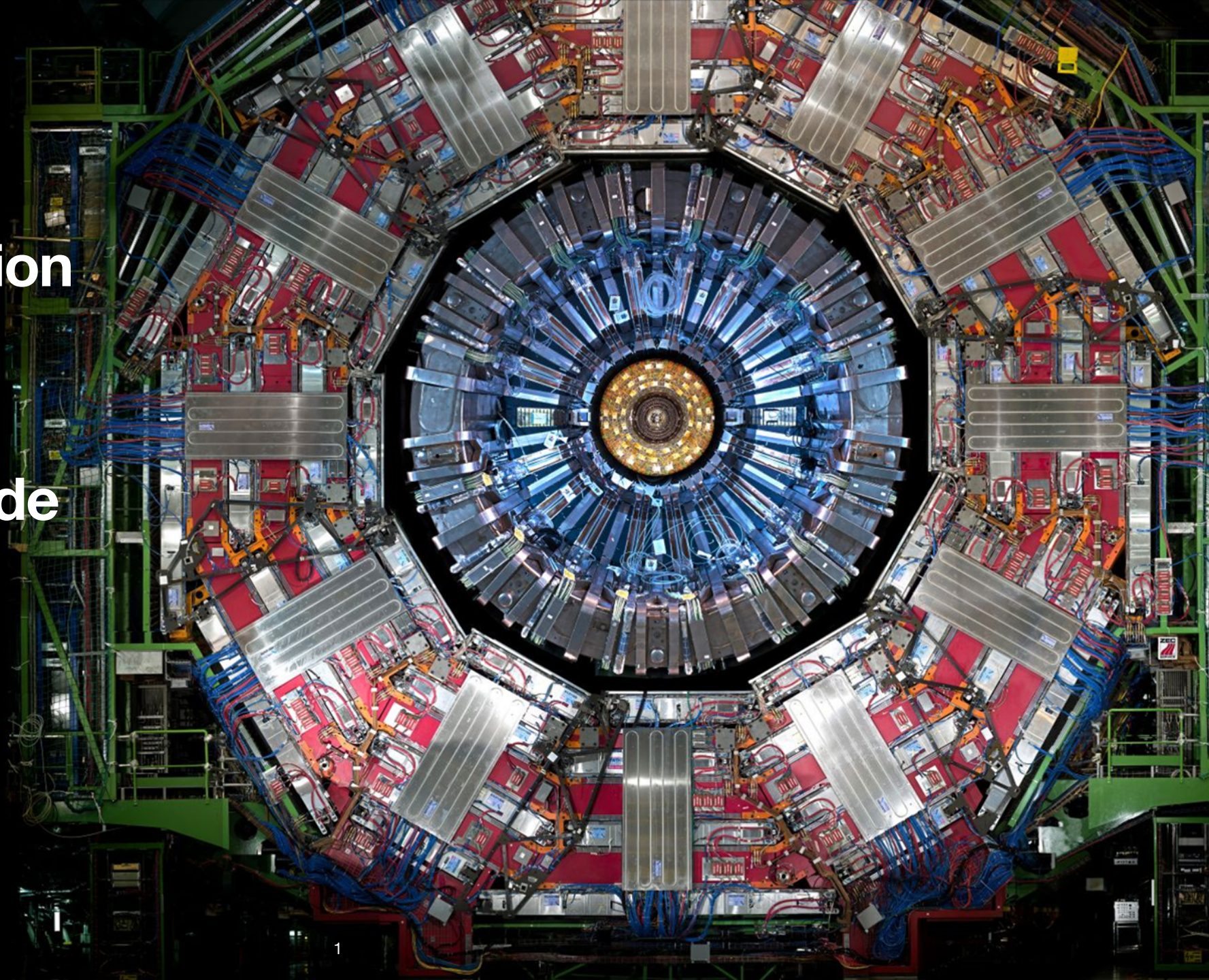
EP-ESE
ELECTRONIC SYSTEMS FOR EXPERIMENTS

Functional verification of ECON ASICs in the High-Granularity Calorimeter Upgrade of CMS



Matteo Lupi
on behalf of the CMS collaboration

TWEPP2024 - Glasgow - 01.10.24

Summary

- HGCAL
- ECON-T and ECON-D
- Verification strategy
- ECON-D right, case study
- Conclusions

HGCAL

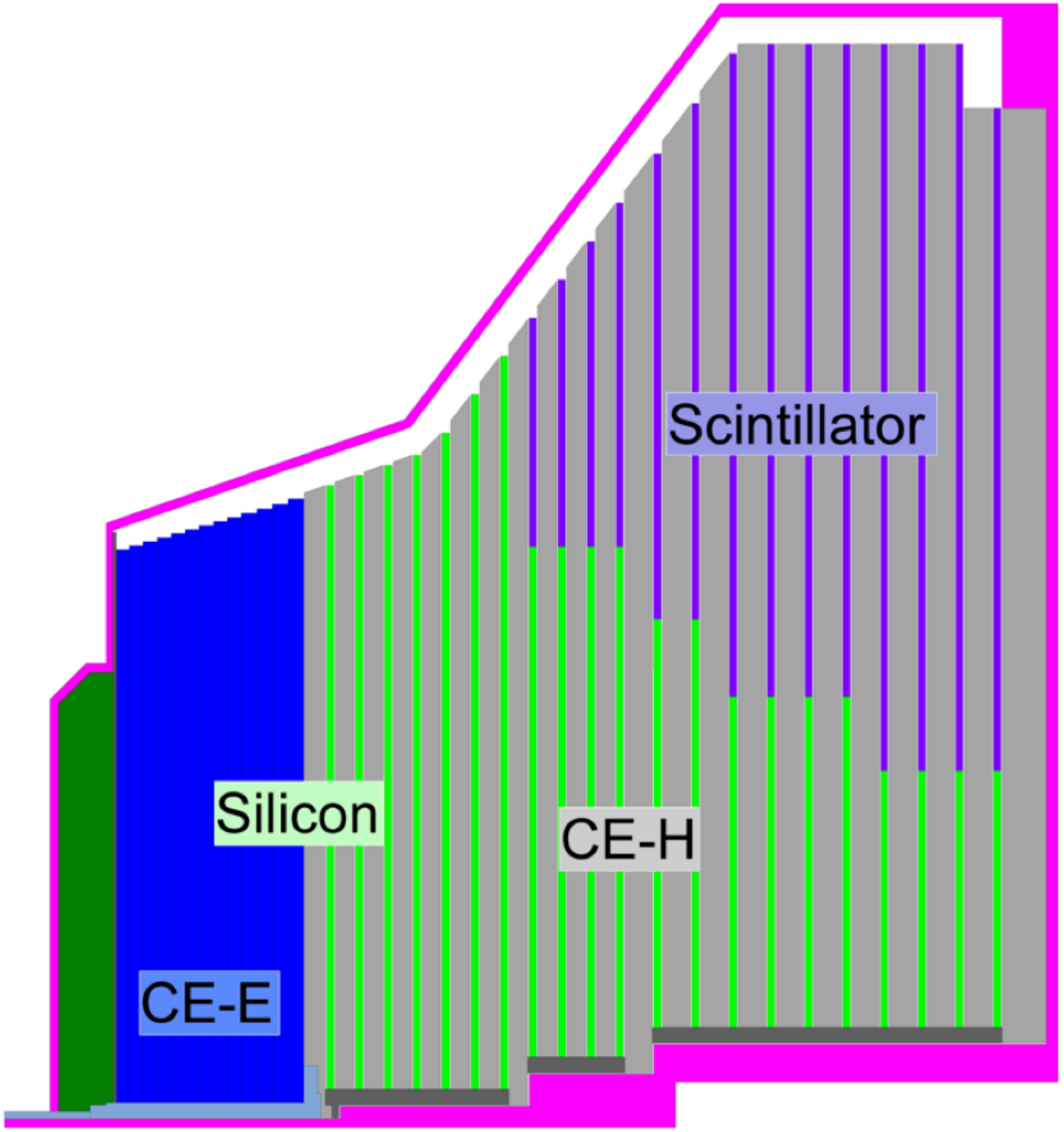
Introduction

High-Granularity Calorimeter (HGCAL)

CMS will replace the current endcap calorimeter system during LS3

HGCAL key features

- **47-layer** sampling calorimeter
- Two main areas
  - **Silicon sensors**
    - 620 m^2
    - 6 M channels
    - ~30 k hexagonal board design
    - Higher hit-rate/radiation area
  - **Scintillating tiles**
    - ~400 m^2
    - ~240 k channels
    - 4 k SiPM-on-tile
    - Lower hit-rate/radiation area

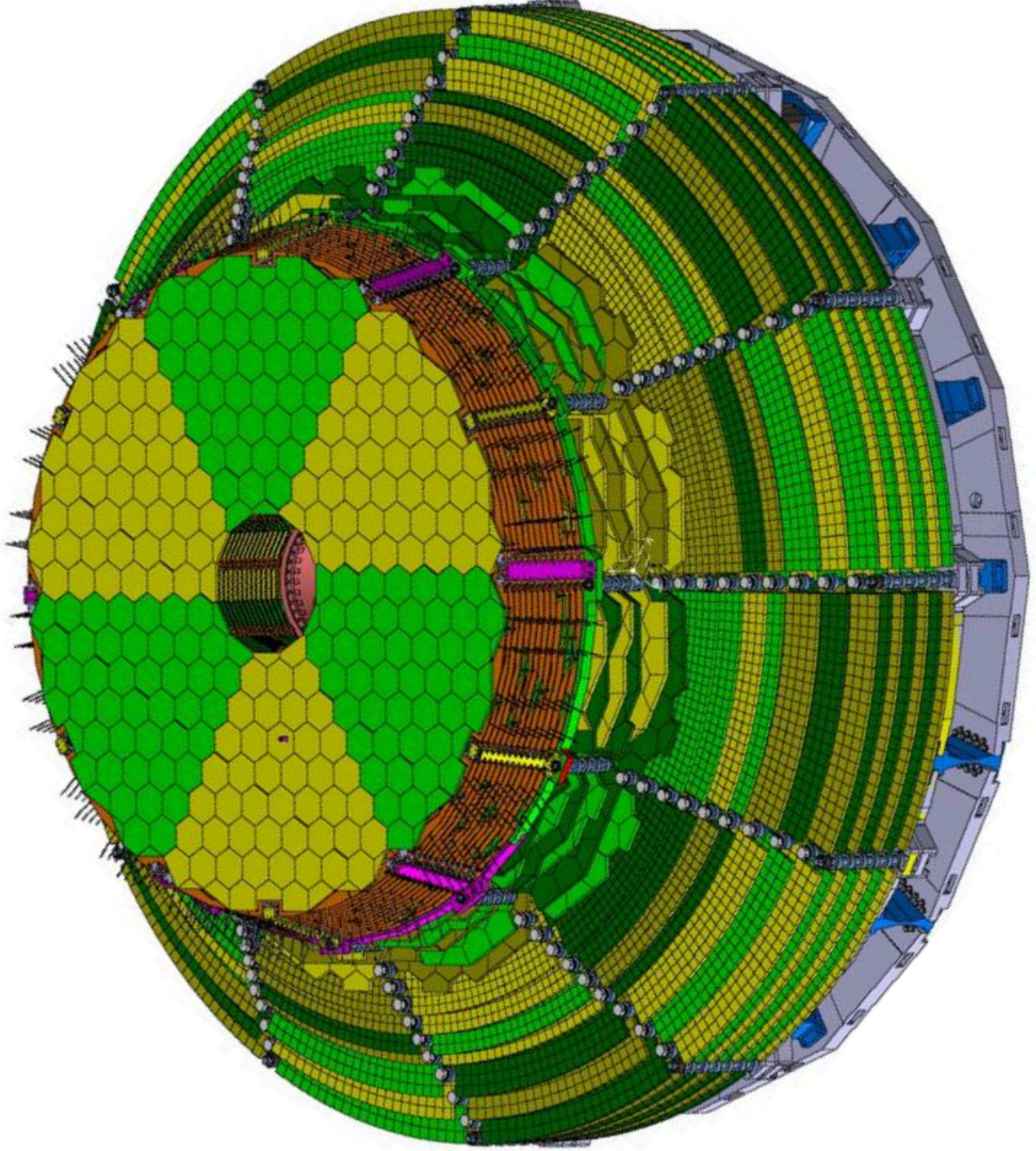


High-Granularity Calorimeter (HGCAL)

CMS will replace the current endcap calorimeter system during LS3

HGCAL key features

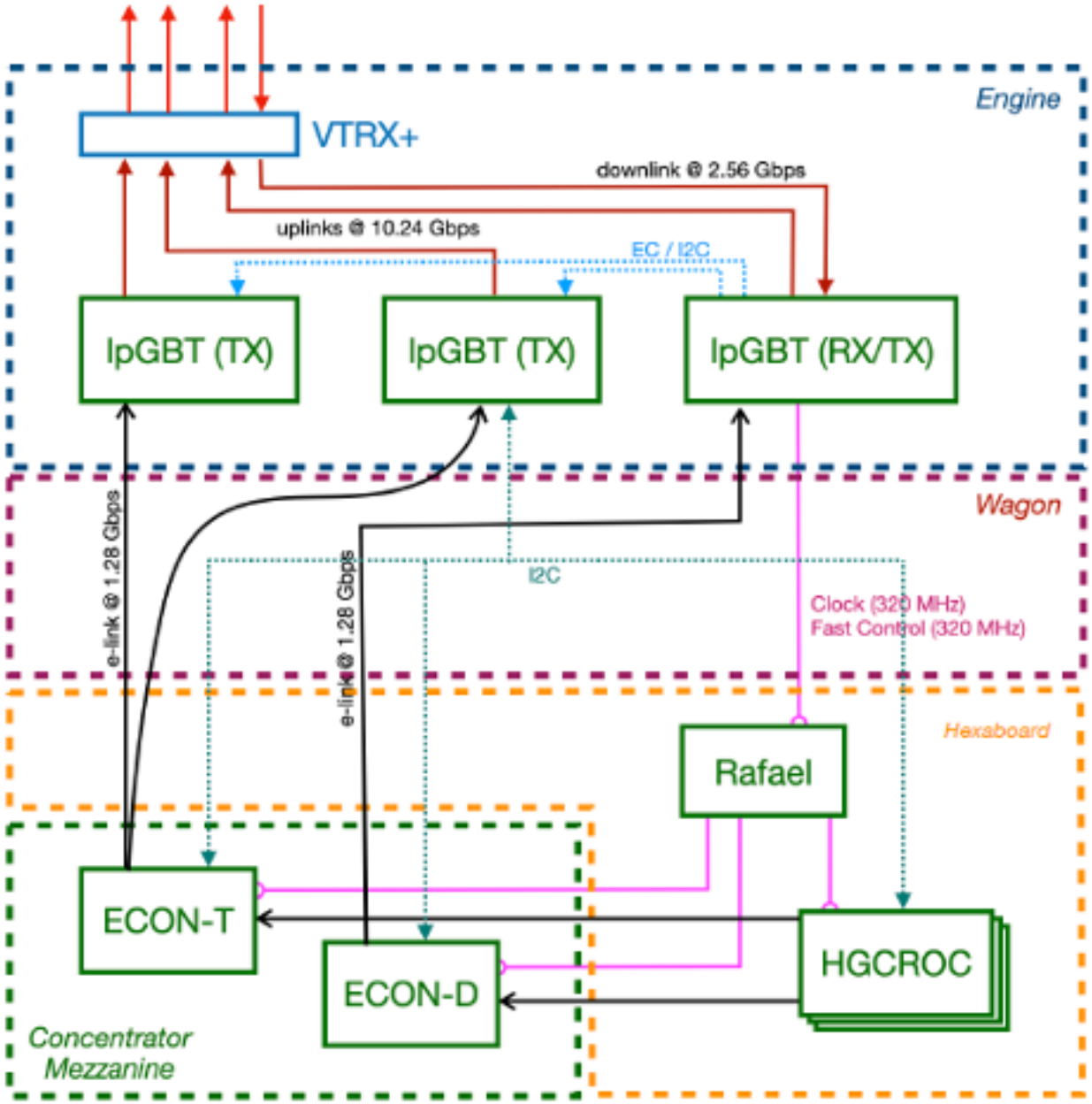
- **47-layer** sampling calorimeter
- Two main areas
  - **Silicon sensors**
    - 620 m^2
    - 6 M channels
    - ~30 k hexagonal board design
    - Higher hit-rate/radiation area
  - **Scintillating tiles**
    - ~400 m^2
    - ~240 k channels
    - 4 k SiPM-on-tile
    - Lower hit-rate/radiation area



Detector modules

Detector modules block diagram

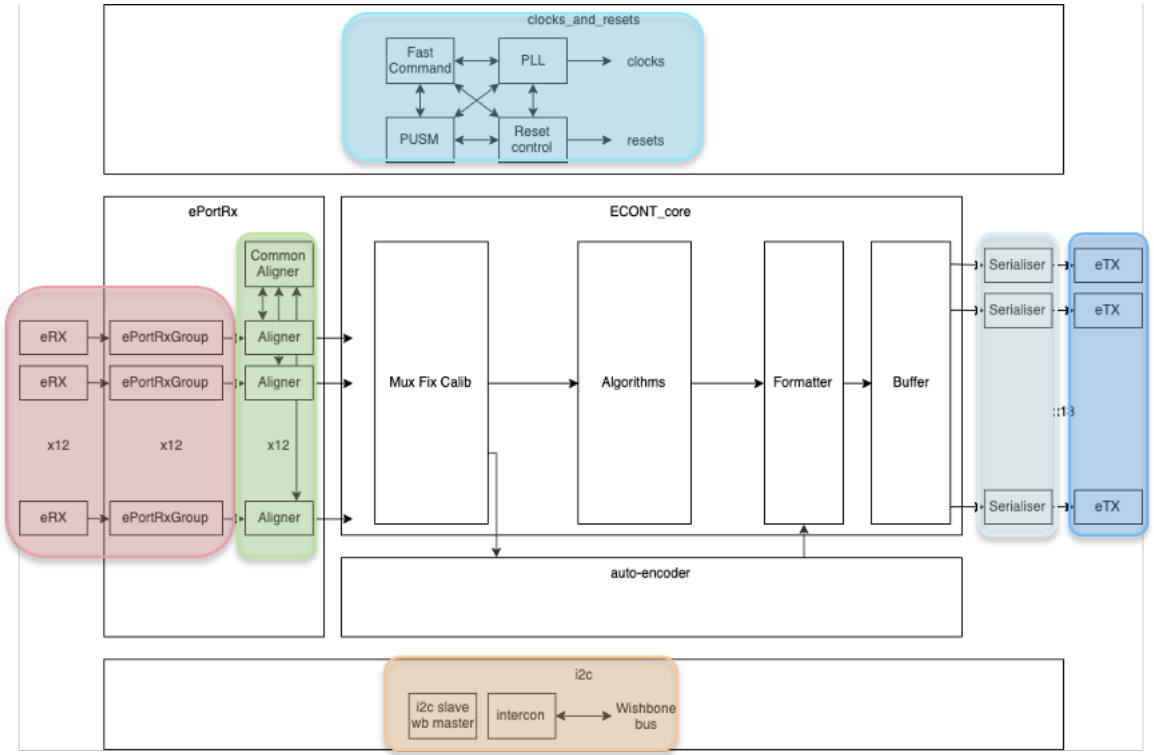
- ***LpGBT* & *VTRX+*** used as link to back-end
- ***Rafael*** fans out trigger and timing signals
- ***HGCROC*** reads out the sensors
- Endcap CONcentrators (***ECONs***) aggregate *HGCROCs* data
  - **ECON-T:** trigger concentrator
  - **ECON-D:** data concentrator



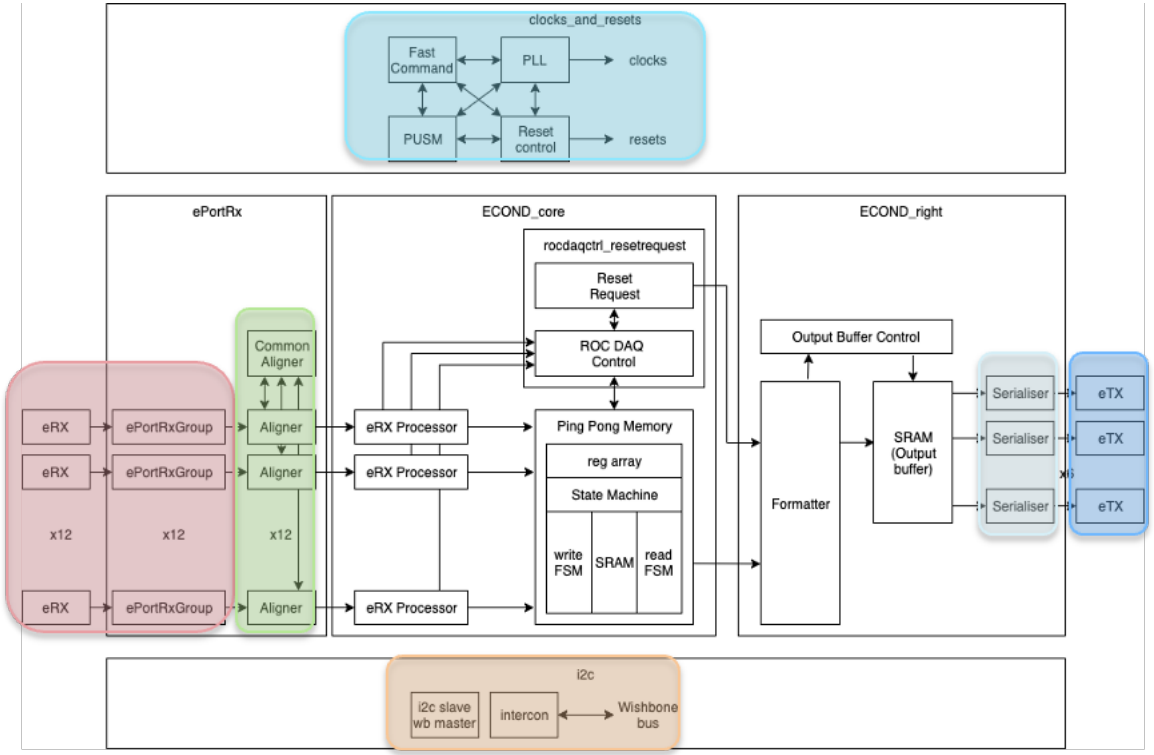
ECON-T and ECON-D

ECON-T and ECON-D

ECON-T



ECON-D



[1] DOI: 10.1088/1748-0221/19/03/C03050

Requirements

With effect on verification

- Multiple **components re-used** between ECON-D and ECON-T
- **Module assembly:**
  - Fast command, fast clock shared with HGCROC
  - Slow control (I2C) shared with HGCROC
  - Multiple detector modules types with different number of HGCROC per ECON
- **ECON-T:** Latency ≤ 400 ns
- **SEE tolerance:** $f_{(hadron,E\geq 20MeV)} = 1 \cdot 10^{6} cm^{-2} s^{-1}$ $\Phi_{(hadron,E\geq 20MeV)} = 1 \cdot 10^{14} cm^{-2}$
- **Power consumption** ≤ 2.5 mW/channel

Verification strategy

Verification strategy

Multiple complementary approaches to target different verification goals

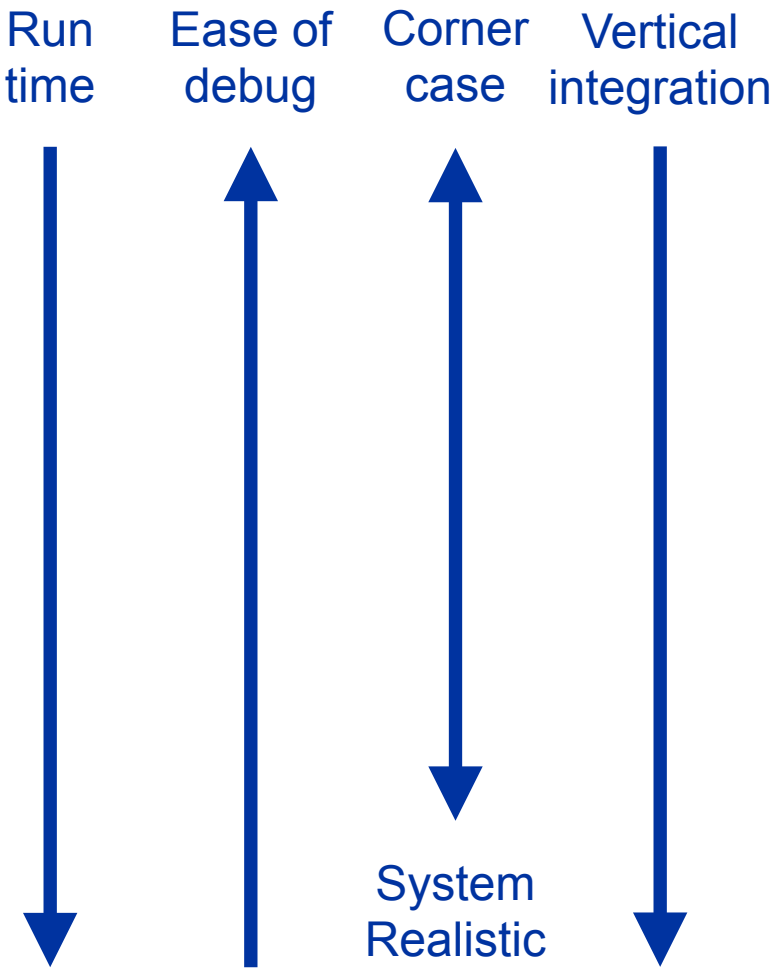
- **Functional verification based on Universal Verification Methodology (UVM)**
  - Industry standard for functional verification
  - Implements constrained-random verification
  - Measurable verification progress through coverage
  - Allows re-use of verification components (UVCs)
  - Used together with a regression manager to aggregate results
- **Formal verification**
  - Proves hard-to-reach conditions
  - Allows running Single-Event Upsets (SEUs) injections [2]
  - Allows proving Triple Modular Redundancy (TMR) correctly implemented
- **Continuous Integration (CI)**
  - Prevent introduction to bug-prone code in codebase
  - Acts as simple gate-keeping
  - Checks that no regression are introduced in development

[2] DOI: 10.1088/1748-0221/18/02/C02023

UVM-based functional verification

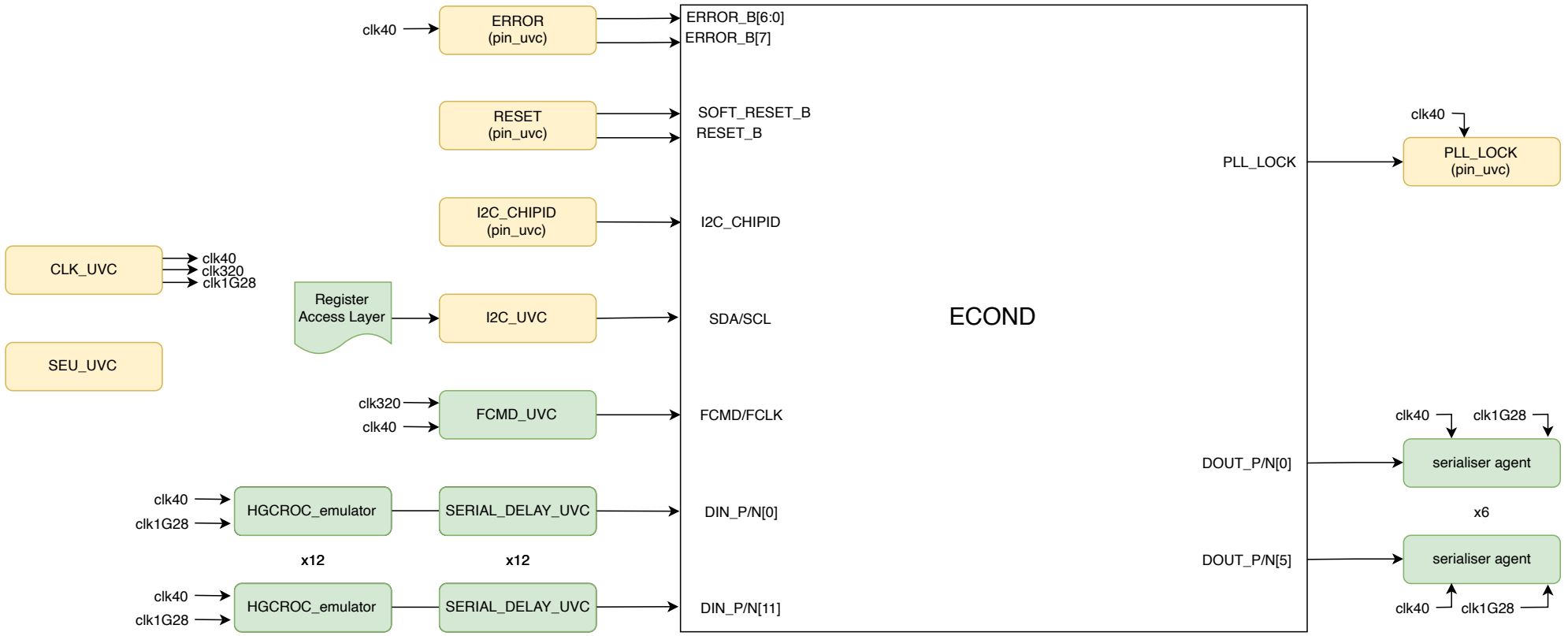
Multiple testbenches target at different verification goals

- **Block-level testbench for hard-to-cover blocks**
  - Allows verifying special conditions
- **ECON-D/ECON-T stand-alone testbench**
  - General testbenches
  - Achieve maximum coverage for Design Under Test (DUT)
  - Two separate data sources
    - HGCROC emulator (UVC)
    - HGCROC physics data (in CSV files)
- **System-level test bench (HGCROCs + ECON)**
  - Co-verify HGCROC trigger/data path with the corresponding ECON
  - Verifies specifications of HGCROC and ECON are compatible
  - Allows identifying bugs at system level



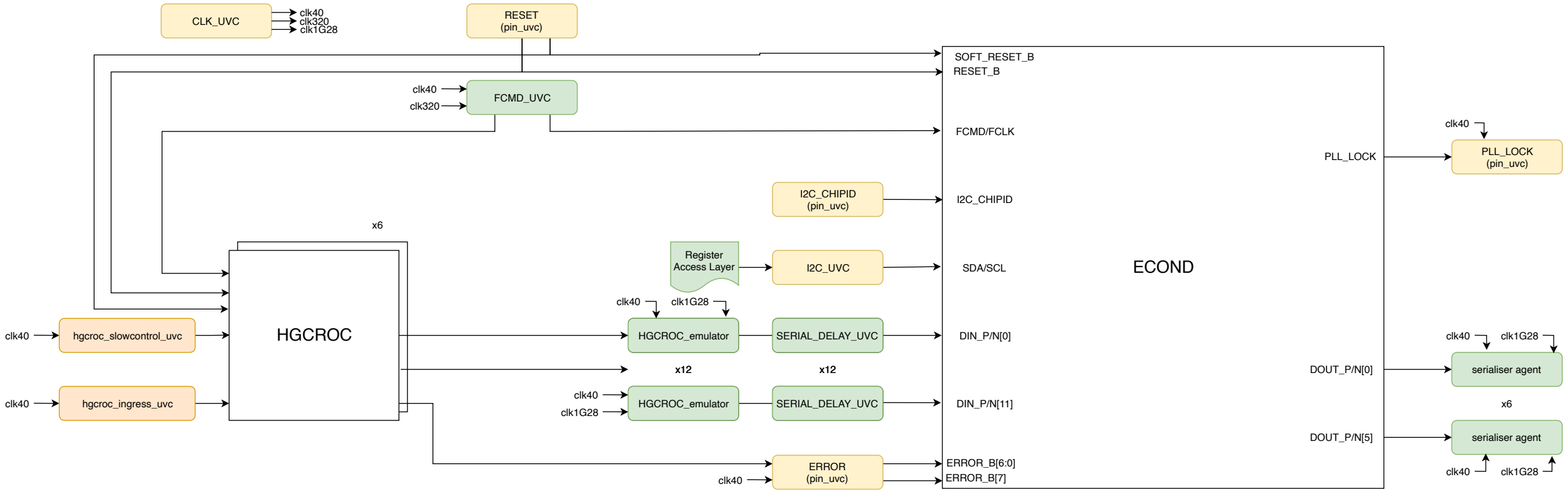
ECON-D

ASIC Testbench



ECON-D

System-level Testbench



Verified design hierarchies

- **HGCROC:**
  - Analog-on-top design
  - Partial TMR-RTL

	RTL	TMR-RTL	TMR-RTL w/ SEU	GL + SDF	GL + SDF w/ SEU	GL+SDF w/ SET
HGCROC		✓				
ECON-D right	✓	✓				
ECON-D	✓	✓	✓	✓	✓	✓
ECON-T	✓	✓	✓	✓	✓	✓

- **ECON-D and ECON-T:**
  - RTL
  - TMR-RTL
  - Gate-level (GL) netlist + SDF (min/typ/max)

- Fault injections run on [3]
  - **Single Event Upsets (SEUs):** TMR-RTL and GL+SDF (all corners)
  - **Single Event Transient (SETs):** GL+SDF (all corners)

- Verification framework designed to be able to work “seamlessly” w/ different hierarchies
  - DUT instantiation depends on `ifdef`
  - Verification components designed with re-use in mind

[3] DOI: 10.1088/1748-0221/18/01/C01038

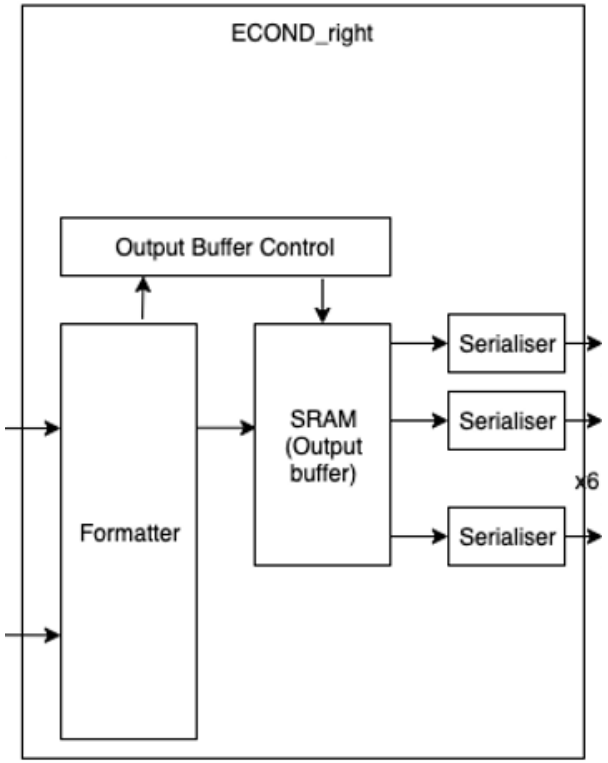
ECON-D right

Case study

ECON-D right testbench

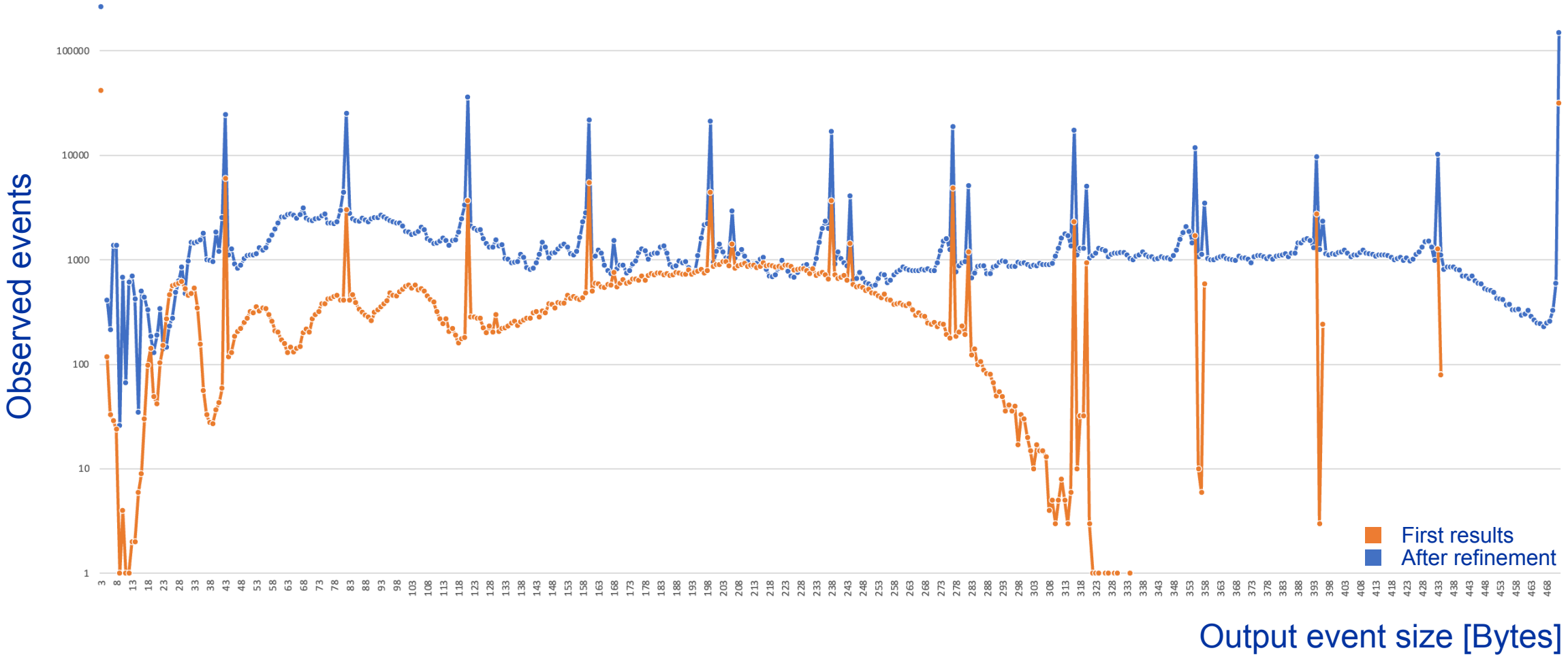
This module-based testbench was developed to help the development of the RTL

- **Design w/ very strict requirements**
  - High number of input signals
  - Real time processing w/ interrupts (request to veto packet)
  - No back pressure
  - Process packets both in- and out-of-order
  - Requires to run many (x1000) tests reach coverage goals
- **Module receives data from a non-triplicated SRAM (w/ ECC)**
  - Module-based testbench allows verifying ECC correction at early development stages



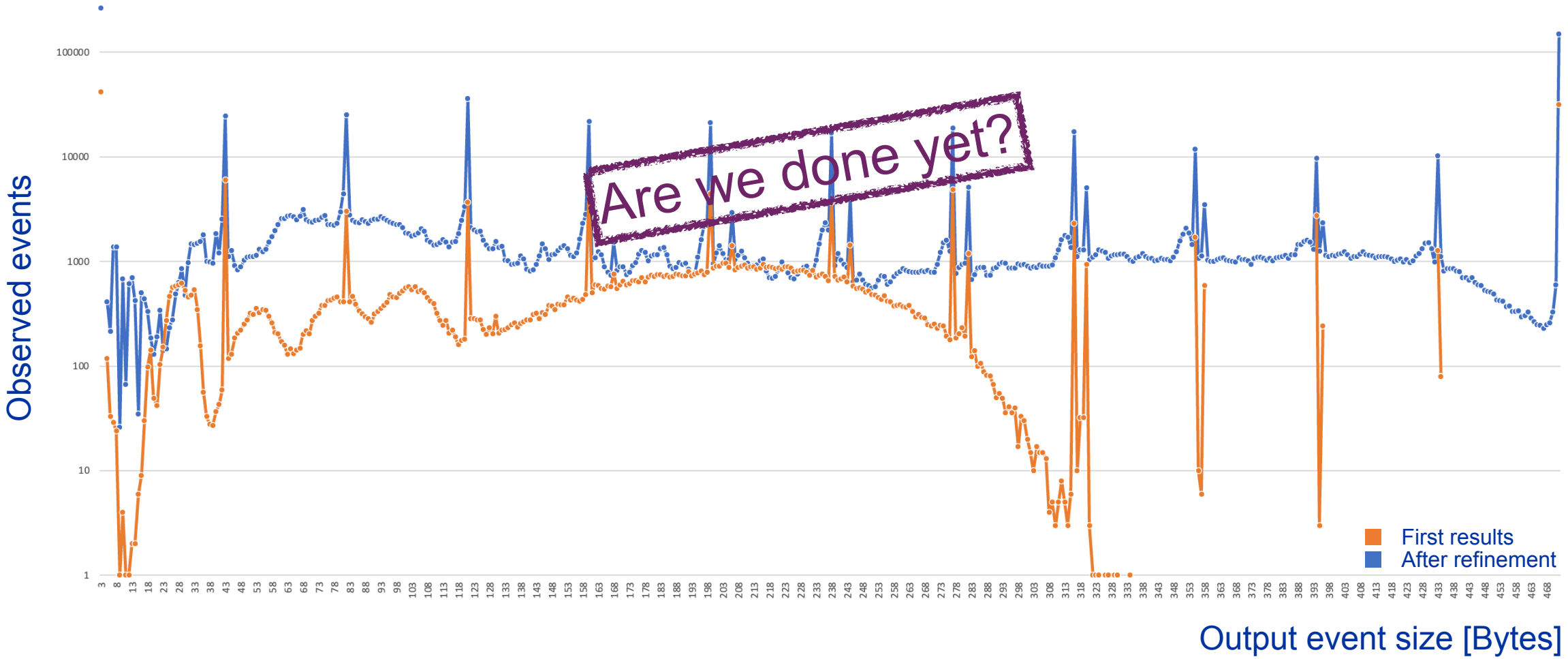
ECON-D right testbench

How coverage helps finding bugs



ECON-D right testbench

How coverage helps finding bugs



ECON-D right testbench

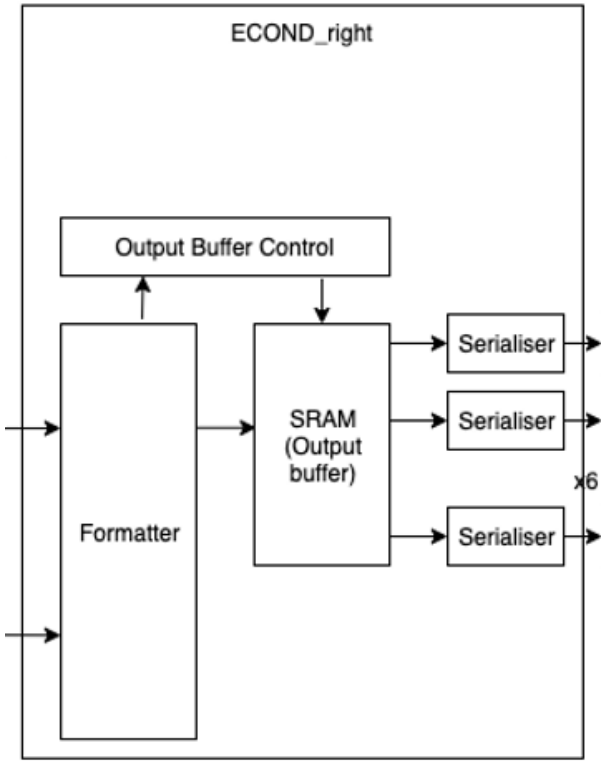
How coverage helps finding bugs

- ECON-D was submitted in two versions
  - Prototype: ECON-D-P1 → MPW: March 2023
  - Production: ECON-D-P2 → ER: December 2023
- The ECON-D right testbench showed complete coverage at first submission

- **However**…

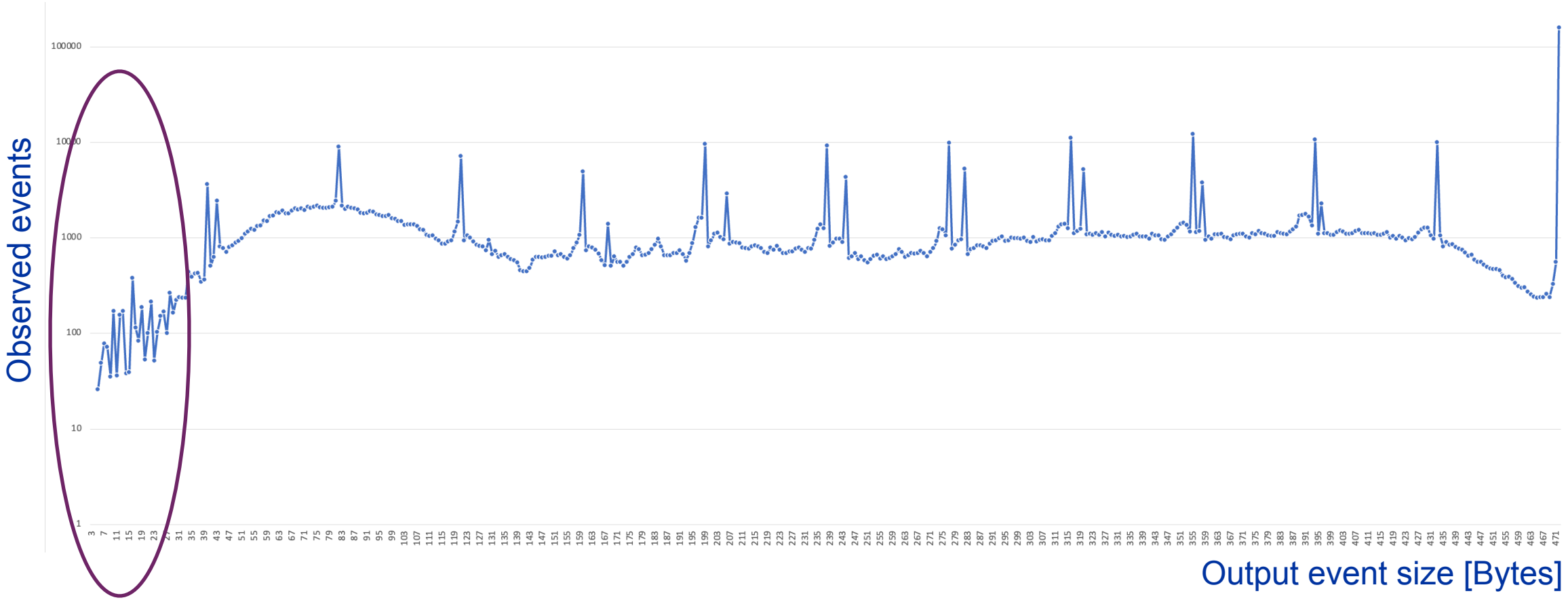
- Some (after tape out) tests showed unexpected failures
  - Packet w/ size 15 Bytes
  - Two input channel (35/36) of a specific HGCROC (on input link 11) present in output data

- Looking back at **coverage w/ inputs from designer**…



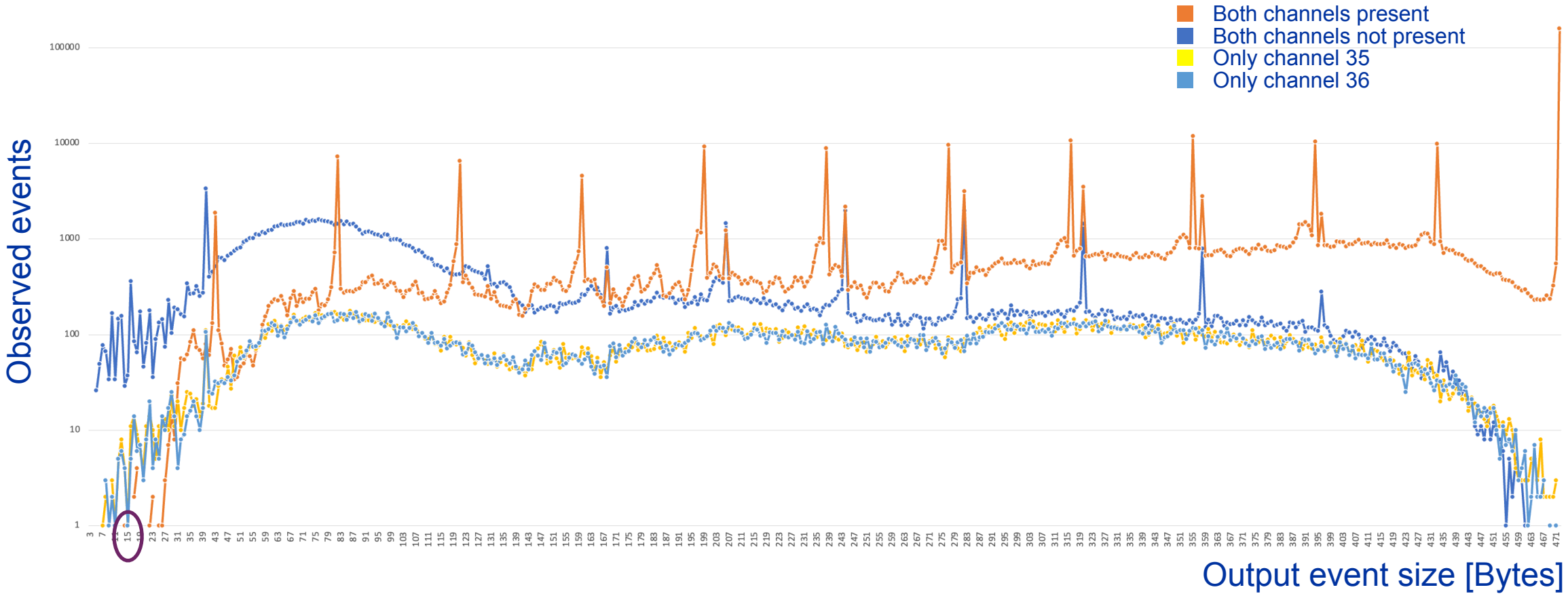
ECON-D right testbench

How coverage helps finding bugs



ECON-D right testbench

How coverage helps finding bugs



Results

Packaged ECONs

ECON-T

ECON-D

Typical run times to achieve coverage target

64 parallel tests running

- **ECON-D right side:**
  - ~4000 tests
  - ~13h machine time
- **ECON-D RTL:**
  - ~2000 tests
  - ~20h machine time
- **ECON-D TMR w/ SEU:**
  - 1500 tests
  - ~35h machine time
- **ECON-D SDF+GL (single delay corner)**
  - ~1200 tests
  - ~40h machine time
- **ECON-D SDF+GL w\ SET (single delay corner)**
  - ~1000 tests
  - ~32h machine time
- **ECON-T RTL:**
  - ~2500 tests
  - ~15h machine time
- **ECON-T TMR w/ SEU:**
  - 1500 tests
  - ~25h machine time
- **ECON-T SDF+GL w\ SEU (single delay corner)**
  - ~1200 tests
  - ~42h machine time
- **ECON-T SDF+GL w\ SET (single delay corner)**
  - ~1000 tests
  - ~35h machine time

Conclusions

- Design of functional verification framework for ECON-D and ECON-T was described
  - **Constrained-random verification** based on UVM testbenches
  - Leverage **re-use of verification components** w/ vertical integration
  - **Coverage** had a key role in verification signoff

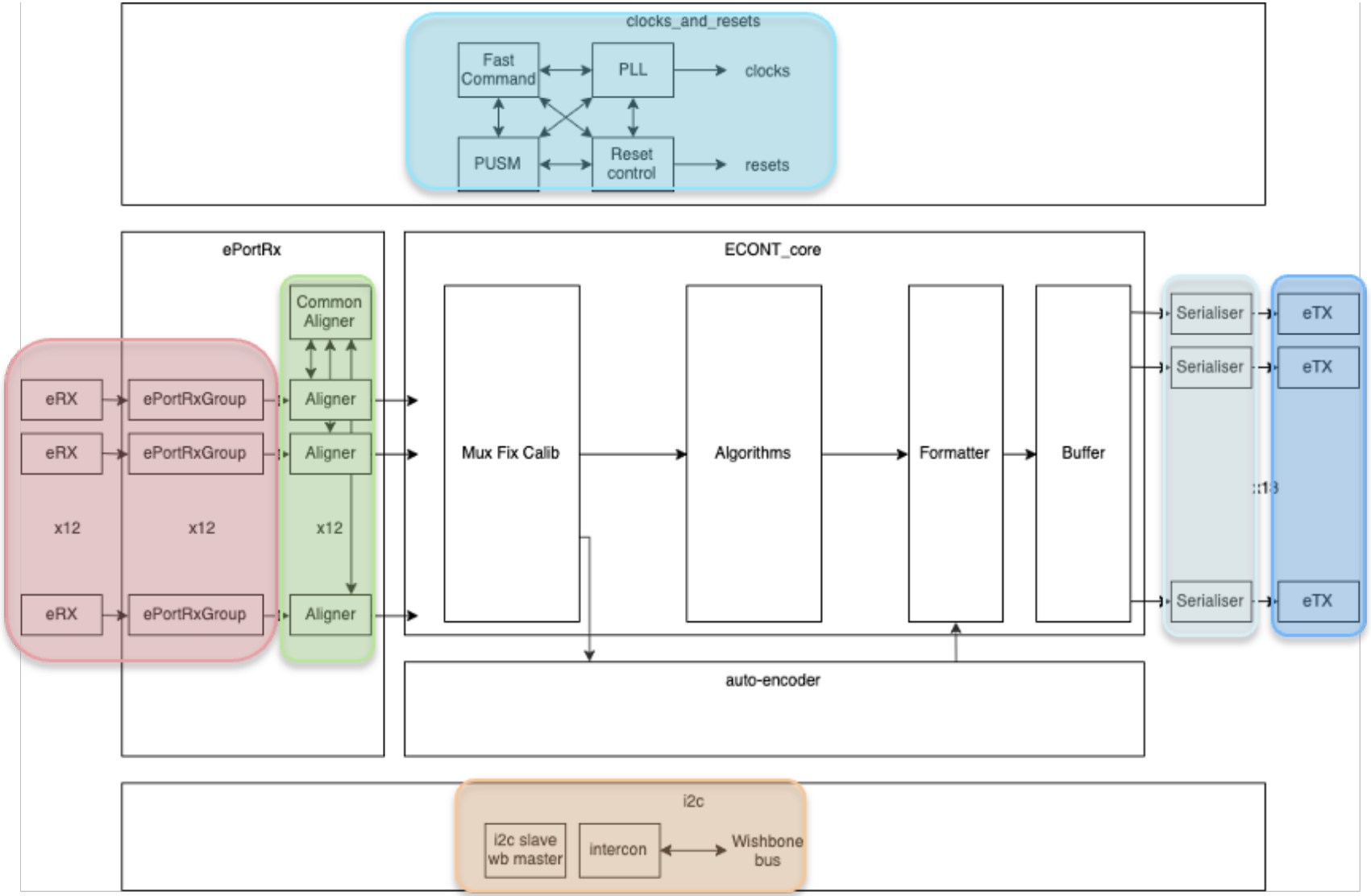
- **To date, ECONs** testing is showing **no functional bugs**

Backup slides

ECON-T

Backup

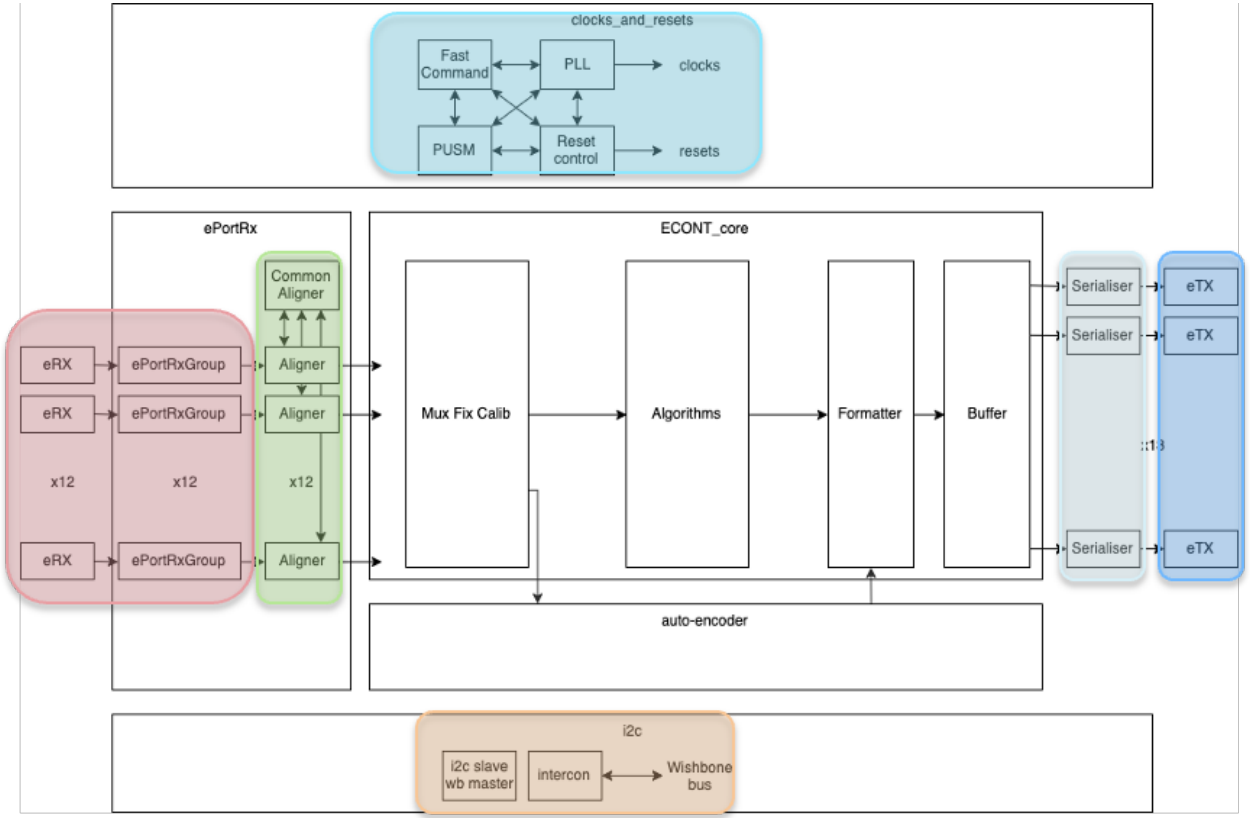
ECON-T


clocks_and_resets
Fast Command
PLL
clocks
PUSM
Reset control
resets
ePortRx
ECONT_core
Common Aligner
eRX
ePortRxGroup
Aligner
eRX
ePortRxGroup
Aligner
x12
x12
x12
eRX
ePortRxGroup
Aligner
Mux Fix Calib
Algorithms
Formatter
Buffer
Serialiser
eTX
Serialiser
eTX
::13
Serialiser
eTX
auto-encoder
i2c
i2c slave wb master
intercon
Wishbone bus

ECON-T

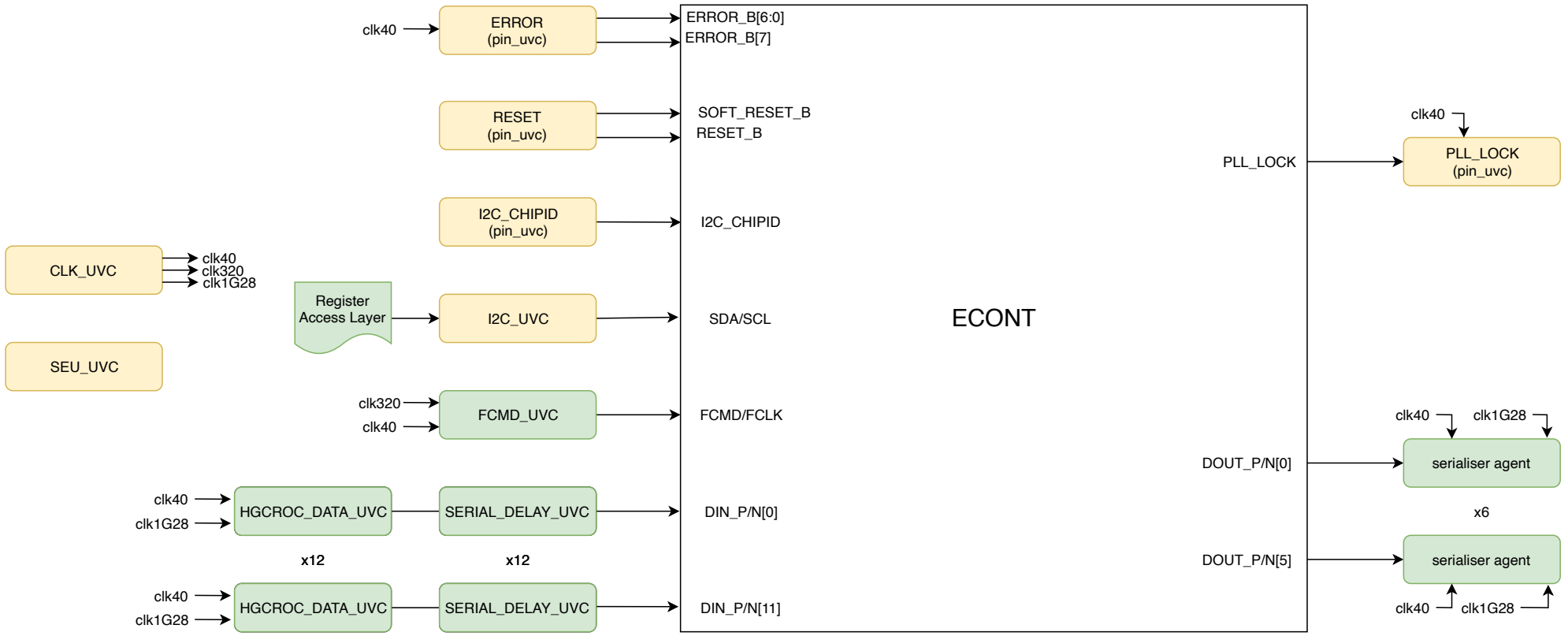
Auto encoder

- Neural-network based encoder for the trigger messages received by HGCROC
- Generated w/ High-Level Syntesis (HLS) of C++ code
  - **No cycle accurate specification**
  - No human-readable RTL
- Reference model
  - Calculate expected output by running it the C++ code with the provided inputs
  - Calculation done via Direct Programming Interface (DPI)



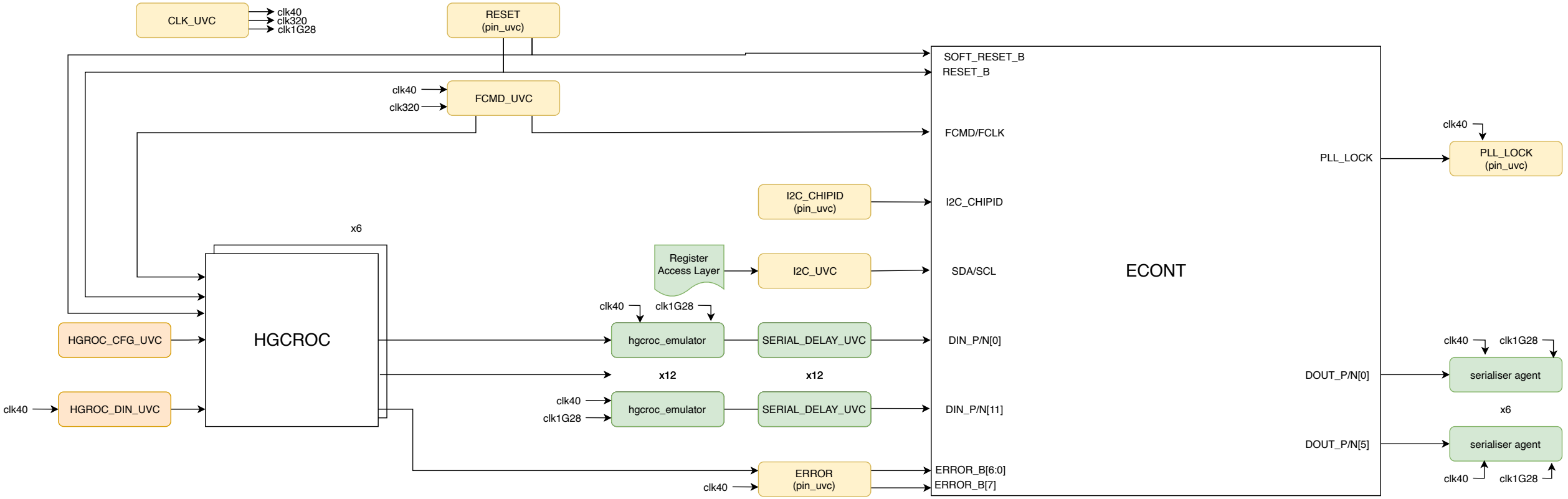
ECON-T

ASIC Testbench



ECON-T

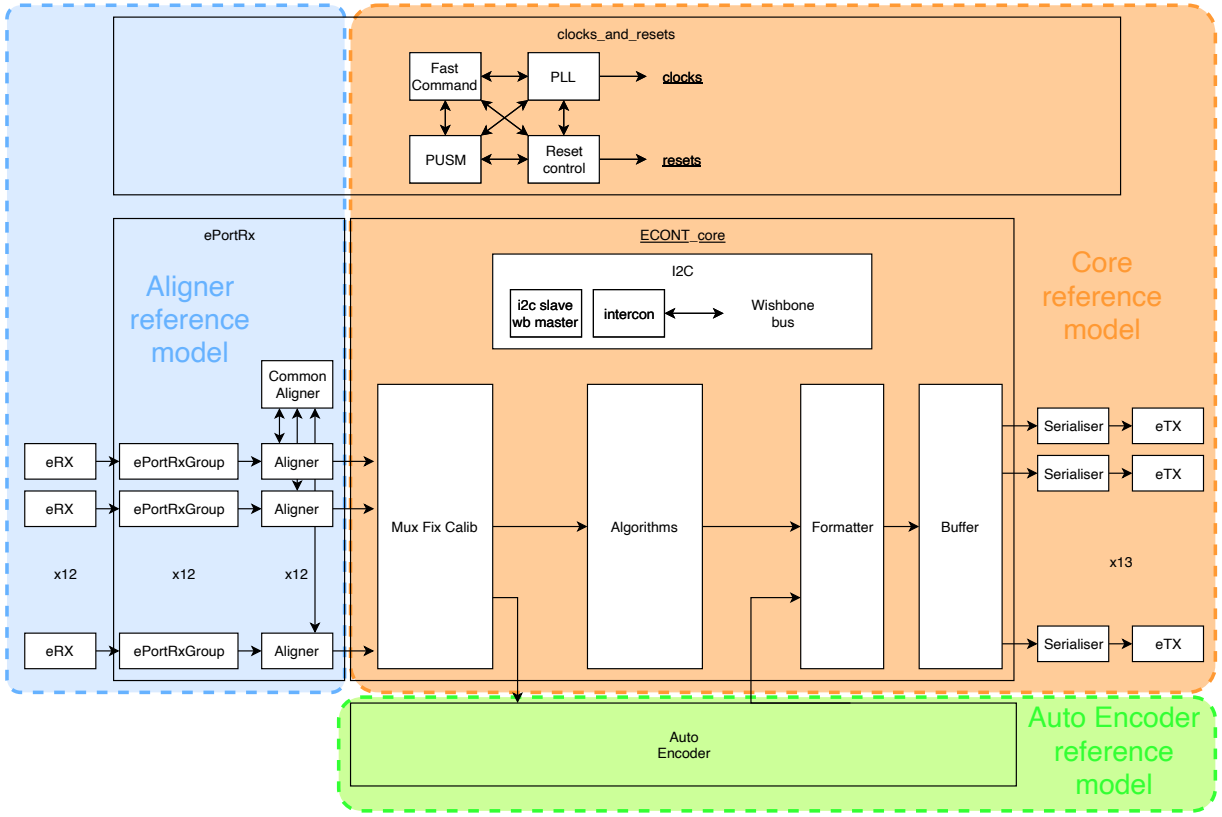
System-level Testbench



ECON-T

Reference model

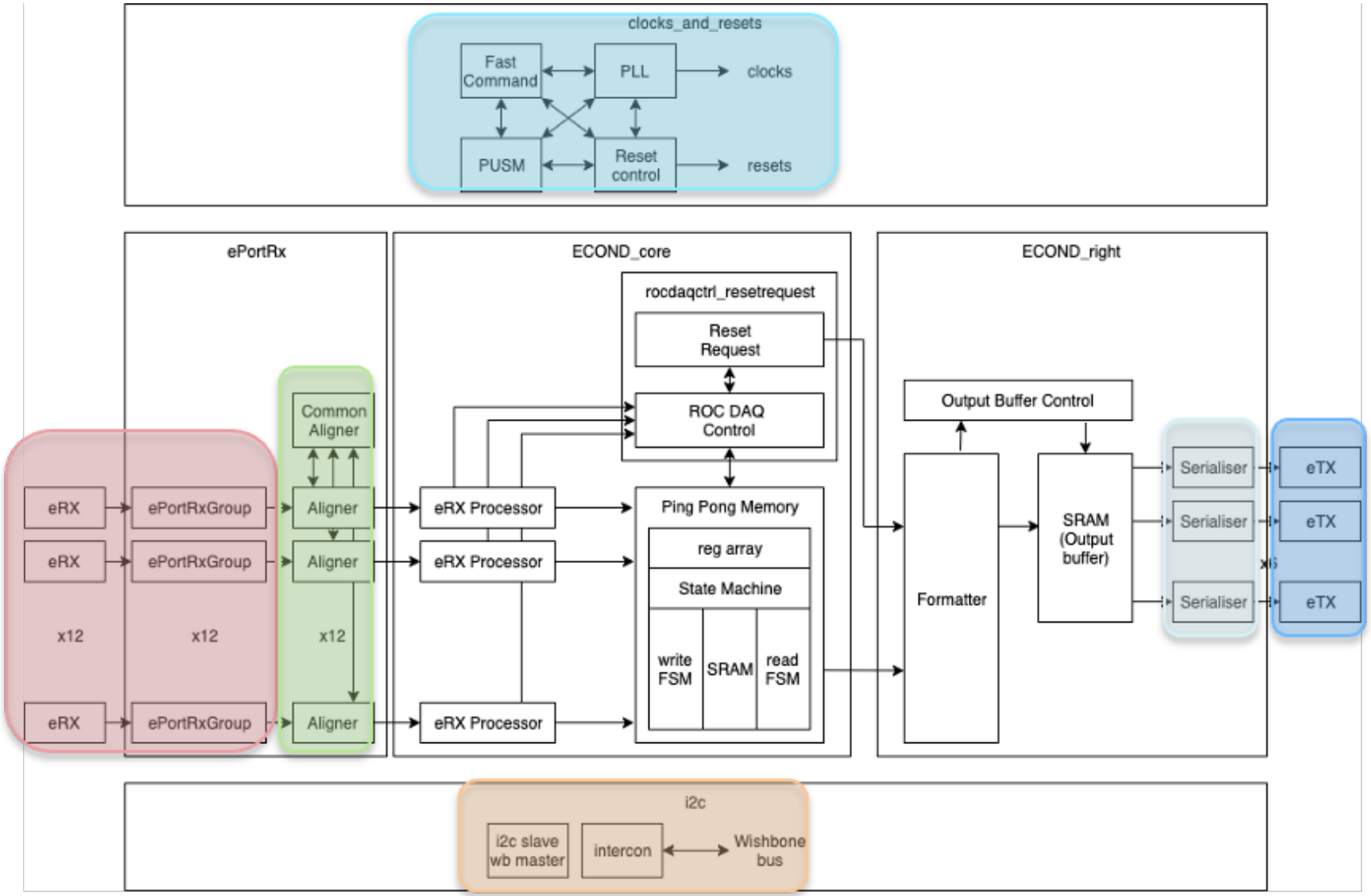
- Reference model built targeting
  - Re-use (aligner)
  - Parallel development
    - Core developed at CERN
      - UVM behavioural model
    - AE developed at FNAL
      - Mixed approach
      - C++ behavioural model
      - UVM timing behaviour
      - UVM configuration & coverage



ECON-D

Backup

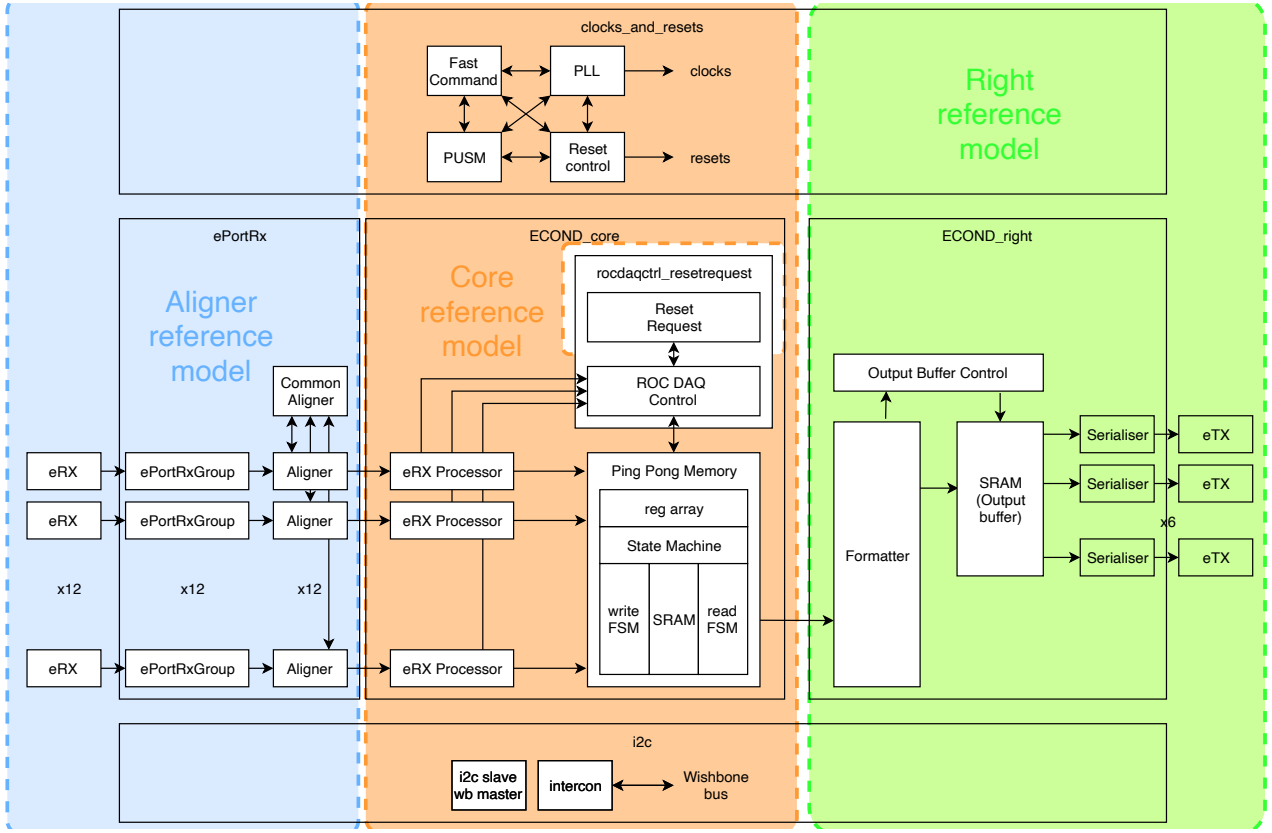
ECON-D


clocks_and_resets
Fast Command
PLL
clocks
PUSM
Reset control
resets
ePortRx
ECOND_core
ECOND_right
rocdaqctrl_resetrequest
Reset Request
ROC DAQ Control
Common Aligner
eRX
ePortRxGroup
Aligner
eRX Processor
Ping Pong Memory
reg array
State Machine
write FSM
SRAM
read FSM
x12
Output Buffer Control
Formatter
SRAM (Output buffer)
Serialiser
eTX
i2c
i2c slave wb master
intercon
Wishbone bus

ECON-D

Reference model

- Reference model built targeting
  - Re-use
    - Aligner re-used in ECON-T
    - ECON-D right (block-level testbench)
  - Parallel development
    - Aligner developed at FNAL/CERN
    - Core developed at FNAL
    - Reset request developed at FNAL
    - ECON-D right developed at CERN



ECON-D output packet

Header

Payload

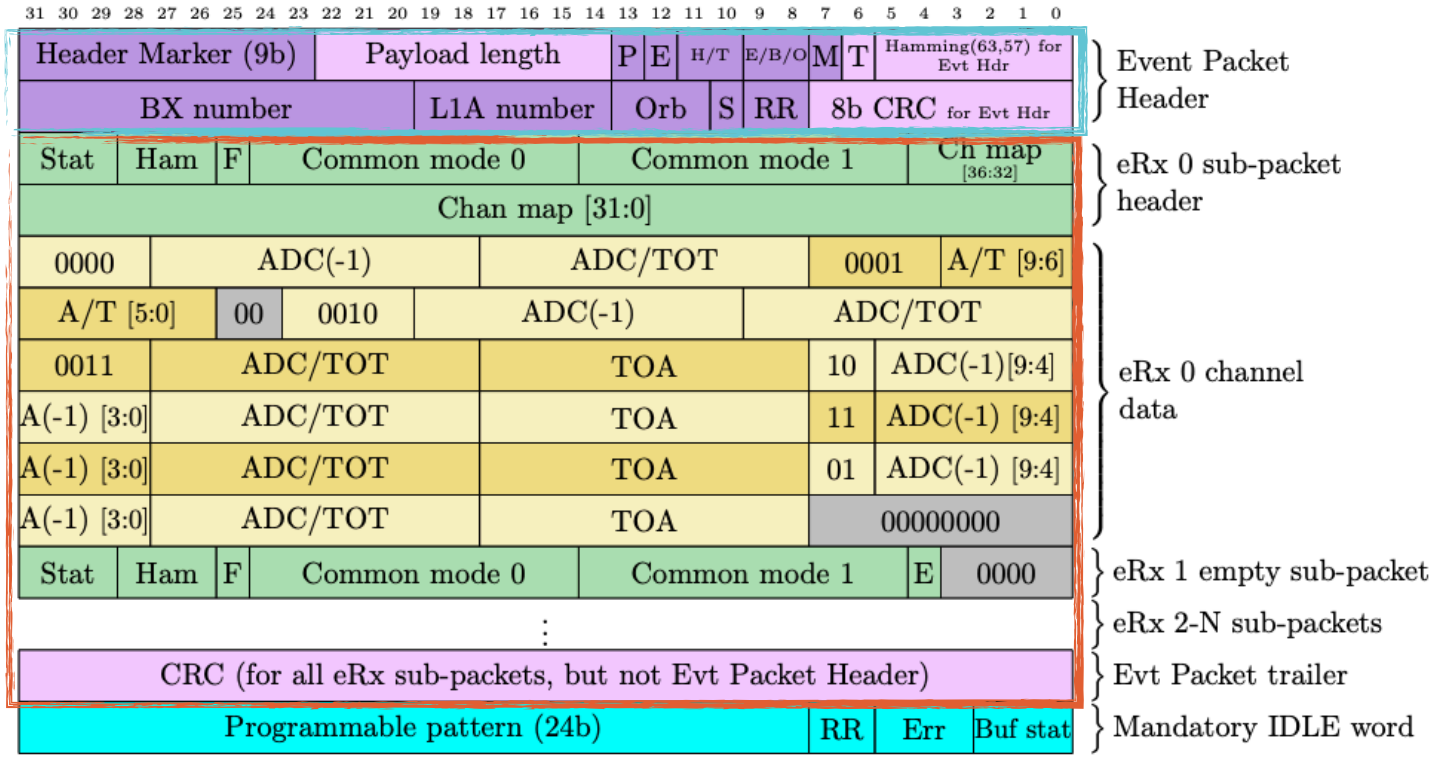
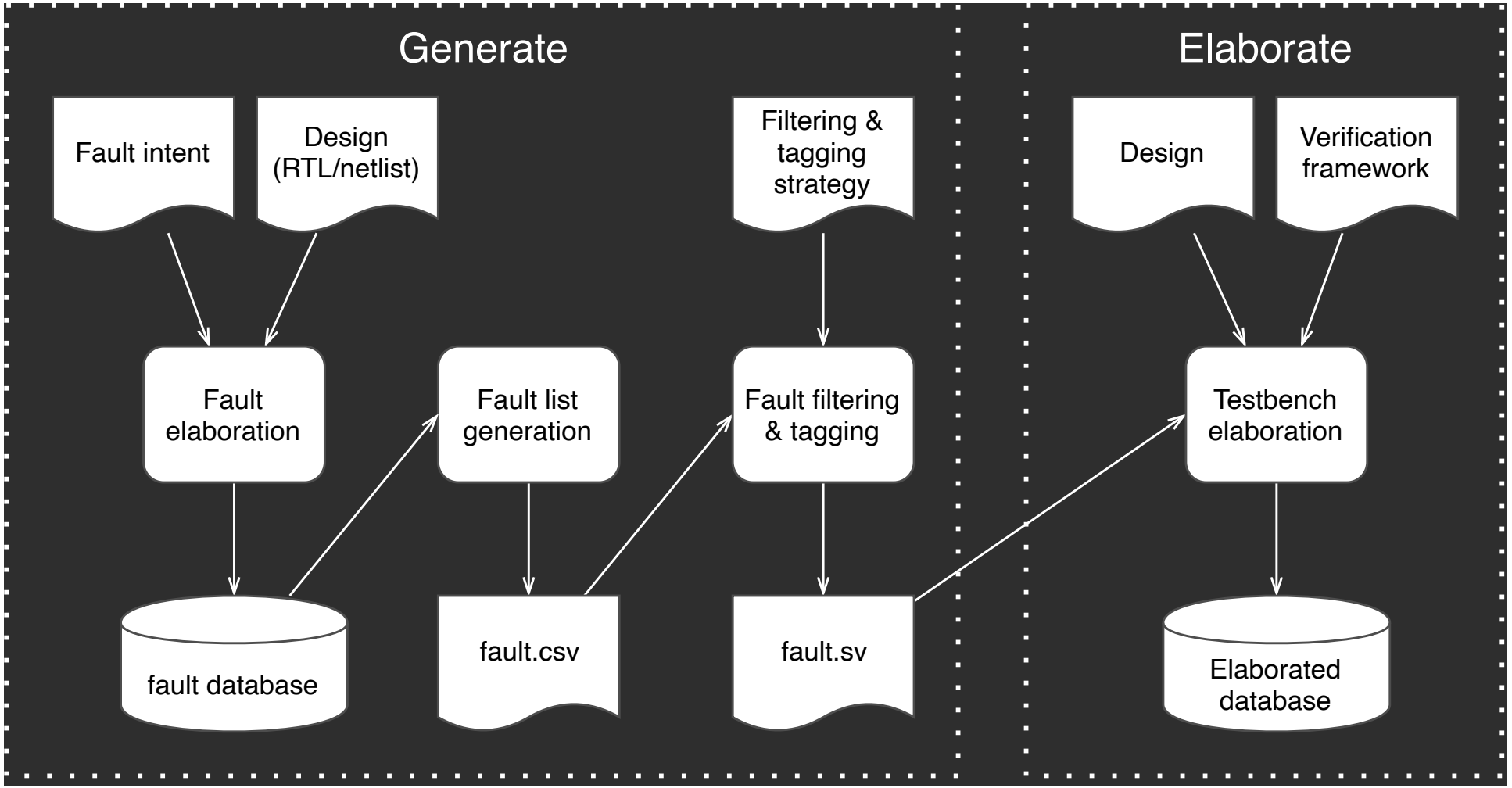


Figure 33 : Standard ECON-D Event Packet data format for N active eRx.

Fault injections in simulation

Framework


Generate
Elaborate
Fault intent
Design (RTL/netlist)
Filtering & tagging strategy
Design
Verification framework
Fault elaboration
Fault list generation
Fault filtering & tagging
Testbench elaboration
fault database
fault.csv
fault.sv
Elaborated database

Reference vs fault simulations