CAD Algorithms and Performance of Malibu: An FPGA with Time-Multiplexed Coarse-Grained Elements

David Grant

Supervisor: Dr. Guy Lemieux

April 8, 2011

Motivation

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

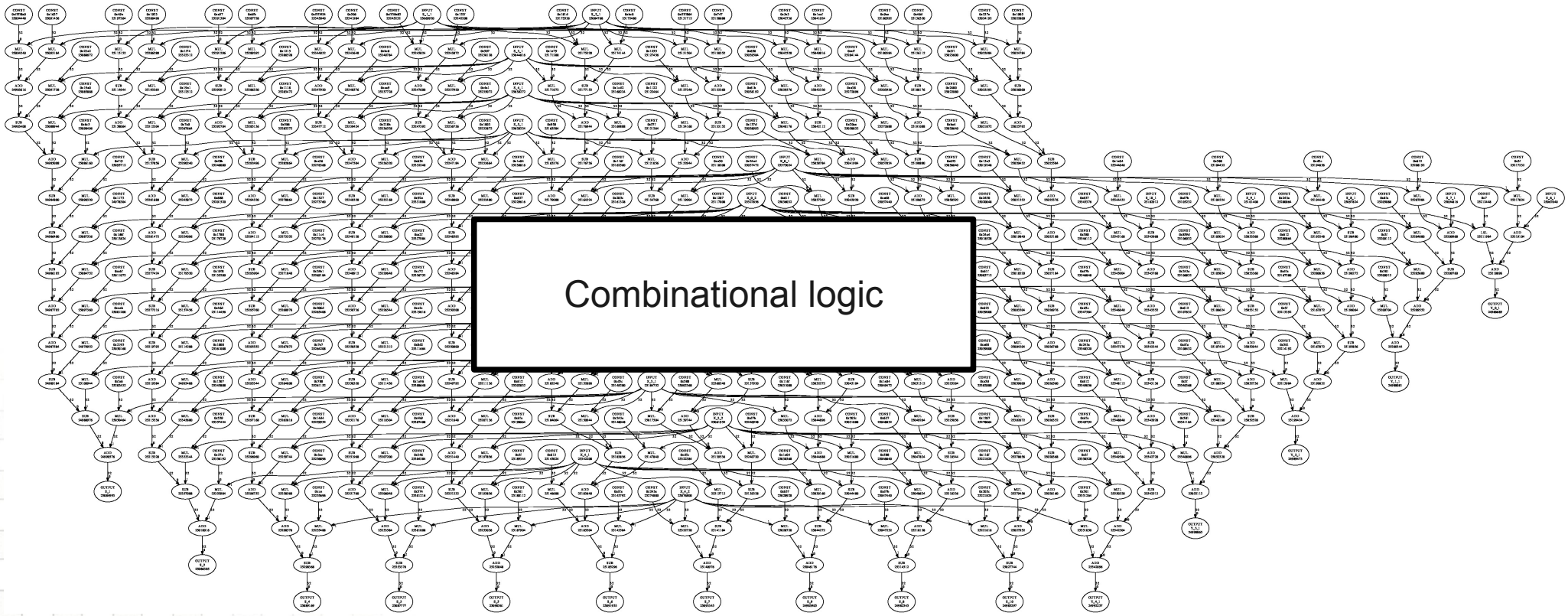
Motivation

- Problems with FPGAs
  - Synthesis time
    - CAD runtime can take hours or days
  - Density
    - FPGAs have a fixed capacity
    - A large circuit may not fit
- Not A Problem
  - Circuit speed
    - FPGAs are fast
    - Preserve as much as possible

Motivation

- Benchmark: *chem*

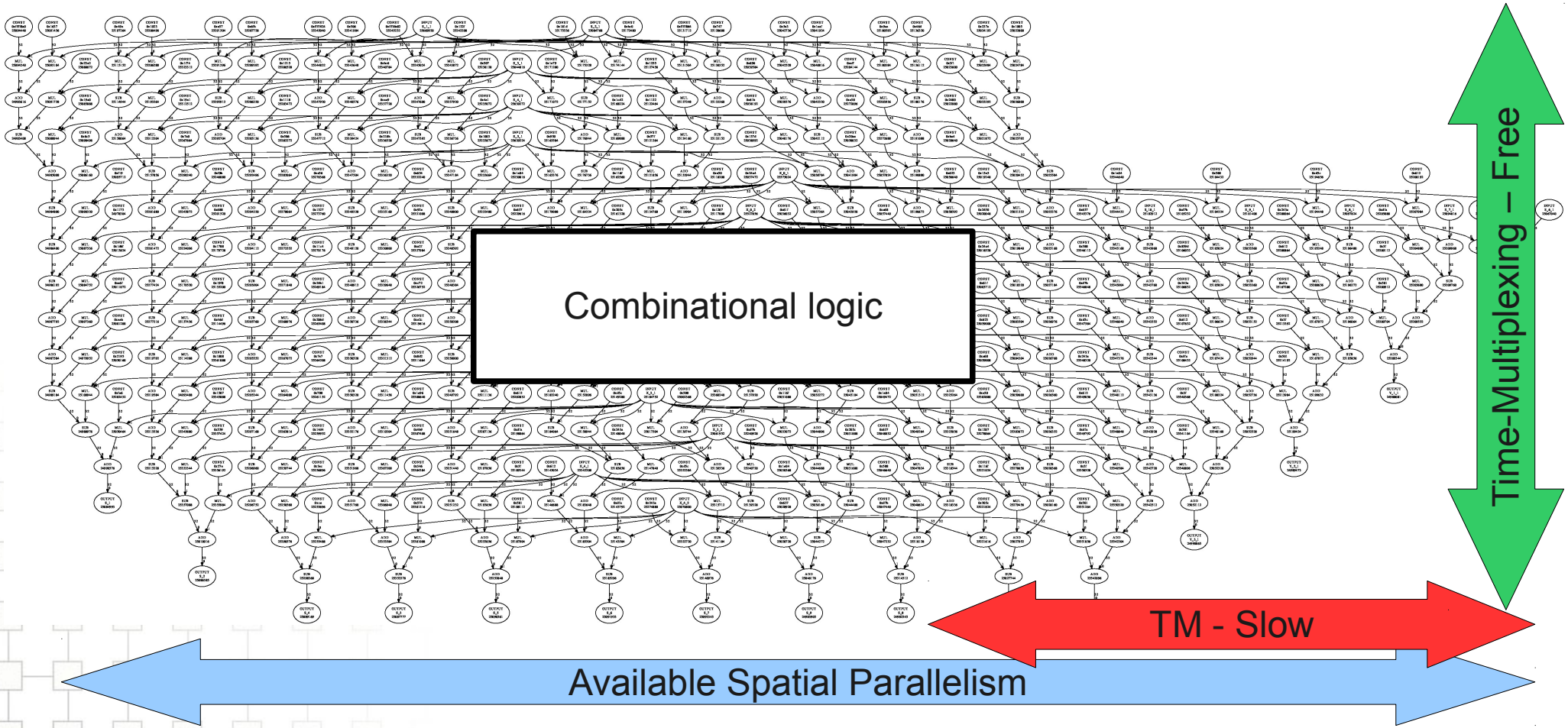
Inputs and register outputs



Outputs and register inputs

Motivation

- Benchmark: *chem*



Motivation

- Solution
  - Preserve the coarse-grained features of the circuit
    - Fast CAD tools
  - Divide up the circuit and run it on an array of processors
    - Improved density because of time multiplexing
  - Create coarse-grained-aware CAD tools

Statement of Thesis

"To investigate mapping coarse-grained circuits onto an FPGA-like architecture comprised of fine-grained and time-multiplexed, coarse-grained resources"

- Objective: **Find a better way to map** coarse-grained circuits onto reconfigurable devices vs. existing commercial tools and approaches
  - Faster synthesis
  - Increased density
  - Limited decrease in circuit performance
  - Focus on CAD, not architecture or synthesis

Thesis Contributions

1. Malibu: A hybrid coarse-grained / fine-grained architecture
   - Time-multiplexed coarse-grained resources
     - Improves density
     - Enables faster CAD tools
   - Fine-grained resources
     - Improves circuit speed for some circuits

2. M-CAD: An FPGA-CAD based tool flow
   - Traditional FPGA CAD + support for coarse-grained resources
   - For coarse-grained circuits vs. QuartusII
     - 77x faster synthesis vs. QuartusII (commercial CAD tool)
     - 10x density at 0.01x circuit speed
     - 0.99x density at 1.45x circuit speed

Thesis Contributions

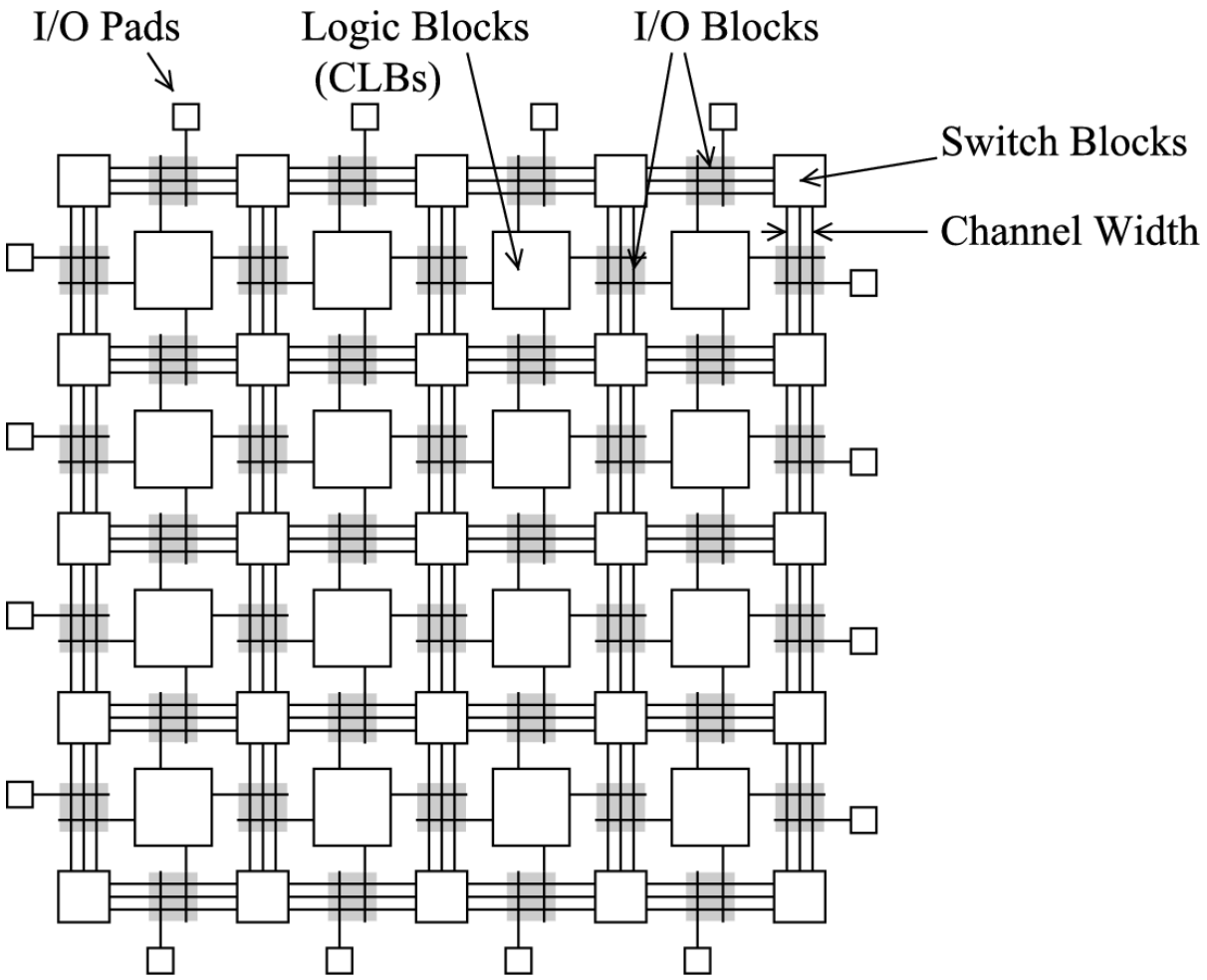
3. M-HOT: A height-oriented tool flow
   - Integrated placement, routing and scheduling
   - For coarse-grained circuits vs. Quartus
     - 31x faster synthesis
     - 2.74x density at 2.71x circuit speed

Related Work

Project	Resources	Time-Mux	Host CPU	Global Mem	Source Language
ADRES	coarse	alus only	vliw	yes	netlist
Ambric	coarse	alus only	-	-	Java
RaPiD	coarse	-	mips	-	unknown
MorphoSys	coarse	alus only	risc	yes	netlist
PipeRench	coarse	alus only	sparc	yes	C
RAW	coarse	coarse	-	-	C
SPR	**coarse+fine**	**coarse**	**-**	**-**	**C-like (Macah)**
WaveScalar	coarse	-	-	-	C
FPGA	fine	-	-	-	VHDL/Verilog
Tabula	fine	fine	-	-	VHDL/Verilog
TSFPGA	fine	fine	-	-	netlist
VEGA	fine	fine	-	-	netlist
DP-FPGA	coarse+fine	-	-	-	netlist
MALIBU	**coarse+fine**	**coarse**	**-**	**-**	**Verilog**

Overview

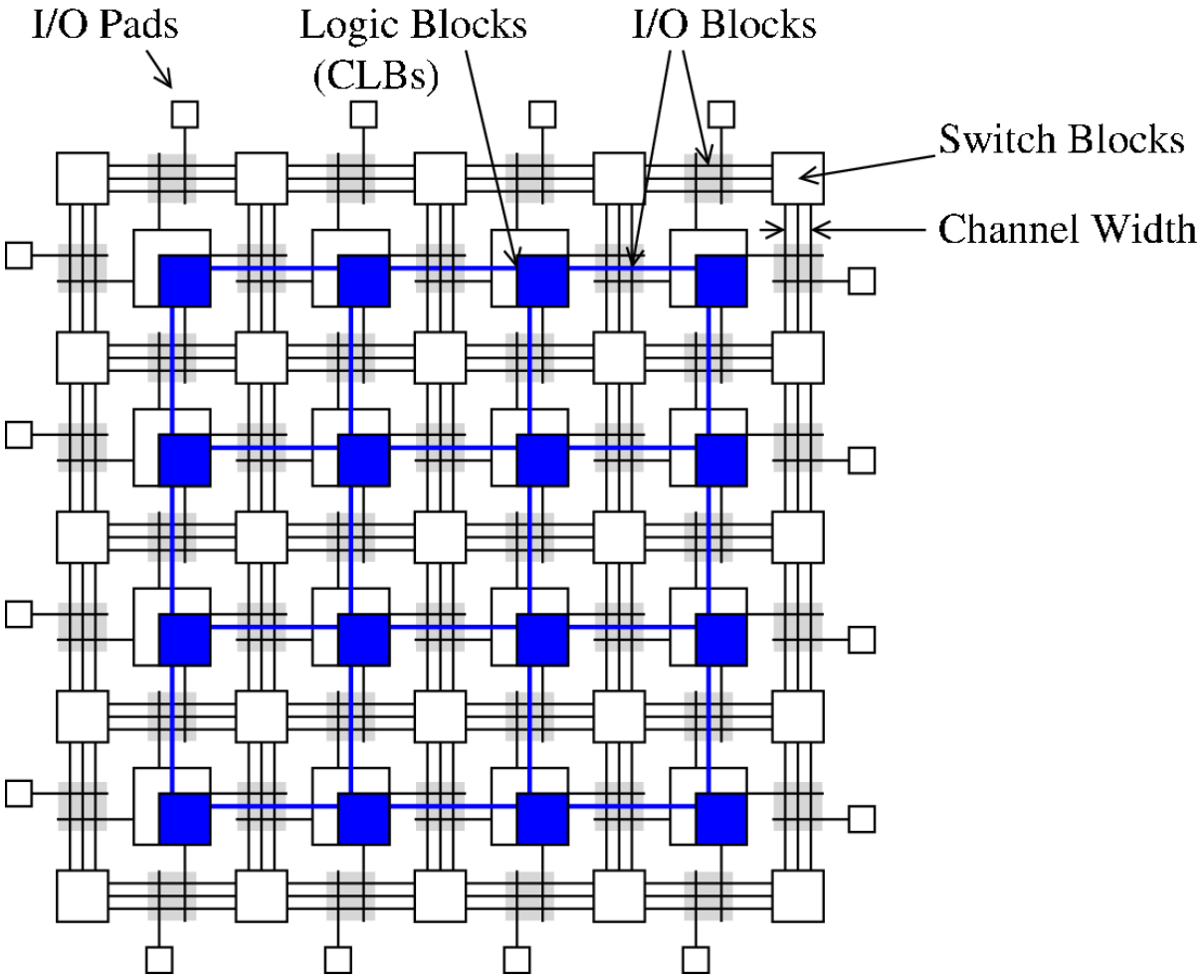
- Motivation
- **Malibu Architecture**
- Front-End Synthesis
- M-CAD tool flow
- M-HOT tool flow
- Conclusions



Traditional Island-Style FPGA

Overview

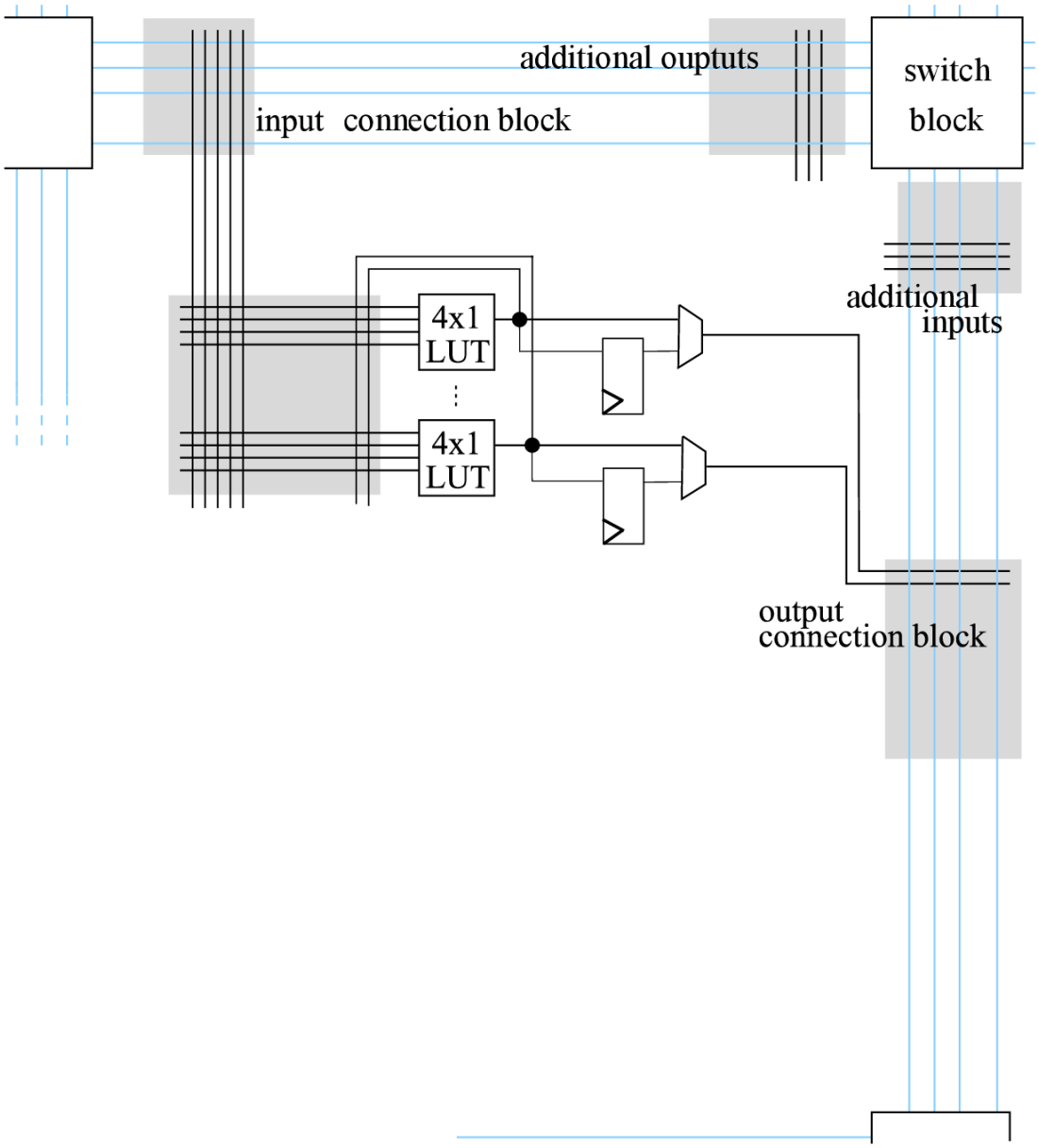
- Motivation
- **Malibu Architecture**
- Front-End Synthesis
- M-CAD tool flow
- M-HOT tool flow
- Conclusions



Traditional Island-Style FPGA
+ Coarse-Grained Blocks

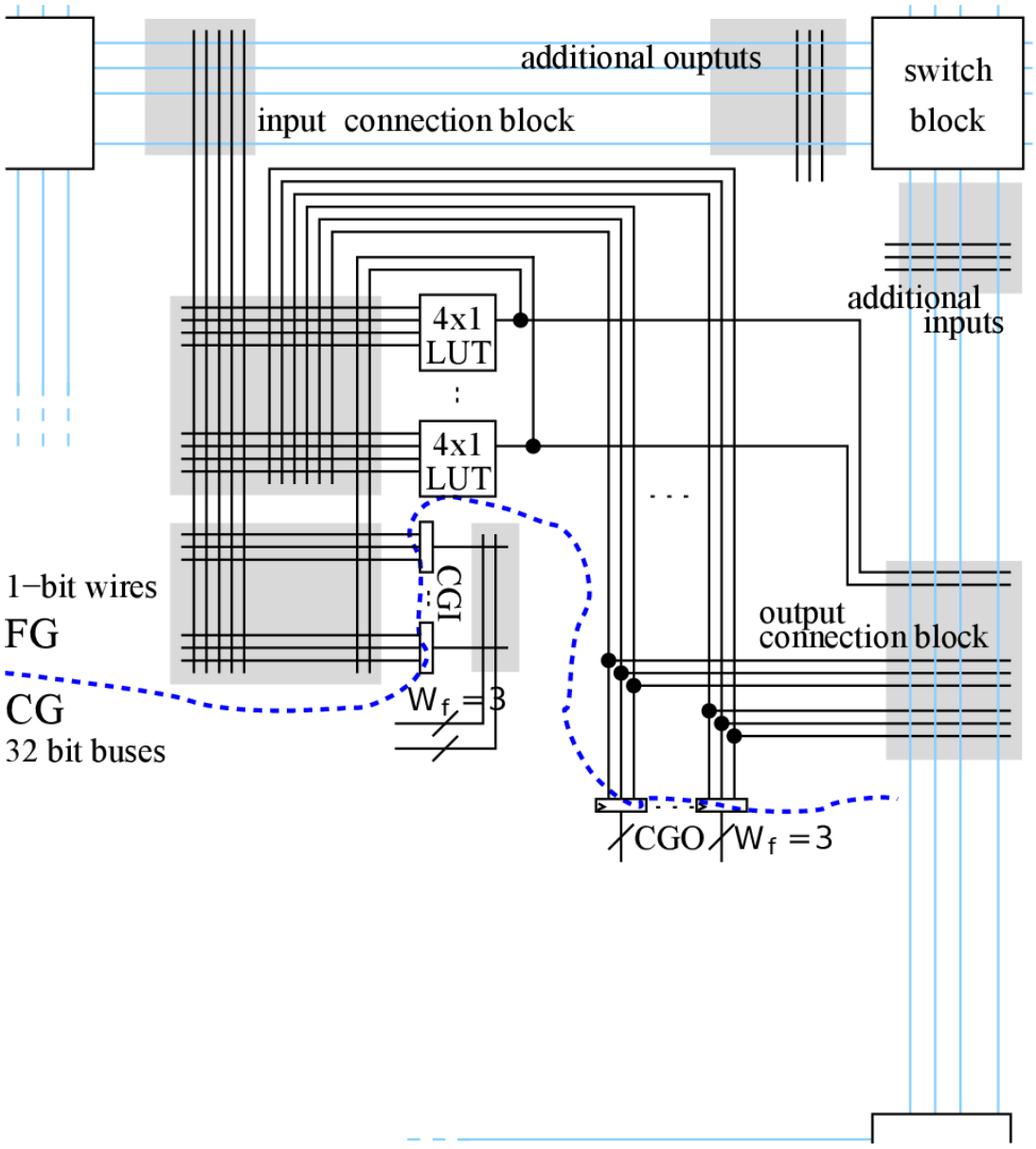
Malibu Architecture

- Traditional FPGA CLB
  - Add CG to this



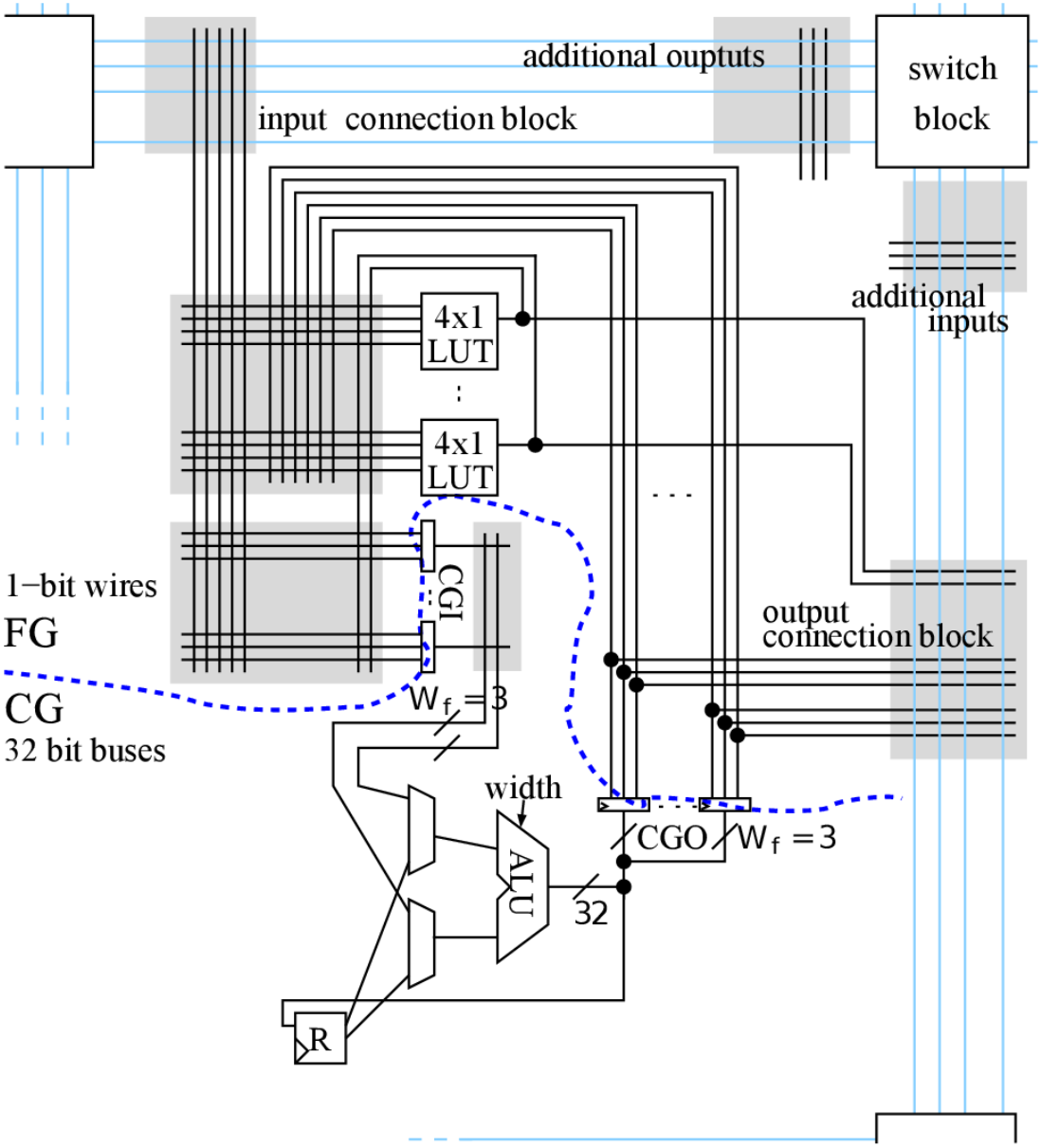
Malibu Architecture

- Add Coarse-Grained inputs and outputs
  - CGOs
    - Into FG
  - CGIs
    - Out of FG



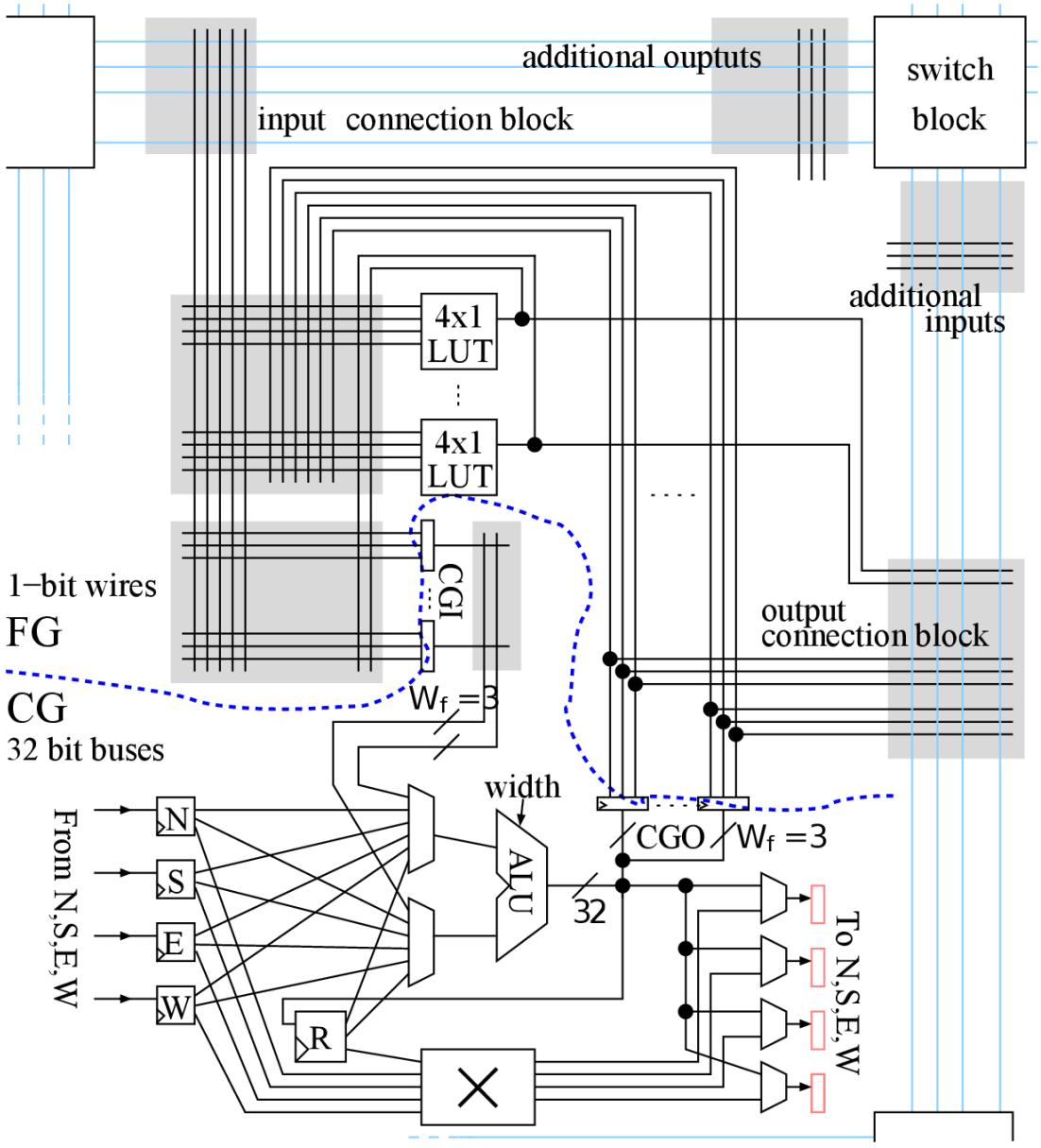
Malibu Architecture

- Add an ALU and register file



Malibu Architecture

- Add coarse-grained communication
- User clock
  - Target 20-200 MHz
- System Clock
  - 1 GHz
- Time Mux Instructions
  - SL instructions
  - User clock ≤ 1GHz / SL

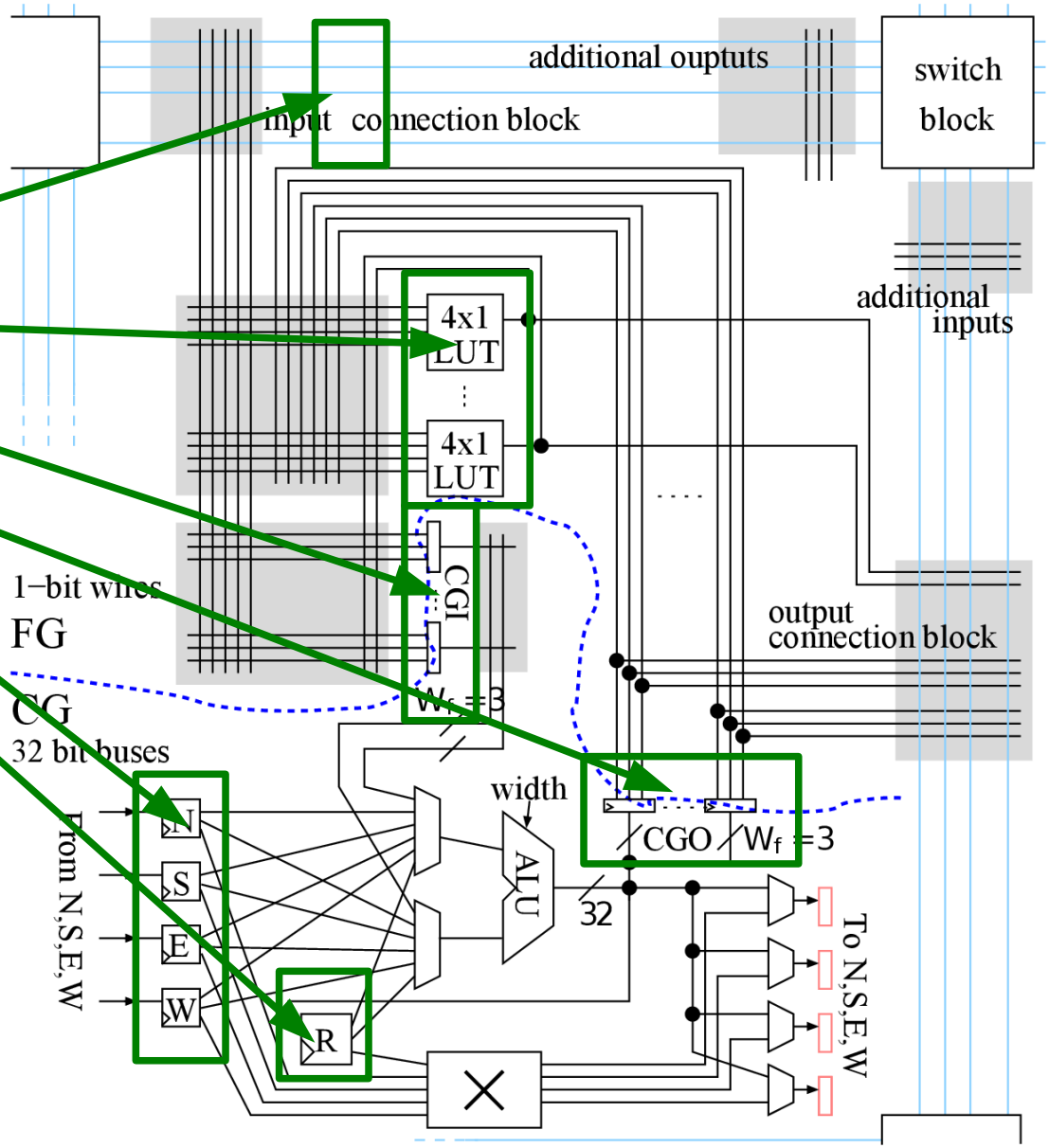


Malibu Architecture

- Parameter Values

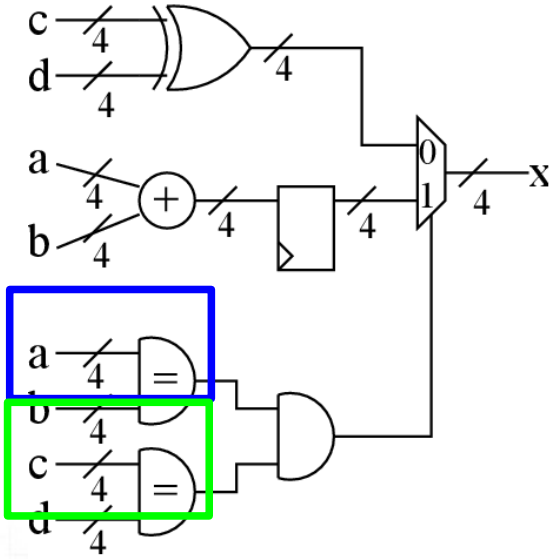
	Speed	Density
CW	120	120
N	10	10
N_cgi	16	16
N_cgo	4	4
NSEW_len	16	32
R_len	128	256
Instr_len	256	1024

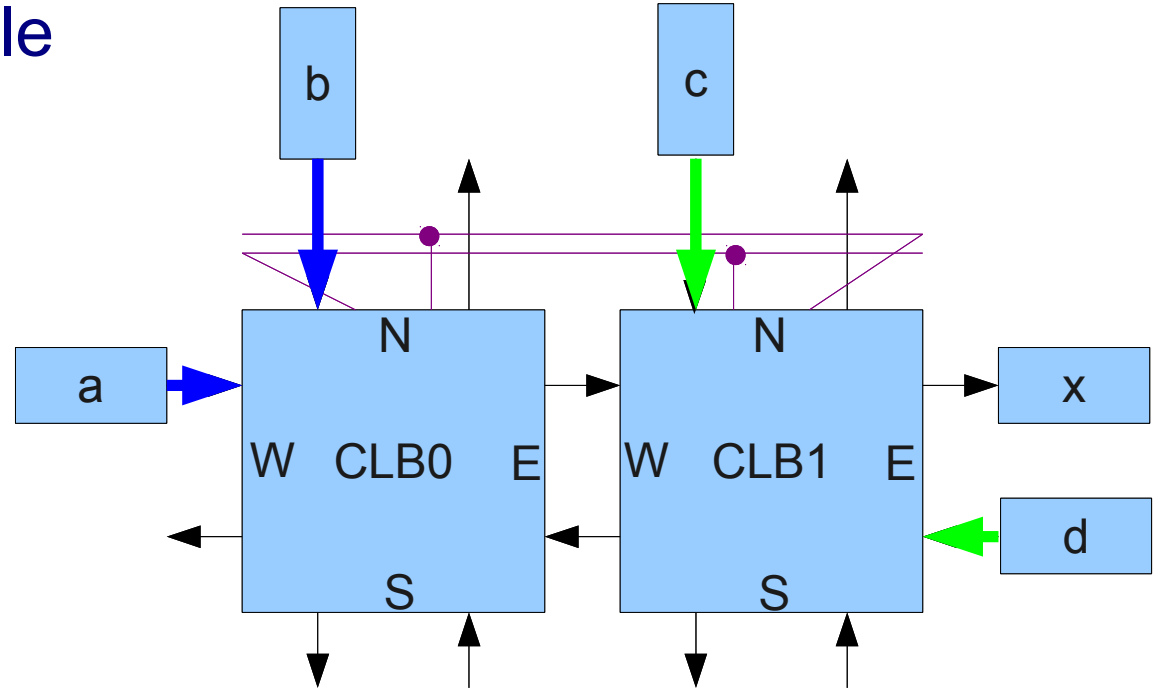
- Speed: Maximize circuit speed at minimum area
- Density: Minimize area at max speed



Circuit Example

- Circuit Execution Example

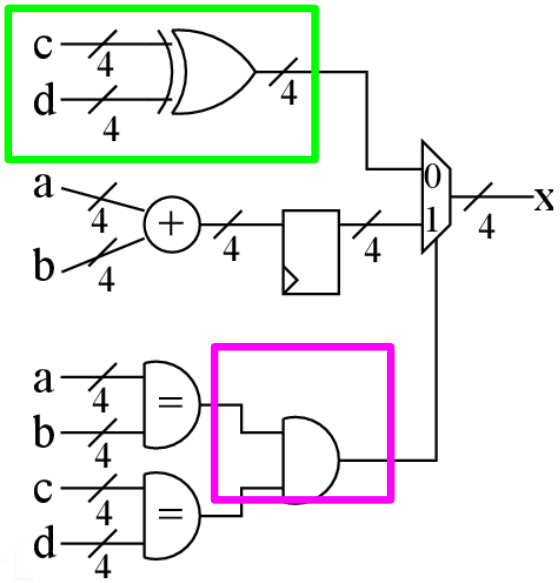


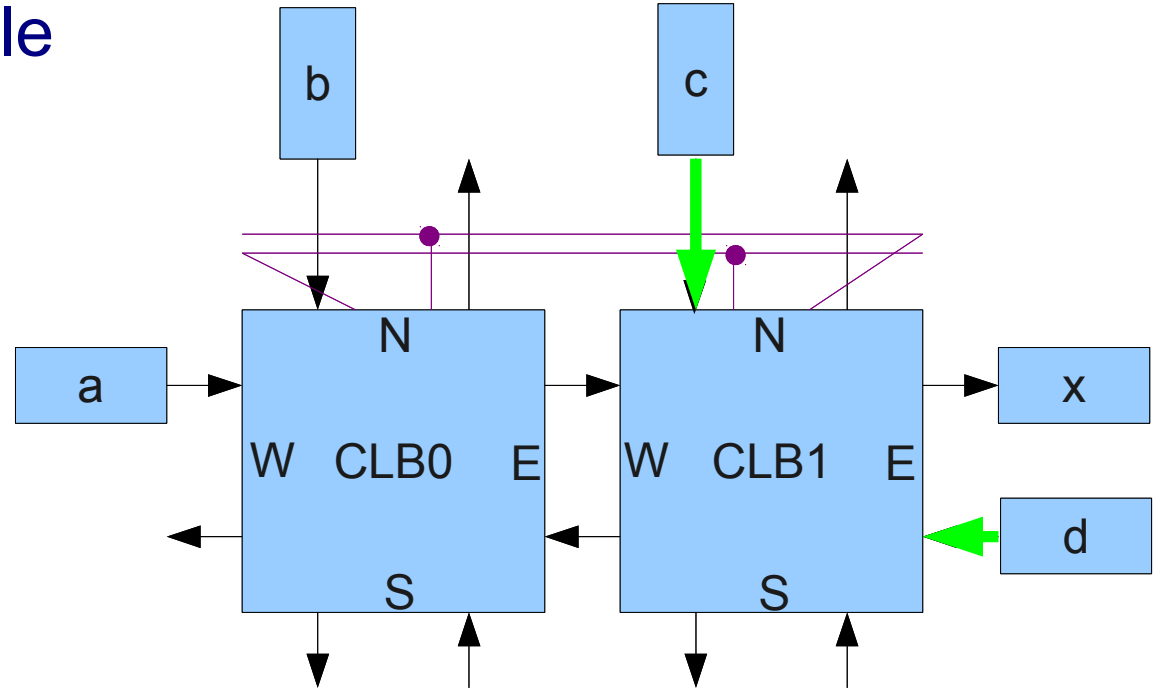


Time	CLB0 CG	CLB0 FG	CLB1 CG	CLB1 FG
0	EQ N0,W0 -> CGO0		EQ N0,E0 -> CGO0	
1	NOP	LUT	XOR N0,E0 -> R0	
2	ADD N0,W0 -> E0		MUX W0,R0,CGI0->E0	

Circuit Example

- Circuit Execution Example

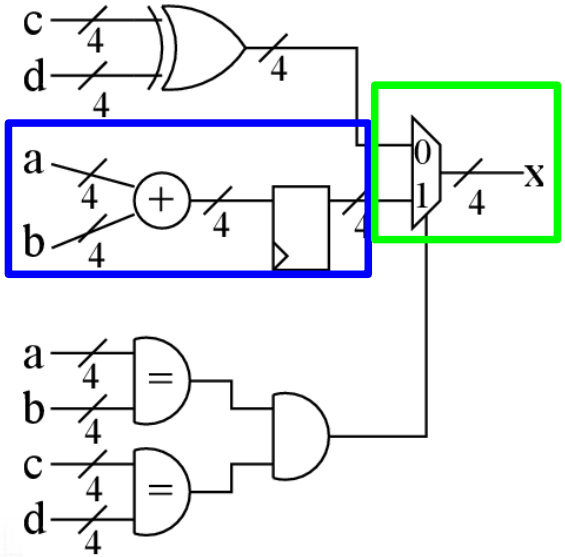


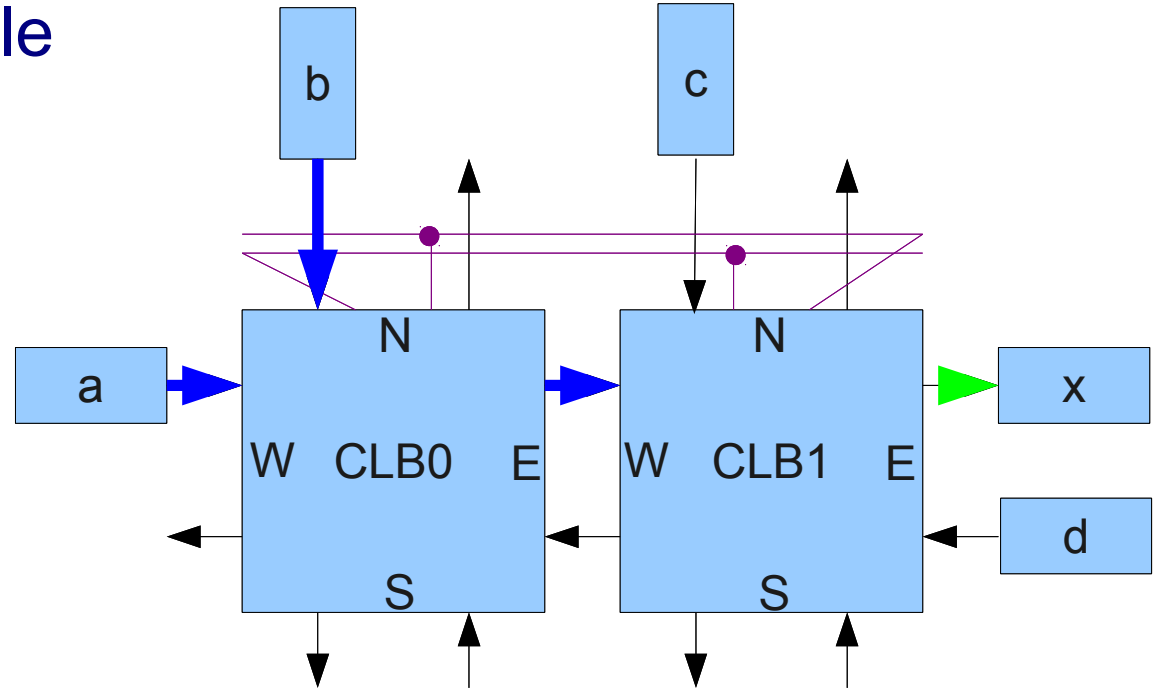


Time	CLB0 CG	CLB0 FG	CLB1 CG	CLB1 FG
0	EQ N0,W0 -> CGO0		EQ N0,E0 -> CGO0	
1	NOP	LUT	XOR N0,E0 -> R0	
2	ADD N0,W0 -> E0		MUX W0,R0,CGI0->E0	

Circuit Example

- Circuit Execution Example

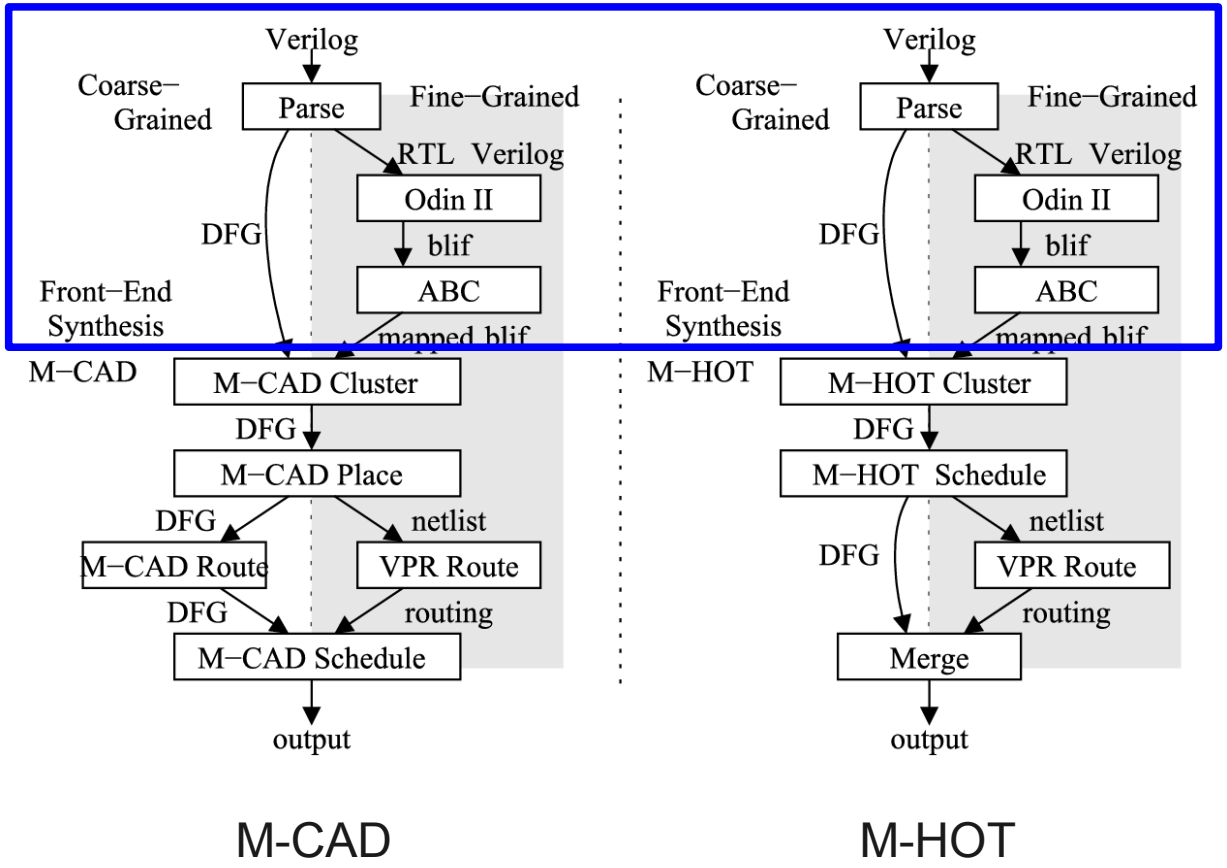




Time	CLB0 CG	CLB0 FG	CLB1 CG	CLB1 FG
0	EQ N0,W0 -> CGO0		EQ N0,E0 -> CGO0	
1	NOP	LUT	XOR N0,E0 -> R0	
2	ADD N0,W0 -> E0		MUX W0,R0,CGI0->E0	

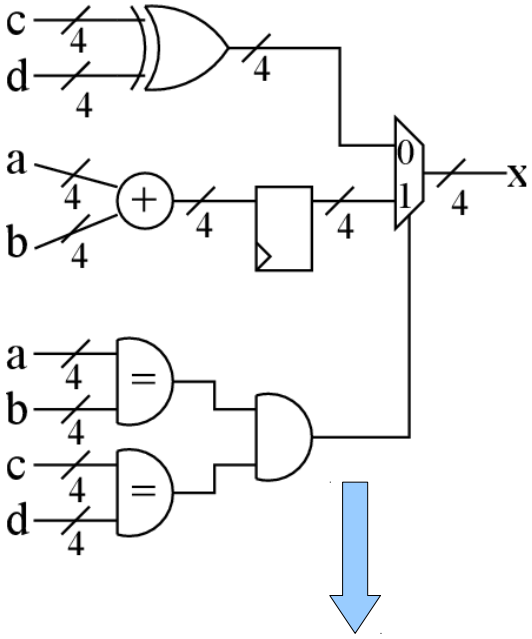
Overview

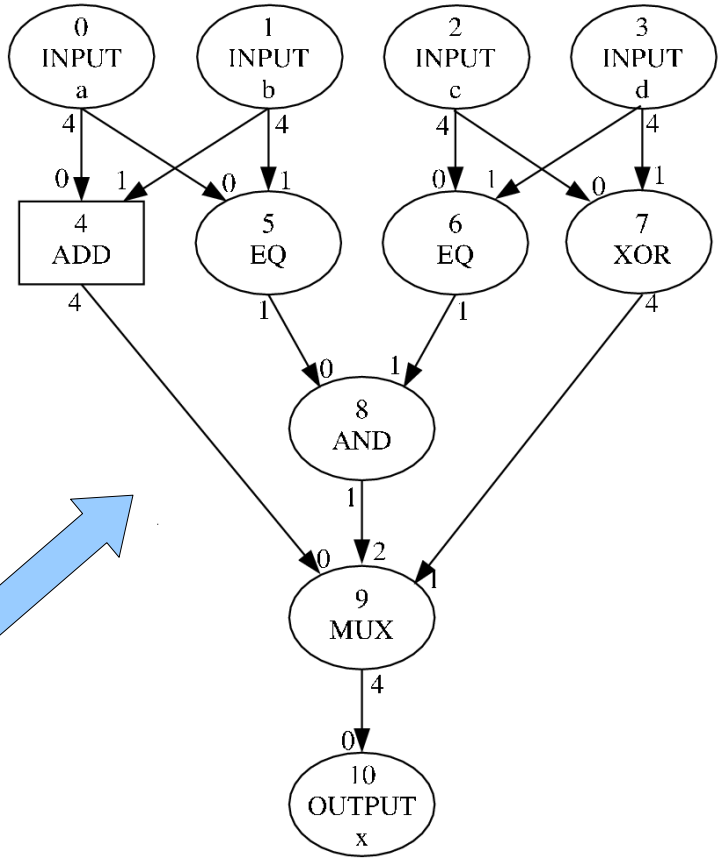
- Motivation
- Malibu Architecture
- **Front-End Synthesis**
- M-CAD tool flow
- M-HOT tool flow
- Conclusions



Synthesis

- Example



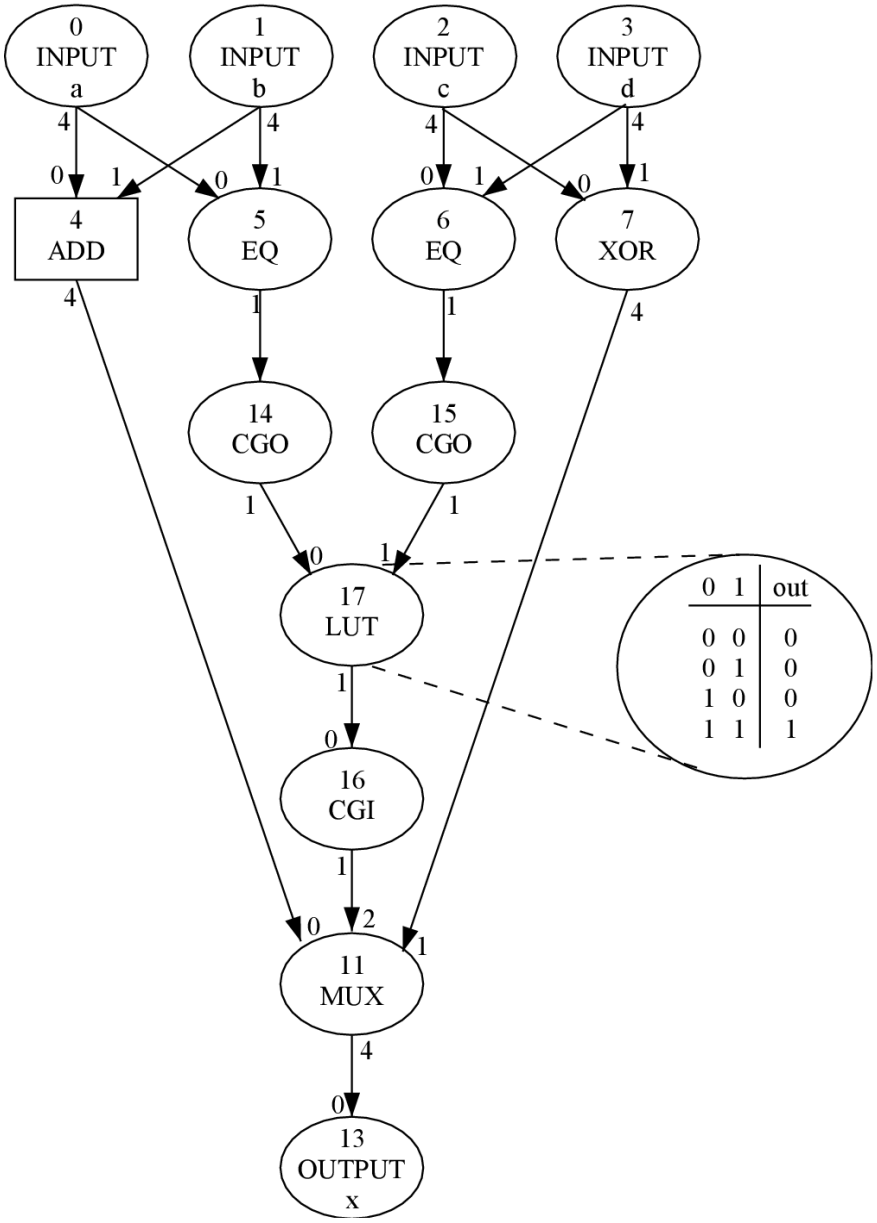



```
assign c1 = (a == b) ? 1'b1 : 1'b0;
assign c2 = (c == d) ? 1'b1 : 1'b0;
assign t2 = c ^ d;
assign x = (c1 & c2) ? t1 : t2;

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

Front-End Synthesis

- Parse
  - Modified Verilator to build CDFG
    - OdinII incomplete
    - Verilator uses words
  - Apply word-level optimizations
- Coarse-Grained Synthesis
  - Map to Malibu architecture
- Fine-Grained Synthesis
  - Synthesize FG logic to LUTs
  - Use OdinII and ABC



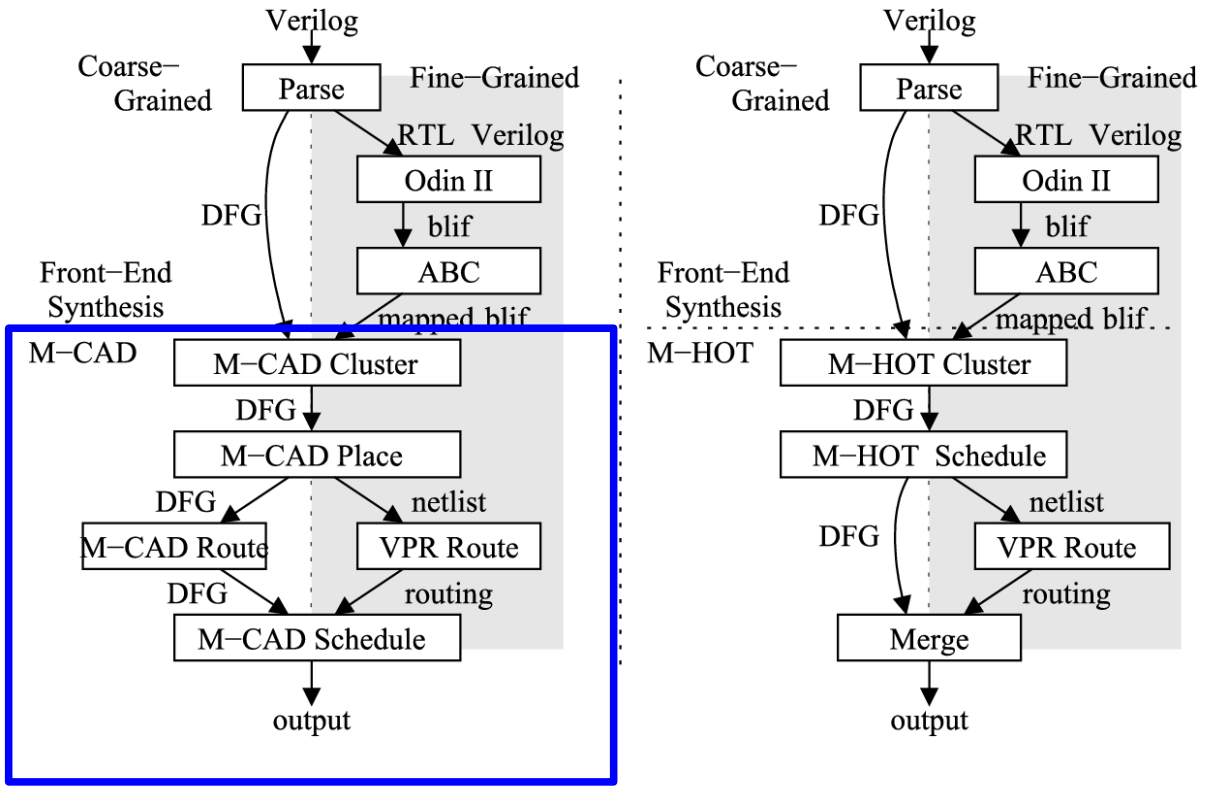
Front-End Synthesis

- Results for Front-End Synthesis
  - Works well in most cases
    - Very well for CG circuits
    - Some circuits have far too many nodes, needs optimizations
  - CG only
    - No fine-grained signals
  - Good
    - FG only used for control
  - Bad
    - Uses constructs that map poorly to Malibu

	Circuit	ALMs	Nodes
CG only	me	5148	5954
	fft16	6412	2120
	chem	3526	568
	fft8	2075	800
	honda	1216	249
	mcm	1057	232
	wang	797	134
	pr	646	176
Good	ac97_ctrl	1254	4911
	aes core	1154	3380
	dir	1150	884
	spi	488	664
	pci_master	137	957
Bad	ethernet	6868	9693
	wb_conmax	5349	17917
	dma	1714	18514
	tv80	850	12186
	jpeg enc	791	4486
	systemcaes	716	3043
	des	298	4114
	systemcdes	237	1688

Overview

- Motivation
- Malibu Architecture
- Front-End Synthesis
- **M-CAD tool flow**
- M-HOT tool flow
- Conclusions

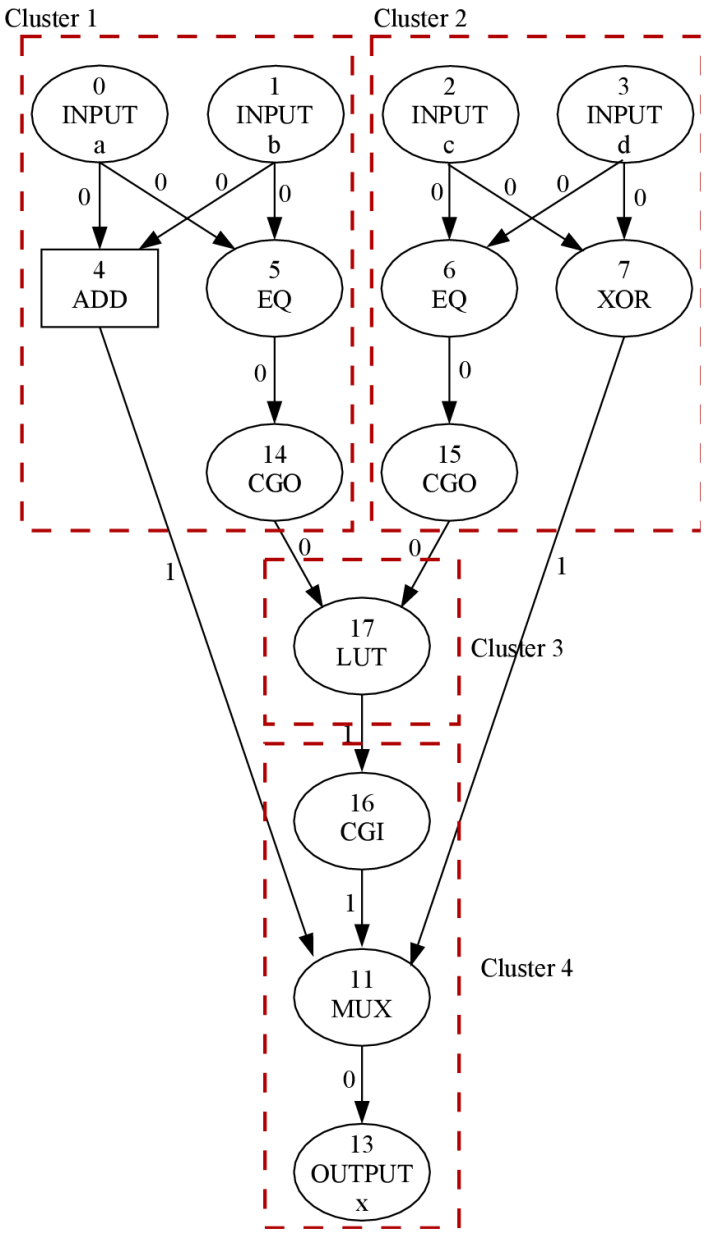


M-CAD

- Approach based on FPGA-CAD
  - Separate clustering, placement, routing, scheduling
- Advantages
  - Fast synthesis time
  - Can target a specific number of CLBs
  - Uses traditional FPGA-CAD algorithms
- Disadvantages
  - Lack of future information is a more of a problem in placement

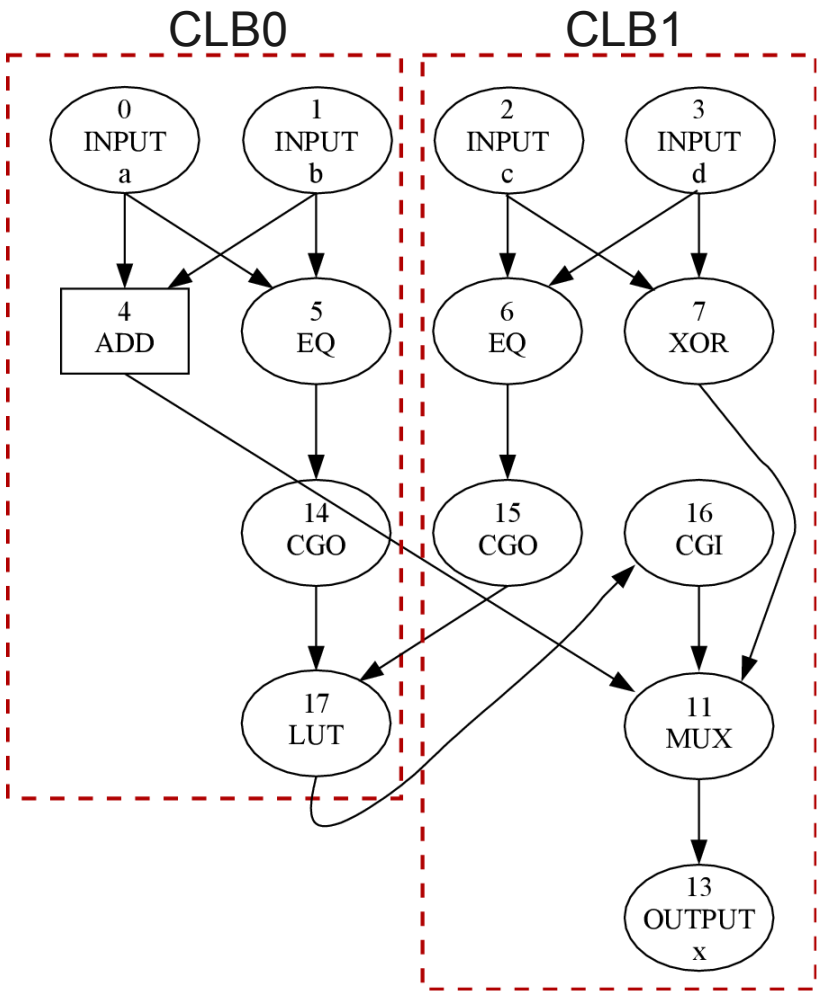
M-CAD

- M-CAD Cluster
  - Reduce problem size for placement
  - Objectives
    - Minimize number of communications
    - Balance operations
  - Create 2 clusters per CLB
  - Use hMETIS (recursive bisection)
    - Connect some nodes with high weights
      - CGO+source
      - CGI+sink
      - LOAD/STORE



M-CAD

- M-CAD Place
  - Assign clusters to CLBs
  - Minimize critical path
  - Use VPR's timing driven placement algorithm (annealing)
    - Approx model for placement
    - Timing delay changed to an integer
      - allow mixing of routing networks
      - CG = mh
      - FG = ceil(mh/10)
      - 10 = number of hops a FG signal can travel in a 1 GHz cycle



M-CAD

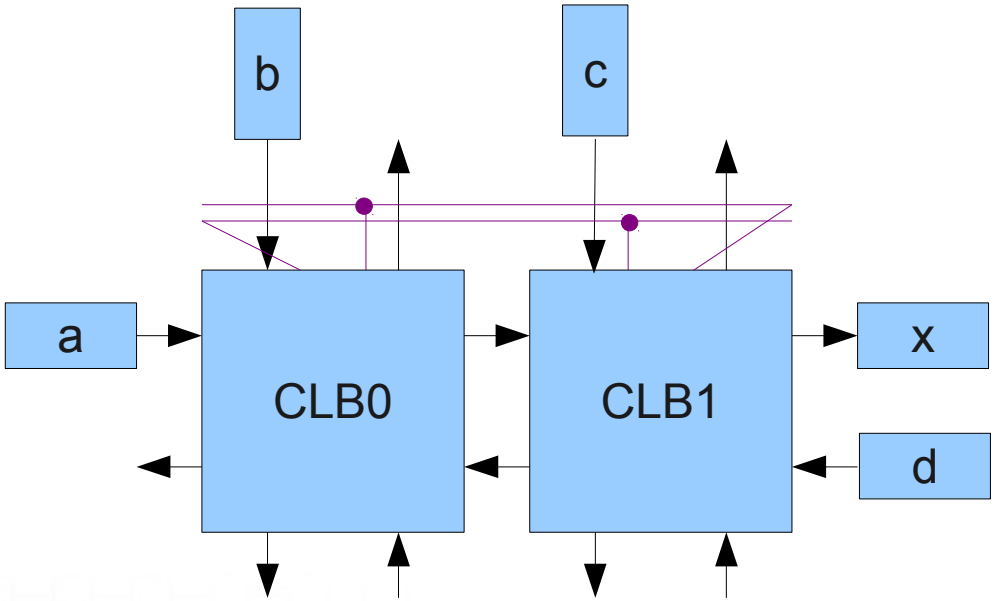
- Fine-Grained Routing
  - FG nets passed to VPR for routing
    - Netlist must include CG objects too for criticality
    - all cg nets marked DNR
  - Architecture is based on iFAR (n10k04l04) 65nm arch
    - Length-four wires changed to length-one, without changing timing to over-compensate for CG
  - Elmore delays and FG routing solution merged back into CDFG
- Coarse-Grained Routing
  - Horizontal-then-vertical routing strategy (O(n))
  - Ignore congestion and conflicts

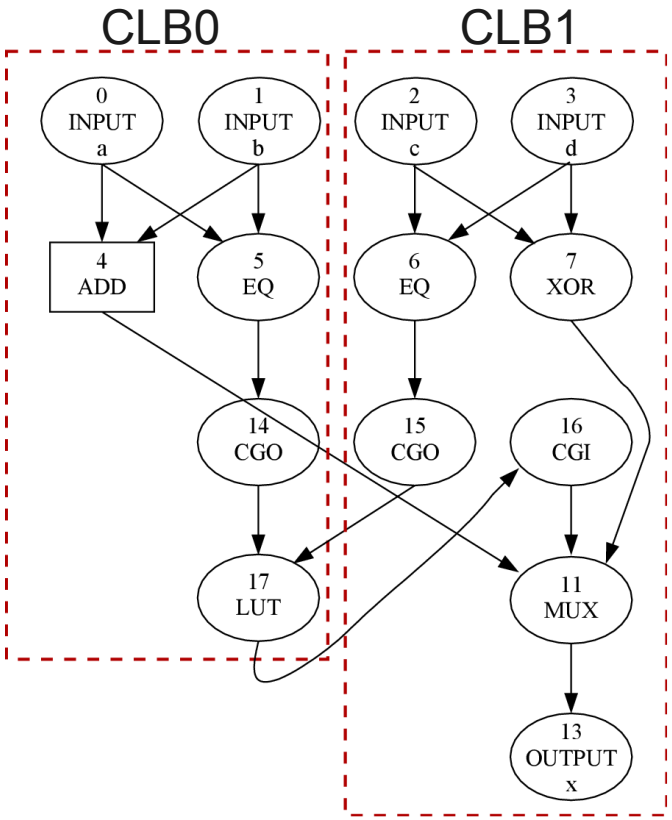
M-CAD

- Schedule
  - Input is netlist and criticality from placement
  - Assign operations to timeslots in each CLB
  - Based on list scheduling
    - Nodes ranked by criticality
  - Coarse-grained routing conflicts
    - Delay less-critical signal using a “hold slot”

M-CAD

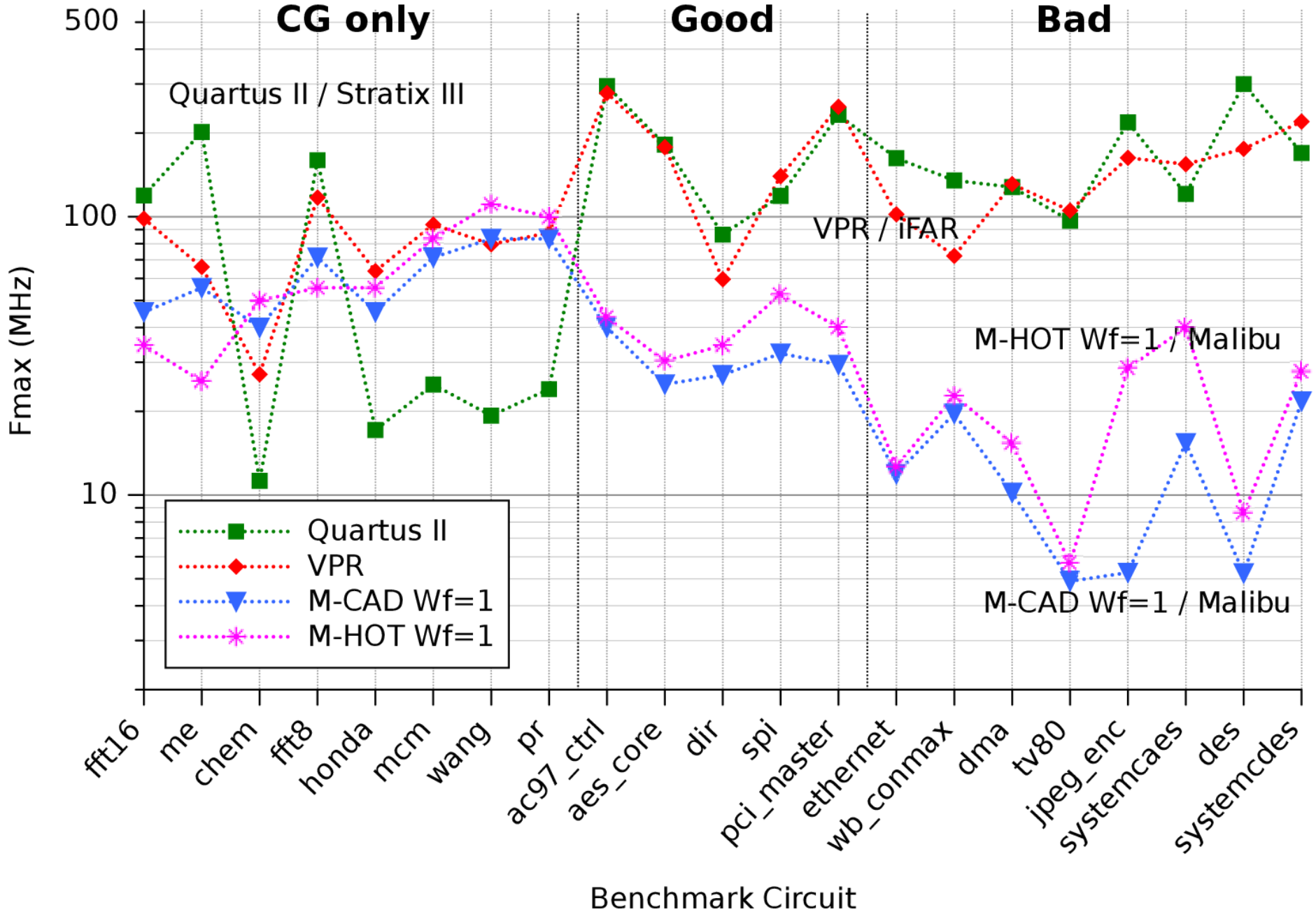
- Final schedule



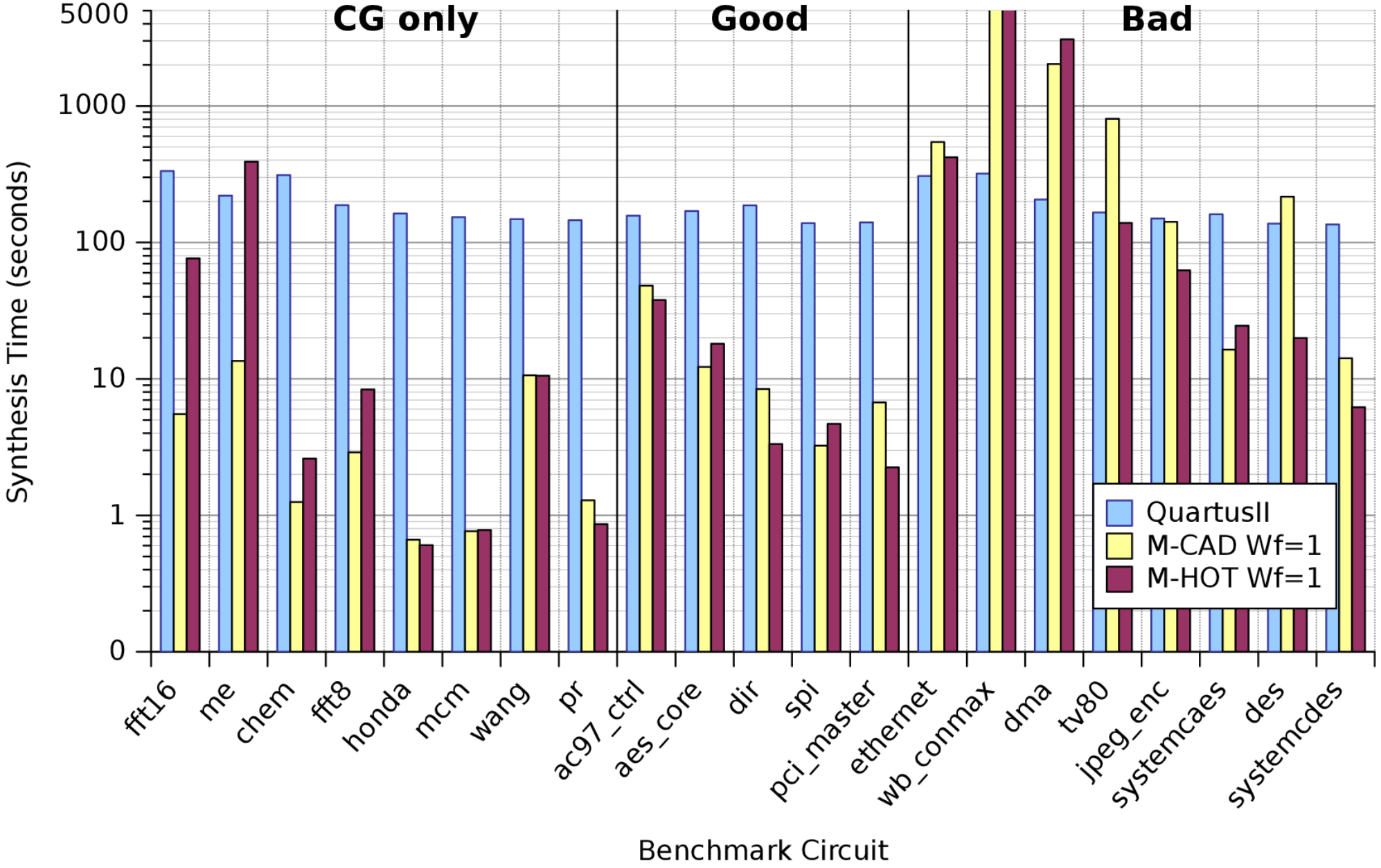


Time	CLB0 CG	CLB0 FG	CLB1 CG	CLB1 FG
0	EQ N0,W0 -> CGO0		EQ N0,E0 -> CGO0	
1	NOP	LUT	XOR N0,E0 -> R0	
2	ADD N0,W0 -> E0		MUX W0,R0,CGI0->E0	

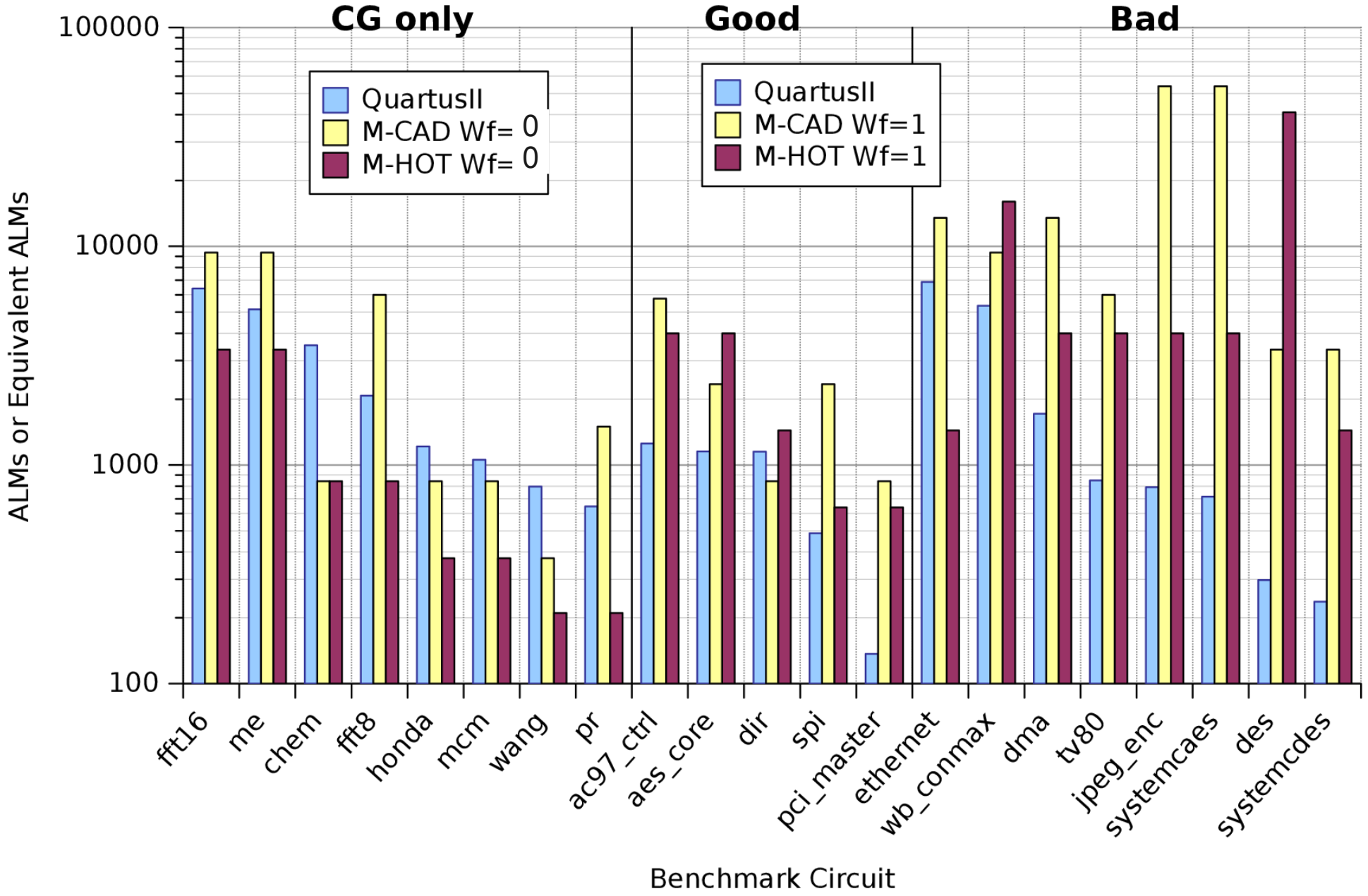
M-CAD Results -- Fmax



M-CAD Results – Synthesis Time



M-CAD Results -- Area



M-CAD

- Results (compared to Quartus II / Stratix III)
  - CG Only

	Wf=0
Synthesis Time Speedup:	77.0x
User Clock Speed:	1.45x
Density:	0.69x (0.99x)

  - Good Only

	Wf=0
Synthesis Time Speedup:	36.1x
User Clock Speed:	0.14x
Density:	0.22x (0.31x)

  - Bad Only

	Wf=0
Synthesis Time Speedup:	5.07x
User Clock Speed:	0.05x
Density:	0.095x

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

M-CAD

- Results (compared to Quartus II / Stratix III)
  - CG Only

	Wf=0
Synthesis Time Speedup:	77.0x
User Clock Speed:	1.45x
Density:	0.69x (0.99x)

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

  - Good Only

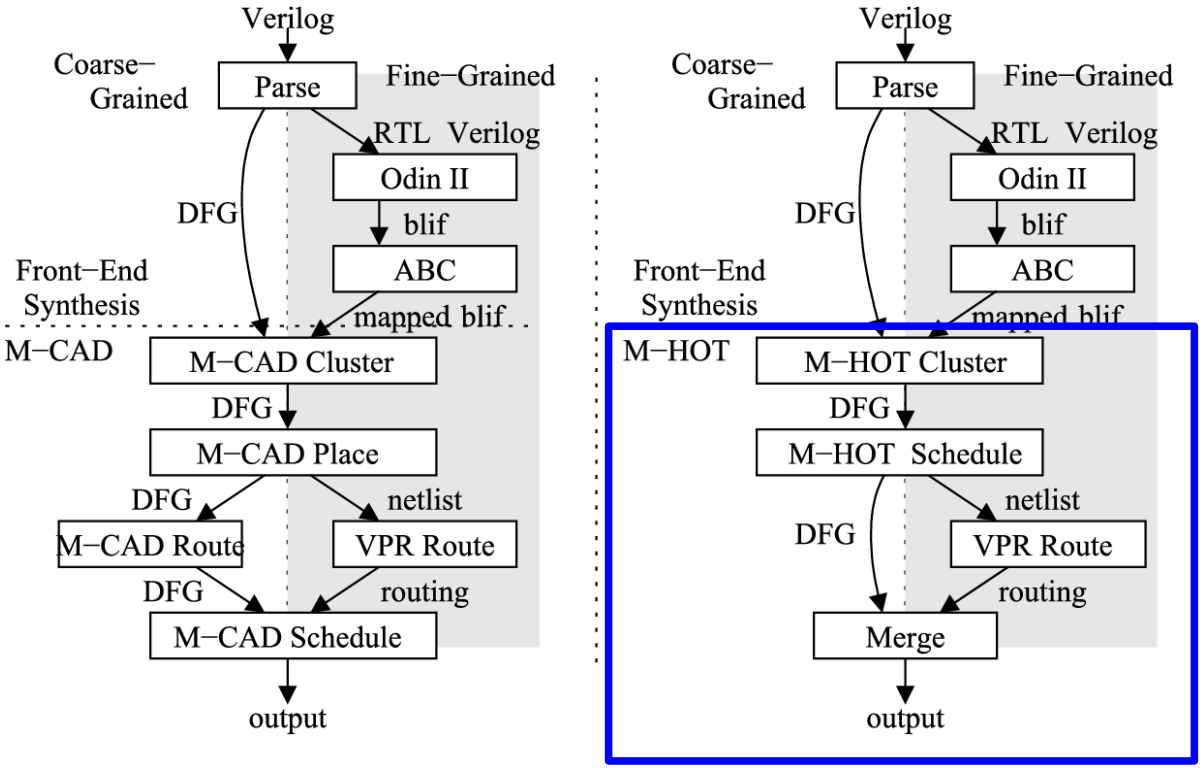
	Wf=0	Wf=1
Synthesis Time Speedup:	36.1x	15.5x
User Clock Speed:	0.14x	0.18x
Density:	0.22x (0.31x)	0.36x

  - Bad Only

	Wf=0	Wf=1
Synthesis Time Speedup:	5.07x	0.56x
User Clock Speed:	0.05x	0.06x
Density:	0.095x	0.093x

Overview

- Introduction
- Malibu Architecture
- Front-End Synthesis
- M-CAD tool flow
- **M-HOT tool flow**
- Conclusions

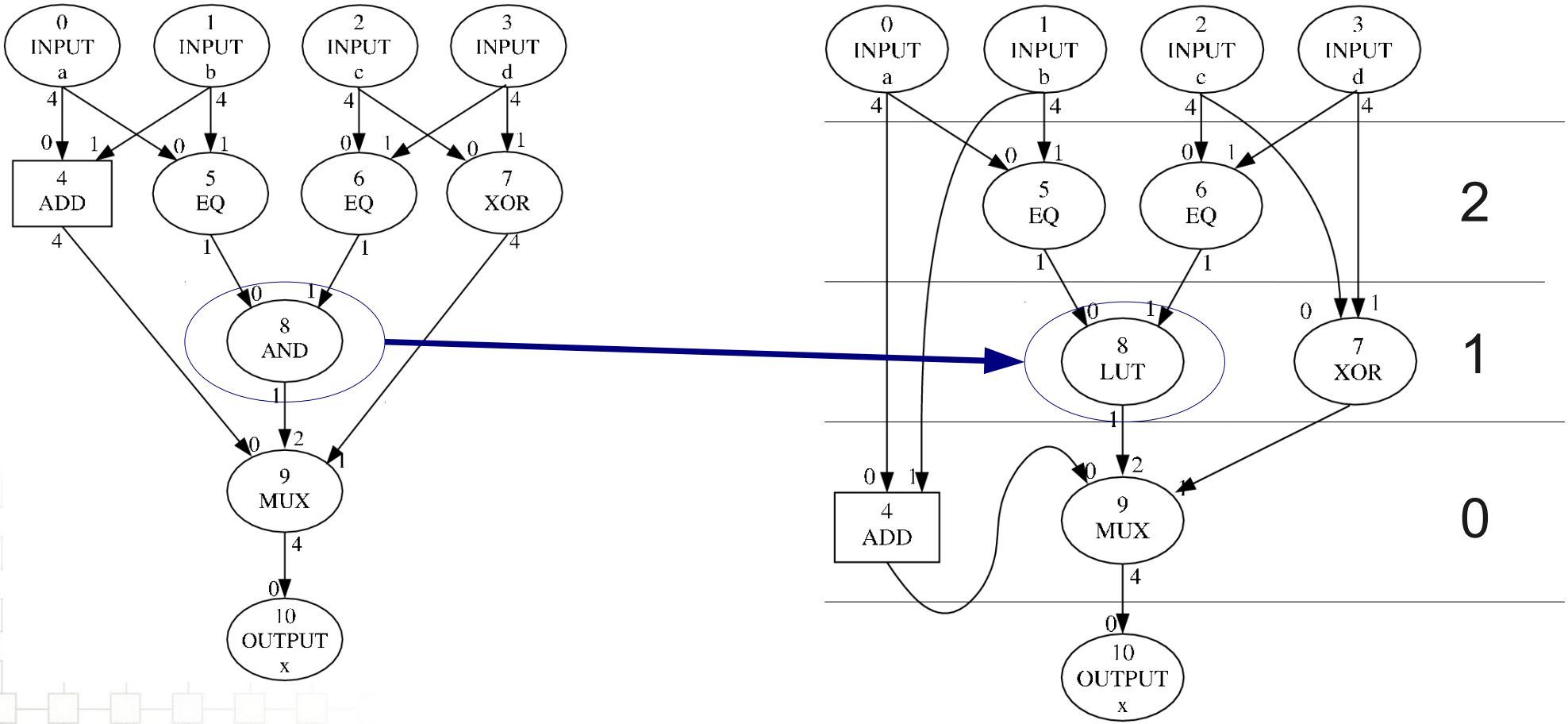


M-HOT

- Based on a modulo graph scheduler from Park et al.
  - Integrated placement, routing, scheduling
  - Divides problem into heights and anneals each height
- Advantages
  - Better quality than separate flows
  - Can target any number of CLBs
- Disadvantages
  - Slower, especially when ALAP tree is small or imbalanced

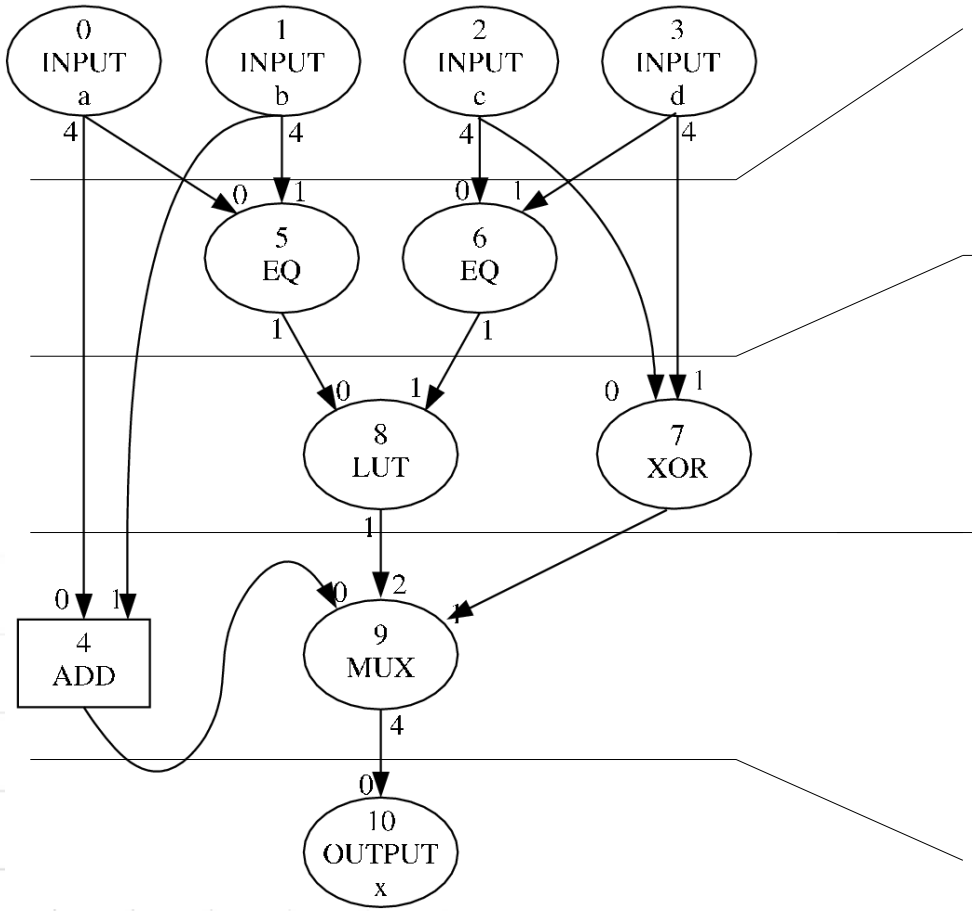
M-HOT

- CDFG and ALAP output of Front-End Synthesis



M-HOT

- 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

M-HOT vs. M-CAD

- Results (compared to Quartus II / Stratix III)
  - CG Only

	CAD Wf=0
Synthesis Time Speedup:	77.0x
User Clock Speed:	1.45x
Density:	0.69x (0.99x)

  - Good Only

	CAD Wf=0	CAD Wf=1
Synthesis Time Speedup:	36.1x	15.5x
User Clock Speed:	0.14x	0.18x
Density:	0.22x (0.31x)	0.36x

  - Bad Only

	CAD Wf=0	CAD Wf=1
Synthesis Time Speedup:	5.07x	0.56x
User Clock Speed:	0.05x	0.06x
Density:	0.095x	0.093x

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

M-HOT vs. M-CAD

- Results (compared to Quartus II / Stratix III)
  - CG Only

	CAD Wf=0	HOT Wf=0
Synthesis Time Speedup:	77.0x	30.9x
User Clock Speed:	1.45x	2.71x
Density:	0.69x (0.99x)	2.74x

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

  - Good Only

	CAD Wf=0	CAD Wf=1
Synthesis Time Speedup:	36.1x	15.5x
User Clock Speed:	0.14x	0.18x
Density:	0.22x (0.31x)	0.36x

  - Bad Only

	CAD Wf=0	CAD Wf=1
Synthesis Time Speedup:	5.07x	0.56x
User Clock Speed:	0.05x	0.06x
Density:	0.095x	0.093x

M-HOT vs. M-CAD

- Results (compared to Quartus II / Stratix III)
  - CG Only

	CAD Wf=0	HOT Wf=0
Synthesis Time Speedup:	77.0x	30.9x
User Clock Speed:	1.45x	2.71x
Density:	0.69x (0.99x)	2.74x

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

  - Good Only

	CAD Wf=0	CAD Wf=1	HOT Wf=0
Synthesis Time Speedup:	36.1x	15.5x	7.62x
User Clock Speed:	0.14x	0.18x	0.19x
Density:	0.22x (0.31x)	0.36x	0.40x (0.65x)

  - Bad Only

	CAD Wf=0	CAD Wf=1	HOT Wf=0
Synthesis Time Speedup:	5.07x	0.56x	0.27x
User Clock Speed:	0.05x	0.06x	0.10x
Density:	0.095x	0.093x	0.51x

M-HOT vs. M-CAD

- Results (compared to Quartus II / Stratix III)
  - CG Only

	CAD Wf=0	HOT Wf=0
Synthesis Time Speedup:	77.0x	30.9x
User Clock Speed:	1.45x	2.71x
Density:	0.69x (0.99x)	2.74x

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

  - Good Only

	CAD Wf=0	CAD Wf=1	HOT Wf=0	HOT Wf=1
Synthesis Time Speedup:	36.1x	15.5x	7.62x	20.9x
User Clock Speed:	0.14x	0.18x	0.19x	0.27x
Density:	0.22x (0.31x)	0.36x	0.40x (0.65x)	0.41x

  - Bad Only

	CAD Wf=0	CAD Wf=1	HOT Wf=0	HOT Wf=1
Synthesis Time Speedup:	5.07x	0.56x	0.27x	1.06x
User Clock Speed:	0.05x	0.06x	0.10x	0.14x
Density:	0.095x	0.093x	0.51x	0.22x

Overview

- Motivation
- Malibu Architecture
- Front-End Synthesis
- M-CAD tool flow
- M-HOT tool flow
- **Conclusions**

Conclusions

- Achieved Thesis Objective
  - Fast synthesis for coarse-grained circuits
  - Improved density (by trading circuit speed)
  - Improved speed (by trading density)
- New Architecture: Malibu
  - Add time-multiplexed coarse-grained resources to an FPGA
- M-CAD
  - Fast
  - Placement not ideal with time-multiplexing due to no information sharing
- M-HOT
  - Better circuit speed, better density
  - Slower synthesis

Future Work

- Improve M-CAD and M-HOT (focus of thesis)
  - At best 2x-3x performance improvement might exist
    - Assumes routing delays = 0, perfect placement, infinite resources
  - Focus on placement

- Improvements to Architecture and Front-End Synthesis
  - Architecture: FG/CG interface can better support signal mixing
  - Architecture: Multiple ALUs, minimize memory ports, better memory technology
  - Synthesis: Parallel cases, better logic optimizations
  - Synthesis: More architecture aware

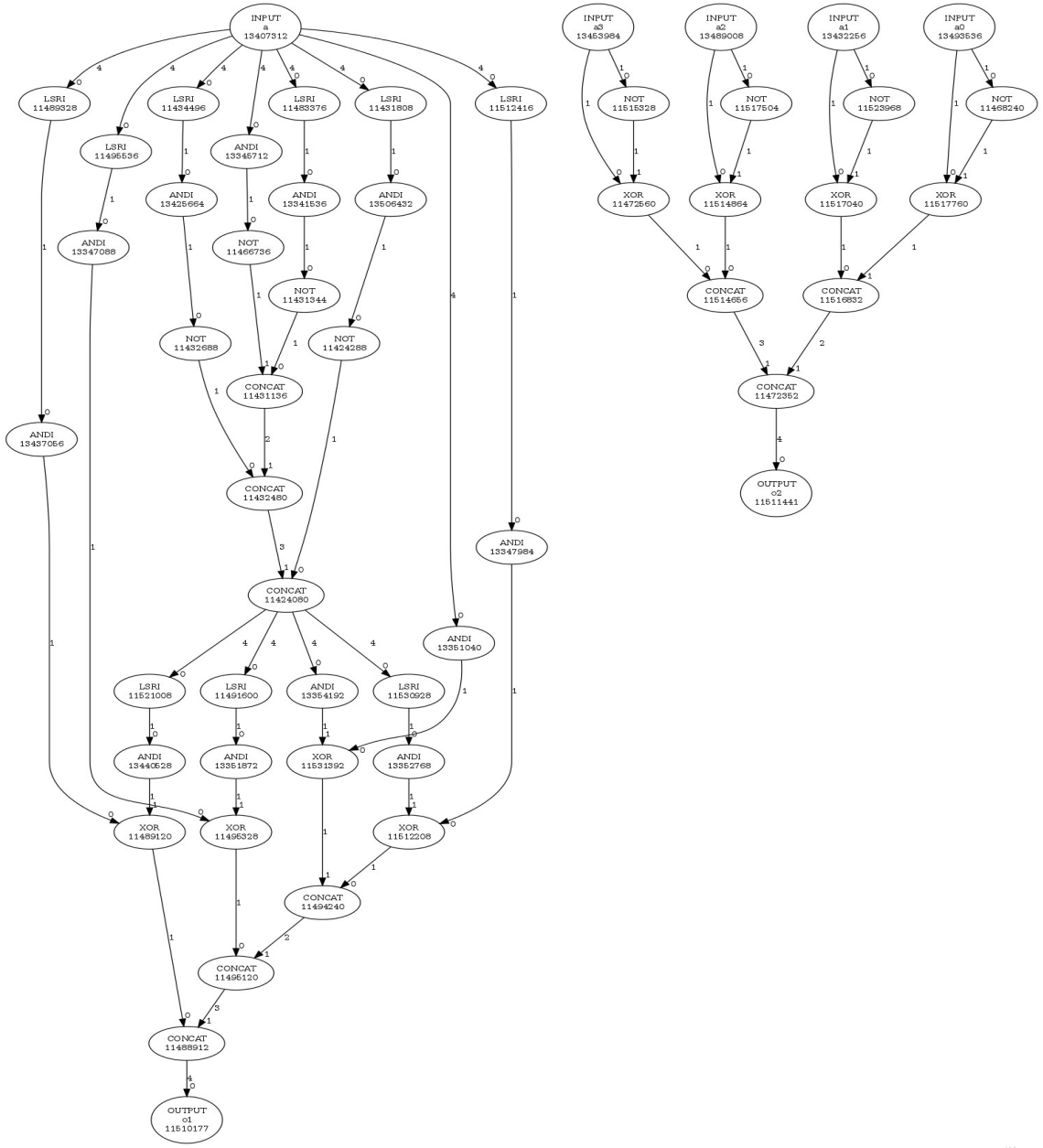
Publications

- D. Grant and G. G. Lemieux. A spatial computing architecture for implementing computational circuits. In *Proc. MNRC*, pages 41-44, Oct. 2008.
  - Malibu Architecture (coarse grained only)
- D. Grant, G. Smecher, G. G. Lemieux, and R. Francis. Rapid synthesis and simulation of computational circuits in an MPPA. In *Proc. FPT*, pages 151-158, Dec. 2009.
  - M-CAD CAD tool flow (coarse grained only)
- D. Grant, G. Smecher, G. G. Lemieux, and R. Francis. Rapid synthesis and simulation of computational circuits in an MPPA. To Appear In *Journal of Signal Process Systems*, 15 pages, 2010.
  - M-CAD experimentation, max. theoretical clock speed, CLB area calculation
- D. Grant, C. Wang, G. G. Lemieux. A CAD Framework for MALIBU: An FPGA with Time-multiplexed Coarse-Grained Elements. To Appear In *Proc. Field-Programmable Gate Arrays (FPGA)*, 10 pages, 2011.
  - Arch and M-CAD fine-grained
  - M-HOT

EOF

Bad Constructs

- A bus to store individual bits
  - wire [3:0] flags;
  - Then using each flag as though it was a separate wire
  - Generates shift/mask for both reads and writes



Bad Constructs

- Creating multi-bit operations manually instead of using a high-level operator
  - Use +,-,*,<<,>>, ...
  - Don't build an adder manually

Bad Constructs

- A “case” LUT
  - case (in)
    - 8'b0000000: out <= 4'b0000;
    - 8'b0000001: out <= 4'b1010;
    - ...
  - Should be:
    - wire [3:0] lut [7:0];
    - Initial values in an initial block

Bad Constructs

- Series of case statements in an FSM
  - Need support for synopsys full parallel case
  - ex. tv80 ~190 states in a sequence