RELIABILITY TESTS OF THE LHC BEAM LOSS MONITORING FPGA FIRMWARE

Csaba F. Hajdu

09/04/2010

Outline

- Introduction
  - The LHC Beam Loss Monitoring system
  - The Threshold Comparator firmware
  - Verification: Motivation
- Verification methodology
  - Simulation
  - Hardware-based testing
  - Software-based testing
- Procedure for new release
- Conclusion

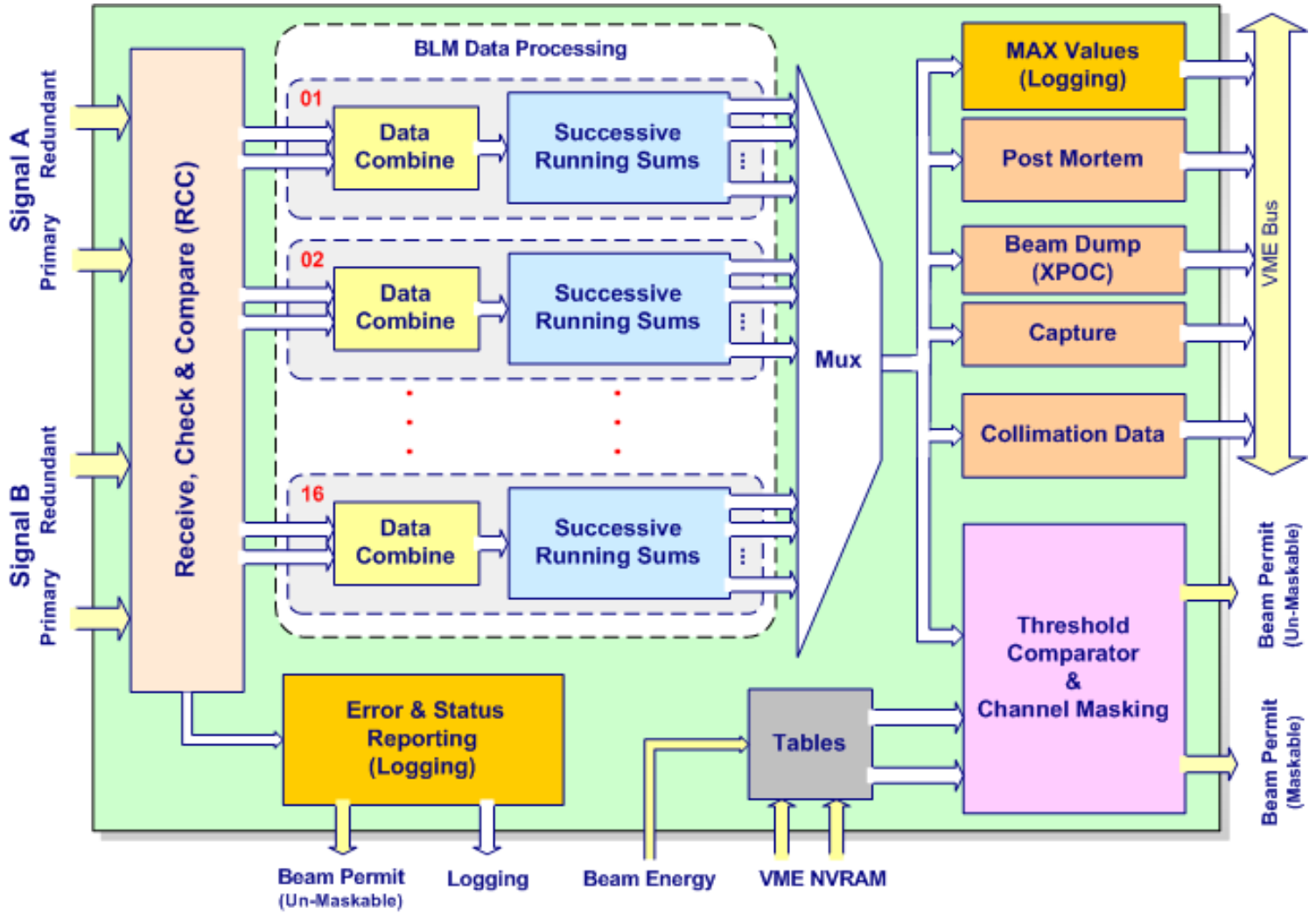
1/1. Introduction

The LHC Beam Loss Monitoring system

- The LHC Beam Loss Monitoring system
  - One of the most critical protection systems
    - Protects magnets from quenches, collider from damage
    - Provides data for diagnostics and machine tuning
  - Nearly 4'000 monitors
    - Ionization Chambers (IC), Secondary Emission Monitors (SEM)
  - Data acquisition: Current to Frequency Converter (CFC)
  - Data processing: Threshold Comparator (TC)
    - FPGA-based
    - Calculation of integrals over different time windows
    - Beam abort trigger when necessary

1/2. Introduction

The Threshold Comparator firmware



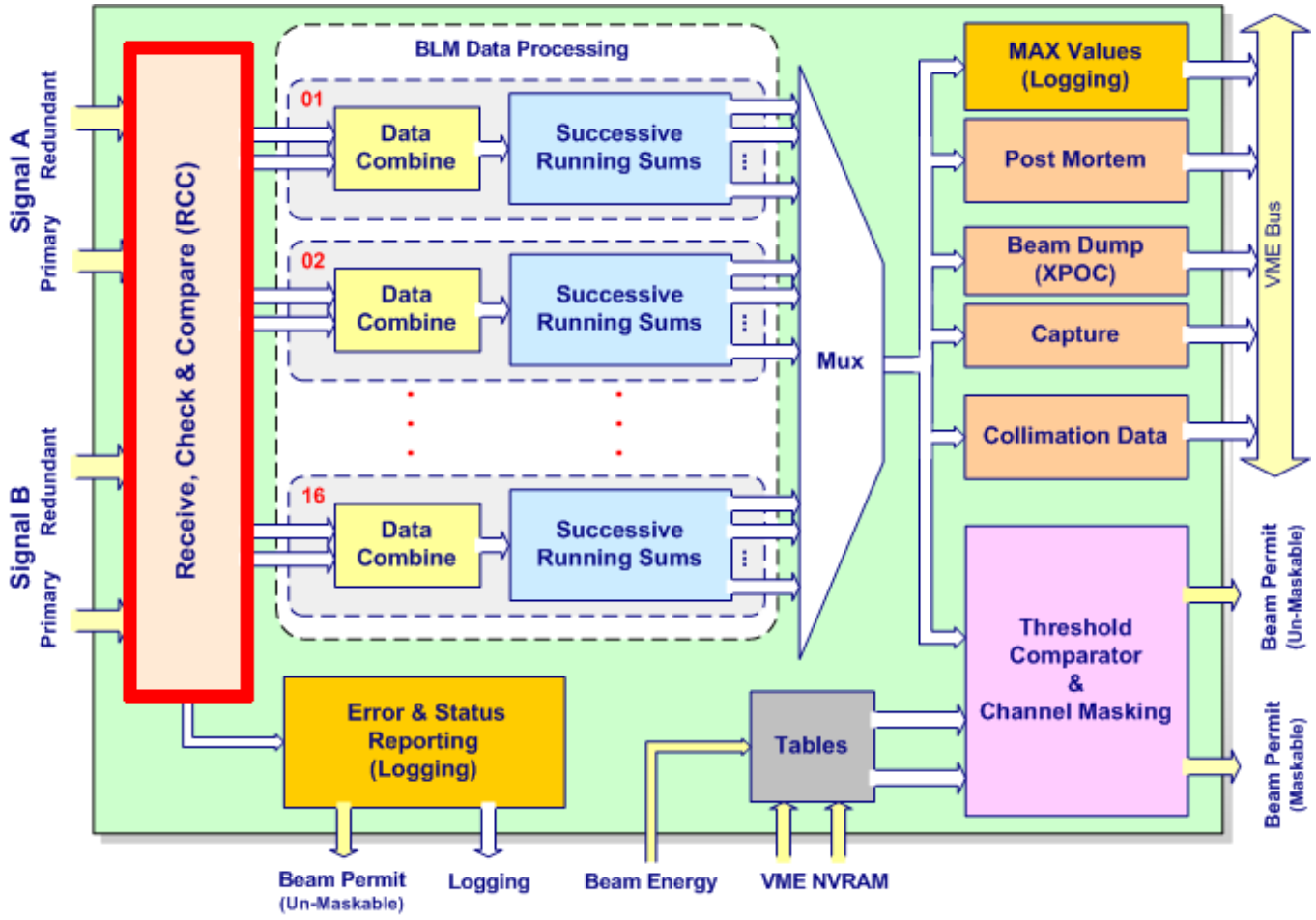
1/3. Introduction

Verification: Motivation

- Verification of the Threshold Comparator firmware
  - Verification: about 70% of the design cycle (Doulos)
- Sequential digital design of great complexity
  - Exhaustive verification impractical
  - Comprehensive verification environment
    - Different approaches of verification targeting different aspects of design
      - Functional simulation
      - Hardware-based approach
      - Software-based approach

2/1. Simulation

Scope



2/2. Simulation

The RCC block

- The RCC block
  - Receives the redundant input signals from TLKs
  - Checks packets for errors
    - CRC, Card ID, Frame ID, link unavailable
    - Can issue beam abort triggers as required
  - Decides which packet to use for further processing

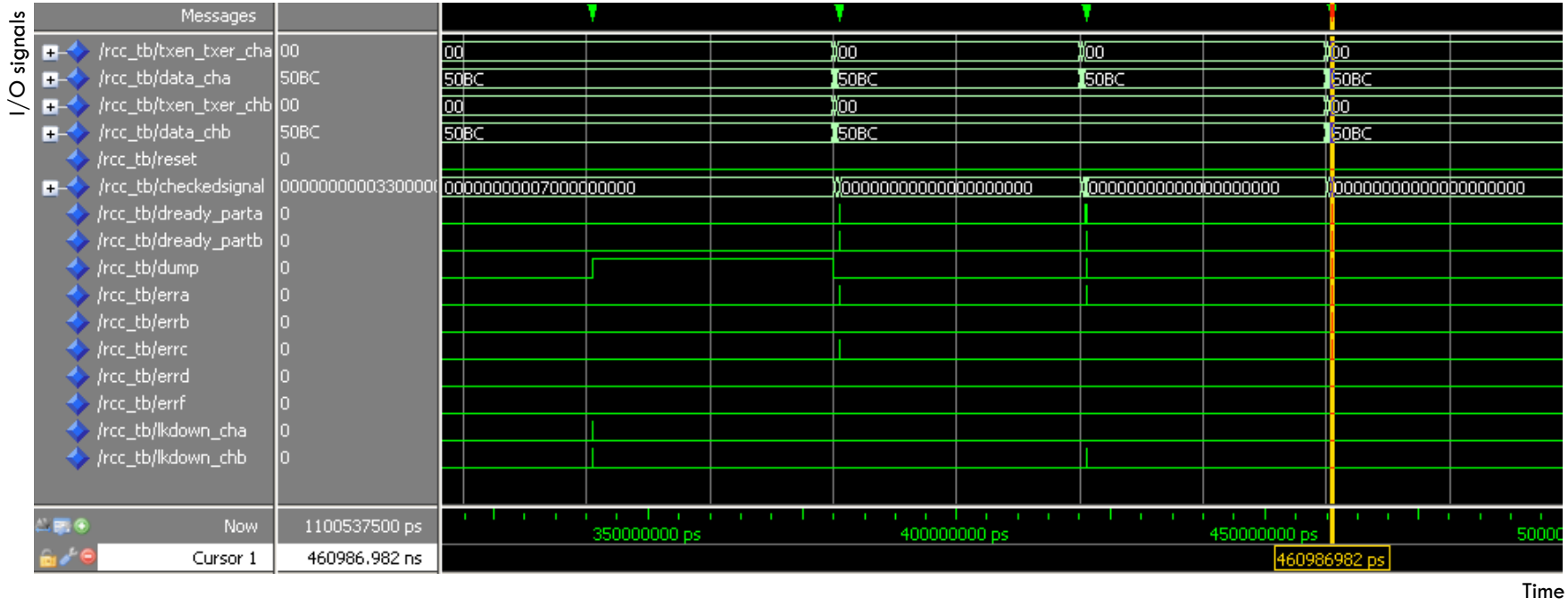
2/3. Simulation

Description

- Functional simulation
  - Verify compliance to functional requirements of spec
- Testbench (ModelSim)
  - “Black box” design methodology
    - Based only on specification
  - Stimulus read from file
  - Automatic checking of outputs
    - Comparison of versions
      - Regression testing
      - Detection of new bugs

2/4. Simulation

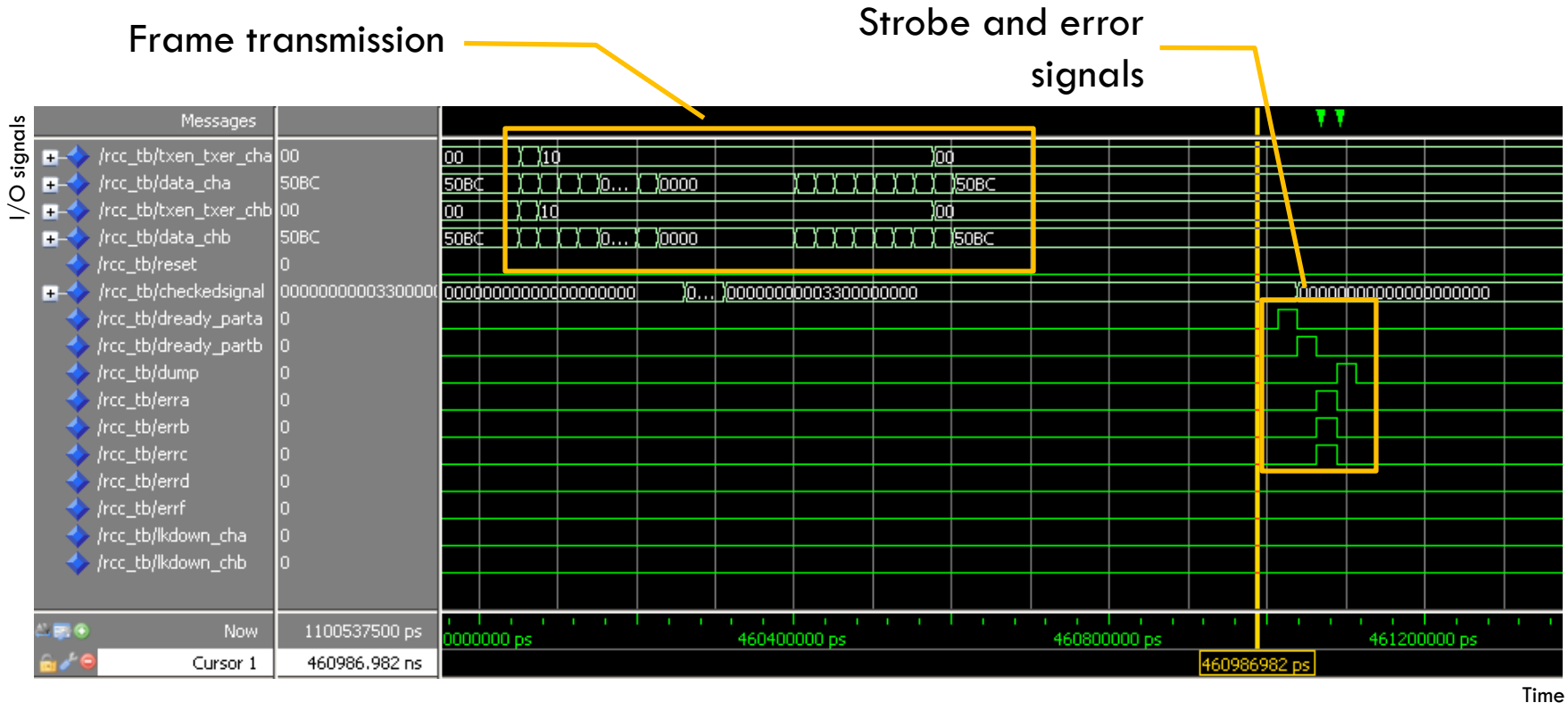
Sample output waveforms



Output waveform for optical inspection

2/5. Simulation

Sample output waveforms



2/6. Simulation

Automatic output checker – Description

- Automatic output checker
  - Checks the output of the block vs. the expected output based on stimulus from file
  - Complements visual inspection

```
507
508         -- Both links are down
509         IF (TxData_ChA.LKup = '0') AND (TxData_ChB.LKup = '0') THEN
510           WAIT UNTIL (ERRA OR ERRB OR ERRC OR ERRD OR ERRF) = '1' FOR 1000 ns;
511           IF (ERRA OR ERRB OR ERRC OR ERRD OR ERRF) = '1' THEN
512             REPORT "Both links are down, error flag INCORRECTLY RAISED!" SEVERITY ERROR;
513             WAIT FOR 30 ns;     -- Wait one clock cycle and some more
514           END IF;
515
516           IF DUMP = '1' THEN
517             REPORT "Both links are down, dump correctly generated";
518           ELSE
519             REPORT "Both links are down, DUMP MISSING!";
520           END IF;
521         -- At least one of the links is up
522         ELSE
```

Code snippet of the automatic output checker

2/7. Simulation

Automatic output checker – Sample outputs

Transcript

```
# ** Note: Dump correctly generated by error flags
#    Time: 421067500 ps  Iteration: 0  Instance: /rcc_tb
# ** Note: ChA CRC error correctly reported
#    Time: 461042500 ps  Iteration: 0  Instance: /rcc_tb
# ** Note: ChB CRC error correctly reported
#    Time: 461042500 ps  Iteration: 0  Instance: /rcc_tb
# ** Note: CRC comparison error correctly reported
#    Time: 461042500 ps  Iteration: 0  Instance: /rcc_tb
# ** Note: Dump correctly generated by error flags
#    Time: 461067500 ps  Iteration: 0  Instance: /rcc_tb
# ** Note: ChB CRC error correctly reported
#    Time: 501042500 ps  Iteration: 0  Instance: /rcc_tb
# ** Note: CRC comparison error correctly reported
#    Time: 501042500 ps  Iteration: 0  Instance: /rcc_tb
# ** Note: ChA CRC error correctly reported
#    Time: 581042500 ps  Iteration: 0  Instance: /rcc_tb
# ** Note: CRC comparison error correctly reported
#    Time: 581042500 ps  Iteration: 0  Instance: /rcc_tb
# ** Note: ChA CRC error correctly reported
#    Time: 621042500 ps  Iteration: 0  Instance: /rcc_tb
# ** Note: CRC comparison error correctly reported
#    Time: 621042500 ps  Iteration: 0  Instance: /rcc_tb
# ** Note: ERRFpc correctly reported
#    Time: 701042500 ps  Iteration: 0  Instance: /rcc_tb
```

Error-free output

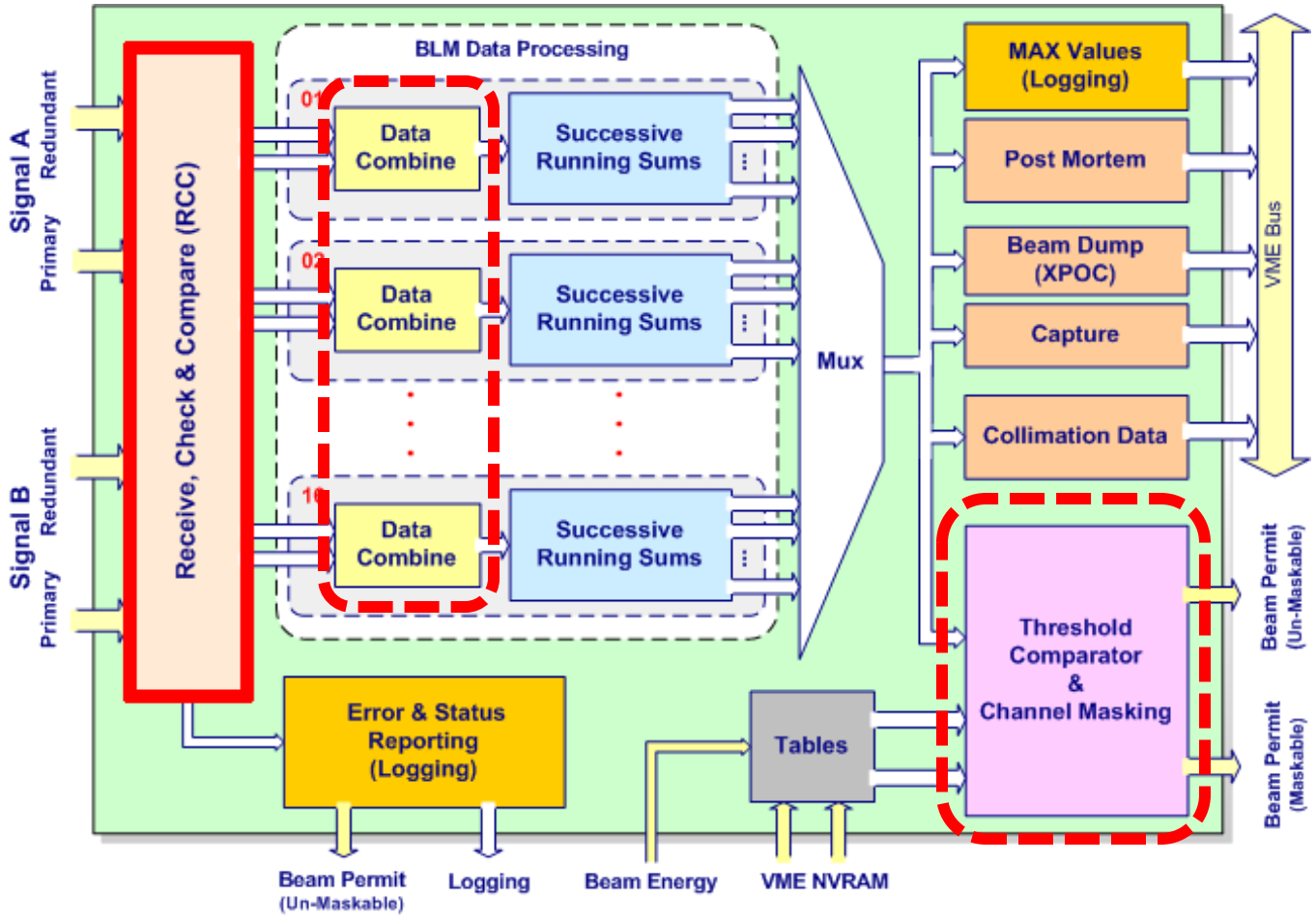
Transcript

```
# ** Error: CID comparison error INCORRECTLY REPORTED!
#    Time: 821067500 ps  Iteration: 0  Instance: /rcc_tb
# ** Error: ERRF INCORRECTLY REPORTED!
#    Time: 821067500 ps  Iteration: 0  Instance: /rcc_tb
# ** Error: CID comparison error INCORRECTLY REPORTED!
#    Time: 861067500 ps  Iteration: 0  Instance: /rcc_tb
# ** Error: ERRF INCORRECTLY REPORTED!
#    Time: 861067500 ps  Iteration: 0  Instance: /rcc_tb
# ** Error: CID comparison error INCORRECTLY REPORTED!
#    Time: 901067500 ps  Iteration: 0  Instance: /rcc_tb
# ** Error: ERRF INCORRECTLY REPORTED!
#    Time: 901067500 ps  Iteration: 0  Instance: /rcc_tb
# ** Error: Both links are down, error flag INCORRECTLY RAISED!
#    Time: 941042500 ps  Iteration: 0  Instance: /rcc_tb
# ** Note: Both links are down, dump correctly generated
#    Time: 941072500 ps  Iteration: 0  Instance: /rcc_tb
# ** Error: CID comparison error INCORRECTLY REPORTED!
#    Time: 981067500 ps  Iteration: 0  Instance: /rcc_tb
# ** Error: ERRF INCORRECTLY REPORTED!
#    Time: 981067500 ps  Iteration: 0  Instance: /rcc_tb
# ** Error: Both links are down, error flag INCORRECTLY RAISED!
#    Time: 1021042500 ps  Iteration: 0  Instance: /rcc_tb
# ** Note: Both links are down, dump correctly generated
#    Time: 1021072500 ps  Iteration: 0  Instance: /rcc_tb
```

Output with errors

3/1. Hardware-based check

Scope



3/2. Hardware-based check

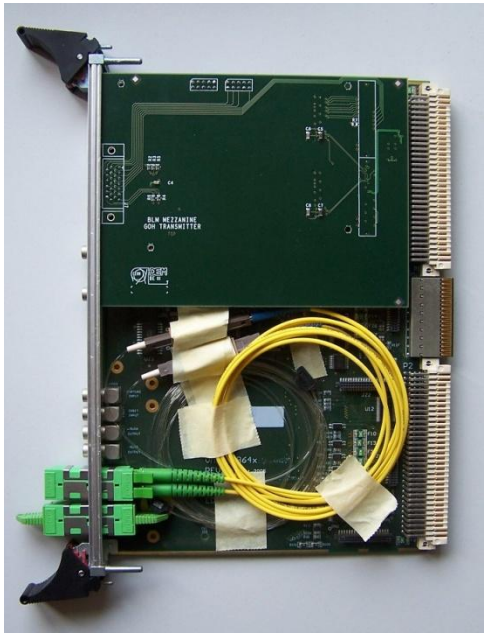
Motivation

- In situ test of the TC in VME crate
  - Emulation of output signals of CFC
- TC architecture
  - BI standard DAB64x card
  - Optical receiver mezzanine
    - 4 TLKs, signal from 2 CFCs
  - FPGA processing firmware

3/3. Hardware-based check

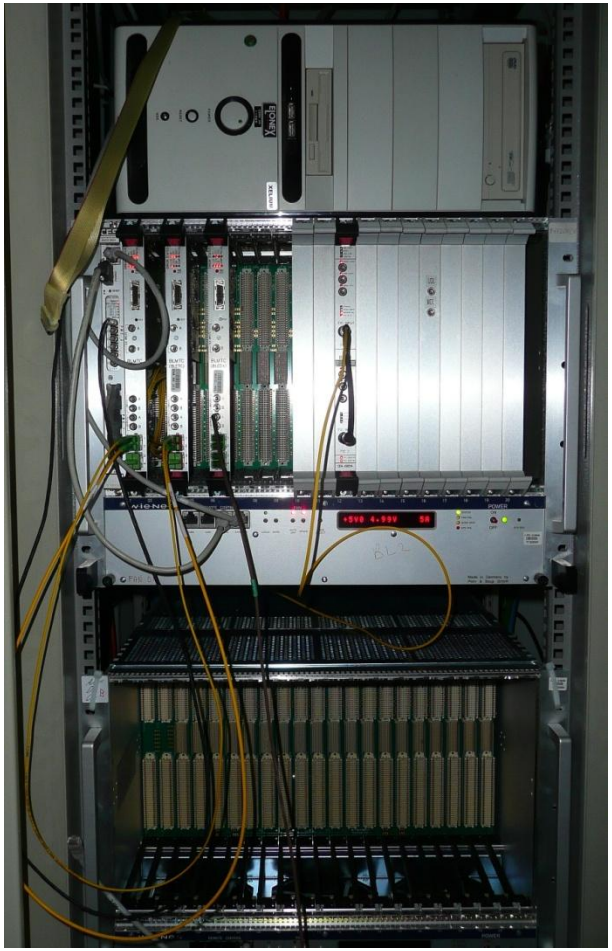
The “Vertical Slice” card

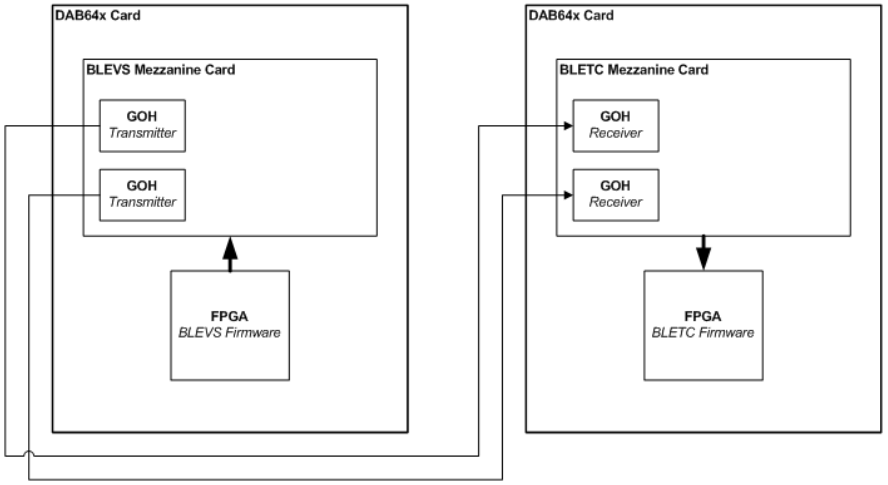
- New mezzanine for DAB64x card
  - Two standard Gigabit Optical Hybrid transmitters
    - Emulation of one CFC; more with optical splitters
- Custom FPGA firmware



3/4. Hardware-based check

Installation





3/5. Hardware-based check

Features

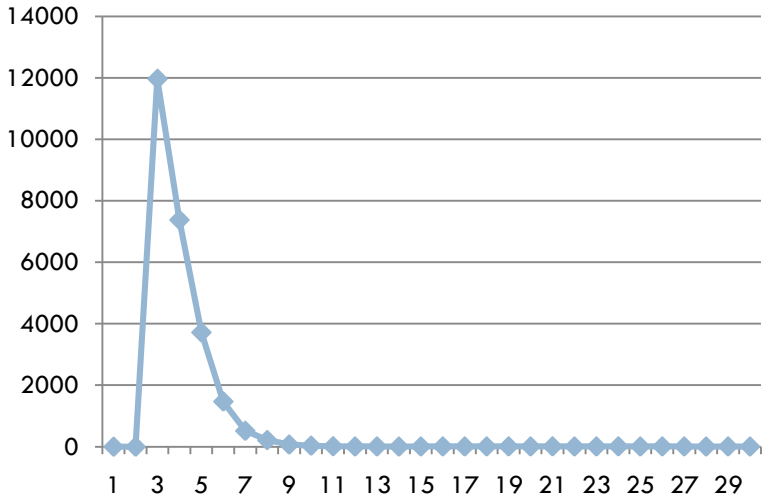
- Arbitrary Tx data
  - Comparison of different TC firmware versions
  - Playback of LHC capture data for analysis
- Tx errors
  - CRC, CID, FID
- Wrong configuration
- Errors in physical layer

- Manual testing procedure
  - Results read out in Expert application

3/6. Hardware-based check

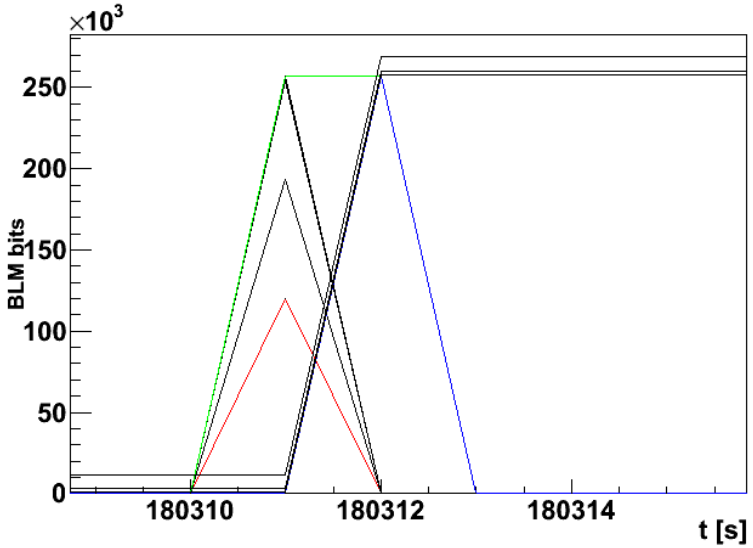
Features – Playback of capture data

INPUT



Samples from an IC in LHC tunnel: loss of a single bunch (40us)

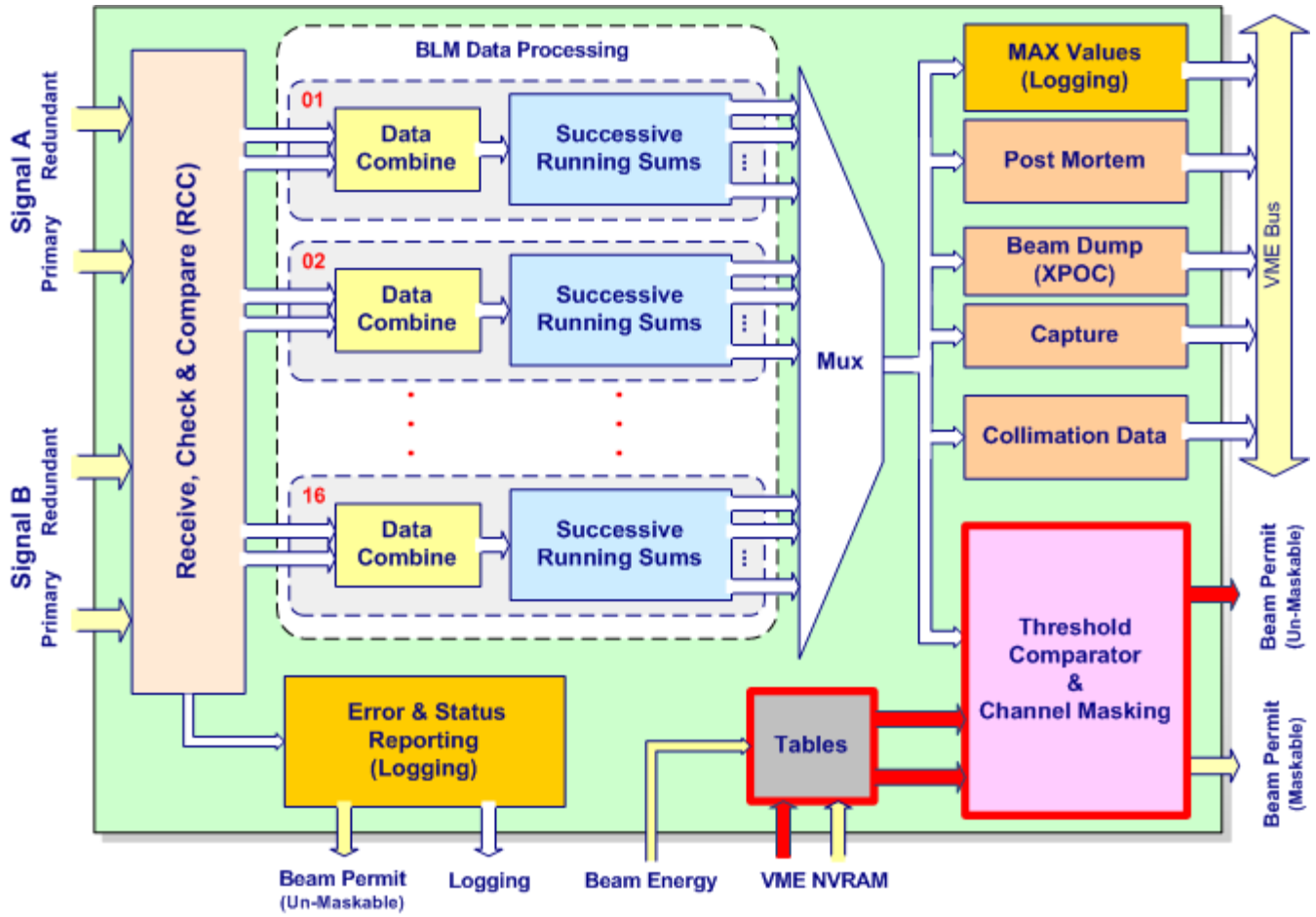
OUTPUT



12 Running Sums (integrals)

4/1. Software-based check

Scope



4/2. Software-based check

Description

- Exhaustive verification of the behavior of the Threshold Comparator block
  - Every threshold value has to trigger one by one
    - 16 cards/crate
    - 16 detectors/TC card
    - 12 integration windows/detector
    - 32 beam energy levels
      - 98'304 testcases/crate
- VME readout check
  - The same testcase repeated 500'000 times

4/3. Software-based check

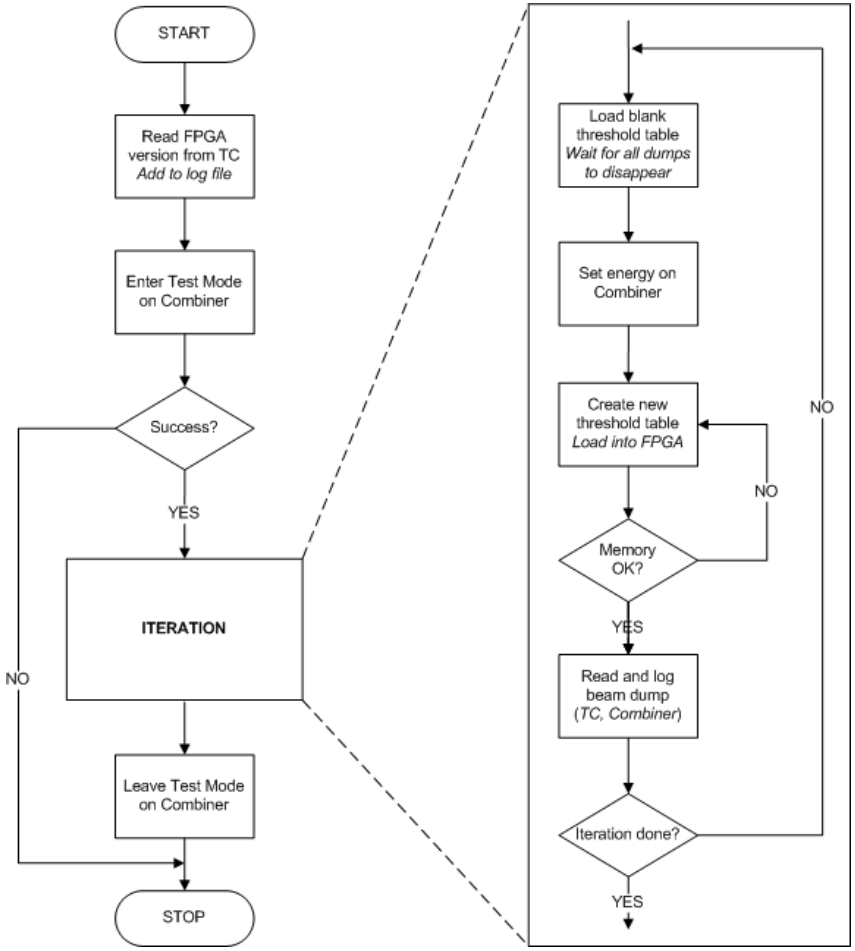
The "Exhaustive Threshold Triggering"

- VME crates
  - "Front End Computer" (FEC)
  - Combiner (CS)

- Algorithm running on FEC
  - Loads threshold maps causing one chosen threshold to trigger a beam abort
  - Checks the result on TC and CS

4/4. Software-based check

Algorithm



5. New release

- Procedure
  - A few hours of manual testing
    - Carry out simulation with testbench
      - If RCC block has been modified
    - Perform hardware-based test
      - CRC errors
      - CID errors
      - FID errors
      - Lost frames
  - 24 hours of automatic testing
    - Execute the software-based “Exhaustive testing”
- All tests need to pass!

- Inspection of the code changes by the verification engineer
  - Independent review

6. Conclusion

- Numerous bugs identified and fixed
- Number of bugs found vs. time converging
  - Reason for optimism
- No bugs found during operation
  - Sheer good luck?