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

- 75% of logic utilization
- 50% of memory utilization
- 25% of DSP block utilization
- 30% of maximum I/O pins
- 12.5% toggle rate
- 200-MHz clock
- 25°C ambient temperature

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

Source: Altera, 2005
A power-conscious design methodology addresses power at every level of the design hierarchy.

Source: Pedram, 1999
Raise up the Design Level

- Higher design productivity
- Better quality of result
- Fast design space exploration
A Typical FPGA Architecture

Programmable IO

Programmable Logic (CLB)

Programmable Routing
Subtasks in Behavior-Level Synthesis

- **Scheduling** determines when an operation will be executed.

- **Allocation** determines the number of instances of each type of resources.

- **Binding** binds operations, variables, or data-transfers to the resources.
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

\[
P_{in} = \frac{\sum_{i=1}^{N-1} C_{in}(O_i, O_{i+1}) + C_{in}(O_N, O_1)}{2 \times \text{Bit} \_ \text{width} \times (N \times K \times B - 1)}
\]
Resource Power Estimation (1)

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

<table>
<thead>
<tr>
<th>Elements</th>
<th>Num</th>
<th>Est'ed P (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LE</td>
<td>1</td>
<td>0.12</td>
</tr>
<tr>
<td>LE w/ Carry</td>
<td>1</td>
<td>0.04</td>
</tr>
<tr>
<td>DSP</td>
<td>Per output</td>
<td>1.23</td>
</tr>
<tr>
<td>I/O</td>
<td>1</td>
<td>19.31</td>
</tr>
</tbody>
</table>

Global Clock Network Power for Altera's Stratix

\[ y = 1.8378x + 71.047 \]

M4K Power vs. Bitwidth

\[ y = 0.7319x + 6.9164 \]
Resource Power Estimation (2)

- \( P_{\text{resource}} = S_{\text{resource}} \cdot A_{\text{resource}} \cdot P_{LE} \)
  - \( A_{\text{resource}} \) is characterized on the targeted FPGA architecture

- \( P_{DSP} = 1.23 \cdot S_{DSP} \cdot \text{BitWidth} \)

- \( P_{IO} = 19.31 \cdot S_{IO} \)

- \( P_{CLK} = P_{clk-FF} + P_{clk-DSP} \)

- \( P_{\text{memory}} = Mem_{\text{type}}(\text{BitWidth}) \)
## Area Characterization

<table>
<thead>
<tr>
<th>Operation</th>
<th>Resource</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add/Subtract</td>
<td>LE</td>
<td>$N$</td>
</tr>
<tr>
<td>Bitwise and/or/xor</td>
<td>LE</td>
<td>$N$</td>
</tr>
<tr>
<td>Compare ($=, &gt;, \geq$)</td>
<td>LE</td>
<td>$\text{round}(0.67*N+0.62)$</td>
</tr>
<tr>
<td>Shift (with variable shift distance)</td>
<td>LE</td>
<td>$\text{round}(0.045<em>N^2+3.76</em>N-8.22)$</td>
</tr>
</tbody>
</table>
| Multiply                            | DSP9x9   | $N \leq 18: \left\lceil \frac{N}{9} \right\rceil$  
|                                     |          | $N \leq 36: \left\lceil \frac{N}{18} \right\rceil$ |
| Multiplexer                         | LE       | $N \times \text{round}(0.67*K)$      |

$N$ 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

<table>
<thead>
<tr>
<th>Operation</th>
<th>Delay (ns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add/Subtract</td>
<td>0.024*N+1.83</td>
</tr>
<tr>
<td>Bitwise and/or/xor</td>
<td>&lt; 2</td>
</tr>
<tr>
<td>Compare (=, &gt;, ≥)</td>
<td>0.014*N+2.14</td>
</tr>
<tr>
<td>Shift (with variable shift distance)</td>
<td>4.3*10^{-5}<em>N^3–5</em>10^{-3}<em>N^2+0.24</em>N+0.93</td>
</tr>
<tr>
<td>Multiply</td>
<td>N ≤ 9: 3.05</td>
</tr>
<tr>
<td></td>
<td>N ≤ 18: 3.83</td>
</tr>
<tr>
<td></td>
<td>N ≤ 36: 7.69</td>
</tr>
<tr>
<td>Multiplexer (8-to-1)</td>
<td>9.8*10^{-5}<em>N^3–7.4</em>10^{-3}<em>N^2+0.2</em>N+1.07</td>
</tr>
</tbody>
</table>
Outline

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

A State Transition Graph

Global Comparability Graph

Multiplication

ALU operations
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

CDFG+STG

CDFG Simulation

xPlore-Power

Test Vectors

Estimated Power and Delay

RTL

Test Vectors in VWF Format

Quartus II + PowerPlay

Power and Delay after Fitting
Power Estimation Results

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>PowerPlay (mW)</th>
<th>xPlore-Power (mW)</th>
<th>Estimation Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dir</td>
<td>437.7</td>
<td>431</td>
<td>-1.5%</td>
</tr>
<tr>
<td>lee</td>
<td>1814.8</td>
<td>1533.4</td>
<td>-15.5%</td>
</tr>
<tr>
<td>mcm</td>
<td>390.7</td>
<td>423.4</td>
<td>8.4%</td>
</tr>
<tr>
<td>motion</td>
<td>239.3</td>
<td>252.1</td>
<td>5.3%</td>
</tr>
<tr>
<td>pr</td>
<td>1491.3</td>
<td>1536.7</td>
<td>3.0%</td>
</tr>
<tr>
<td>sym_conv</td>
<td>307.2</td>
<td>251.4</td>
<td>-18.2%</td>
</tr>
</tbody>
</table>

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

Critical Path Delay Trend vs. 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)

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>lmXRLF</th>
<th>lmXRLF-Power</th>
<th>xPlore-Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>dir</td>
<td>541.9</td>
<td>447.7</td>
<td>250.2</td>
</tr>
<tr>
<td></td>
<td>160.1</td>
<td>153.7</td>
<td>236.3</td>
</tr>
<tr>
<td>lee</td>
<td>3955.6</td>
<td>4129.0</td>
<td>1627.3</td>
</tr>
<tr>
<td></td>
<td>113.6</td>
<td>107.9</td>
<td>122.9</td>
</tr>
<tr>
<td>mcm</td>
<td>492.9</td>
<td>500.9</td>
<td>203.2</td>
</tr>
<tr>
<td></td>
<td>171.9</td>
<td>174.6</td>
<td>241.1</td>
</tr>
<tr>
<td>motion</td>
<td>56.5</td>
<td>56.6</td>
<td>51.8</td>
</tr>
<tr>
<td></td>
<td>139.3</td>
<td>145.6</td>
<td>142.1</td>
</tr>
<tr>
<td>pr</td>
<td>1418.8</td>
<td>1360.5</td>
<td>1303.8</td>
</tr>
<tr>
<td></td>
<td>114.2</td>
<td>111.0</td>
<td>111.3</td>
</tr>
<tr>
<td>sym_conv</td>
<td>155</td>
<td>155</td>
<td>146.5</td>
</tr>
<tr>
<td></td>
<td>71.2</td>
<td>71.2</td>
<td>73.7</td>
</tr>
</tbody>
</table>
Power and Performance Comparison (2)

Power & Fmax Comparison

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Power</th>
<th>Fmax</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImXRLF</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ImXRLF-Power</td>
<td>0.97</td>
<td>0.99</td>
</tr>
<tr>
<td>Various Algorithms</td>
<td></td>
<td></td>
</tr>
<tr>
<td>xPlore-Power</td>
<td>0.68</td>
<td>1.16</td>
</tr>
</tbody>
</table>

![Bar Chart](Image)
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.