Optimizing latency in Xilinx FPGA Implementations of the GBT

Steffen MUSCHTER (Stockholm University)
Christian BOHM (Stockholm University)
Sophie BARON (CERN)
Jean-Pierre CACHEMICHE (CPPM)
Csaba SOOS (CERN)

Overview

- Introduction
- GBT – general information (reminder)
- GBT implementation in an FPGA
- GBT test design
- GBT optimization
- Test results
- Outlook

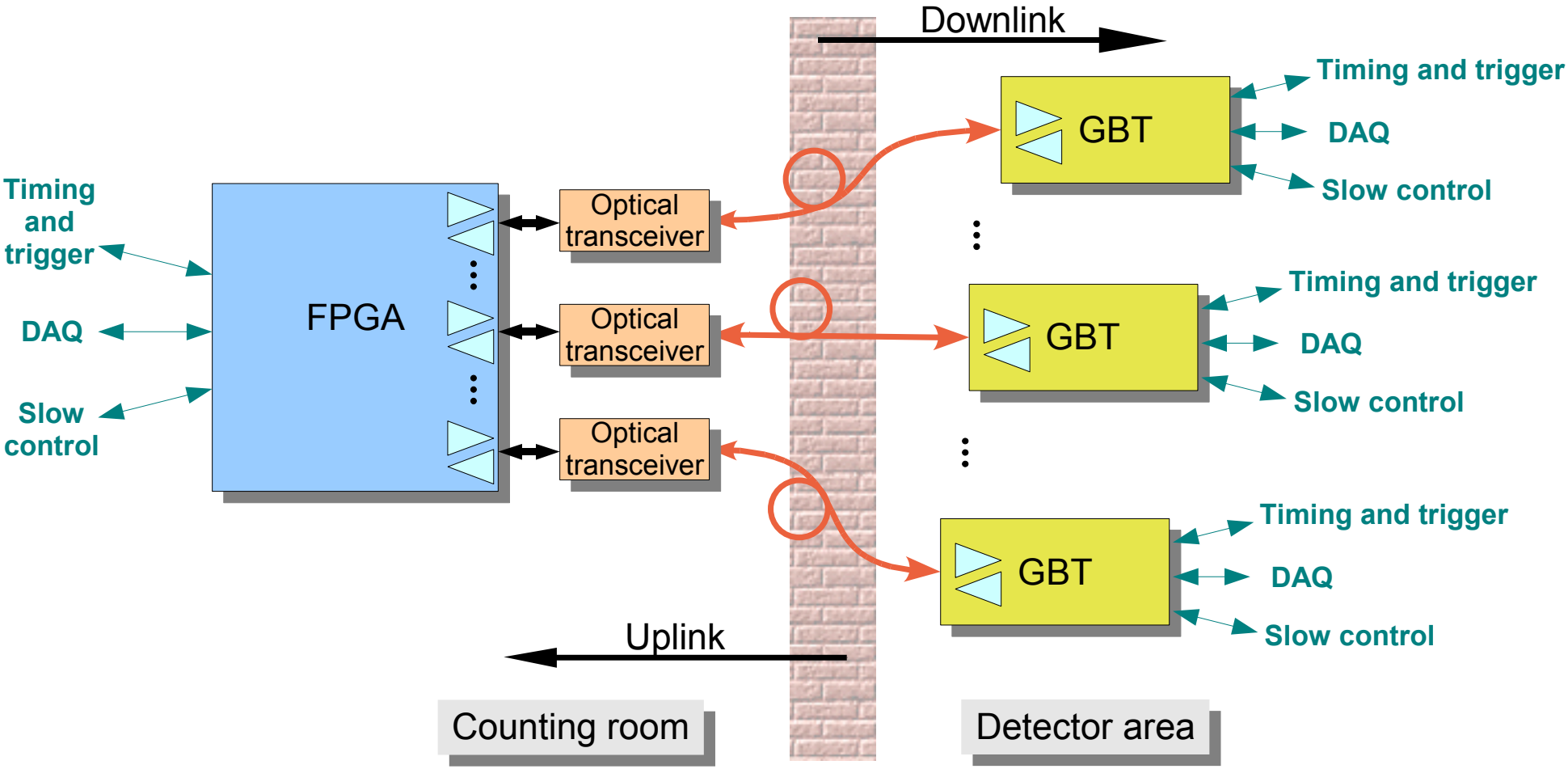
Introduction

- Motivation:
  - GBT Protocol is designed for wide use within the entire LHC
  - Focus of the “Starterkit” was on utilization
  - Until now no studies about GBT – FPGA latency have been available

  => But! Latency is also one of the important parameters especially for use in trigger

- What have we done:
  - We analyzed the GBT protocol in terms of latency
  - We optimized the GBT protocol for low latency with different setups
  - We tested the applied changes extensively with a Virtex 6 and started tests on a Stratix II Gx FPGA

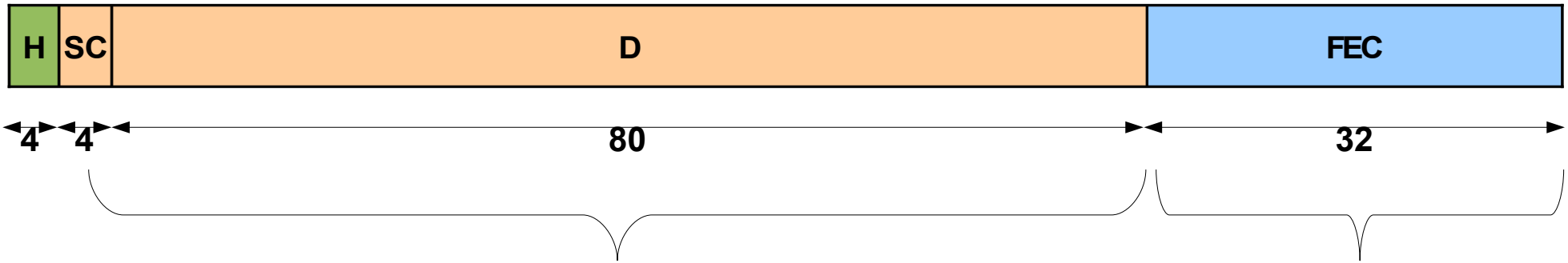
GBT- general information (reminder)*



*Frederic Marin TWEPP 2009

GBT- general information (reminder)*

SLHC frame: 120 bits @ 40 MHz ≡ 4.8 Gbps



User field: 82 bits ≡ 3.28 Gbps

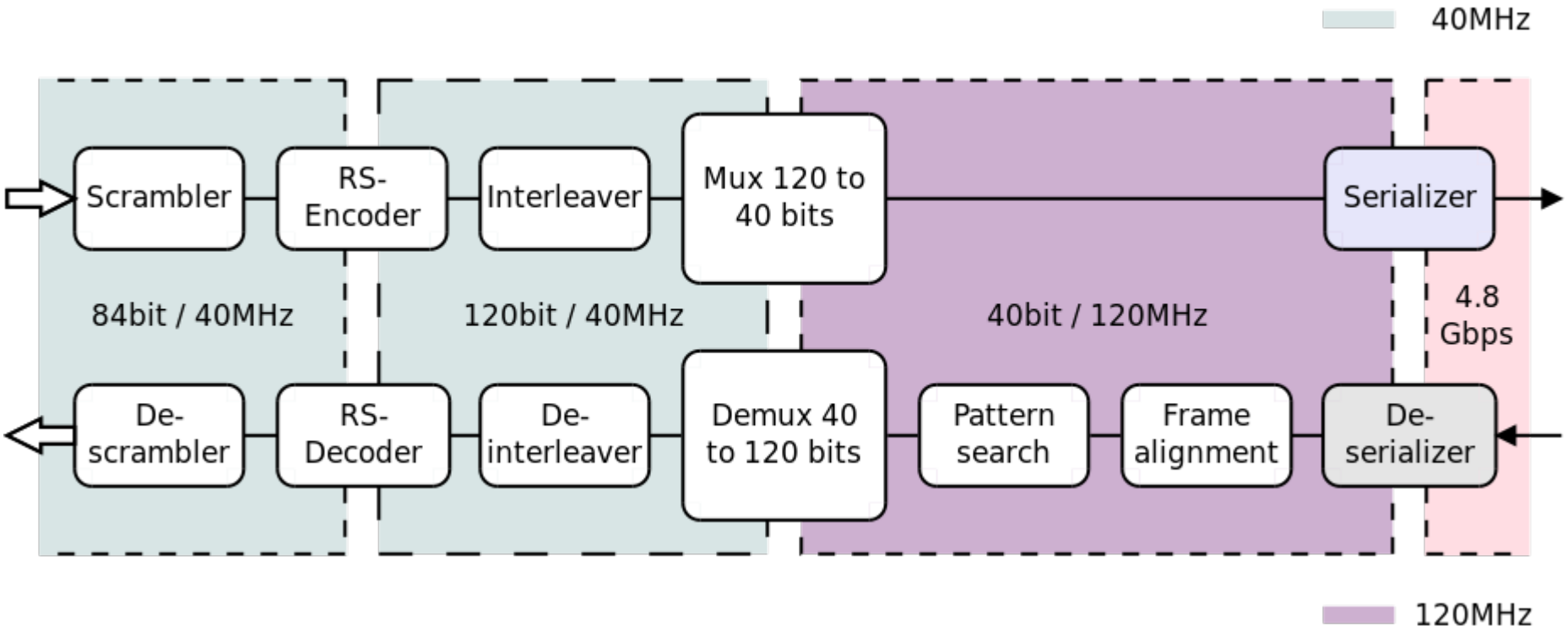
H: Header, 4 bits
SC: Slow Control 4 bits
GBT control 2 bits (80 Mb/s)
Slow control 2 bits (80 Mb/s)
D: Data (3.2 Gbps)
FEC: Forward Error Correction (32 bits)

Line efficiency: 68%

*Frederic Marin TWEPP 2009

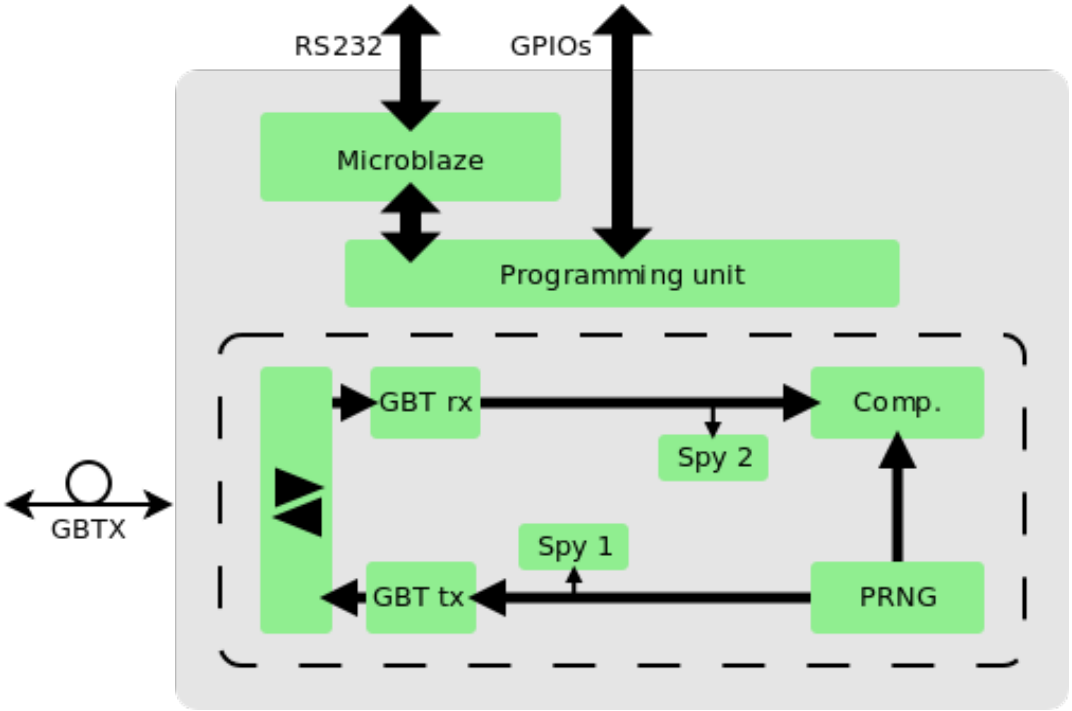
GBT implementation in a FPGA

- Data is transferred along a chain of distinct components
- Each component needs a certain number of clock cycles to operate properly
  - ➔ Finding these numbers is essential for reducing the latency



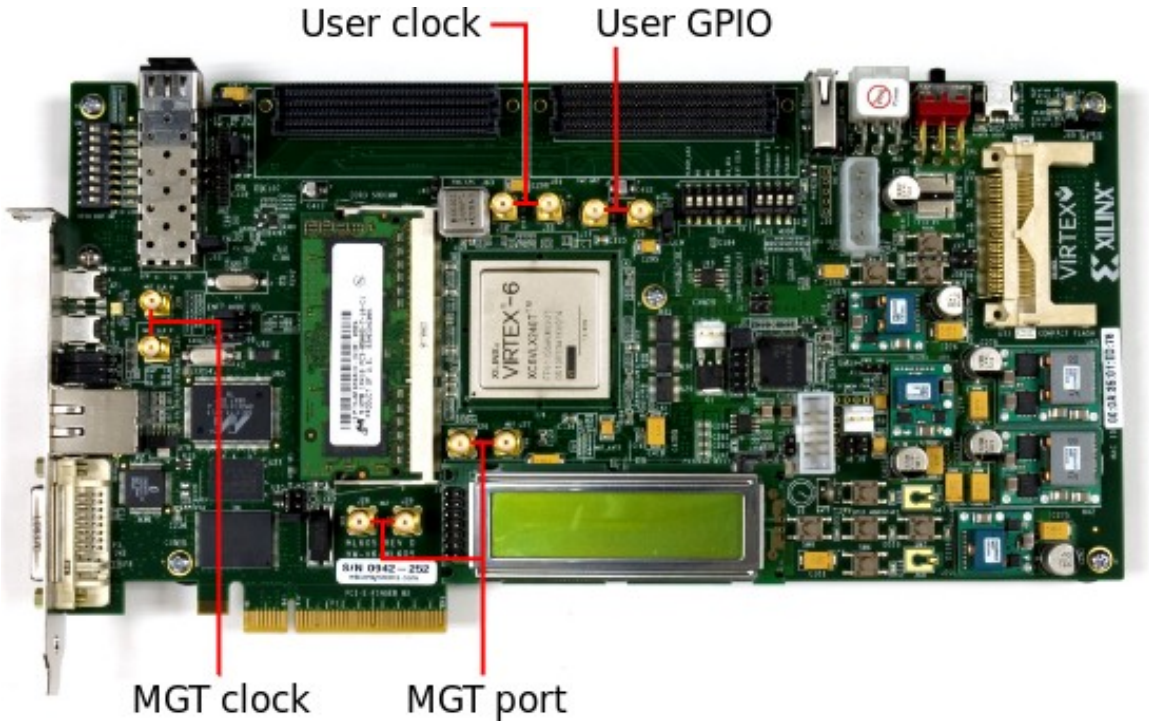
GBT Test system - overview

- The Virtex 6 test system uses:
  - The complete GBT protocol
  - One pseudo random number generator
  - Two spy memories
  - One micro-blaze for controlling and monitoring



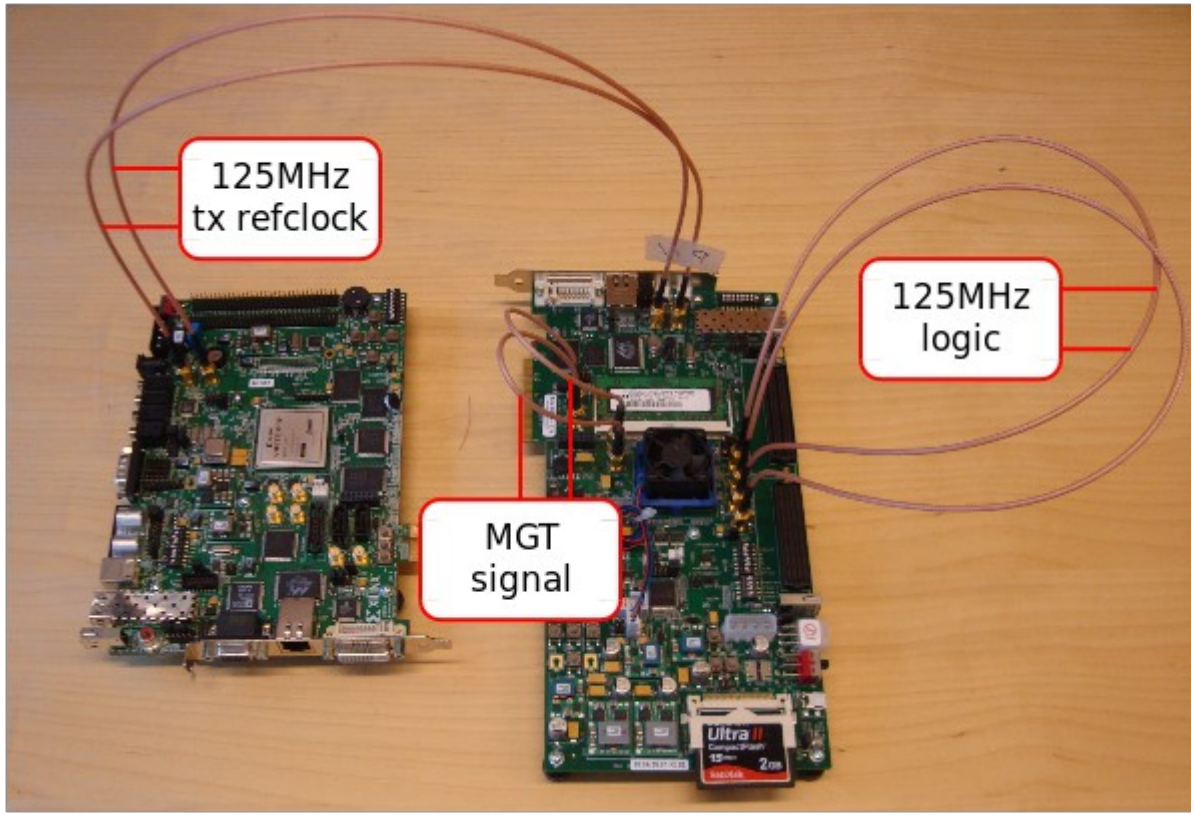
GBT Test system

- ML605 evaluation Board from Xilinx with
  - Virtex 6 LX240T Speedgrade - 1
    (for implementing the test design)
- ML507 evaluation Board from Xilinx
  (as clock source for the gigabit transceiver tx-pll)



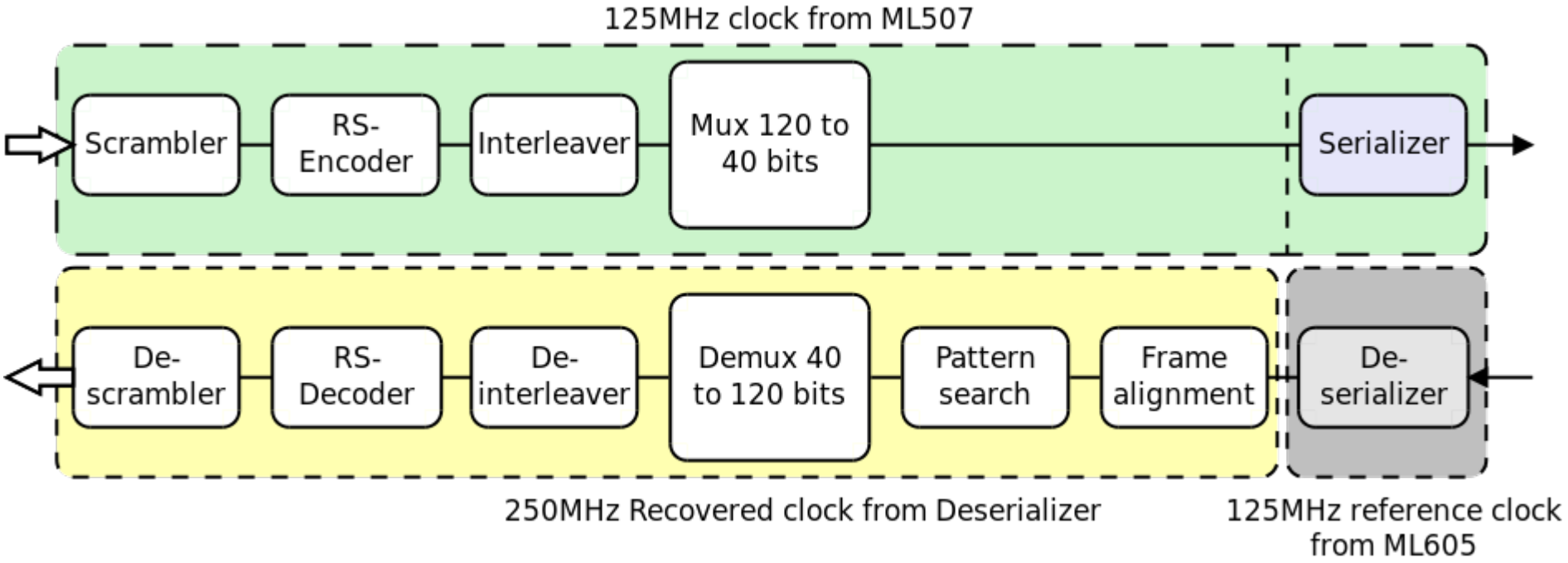
GBT Test system – hardware setup

- The Virtex 6 test system also uses:
  - A gigabit transceiver wired to SMA connectors
  - Two independent clock sources



GBT Testsystem - clock setup

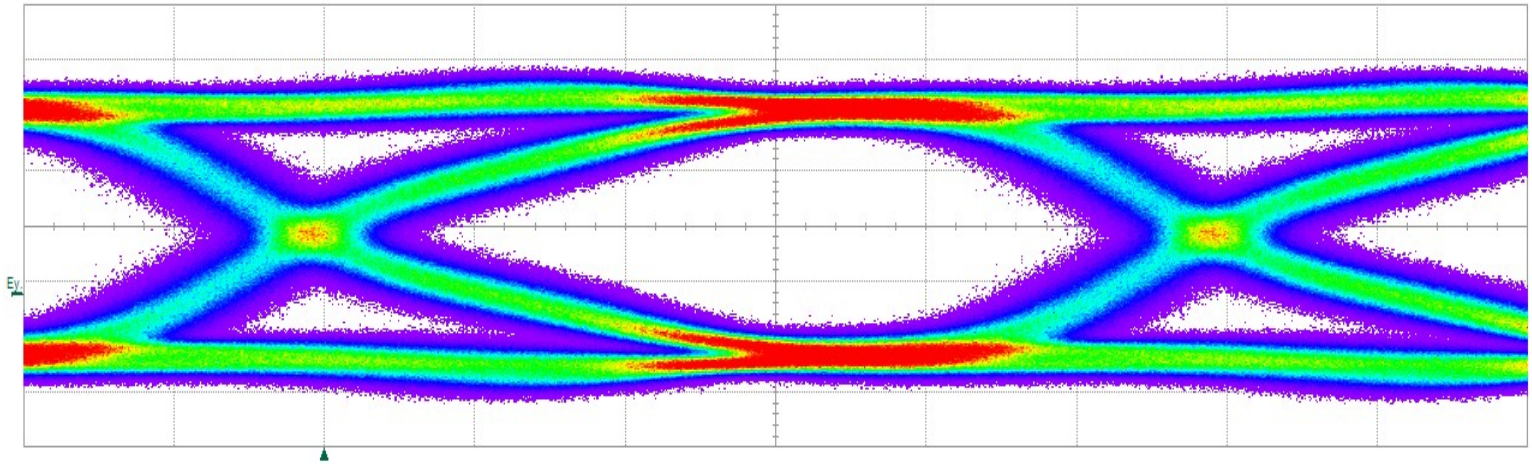
- 125MHz clock from ML507 → tx-pll → MMCM (1)* → internal logic
- 125MHz clock from ML605 → rx-pll
- 250MHz recovered clock → MMCM(2) → GBT - rx



*MMCM → Mixed Mode Clock Manager

GBT Testsystem - performance

- Transmission rate used for testing is 5Gb/s due to clock availability (this corresponds to the maximum rate with speedgrade – 1)
- High quality 125MHz clock is used to feed the tx-pll
  Jitter ≈ 108ps (p-p)*
- Output signal at 5Gb/s with jitter ≈ 60ps (p-p)*

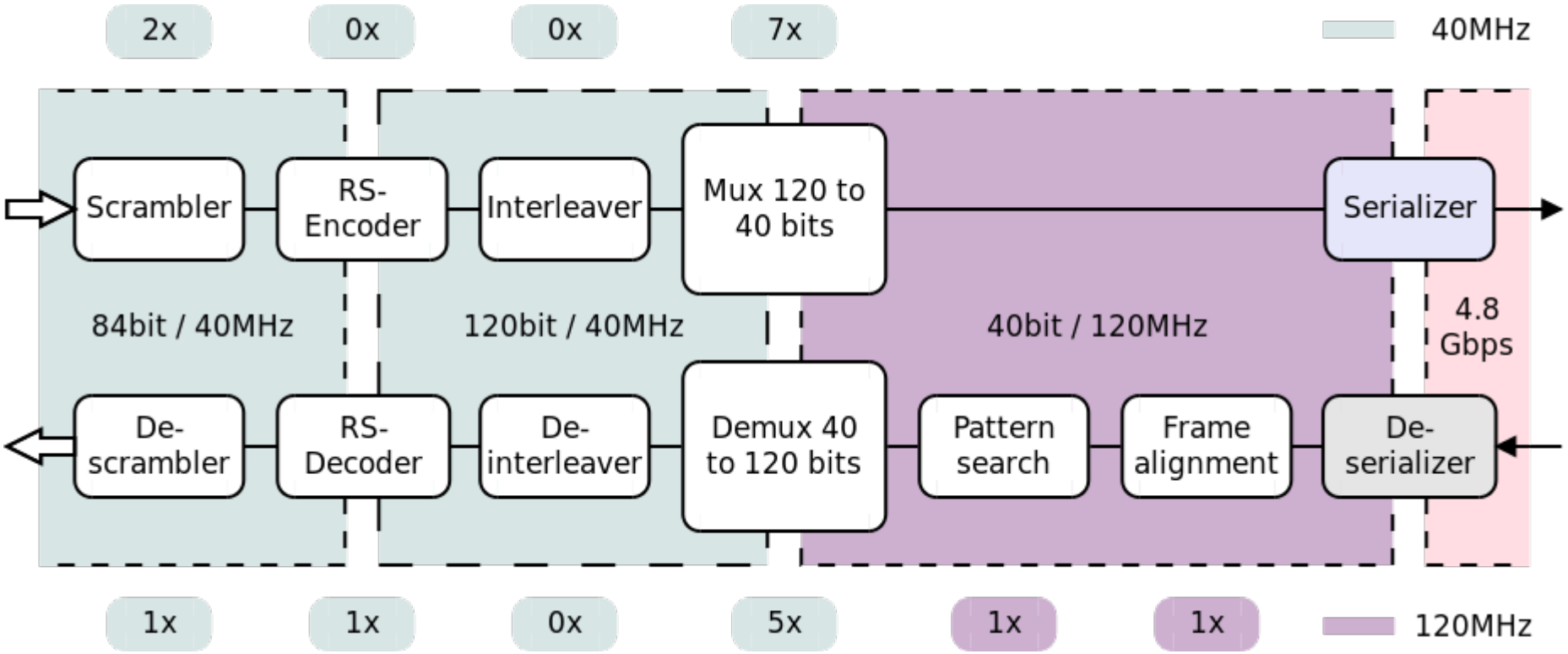


SDA Eye	EyeHeight	EyeOne	EyeZero	EyeAmpl	EyeWidth	EyeCross	EyeAvgPwr
value	628.4 mV	664.5 mV	-214.6 mV	879.2 mV	151.5 ps	49.32 %	224.2 mV
status	✔	✔	✔	✔	✔	✔	✔

*measured with a LeCroy SDA 813Zi

GBT optimization

- First test in a Virtex 6 showed a latency of 21 clock cycles
- Evaluation necessary which components contribute most
- Estimate the latency by looking at the event statements and counters (not exact latency needed)

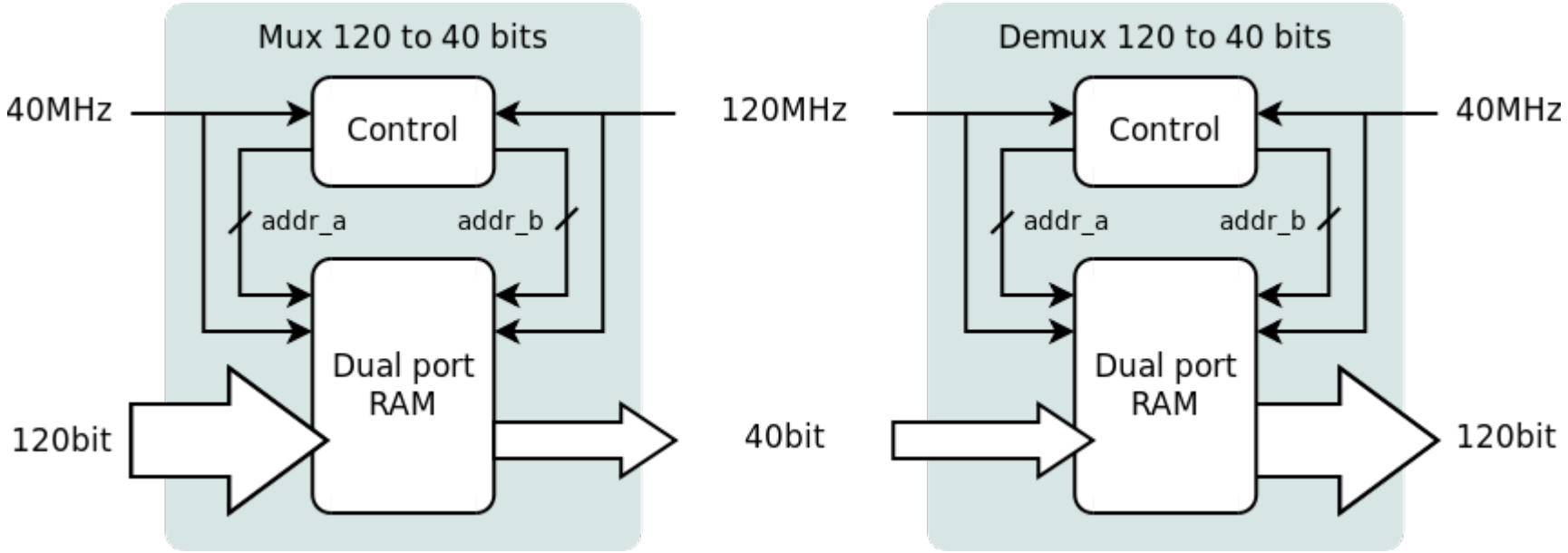


GBT optimization

- Results for estimation:
  - Transceiver chain needs min. 9 clock cycles
  - Receiver chain needs min 8 clock cycles
- Total latency min. 17 clock cycles
  without serializer and deserializer (GTX)
- Transceiver adds min. 1 clock cycle depending on the instantiation and the type of FPGA
- Does not exactly match the experimental results
  (due to not optimized GTX configuration)
- “Mux 120 to 40 bits” and “Demux 40 to 120 bits” identified as largest sources of latency

GBT optimization

- “Mux 120 to 40 bits” and “Demux 40 to 120 bits”
  - Implemented using dual port RAM modules

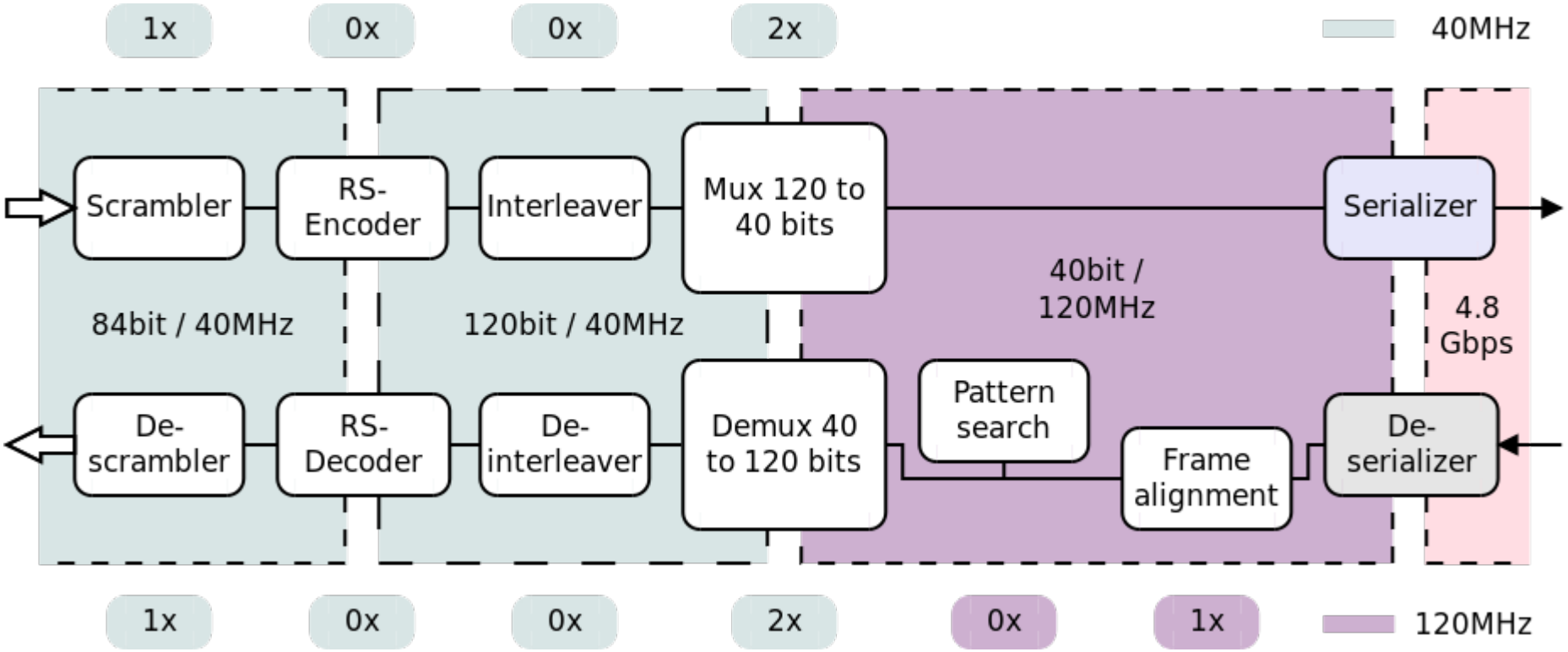


GBT optimization

- “Mux 120 to 40 bits” and “Demux 40 to 120 bits”
  - Counters optimized
  - Latency is now 2 clock cycles for each module
- “Pattern search”
  - Data is not directly processed
  - Clocking could be bypassed
- “Decoder”
  - Does not use a clock, like the Encoder
- “Scrambler” and “Descrambler”
  - Code was optimized, which allowed us to remove one event statement

GBT optimization

- Resulting latency approximation
  - Transmitter chain ≈ 3 clock cycles
  - Receiver chain ≈ 4 clock cycles



GBT optimization

- The utilization in percent refers to a design using V6LX240T with
  - 301440 Slice Registers
  - 150720 Slice LUTs
- Not optimized GBT-tx

Links	1	4	8	12
Slice Registers	260 (0%)	1016 (0%)	2024 (0%)	3032 (1%)
Slice LUTs	350 (0%)	1363 (0%)	2460 (1%)	3622 (2%)

- Optimized GBT-tx

Links	1	4	8	12
Slice Registers	93 (0%)	345 (0%)	681 (0%)	1017 (1%)
Slice LUTs	337 (0%)	1363 (0%)	2460 (1%)	3571 (2%)

GBT optimization

- Not optimized GBT-rx

Links	1	4	8	12
Slice Registers	414 (0%)	1650 (0%)	3298 (1%)	4946 (1%)
Slice LUTs	2147 (1%)	8409 (5%)	15149 (10%)	24000 (15%)

- Optimized GBT-rx

Links	1	4	8	12
Slice Registers	326 (0%)	1298 (0%)	2594 (0%)	3890 (1%)
Slice LUTs	2366 (1%)	9097 (6%)	17747 (11%)	26249 (17%)

- Optimization uses much fewer Registers but slightly more LUTs

GBT optimization

- Results so far
  - Latency of test design → 8 clock cycles including the GTX (protocol ≈ 7 clock cycles)
  - Design is running at 5 Gb/s instead of 4.8 Gb/s due to clocking of the ML605 development board
  - Optimized code does not increase the utilization when instantiated
  - No single error during two days of operation

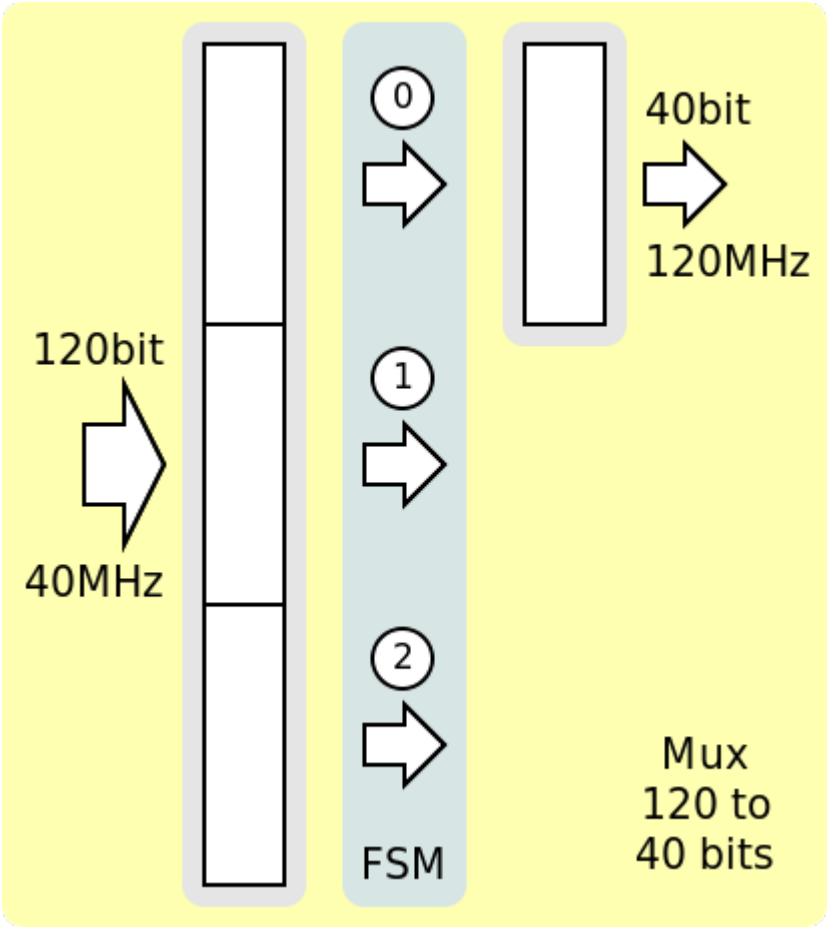
    => Bit error rate better than 10^{-15}

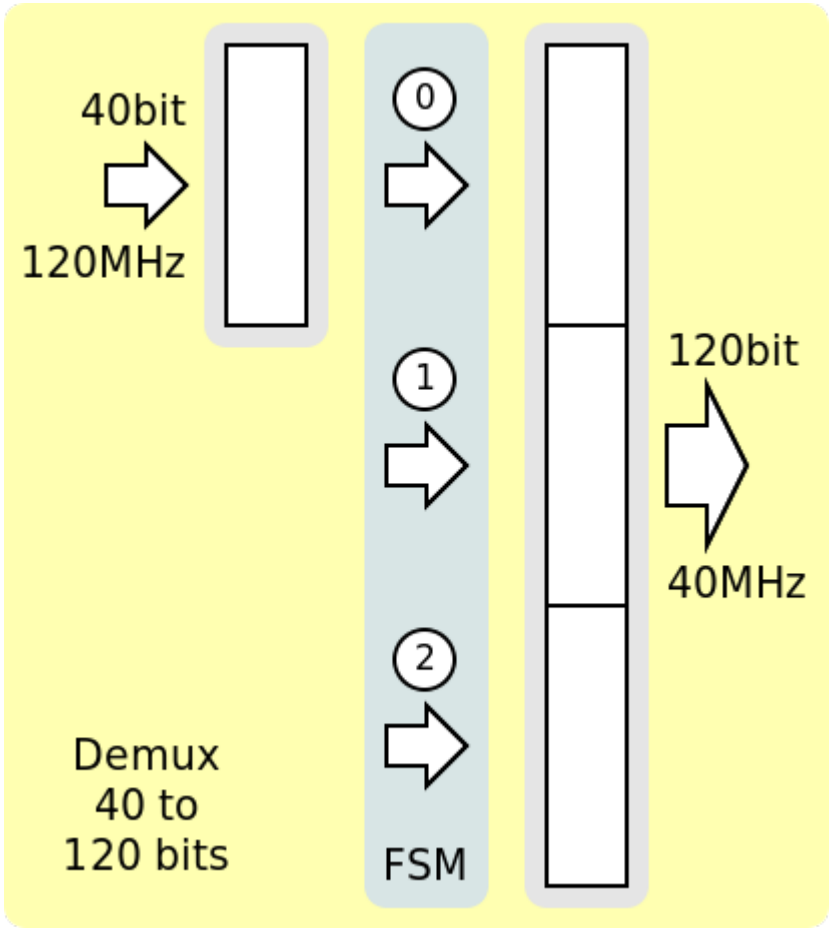
- Next step
  - Try to improve the test design latency below 8 clock cycles

    => only possible by removing the dual port RAM modules

GBT optimization

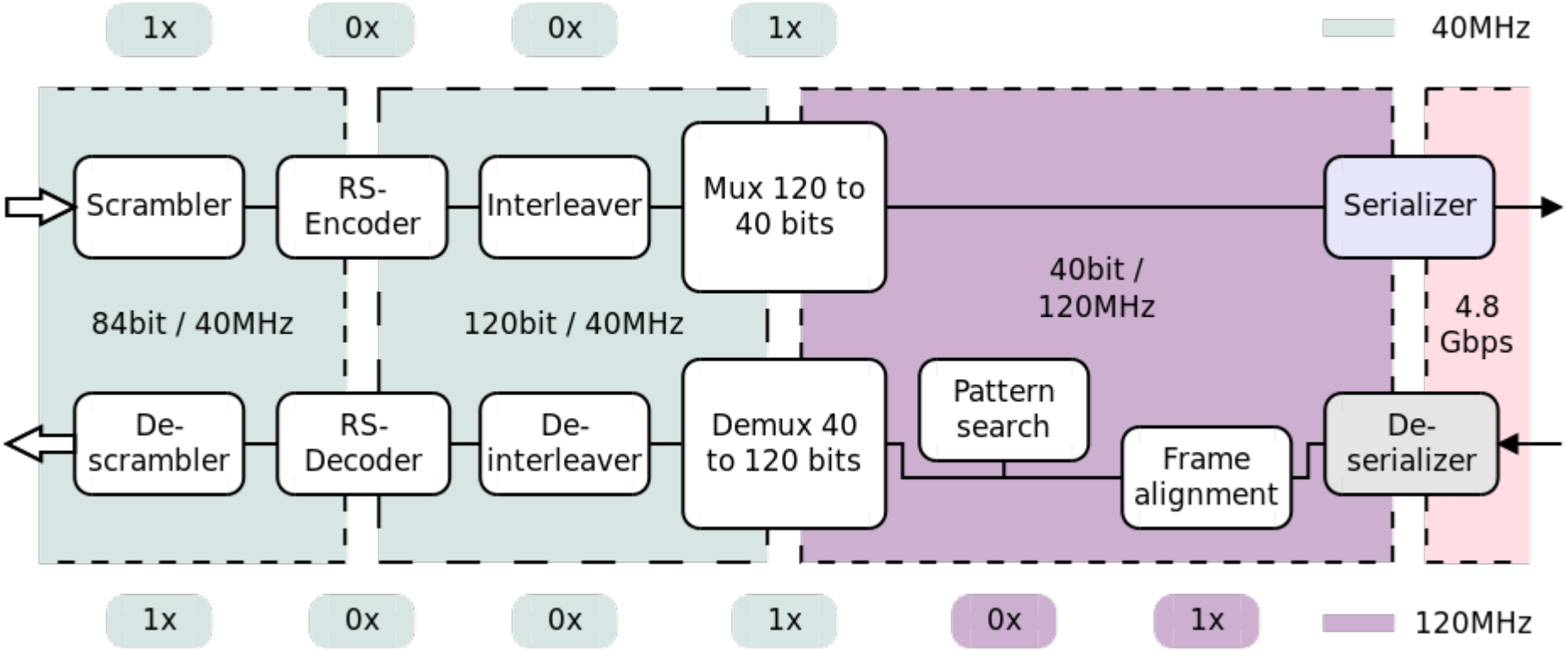
- Using registers instead of dual port RAM
  => More utilization but less Latency





GBT optimization

- The use of registers depends on the performance of the FPGA (e.g. Virtex6 LX240T Speedgrade -1 $f_{max} \approx 300MHz$)
- Utilization should not increase because of previously made optimizations



GBT optimization

- Utilization studies for optimization using registers between 120MHz and 40MHz clock domains
- Not optimized GBT-tx

Links	1	4	8	12
Slice Registers	260 (0%)	1016 (0%)	2024 (0%)	3032 (1%)
Slice LUTs	350 (0%)	1363 (0%)	2460 (1%)	3622 (2%)

- Optimized GBT-tx

Links	1	4	8	12
Slice Registers	246 (0%)	974 (0%)	1946 (0%)	2918 (0%)
Slice LUTs	381 (0%)	1505 (0%)	2771 (1%)	3859 (2%)

GBT optimization

- Not optimized GBT-rx

Links	1	4	8	12
Slice Registers	414 (0%)	1650 (0%)	3298 (1%)	4946 (1%)
Slice LUTs	2147 (1%)	8409 (5%)	15149 (10%)	24000 (15%)

- Optimized GBT-tx

Links	1	4	8	12
Slice Registers	519 (0%)	2070 (0%)	4138 (1%)	6194 (2%)
Slice LUTs	1849 (1%)	7396 (4%)	14949 (9%)	22413 (14%)

- Optimization was successfully tested at 5Gb/s for 2 days without a single error

=> Bit error rate better than 10^{-15}

Results

- Final results:
  - The startup test design latency was 21 cycles including GTX
  - The optimized test design with RAM modules had a latency of 8 cycles (7 cycles only for the protocol)
  - The optimized test design with registers had a latency of 6 cycles (5 cycles only for the protocol)
  - The optimized code does not increase the utilization when instantiated
  - All optimizations were tested carefully with a Virtex 6
  - No single error during two days of operation with both optimizations

    => Bit error rate better than 10^{-15} for both optimizations

Outlook

- Optimization will be available from the GBT-FPGA team soon
- Three different instantiations planned
  - Optimized code with 7 clock cycles
    - Using dual port memories
    - Using only registers
      (if one needs to spare resources)
  - Optimized code using only registers with 5 clock cycles
- Test of optimization on Stratix FPGAs and Virtex 5
  - First test already started and seem to be promising
- Optimized code ready for distribution until the end of the year
  - Could be accessed from the GBT – FPGA website

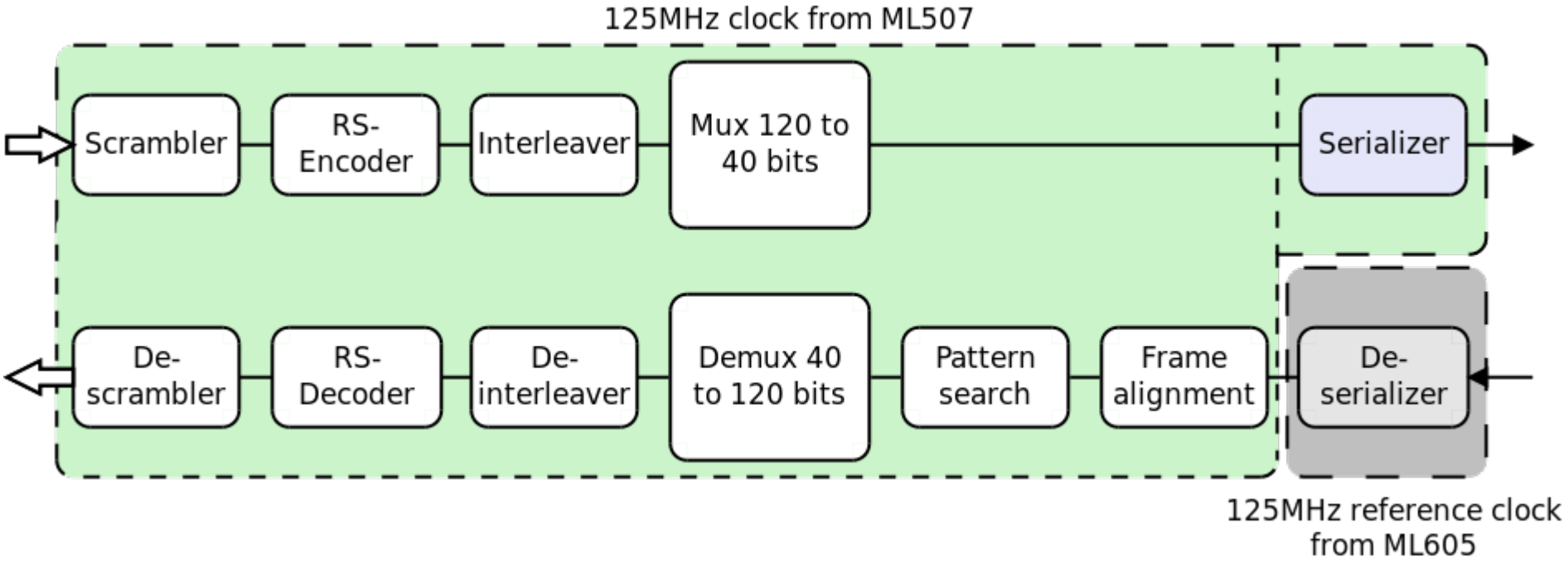
    https://espace.cern.ch/GBT-Project/GBT-FPGA/default.aspx

Thank you

Steffen MUSCHTER (Stockholm University)
Christian BOHM (Stockholm University)
Sophie BARON (CERN)
Jean-Pierre CACHEMICHE (CPPM)
Csaba SOOS (CERN)

GBT Testsystem - clock setup (2)

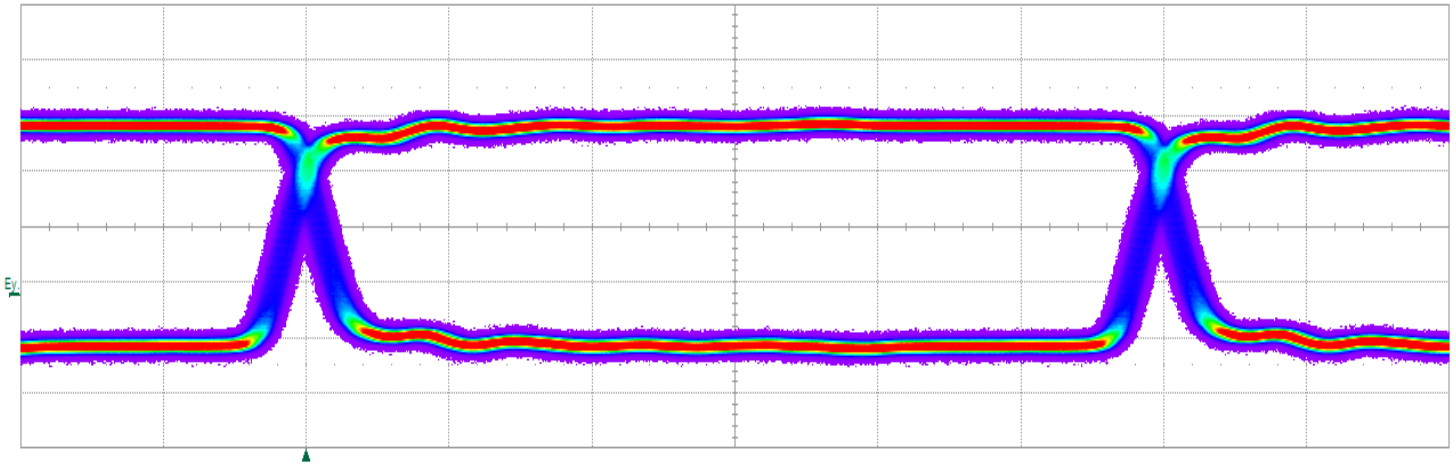
- Add jitter to the internal logic to test the design for stability
- 125MHz clock from ML507 → tx-pll → GPIO (SMA) → User clock (SMA) → MMCM (1)* → internal logic
- 125MHz clock from ML605 → rx-pll



*MMCM ⟶ Mixed Mode Clock Manager

GBT Testsystem – performance (2)

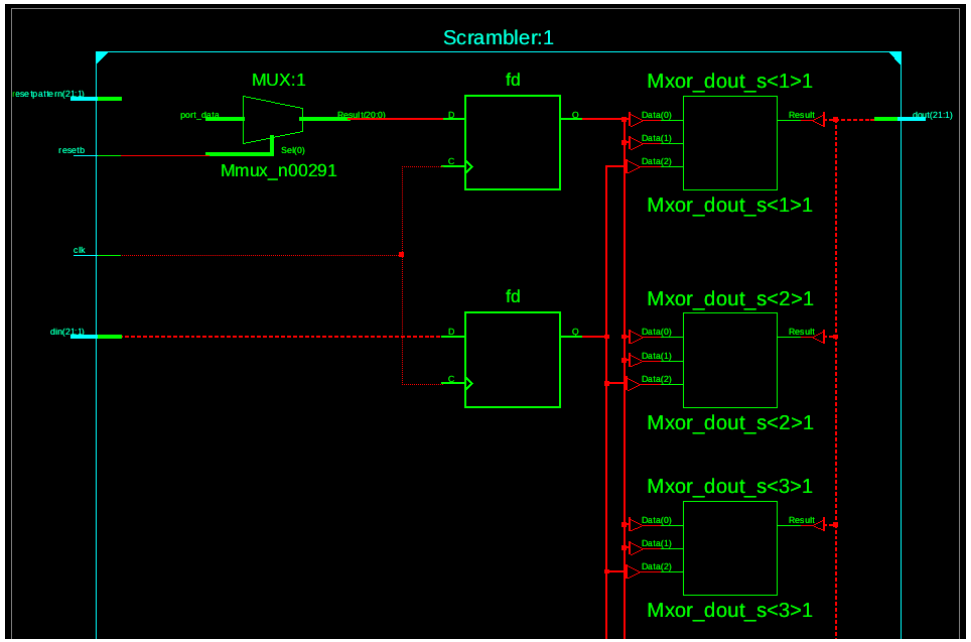
- Clock for internal logic has increased jitter due to loop back through normal GPIOs
  Jitter ≈ 387ps (p-p)
- Both clock setups were tested with the high quality and this clock signal to test the stability of the protocol



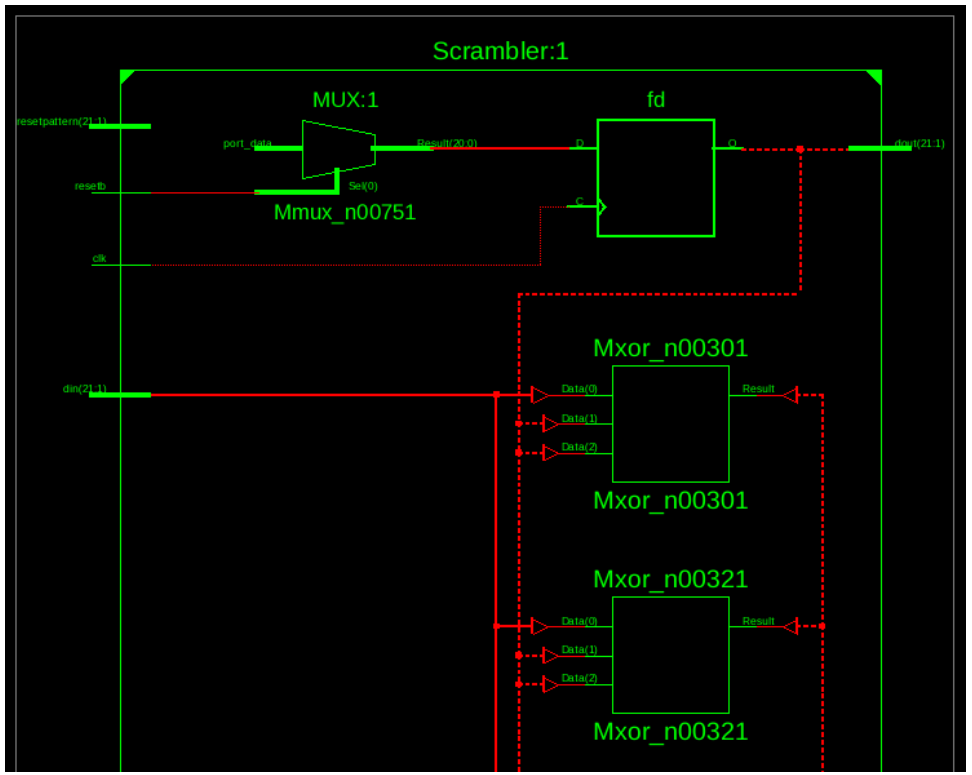
*measured with a LeCroy SDA 813Zi

GBT optimization

- RTL schematic of the Scrambler before and after optimization



before



after