



High-Level Power Estimation and Low-Power Design Space Exploration for FPGAs

**Deming Chen⁺, Jason Cong^{*}, Yiping Fan^{*},
Zhiru Zhang^{*}**

**+ : Department of ECE, University of Illinois, Urbana-
Champaign**

*** : Computer Science Department, University of California,
Los Angeles**

**This work is partially sponsored by NSF under grant CCF-0306682 and Altera, Magma, and
Xilinx under the MICRO Program.**

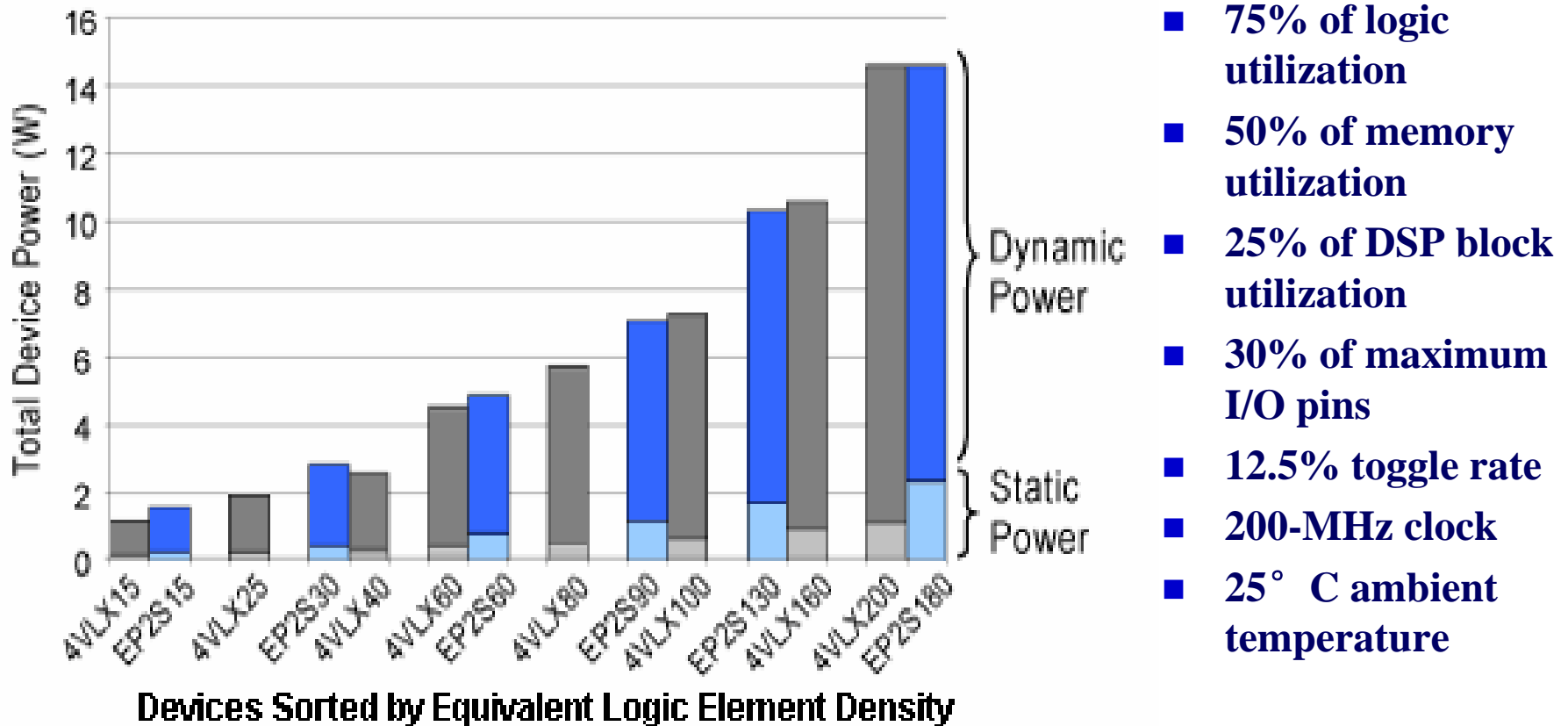


Outline

- **Background and motivation**
- **Related work**
- **Problem formulation**
- **Power estimation**
- **Simultaneous allocation and binding**
- **Experimental results**
- **Conclusions**

Power Consumption of FPGA Chips

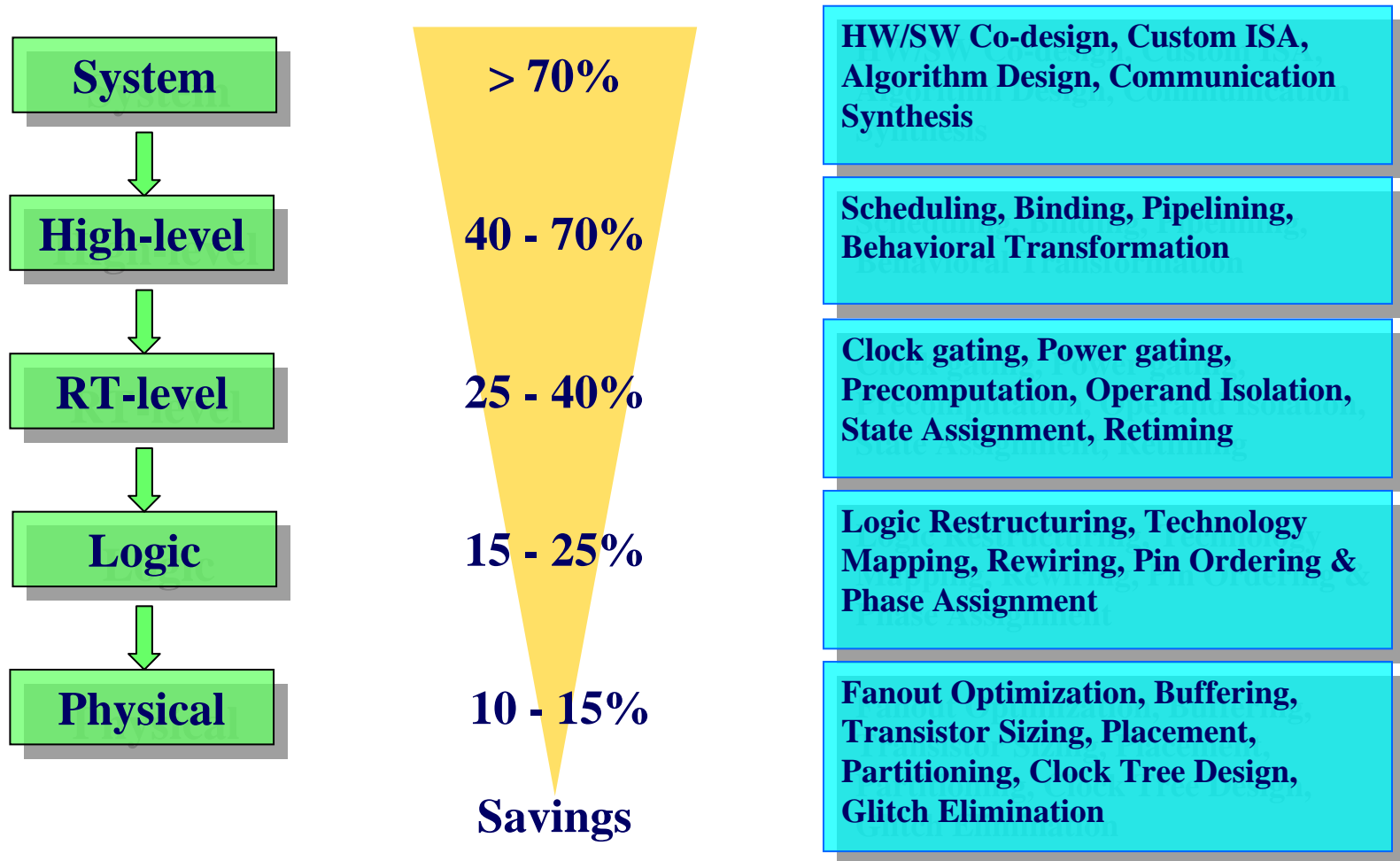
Blue bar: Altera; Gray bar: Xilinx



1.2V, 90nm, 11 layers of metal...

Source: Altera, 2005

Power Saving Opportunities



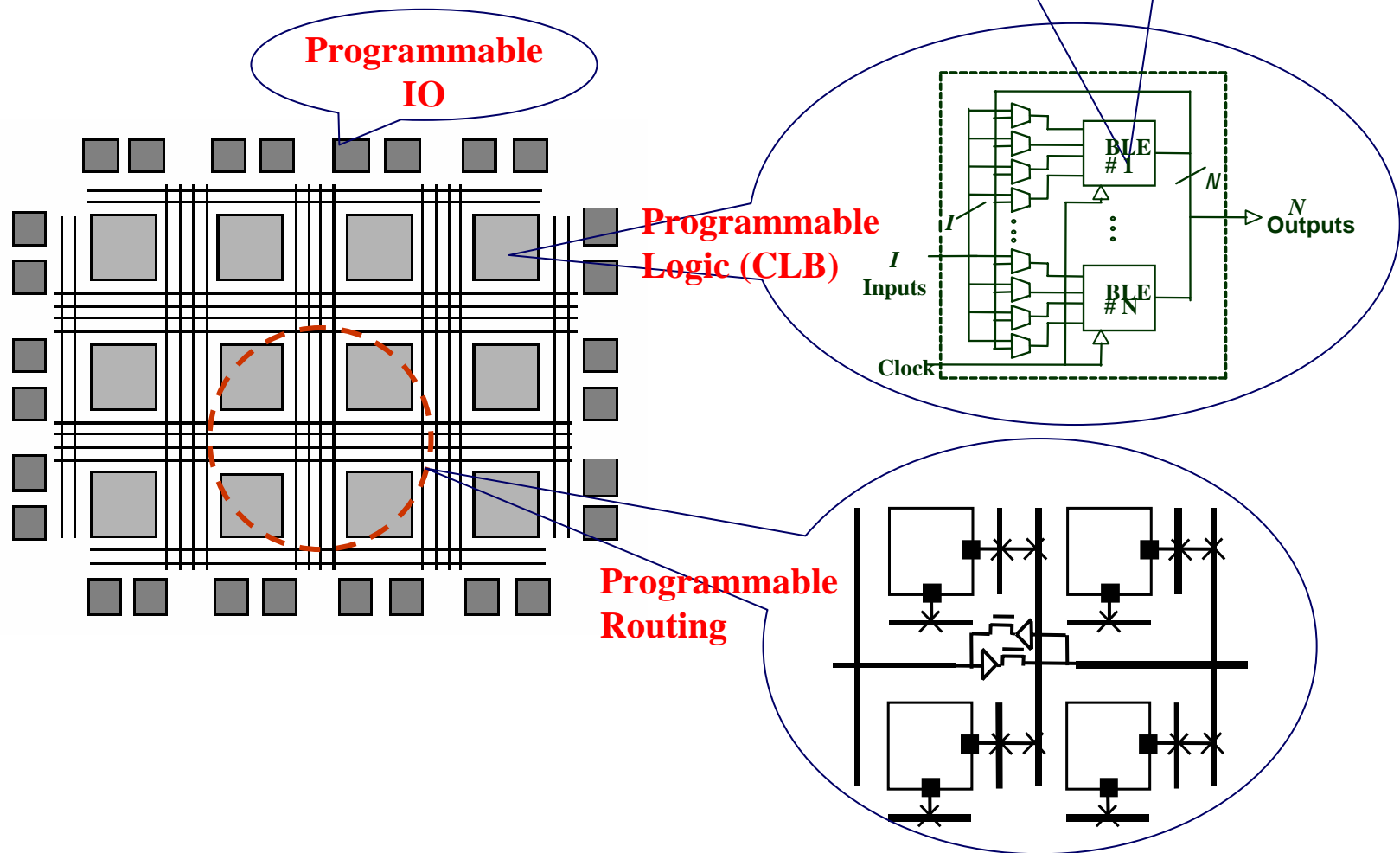
- A power-conscious design methodology addresses power at every level of the design hierarchy



Raise up the Design Level

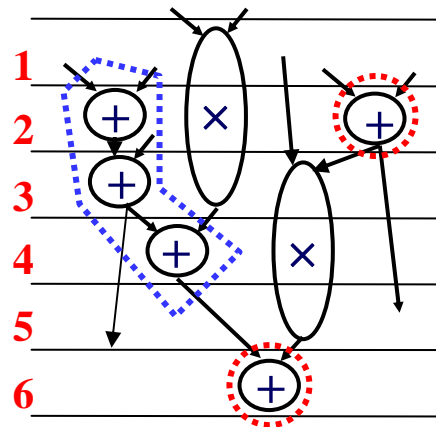
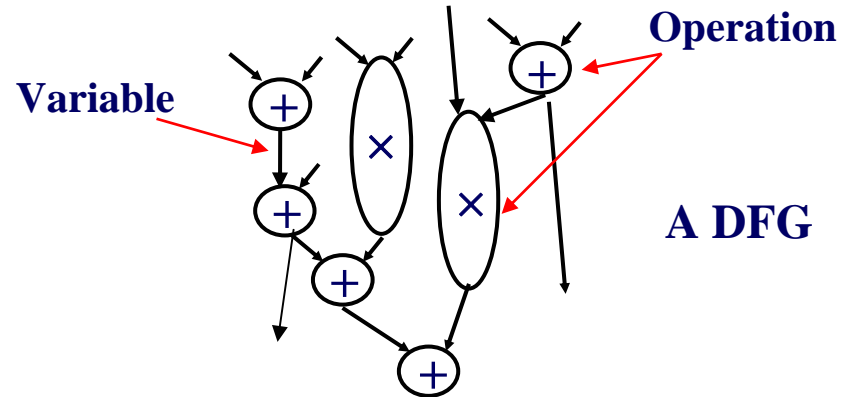
- **Higher design productivity**
- **Better quality of result**
- **Fast design space exploration**

A Typical FPGA Architecture

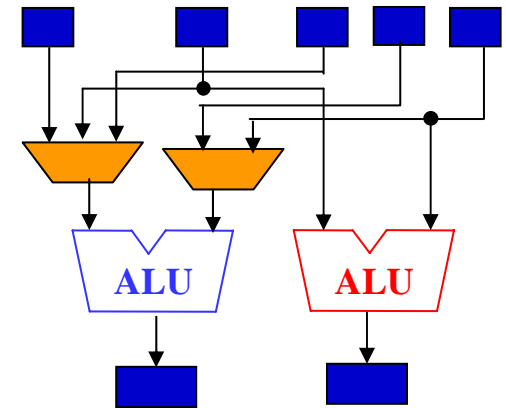


Subtasks in Behavior-Level Synthesis

- **Scheduling** determines when an operation will be executed
- **Allocation** determines number of instances of each type of resources
- **Binding** binds operations, variables, or data-transfers to the resources



Scheduling & allocation



Binding

Related Work

- **The first group solves register binding and functional unit binding separately**
 - ◆ clique partitioning [Tseng, TCAD'86]
 - ◆ weighted bipartite-matching [Huang, DAC'90]
 - ◆ network flow [Chang, DAC'95][Gebotys, DAC'97]
 - ◆ *k*-cofamily [Chen, ASPDAC'04]
- **The second group performs simultaneous functional unit and register binding globally**
 - ◆ simulated annealing [Chen, ISLPED'03][Choi, TODAES'99]
 - ◆ simulated evolution [Ly, TCAD'93]
 - ◆ ILP (integer linear programming) [Gebotys, JSSC'92][Rim, DAC'92]
- **The third group carries out simultaneous optimization one control step at a time**
 - ◆ network flow [Kim, CICC'95][Mujumdar, TCAD'96]

Outline

- **Background and motivation**
- **Related work**
- **Problem formulation**
- **Power estimation**
- **Simultaneous allocation and binding**
- **Experimental results**
- **Conclusions and future work**

Problem Formulation

- **CDFG: control data flow graph** to represent the functional behavior of the circuit
- **STG: state transition diagram** to describe the scheduling result of the circuit
- **Given:** A CDFG G and its STG G'
- **Tasks:** construct a datapath architecture, in which every functional unit is bound to a set of operations, and every register is bound to a set of dataflows.
- **Objectives:** maintain behavior correctness and optimize power and performance for the design on a target FPGA.

Challenges

- **In general**
 - ◆ **Huge design space during behavior-level synthesis**
 - ◆ **Many design parameters are interdependent**
- **Need to consider critical path delay for high performance**
- **Need to explore the correlation between power and performance**
 - ◆ **Optimize power under delay constraint**
 - ◆ **Optimize delay under power constraint**
 - ◆ **Power/delay tradeoff if possible**
- **Need an accurate high-level power estimator**

Contributions of xPlore-Power

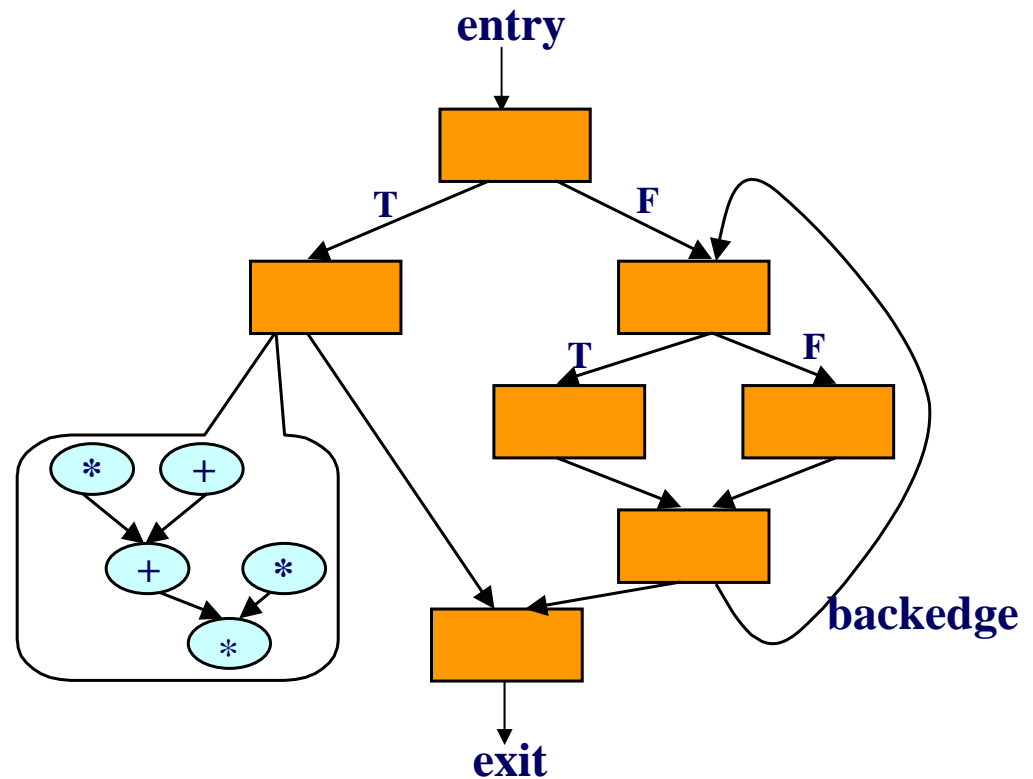
- **Set up CDFG power estimation targeting real FPGA architectures**
 - ◆ *logic elements, DSP cores, memories, ...*
- **Built a flow and evaluated FPGA high-level power estimation and optimization through a commercial gate-level power analyzer**
- **Designed a novel design space exploration engine**
 - ◆ *Form, propagate, and prune synthesis solution points for datapath generation*
 - ◆ *Generate power/delay correlation curve targeting real FPGA architectures*
- **Achieved significant amount of power and performance gain compared to a traditional synthesis algorithm**

Outline

- **Background and motivation**
- **Related work**
- **Problem formulation**
- **Power estimation**
- **Simultaneous allocation and binding**
- **Experimental results**
- **Conclusions and future work**

CDFG Simulation and Profiling

- A two-level CDFG representation
 - ◆ CFG
 - ◆ DFG
- Test vectors
 - ◆ Primary inputs
 - ◆ Global variables
- Profiling results
 - ◆ Basic block utilization ratio
 - ◆ Switching activity information on ports, nodes, memories
 - ◆ Worst case latency



Switching Activity Estimation

- Performs simulation just once at the beginning
 - ◆ computes switching activities for any legal binding without repeating simulations (based on [Bogliolo et.al. ISLPED'99])
 - ◆ Extended to support loops
- Toggle count calculation

$$C_{in}(O_i, O_{i+1}) = \sum_{j=1}^K \sum_{x=1}^B D_H(I_i^{j(x)}, I_{i+1}^{j(x)})$$

$$C_{in}(O_N, O_1) = \sum_{j=1}^K \sum_{x=1}^{B-1} D_H(I_N^{j(x)}, I_1^{j(x+1)}) + \sum_{j=1}^{K-1} D_H(I_N^{j(B)}, I_1^{(j+1)(1)})$$

- Switching activity calculation

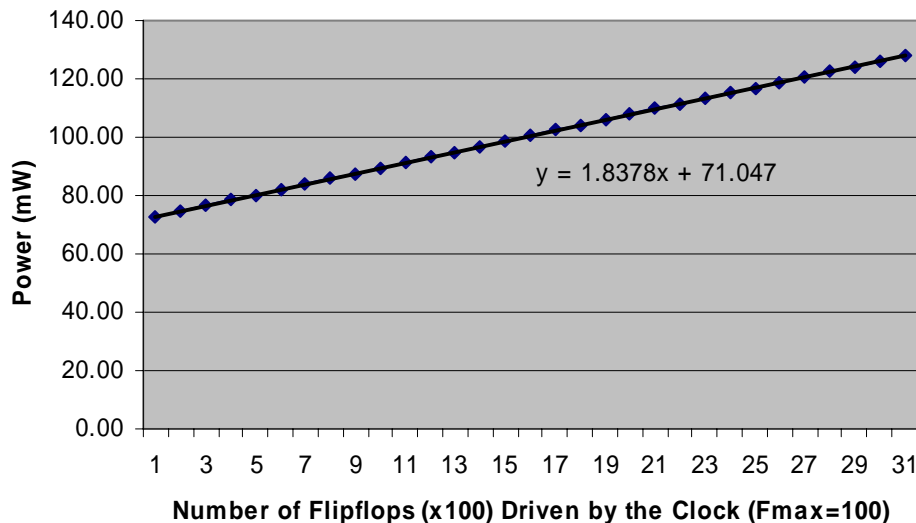
$$P_{in} = \frac{\sum_{i=1}^{N-1} C_{in}(O_i, O_{i+1}) + C_{in}(O_N, O_1)}{2 \times \text{Bit_width} \times (N \times K \times B - 1)}$$

Resource Power Estimation (1)

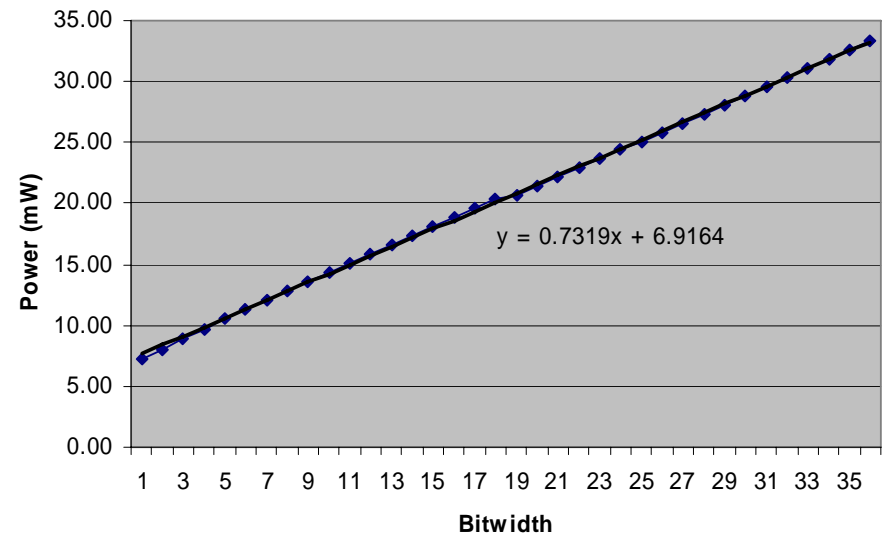
Fmax = 100; Toggle rate = 100% for Altera Stratix Devices

Elements	Num	Est'ed P (mW)
LE	1	0.12
LE w/ Carry	1	0.04
DSP	Per output	1.23
I/O	1	19.31

Global Clock Network Power for Altera's Stratix



M4K Power vs. Bitwidth



Resource Power Estimation (2)

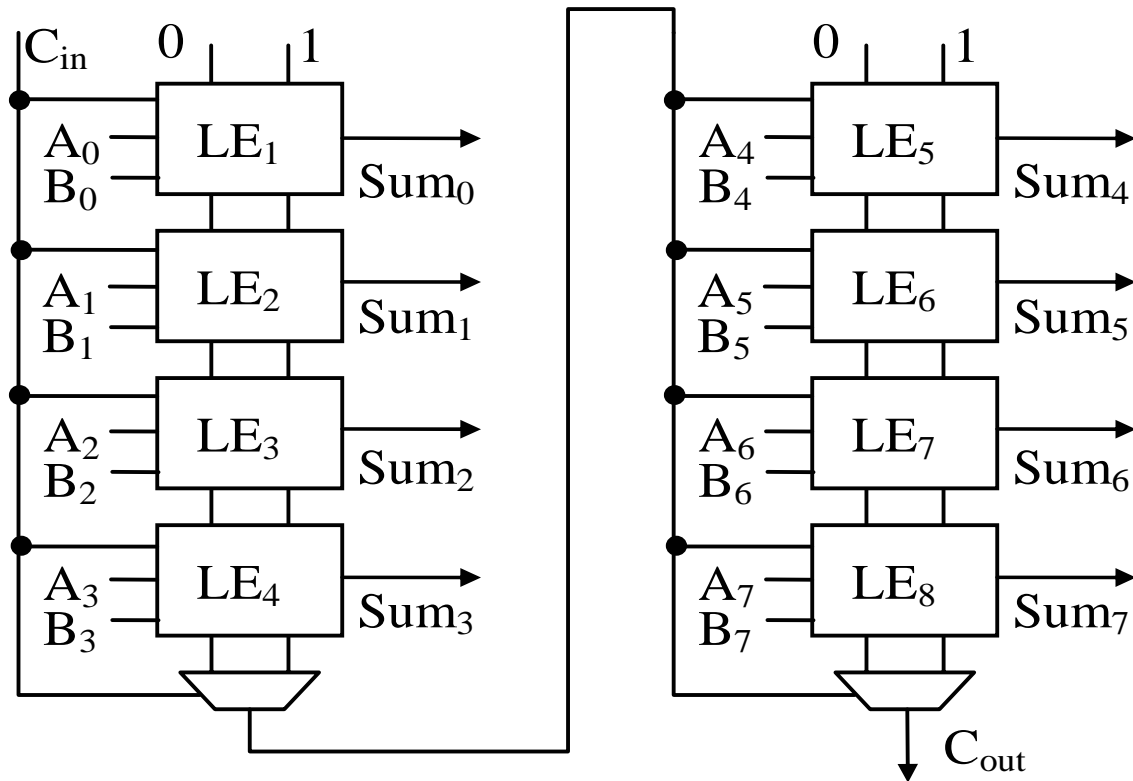
- $P_{resource} = S_{resource} \cdot A_{resource} \cdot P_{LE}$
 - ◆ $A_{resource}$ is characterized on the targeted FPGA architecture
- $P_{DSP} = 1.23 \cdot S_{DSP} \cdot BitWidth$
- $P_{IO} = 19.31 \cdot S_{IO}$
- $P_{CLK} = P_{clk-FF} + P_{clk-DSP}$
- $P_{memory} = Mem_{type}(BitWidth)$

Area Characterization

Operation	Resource	Usage
Add/Subtract	LE	N
Bitwise and/or/xor	LE	N
Compare ($=, >, \geq$)	LE	$\text{round}(0.67*N+0.62)$
Shift (with variable shift distance)	LE	$\text{round}(0.045*N^2+3.76*N-8.22)$
Multiply	DSP9x9	$N \leq 18: \lceil N/9 \rceil$ $N \leq 36: \lceil N/18 \rceil$
Multiplexer	LE	$N*\text{round}(0.67*K)$

N and K represent the bitwidth and the number of input operands, respectively.

An Example: Adder



An 8-bit carry-select adder in Altera Stratix

Delay Characterization

Operation	Delay (ns)
Add/Subtract	$0.024*N+1.83$
Bitwise and/or/xor	< 2
Compare (=, >, \geq)	$0.014*N+2.14$
Shift (with variable shift distance)	$4.3*10^{-5}*N^3-5*10^{-3}*N^2+0.24*N+0.93$
Multiply	$N \leq 9: 3.05$ $N \leq 18: 3.83$ $N \leq 36: 7.69$
Multiplexer (8-to-1)	$9.8*10^{-5}*N^3-7.4*10^{-3}*N^2+0.2*N+1.07$

Outline

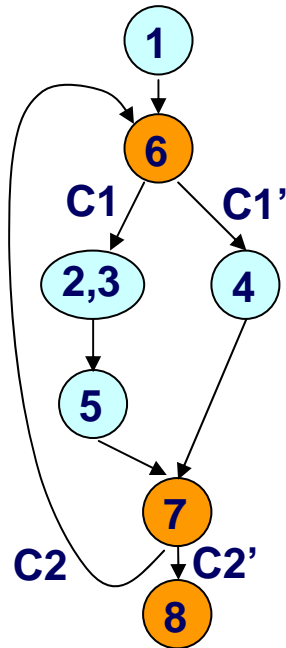
- **Background and motivation**
- **Related work**
- **Problem formulation**
- **Power estimation**
- **Simultaneous allocation and binding**
- **Experimental results**
- **Conclusions and future work**

Global Comparability Graph

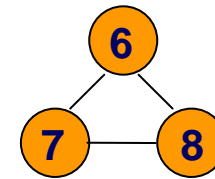
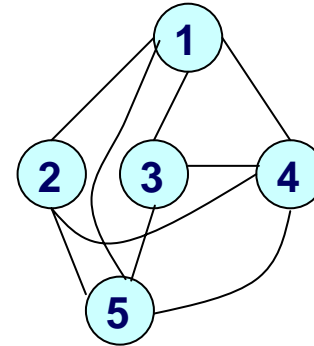
Multiplication



ALU operations



A State Transition Graph



Global Comparability Graph

Design Space Exploration

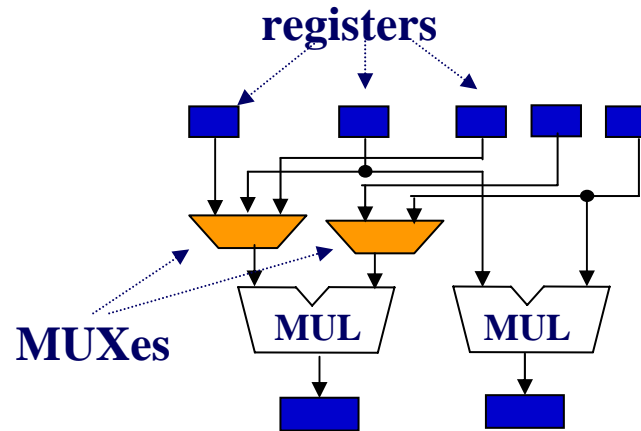
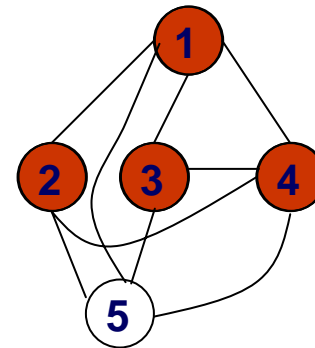
Node 2: (1) (2) two mul
(1, 2) one mul



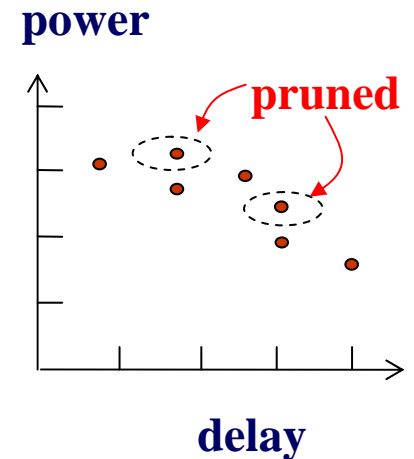
Node 3: (1) (2) (3) three mul
(1, 2) (3) two mul
(1, 3) (2) two mul



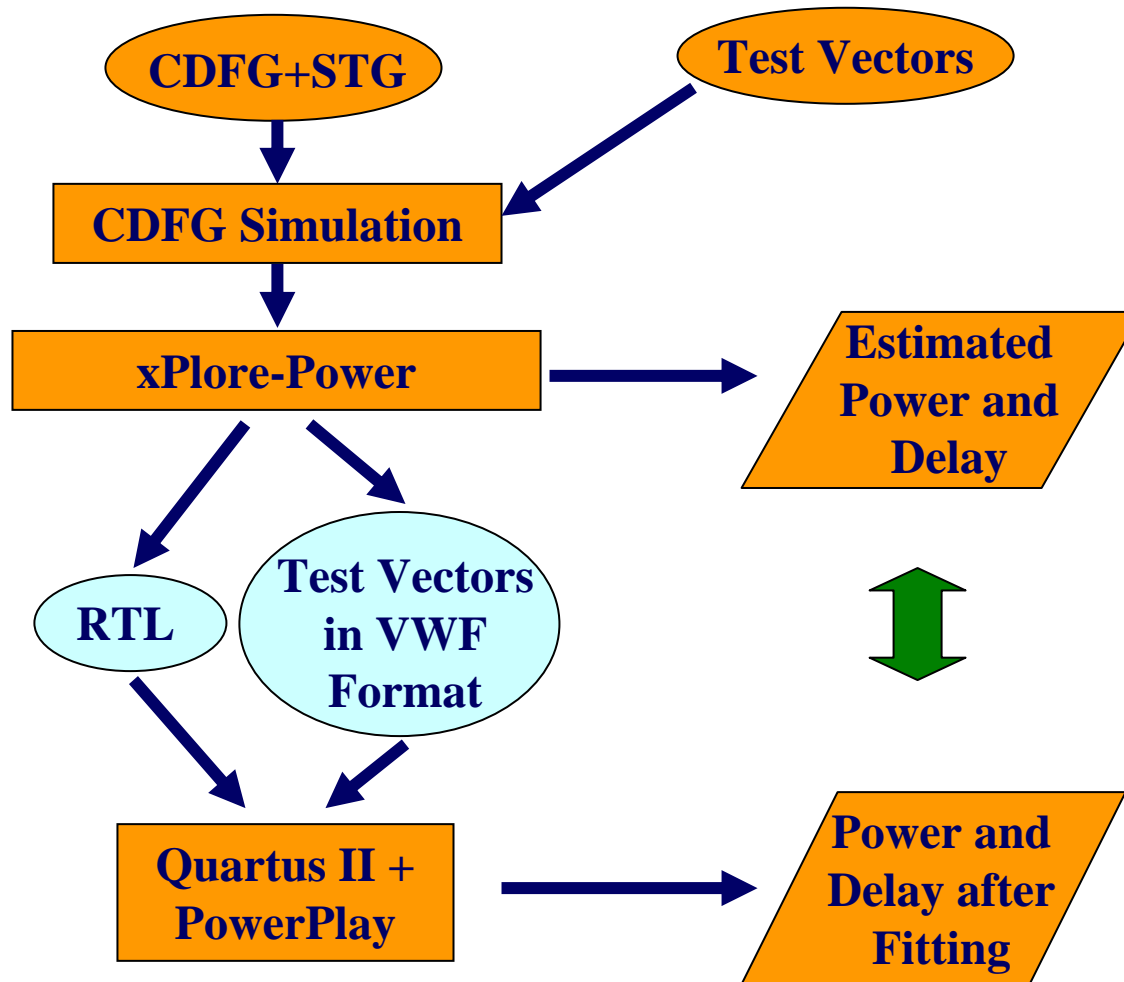
Node 4: (1) (2) (3) (4)
(1, 2, 4) (3)
(1, 2) (3, 4)
(1, 2) (3) (4)
(1, 3, 4) (2)
(1, 3) (2, 4)
(1, 3) (2) (4)



Datapath for
solution (1, 2, 4) (3)



xPlore-Power Experimental Flow

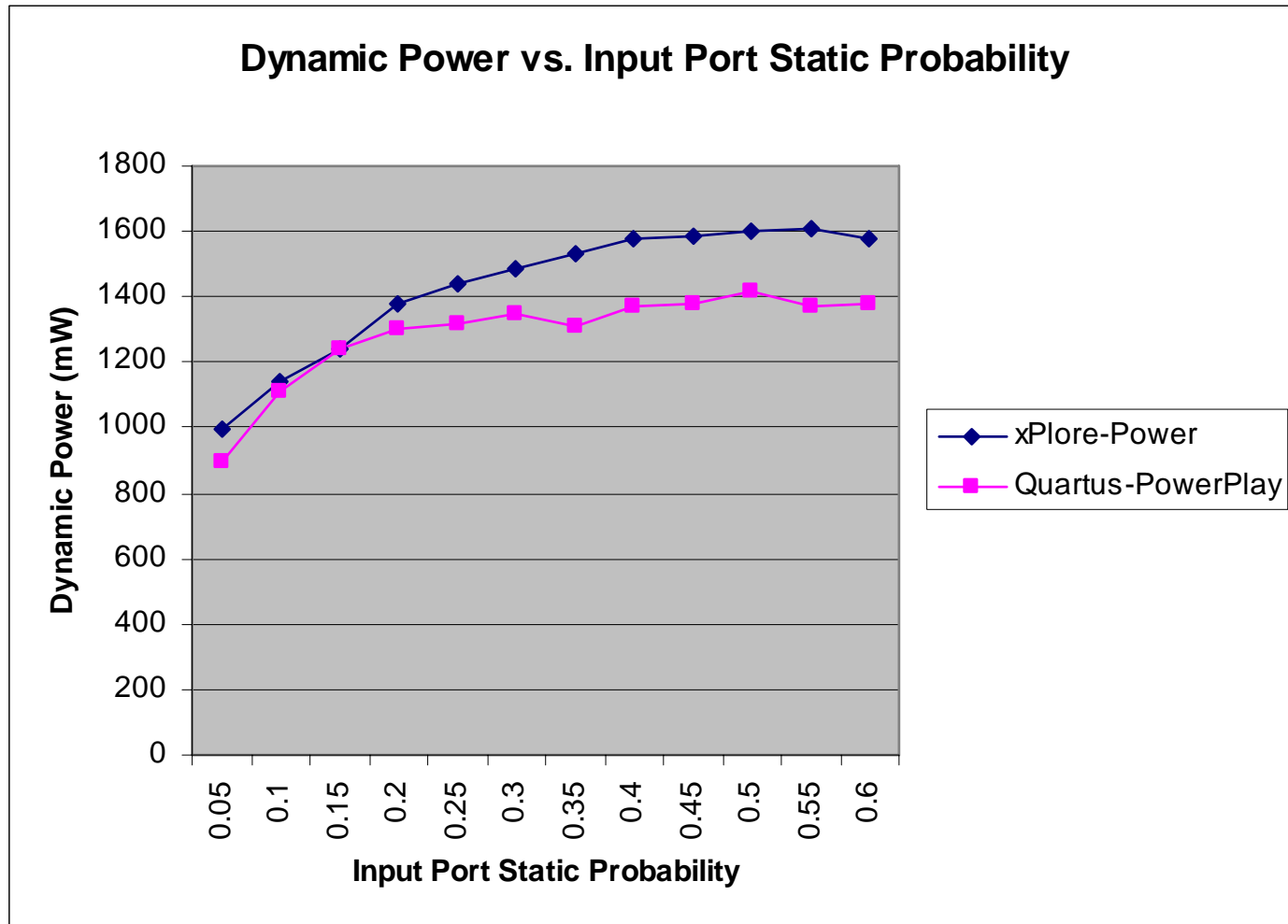


Power Estimation Results

Benchmarks	PowerPlay (mW)	xPlore-Power (mW)	Estimation Error (%)
dir	437.7	431	-1.5%
lee	1814.8	1533.4	-15.5%
mcm	390.7	423.4	8.4%
motion	239.3	252.1	5.3%
pr	1491.3	1536.7	3.0%
sym_conv	307.2	251.4	-18.2%
	Absolute Value Average:		8.7%

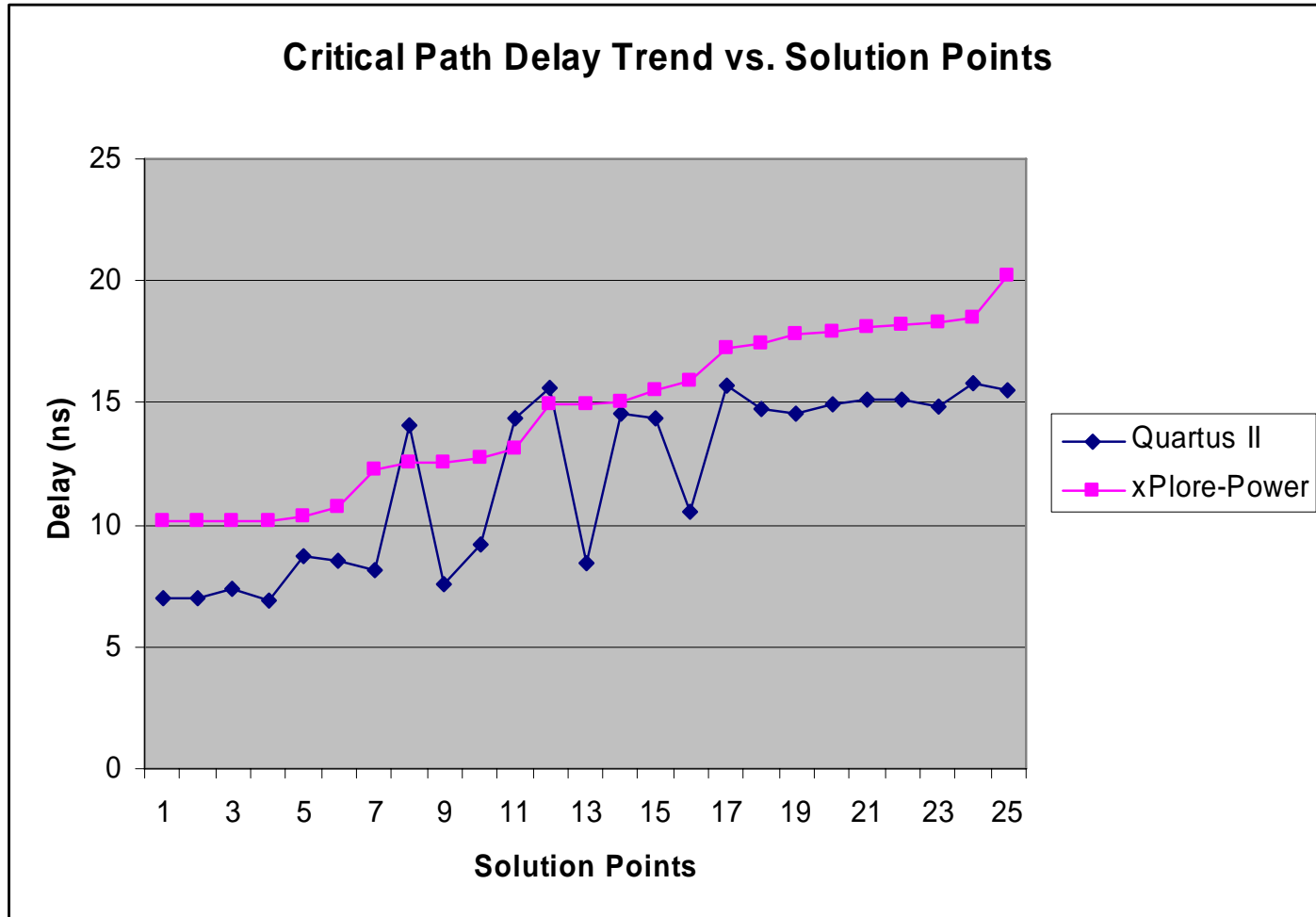
Total power includes 187.50 mW fixed static power for Altera Stratix device EP1S10B672C6

Power vs. Input Static Probability



On benchmark *Pr*

Delay and Power Trend for Solution Points



On benchmark *motion*

Traditional Register Binding

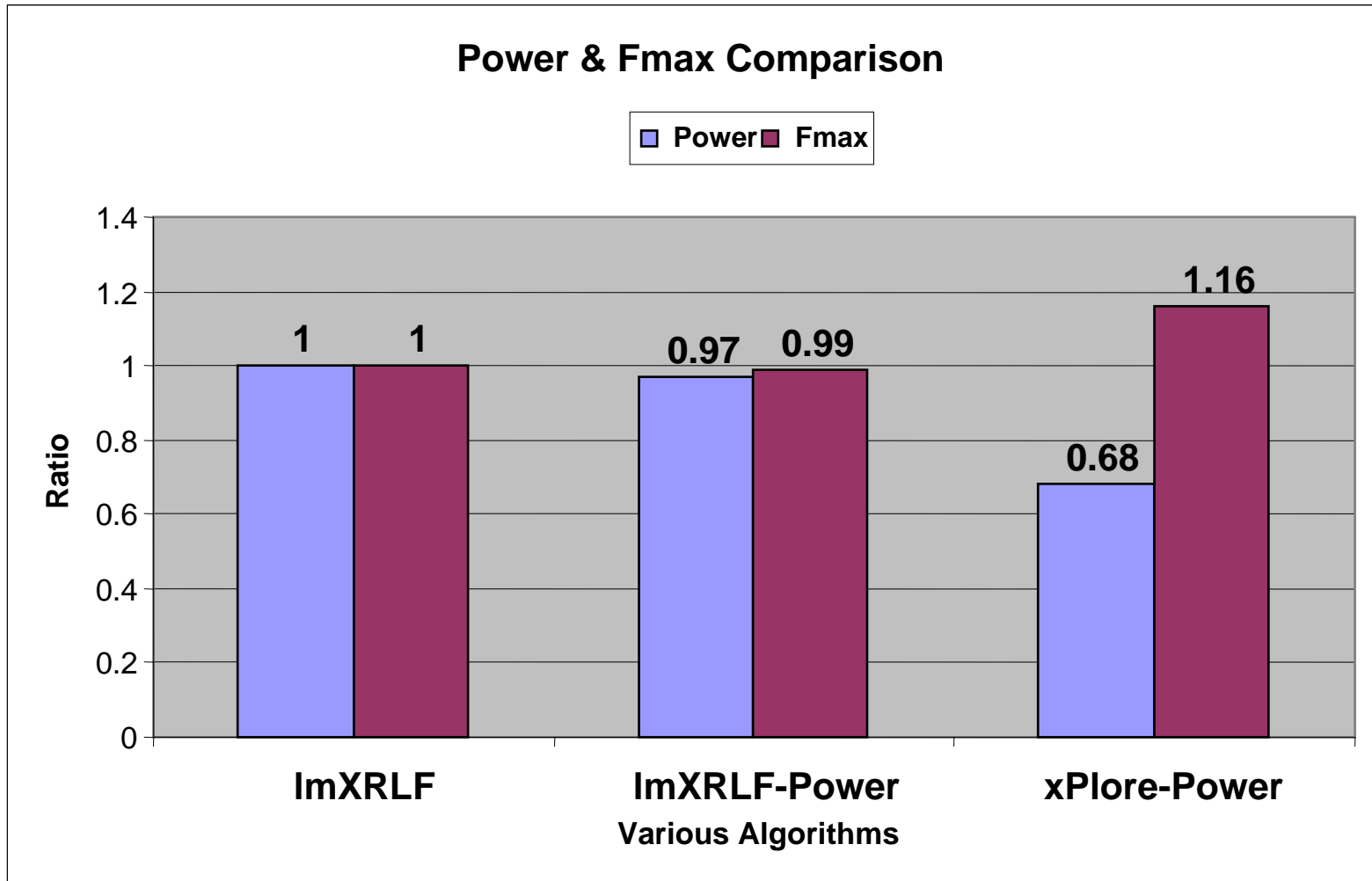
- Target minimum number of resources
- Power and delay of MUX are not explicitly considered
- Can lead to inferior solution especially for FPGA architectures

- Best clique partitioning solution on the comparability graph can achieve minimum resources
- Graph-coloring techniques can be transformed to finding the best clique partitioning solution
 - ◆ lmXRLF: a state-of-art graph coloring algorithm [Kirovski/Potkonjak, *DAC'98*]
 - ◆ lmXRLF-Power: modified lmXRLF to consider switching activities during the coloring process

Power and Performance Comparison (1)

	ImXRLF		ImXRLF-Power		xPlore-Power	
Benchmarks	Power (mW)	Fmax (MHz)	Power (mW)	Fmax (MHz)	Power (mW)	Fmax (MHz)
dir	541.9	160.1	447.7	153.7	250.2	236.3
lee	3955.6	113.6	4129.0	107.9	1627.3	122.9
mcm	492.9	171.9	500.9	174.6	203.2	241.1
motion	56.5	139.3	56.6	145.6	51.8	142.1
pr	1418.8	114.2	1360.5	111.0	1303.8	111.3
sym_conv	155	71.2	155	71.2	146.5	73.7

Power and Performance Comparison (2)



Conclusions

- We concentrated on resource allocation and binding tasks to optimize FPGA power and delay
- We designed a high-level power estimator for a commercial FPGA architecture
- We proposed a new simultaneous allocation and binding optimization algorithm, xPlore-Power, for efficient design space exploration
- Our high-level power estimator is only 8.7% away from a commercial gate-level FPGA power estimator
- Comparing to a traditional graph coloring-based register binding algorithm, xPlore-Power is 32% better on power and 16% better on Fmax after placement and routing