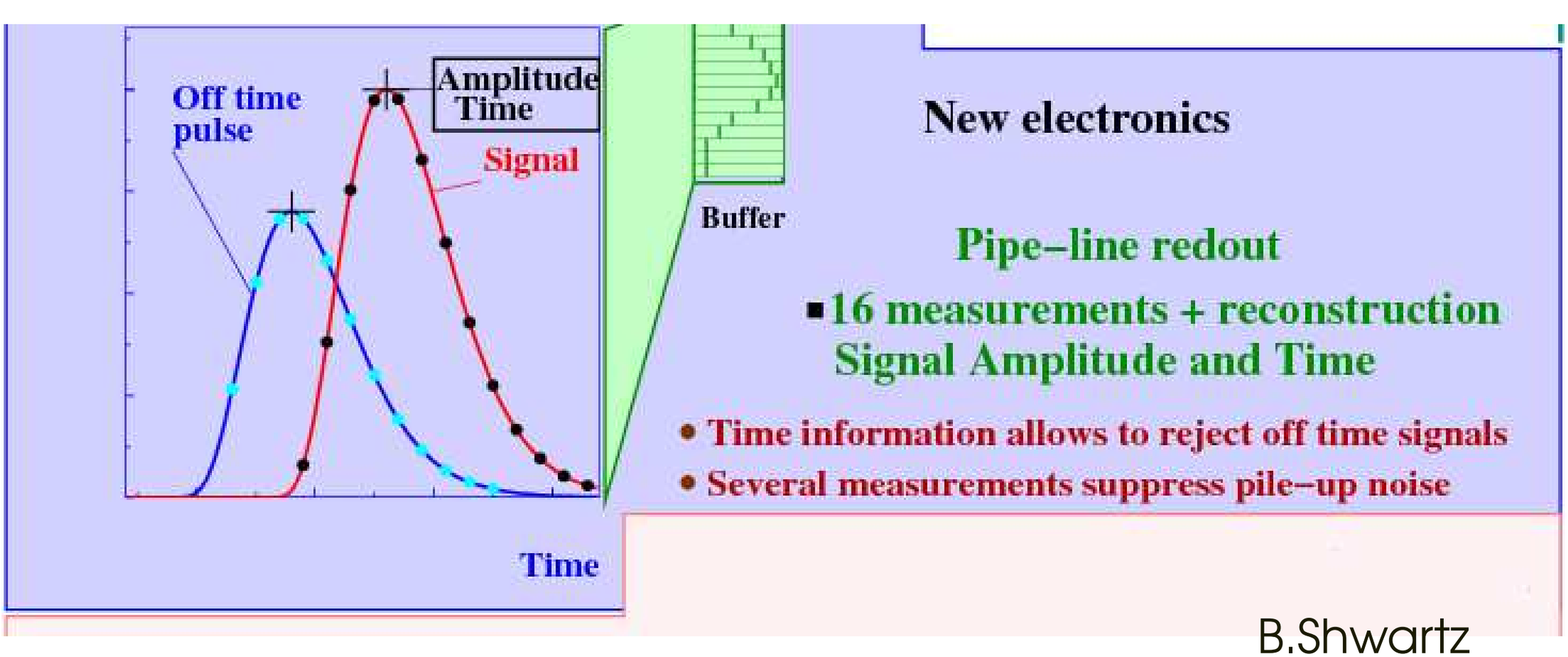


- **Min Lost**: trigger restriction (1) = too little distance
- **FIFO Lost**: trigger restriction (2) = too many pending readouts

(M. Friedl)

Impact on slow readout

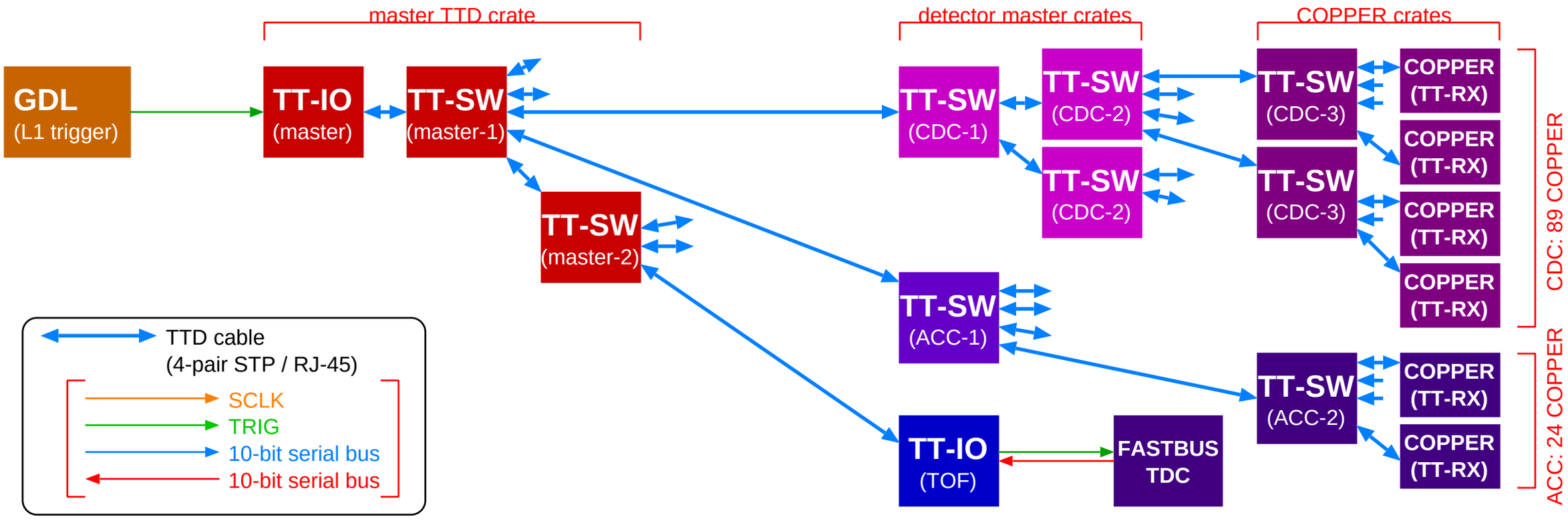
- Readout time is expected to be slow in ECL (and PXD)
  - Minimum buffer separation of 500 ns $\times$ 16 sampling (8 μs)
  - Two or more trigger have to share the same sample
  - For $t_{\rm in} < 500$ ns, the same samples have to be read out could be separated offline ($\sigma(t) \sim 100$ ns for 5 MeV)



B.Shwartz

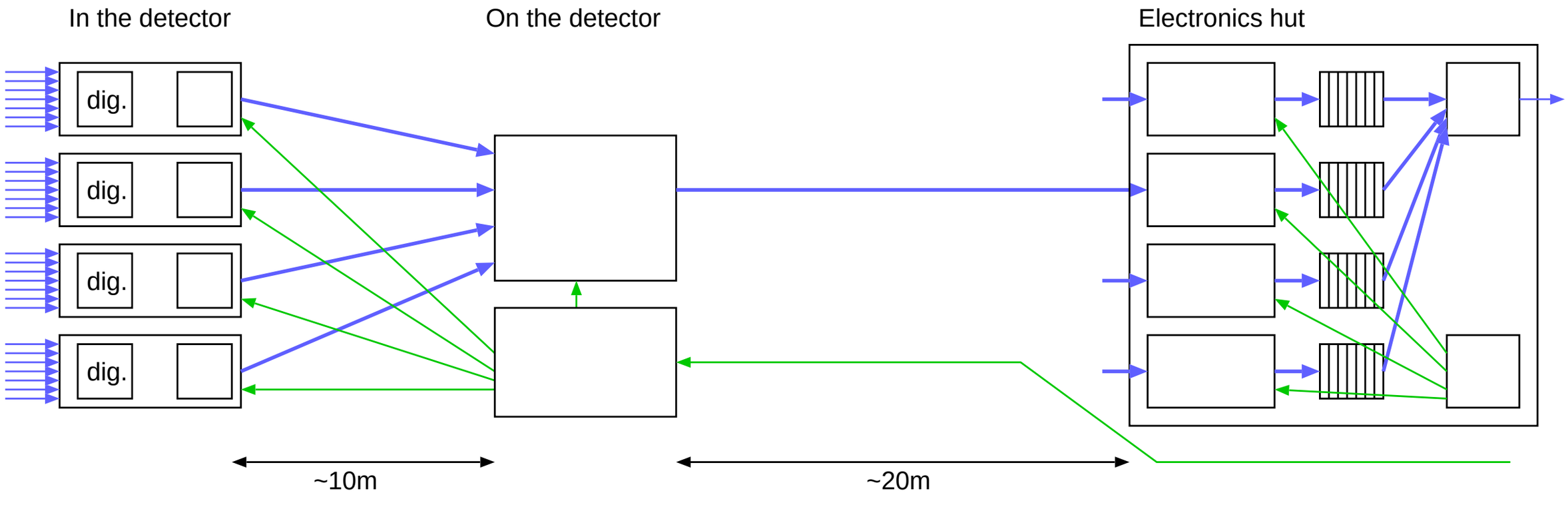
Timing distribution 1 (to COPPER)

- Trigger and system clock (42 MHz) distributed to all COPPER through TTD system (TT-IO $\Rightarrow$ (TT-SW)^n $\Rightarrow$ TT-RX)
- Modules are in operation, will also be used in Super Belle
  Common to every detector
- Serial-bus based control (BUSY signal, reset, etc)
- Trigger-busy handshake at Belle, need something better



Timing distribution 2 (to frontend)

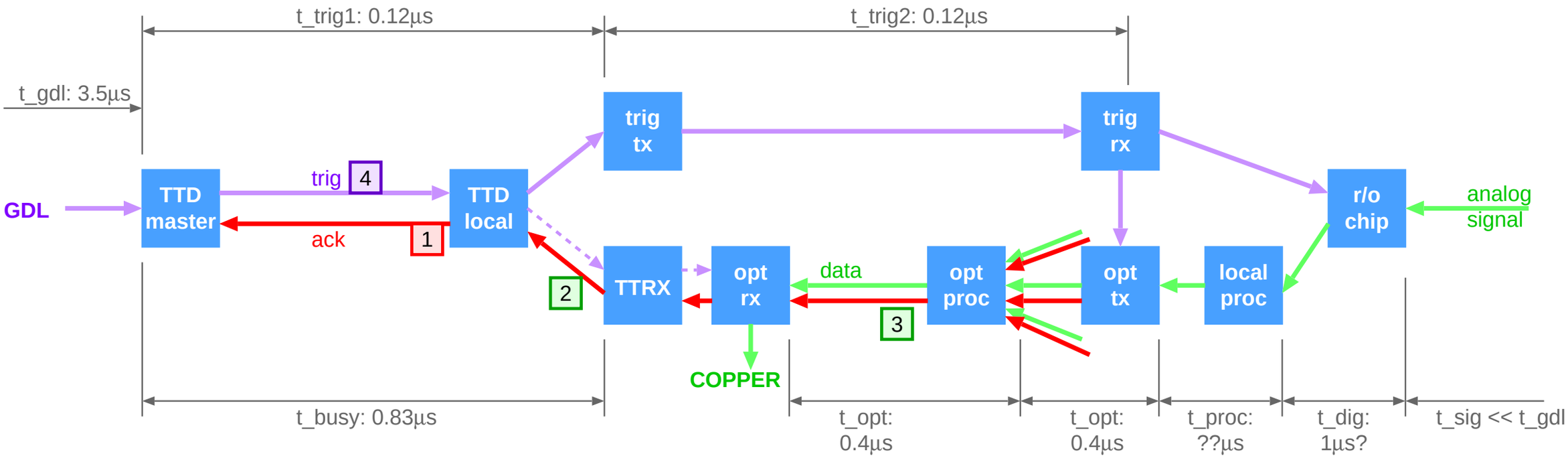
- Most of digitization in/near the detector ➡ Unification
- No COPPER crates outside the E-hut
  No dedicated BUSY collection line from front-end
- Data transfer and timing distribution provided by DAQ group
  Timing signals: clock (SCLK), trigger (TRIG), revolution (REVO)
  (but SVD will distribute by themselves)



TTD ➡ external timing dist. ➡ Front-end

Pipelined trigger distribution

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



System clock (jitter)

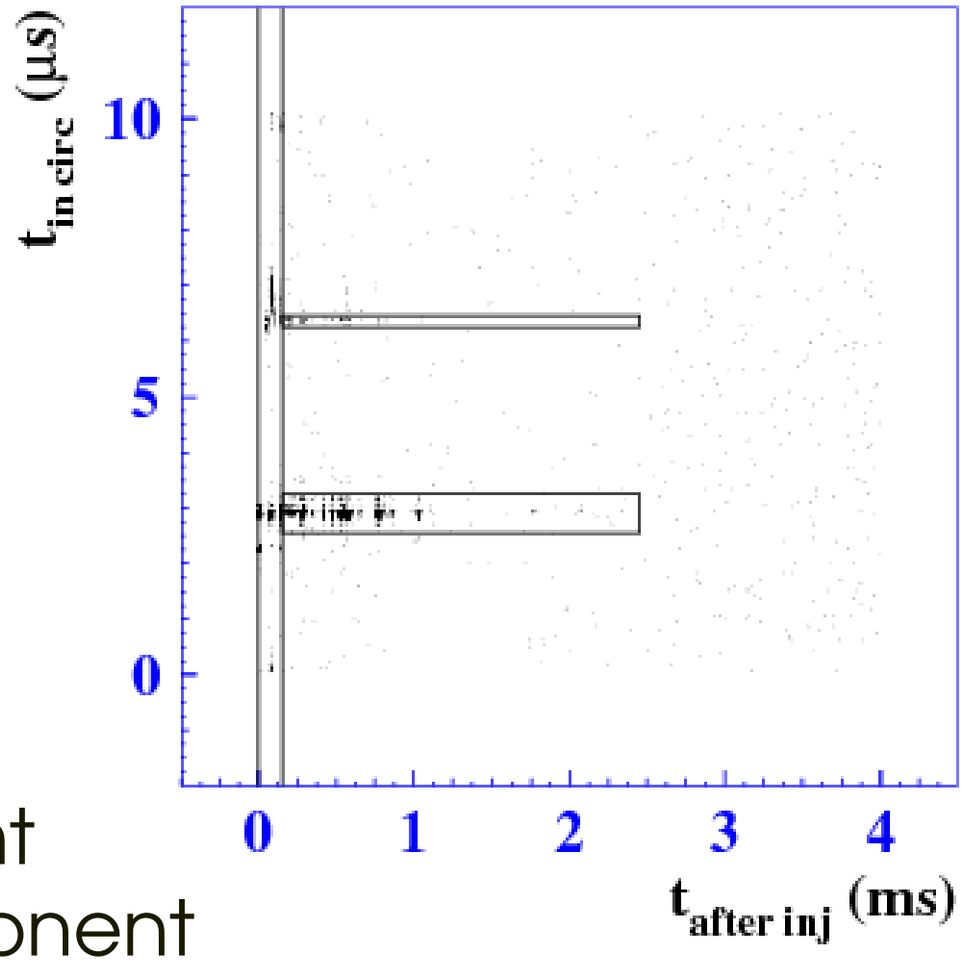
- Only TOP system requires < 10 ps clock-to-clock jitter
  (others should be fine with ~ 100 ps jitter)
  - DAQ receives SCLK from TOP and distribute to others
    (better than DAQ provides SCLK to TOP or use something else)
  - SCLK directly distributed on detector, not through E-hut
  - Beam pickup instead of RF clock?
- How about drift of the clock phase?
  - How much can be calibrated offline?
  - Phase-lock-loop between E-hut and detector
- RocketIO clock
  - High quality clock requirement (< 100ppm)
  - RF clock (127 MHz) is not good enough
    O(10h) lifetime in RocketIO link (tested in E-hut)
  - RocketIO will be fully asynchronous from the RF clock
    (driven by on-board Xtal)

System clock (frequency)

- SCLK 42 MHz (RF/12)
  - Compatiblity with LHC technologies (AMT3, APV25, . . .)
  - No more reason? — 32 or 63 MHz could be an option
    Electronics for $1/2^n$ has better performance than $1/(3 \times 2^n)$
- SCLK 32 MHz option?
  - Introduces more latency in the TTD handshake
  - APV25 allows longer L1 latency, at a cost of deadtime
- SCLK 63 MHz option?
  - COPPER and TTD? If it works, it reduces the latency
  - Existing COPPER and TTD to be modified (replace Xtal)
- Mixed SCLK system
  - All three clocks (63/42/32 MHz) will be available, users can choose in the frontend and FINESSE SCLK or RCLK
  - 42/32 MHz are generated from 63 MHz when needed

Injection veto deadtime

- Injection noise
  - Short component — all over the ring
  - Long component — only in the injected bunch every 10 μs (two components?)



- Injection veto for the L1 trigger
  - 150 μs veto for short component
    10% times 2.5ms for long component
  - 50 Hz injection ➡ ~2% deadtime
- Injection effects on PXD?
  - Takes 10 μs to readout
  - Always affected by the long component

Status / Plans

- Experiences have been gained
  - System control using serial bus (similar in RocketIO)
  - Multi clock domain operation (64 MHz sync'ed trigger and 42 MHz sync'ed readout)
- Test setup using TT-IO module
  - RocketIO characteristics have been studied
- Plans for module development
  - FINESSE with Virtex5 FPGA for various tests
  - new TT-IO (with Virtex5 FPGA) for better clock handling

Proposal

- Minimum interval between two triggers: t_{in} = 200 ns
  - close to the APV25 limitation, requires duplicated ECL event
- Depth of the fast FIFO: $N_{\mathrm{buf}} \geq 5$
  - APV25 limitation, enables data integrity with small deadtime
- Time to flush the fast FIFO: $t_{\mathrm{out}} < 20\ \mu s$
  - APV25 limitation for 42 MHz clock, still under discussion
  - Don't design anything slower
    An equivalent latency for a variable latency design
- Backend should not slow down for these requirements

The readout speed may not follow this model, e.g. when the latency is data-size dependent. It has to be simulated to meet the requirements.

More material

APV25 Pipeline

- **192 pipeline cells** (actually a ring buffer)
- After APV receives a trigger, the corresponding pipeline cells are labelled in an index FIFO in order not to be overwritten before the event is completely read out
- **Index FIFO** has **32 cells**

  → In worst case, **160 pipeline cells** always remain active

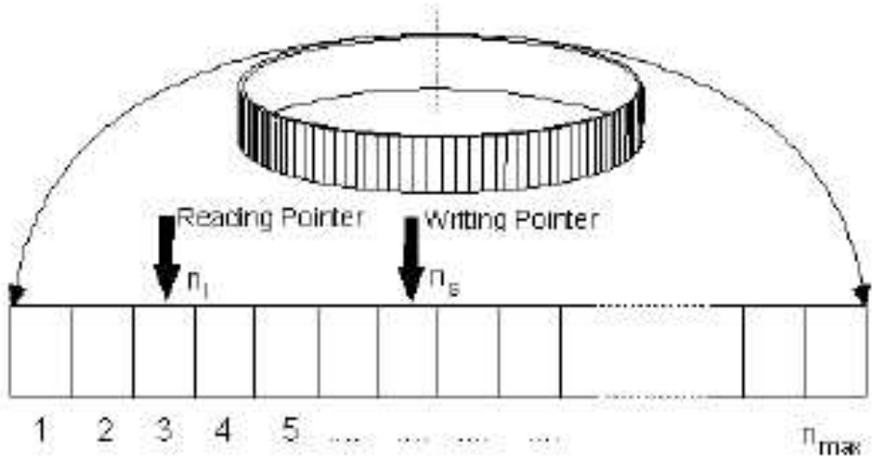
- = **3.8µs @ 42.3MHz** clock (RF/12)

  → 3.5µs max. latency for L1

or

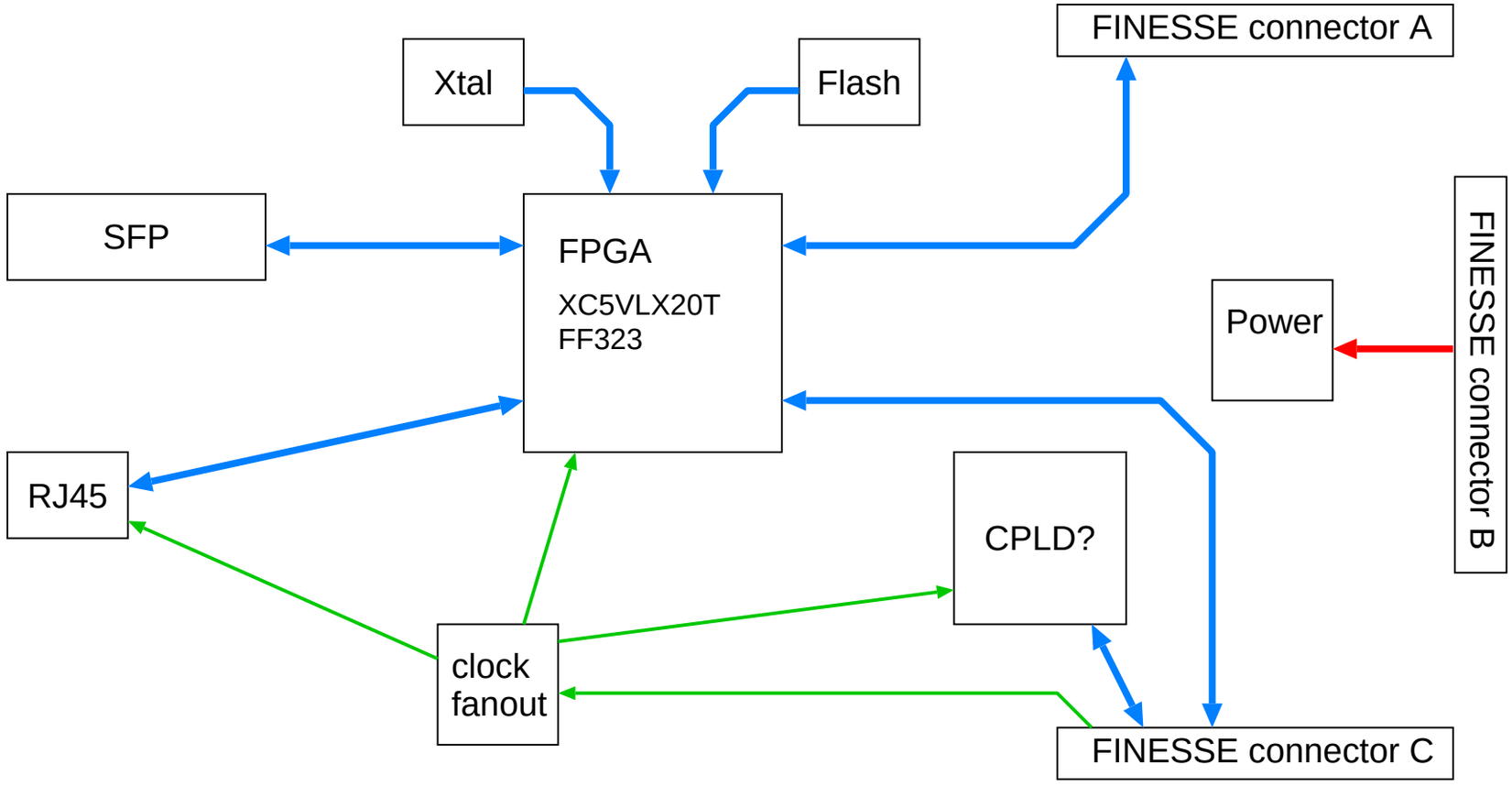
- = **5.0µs @ 31.8MHz** clock (RF/16)

  → 4.7µs max. latency for L1



- Preferred by **Iwasaki-san**, but has implications on trigger rate (see below)

MGT-FINESSE



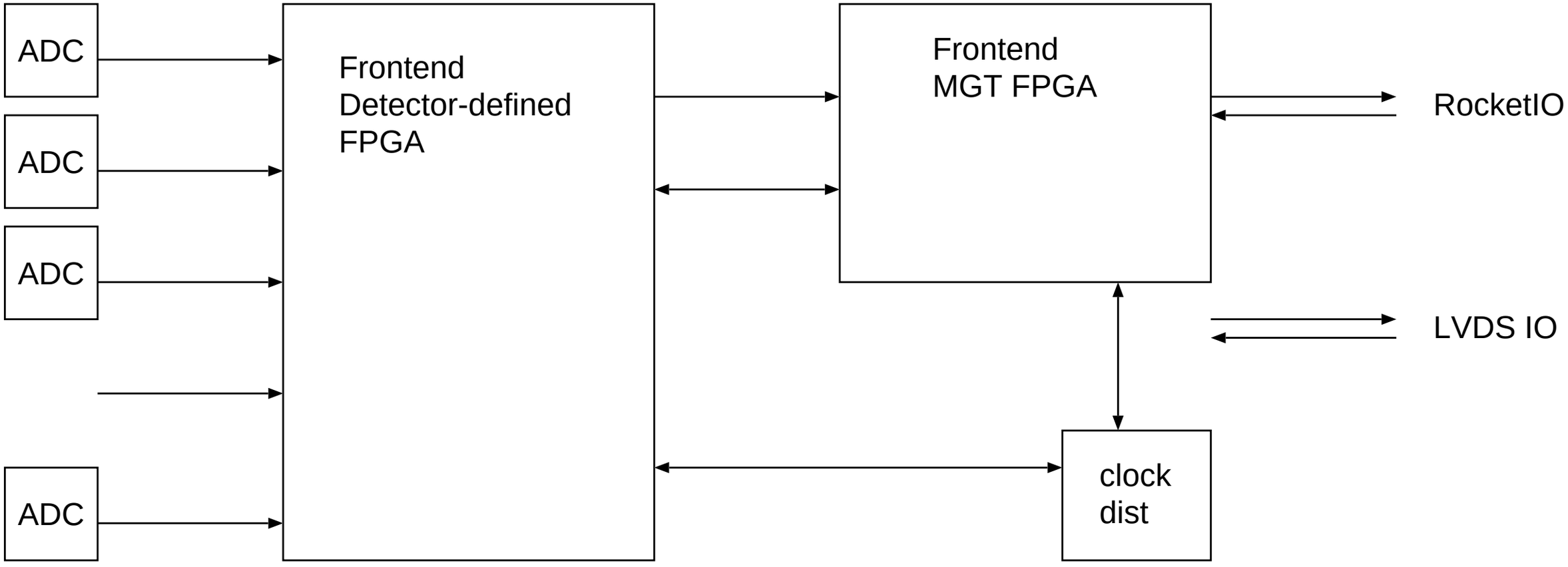
- Main components: Virtex-5, SFP and FINESSE connectors
- RJ-45 for the trigger distribution option 1
  Xtal for the asynchronous RocketIO option
- CPLD for on-the-fly reprogramming of Virtex5
- 127 MHz received through RCLK line (SCLK is 42.33 MHz)

FINESSE interface

- 2× 40-pin connectors (+ 36-pin connector for ±5V and +3.3V)
- 7-bit address, 8-bit data for control from/to COPPER
- 32-bit data to COPPER FIFO
  FIFO write clock from FINESSE
  (could be RXUSRCLK2)
- Two independent LVDS clock lines
  SCK = 42.33 MHz, also used in COPPER
  RCK = user deefined
- Several pins can be used for other purpose by reprogramming TT-RX
  `(IO2, IENA, ABRT, TYP[0..3])`

FF0	IRSTB
FF1	IENA
FF2	IO2
FF3	TYP0
FF4	TYP1
FF5	TYP2
FF6	TYP3
FF7	TAG0
FF8	TAG1
FF9	TAG2
FF10	TAG3
FF11	TAG4
FF12	TAG5
FF13	TAG6
FF14	TAG7
FF15	LD0
FF16	LD1
FF17	LD2
FF18	LD3
FF19	LD4
FF20	LD5
FF21	LD6
FF22	LD7
FF23	LA0
FF24	LA1
FF25	LA2
FF26	LA3
FF27	LA4
FF28	LA5
FF29	LA6
FF30	LRW
FF31	CSB
GATE	TRGP
FRSTB	TRGN
FWENB	REVP
FWCLK	REVN
FFUL	RCKP
NWFF	RCKN
ABRT	SCKP
BUSY	SCKN

Frontend interface



- A separate FPGA+clock receiver on the frontend card
- Interface to the detector logic: same as FINESSE

Protocol for RocketIO (rough idea)

- Need a very simple protocol
- 8b10b encodign/decoding (64b66b is not needed)
- 32-bit word, data driven at a half clock rate (USRCLK2)
- Packets from front-end to FINESSE
  - Sync packet (SYN) — RF revolution signal
  - Trigger info packet
  - Clock phase correction packet?
  - Other request packets
- Packets from front-end to FINESSE
  - Interrupt packet: single word with K-code (ACK/NAK/RST)
  - Sync packet (SYN)
  - Data packet: one frame per one event
  - Status packet: return for the query request, not to mix with the data packet