The SFRA: A Corner-Turn FPGA Architecture

Paper by Nicholas Weaver
John Hauser, John Wawrzynek
Berkeley CS, FPGA'04
ACME Seminar, January 23, 2004
Mark L. Chang

Conventional FPGAs

- Conventional FPGAs do not pre-determine clock frequency
- Interfacing with microprocessor requires compatible clocking schemes

Fixed-Frequency FPGAs

- Operate at a set clock rate regardless of configuration
- Easier to integrate with other computation blocks
- Higher clock rates
- Good for computations that can be pipelined!

Interconnect and Routing

- Interconnect delay dominates conventional FPGAs
- Fixed-frequency generally requires pipelining the interconnect and routing structures
- Previous fixed-frequency arrays
  - Difficult placement problems
  - Highly restrictive interconnect topologies

Other Fixed-Frequency FPGAs

- Garp
  - MIPS processor with reconfigurable coprocessor
  - Interconnect fabric supported only limited connectivity
- RaPiD / PipeRench
  - Coarse-grained array of functional units
  - Explicitly pipelined interconnect

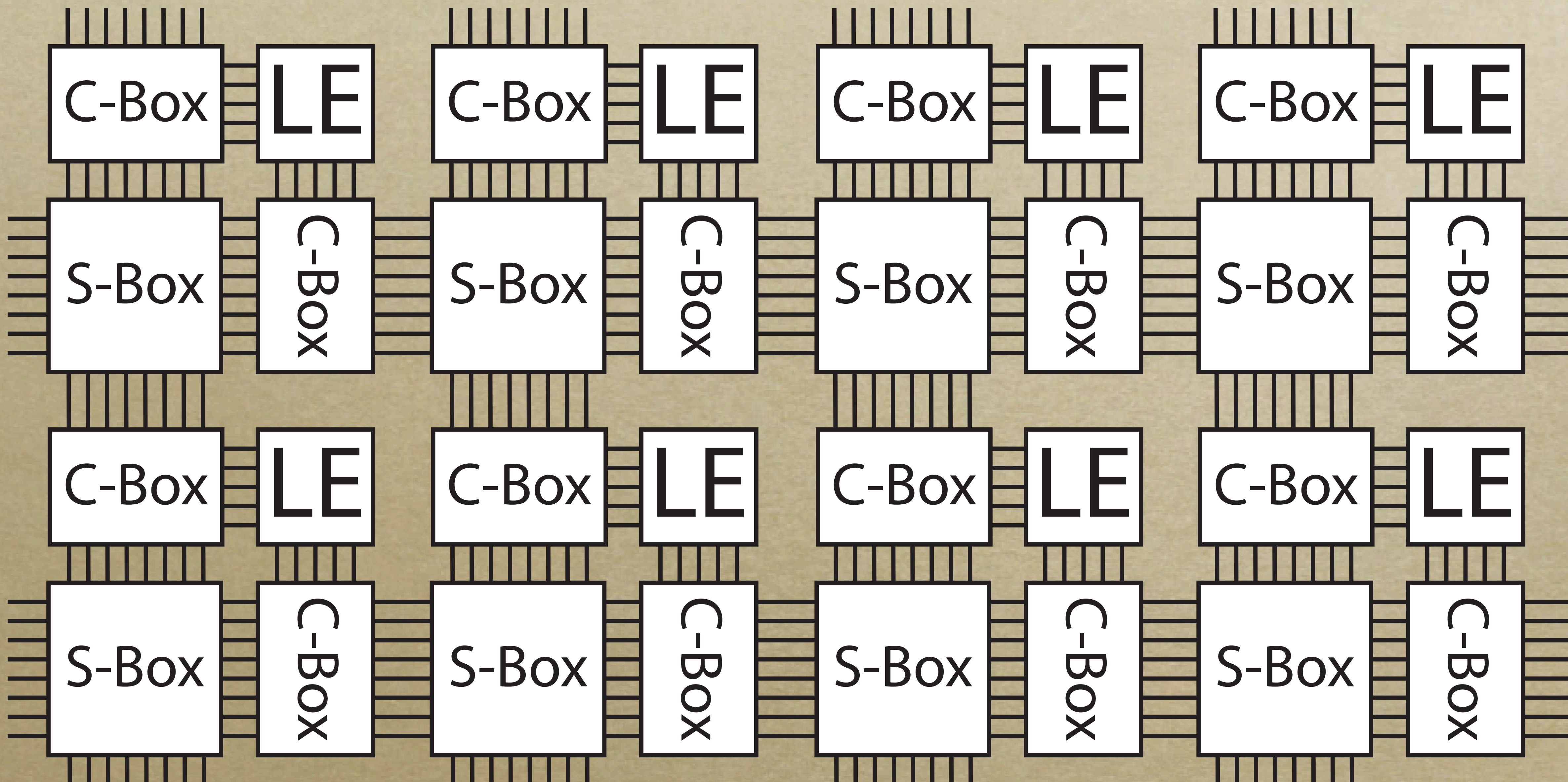
More Fixed-Frequency FPGAs

- HSRA
  - Pipelined H-tree for the routing structure
  - Retiming chains, programmable delay shift registers on all inputs to balance delay
- Introduces a significant placement problem
  - Not Manhattan
  - Recursive biparitioning for placement
  - Good for random logic applications
  - Bad for datapath-oriented circuits

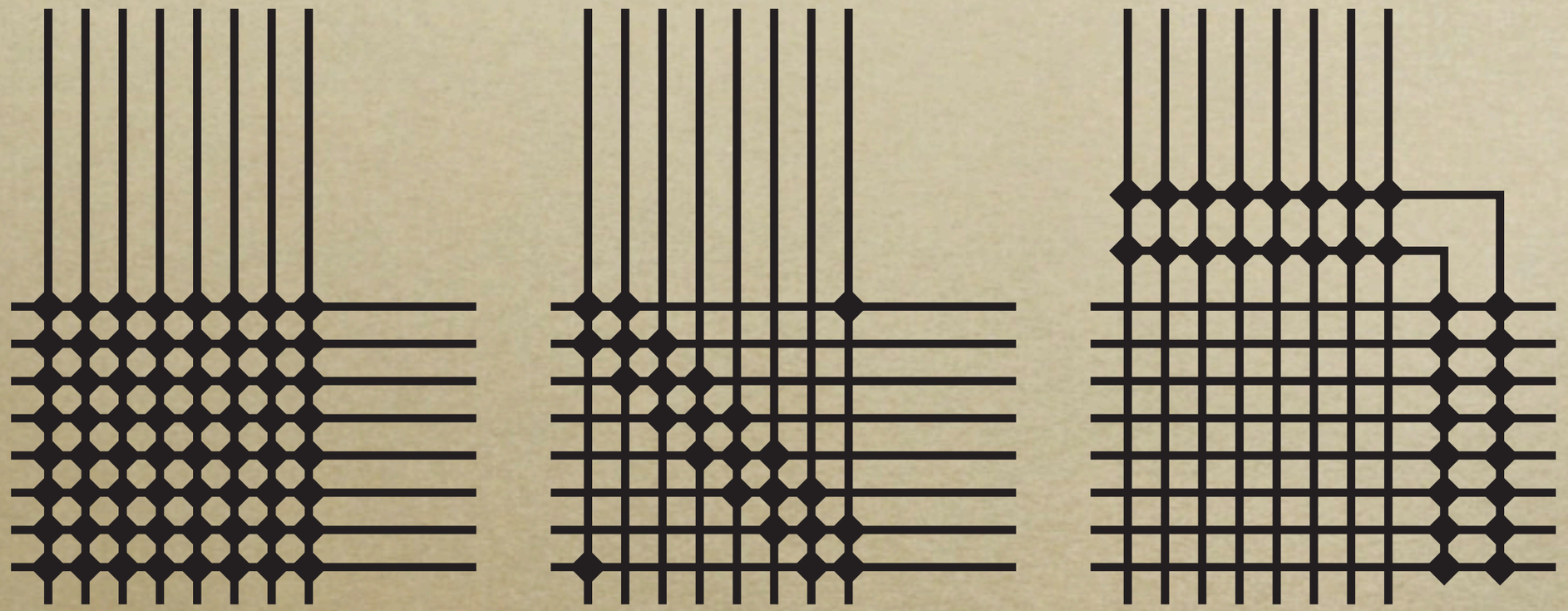
The Problem

- Want to leverage conventional placement and synthesis tools and techniques
- Interconnect must be pipelined
- But adding pipelined switches to a conventional FPGA would be impractical
  - Too many registers!
  - Could complicate the routing problem if switches are not fully registered

The FPGA Landscape



Corner-Turn Interconnect



- Any-to-any connectivity for very signals
- Maintains Manhattan placement
- Can be efficiently pipelined

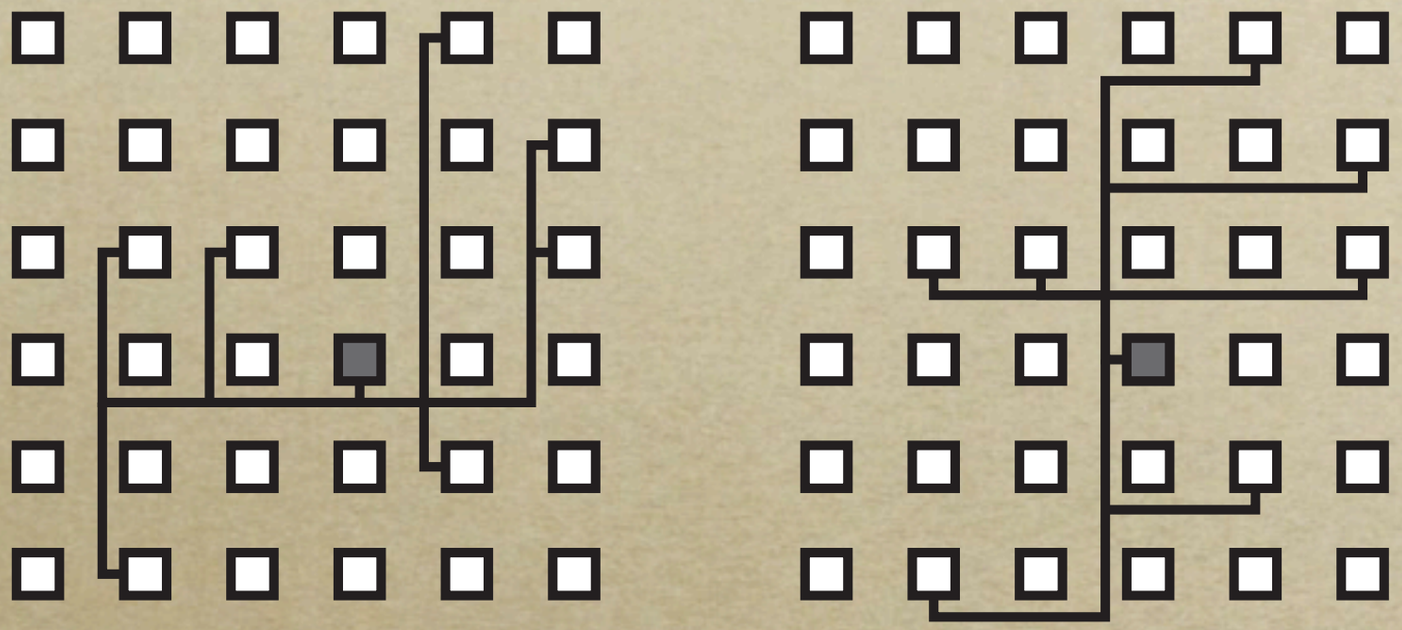
Routing Corner-Turn FPGAs

- Global Routing
  - Assigns signals to particular channels and turns
  - Attempts to minimize the number of turns taken by each signal
  - Done in 5 stages
- Detailed routing
  - Performs wire assignment within each channel
  - Greedy channel-packing technique (O(N logN))

5-Stage Global Routing

- 1 - Direct routing
  - Route all no turn nets which can be routed
- 2 - Fanout routing
  - Attempt to route large fanout nets to maximize sharing of turns
  - Sort all remaining nets by degree of fanout
  - Starting with highest, route all nets with fanout > 4

Fanout Routing



- Possibly route horizontally or vertically
- Use orientation that uses fewer turns
- Leave unroutable nets to later routing stages
- Lock down routed nets

5-Stage Global Routing

- 3 - Pushrouting
  - Consider all nets as individual point-to-point signals
  - Prioritize shorter nets and nets on critical paths
  - For each net, only two possible one-turn routes
  - If either route is free, assign the turn to this net
  - If not free, depth-first search routed nets that use the two turns to see if an adjustment can be made to allow routing of the net

5-Stage Global Routing

- 4 - Zig-zag routing
  - Remaining nets are examined to find possible two-turn routes
  - Prioritize longer nets, as they have more possible turns along their route

5-Stage Global Routing

- 5 - Rip-up-and-reroute
  - We want to route any remaining nets
  - For each unrouted net, determined if it can be pushrouted or zig-zag routed
  - If not, examine the two one-turn route switches
  - Breath [sic] first search on all nets using these turns
  - Select on randomly (weighted toward longer nets)
  - Rip-up-and-reroute

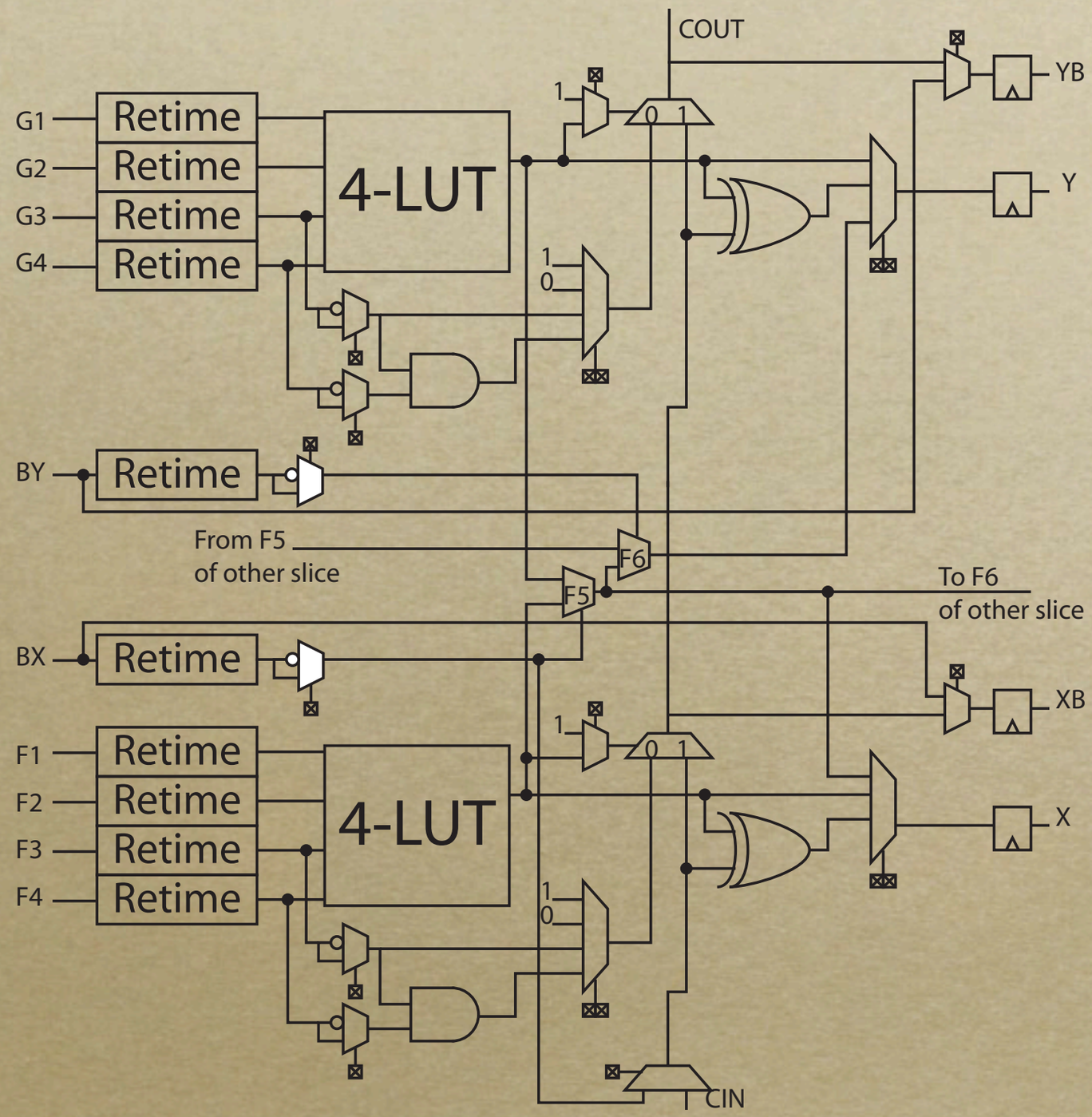
Retiming

- In a fixed-frequency FPGA we must either
  - Restrict the user's designs to meet the array's pipeline requirements
  - Automatically transform designs to meet the array's constraints
- Feed-forward designs can be repipelined
- Feed-backward designs must use C-Slow

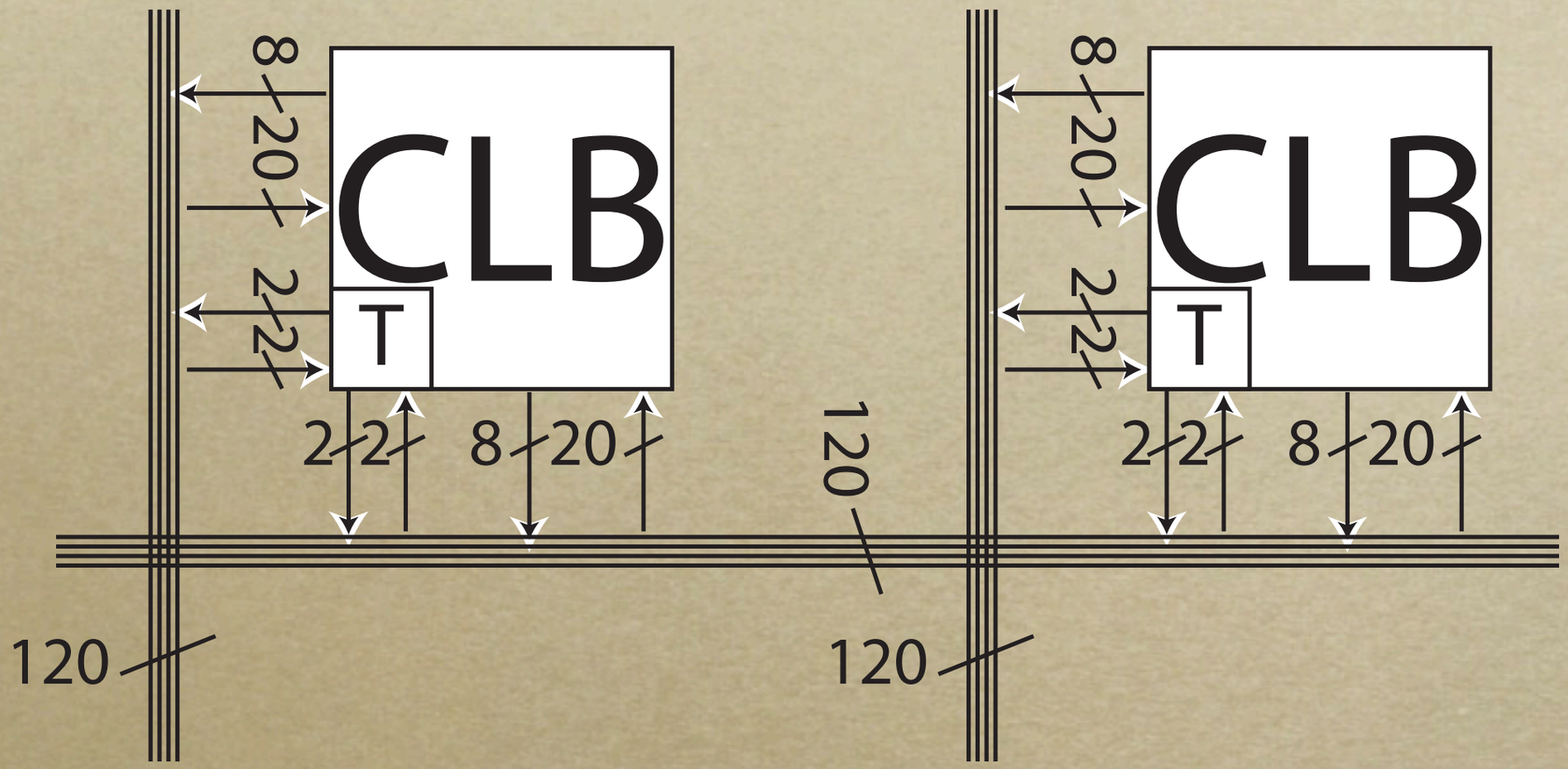
The SFRA Architecture

- Generally compatible with Xilinx Virtex
- Designs must use a single global clock
- Resets and clock enables must be expressed as combinational logic rather than using primitives
- Cannot use LUTs as RAMs or SRL16s

The SFRA Slice

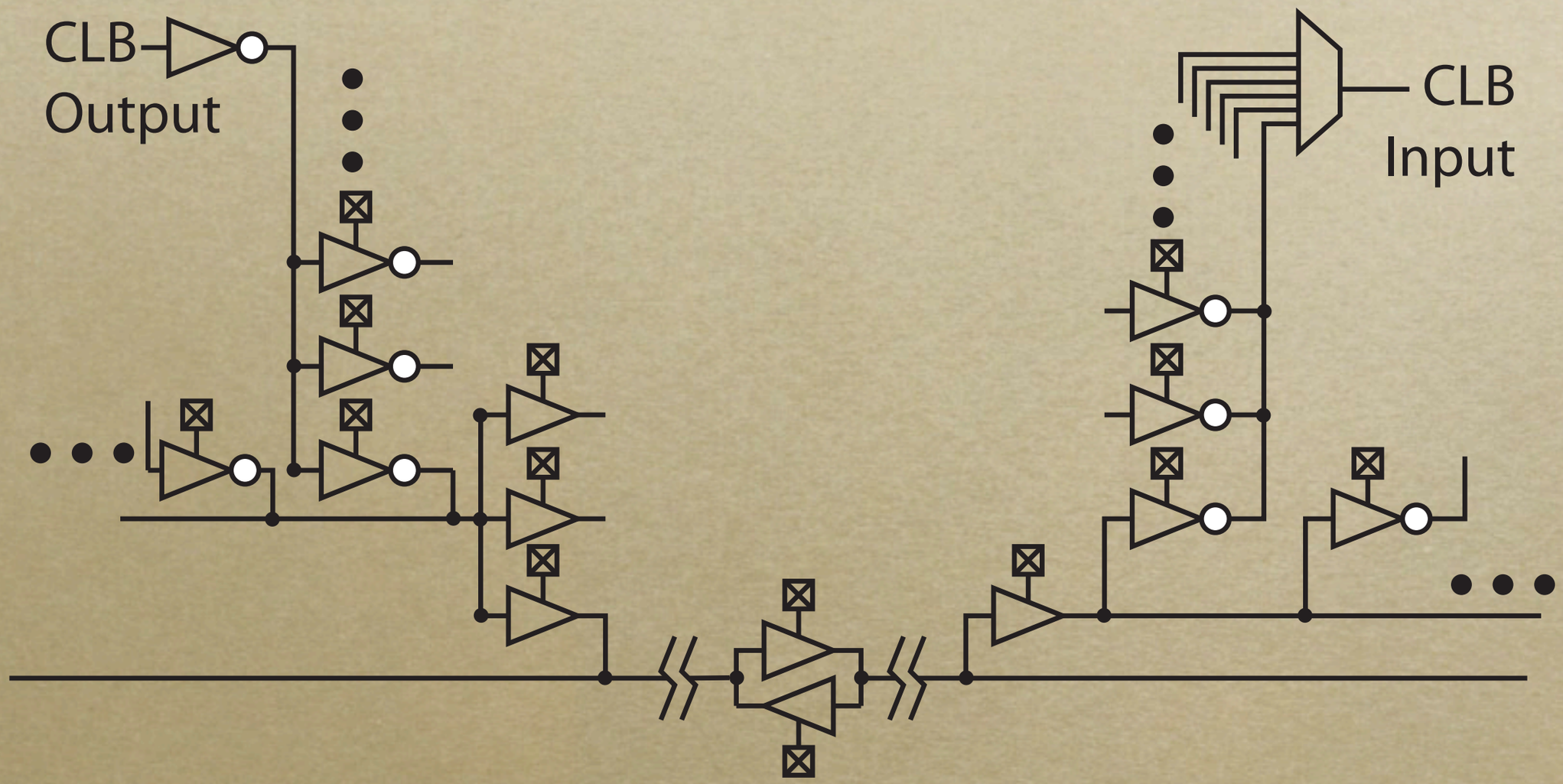


The SFRA Interconnect

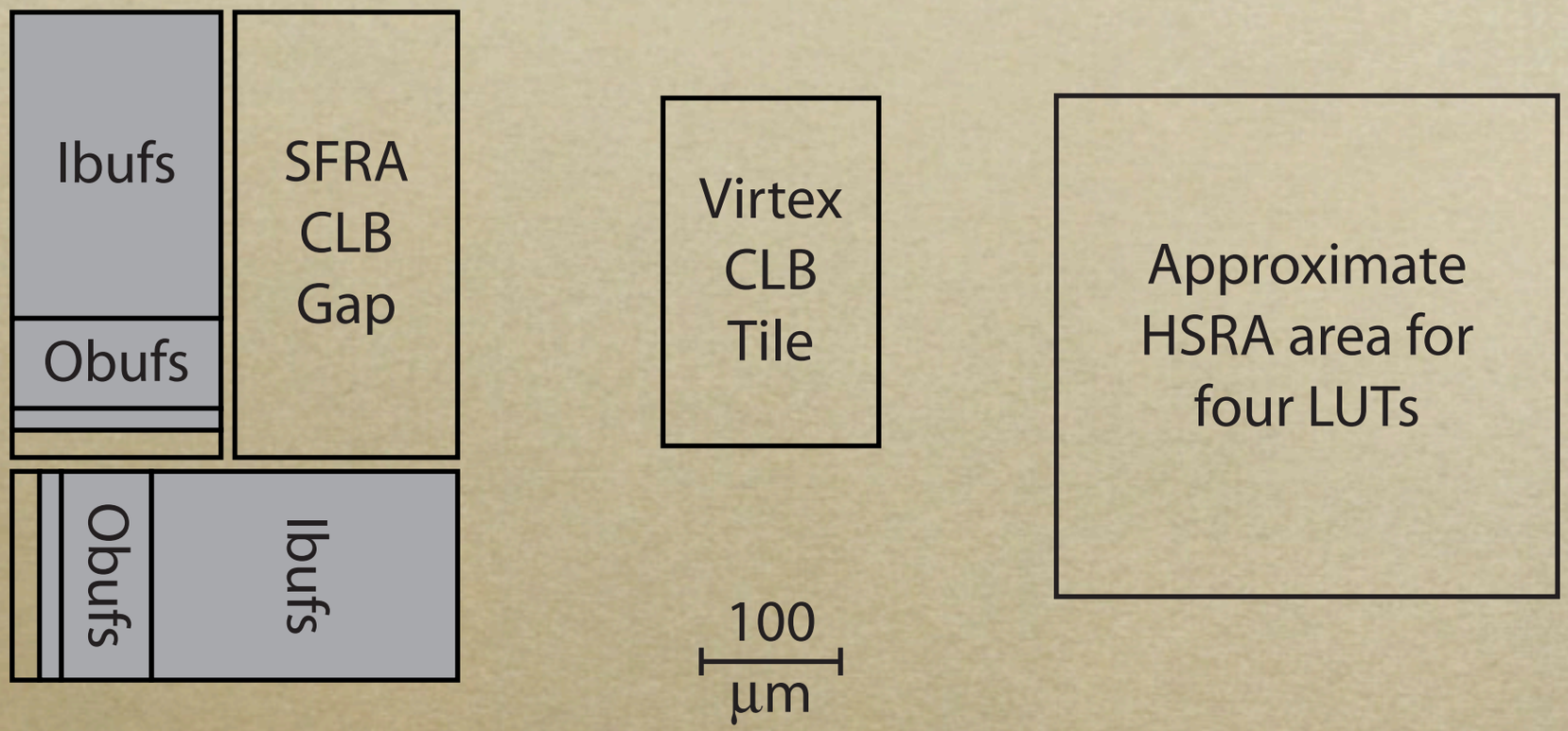


- I/O on any horizontal and vertical wire
- Every three CLBs, bidirectional buffer break
- Every nine CLBs, bidirectional register break

The SFRA Interconnect



The SFRA Tile



- 180nm process, 160,000 square microns
- 3.9x larger than Xilinx Virtex E, 1.5x HSRA
- 300MHz in simulation

The SFRA Tool Flow

- Begins with Xilinx toolset
  - Design entry
  - Placement
  - Mapping
- Custom tools
  - Initial retiming to find critical path
  - Global routing
  - Detailed routing
  - Retiming

Evaluating the SFRA

- Benchmarks
  - AES encryption
  - Smith/Waterman sequence matching
  - Synthetic datapath (core of 32-bit CPU)
  - LEON 1 microprocessor core (SPARC)
- Xilinx benchmarks done on a Spartan II @ 250nm
  - Clock rates scaled by 1.4

Evaluating the SFRA

- Benchmarks placed using Xilinx tools
- Maximum effort for all tools
- C-Slow retimed Xilinx implementations
- Used hand- and automatically-placed circuits

Tool Runtime Results

Benchmark	Xilinx Routing Time	Custom Xilinx Retiming Time	SFRA Routing Time	SFRA Toolflow Time
AES (hand placed)	405 s	73 s	2 s	6 s
AES (autoplaced)	542 s	77 s	2 s	6 s
Smith/Waterman (hand placed)	104 s	178 s	2 s	8 s
Smith/Waterman (autoplaced)	102 s	169 s	2 s	9 s
Synthetic Datapath (hand placed)	284 s	47 s	2 s	5 s
Synthetic Datapath (autoplaced)	307 s	50 s	2 s	6 s
LEON (autoplaced)	432 s	hours	11 s	62 s

SFRA Performance Results

Benchmark	Adjusted Xilinx Clock Rate	SFRA C-slow Factor	Expected Speedup for 300-MHz SFRA	Area-Normalized Speedup
AES (hand placed)	67 MHz	24-slow	4.4x	1.1x
AES (autoplaced)	67 MHz	27-slow	4.4x	1.1x
Smith/Waterman (hand placed)	66 MHz	31-slow	4.4x	1.1x
Smith/Waterman (autoplaced)	60 MHz	37-slow	5.0x	1.3x
Synthetic Datapath (hand placed)	77 MHz	21-slow	3.9x	1.0x
Synthetic Datapath (autoplaced)	70 MHz	23-slow	4.2x	1.1x
LEON (autoplaced)	38 MHz	67-slow	7.9x	2.2x

Unretimed Xilinx

SFRA Performance Results

Benchmark	Adjusted Xilinx Clock Rate and C-slow Factor	SFRA C-slow Factor	Expected Speedup for 300 MHz SFRA	Area Normalized Speedup
AES (hand placed)	147 MHz 5-slow	24-slow	2.0x	.5x
AES (autoplaced)	123 MHz 5-slow	27-slow	2.4x	.6x
Smith/Waterman (hand placed)	120 MHz 3-slow	31-slow	2.5x	.6x
Smith/Waterman (autoplaced)	117 MHz 3-slow	37-slow	2.5x	.6x
Synthetic Datapath (hand placed)	127 MHz 3-slow	21-slow	2.4x	.6x
Synthetic Datapath (autoplaced)	123 MHz 3-slow	23-slow	2.4x	.6x
LEON (autoplaced)	64 MHz 2-slow	67-slow	4.6x	1.2x

C-Slow Retimed Xilinx