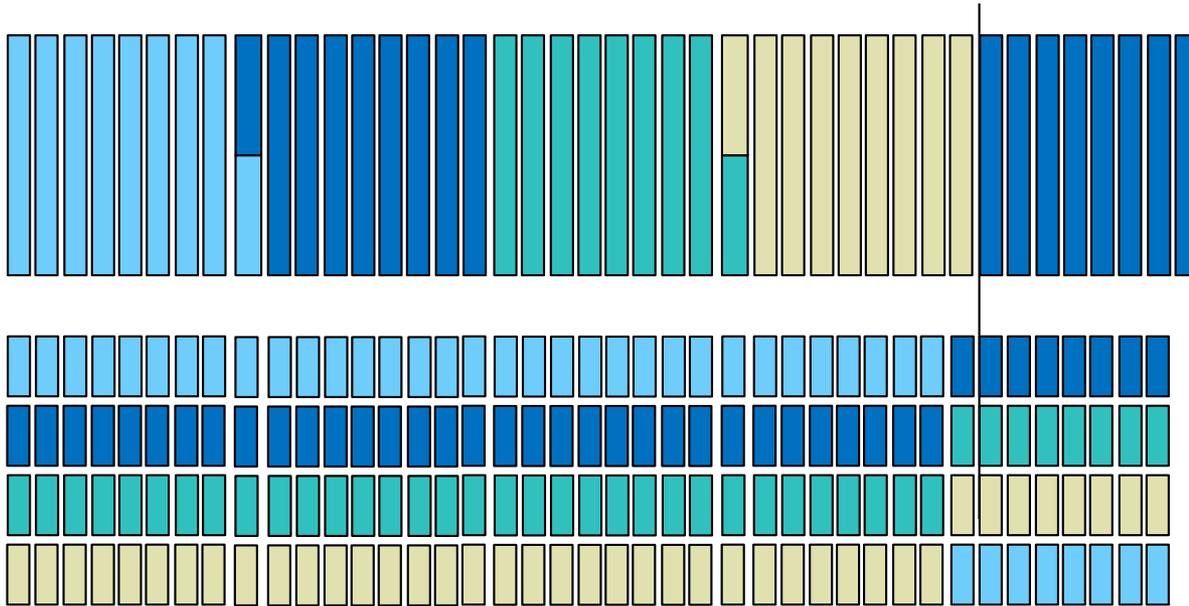


400GE FEC Breakout Architecture Analysis

Martin Langhammer
Altera Corporation

IEEE802.3bs 400GE Task Force
July, 2015

1x400GE ASIC Breakout Problem



1x400G KP4 RS(544,514) requires 8.5 clocks per codeword @ 64 symbols per clock
4x100G KR4 RS(528,514) requires 33 clocks per codeword @ 4x16 symbols per clock

1x400G FPGA Breakout Problem

- ◀ There is none
 - Just reconfigure device with required FEC
- ◀ Still important for FPGA market
 - Hard embedded FECs

Method

◀ Build ASIC Decoder

- 64 symbol width
- 1x400G, 8.5 clocks per KP4 codeword
- ASIC pipeline depth for correct area model

◀ Fit into FPGA

- Only relative sizing needed for options
- Large reported area variance between ASIC cores anyways

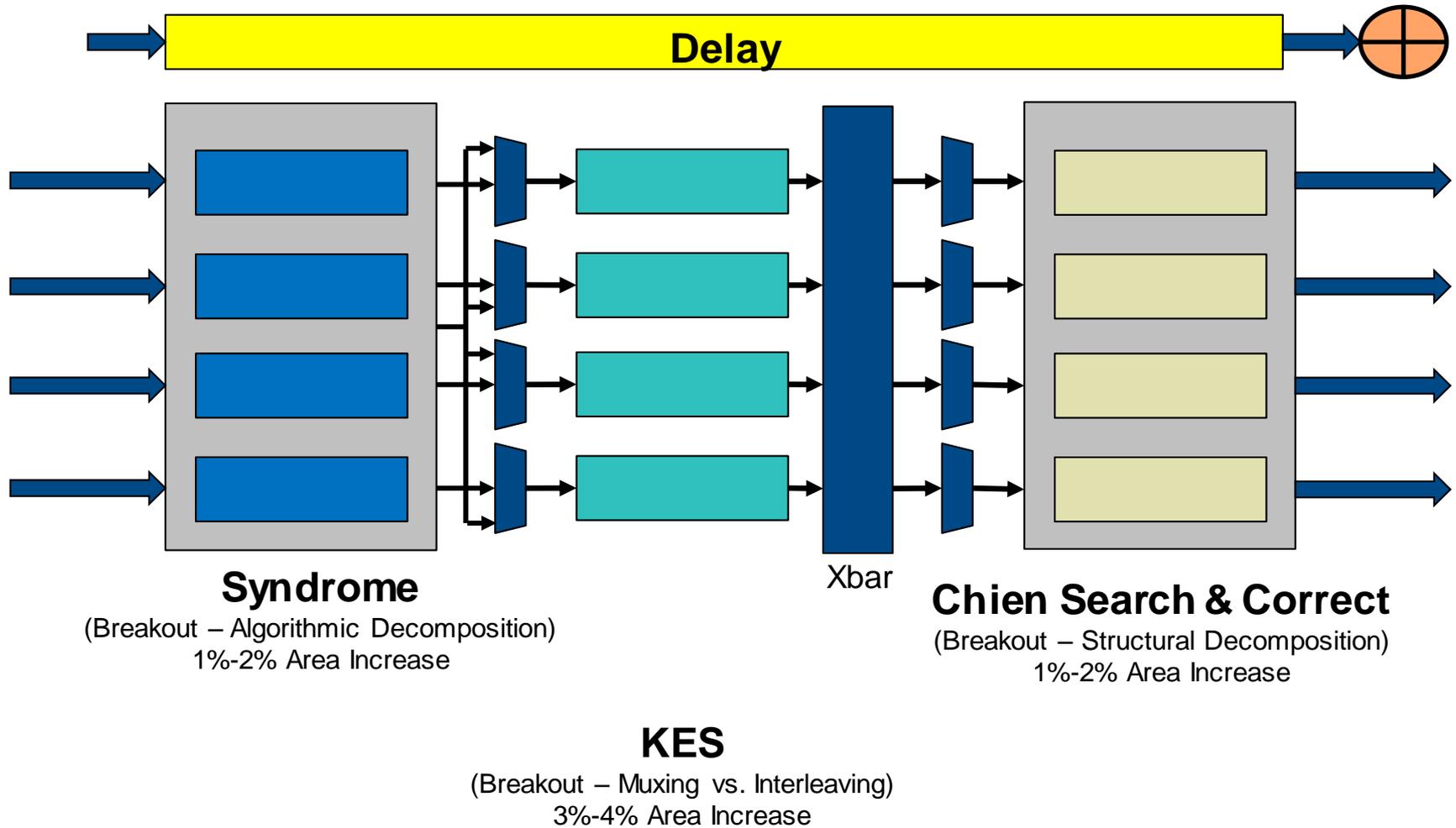
Results

- ◀ 1x400G KP4 ASIC: 55645
- ◀ 1x400G KP4 & 4x100G KR4 ASIC: 61518
- ◀ 11% area increase
- ◀ Recoverable to 5% area increase
 - Some duplicated calculations due to schedule of original work
- ◀ Increase Analysis : 5%
 - Function Logic Increase : 1%-2%
 - Muxing, Staging - 3%-4%

◀ Caveats

- *Treat size as dimensionless number*
 - ◀ *Don't compare to FPGA results – very different component structures*
- *Preliminary Design: optimization possible, especially latency*

Breakout Architecture



Architecture Notes

◀ Direct Breakout Supported

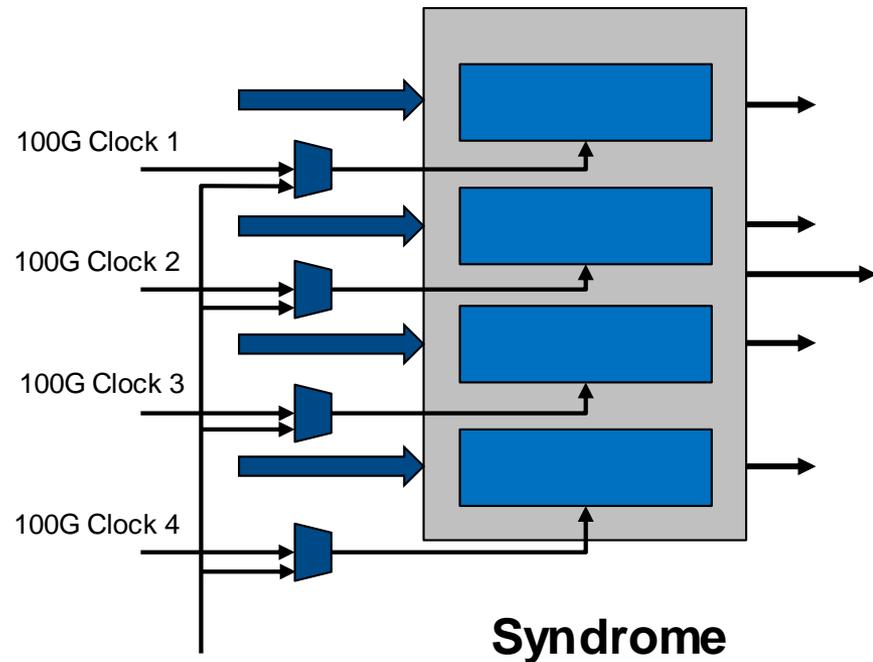
- Based on 1x400G Core
- Dynamic Switching between 1x400G and 4x100G
 - ◀ Using the same datapath elements
- No TDM
 - ◀ One datapath
- No duplicated engines
 - ◀ Simpler design, may add 10%-20% area
- No additional memory
 - ◀ All data, 1x400G or 4x100G sent to core as soon as it is received
- No material latency change ($\ll 5$ clocks)

◀ Currently Breakout Lanes are in Lockstep

- Straightforward to make completely independent

Multi-Clock Considerations

- ◀ Clock muxes at top of tree, similar to muxes used for configurable channel bonding or DFT scan
- ◀ Timing constraints applied using case analysis or overlapping clocks
- ◀ Shared clock is synchronous across the trees
- ◀ Individual clocks are mutually asynchronous
- ◀ Shared clock and individual clocks are physically exclusive
- ◀ All clocks use same period (nominal rate + maximum ppm offset)
- ◀ All clocks must account for mux insertion delay (generated clocks with mux input as source)



The 1x400G syndrome block can be algorithmically decomposed with ≈ 0 area cost (other blocks are decomposed by muxing or structurally). In 1x400G mode, only one clock is used, in 4x100, each decomposed lane can support a separate clock.

Latency

- ◀ Similar to reported individual core results
- ◀ Proportional to KES architecture
 - Codeword input time constant for any architecture
- ◀ 2 KES ASIC options
 - 1 clock per check symbol (KP4 = 30 clocks, KR4 = 14 clocks)
 - ◀ Simplest, lowest latency
 - 2 clocks per check symbol (KP4 = 60 clocks, KR4 = 28 clocks)
 - ◀ Easier to for timing closure, longer latency
- ◀ Latency : 2 clock KES
 - 1x400G KP4 RS(544,514) = 137ns
 - 4x100G KR4 RS(528,514) = 120ns
- ◀ Latency : 1 clock KES
 - 1x400G KP4 RS(544,514) = 90ns
 - 4x100G KR4 RS(528,514) = 74ns

Un-optimized latencies
Optimized latencies 10ns-15ns less

Effect of Error Marking

- ◀ Will not affect this analysis
- ◀ If no acceleration (parallel Chien search), 1x400G actually faster than 4x100G
 - 12.5ns vs 37.5ns increase
- ◀ Same tradeoffs in terms of parallelization of acceleration can be made in both cases.

Other considerations

◀ Wiring and Mux density

- Will be increased for breakout support
- Effect TBD

◀ Timing closure

- 1x400G architecture possibly more difficult than 4x100G
 - ◀ ASIC or FPGA
 - ◀ But may be same for system POV
- Breakout support will further increase bus length and width

◀ Other breakout possibilities

- 4x100G KP4 may be possible using very little additional hardware
 - ◀ Estimated final latency 1x400G KP4: 80ns
 - ◀ Estimated final latency 4x100G KP4: 117ns

Encoder

- ◀ Not strictly part of this analysis, but....
- ◀ Breakout not directly supportable KP4 \Leftrightarrow KR4 with known methods
 - By any method or architecture
- ◀ 4x100G KP4 \Leftrightarrow 100G KP4 trivial by definition
- ◀ 1x400G KP4 \Leftrightarrow 100G KP4 relatively straightforward
 - Very low latency of encoder allows for inexpensive (latency) implementation of TDM
 -

Conclusions

- ◀ 1x400G and 4x100G breakout directly supportable
 - Using a single core architecture
 - 5%-10% larger than 1x400G monolithic
 - ◀ Based on both achieved results and architectural analysis

Thank You