A CAD Framework for MALIBU: An FPGA with Time-multiplexed Coarse-Grained Elements

David Grant

Supervisor: Dr. Guy Lemieux

FPGA 2011 -- Feb 28, 2011

Motivation

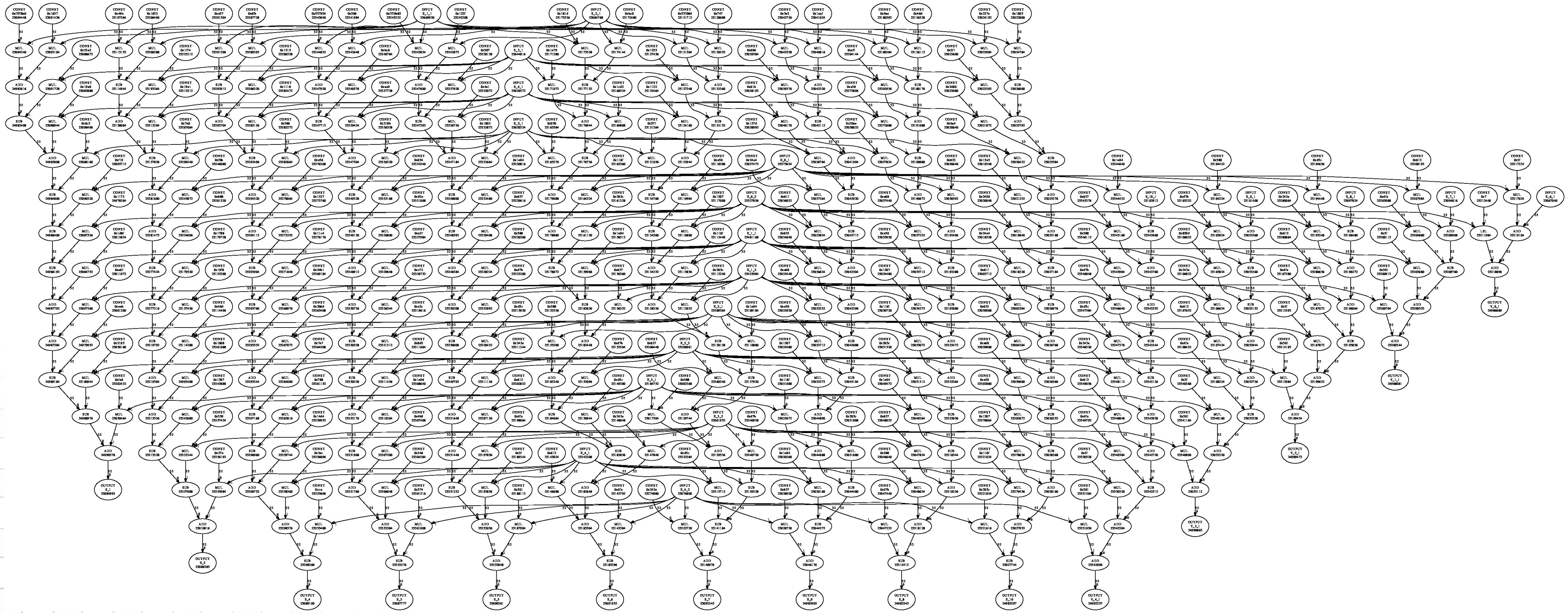
- Growing Industry Trend: Large FPGA Circuits
  - Often from C-to-Hardware or system generators
    - ex. molecular dynamics, rendering, nuclear simulation
  - Word-oriented
  - Millions of gates

Motivation

- Problems with FPGAs
  - CAD runtime can take hours or days
  - Fixed capacity
    - A large circuit may not fit
  - Inefficient use of resources
    - Resources sit idle most of the time

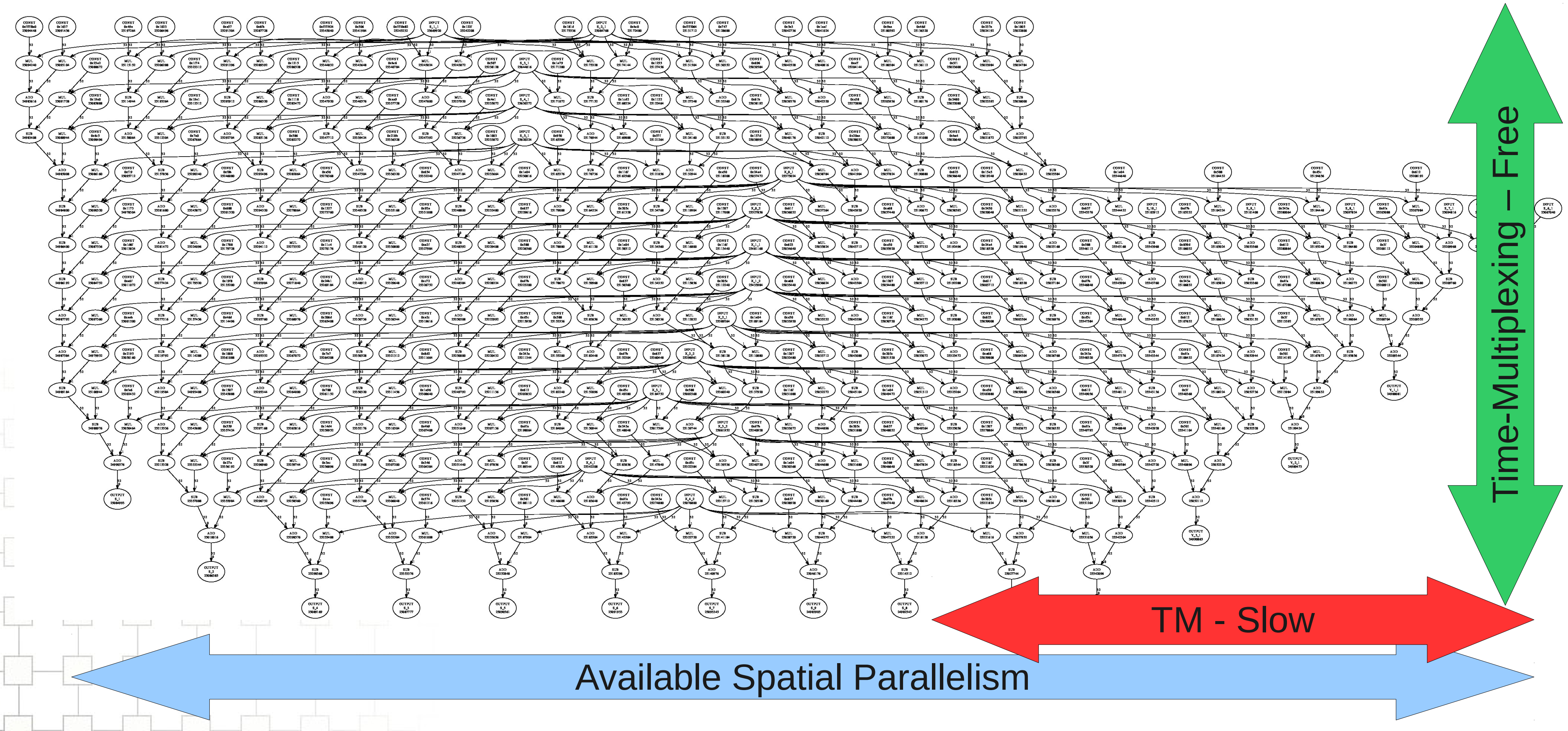
Motivation

- Benchmark: *chem*



Motivation

- Benchmark: *chem*

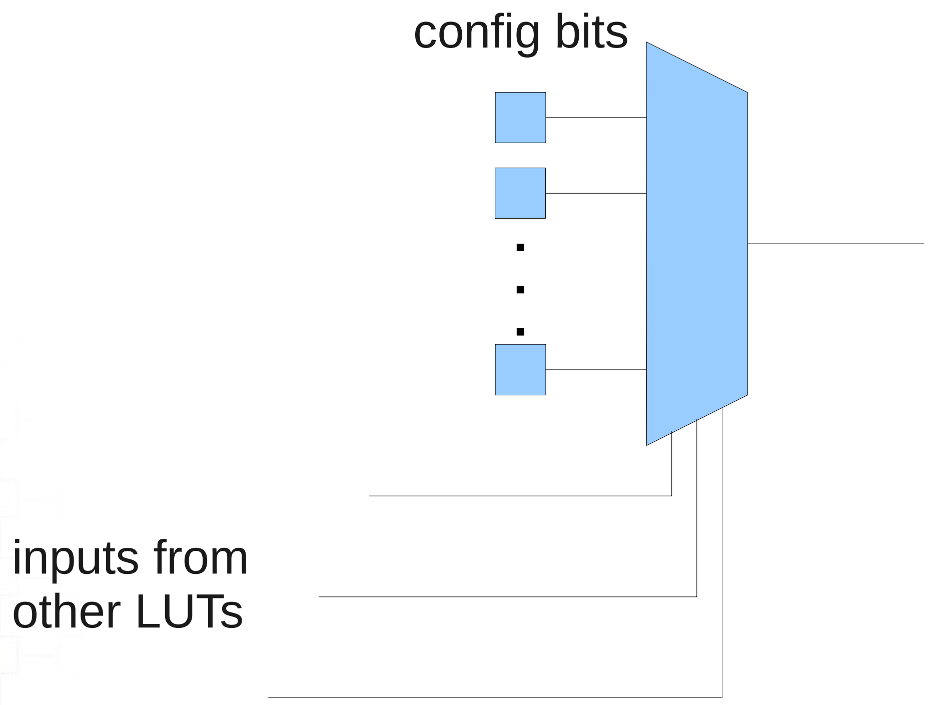


Motivation

- Solution
  - ➔ Divide up the circuit and run it on an array of processors
    - Preserve the coarse-grained features of the circuit
  - ➔ Create coarse-grained-aware CAD tools

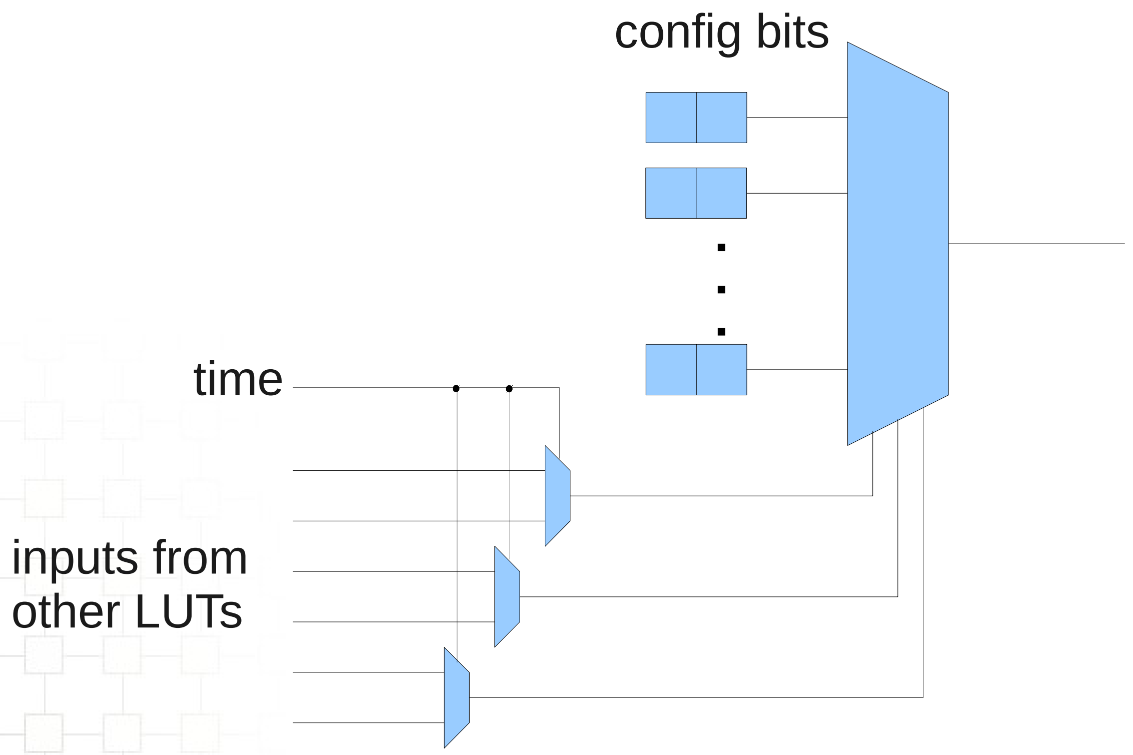
Time-Multiplexing

- Time-Multiplexed FPGAs
  - Look-up Table (LUT)



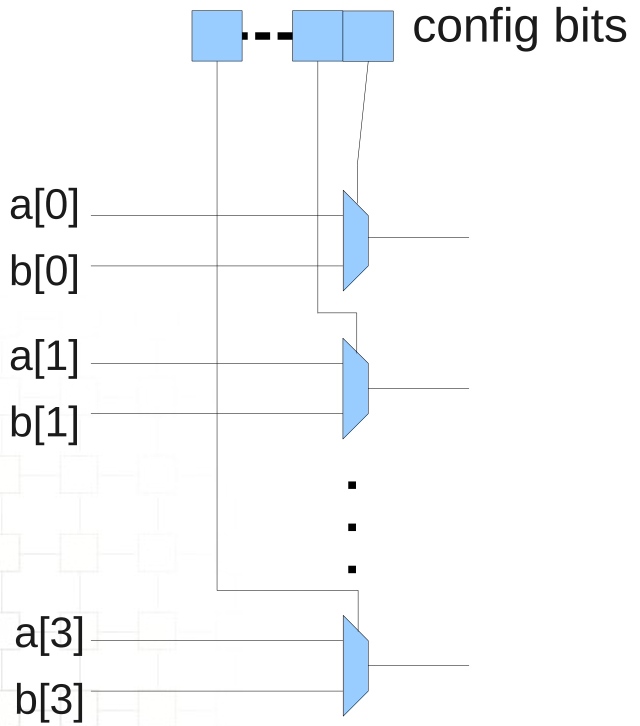
Time-Multiplexing

- Time-Multiplexed FPGAs
  - Time-multiplexed LUT
  - Multiplexer is shared

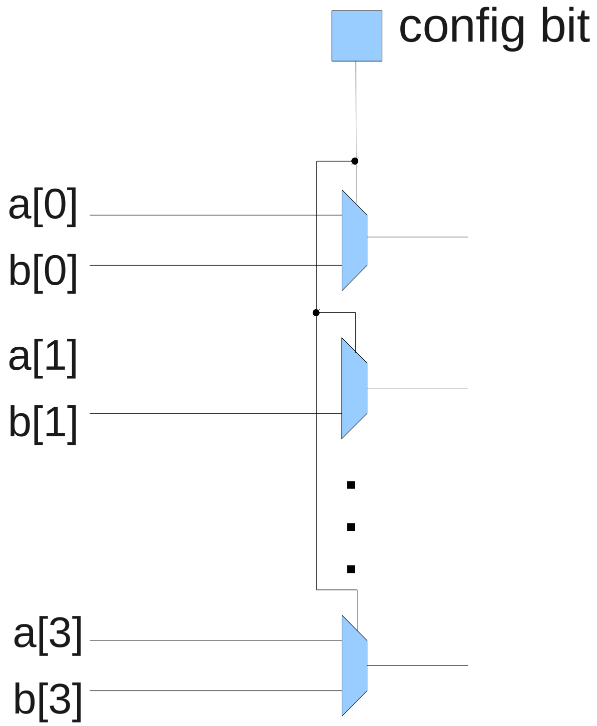


Datapath FPGAs

- Datapath FPGAs
  - Config bit sharing
  - 1.1x density, same performance



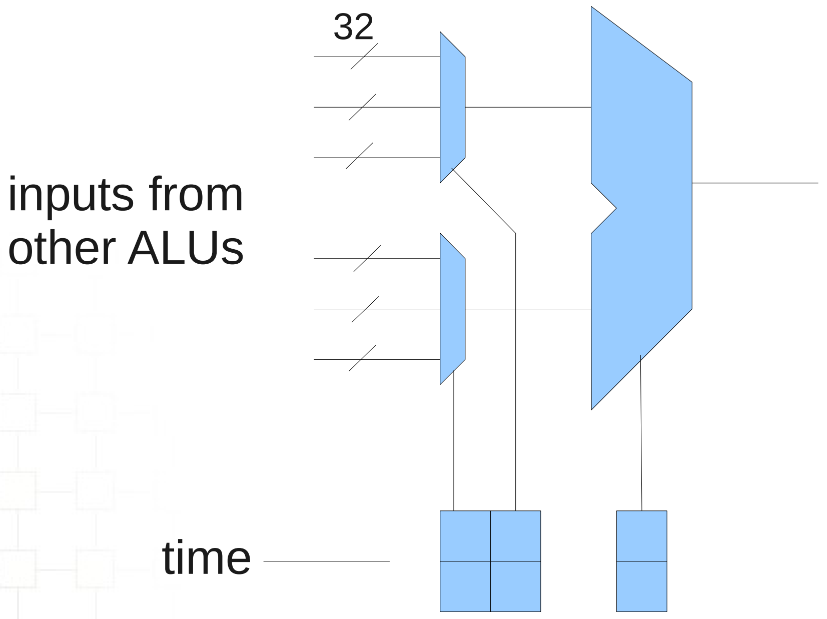
Routing Mux



Datapath Routing Mux

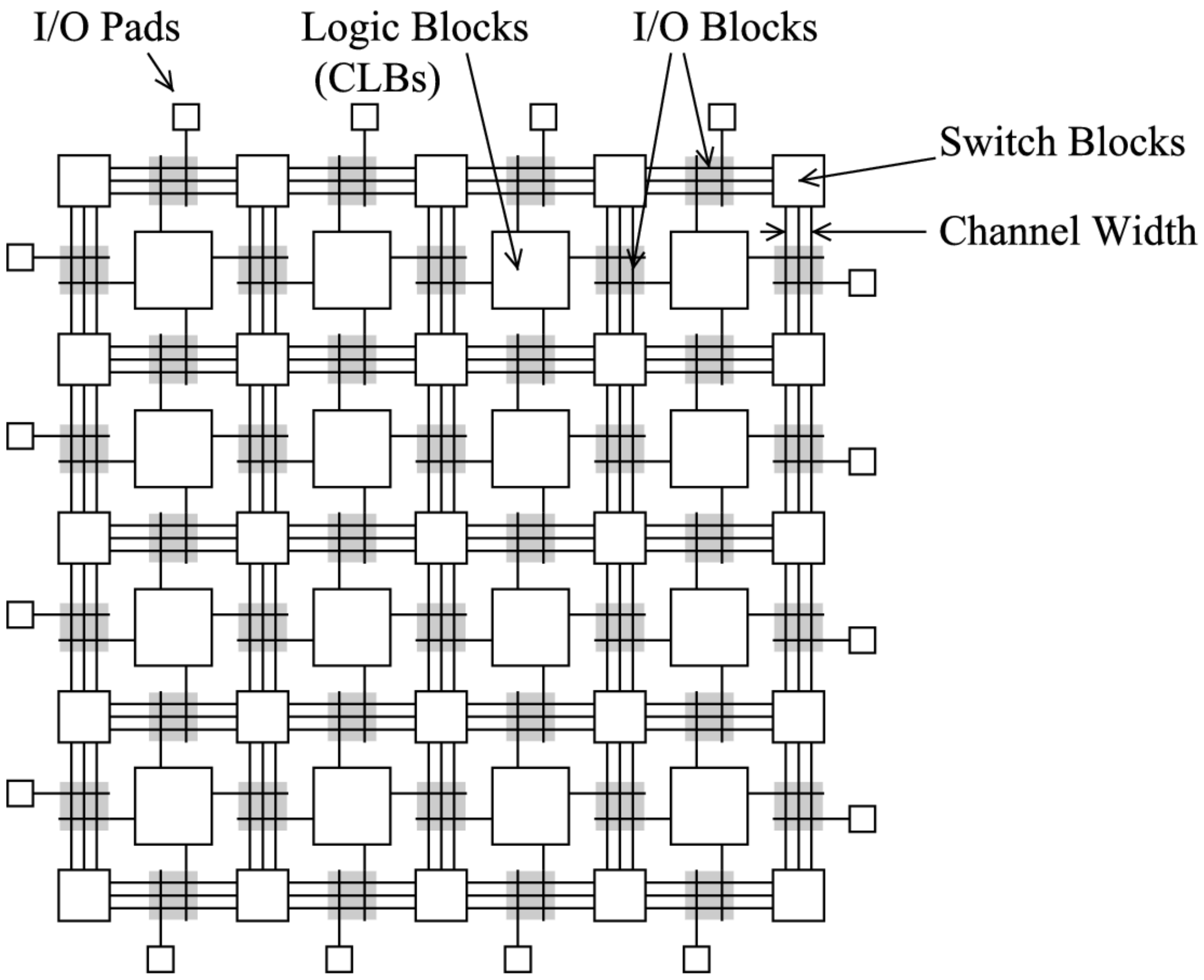
Time-Multiplexing and Datapath

- Where we're going
  - Coarse-grained(datapath) time-multiplexed resources
  - ALU is shared



Overview

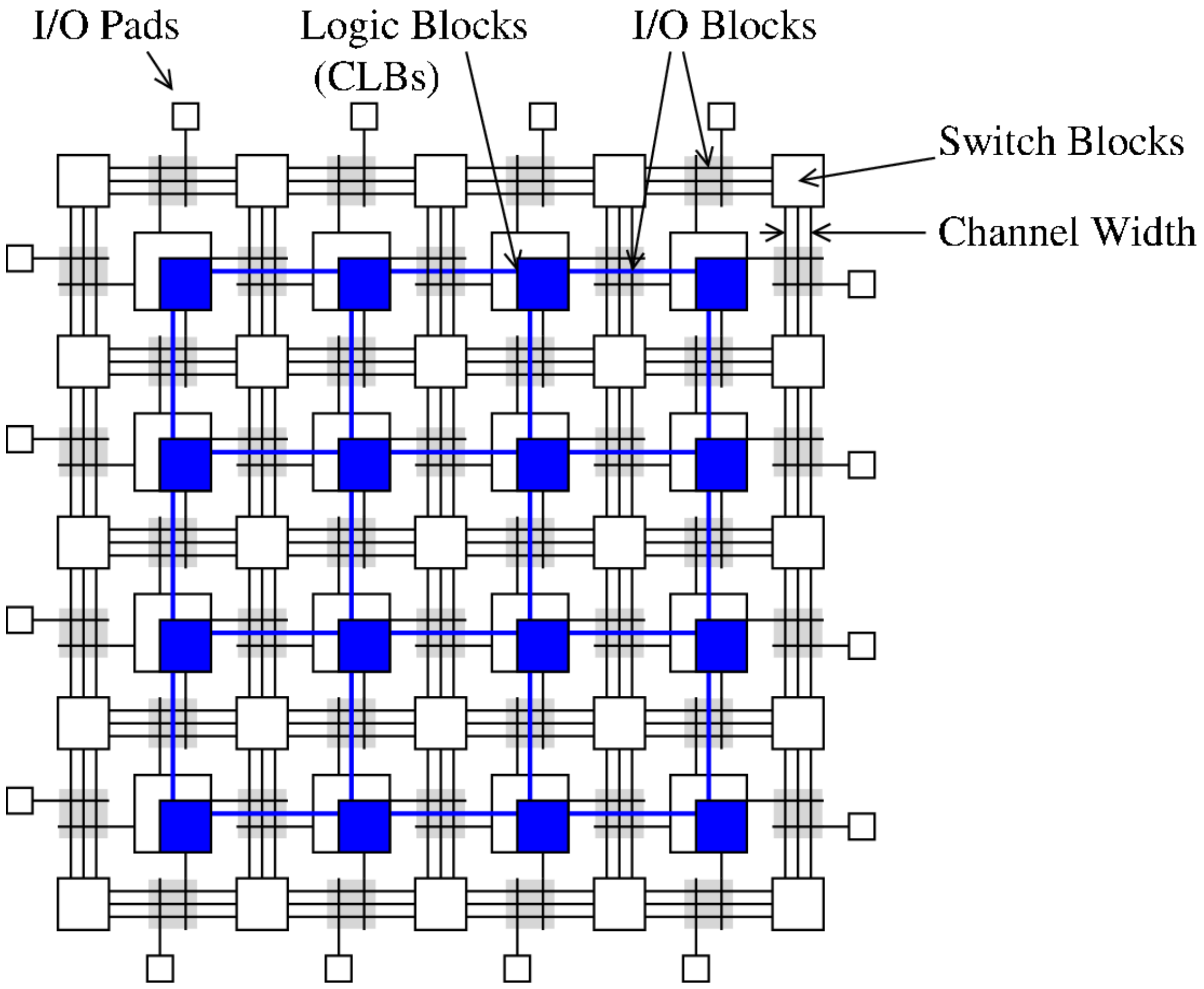
- Motivation
- **Malibu Architecture**
- Synthesis
- Results



Traditional Island-Style FPGA

Overview

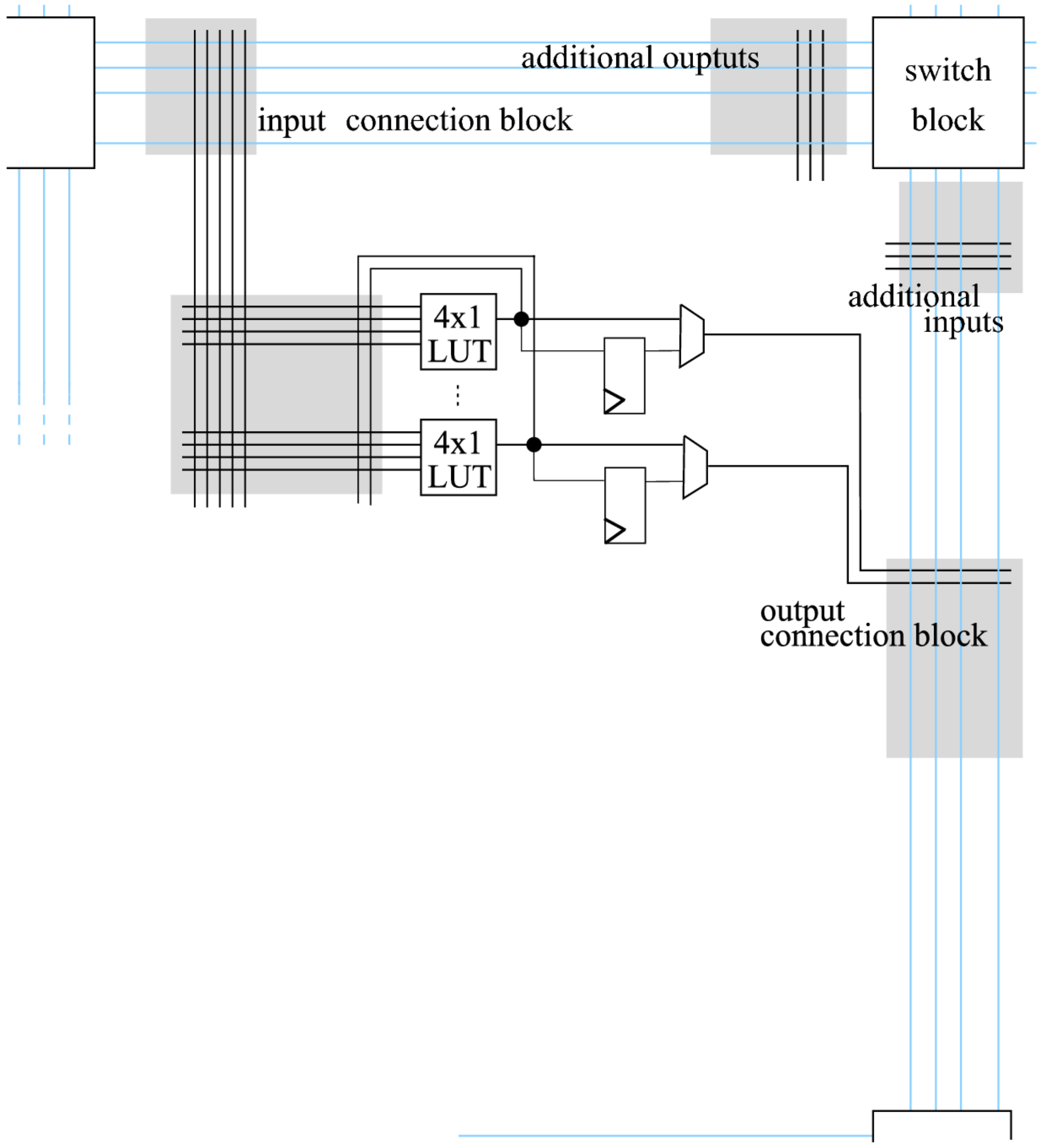
- Motivation
- **Malibu Architecture**
- Synthesis
- Results



Malibu Architecture

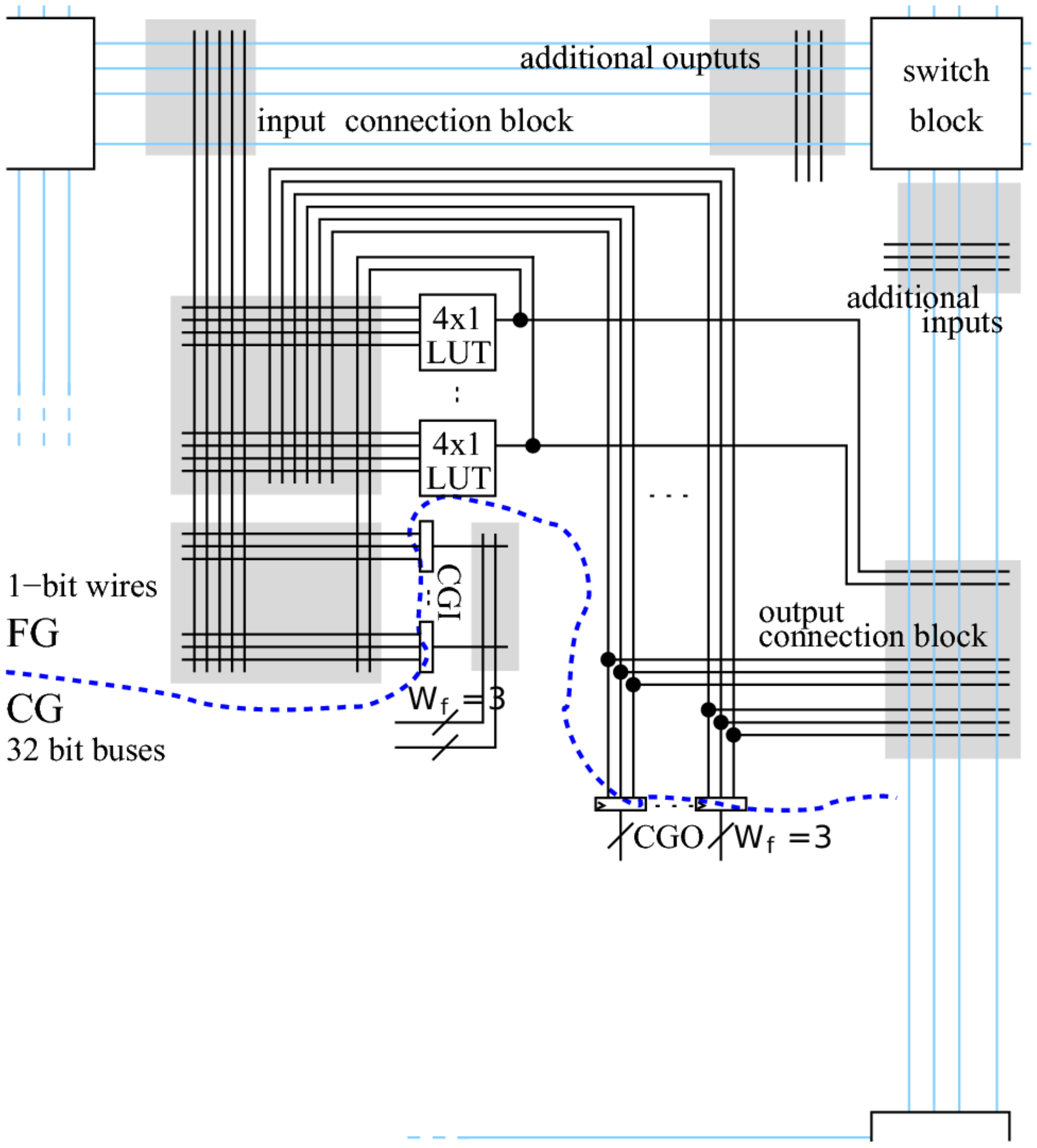
Malibu Architecture

- Traditional FPGA CLB



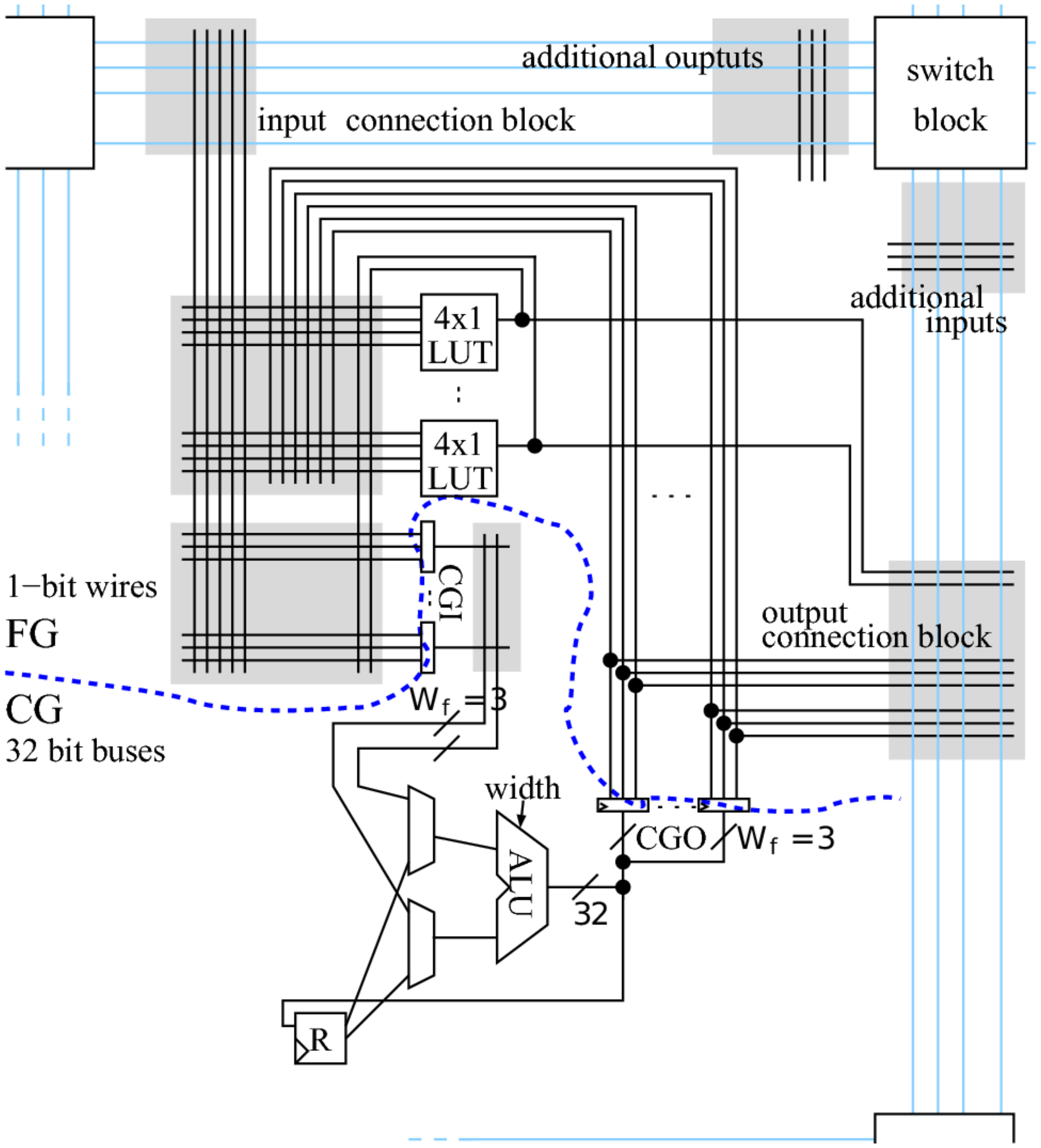
Malibu Architecture

- Add Coarse-Grained inputs and outputs



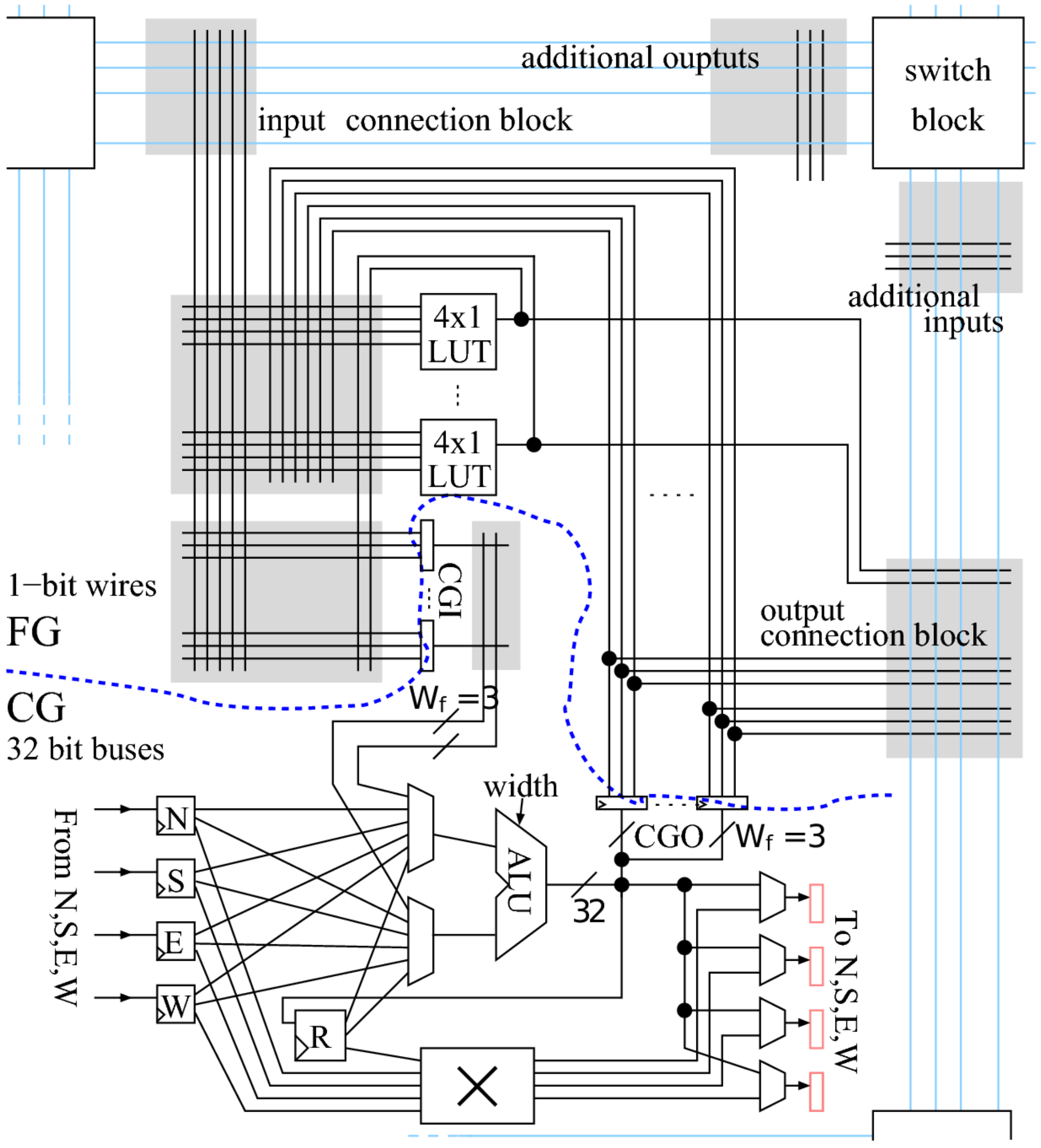
Malibu Architecture

- Add an ALU and register file



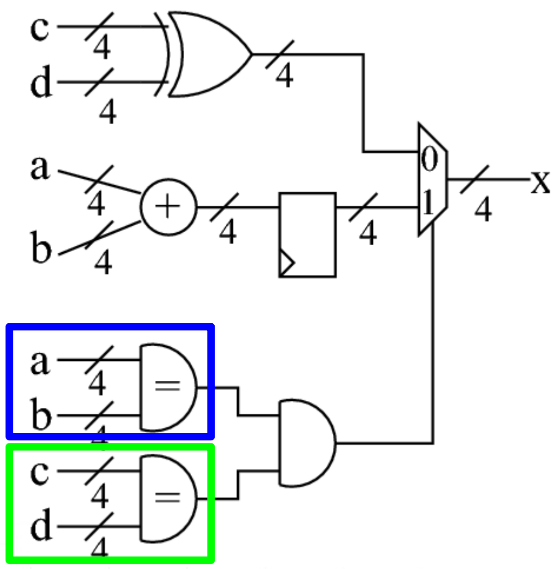
Malibu Architecture

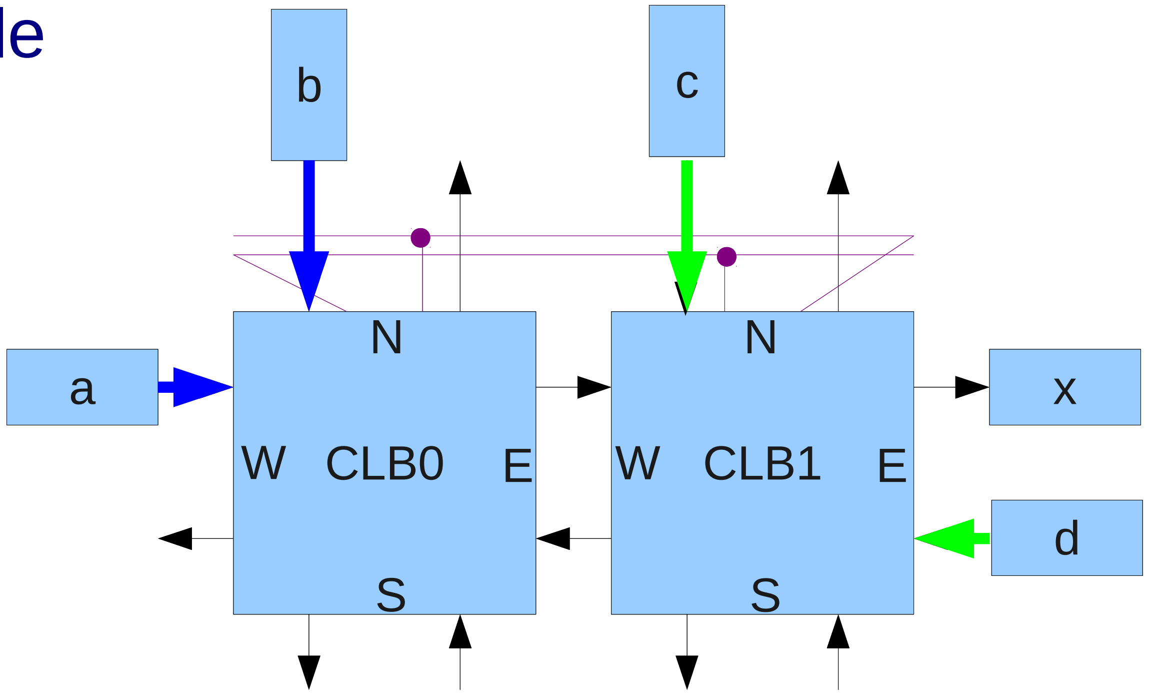
- Add coarse-grained communication
- Each CG has a schedule
  - SL instructions
  - Fmax = 1GHz / SL



Malibu Architecture

- Circuit Execution Example

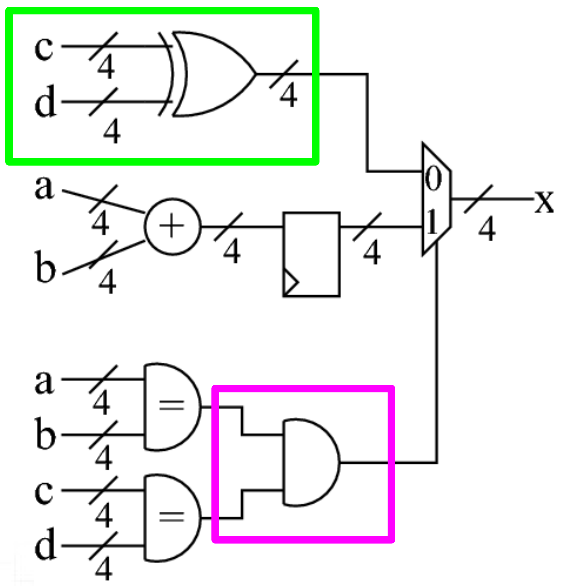


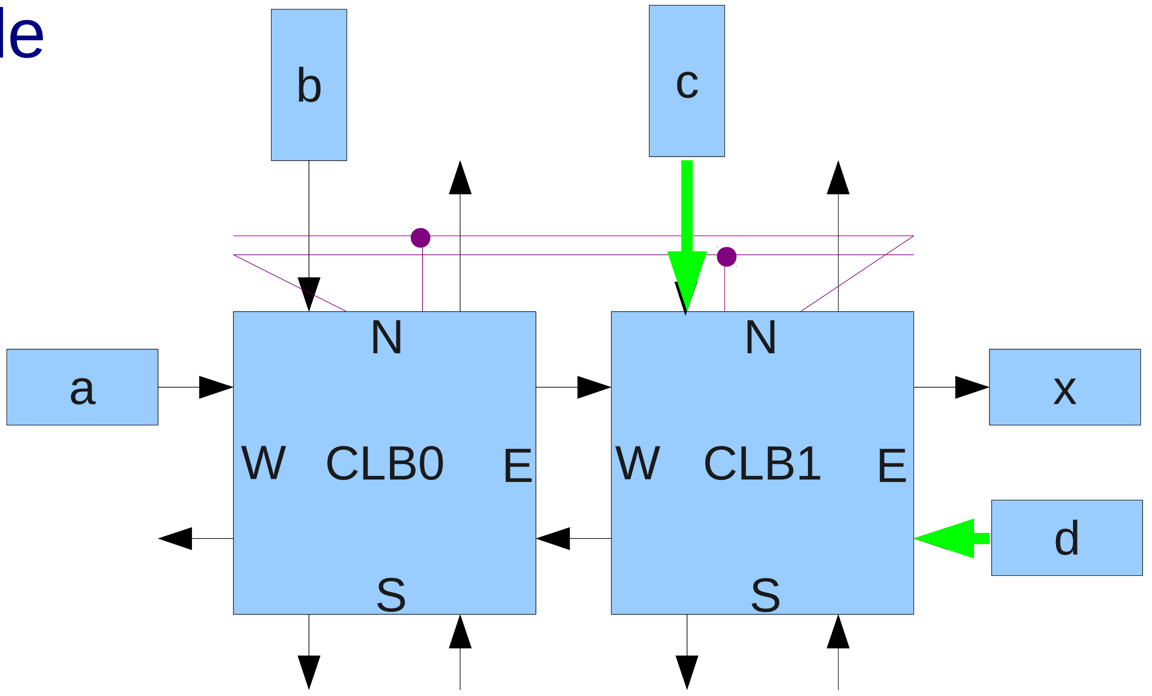


Time	CLB0 CG	CLB0 FG	CLB1 CG	CLB1 FG
0	EQ N0,W0 -> CGO0		EQ N0,E0 -> CGO0	
1	NOP	LUT	XOR N0,E0 -> R0	
2	ADD N0,W0 -> E0		MUX W0,R0,CGI0->E0	

Malibu Architecture

- Circuit Execution Example

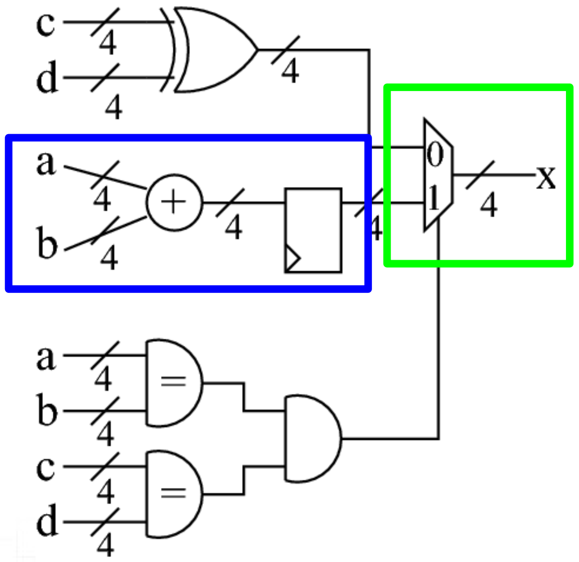


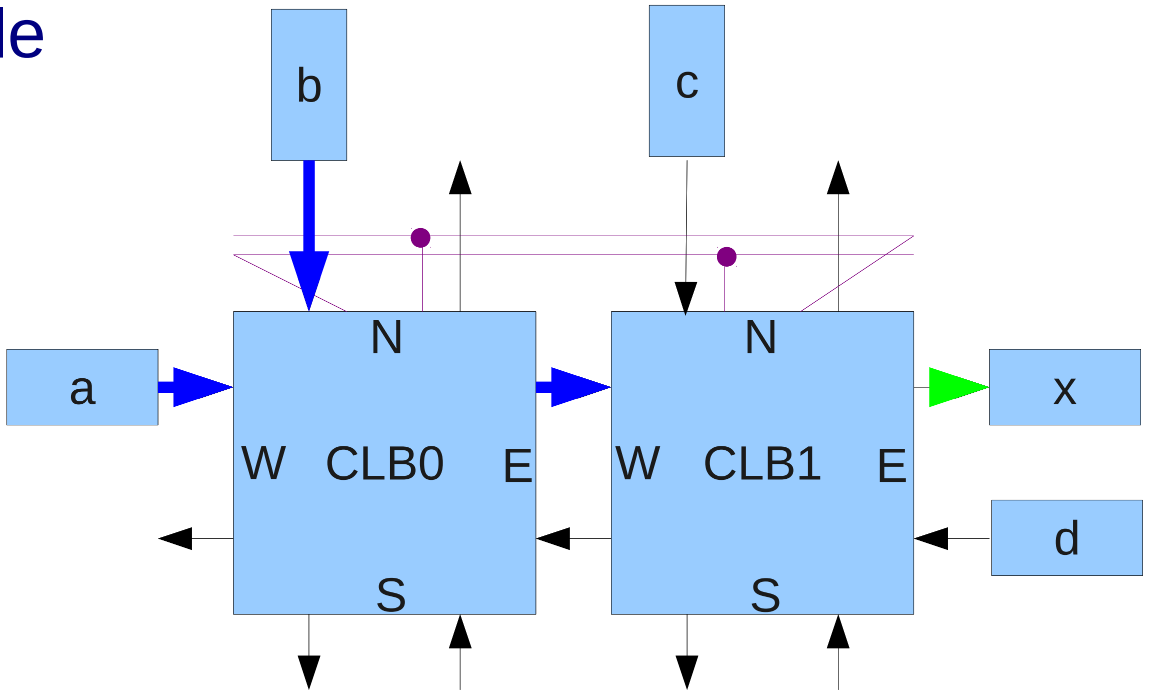


Time	CLB0 CG	CLB0 FG	CLB1 CG	CLB1 FG
0	EQ N0,W0 -> CGO0		EQ N0,E0 -> CGO0	
1	NOP	LUT	XOR N0,E0 -> R0	
2	ADD N0,W0 -> E0		MUX W0,R0,CGI0->E0	

Malibu Architecture

- Circuit Execution Example

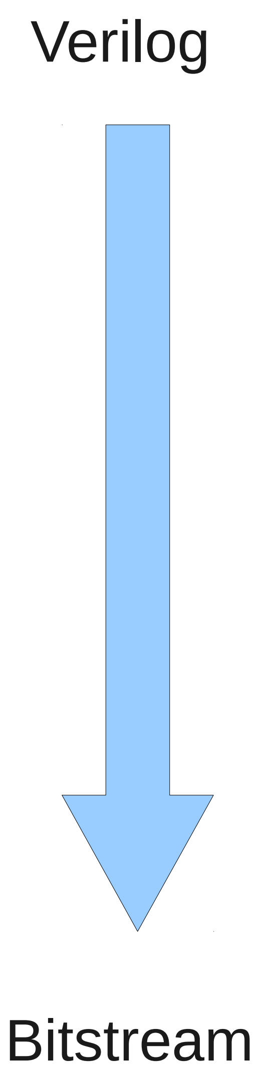




Time	CLB0 CG	CLB0 FG	CLB1 CG	CLB1 FG
0	EQ N0,W0 -> CGO0		EQ N0,E0 -> CGO0	
1	NOP	LUT	XOR N0,E0 -> R0	
2	ADD N0,W0 -> E0		MUX W0,R0,CGI0->E0	

Overview

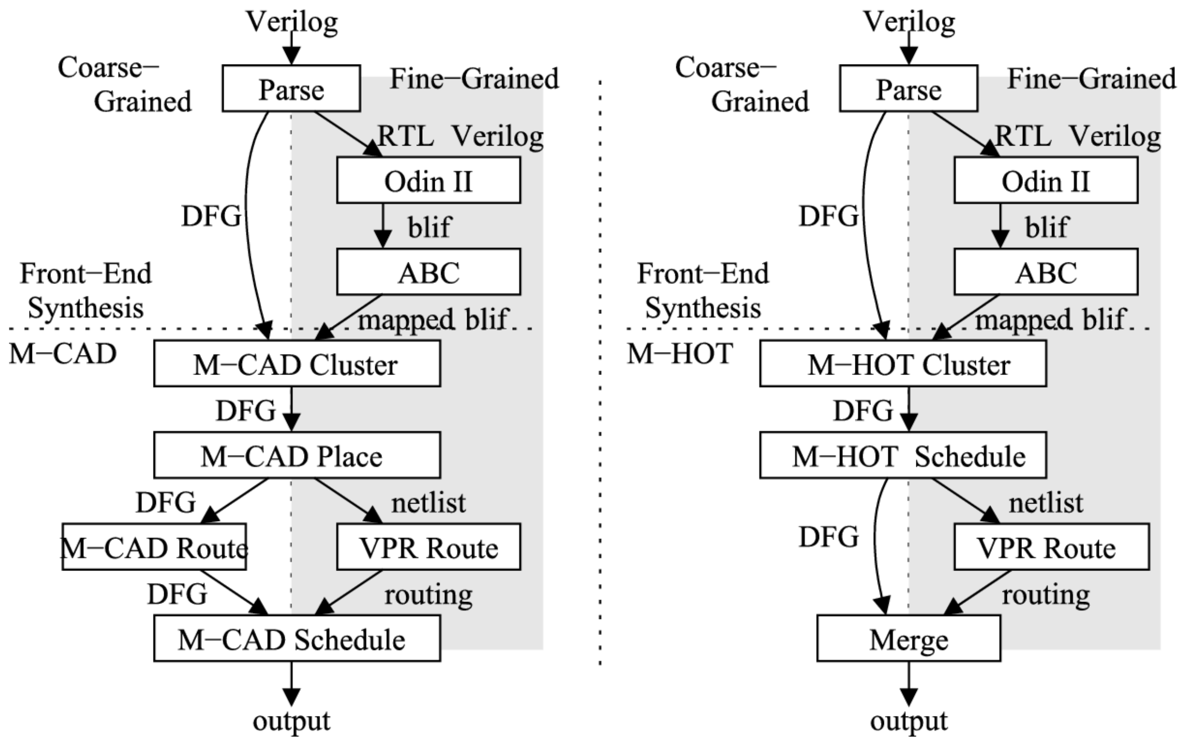
- Motivation
- Malibu Architecture
- **Synthesis**
- Results



Overview

- Motivation
- Malibu Architecture
- **Synthesis**
- Results

Verilog

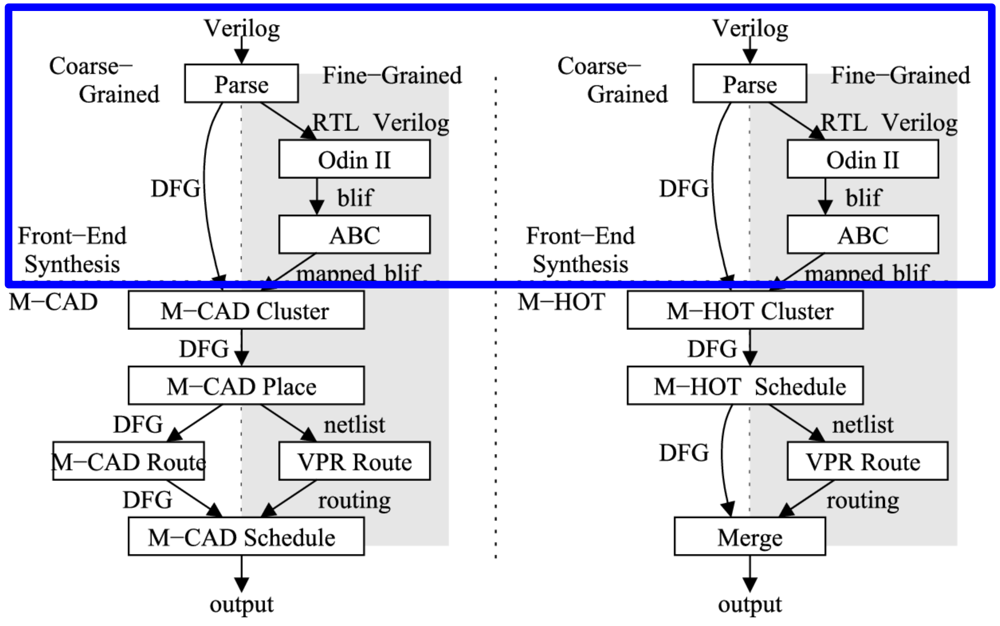


M-CAD M-HOT

Bitstream

Front-End Synthesis

- Parse and Elaborate
  - Use Verilator
  - Construct a CDFG
  - Optimize



- Coarse-Grained Synthesis
  - Map CDFG to Malibu instructions
  - Various CDFG transformations

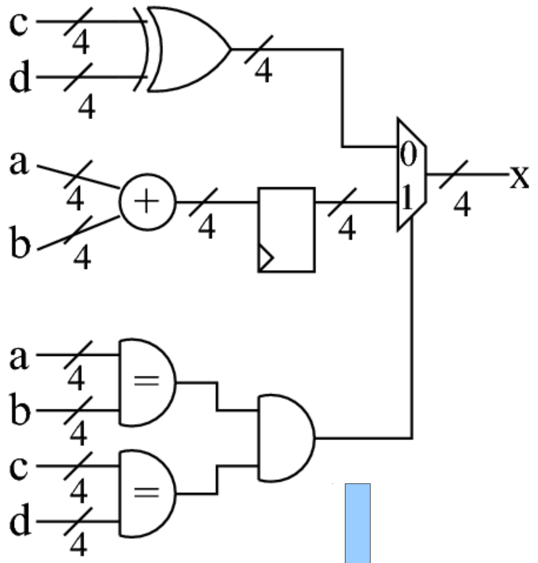
- Fine-Grained Synthesis
  - Extract signals ≤ W_f
  - Use OdinII and ABC to synthize to LUTs

Back-End Synthesis

- M-CAD
  - ➔ Traditional FPGA-CAD flow
  - ➔ Separate Placement, Routing, <u>Scheduling</u>
- M-HOT
  - ➔ Integrated placement, routing, scheduling
  - ➔ Divides problem into levels, place+route each level
- Both Approaches
  - ➔ Can target any-sized architecture
  - ➔ Can trade area for performance
  - ➔ Fast

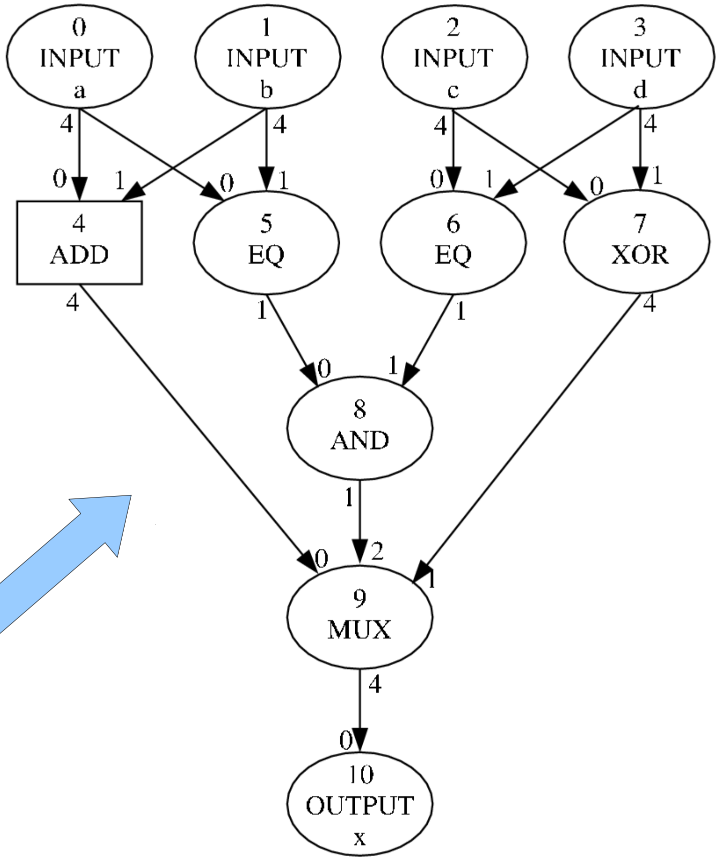
Synthesis

- Example



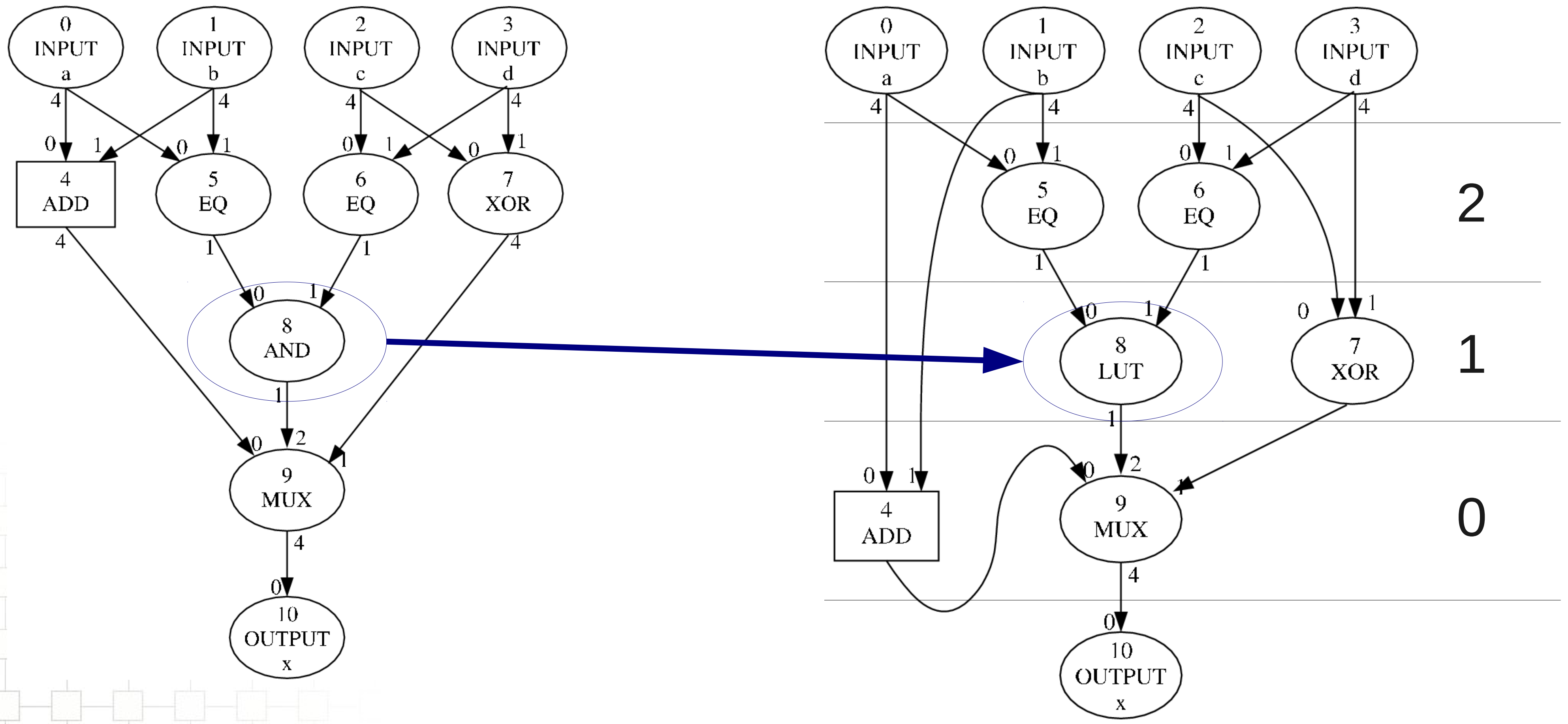
```
assign c1 = (a == b) ? 1'b1 : 1'b0;
assign c2 = (c == d) ? 1'b1 : 1'b0;
assign t2 = c ^ d;
assign x = (c1 & c2) ? t1 : t2;

always @(posedge clk) begin
   t1 <= a + b;
end
```



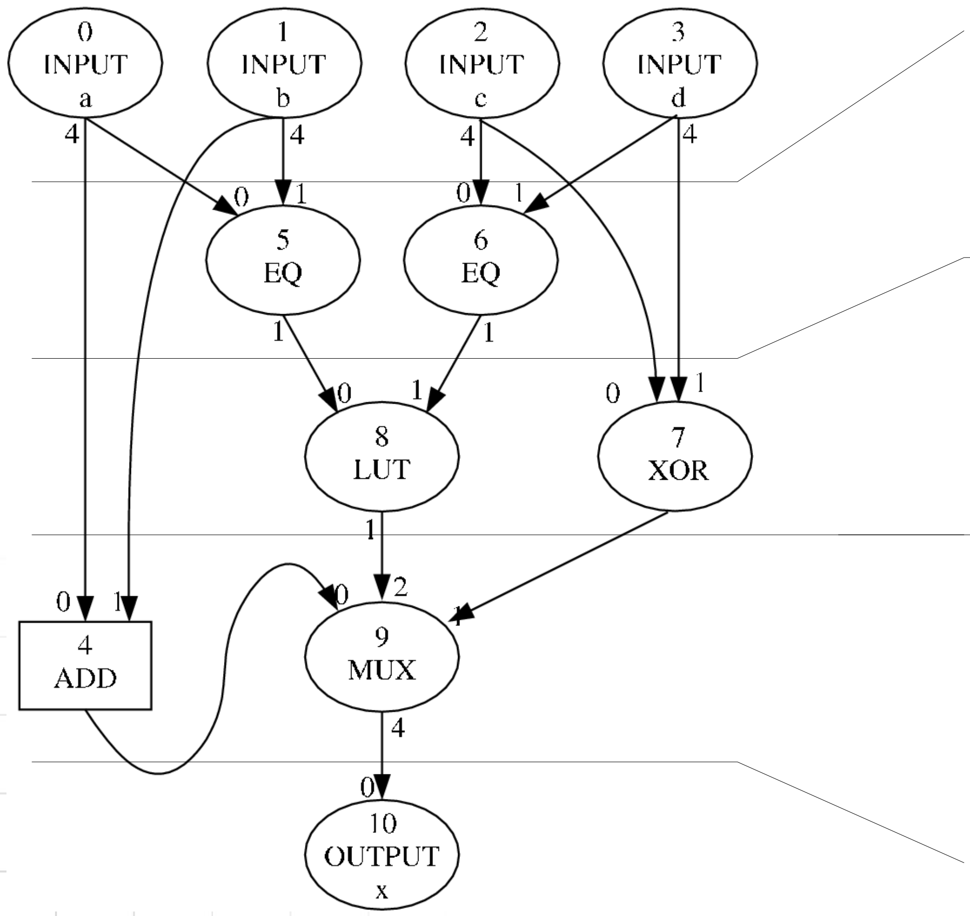
Synthesis

- CDFG and ALAP output of Front-End Synthesis



Synthesis

- M-HOT Place, Route, Schedule each height



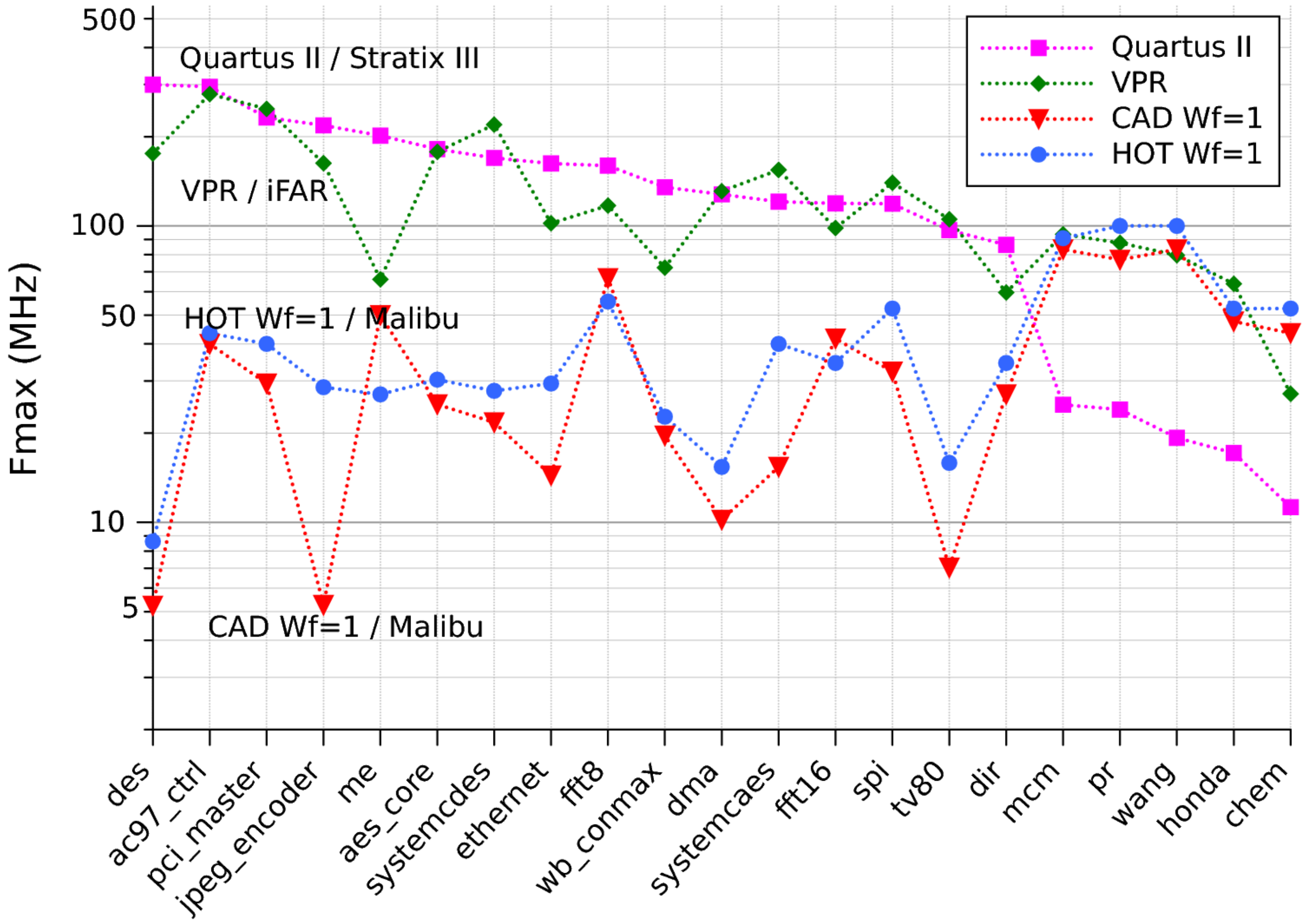
CLB0	CLB1
5: EQ - -	6: EQ - -
5: EQ 8: LUT -	6: EQ 7: XOR -
5: EQ 8: AND 4: ADD	6: EQ 7: XOR 9: MUX

Overview

- Motivation
- Malibu Architecture
- Synthesis
- **Results**

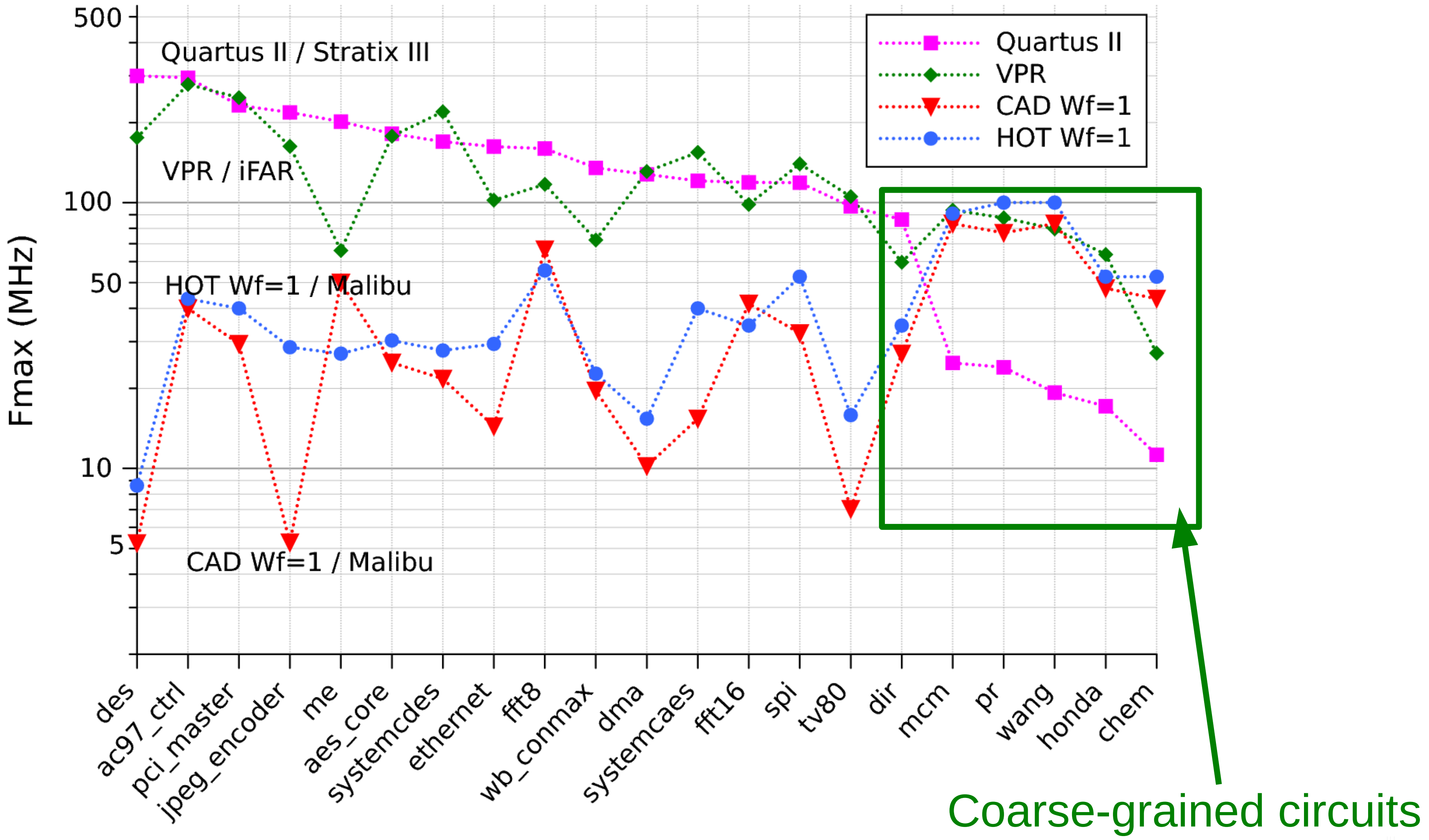
Results

- Frequency (MHz) for each benchmark



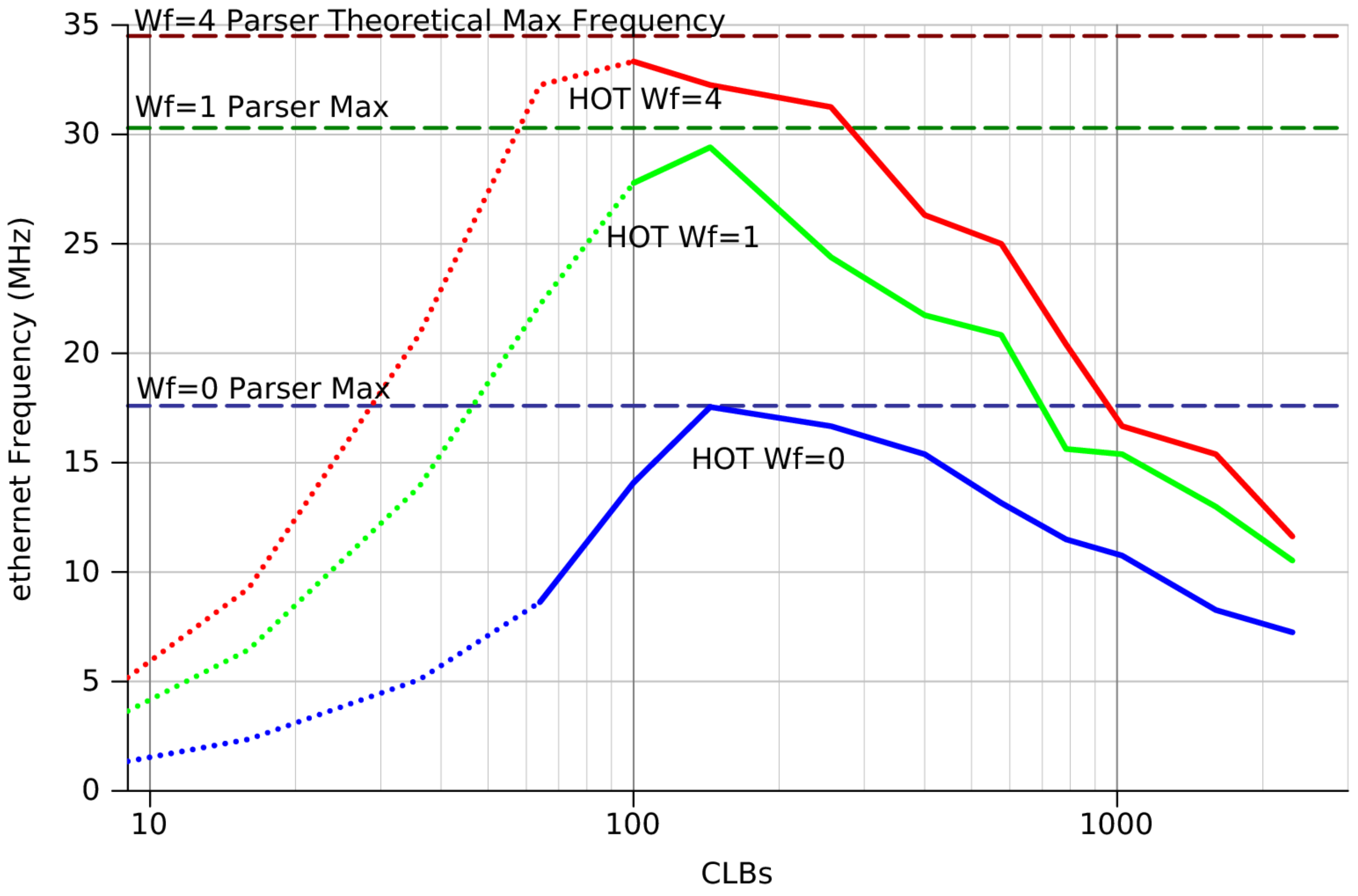
Results

- Frequency (MHz) for each benchmark



Results

- Area vs. Performance tradeoff



Results

- Results (compared to Quartus II / Stratix III)

	M-HOT	M-CAD
Synthesis Time Improvement:	30.9x	77.0x
User Clock Speed:	0.12x	0.07x
Density:	1.48x	0.67x

10x = 10 times better than the Quartus result
1x = same as Quartus
0.1x = 1/10th the Quartus result (10 times worse)

Future Work

- Improve Front-End Synthesis
  - ➔ Needs to be both coarse-grain and fine-grain aware
  - ➔ Coarse-grained optimizations
- Improve M-CAD and M-HOT
  - ➔ Possibly 2x-3x performance improvement

Thanks

- Purpose
  - Implement a circuit on a coarse-grained/fine-grained architecture
- Malibu Architecture
  - FPGA with time-multiplexed coarse-grained resources
  - Can trade density for performance
- Synthesis (M-CAD and M-HOT)
  - Fast, up to 250x faster than QuartusII
  - Fmax results within 1/10th of an FPGA