



ATLAS Level-1 Muon Topological Trigger Information for Run 2 of the LHC

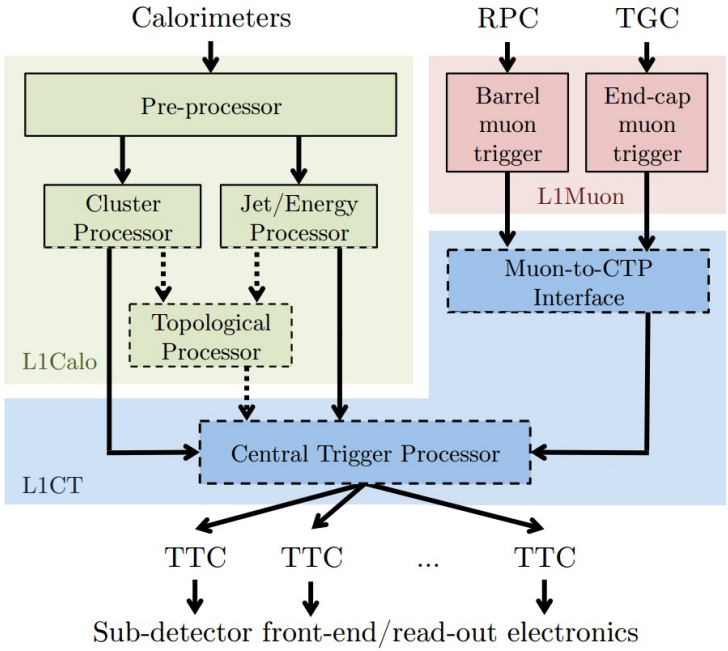
Marcos Vinicius Silva Oliveira

Outline

- Introduction
- Level-1 muon trigger
- Feasibility tests
  - Development
  - Results
- MUCTPI firmware upgrade for Run 2
  - Development
  - Verification
- Integration tests
- Conclusions

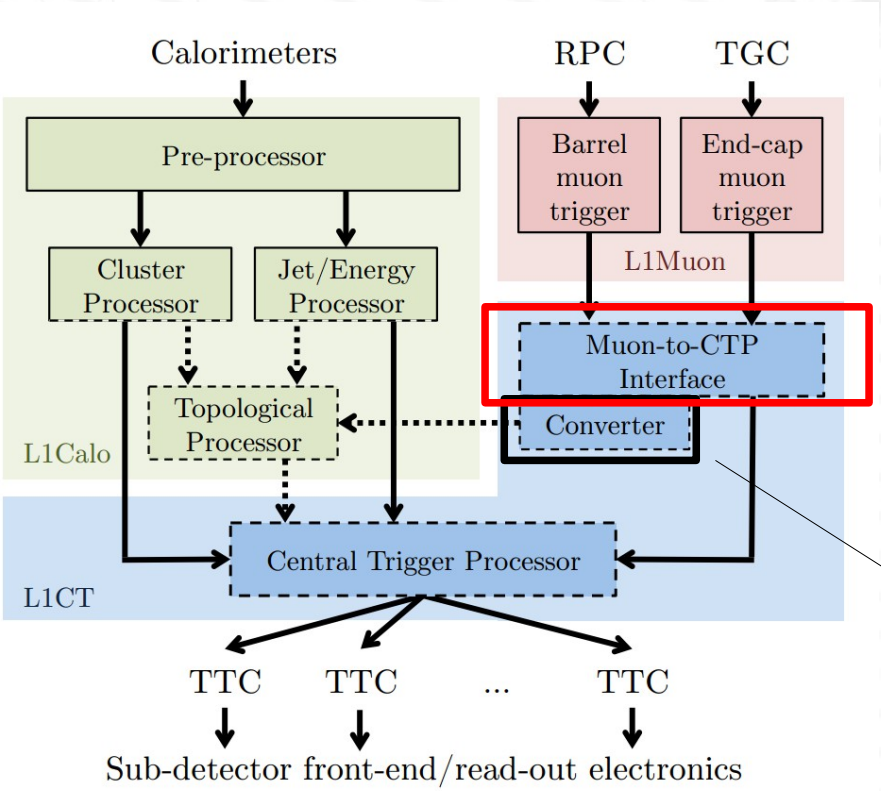
Motivation

- **Higher the luminosity, higher the energy cuts ► low-momentum objects are prescaled**
- **Trigger efficiency for some Physics channels can be improved with topological information**
  - **Examples: B-hadrons decaying to two low-*pT* muons and lepton flavor violation decays.**
  - **Only calorimeter topological information was foreseen in Run 2**
    - **Physics processes involving muons ► loss of trigger efficiency**



Motivation

- **MUCTPI firmware upgrade ► Muon topological information in Run 2**
  - **Sustain trigger efficiency for Physics channels involving muons**
- **Electro-optical interface used to forward MUCTPI data to Topological Processor**



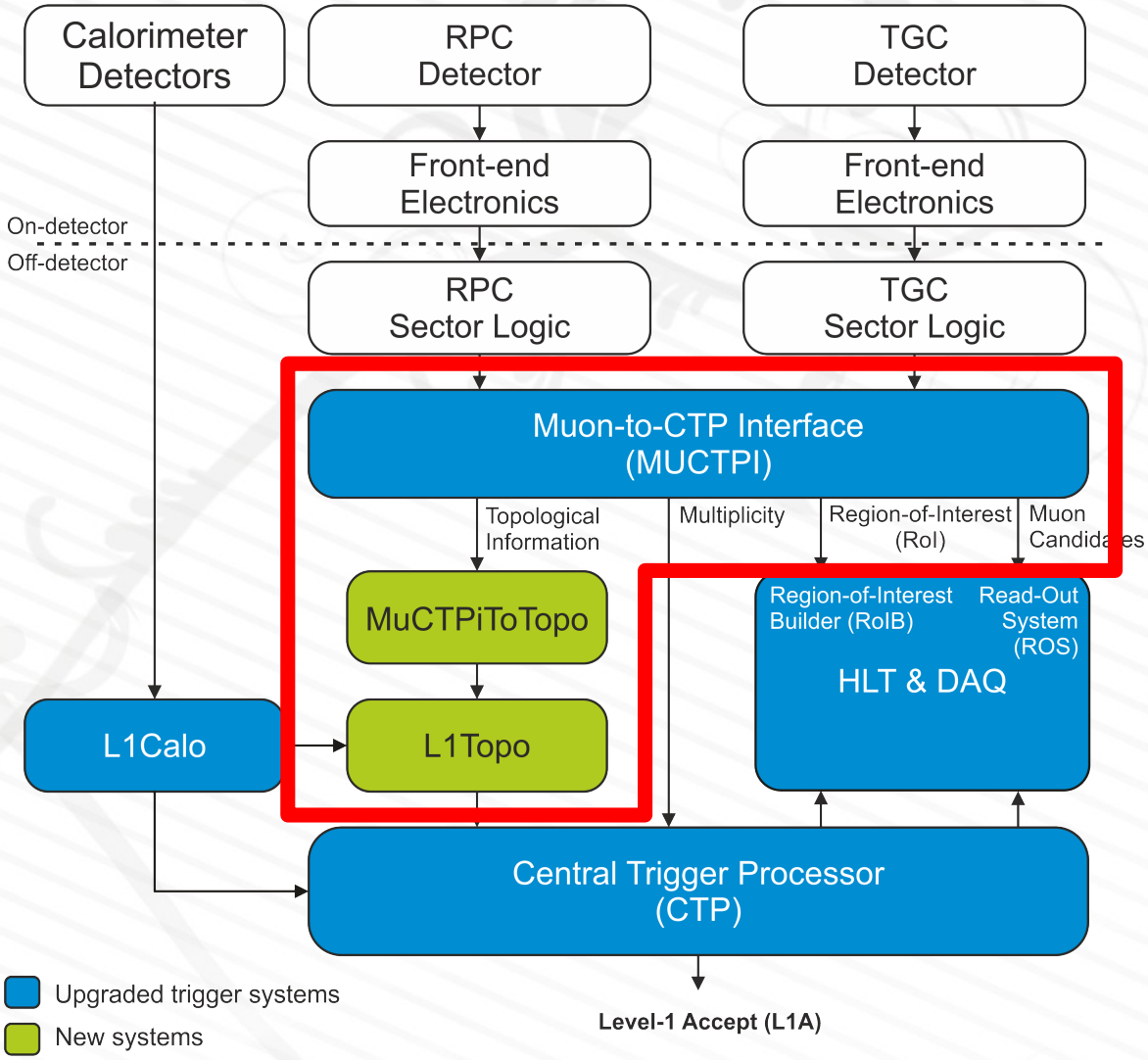
FW upgrade
using existing electrical trigger outputs

MuCTPiToTopo
(developed by an external institute)

ATLAS Level-1 Muon Trigger

ATLAS Level-1 Trigger system

- Implemented with custom electronics.
- Reduces rate from 40 MHz down to 100 kHz
- Adding topological information in Run 2 ► L1Topo
- The CTP receives flags from L1Topo

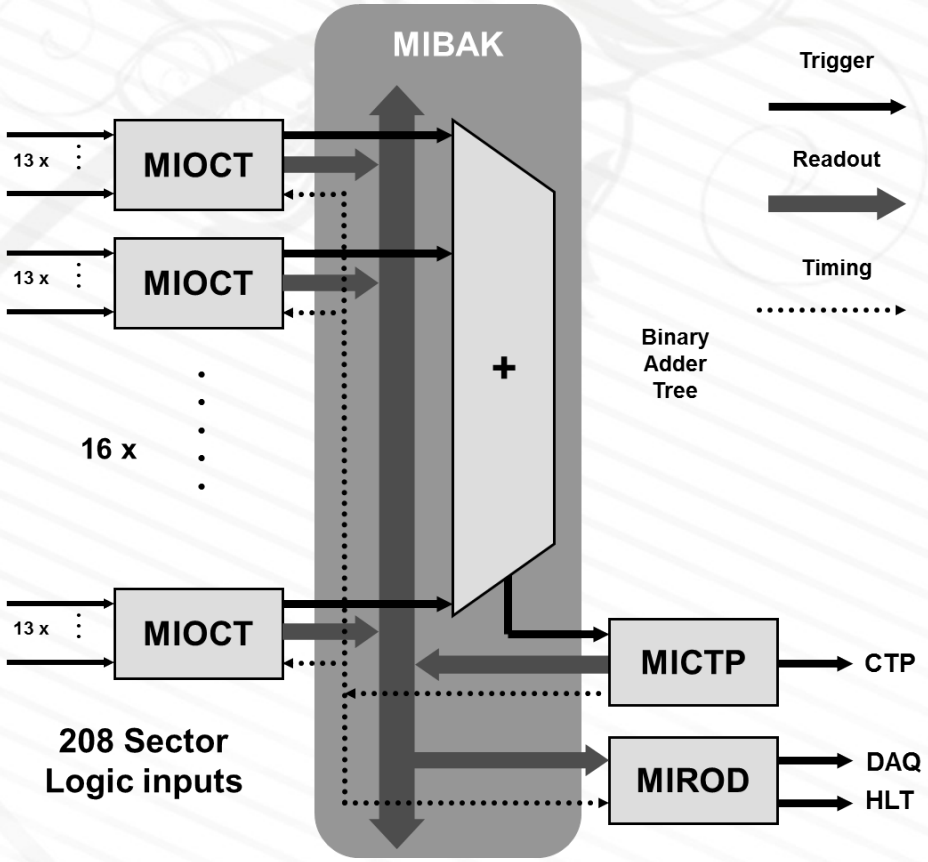


Muon trigger detectors

- Provide multiplicity and approximate energy range
- Fast response (< 1BC)
- Resistive Plate Chambers (RPC) in the barrel
- Thin Gap Chambers (TGC) in the end-cap
- Momentum of muons is measured from magnetic deflection of muon tracks in the toroid magnets
- Higher the momentum, smaller the bending
- Trigger sector logic reconstructs muon tracks
  - Sends sector data with six levels of transverse momentum (p_T) to the MUCTPI

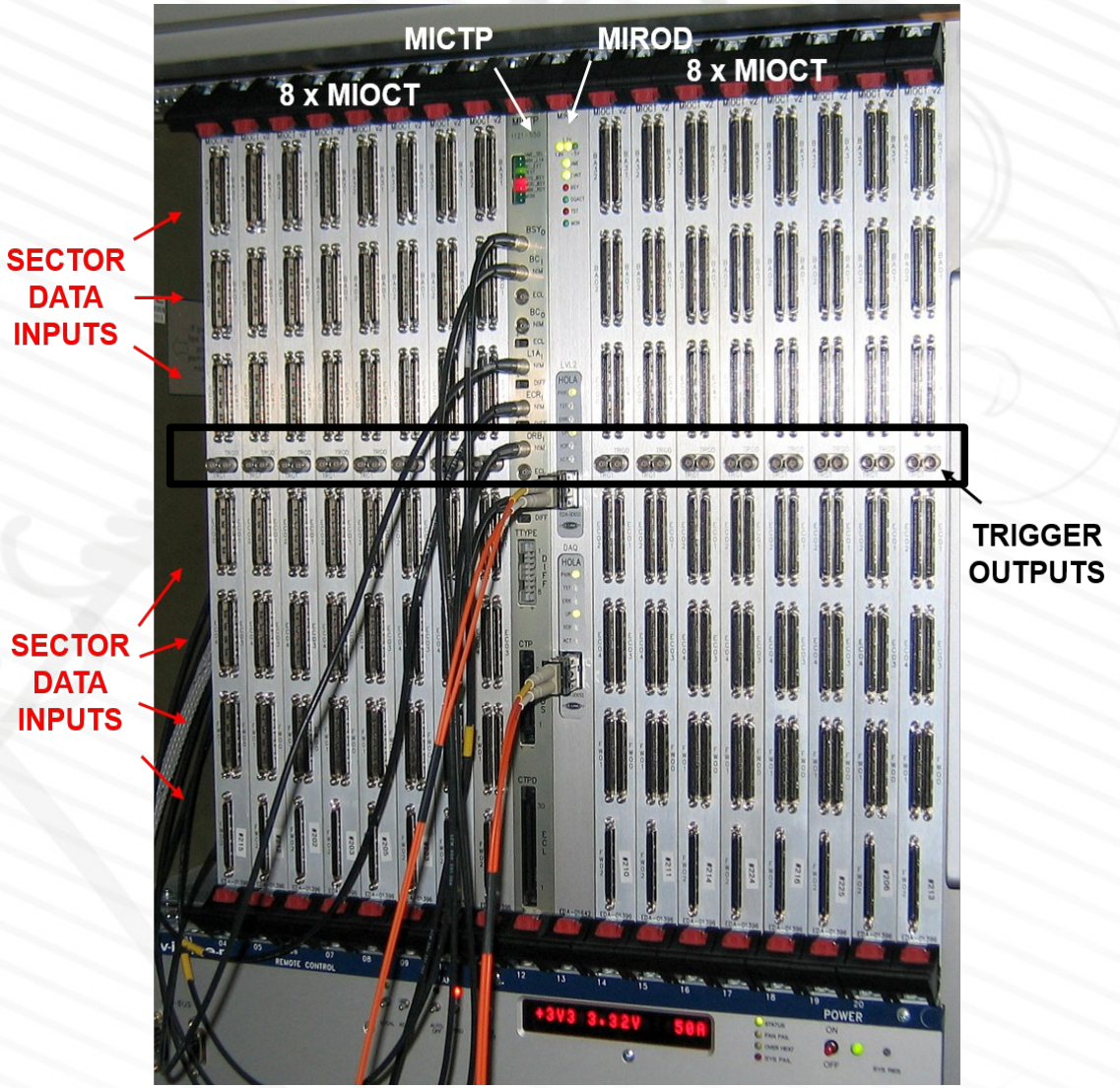
MUCTPI

- Calculates total number of muon candidates for each of 6 pT thresholds
- Avoids double counting of muon candidate in overlap regions
- Sends multiplicity to CTP
- Consists of:
  - 16 octant modules (MIOCT)
    - Generates the local multiplicity
    - Handles overlap
    - Sends flags to CTP using the electrical trigger outputs
  - MIBAK
    - Calculates total multiplicity
    - Distributes timing and control signals
    - Implements readout bus
  - MICTP
    - Receives trigger & timing
    - Forwards total multiplicity to CTP



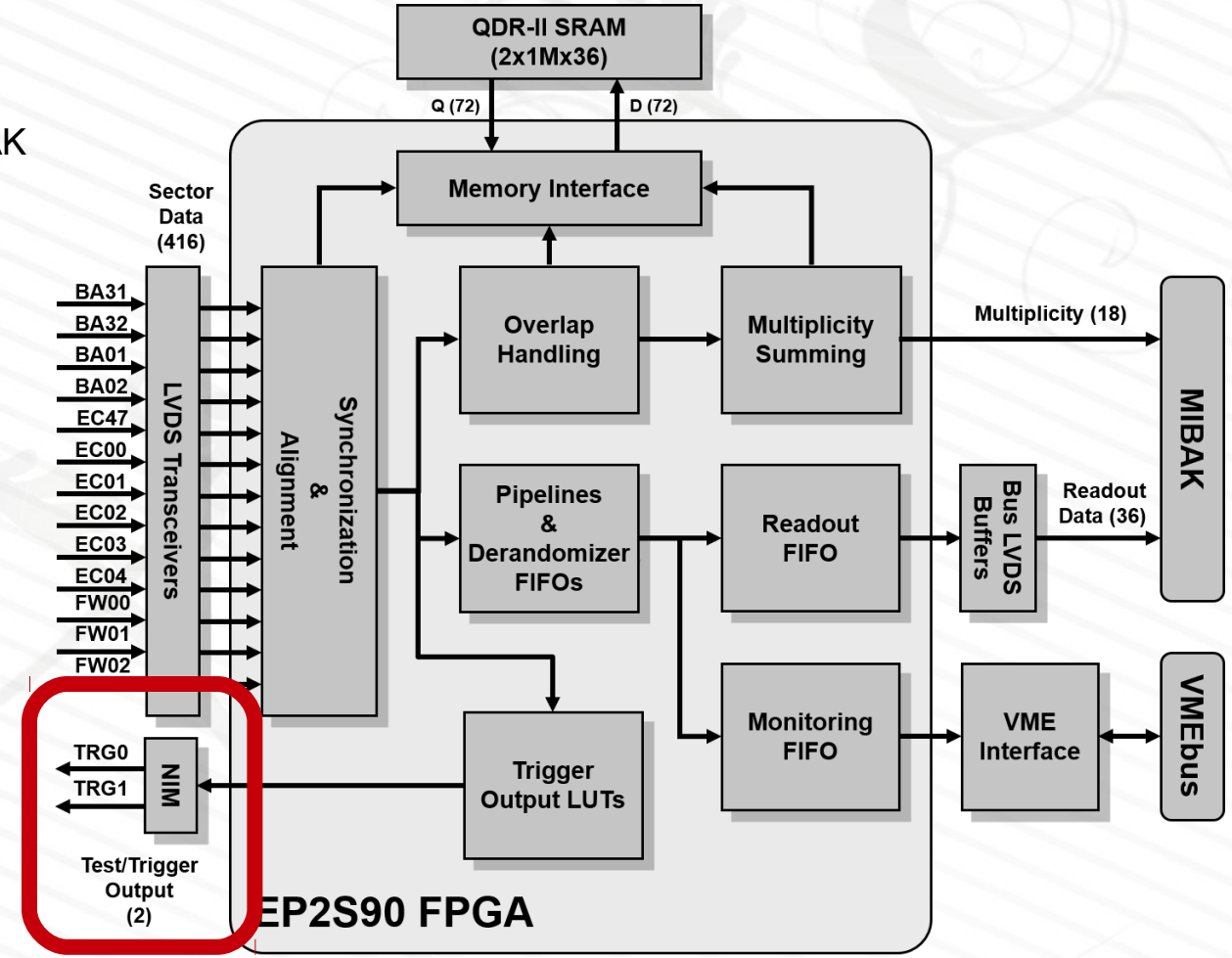
MUCTPI (cont.)

- MIROD
  - Collects readout data via MIBAK
  - Sends them to HLT and DAQ systems



MIOCT

- Synchronization using programmable length pipelines
- Overlap handling using programmable LUTs
- Sends local multiplicity to MIBAK
- SRAM memory for validation and to capture sector input data
- Sends trigger flags to CTP **at the BC rate (40 Mbps)** using the electrical trigger outputs



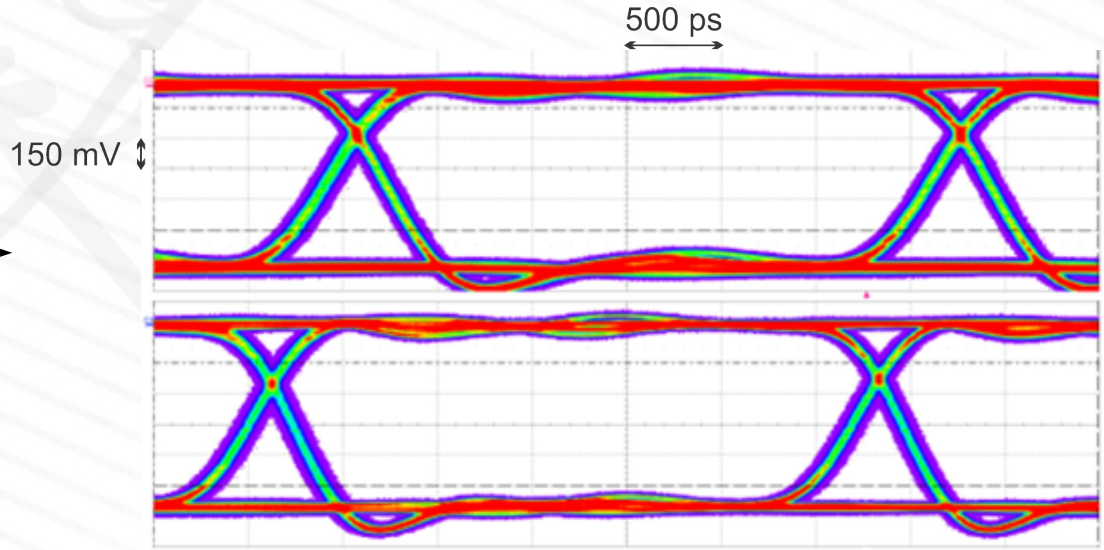
Feasibility tests

Feasibility tests

Can the MIOCT trigger outputs be used to transmit topological information to L1Topo?

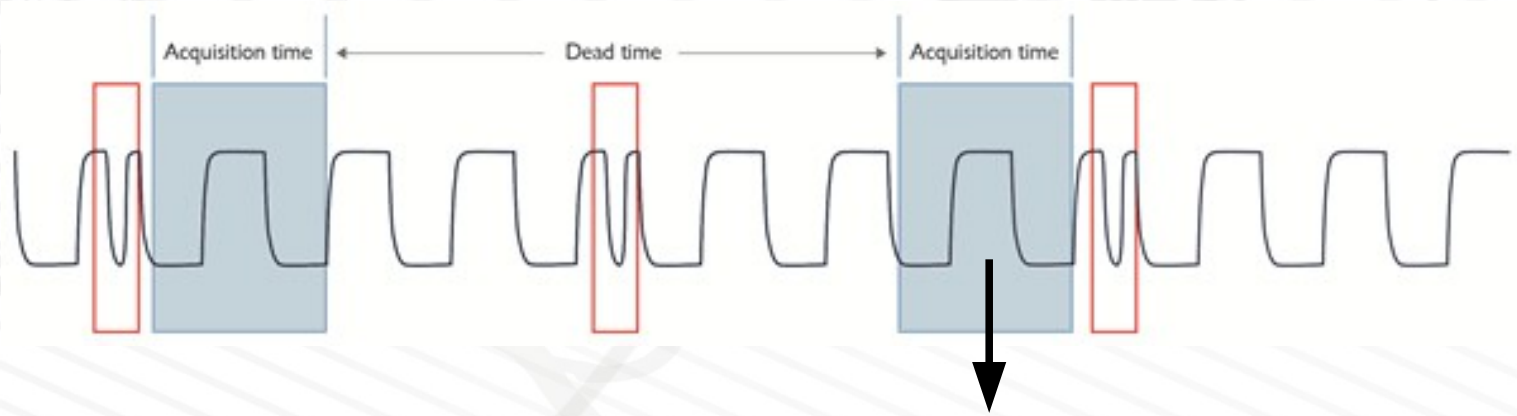
- Verify operation at high rates (40 Mbps is not enough)
- MIOCT firmware modification to send PRBS-31 data at 320 Mbps.
- Open eye-diagram with outputs sending data at 320 Mbps



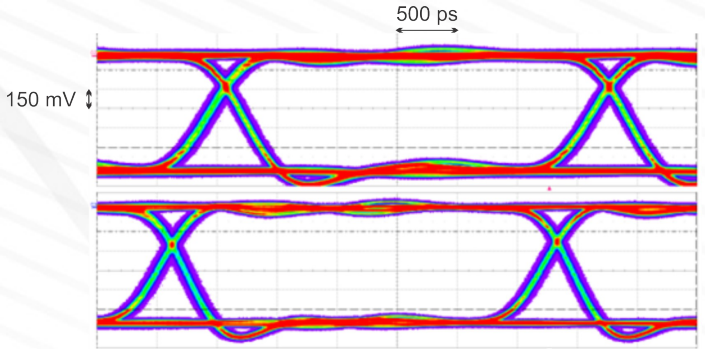


Feasibility tests

- Measurement characteristics
  - Oscilloscope / MuCTPiToTopo ► different receivers
  - Data are not stored for further checking
  - Oscilloscope ►long dead time ► measures only high BER



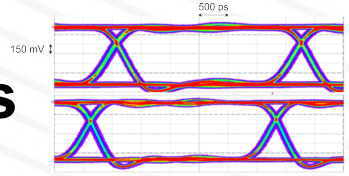
- Development of a test system
  - Measure low BER
  - Generate a phase scan
  - Check topological data



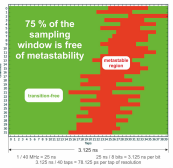
Summary

Feasibility tests

Development of an error rate test system

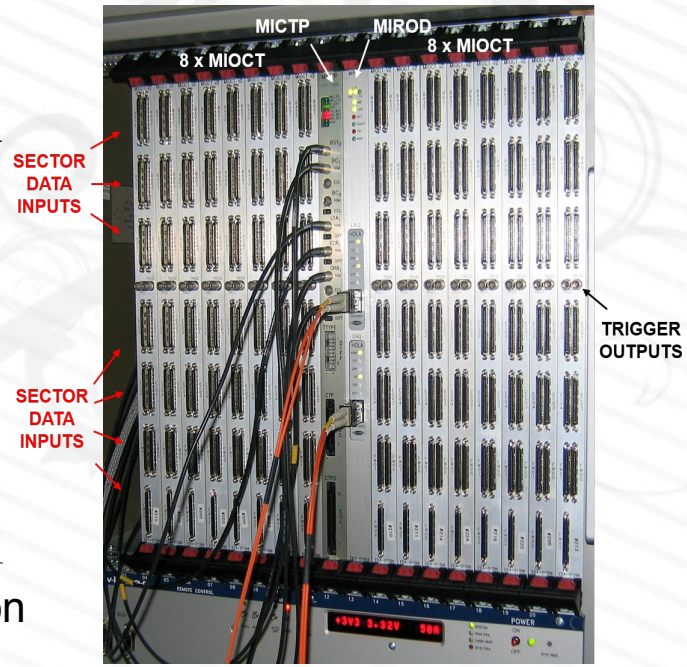


Maximum bit rate



Topological information

MUCTPI firmware upgrade



Integration tests in the lab

Verification of the upgraded MUCTPI system

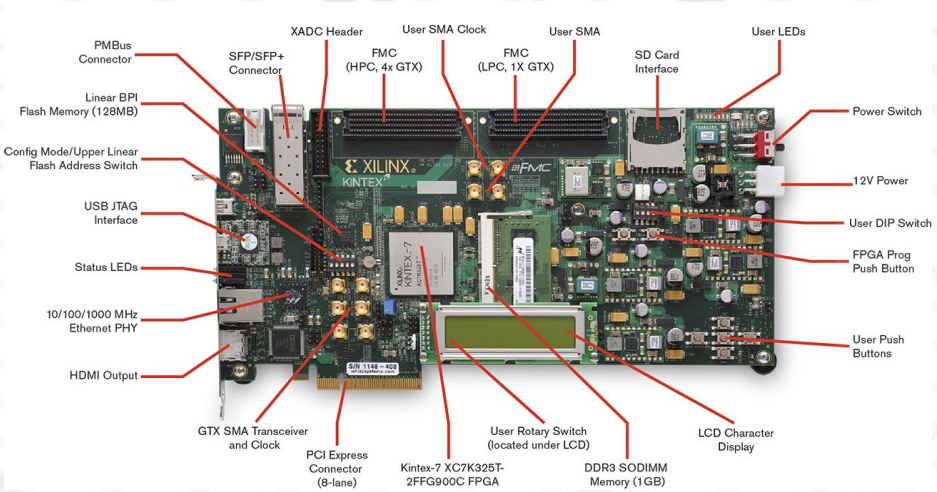
Tests in USA15

Test results in the lab

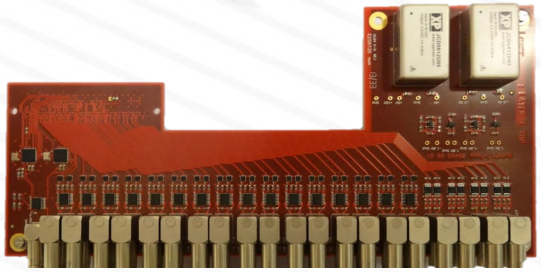
Development of the error rate test system

Test system hardware platform

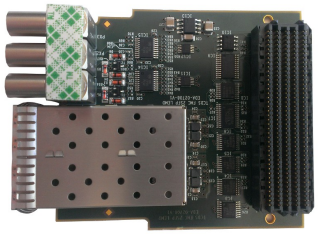
- Xilinx Kintex-7 FPGA board
  - FPGA with programmable internal delay lines
  - Gigabit Ethernet interface
  - DDR3 memory module
  - 2 FMC slots



- MUCTPI receiver mezzanine card
  - 32 NIM inputs and 8 NIM outputs
  - NIM / LVDS level converters

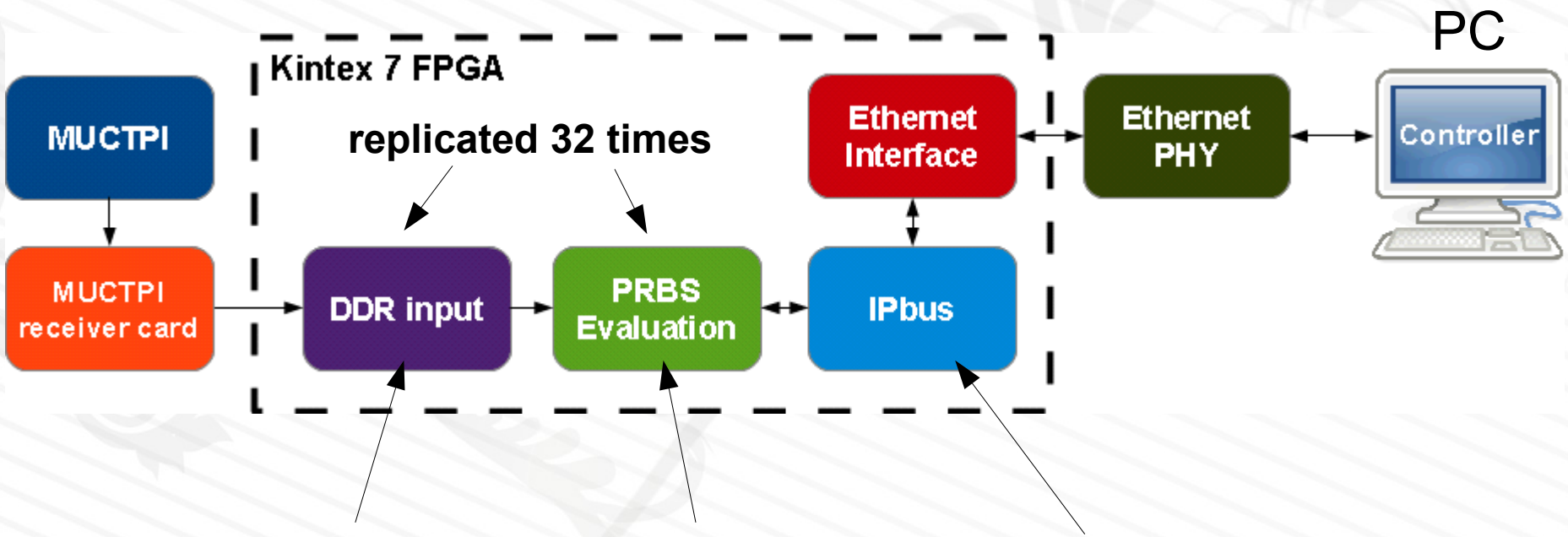


- ECL-LVDS level converter mezzanine card



Test system firmware

- Firmware block diagram



Adjust input phase

Detect transitions

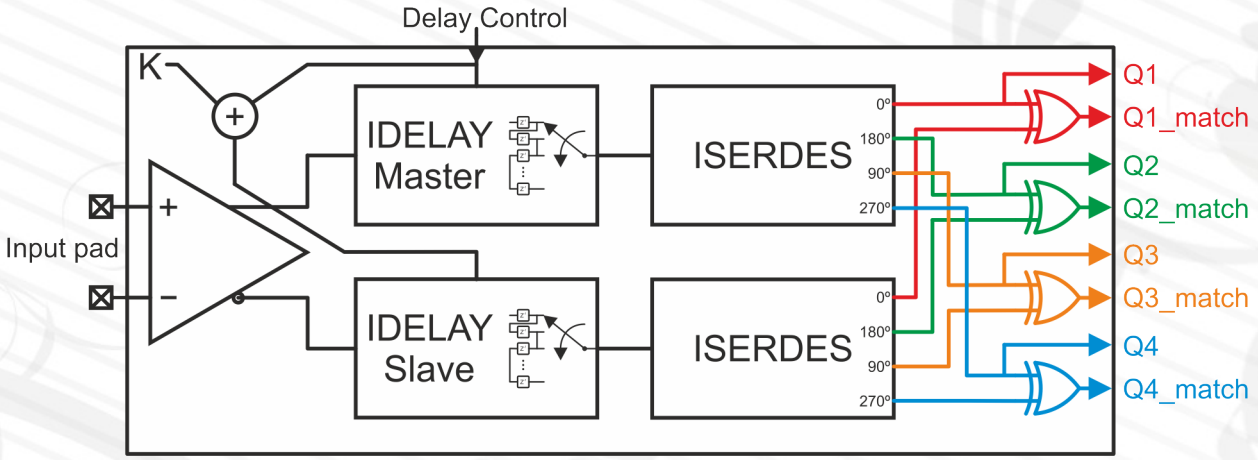
PRBS receiver

Count bit errors

Control the hardware

Read out test results

DDR input



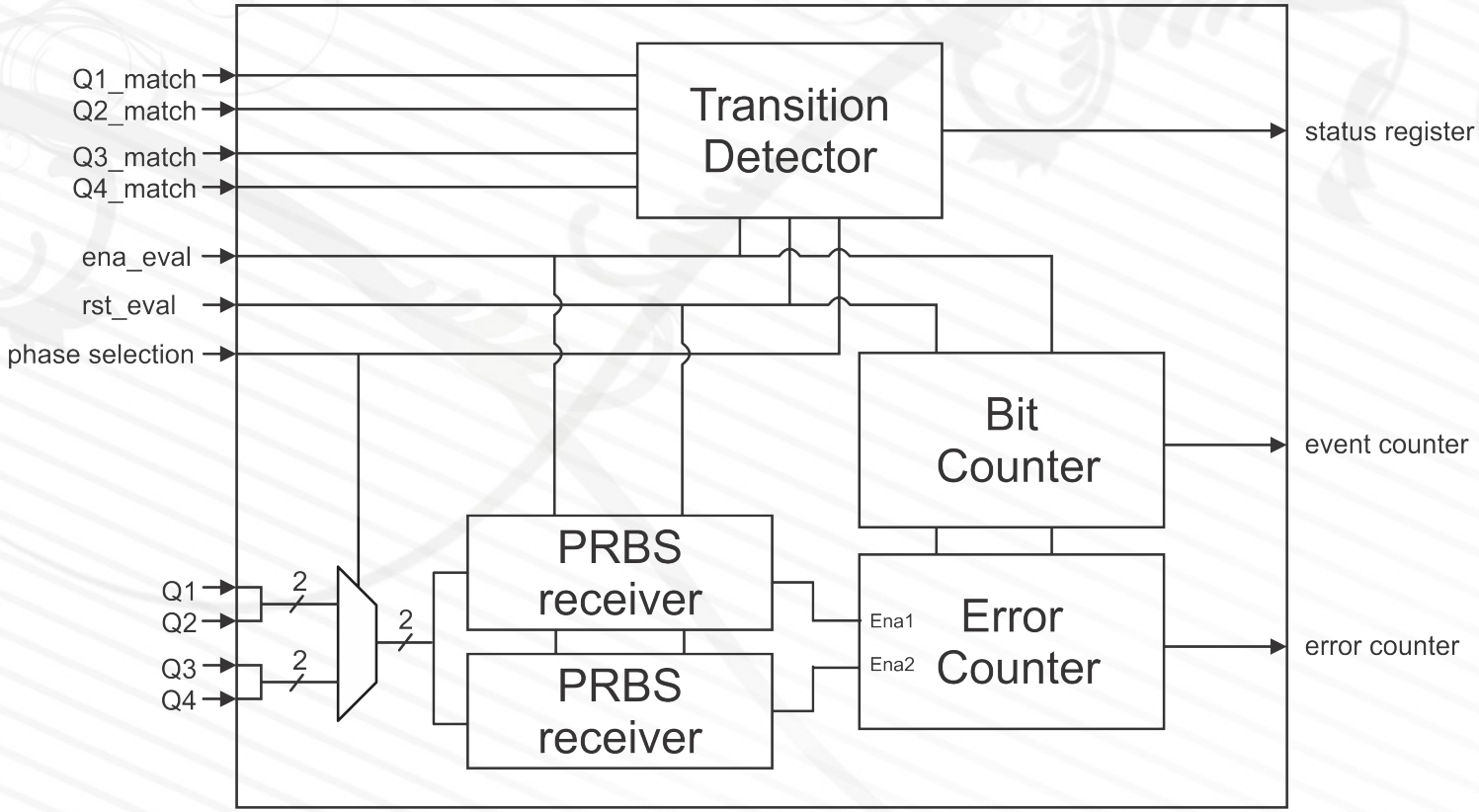
- Implements master and slave internal delay lines
- Adjusts the input phase from 0 to 3.125 ns in steps of 78.125 ps
- The delay offset between master and slave paths is one step
- XOR gates are used to compare the data from both paths to detect transitions

PRBS evaluation circuit

1) Monitor transition at a given sampling phase

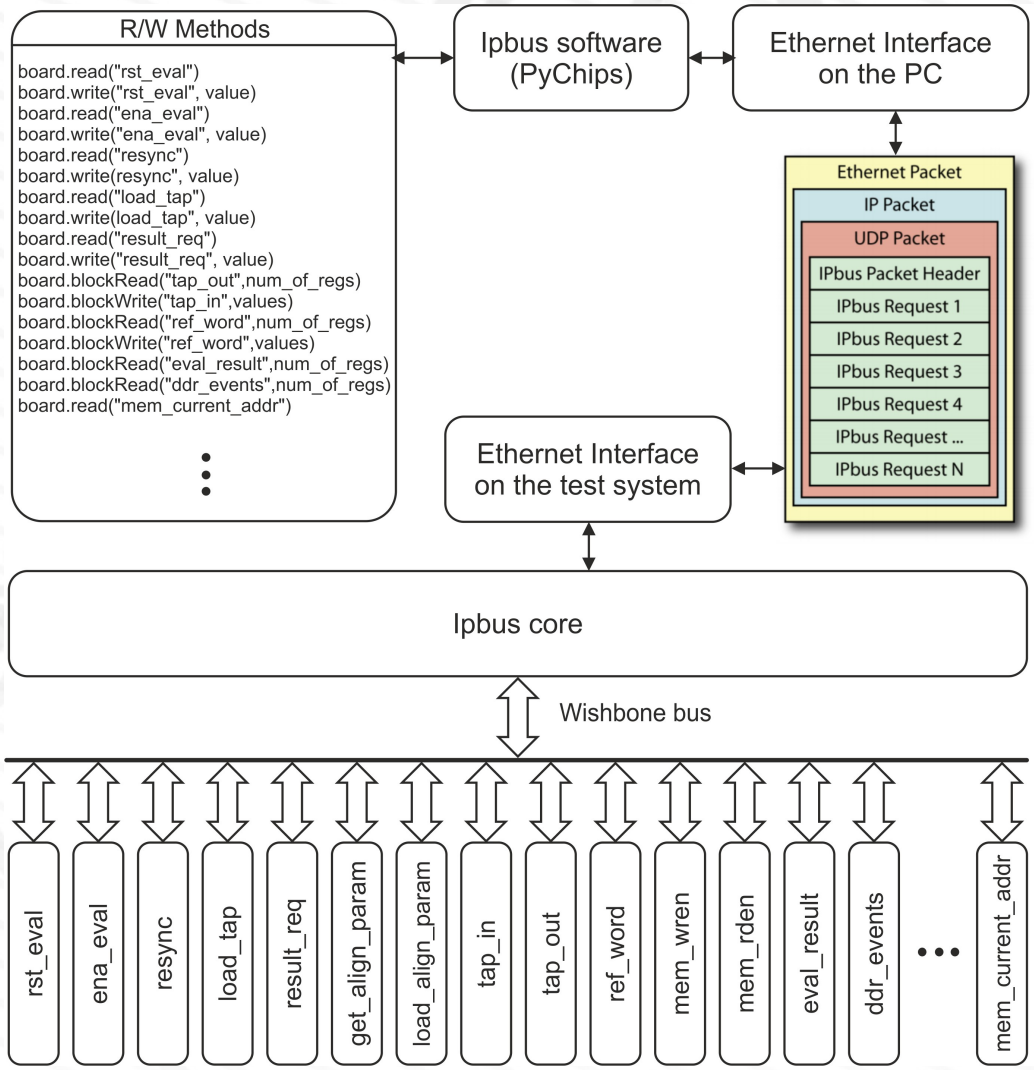
2) Compare the received PRBS data to a reference

3) Count the number of errors



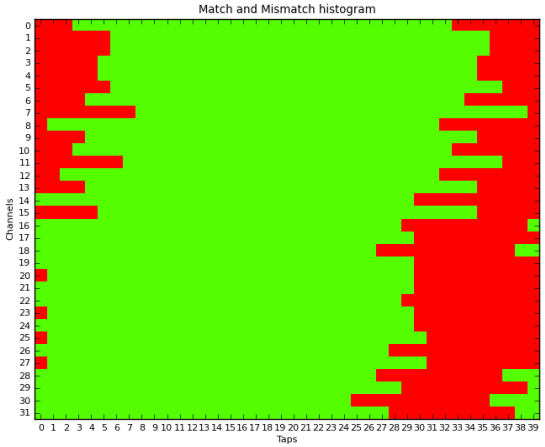
IPbus

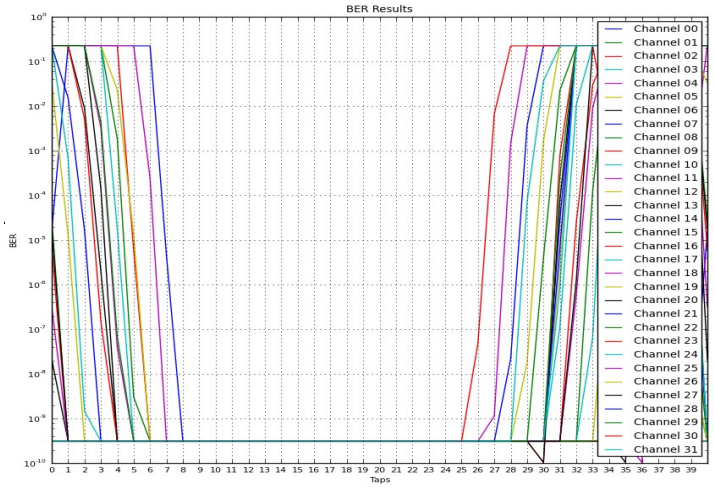
- Control / Readout via Ethernet
  - Configure registers in the FW
  - Read out test results
- Requests are encapsulated into an Ethernet packet
- IPbus core translates the requests to Wishbone transactions



Test system software

- Software written in Python scripting language
- Control and read out results from the test system generating:







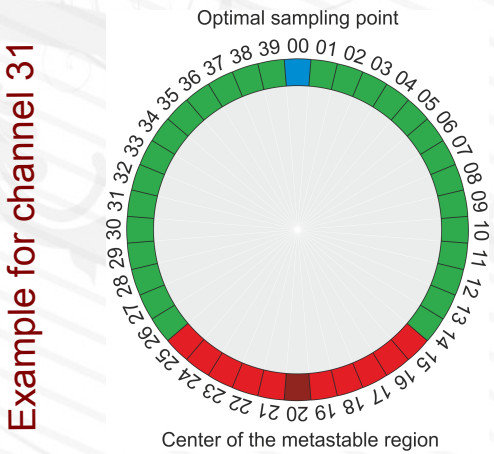
Taps	Channel 00 Errors	Channel 00 Events	Channel 00 BER	Channel 01 Errors	Channel 01 Events	Channel 01 BER
00	160382702	320710744	5.00085e-01	160382702	320710744	5.00085e-01
01	160453066	320852190	5.00084e-01	160453066	320852190	5.00084e-01
02	160350960	320647438	5.00085e-01	160350960	320647438	5.00085e-01
03	160332272	320609838	5.00085e-01	160332272	320609838	5.00085e-01
04	160391364	320728204	5.00085e-01	160391364	320728204	5.00085e-01
05	160432752	320811246	5.00085e-01	160432752	320811246	5.00085e-01

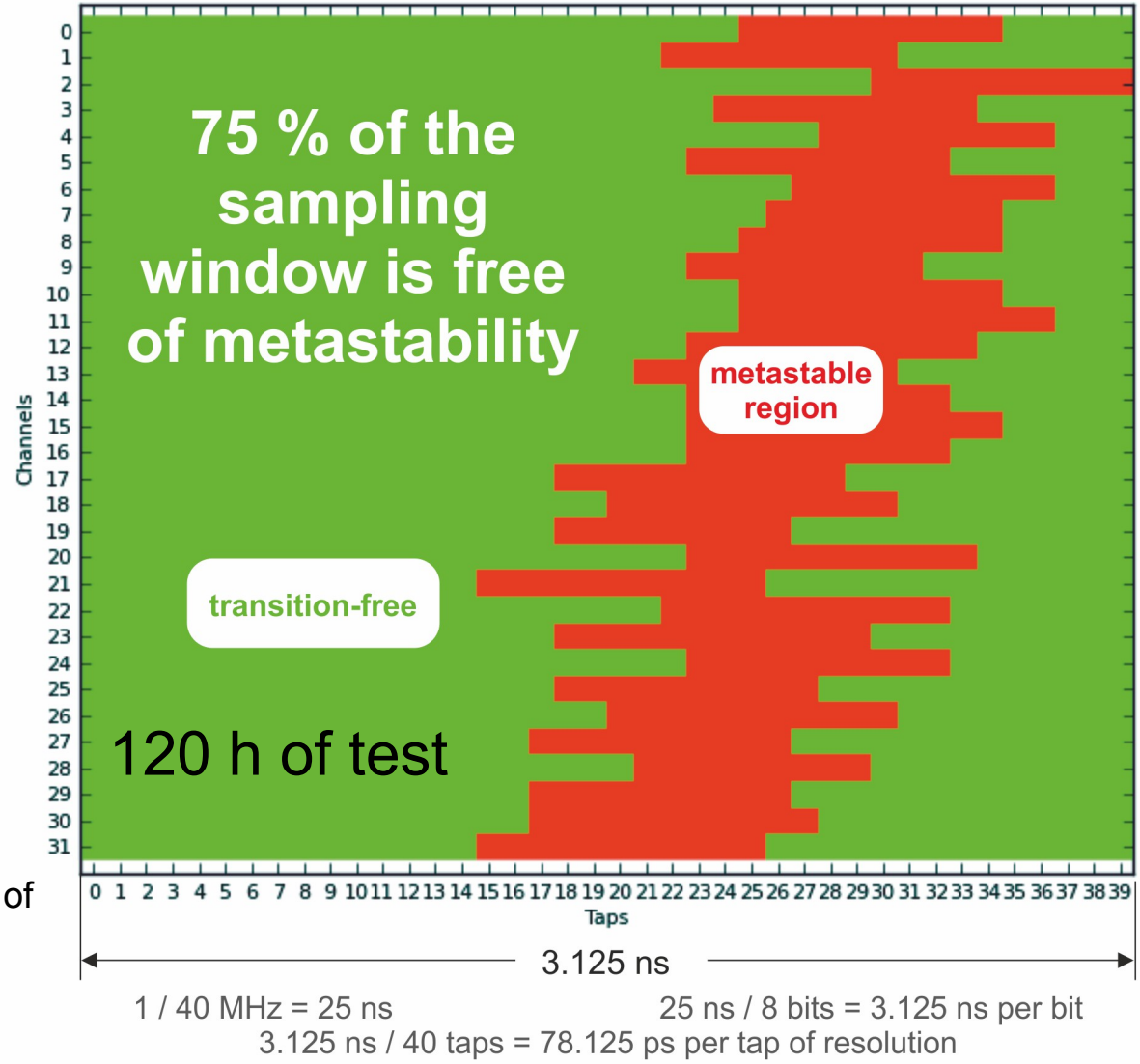
```
prbs_eval - INFO - ########################
prbs_eval - INFO - # Loading the tap: 00. #
prbs_eval - INFO - ########################
prbs_eval - INFO - Loading taps.
prbs_eval - INFO - Innitializing registers.
prbs_eval - INFO - Resetting evaluation.
prbs_eval - INFO - Synchronization request.
prbs_eval - INFO - Activating the evaluation.
prbs_eval - INFO - Wainting for 0.1 seconds until request the results.
prbs_eval - INFO - Desactivating the evaluation.
prbs_eval - INFO - Printing the results.
prbs_eval - INFO - CH.: 00    Match: True Errors: 0000000000  Events: 31638256.
prbs_eval - INFO - CH.: 01    Match: True Errors: 0000000000  Events: 31638256.
```

Test results

Test results

- Transition detection
  - Comfortable timing margin
  - Reliable data reception can be achieved sampling data at the optimal sampling point

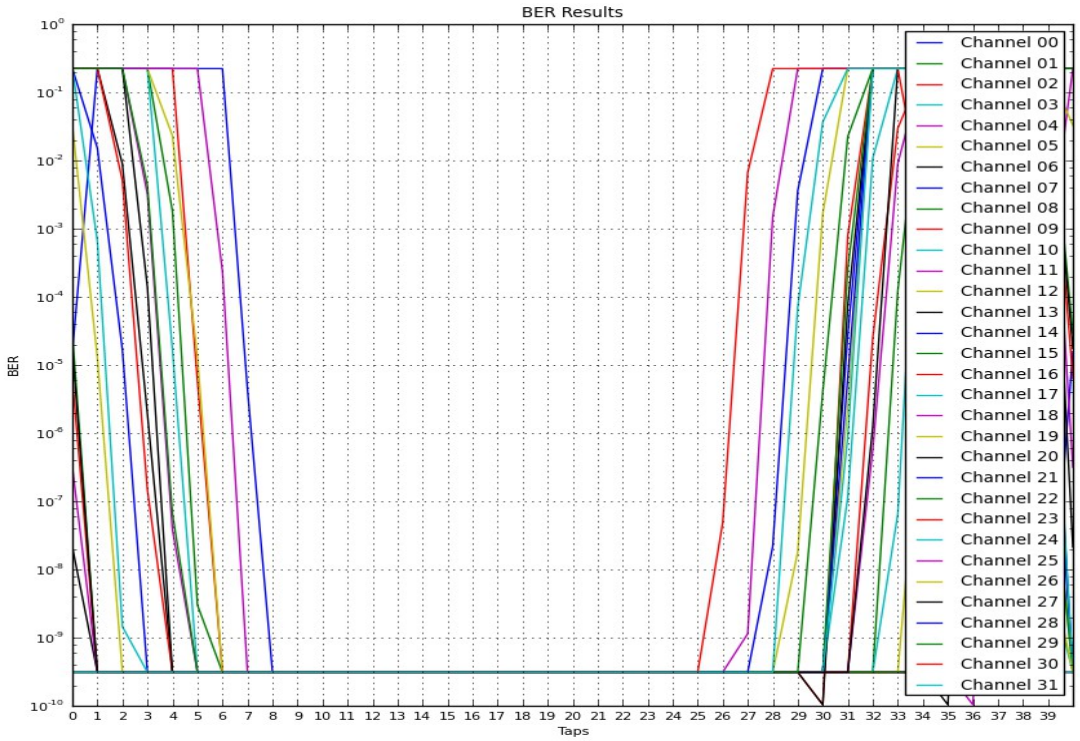




  - Additional tests of 120 h ► **no** drift of the clock phase > 80 ps

Bit error rate results

- No errors in 120 h
- BER < 10^{-15} @ 98.8 % C.L.
  - Upper limit of 1 x 10^{-15} ► 32 MUCTPI channels @ 320 Mbps during 24 h > LHC fill
  - More tests were performed (total > 500 h)

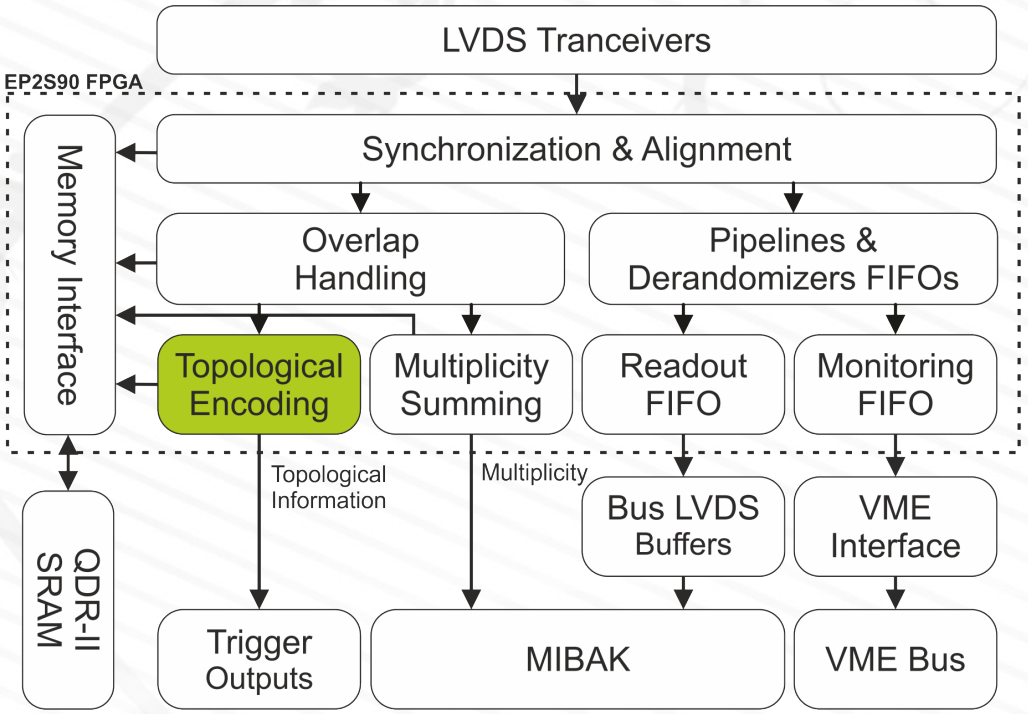


1 e-9 upper limit BER plot example

MUCTPI upgrade for LHC Run 2

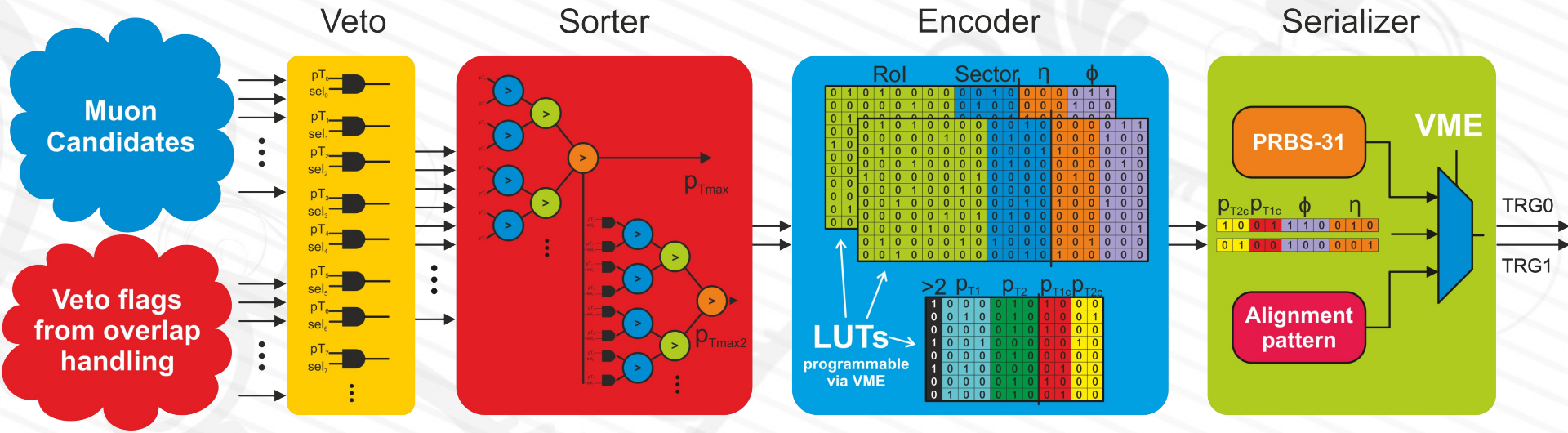
MUCTPI upgrade

- Topological encoding unit added
  - Extract & encode topological information from the two highest-p_T candidates
  - Transmit the data through the electrical trigger outputs at 320 Mbps.
- In addition:
  - Write debugging information from the topological encoding unit to memory
  - Rate monitoring counters
    - One candidate
    - Two candidates
    - More than two candidates.
  - Topological information is serialized using DDR outputs.



MUCTPI upgrade

- Topological Encoding unit block diagram



- Logic is clocked at 4 times the BC (160 MHz)
- Veto unit
  - Suppress candidates flagged by the overlap handling unit

Sorter unit

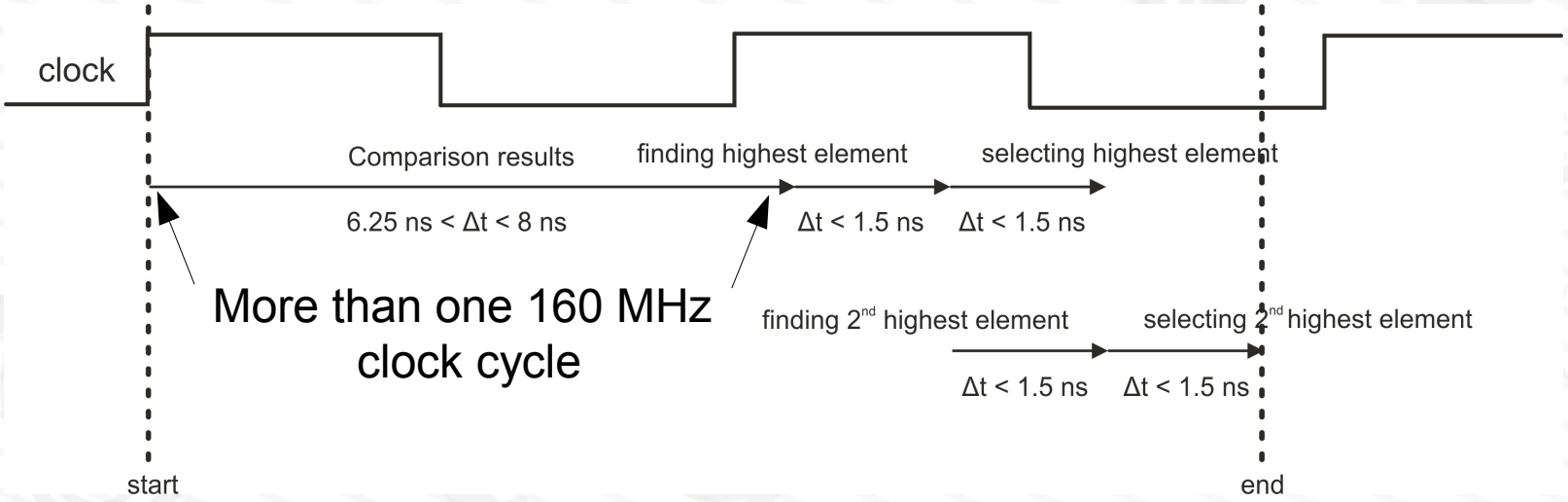
- Sorts the muon candidates according to their p_T values
- Outputs the sector number, RoI and p_T value of the 1st and 2nd highest-p_T candidates
- Sequential sorting too slow ► parallel implementation
- Multiple comparators are used to find the 2 highest-p_T candidates from 26 candidates
  - For instance, 325 comparators are implemented to sort 26 candidates

Example for five candidates

	b	**c**	**d**	**e**
a	$p_{Ta} \geq p_{Tb}$	$p_{Ta} \geq p_{Tc}$	$p_{Ta} \geq p_{Td}$	$p_{Ta} \geq p_{Te}$
b	$\overline{(p_{Ta} \geq p_{Tb})}$	$p_{Tb} \geq p_{Tc}$	$p_{Tb} \geq p_{Td}$	$p_{Tb} \geq p_{Te}$
c	$\overline{(p_{Ta} \geq p_{Tc})}$	$\overline{(p_{Tb} \geq p_{Tc})}$	$p_{Tc} \geq p_{Td}$	$p_{Tc} \geq p_{Te}$
d	$\overline{(p_{Ta} \geq p_{Td})}$	$\overline{(p_{Tb} \geq p_{Td})}$	$\overline{(p_{Tc} \geq p_{Td})}$	$p_{Td} \geq p_{Te}$

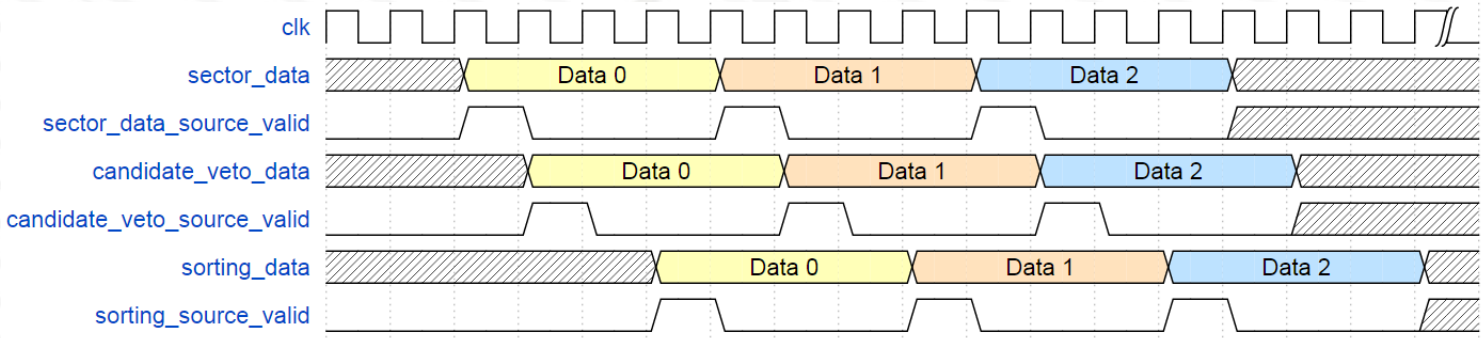
Sorter unit

- Timing diagram



– Multicycle timing constraint of the sorting unit ► path propagation delay = 2 clock cycles

– Topological encoding timing diagram:



Encoder unit

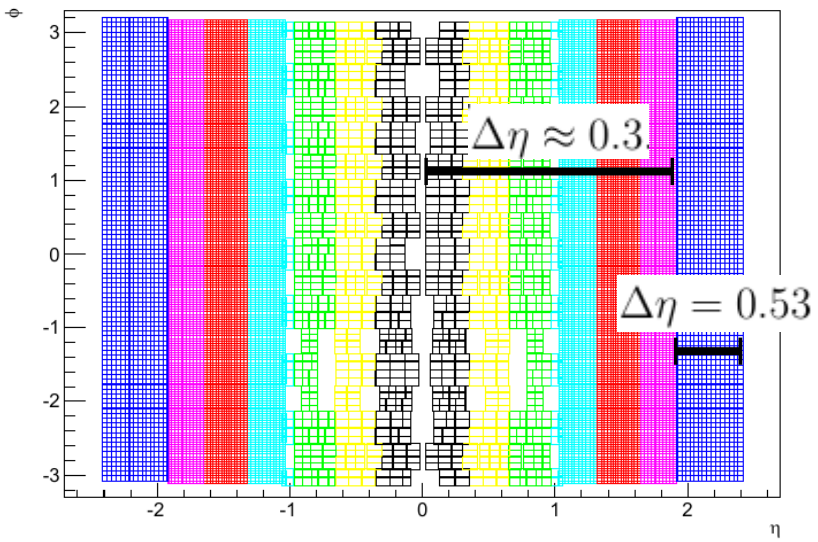
- 8 bits of topological information were allocated based on best trigger efficiency from physics simulations

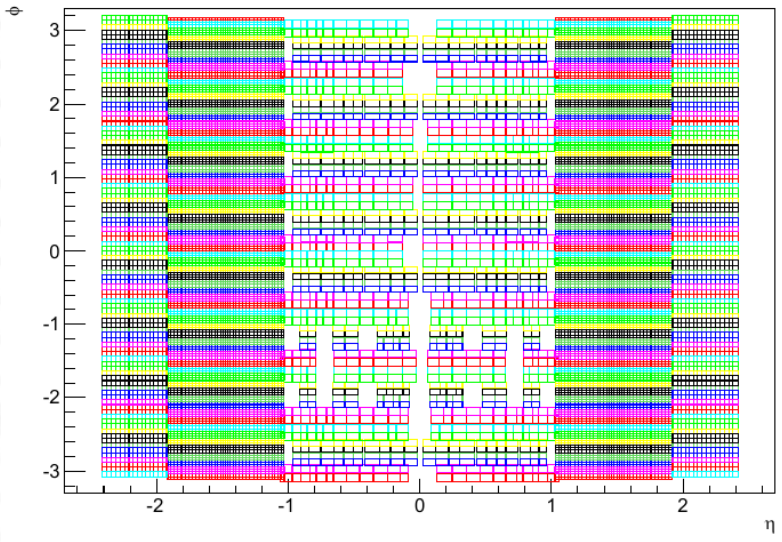
Bit position	7	6	5	4	3	2	1	0
Trigger output	η			ϕ			p_T	

→ $\Delta\eta \times \Delta\phi \approx 0.3 \times 0.1$

- 7 codes out of 8 used to encode Eta
- Remaining code used to indicate no candidate

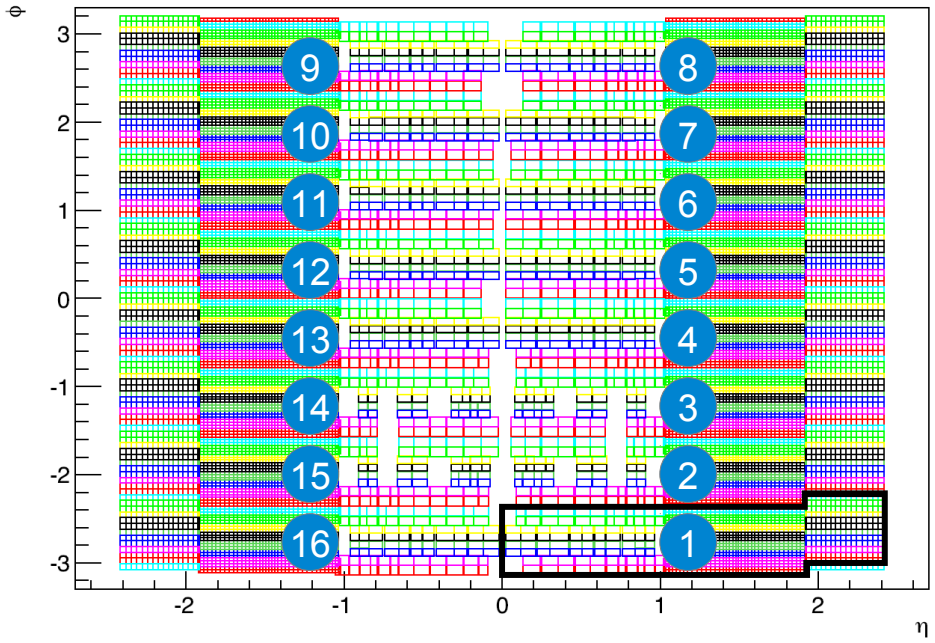
- 8 codes used to encode
- Uniform bin size $\Delta\phi < 0.1$





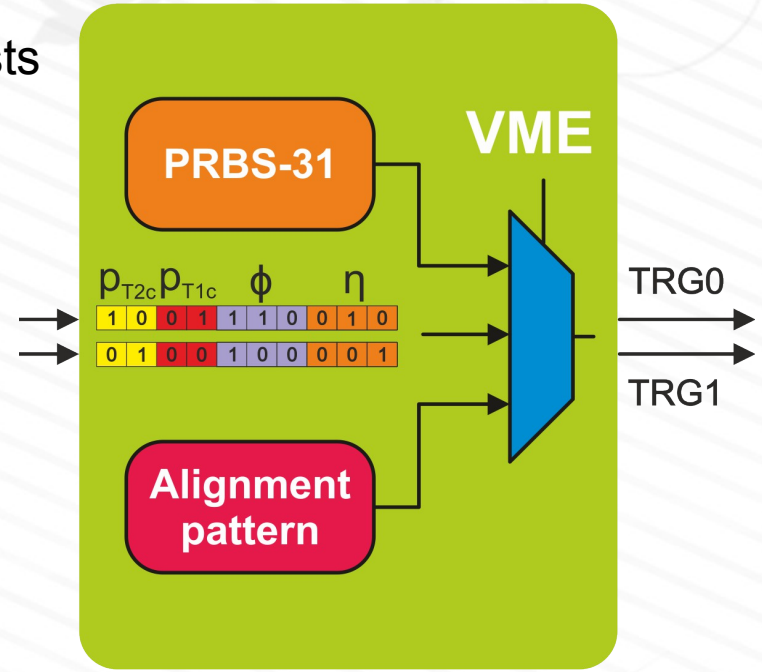
Geometry

- Each MIOCT encodes **7 x 8 = 56** geometric locations
- Global Eta, Phi coordinate based on octant number
- MUCTPI system encodes **16 x 56 = 896** geometric locations



Serializer

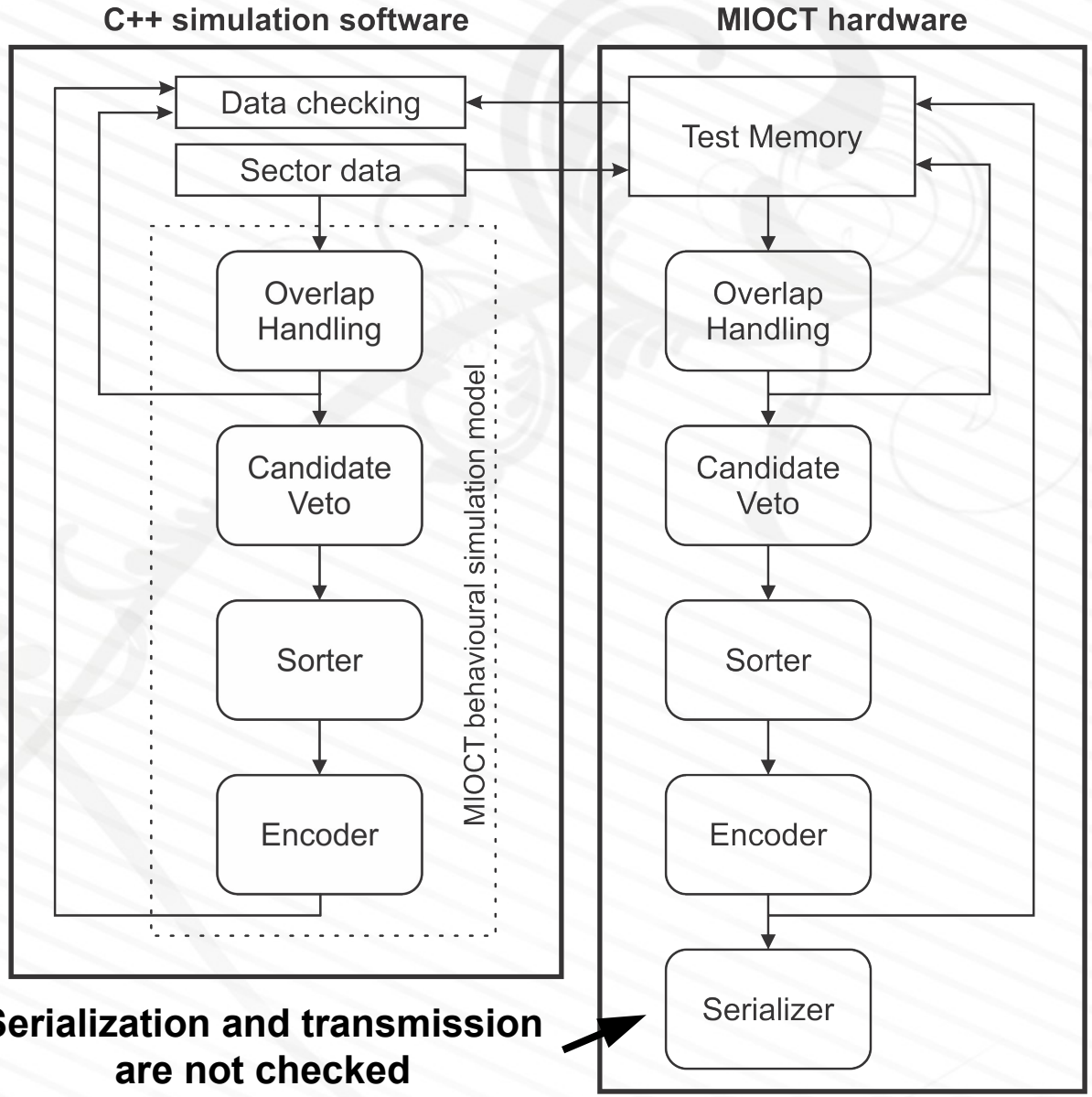
- Serializes 8-bit word at 320 Mbps
  - Uses DDR register outputs
- The two electrical trigger outputs carry the information from the 1st and 2nd highest-p_T candidates respectively
- Generates a fixed 16-bit training pattern
  - Used for phase and word alignment in MuCTPIToTopo
- PRBS generator same as used in the feasibility tests
  - Used for testing



Verification tests of the MUCTPI firmware upgrade

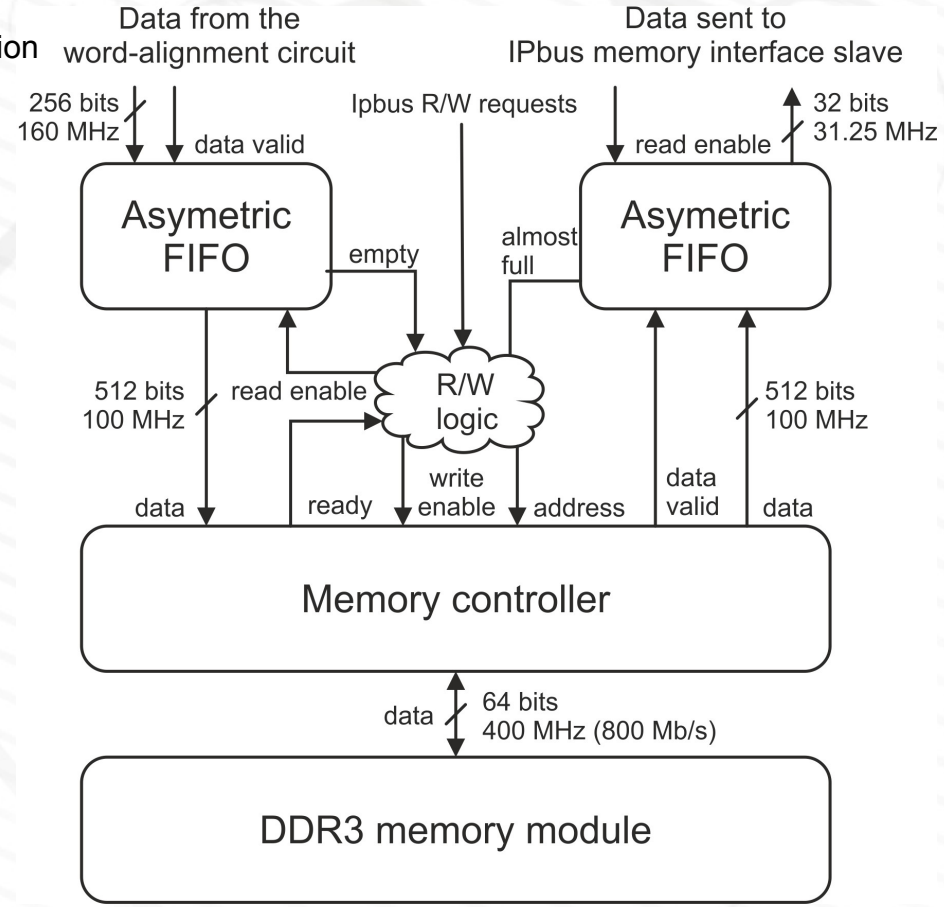
In-system verification

- Checks MIOCT hardware in operation
- Compare results from the hardware to a software model (C++)
- Write constrained random generated sector data to a test memory
- Test Memory:
  - Injects sector data to overlap handling unit
  - Reads out the generated veto flags and topology information
- Checked data from ~ 1.4 billions of bunches
  - No errors



Upgrade of the error rate test system and test result

- The error rate test system was upgraded
  - Receive serialized data from 32 MUCTPI electrical trigger outputs
  - Align them
  - Write them into a memory module
    - Where they can be read out by a computer for verification
- Word alignment circuit compares input data to predefined 8-bit reference value.
- Memory interface receives data from 32 channels
  - Groups it in 512-bit words
  - Writes them to the external memory module
- Test result:
  - Checked data from ~ 1 x 10^{10} bunches
    - No errors



Summary of the Verification tests

- Functional simulation: No errors in 185 thousand test patterns
- In-system verification: No errors in 1.4 billion test patterns
- Verification using the test system: No errors in 1×10^{10} test patterns

MUCTPI topological information can be transmitted and received reliably

Integration Tests

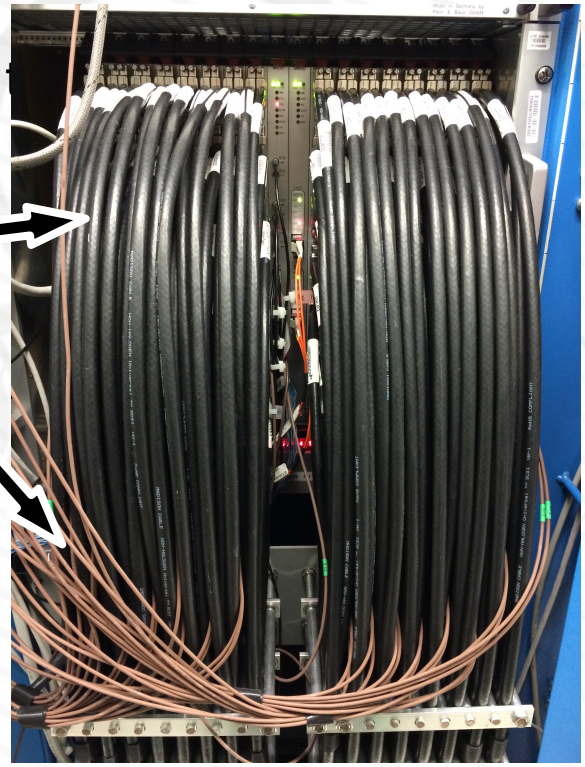
Integration tests

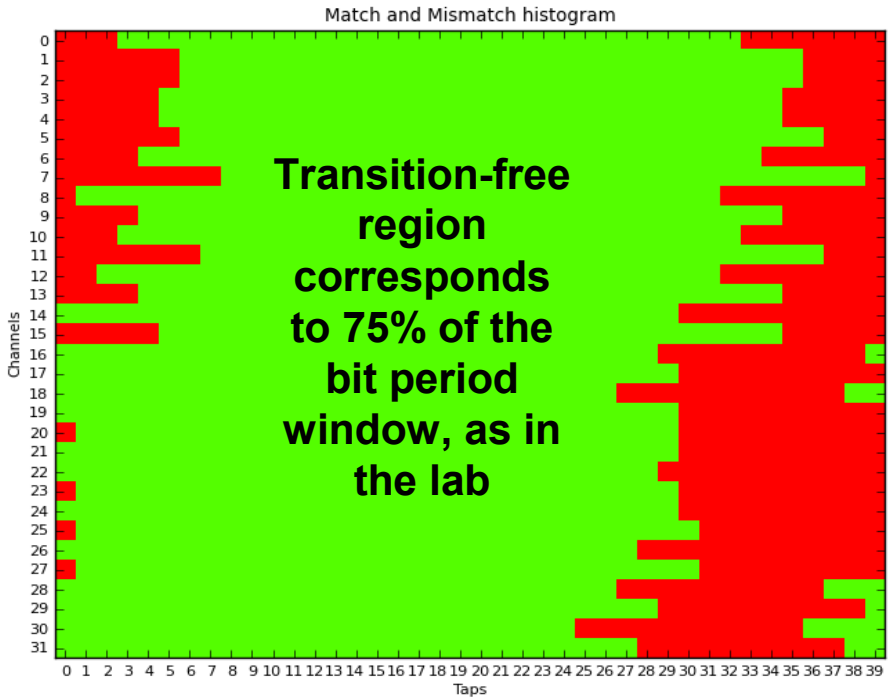
- Integration tests of **MUCTPI** with **MuCTPiToTopo** interface and **L1Topo** processor
  - Goal: Verify the interface between the systems.
- Received data by MuCTPIToTopo and L1Topo were written to on-chip memories, from where they were read out for verification
- Test MUCTPI / MuCTPiToTopo ► **2.6 M bunches** in an overnight test.
- L1Topo was included ► **258 k bunches** in a second overnight test

No errors were seen in the data transfer between MUCTPI, MuCTPiToTopo and L1Topo

MUCTPI hardware tests in USA15

- The MUCTPI system installed in USA15 was tested using t
  error rate test system
  - sector data input from the trigger sector logic
  - trigger output cables connected to the error rate test system







Different phase offset due to the different timing of the system in USA15

Conclusions

Conclusions

- Feasibility test demonstrated that the MUCTPI can reliably transmit data to L1Topo using the electrical trigger outputs operating at a bit rate of **320 Mbps**
- The BER is estimated to be lower than **1 x 10^{-15} (24 h)** with confidence level of **98.80 %**
- MUCTPI firmware was modified to send muon topological information to L1Topo
- Upgraded MUCTPI was checked by:
  - Computer simulation
  - In-system verification
  - Verification using the upgraded error rate test system
    - Serialization and transmission of topological information were also checked
- No errors were seen in the interface between MUCTPI, MuCTPiToTopo, and L1Topo in the integration test

Conclusions

- Test of the electrical trigger outputs of the MUCTPI system in USA15 was performed using the test system. Same timing margin as in the system in the laboratory
- System installed is ready for commissioning tests

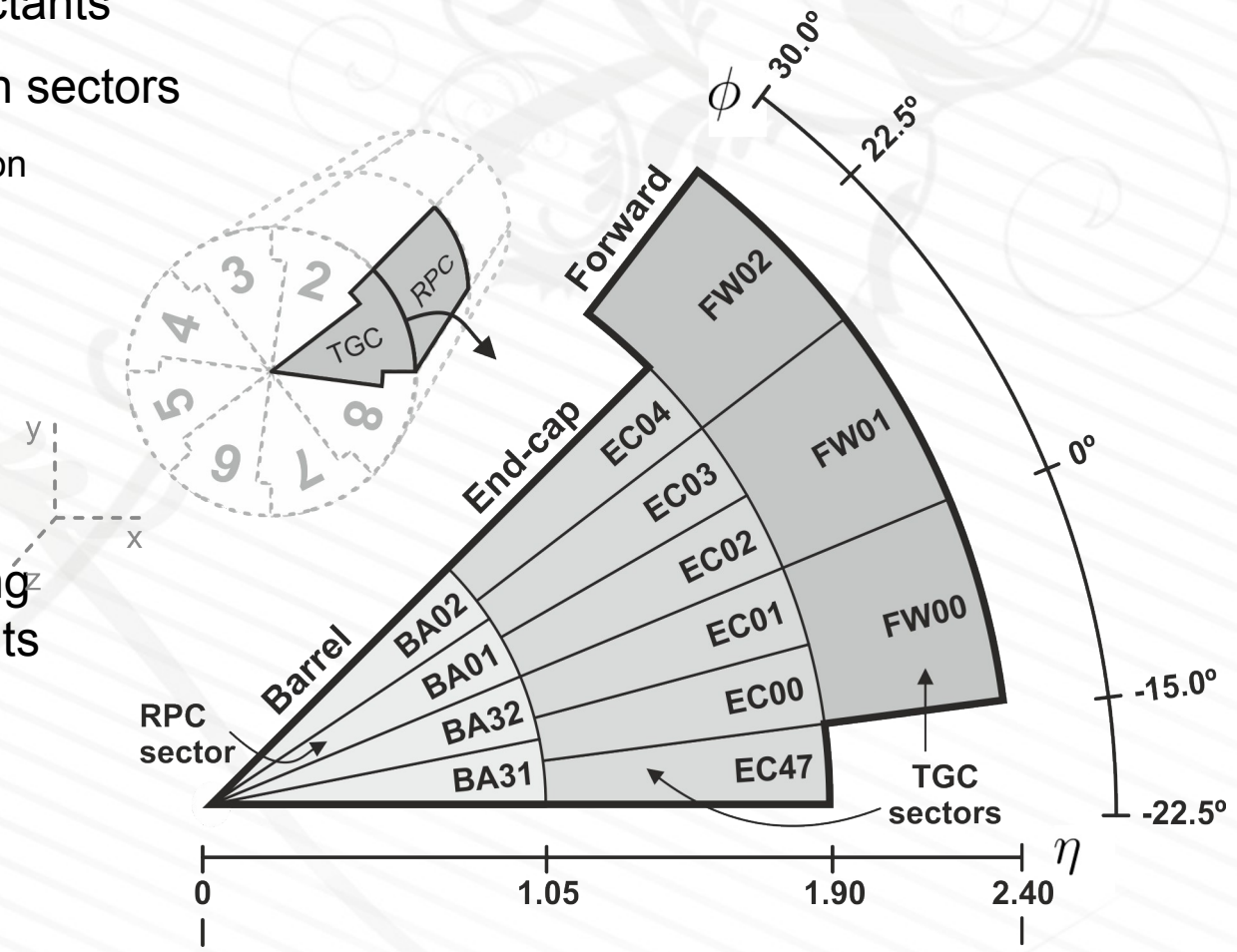
- Next steps:
  - Commissioning of the system in USA15 with L1Topo

BACKUP

Muon trigger detectors segmentation for the MUCTPI

- The muon trigger system is segmented in two halves
- Each half is segmented in octants
- Each octant contains thirteen sectors
  - Four sectors from the barrel region
  - Six sectors from the end-cap region
  - Three sectors from the forward region

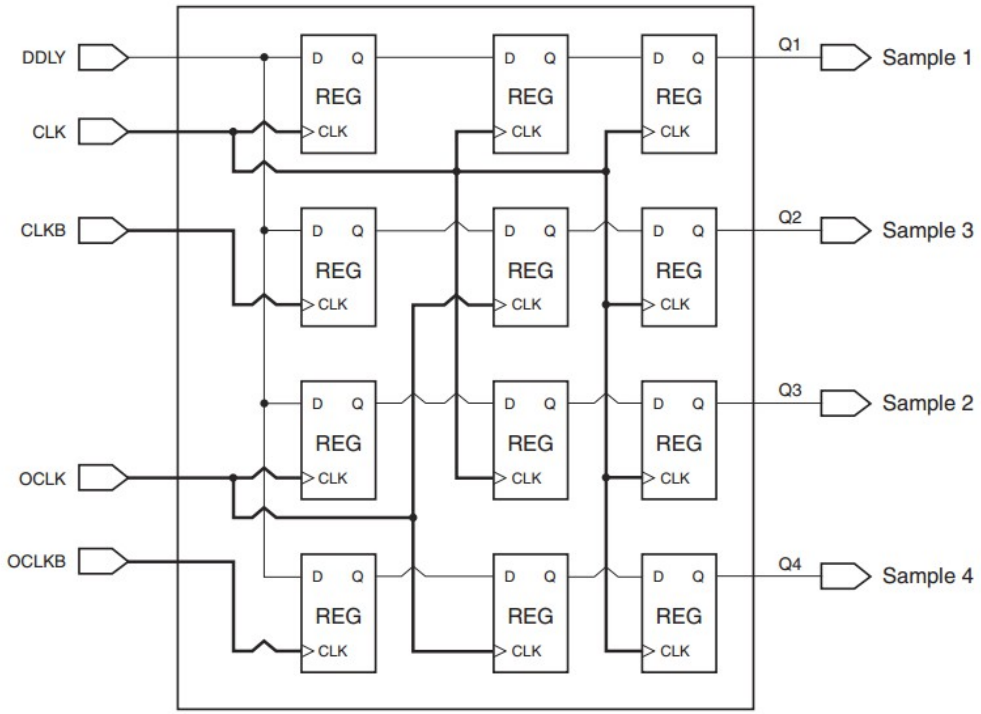
- The **MUCTPI** generates the total multiplicity combining the results from the 16 octants



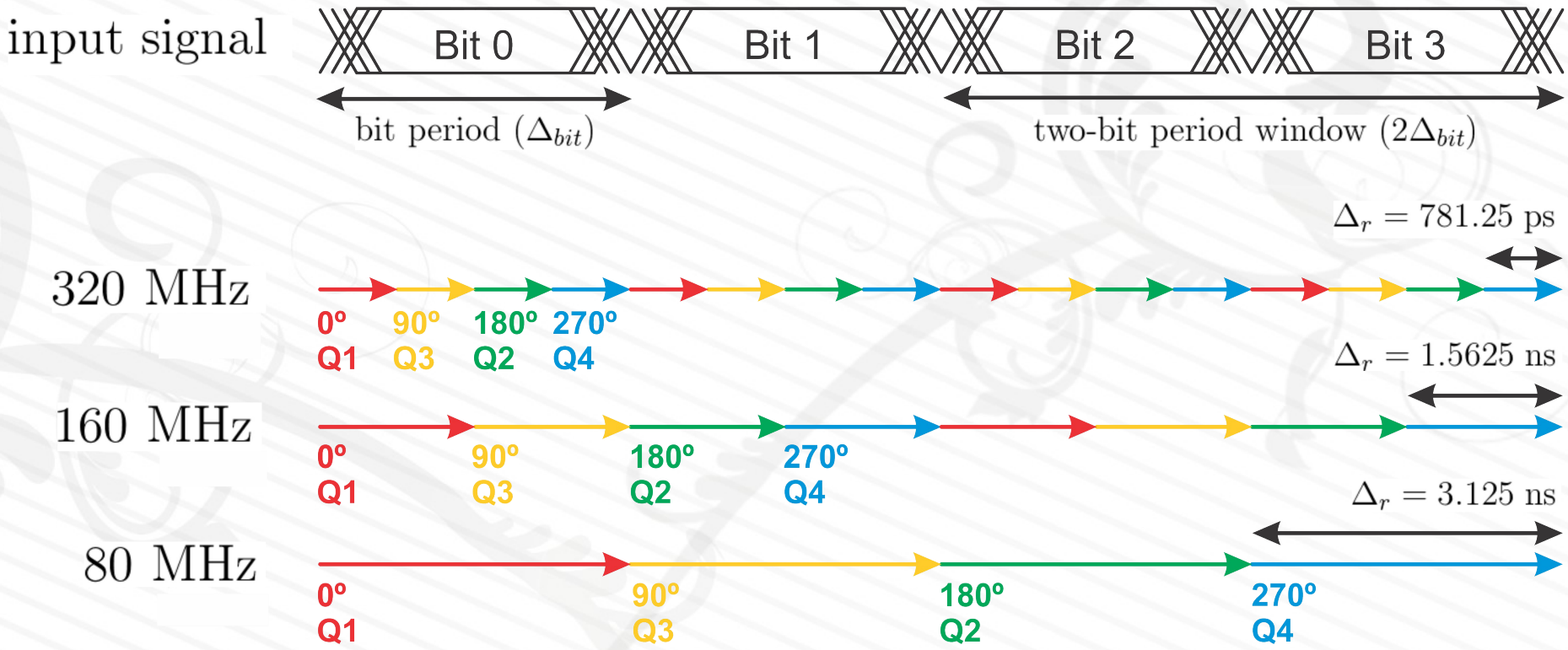
DDR input

- The programmable delay lines offer a maximum delay of 2.5 ns using selectable taps at intervals $\Delta_t = 78.125\,\text{ps}$
- However, the bit period is 3.125 ns in 320 Mbps
- Therefore oversampling the input data is necessary
- Internal serialization and deserialization blocks are used to sample the data at four different positions using four orthogonal clocks (90º phase-shifted)
- Now the delay required is given by:
  - Fclk represents the receiver clock frequency used to sample the data

$$\Delta_r = \frac{1}{4F_{clk}},$$



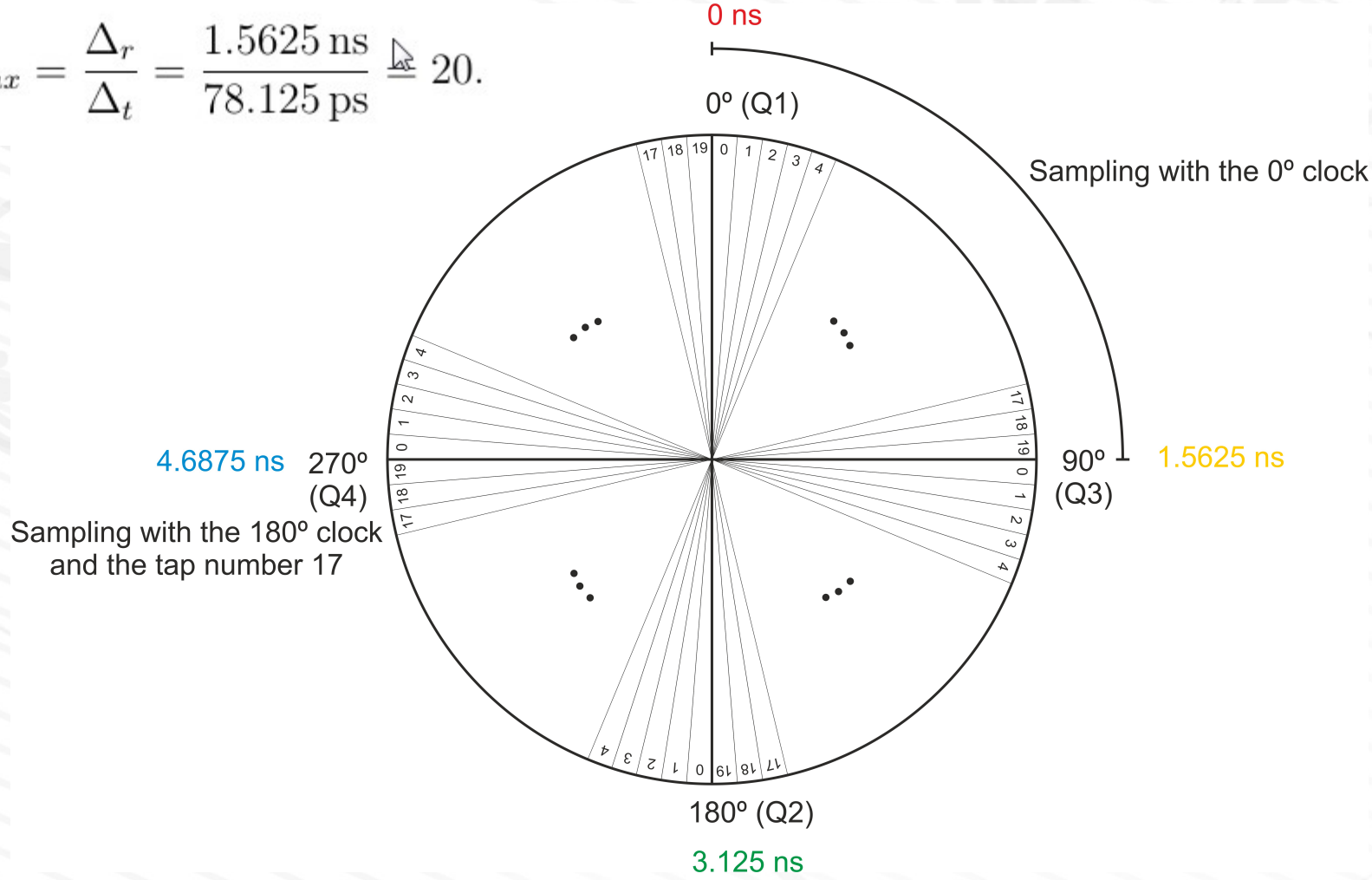
DDR input



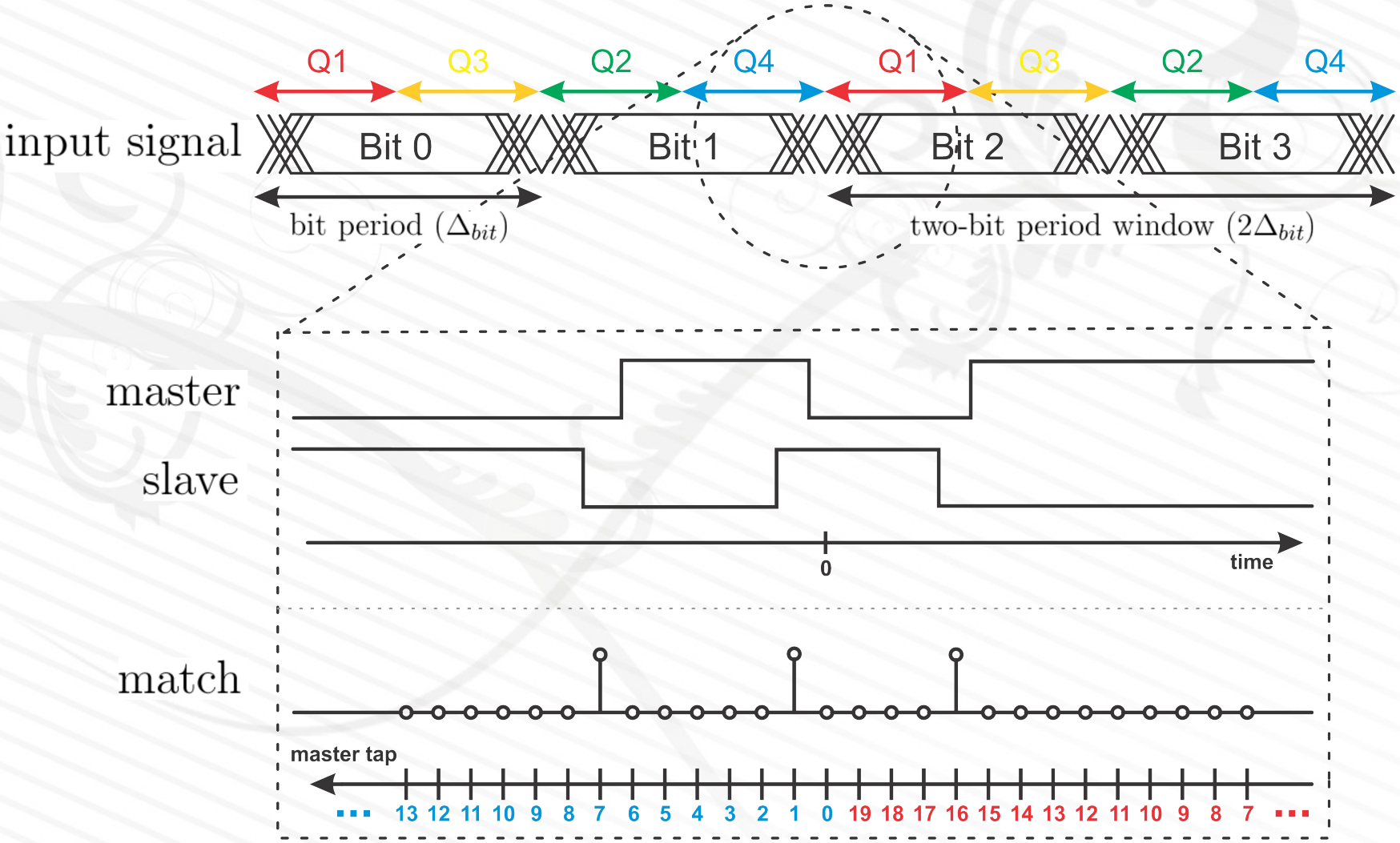
- Oversampling allows the receiver to run at 320 MHz or 160 MHz
- 160 MHz is selected because it simplifies the timing of the system
- The tap selection has to be limited in order to avoid overlap between the four segments of the sampling window

DDR input

$$N_{tmax} = \frac{\Delta_r}{\Delta_t} = \frac{1.5625\,\text{ns}}{78.125\,\text{ps}} \triangleq 20.$$



DDR input

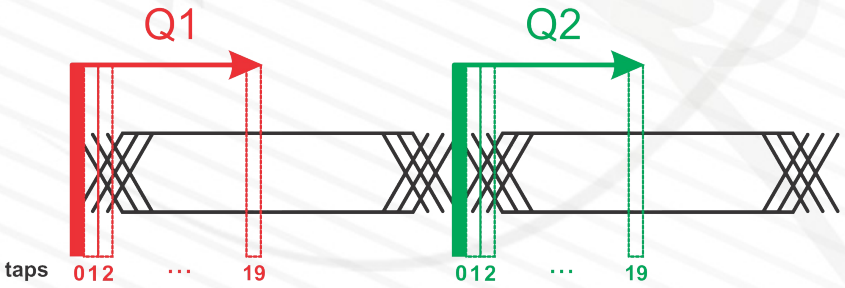

Q1
Q3
Q2
Q4
Q1
Q3
Q2
Q4
input signal
Bit 0
Bit 1
Bit 2
Bit 3
bit period (Δ_{bit})
two-bit period window ($2\Delta_{bit}$)
master
slave
0
time
match
master tap
13 12 11 10 9 8 7 6 5 4 3 2 1 0 19 18 17 16 15 14 13 12 11 10 9 8 7

DDR input

- Measurement is divided into steps
- Each step monitors two match ports simultaneously
- The steps are grouped in two stages
  - First 20 steps Q1 and Q2 are monitored, then Q3 and Q4.

Stage I

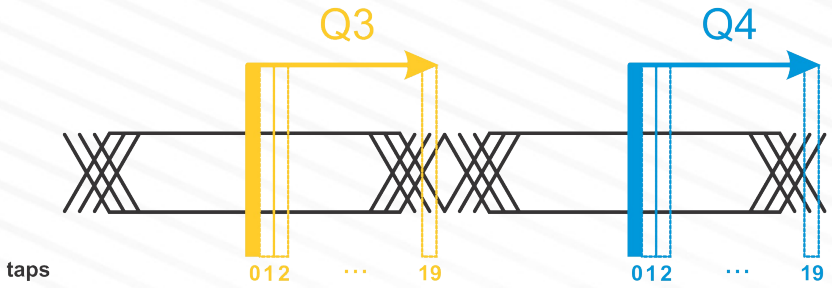
Steps from 0 to 19
Positions from 0 to 19 and 40 to 59



The first and third quarters of the sampling window are checked

Stage II

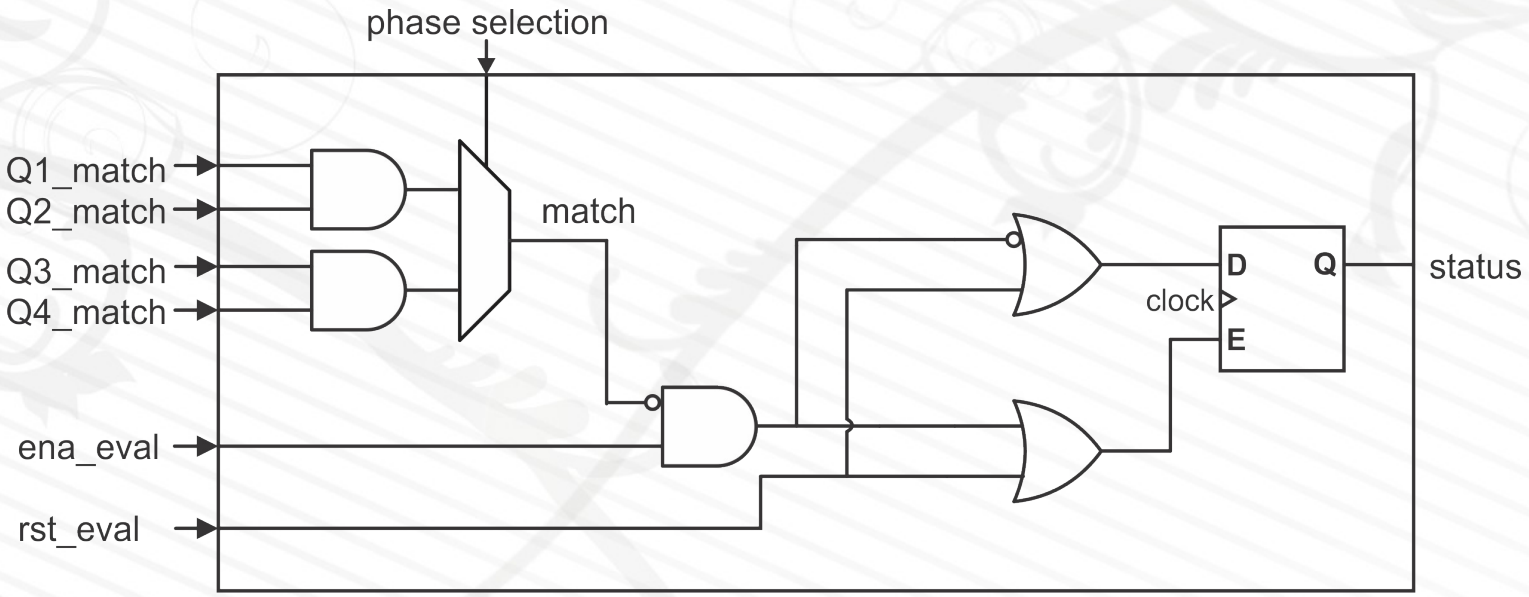
Steps from 20 to 39
Positions from 20 to 39 and 60 to 79



The second and fourth quarters of the sampling window are checked

PRBS evaluation

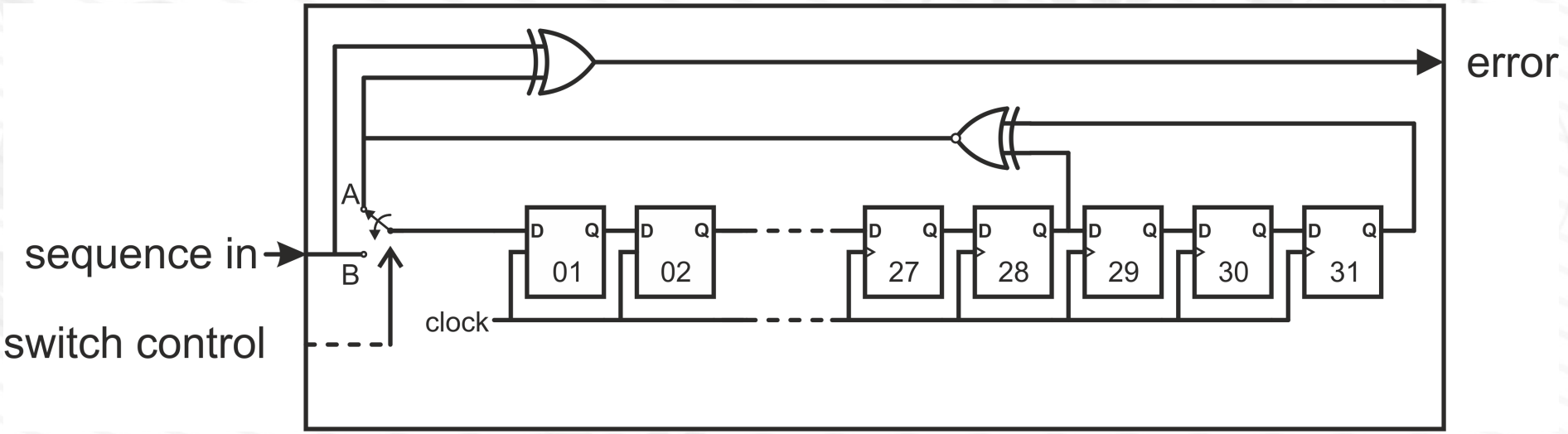
Transition detection circuit



The status is generated by latching the match signal low in a register

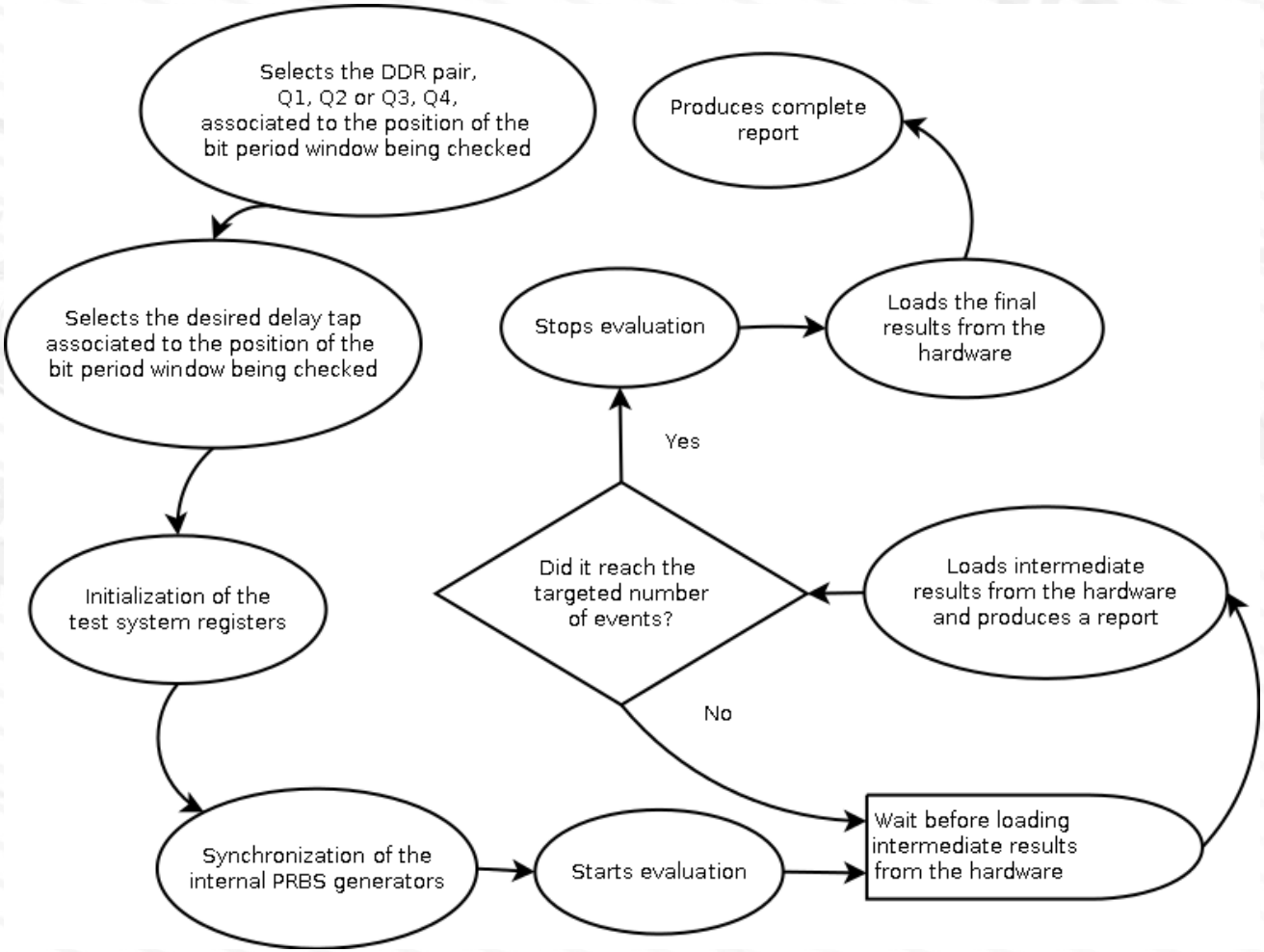
PRBS receiver

- Compares the input data to an internally generated pseudo-random sequence



- B: Register are loaded with the input data
  - Loading the 31 registers takes less than 100 ns
- A: Feedback path is connected
  - Generates a free running sequence synchronized with the input pattern

Software FSM



Sorter unit

- The second highest-pT candidate is found inverting the comparison results involving the highest-pT candidate in a second matrix
- Example: the element a is the highest candidate

	b	c	d	e
a	$p_{Ta} \geq p_{Tb}$	$p_{Ta} \geq p_{Tc}$	$p_{Ta} \geq p_{Td}$	$p_{Ta} \geq p_{Te}$
b	$\overline{(p_{Ta} \geq p_{Tb})}$	$p_{Tb} \geq p_{Tc}$	$p_{Tb} \geq p_{Td}$	$p_{Tb} \geq p_{Te}$
c	$\overline{(p_{Ta} \geq p_{Tc})}$	$\overline{(p_{Tb} \geq p_{Tc})}$	$p_{Tc} \geq p_{Td}$	$p_{Tc} \geq p_{Te}$
d	$\overline{(p_{Ta} \geq p_{Td})}$	$\overline{(p_{Tb} \geq p_{Td})}$	$\overline{(p_{Tc} \geq p_{Td})}$	$p_{Td} \geq p_{Te}$

Inverted comparisons

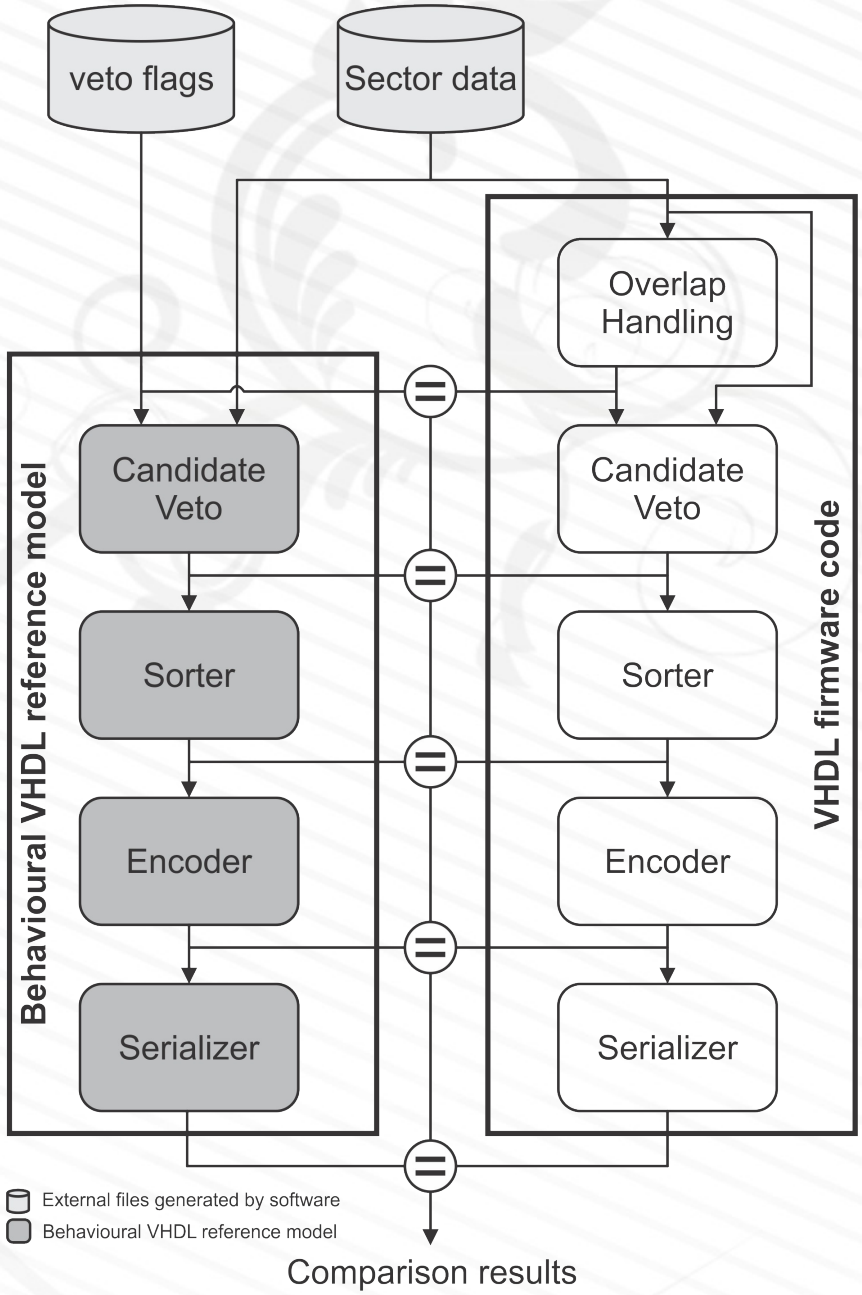
Transverse momentum encoding

- Two bits are used to encode the pT threshold, three of four codes are used to map the original six pT thresholds onto three programmable pT threshold values

	First candidate	Second candidate
00	p_{T_I}	p_{T_I}
01	$p_{T_{II}}$	$p_{T_{II}}$
10	$p_{T_{III}}$	$p_{T_{III}}$
11	not used	more than two candidates

VHDL functional simulation

- The VHDL functional simulation test bench consists of:
  - A VHDL behavioural reference model
  - A VHDL code of the firmware
- The results of the firmware simulation are compared to the reference response
- The veto flags and the four stages of the topological encoding units are checked individually
- The VHDL firmware code was verified by functional simulation using sector data information from approximately 185 k bunches without errors.



Acknowledgements

- I would like to thank my supervisors Stefan Haas and Augusto Santiago
- I am also grateful for the support from UFJF, CERN and CAPES
- I would like to acknowledge with gratitude my officemates Manoel Barros, Marco Ghibaudi and Dimitrios Kolotouros