In2p3

# Implementing the GBT data transmission protocol in FPGAs


Sophie BARON (CERN)
Jean-Pierre CACHEMICHE (CPPM)
**Frédéric MARIN (CPPM)**
Paulo MOREIRA (CERN)
Csaba SOOS (CERN)

# Outline

- Introduction
- GBT ASIC presentation (reminder)
- Typical architecture of use
- GBT protocol implementation on FPGA
- Test setup and results
- Further work
- Reference design availability
- Conclusion

# Introduction

- Aim of our study:
  - Interoperability between GBT chip and FPGAs must be guaranteed
  - Check the compatibility of FPGAs (Xilinx and Altera) with the GBT protocol

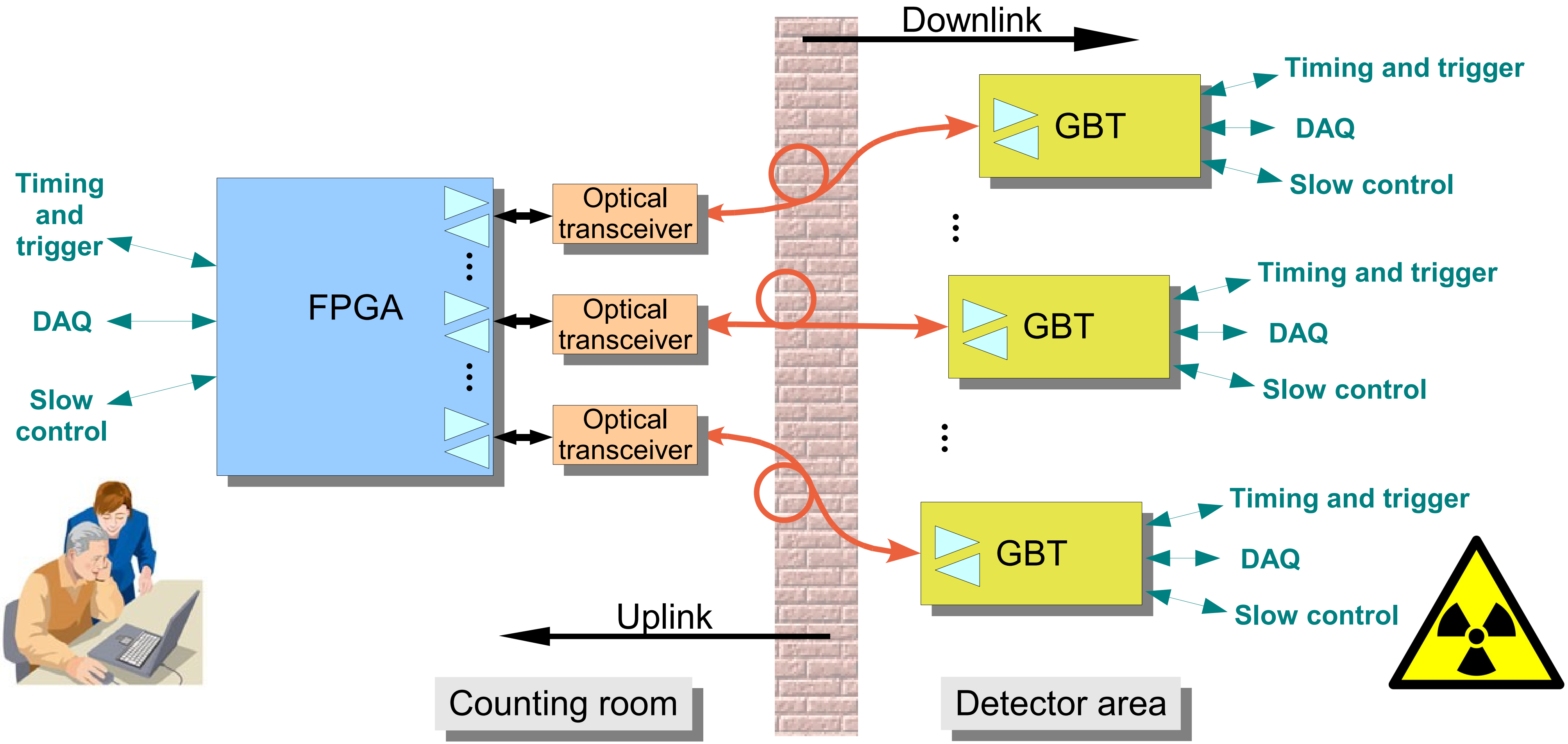
- Warning:
  - This presentation will give some results and numbers concerning the implementation of the GBT protocol on Altera and Xilinx FPGAs but be aware that they cannot be used in any case as a tool of comparison of the two vendors

# GBT ASIC presentation

- **Motivations**
  - Radiation hardened chip compatible with Super LHC luminosity
  - Reduce the number of optical fibres installed
  - Concentrate R&D effort

- **GBT ASIC characteristics**
  - General purpose bidirectional optical serial link running at 4.8 Gbps, on which transit:
    - Data
    - Trigger, Timing and Control
    - Slow control
  - Custom transmission protocol
    - Able to correct up to 16 consecutive erroneous bits

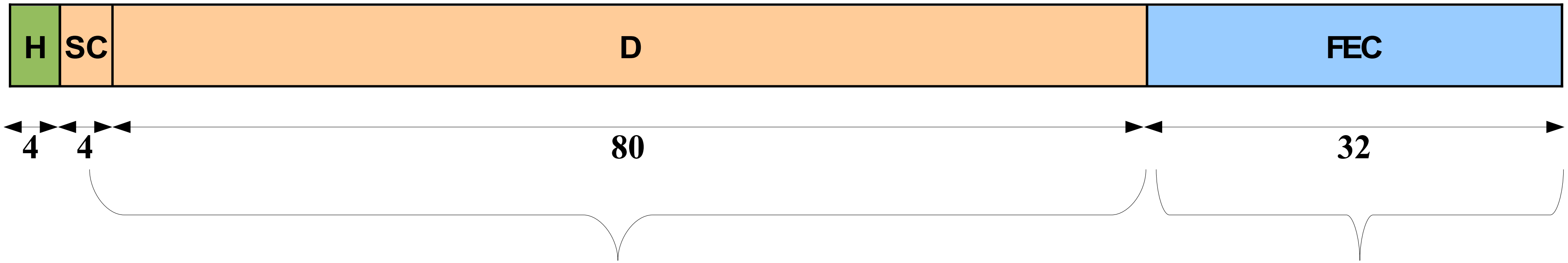
# Typical architecture




GBT protocol needs to be compatible with the FPGAs and vice versa

# Frame description

**SLHC frame: 120 bits @ 40 MHz ≡ 4.8 Gbps**




**User field: 82 bits ≡ 3.28 Gbps**

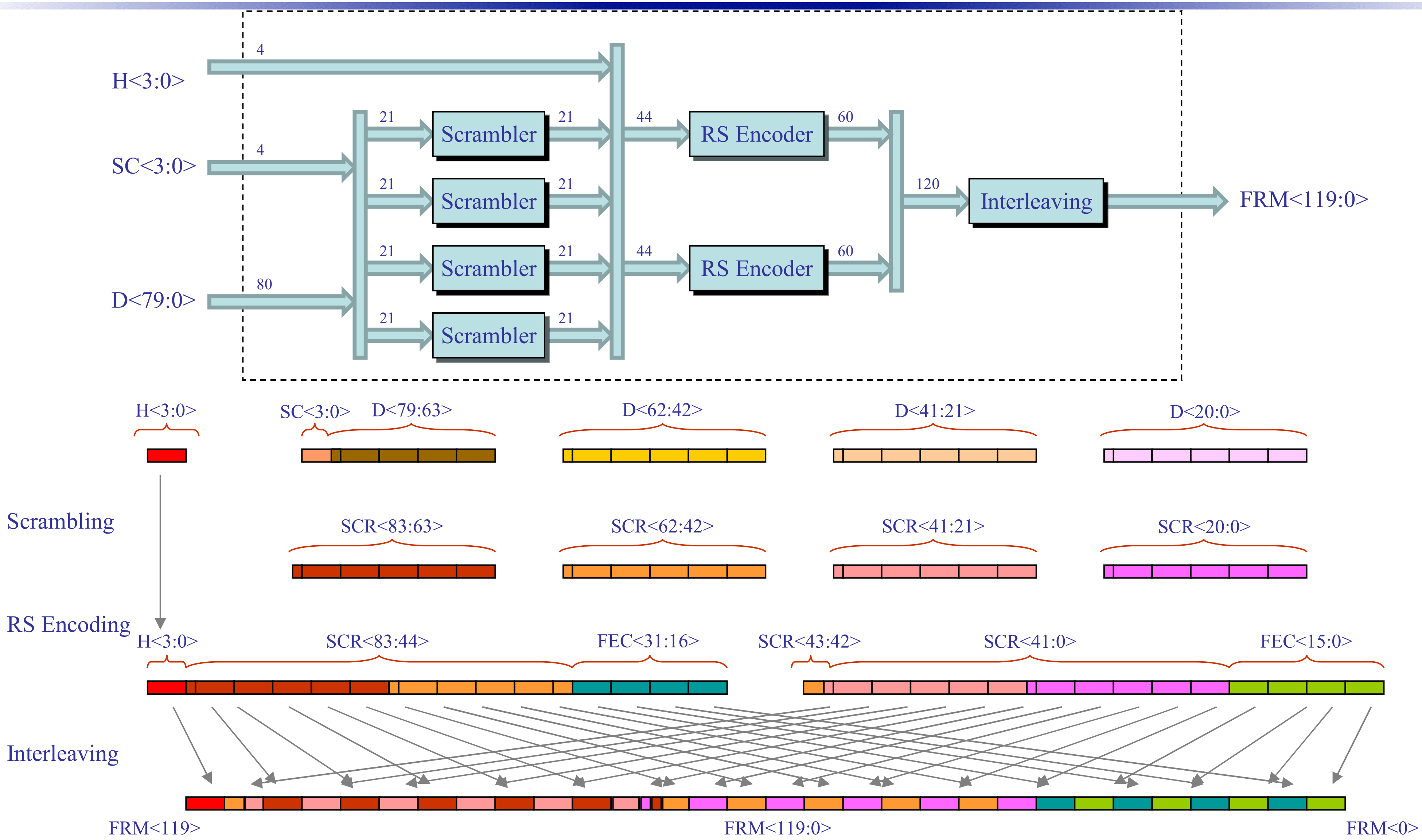
**Reed-Solomon overhead allowing to correct up to 16 consecutive erroneous bits**

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%**

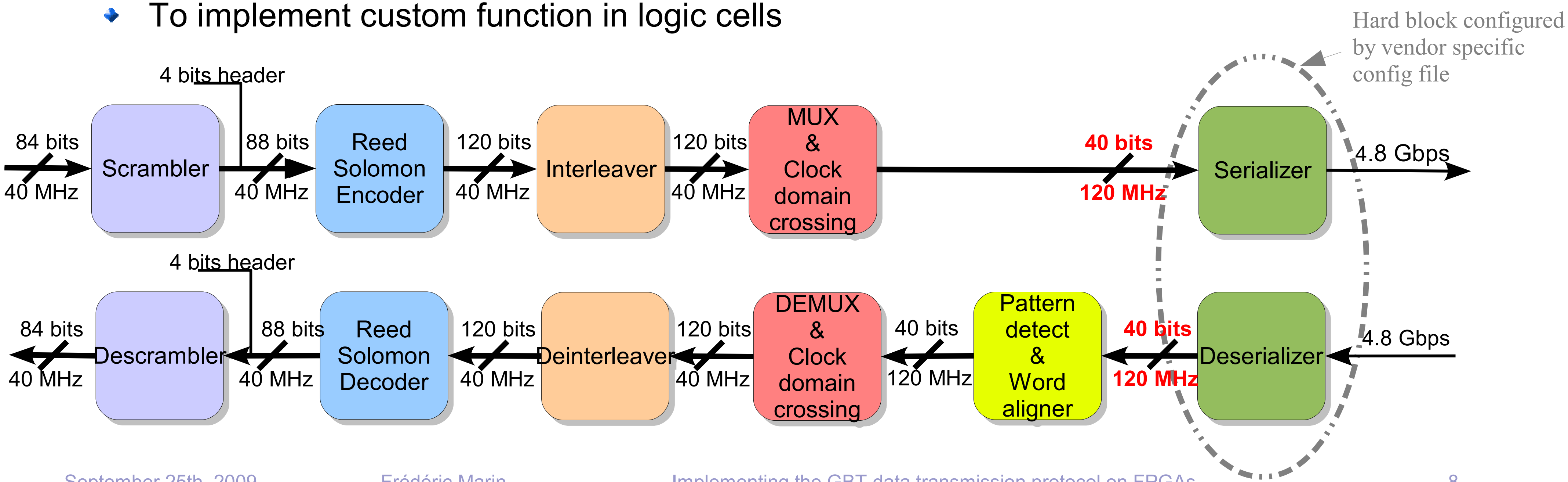
# GBT protocol – Encoding scheme overview




The frame is shifted out MSB first, that is: FRM<119>, FRM<118>, … FRM<0> (The header shifts out first)

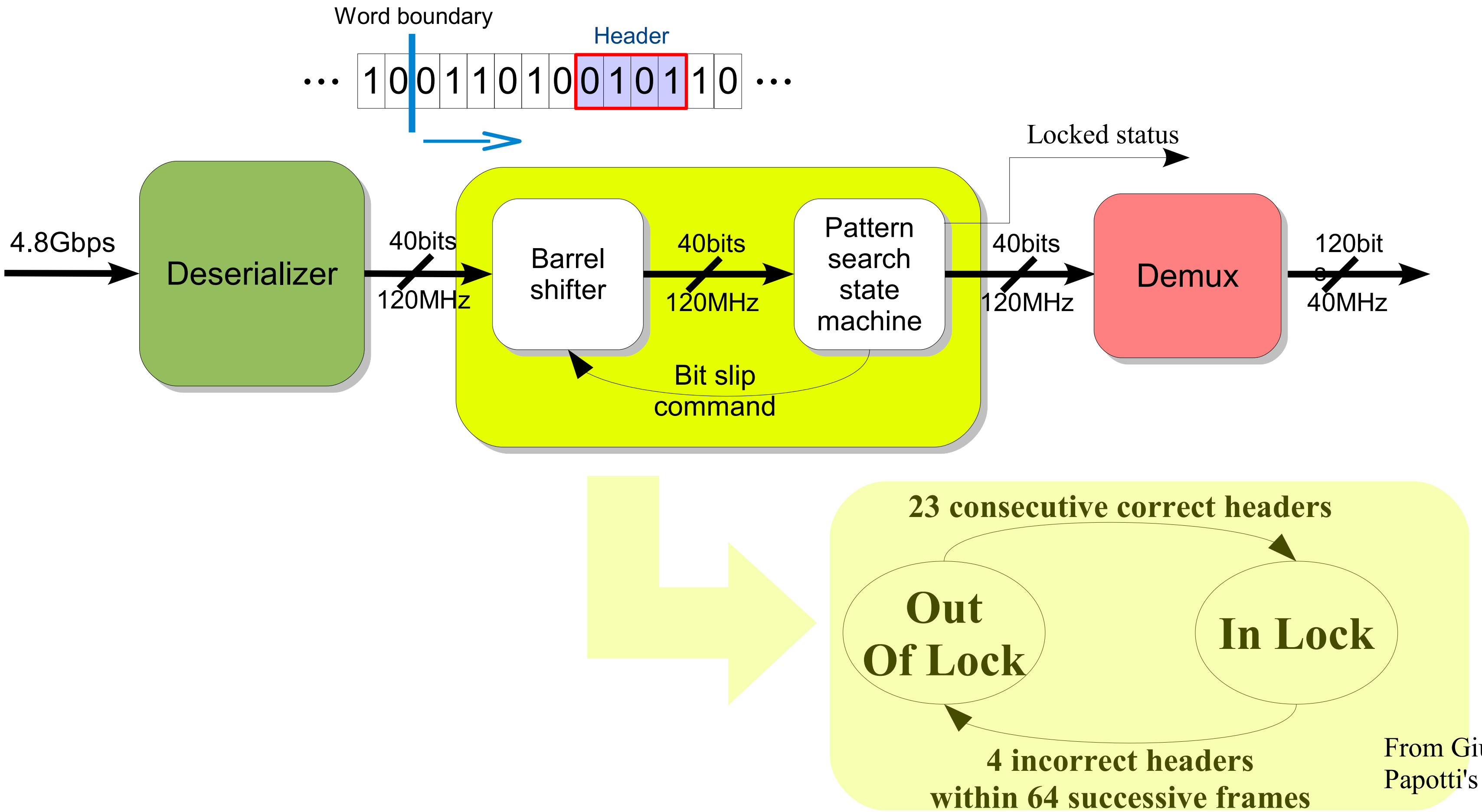
# FPGA implementation

- FPGA devices from Altera or Xilinx embed serial transceivers configurable for several standard telecommunication protocols (XAUI, Gigabit Ethernet, PCIe, ...)
- **But:**
  - Transceivers include neither scrambler nor Reed Solomon encoder nor interleaver
  - Their word aligner/comma detection module are not flexible/configurable enough to fit our needs
  - The width of the parallel interface of the ser/des can only take predefine values
- So the way to implement the GBT protocol on FPGA is:
  - To use the transceiver in the most basic way (ser/des function)
  - To implement custom function in logic cells



# Pattern search for data alignment

- The pattern search and alignment functions are achieved with two entities:
  - A barrel shifter allowing to bit shift the incoming data
  - A state machine looking at the expected header position in the frame and giving the “locked” status




From Giulia Papotti's thesis

# Resource usage

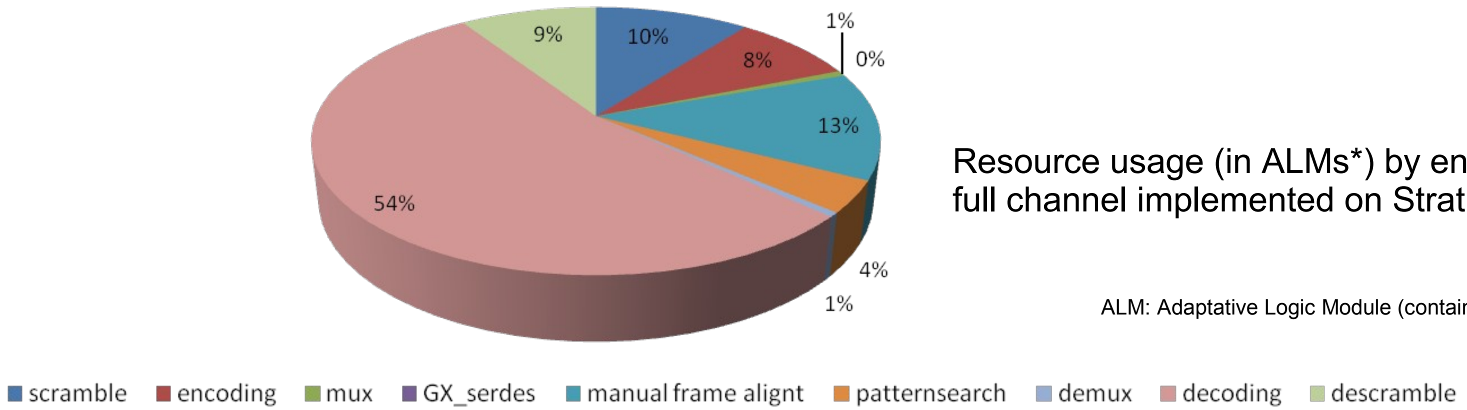
- Implementing several channels leads to an important usage of logic cells resources

| Max usable/available nb of channel | Altera Stratix II GX | Logic cells usage in % |
|---|---|---|
| 8/8 | EP2SGX30D | 92% |
| 12/12 | EP2SGX60D | 78% |
| 16/16 | EP2SGX90E | 69% |
| 20/20 | EP2SGX130G | 59% |
| 24/24 | | |

| Max usable/available nb of channel | Xilinx Virtex 5 | Logic cells usage in % |
|---|---|---|
| 3/8 | XC5VFX30T | 87% |
| | | |
| 10/16 | XC5VFX100T | 93% |
| 13/20 | XC5VFX130T | 94% |
| 20/24 | XC5VFX200T | 96% |

This table gives only ressource usage in terms of logic cells but other ressources are used such as DSP blocks or RAM.
Differences between Altera and Xilinx logic cells usage can be explained by the different policies they adopt in term of ratio between number of logic cells and number of transceivers.

- The Reed Solomon decoder takes most of the resources

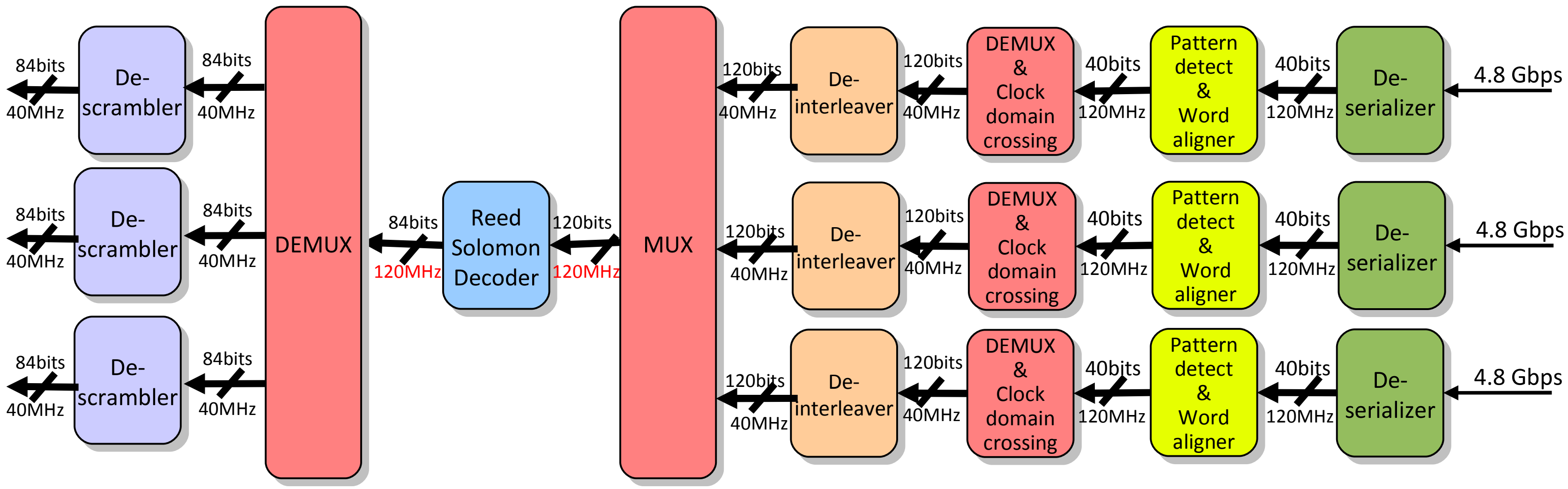



Resource usage (in ALMs*) by entity for one full channel implemented on Stratix II GX

ALM: Adaptative Logic Module (contains 2LUTs and 2 registers)

# Optimization

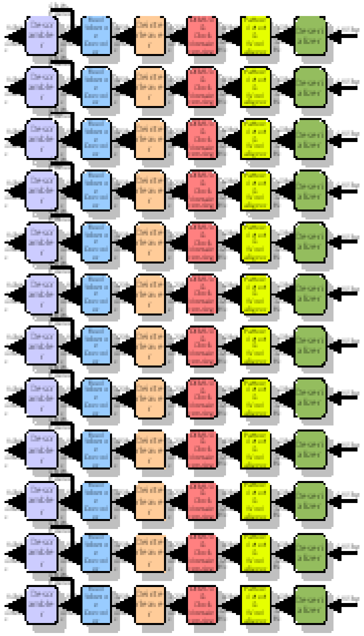
- Resource sharing technique:
  - Use one decoding module for n channels by increasing its working frequency by n



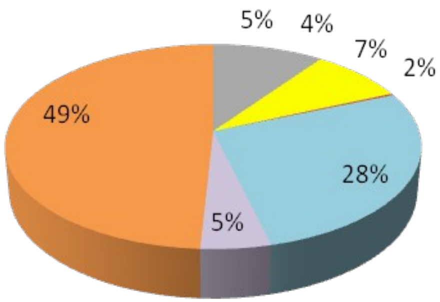

  - Gain not as significant as expected because multiplexing and demultiplexing functions use many registers

# Optimization results for 12 links

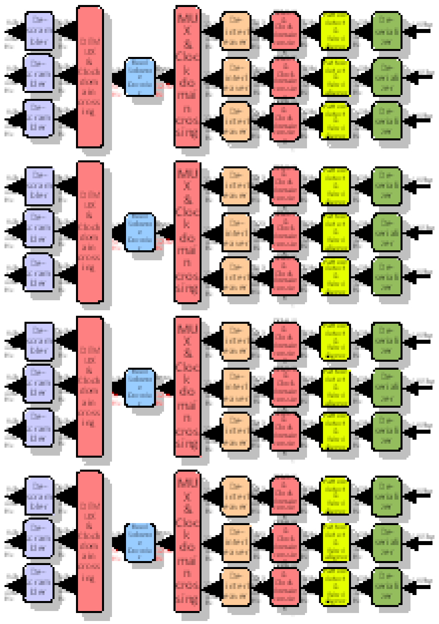
## No optimization



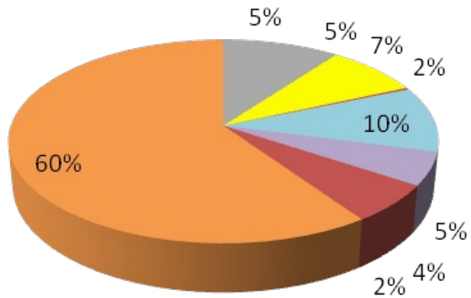
51% occupancy
ALMs of StratixIIGX90




## Optimization: one decoder for 3 links



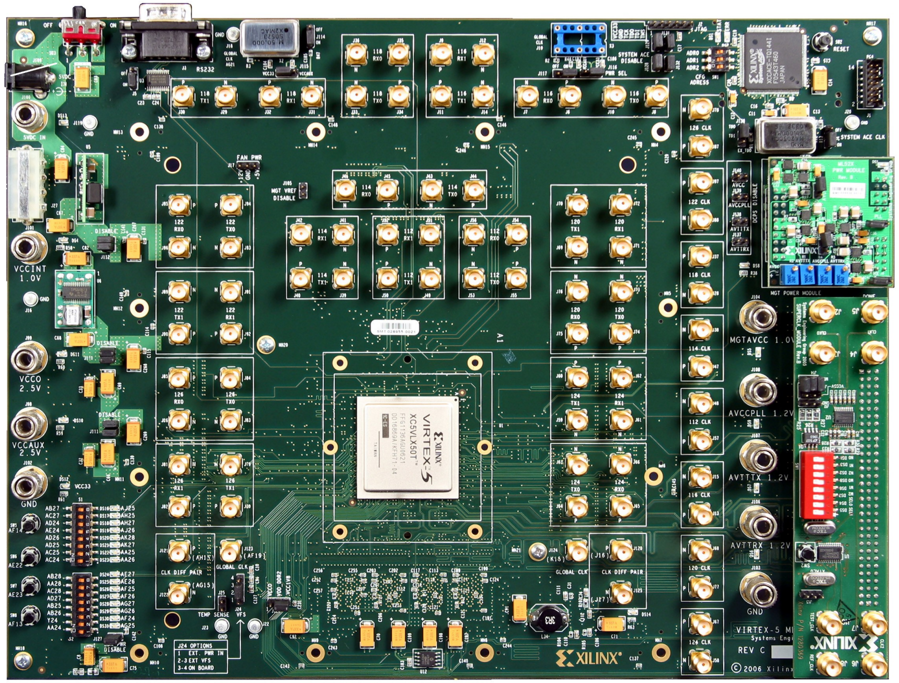
40% occupancy
ALMs of StratixIIGX90



# Tests

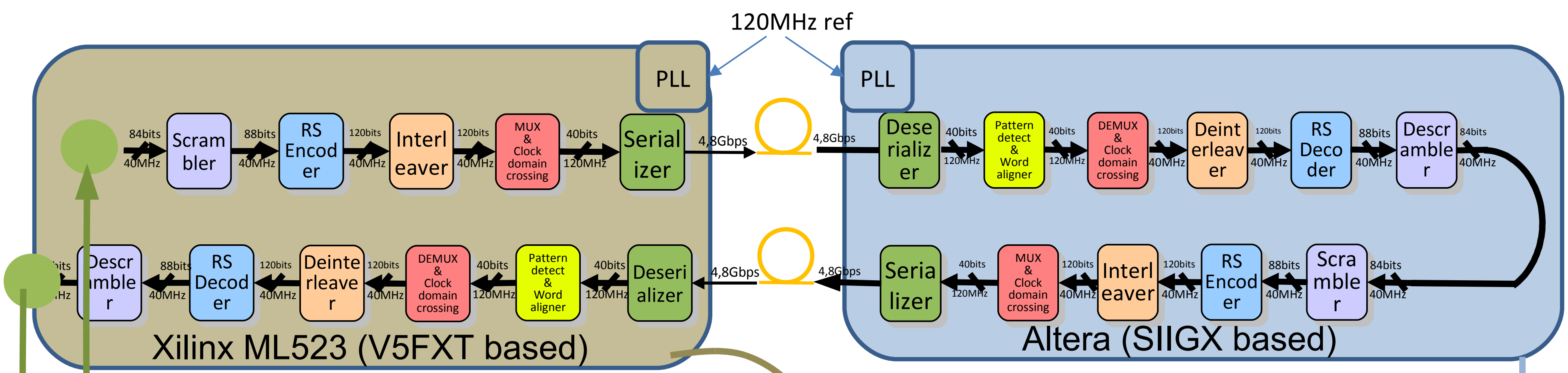
- Tests of implementation have been done using two different platforms:

**Xilinx Virtex 5 ML523 platforms**

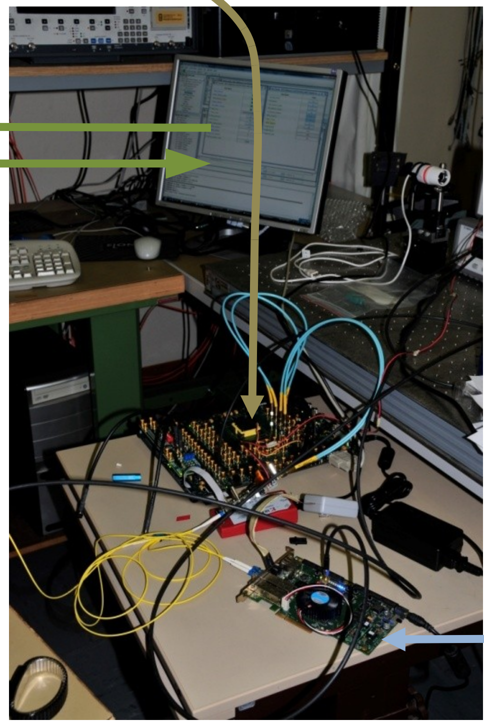


**Altera StratixIIGX PCI express development board**

# Tests (2)




Generator
Error counters

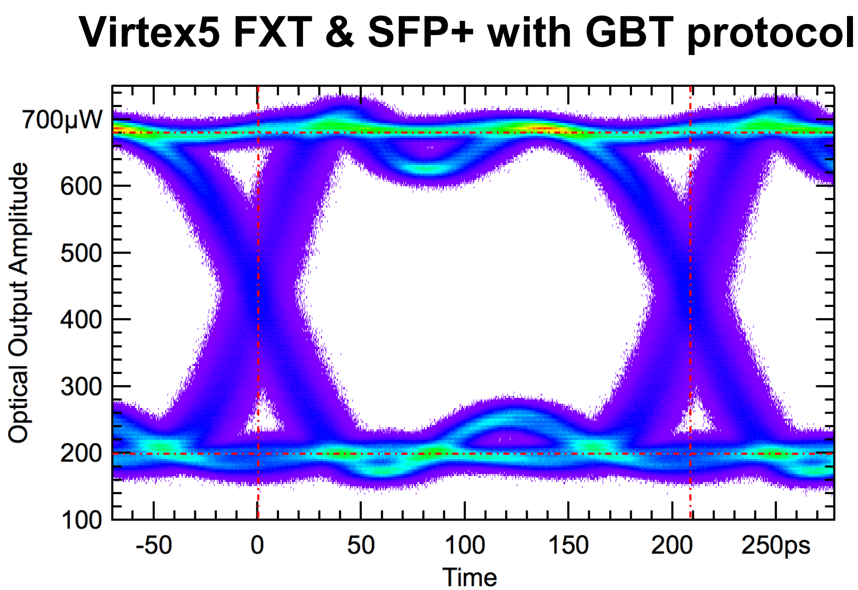


- 120MHz ref generated by an AGILENT JBERT N4903
- Electrical (SMA) to optical mezzanine on Xilinx side
- Optical transceivers SFP+ 1300nm types from MergeOptics
- Generator and Error detection controlled and monitored through ChipScope.
- Generated words: constants or flying bits

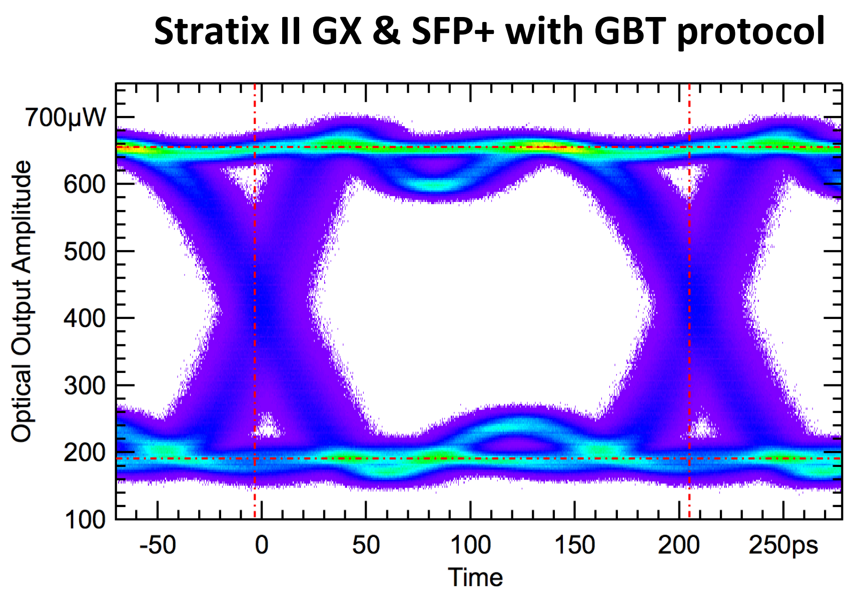
**No error detected during 2 hours of running**

# Measurements

## Xilinx and Altera platforms both show excellent performances




Total Jitter @BER$10^{-12}$ = 78 ps pkpk




Total Jitter@BER$10^{-12}$ = 90 ps pkpk

- **Test conditions:**
  - ML523 (Virtex 5 FXT) for Xilinx, Stratix II GX PCIe for Altera both powered by the power supply given in the kit
  - Reference clock generated by J-BERT 4903A from Agilent
  - Same SFP+ 1300nm optical transceiver from MergeOptics, 50 cm of optical fiber, for both platforms
  - Measurements made with Lecroy SDA100G sampling scope equipped with 10 GHz optical sampling head

**Remark:** the Stratix II GX performances are slightly lower because the reference clock could not be as ideally connected as for the Virtex 5 due to the plateform connectivity

# Further work

- Continue to optimize the implementation:
  - One idea could be to pipeline Reed Solomon decoder to be able to multiplex more channels
- Test the compatibility with real GBT device
- Implement this protocol on the latest devices (StratixIV GX and Virtex 6)
- Study implementation of constant phase/latency of embedded transceivers (see Ioannis Papakonstantinou slides)

# Design availability

- Reference designs implementing one or several full channels for both Xilinx and Altera devices exist

- Firmware starter kit will be offered to the interested people. It will contain:
  - Source code for both implementations (Altera & Xilinx)
  - Documentation
  - Basic support
  - You will just need to provide the hardware

- You just need to contact Sophie Baron (Sophie.Baron@cern.ch) giving
  - Your name
  - The description of your project

# Summary

- First time the GBT protocol runs on real platforms
  - These platforms will be used to test the GBT ASIC
- GBT protocol runs on both major FPGAs vendors
  - Interoperation proven
- Further resource usage optimization under investigation
- Firmware starter kit will be available soon

References:


- Giulia Papotti's thesis: https://espace.cern.ch/GBT-Project/GBTX/Publications/Forms/AllItems.aspx
- Csaba Soos and Paulo Moreira slides @ TWEPP09
- GBT project web site: http://cern.ch/proj-gbt

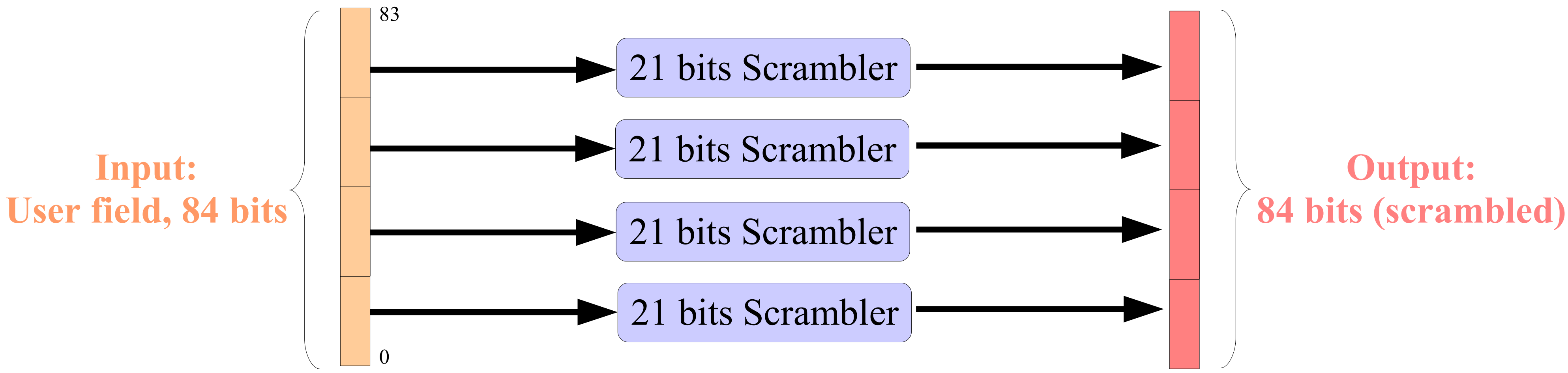
**Thanks for your attention**

# Back up slides

# Scrambler

- ***Aim of the scrambling:***
  - Reduce the occurrence of long sequences of '0' (or '1') only;
  - Achieve a good DC-balance

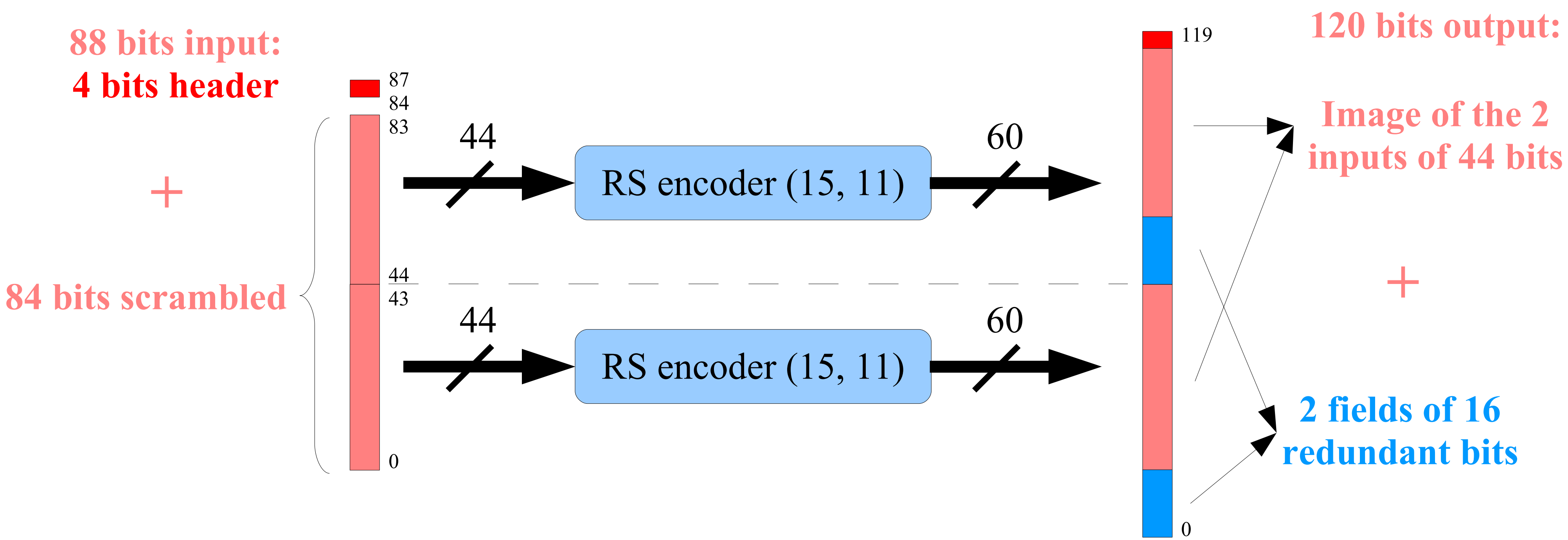
- ***Scrambler characteristics:***
  - Doesn't add any redundancy/overhead
  - Has a latency of 1 clock cycle
  - Generates a pseudo random signal



# Reed Solomon encoder

- ***Aim of the Reed Solomon encoding:***
  - Protect the data against transmission errors

- ***Reed Solomon encoder characteristics:***
  - Composed of 2 RS encoder (15,11), symbol = 4 bits
  - Only combinatorial logic
  - Allows to correct up to 2 symbols (8 consecutive bits for example)



# Interleaver

- ***Aim of the interleaving:***
  - Improve the capacity of the code to correct long burst of errors

- ***Interleaver characteristics:***
  - Interleaving made at a symbol level (4 bits)
  - Only "routing", no clock cycle
  - Increases the code correction capability from 2 to 4 consecutive symbols (16 consecutive bits) without increasing overhead