Avalon Microelectronics

ECOC 2009

Comparison of current FPGA technology for 100G

September 22, 2009

Presentation Overview

- Background
  - Typical Long Haul Optical System and purpose of FEC
  - Quick FEC primer
  - Why use FPGAs
- Algorithm Review
  - G.975.1 Annex 4
  - Dual BCH
  - Dual BCH Product Code
- Comparison of FPGA architecture
  - Xilinx versus Altera

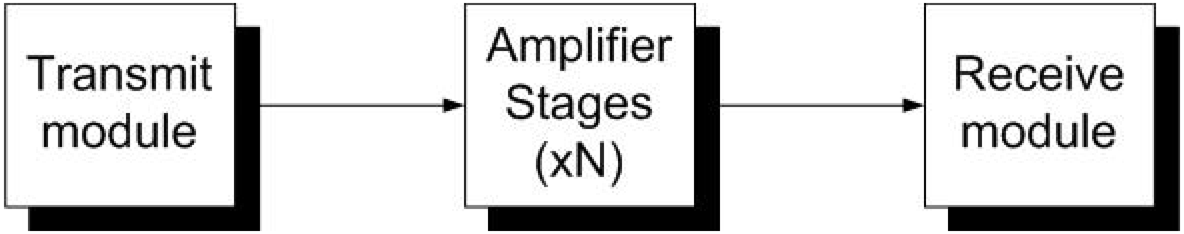
Presentation Overview

- Impementation of the Algorithms
  - Xilinx Virtex 5, Virtex 6; Altera Stratix 4
- Results of Case Study
  - Xilinx Virtex 5, Virtex 6; Altera Stratix 4
  - Place and routed designs
- Further Considerations
  - Algorithm Type
  - RAM size and efficiency
  - Tool requirements
  - Device Considerations

Typical Long Haul Optical System

Background

- Typical system has at least one amplifier stage



- Purpose of FEC is to mitigate bit errors due to additive noise, not for dispersion compensation or other error models

Quick FEC primer

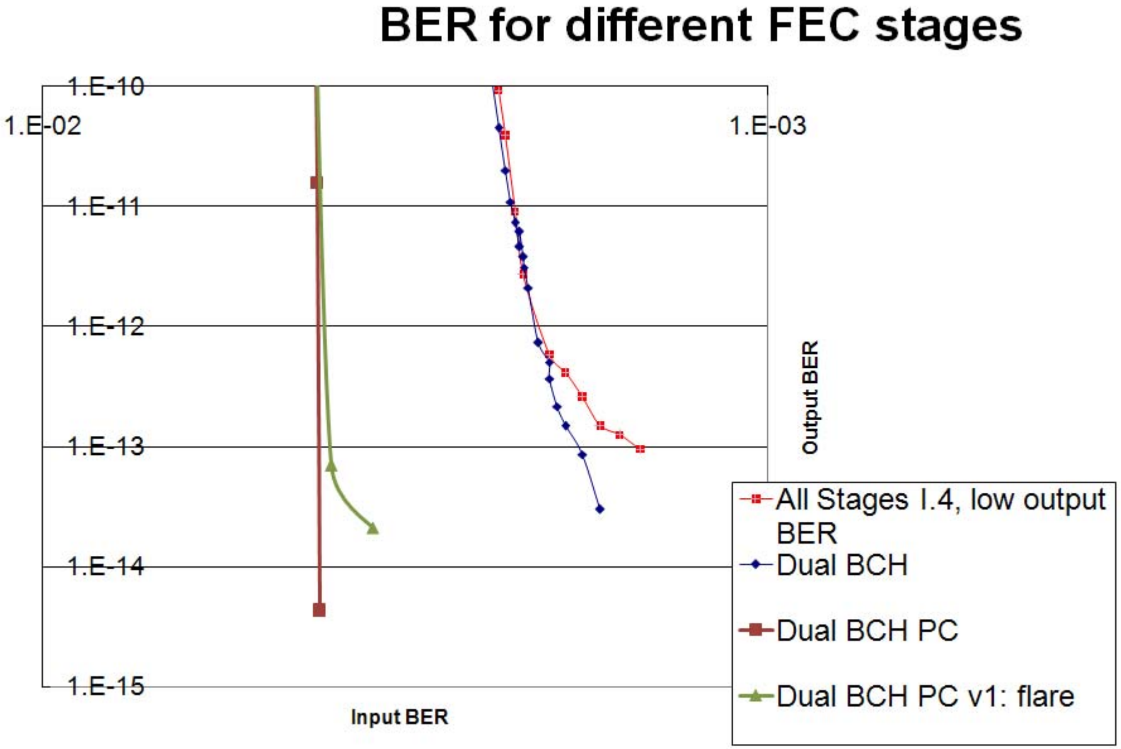
Background

- Forward Error Correction is a process of adding overhead to a transmitted signal S(x) so the receiver has a better chance of recovering the transmitted S(x)
- Net Electrical Coding Gain (NECG) is the measurement used to calculate the gain of an FEC that includes the overhead penalty
- Flaring is an undesired effect in FEC codes that is the result of a sequence of errors that the FEC is not able to correct
    - Can be caused by interaction between embedded FEC codes

Why Use FPGAs?

Background

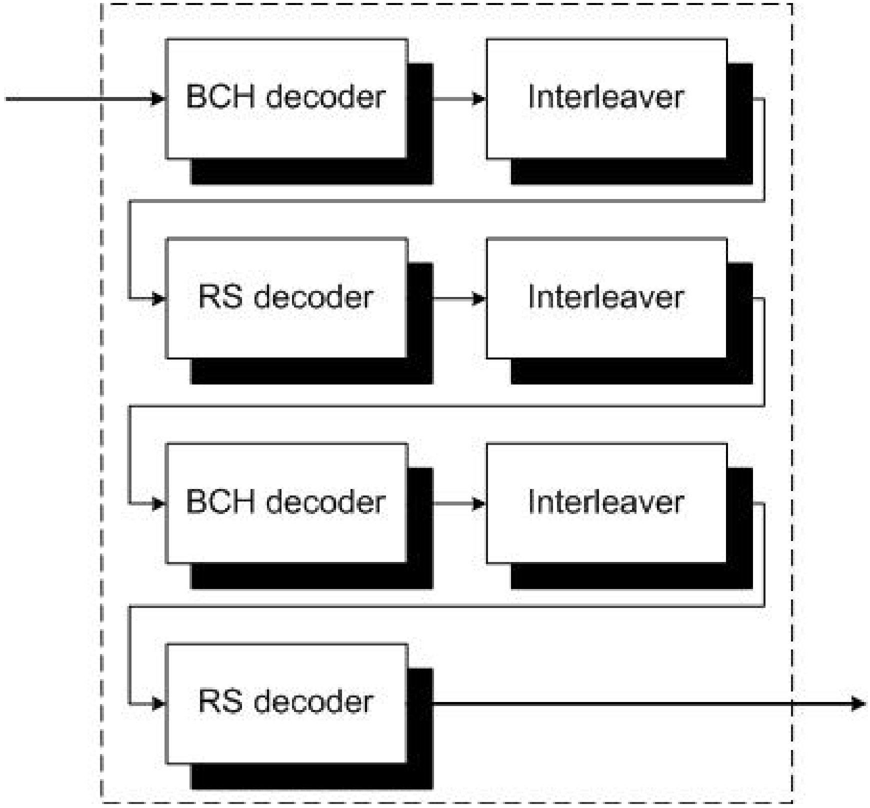
- Low output Bit Error Rate (BER) data points are difficult to simulate due to simulation time in the region of interest
    - Typical regions of interest: 10^{-12} to 10^{-15} output BER
- FPGAs allow prototyping of different FEC algorithms and observation of their effects at low output BER



G.975.1 Annex 4

Algorithm Review

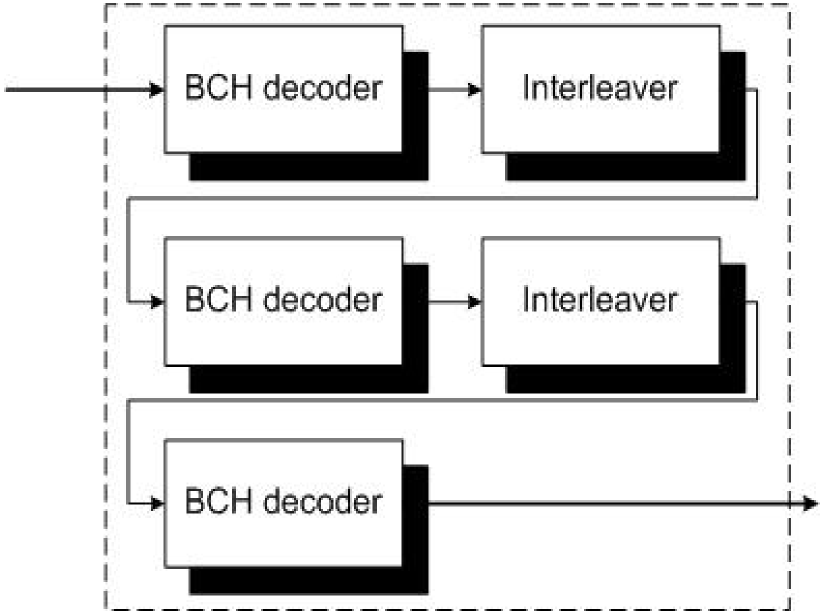
- Popularized by AMCC's Rubicon and follow on devices
- 8.3 dB NECG



Dual BCH

Algorithm Review

- Smaller than I.4
- Better suited for AWGN style error model
- Still subject to flaring at low output BER
- 8.3 NECG

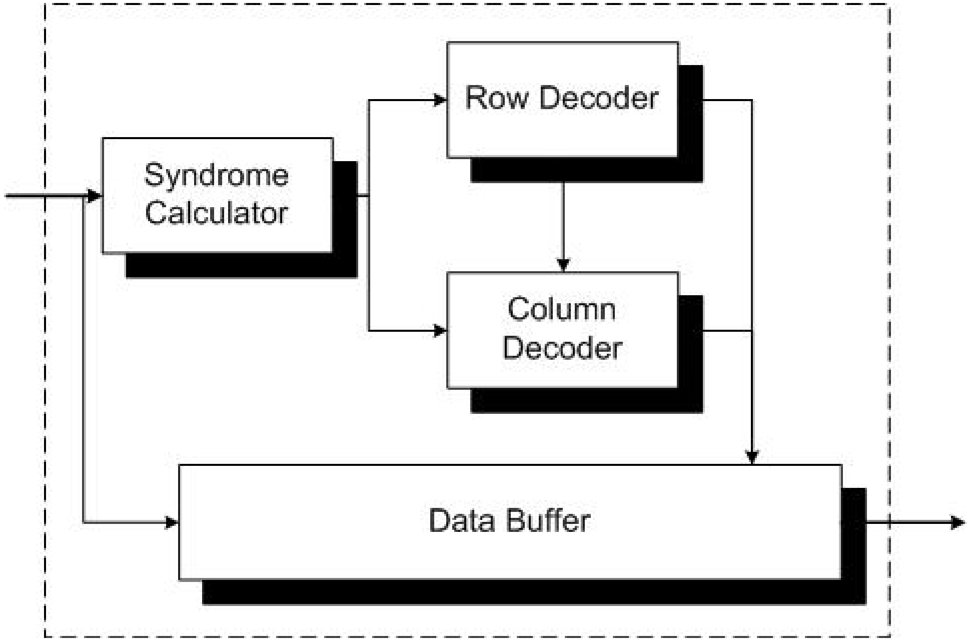


Dual BCH – product code

Algorithm Review

- Smaller than both previous algorithms
- Not subject to flaring
- 9.2 dB NECG

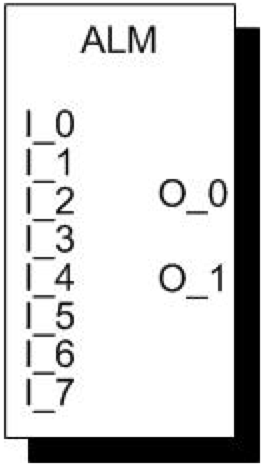
- The development of this code would only be possible with FPGAs due to effects at low output BER



Xilinx versus Altera

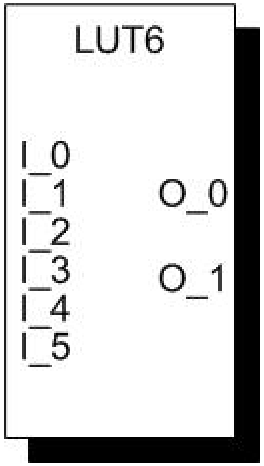
Comparison of FPGA Architecture

- Many differentiators (I/O, RAMs, power) but for logic the comparable element is the Look Up Table (LUT) structure



- **Altera ALM highlights**
- Highly configurable
- Combination of standard LUT and muxes

- **Xilinx LUT6 highlights**
- High speed
- Small footprint
- Third party synthesis does a good job



- There is no straight apples-to-apples comparison, only synthesized circuits can be used to compare

Xilinx Virtex 5, Virtex 6; Altera Stratix 4

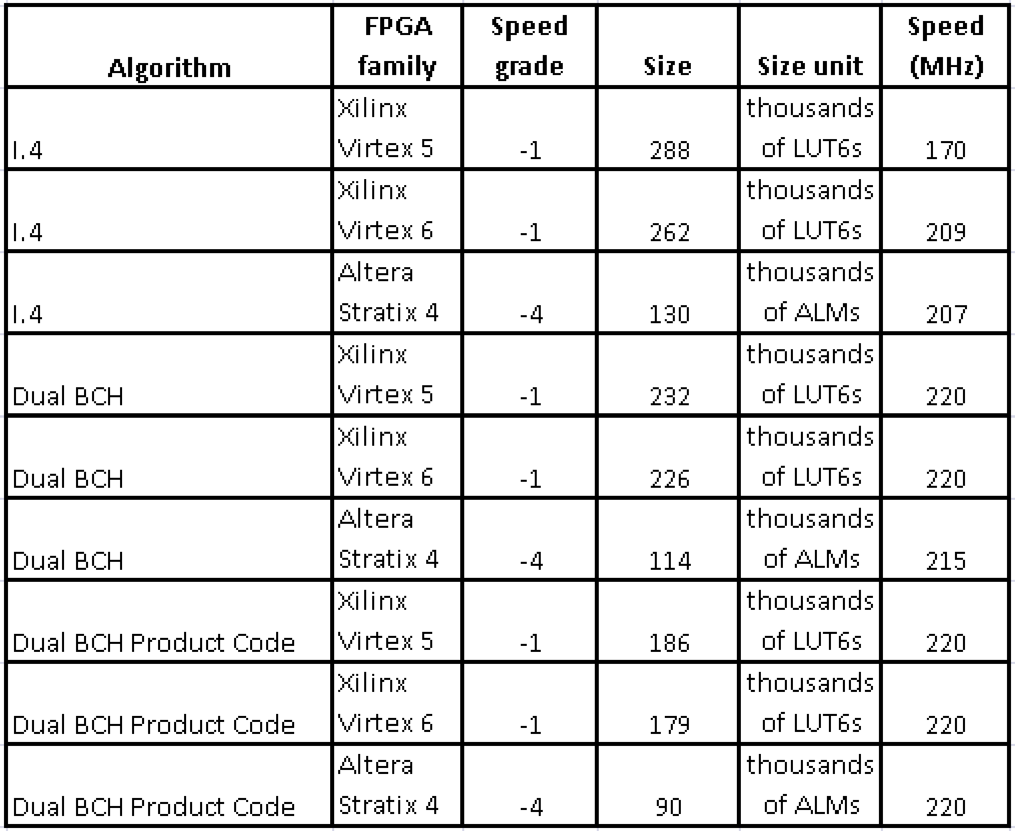
Implementation of the Algorithms

- Implemented using best available methodologies
- Virtex 5:
  - Synthesis: Synopsys Synplify Pro 8.8.0.4
  - Place and Route: Xilinx ISE 10.1.3
- Virtex 6:
  - Synthesis: Synopsys Synplify Pro c200906
  - Place and Route: Xilinx ISE 11.2
- Stratix 4:
  - Synthesis: Quartus 9.0 Service Pack 2
  - Place and Route: Quartus 9.0 Service Pack 2
- Various combinations of tools will produce different results
- Various options within the tools will produce different results

Xilinx Virtex 5, Virtex 6; Altera Stratix 4

Results of Case Study

- Lowest speed grade for all families



Algorithm	FPGA family	Speed grade	Size	Size unit	Speed (MHz)
I.4	Xilinx Virtex 5	-1	288	thousands of LUT6s	170
I.4	Xilinx Virtex 6	-1	262	thousands of LUT6s	209
I.4	Altera Stratix 4	-4	130	thousands of ALMs	207
Dual BCH	Xilinx Virtex 5	-1	232	thousands of LUT6s	220
Dual BCH	Xilinx Virtex 6	-1	226	thousands of LUT6s	220
Dual BCH	Altera Stratix 4	-4	114	thousands of ALMs	215
Dual BCH Product Code	Xilinx Virtex 5	-1	186	thousands of LUT6s	220
Dual BCH Product Code	Xilinx Virtex 6	-1	179	thousands of LUT6s	220
Dual BCH Product Code	Altera Stratix 4	-4	90	thousands of ALMs	220

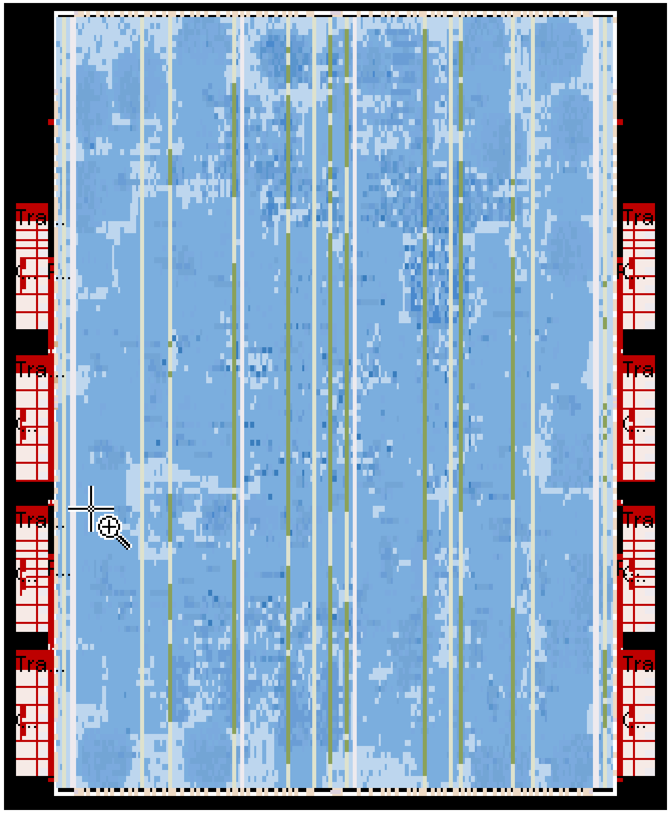
- Rate Calculation:
- 111.8Gb/s
  - OTU4 specification
- 512 bit wide datapath

- Implies 218.4 MHz core operation

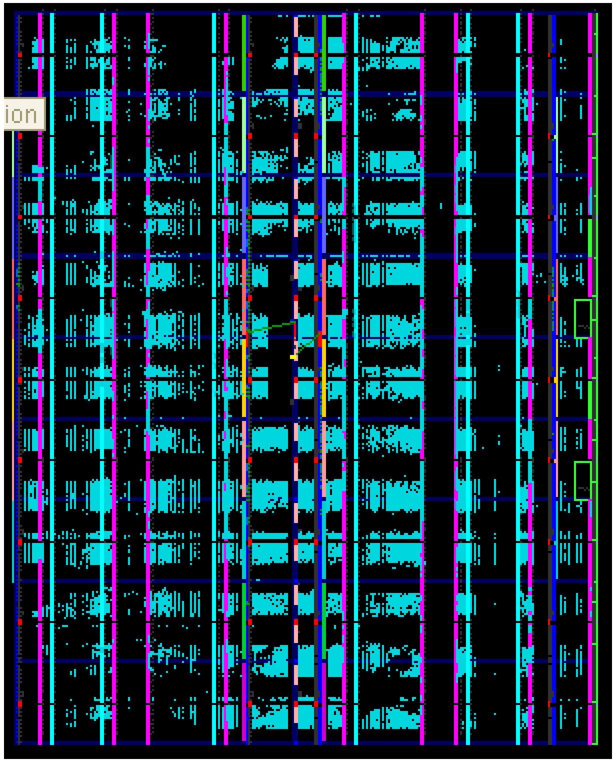
Place and Routed Designs

Results of Case Study

Altera Stratix4



Xilinx Virtex 6



Algorithm Type

Further Considerations

- The error model is AWGN with an input BER of less than 1 x 10^{-2}
- I.4: BCH, RS
  - BCH is a good choice for the inner, or initial, decoder
  - The RS is a poor choice for the outer, or secondary, decoder as the resulting error signature is not bursty
- Dual BCH
  - Having two large BCH codes complements the error model
  - Smaller size helps FPGA implementation
- Dual BCH Product Code
  - Smaller error correction capability per code
  - Smaller codes allow for multiple iterations between decoders
  - Almost a dB of gain while significantly smaller than previous codes

RAM size and efficiency

Further Considerations

- RAM is the second most significant factor on FECs in FPGAs behind LUT utilization
- RAMs pose a difficult routing breakout problem in FPGAs and cannot be practically 100% utilized at these speeds
- Choosing FEC codes and interleaving schemes that effectively use FPGA RAMs are doubly useful:
  - Increase the gain of the algorithm
  - Reduce the difficulty for the Place and Route tools to complete the design

Tool requirements

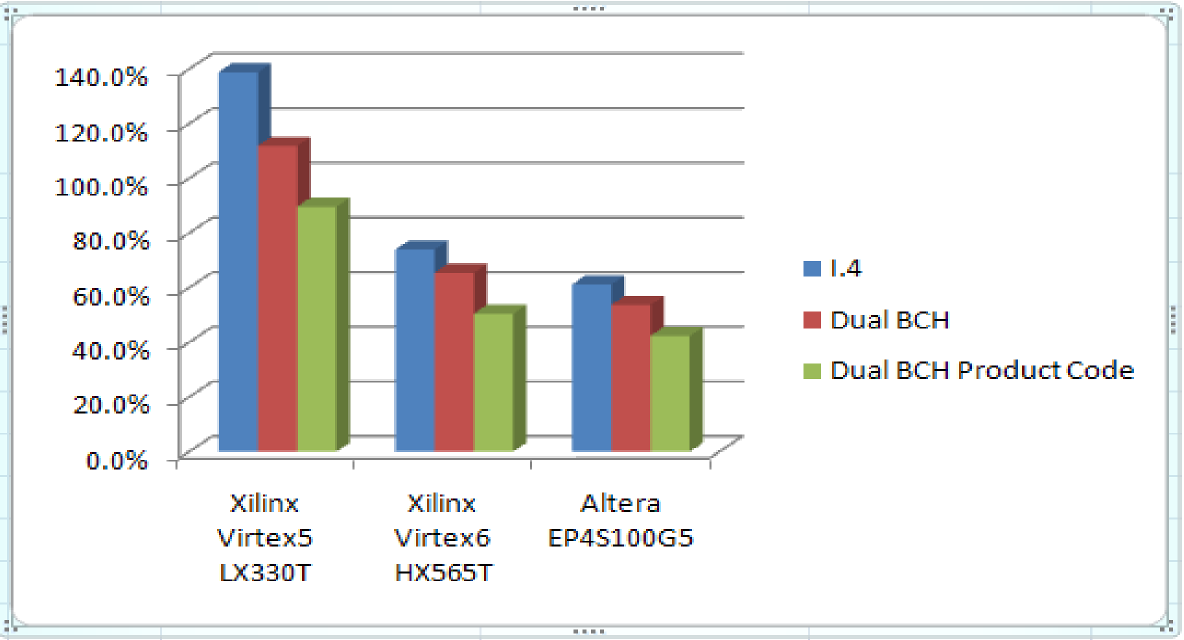
Further Considerations

- RAM floorplanning was required for all implementations
- 64 bit linux machines were required
  - One tool took more than 10GB of memory in one stage
- FPGA tools have various settings (effort, seed, etc) that greatly affect performance
- Different versions of tools utilize different algorithms; some algorithms can be more effective than others at increasing speed or decreasing size of FEC implementations
- FPGAs require more ASIC like place and route tools

Device Considerations

Further Considerations

- RS decoder for I.4 had reduced performance because of the symbol nature of an RS over BCH
- Implementation requires largest currently available FPGAs
- Only 40nm FPGAs are capable of single device 100G operation



Summary

- FPGAs are useful for prototyping FEC performance
- Use various tool combinations to achieve best performance
- It is possible to optimize FEC code for FPGAs
- 100G high gain FEC can be achieved using current FPGA technology

Avalon Microelectronics

Thanks!