Salsa ASIC : Interfaces

Irakli Mandjavidze on behalf of the Salsa collaboration

Irfu, CEA Saclay
Gif-sur-Yvette, 91191
France

2nd DRD1 collaboration meeting & topical workshop on electronics
19/Jun/2024

Outlook

- Possible use cases for Salsa
- “Traditional” integration within a frontend
  → Heterogeneous interface
- “Alternative” integration scheme
  → Unified interface
- Example of envisaged integration in EIC readout
- Summary

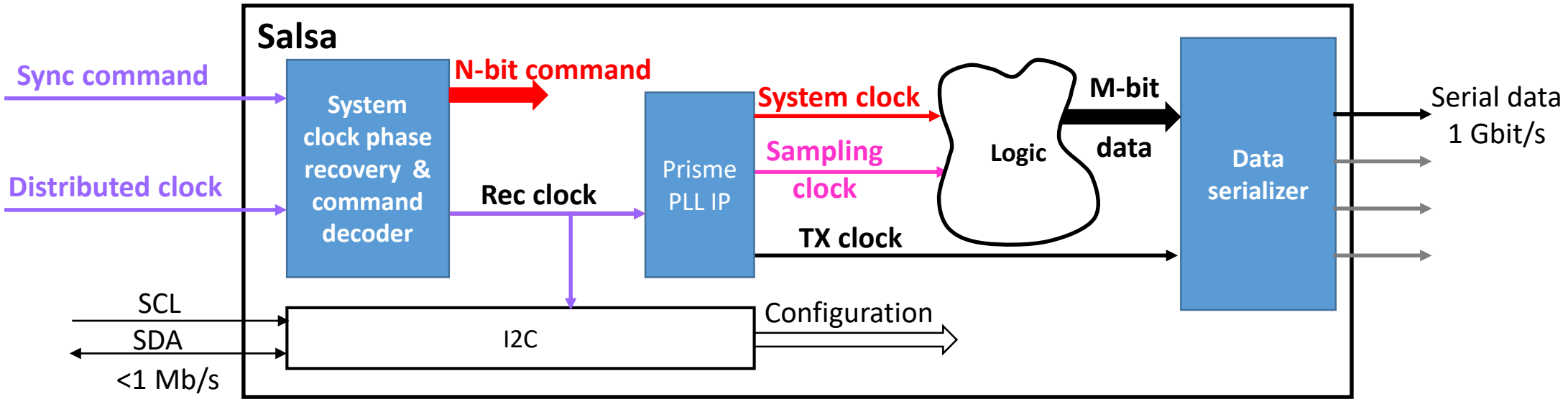
Some use cases that concern Salsa interfaces

- Setups of different nature
  - → Large collider experiments, moderate size fixed-target experiments, small standalone setups
    - Different level of integration depending on channel count and complexity of the experiment
  - → Repeating beam structure or continuous particle flow
    - Synchronous or asynchronous experiments w/ or w/o relationship between the system clock and physics events
      - Variety of clocks : 40 MHz @ LHC; 100 MHz @ EIC; 250 MHz @ CEBAF; 53 MHz @ Fermilab test facility; …
    - More or less rich set of system level synchronous commands to interpret and follow

- Environmental differences
  - → Magnetic field, radiation, space limitation
    - More or less compact design with powering and cooling restrictions

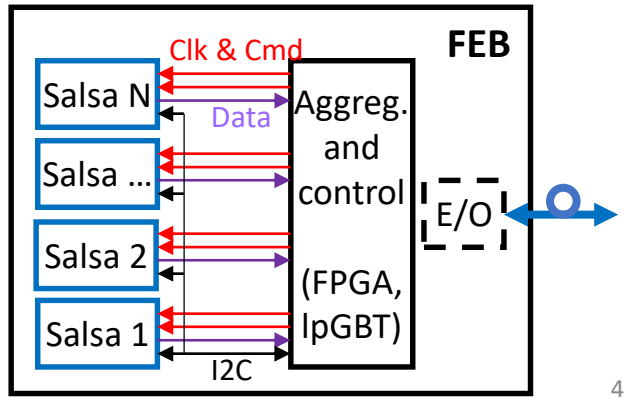
- Differences in the readout strategies
  - → Triggered or streaming or a mix of both
  - → Contribution to trigger generation

Support for “traditional” prevalent interface

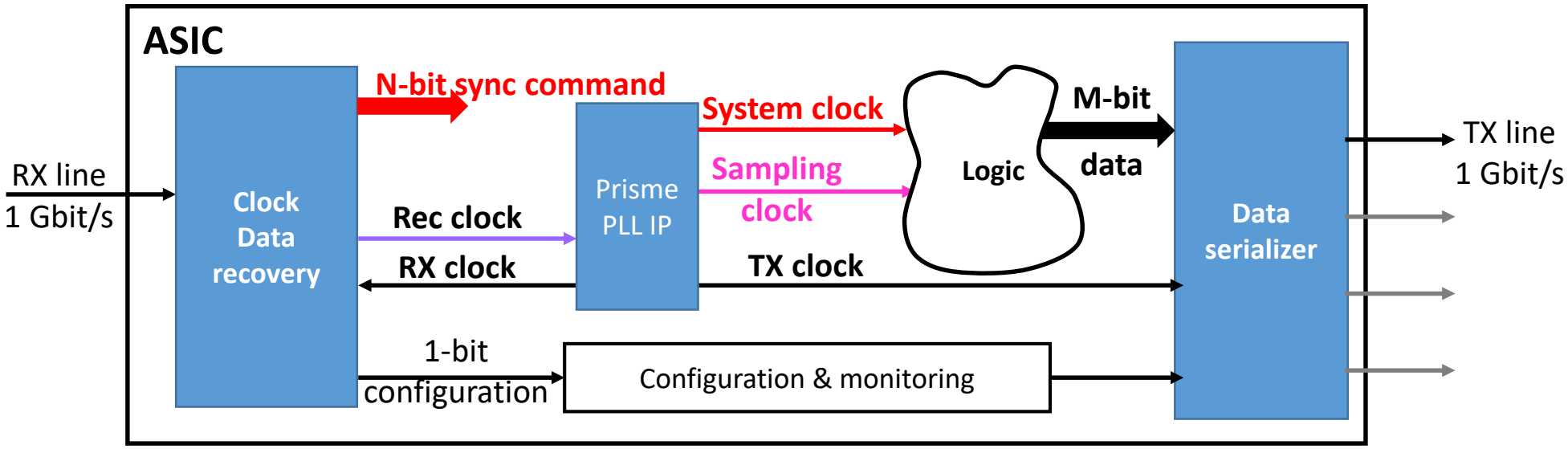


- Important number of heterogeneous external interface signals proper for each functionality
  - → Clock_diff_in, SynCmd_diff_in
    - Synchronous commands decoding options in backup
  - → SCL_in, SDA_io
    - Configuration of ASICs on a FE board in series : longer startup and recovery times
  - → Up to 4 Data_diff_out serial links
  - → Additional IOs like Trigger_diff_in, TrigPrim_diff_out

- May require an on-board companion “intelligence” for control & aggregation

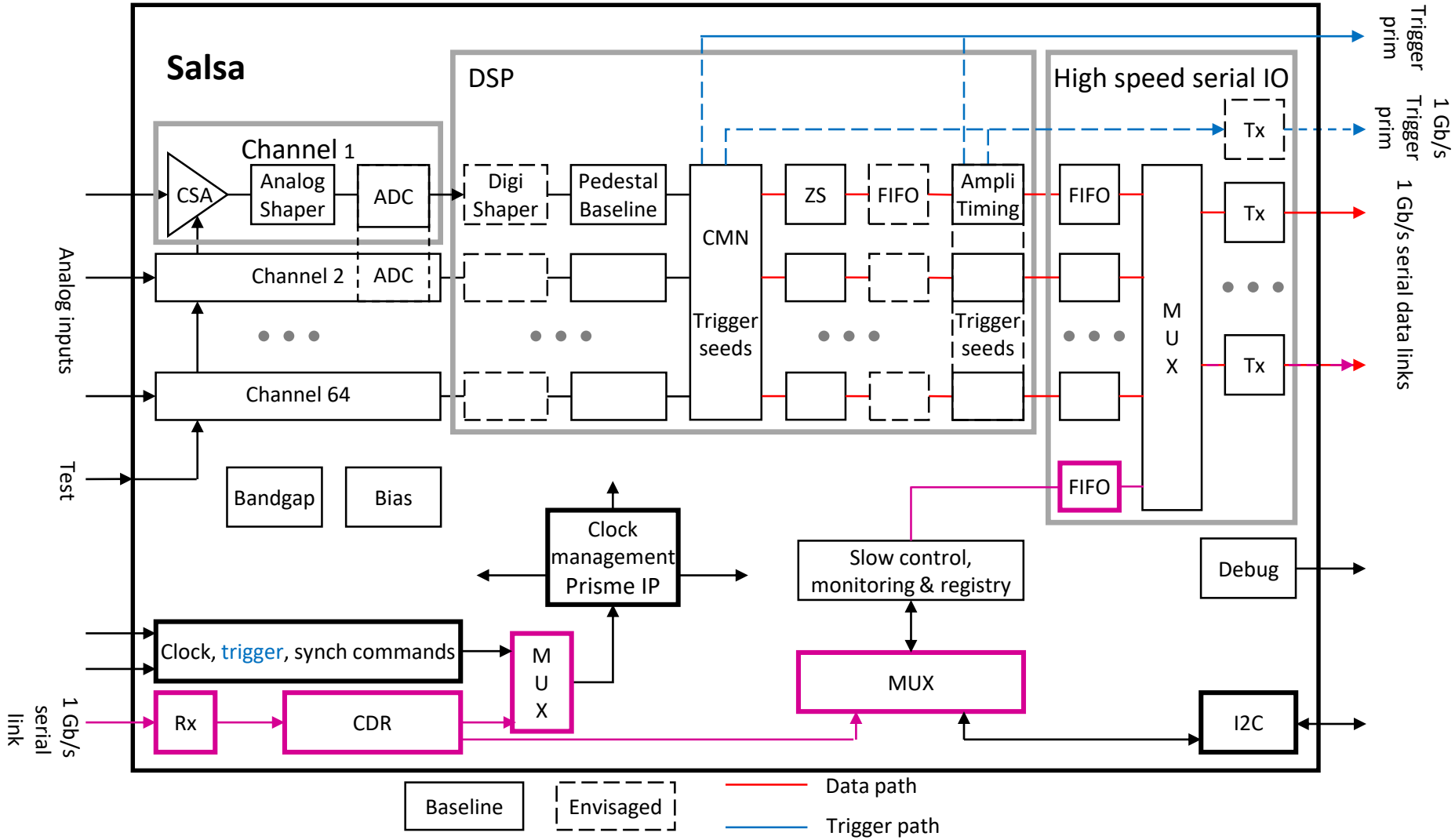


Alternative “unified” interface



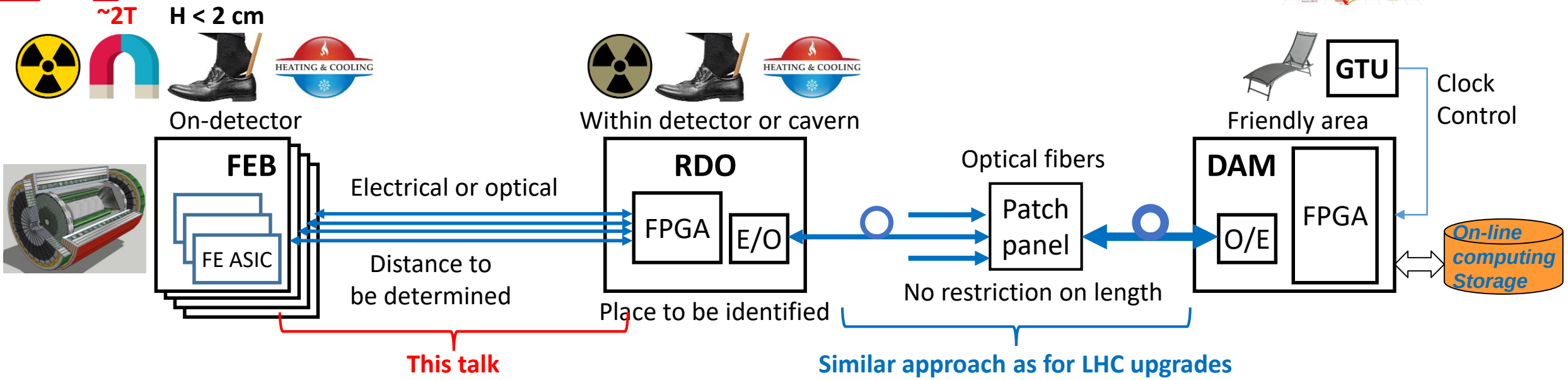
- Single encoded RX line for Clock, SynCmd, Trigger, configuration and monitoring
  - → Minimal external interface: a single diff RX line + at least one diff TX line
    - Simplest case: only 4 pins (Rx_p / Rx_n + Tx_p / Tx_n) to communicate with the chip
    - Parallel configuration of ASICs possible : fast startup and recovery time
- Relatively complex initialization phase requiring collaboration from the remote partner
  - → Clock recovery phase followed by
  - → Data reception and transmission phase

Integrating unified interface in Salsa



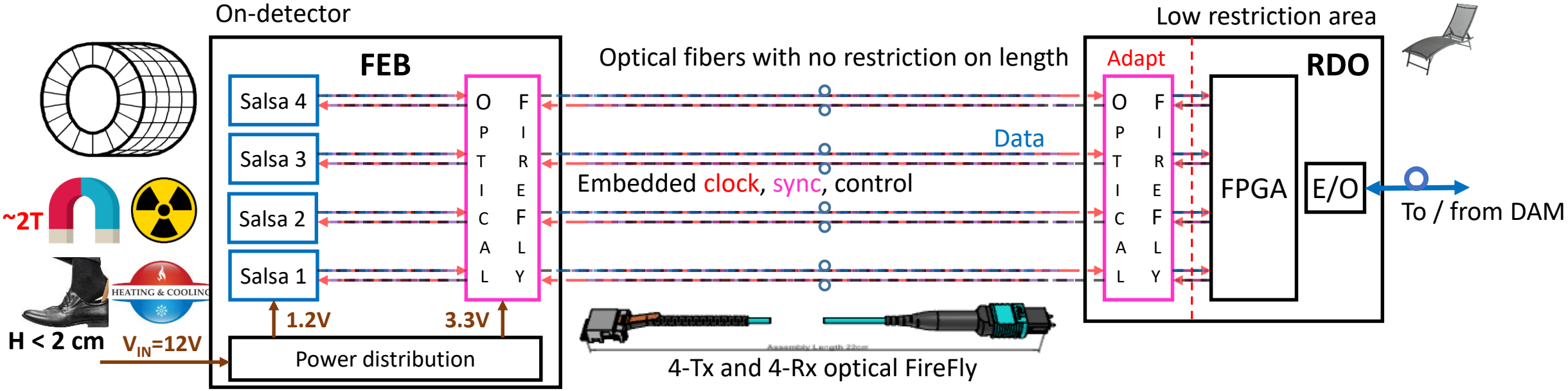
- Low complexity CDR design in progress to check viability
- Work in progress on detailed specifications

EIC example : streaming readout



- FEB – frontend board with readout ASICs
  → Sub-detector specific
- RDO – readout module – first stage of FEB data aggregation, last stage to dispatch clock & control
  → Mostly common design framework between sub-detectors, different form factor
- DAM – data aggregation module – interface with computing and global timing and control unit (GTU)
  → Common design for all sub-detectors
- Downstream towards detector : clock, control, monitoring
- Upstream towards storage : physics, calibration, monitoring data

EIC example : 256-channel FEB with optical interface

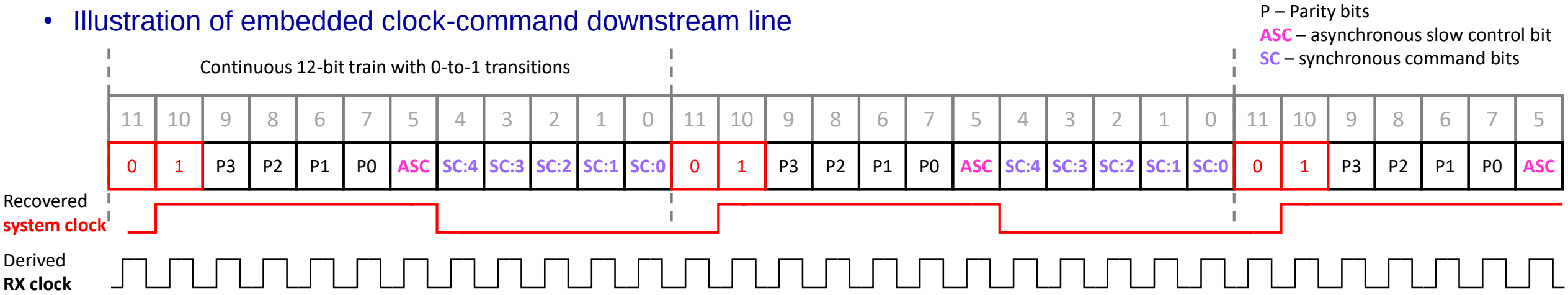


- FEB
  - → ASICs directly connected to 4-lane bidirectional parallel optic FireFly transceivers from Samtec
    - Single 1 Gbit/s Rx line encoding clock, sync run-control and asynchronous slow control and monitoring commands
    - Single 1 Gbit/s Tx line for physics, calibration, control and monitoring data
  - → Low active component count
    - Easier to adapt to challenging on-detector environment
    - Samtec FireFly : reported to stand TID of 50-100 krad and neutron fluence of at least 5×10^{11} n_{eq} / cm^2

- Optimal tradeoff between complexity of the on-detector electronics and its power consumption

Data to Salsa over the serial RX link

- Illustration of embedded clock-command downstream line



→ Periodic 0-to-1 transition for system clock recovery and jitter cleaning in the Prisme PLL IP
  - Inspired from original idea in
    D. Calvet, Clock-Centric Serial Links for the Synchronization of Distributed Readout Systems, IEEE TNS, V 67, N 8, 2020

→ Hamming error detection and recovery bits
  - Use of Reed Solomon RS(7,5) code is considered but requires higher link speeds

→ Up to 32 synchronous commands at every system clock cycle

- Link bandwidth for typical system clock frequencies

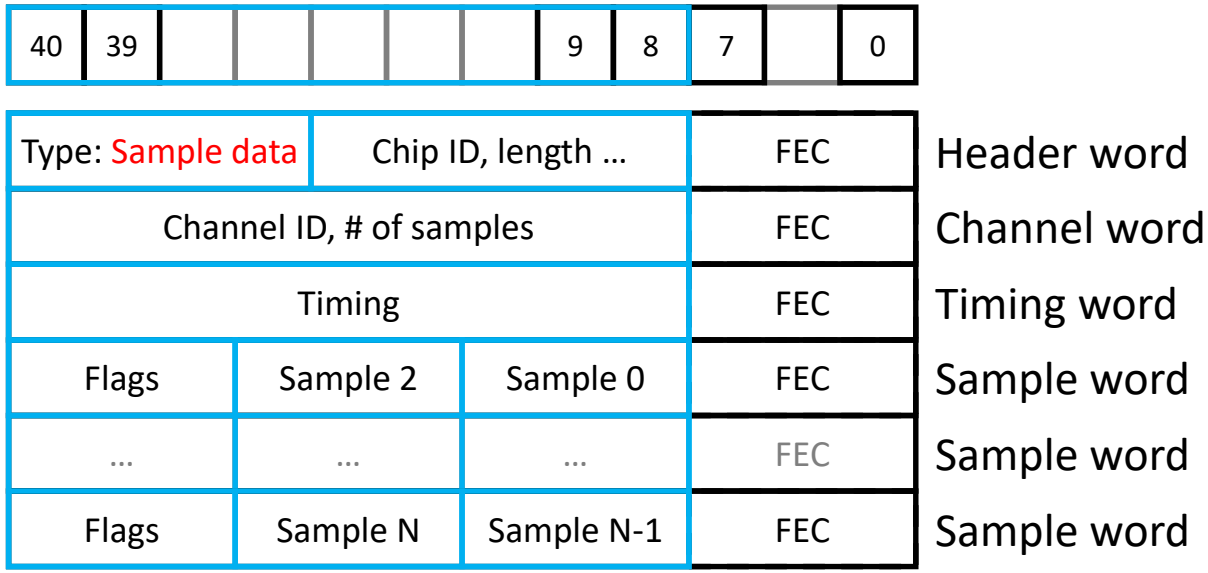
→ 40 MHz @ CERN : 480 Mbit/s
  - Potential to transmit 20-bit asynchronous slow control commands @ 2 MHz

→ 100 MHz @ EIC : 1.2 Gbit/s
  - Potential to transmit 20-bit asynchronous slow control commands @ 5 MHz
  - Considering a protocol with lower number of bits per system clock cycle
    - Lower link speed – easier integration with low end FPGAs

Data from Salsa over the serial TX link

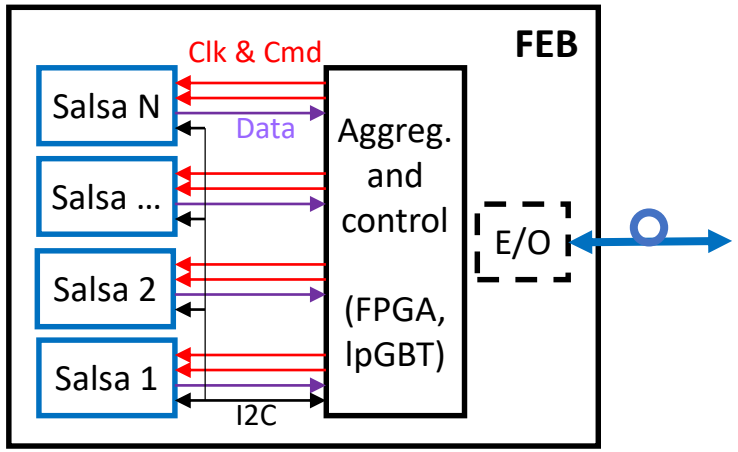
40-bit transmission unit

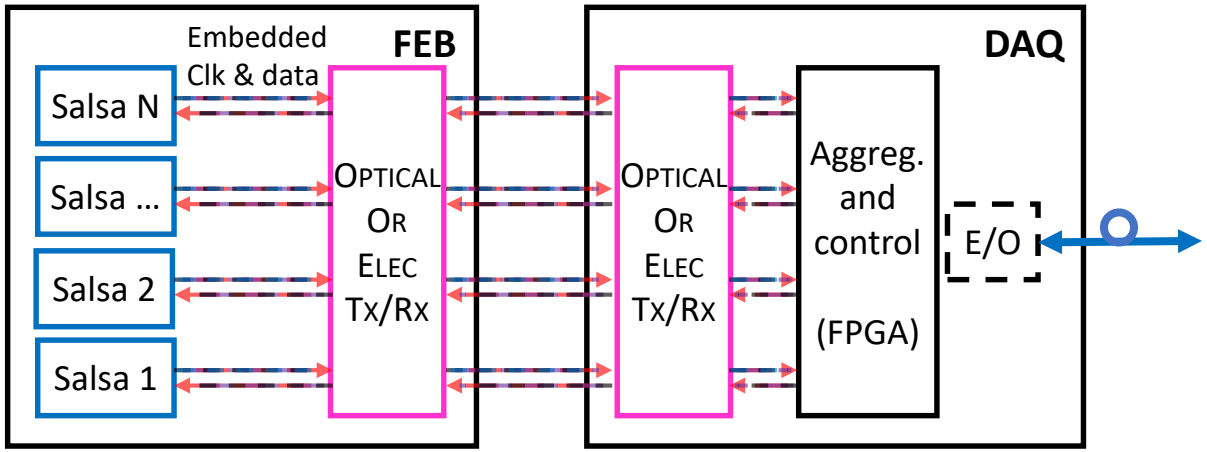
- Fixed 40-bit length transmission units
  - → 32-bit user payload + 8-bit FEC
    - (6+1)-bit Hamming with possibility to detect 2 errors and correct 1 error
    - RS(15,13) Reed Solomon is considered with possibility to correct 4 errors
- Salsa packets transmitted word by word
  - → 20% overhead due to FEC
- Illustration on Salsa packet with signal samples
  - → Packet type and chip identifiers
  - → Channel
  - → Timing binds samples to system clock
    - Sampling and system clocks might be different
      - Frequency and phase
  - → Succession of 12-bit ADC values
  - → Information on processing of samples
    - *e.g.* Raw, ZS, due to neighboring logic
- Example of a slow control packet in backup



Summary

- If successful, flexibility in Salsa integration
  → Support for **local companion ASIC** as well as **remote intelligence** schemes





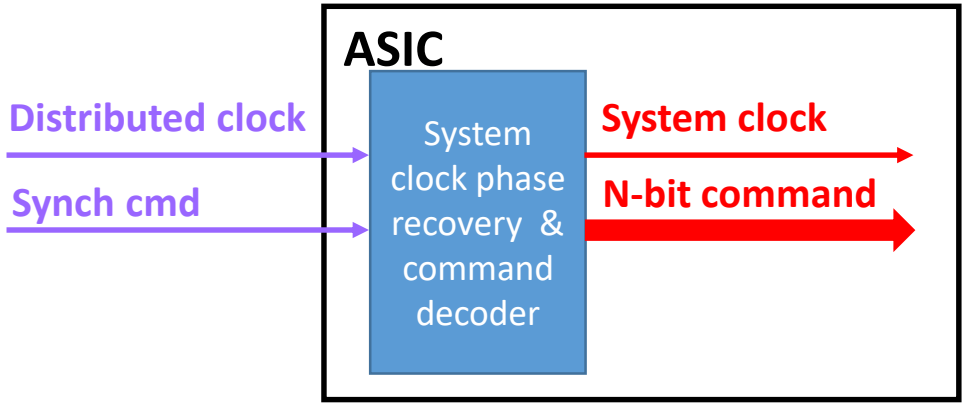
  → Low pin count unified interface : potential to control and aggregate data from a large number of ASICs
  → Less than 1 Gbit/s links to interface with standard FPGA IOs and SERDES IPs
  → Wide range of system clock choices with Prisme mixed analog-digital PLL IP
    - 40 – 120 MHz with aimed recovered clock jitter better than 10 ps RMS
- Work well advanced to fix specifications
- Prototypes under development and tests
- Appeal to DRD1 collaboration to understand the access conditions to the CERN radiation facilities
  → Validate radiation tolerance of the design

Backup

Traditional interface : detecting synchronous commands every clock cycle

- Distributed clock must be a multiple of system clock to decode N-bit synchronous command
  - SDR mode: N * SysClock
  - DDR mode: N * SysClock / 2

  → System clock phase needs to be recovered
  - Reserved bit field with a unique bit pattern



- LHC example : 40 MHz bunch crossing clock

  → 320 MHz distributed clock with 320 Mbit/s synchronous command line

  → 8-bit synchronous command sequence @ every bunch crossing

- EIC example : 100 MHz bunch crossing clock
  - Reminder : no lpGBT equivalent ASIC – use of COTS FPGAs for a subset of its functionality

  → 8-bit synchronous commands would require 800 MHz clock distribution

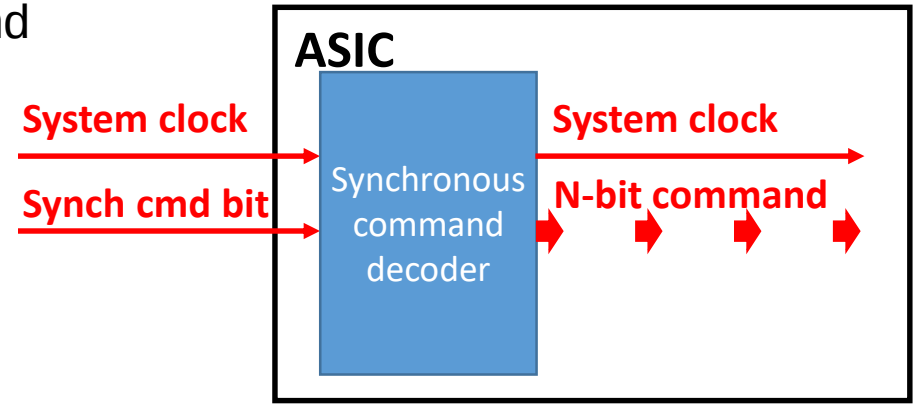
  → A possibility : 300 MHz distributed clock with 600 Mbit/s DDR interface for commands
  - 6-bit recovered sequence would allow decoding of 16 different commands

  → High rate clock distribution regarded as impractical by the collaboration

  → 16 unique commands might be not enough either

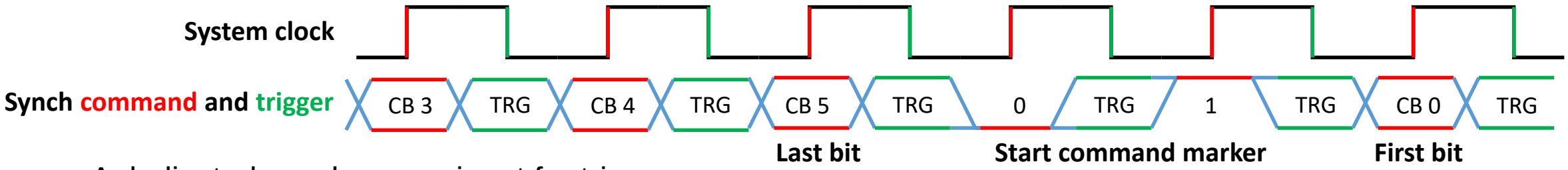
Traditional interface : bit-by-bit aggregation of synchronous commands

- System clock is distributed to ASIC
  - → Requires N clock cycles to construct N-bit synchronous command
    - Arrival time of the very first bit determines command timing



- EIC example : 100 MHz bunch crossing clock
  - → 8-bit sequence determines 6-bit command
  - → Synchronous commands can be received every 8th BX
    - 150 commands per EIC revolution

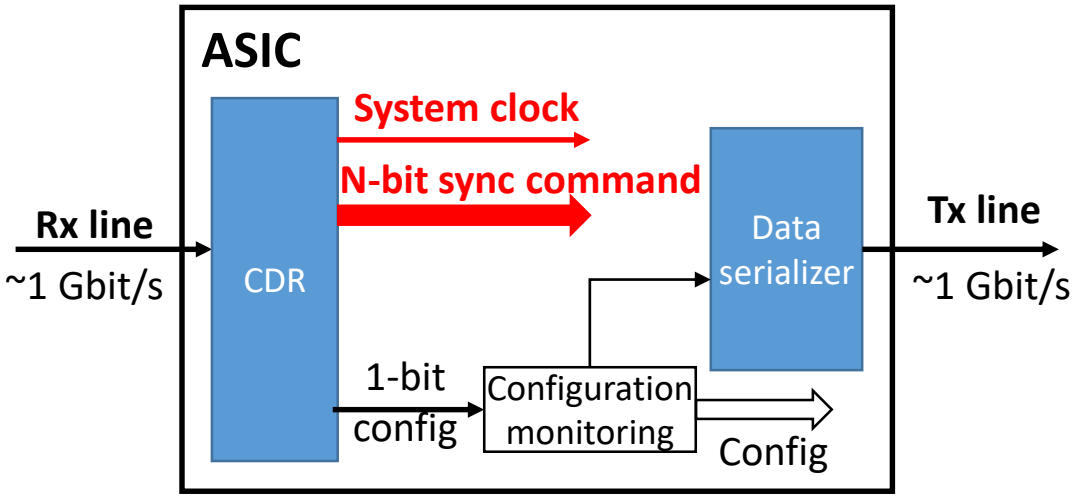
- Triggered readout requires special arrangement
  - → SDR interface imposes dead time if trigger is one of the synchronous commands
    - Dead-time : N x system clock period
  - → DDR interface alleviates the problem



  - → A dedicated synchronous input for trigger
    - Requires an extra distribution tree

Unified interface : data to Salsa : low bit error ratio

- EIC example : 1 Gbit/s RX line
  - → 100 MHz embedded clock
  - → 8-bits per bunch crossing
    - Horizontal parity bit
    - Asynchronous slow control and monitoring bit
    - 6 bits for synchronous commands



- Illustration of embedded clock-command line

9	8	7	6	5	4	3	2	1	0																				
0	1	HP	ASC	SC:5	SC:4	SC:3	SC:2	SC:1	SC:0	0	1	HP	ASC	SC:5	SC:4	SC:3	SC:2	SC:1	SC:0	0	1	HP	ASC	SC:5	SC:4	SC:3	SC:2	SC:1	SC:0

0-to-1 transition for system clock recovery →

  - → Periodic 0-to-1 transition for system clock recovery in the Prisme PLL IP
    - Inspired from original idea in D. Calvet – Ref.
  - → Potential to have up to 64 synchronous commands
  - → Potential to transmit 20-bit slow control commands @ 5 MHz

HP – horizontal parity
ASC – asynchronous slow control bit
SC – synchronous command bits

- Reality is more complex especially to ensure SEU recovery
  - → Larger error correction code, lower number of synchronous commands

Unified interface : data from Salsa over the serial TX link

40-bit transmission unit

- Fixed 40-bit length transmission units
  - → 32-bit user payload + 8-bit FEC
    - 7-bit Hamming with possibility to correct 2 errors
    - RS(15,13) Reed Solomon is considered with possibility to correct 4 errors
- Salsa packets transmitted word by word
  - → 20% overhead due to FEC
- Illustration on Salsa packet with response to registry read slow control request
  - → Packet type and chip identifiers
  - → Start address
  - → Values read from successive registers