Physical Hierarchy Generation

Jason Cong
UCLA Computer Science Department
Email: cong@cs.ucla.edu
Tel: 310-206-2775
http://cadlab.cs.ucla.edu/~cong

Outline

- Global interconnects in nanometer technologies
- Interconnect-centric design flow
- Physical hierarchy generation
  - Motivation
  - Approaches
- Results and on-going work
Global/Local Interconnect Delays vs. Gate Delays

Optimization is obtained buffer insertion/sizing and wire sizing

Clock cycles required for traveling 2cm line under BIWS (buffer insertion and wire sizing)

Estimated by IPEM
On NTRS'97 technology

Driver size: 100x min gate
Receiver size: 100x min gate
Buffer size: 100x min gate
How Far Can We Go in Each Clock Cycle

- NTRS’97 0.07um Tech
- 5 G Hz across-chip clock
- 620 mm² (24.9mm x 24.9mm)
- IPEM BIWS estimations
  - Buffer size: 100x
  - Driver/receiver size: 100x
- From corner to corner:
  - 7 clock cycles

Two Important Implications

- Interconnects determine the system performance
  - Interconnect/communication-centric design methodology
- Need multiple clock cycles to cross the global interconnects in giga-hertz designs
  - Pipelining/retiming on global interconnects
Interconnect-Centric Design Methodology

- **Proposed transition**
  
  device/function centric \[\rightarrow\] interconnect/communication centric

- **Analogy**
  
  Data/Objects \[\rightarrow\] Programs

Interconnect-Centric IC Design Flow
Under Development at UCLA

- Architecture/Conceptual-level Design
- Design Specification
- Interconnect Planning
  - Physical Hierarchy Generation
  - Fourplane/Course Placement with Interconnect Planning
  - Interconnect Architecture Planning
- Synthesis and Placement under Physical-Hierarchy
  - Performance-driven Global Routing
  - Pseudo Pin Assignment under Noise Control
- Interconnect Layout
  - Route Planning
  - Point-to-Point Routing
- Interconnect Synthesis
  - Performance-driven Global Routing
- Interconnect Optimization (TRIO)
  - Topology Optimization with Buffer Insertion
  - Wire sizing and spacing
  - Simultaneous Buffer Insertion and Wire Sizing
  - Simultaneous Topology Construction with Buffer Insertion and Wire Sizing
- Interconnect Performance Estimation Models (IPEM)
  - OWS, SDWS, BSWS
- Final Layout
Interconnect-Centric IC Design Flow
Under Development at UCLA

Architecture/Conceptual-level Design

Design Specification

Final Layout

Interconnect Planning
- Physical Hierarchy Generation
- Floorplan/Coarse Placement with Interconnect Planning

Interconnect Optimization (TRIO)
- Topology Optimization with Buffer Insertion
- Wire sizing and spacing
- Simultaneous Buffer Insertion and Wire Sizing
- Simultaneous Topology Construction with Buffer Insertion and Wire Sizing

Interconnect Synthesis
Performance-driven Global Routing
- Pseudo Pin Assignment under Noise Control

Interconnect Layout
Route Planning
- Point-to-Point Gridless Routing

Interconnect Performance Estimation Models (IPEM)
- OWS, SDWS, BISWS

Synthesis and Placement under Physical Hierarchy
- Performance-driver
- Pseudo Pin Assignment

Interconnect Route Planning

Structure view
Functional view
Physical view
Timing view
Interconnect Planning

- Physical Hierarchy Generation
- Floorplan/Coarse Placement with Interconnect Planning
- Interconnect Architecture Planning

Physical Hierarchy Generation

Motivation:
- Designs are hierarchical due to high complexity
- Design specification (in HDL) defines logic hierarchy
- The common practice partitions the design following the logic hierarchy
- But logic hierarchy may not be suitable to be embedded on a 2D silicon surface, resulting poor interconnect
Example of Logic Hierarchy

Verilog

module cpu(pj_su, pj_boot8, ...);
input ...;
output ...;
	IU fpui(fpair(iu rs2_e), fpbin(iu rs1_e), fpinp(fpop),
		fpout(fpu_data_e), fpu(fpop),
		fpv(fpu clk), ...);
	SMU smu(iu optop_in(iu optop_din), ...);
	DCRAM dcram_tag_shell(dcram shell(data_in), ...);
	DCU dcu(iu_data_in, ...);

dendmodule

Example of Logic Hierarchy in Final Layout

By courtesy of IBM (Tony Drumm)
Example of Logic Hierarchy in Final Layout

What Have We Learned?

- Logic hierarchy may not map well to physical hierarchy
- Floorplanning of logic blocks in RT-level may be a bad idea

Alternatives?

- Synthesis under physical hierarchy!
Physical Hierarchy Generation
Problem Formulation

Logical Hierarchy

Assign modules to physical hierarchy with interconnect estimation and optimization

Impact of Physical Hierarchy Generation

Define the Global Interconnects

Examples: Global interconnects defined by two different physical hierarchy
Synthesis under Physical Hierarchy

Difficulties in Physical Hierarchy Generation

- How to consider retiming/pipelining over global interconnect
  
  Use of the concepts of sequential arrival/required times

- How to handle the high complexity of “almost flattened” designs
  
  Use multi-level optimization techniques
Need of Considering Retiming during Placement
- Retiming/pipelining on global interconnects

✦ Multiple clock cycles are needed to cross the chip
✦ Proper placement allows retiming to hide global interconnect delays.

Placement 1

Before retiming, \( t = 5.0 \)
After retiming, \( t = 3.0 \)

Placement 2

Before retiming, \( t = 4.0 \)
Better Initial Placement!!
Sequential Arrival Time (SAT)

Definition [Pan et al, TCAD98]
- $l(v) = \max\{\text{delay from PIs to } v\text{ after opt. retiming under a given clock period } f\}$
- $l(v) = \max\{l(u) - f \cdot w(u,v) + d(u,v) + d(v)\}$

Relation to retiming: $r(v) = \frac{l(v)}{f} - 1$

Theorem: $P$ can be retimed to $f + \max\{d(e)\}$ iff $l(\text{POs}) \leq f$

With loops, problem is difficult
- Topological order does not exist!
- Start with a min $l$-value for each node and iteratively improve it
- Convergence is guaranteed in $O(n)$ iterations if the circuit can be retimed to the target cycle time

Outline of our approach
- $\text{SAT(PI)} = 0$, $\text{SAT(others)} = -?$
- Relax one vertex at a time and update $l$-values
- Complexity is $O(VE)$
Sequential Arrival Time (SAT)

d(\gamma) = 1, \ d(e) = 2

Is \ ? = 4.5 possible?

<table>
<thead>
<tr>
<th>Iter#</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-?</td>
<td>-?</td>
<td>-?</td>
<td>-?</td>
<td>-?</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1.5</td>
<td>-?</td>
<td>-?</td>
<td>-?</td>
<td>-?</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>-1.5</td>
<td>1.5</td>
<td>1.5</td>
<td>-?</td>
<td>-?</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>-1.5</td>
<td>1.5</td>
<td>4.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>-1.5</td>
<td>1.5</td>
<td>4.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>-1.5</td>
<td>1.5</td>
<td>4.5</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Cycle time 4.5 is possible as \ l(g) = 4.5
Multi-Level Framework

- Coarsening
- Problem sizes
  - Multi-level coarsening generates smaller problem sizes for top levels
  - Faster optimization on top levels
  - Different levels explore different aspects of the solution space
  - Refinement on good solutions from coarser levels can be fast and simple with good solution quality

Successes of Multi-Level Approach

- First used to solve partial differential equations (multi-grid method)
- Successfully applied to circuit partitioning (hMetis [Karypis et al, 1997])
  - Best partitioner for cut-size minimization
- Successfully applied to physical hierarchy generation (HPM and GEO [Cong et al, DAC’00 & ICCAD’00])
  - 30-40% delay reduction compared to hMetis
- Successfully applied to circuit placement [Chan et al, ICCAD’00]
  - 10x speed-up over GordianL
Physical Hierarchy Generation:
Multi-Level Coarse Placement & Retiming

- Bottom-up multi-level clustering
- Coarse placement at each level using multi-way weighted min-cut or SA
- Sequential timing analysis at each level

Hierarchical Approach vs. Multi-Level Approach

- Hierarchical approach: higher-level design constrains lower-level designs
  - Not sufficient information at higher-level
  - Mistake at higher level is impossible or costly to correct
- Multi-level approach: finer-level design refines coarse-level design
  - Converge to better solution as more details are considered
Coarsening for Physical Hierarchy Generation  
(Multi-level Clustering)

- Follow logic hierarchy:
- Connectivity based clustering:
  - hMetis [Karypis et al, DAC’97]
  - Hyper-edge coarsening
  - ESC [Cong and Lim, ICCAD’00]
  - Global edge separability based clustering
- Performance driven multi-level clustering:
  - TLC [Cong and Romesis, DAC’01]

ESC Clustering

- Edge separability [Cong & Lim, ASPDAC00]
  - Min # of edges to separate x and y: x-y mincut

- ESC clustering algorithm
  - Can compute a tight lower-bound $q(e)$ of $?e$ for all edges in $O(n\log n)$ time [Nagamochi & Ibaraki, Algorithmica92]
  - Use $q(e)$ for bottom-up multi-level clustering
  - Produce very good cutsize, comparable to hMetis [KA+97]
**ESC Experimental Results**

- LR [CL+97] bipartitioning on ISPD98 [Alp98] circuits

<table>
<thead>
<tr>
<th>Scale Cutsize</th>
<th>ABS</th>
<th>DEN</th>
<th>REP</th>
<th>RTC</th>
<th>CLO</th>
<th>CON</th>
<th>ESC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>1.09</td>
<td>1.13</td>
<td>1.15</td>
<td>1.31</td>
<td>0</td>
<td>1</td>
<td>1.31</td>
</tr>
</tbody>
</table>

- ABS: Absorption [SS93] (max): weight of edges absorbed into C
- DEN: Density [CS93, HK95] (max): density of C in terms of w(e)
- REP: Rent Parameter [NOP87] (min): entail better placement result
- RTC: Ratio Cut [WC92] (min): identify natural clusters
- CLO: Closeness [SK93] (max): connectivity to neighboring vertices
- CON: Connectivity [SU72] (max): connectivity to neighboring vertices

**Performance Driven Clustering**

**Problem Formulation**

**Inputs:**
- Areas and delays for all modules
- Different inter-cluster delays for different level
- Area constraints on each level of clustering

**Objectives:** Build multi-level clusters that minimized the delay under the area constraints

- Capacity of first-level cluster: 2
- Capacity of second-level cluster: 4
- \(d=1, D_1=2, D_2=4, D_3=8\)
- First solution delay: 35
- Second solution delay: 31
Performance Driven Clustering – TLC Clustering

- Linear space and time complexity (if the network is bounded).
- Two phases (labeling and clustering).
  - First phase: labeling
    - From PIs to POs, visit nodes in topological order
    - Label the node with the maximum delay under the two-level delay model.
  - Second phase: clustering.
    - From POs to PIs, cluster nodes

Node duplication (ND) control
- Full node duplication
- Partial node duplication (depends on node criticality)
- No node duplication

TLC Experimental Results
GEO Experimental Results

- Comparison with existing algorithms
  - hMetis [DAC97] + retiming + slicing floorplan [Algo89]
  - HPM [DAC00] + slicing floorplan [Algo89]
  - GEO: simultaneous partitioning + coarse placement + retiming

Close to 40% delay reduction!

Preliminary Results on Multi-level Coarse Placement

- Multi-level simulated annealing coarse placement engine
- Comparison with GORDIAN:
  - Our engine only turns on wire length optimization
  - Legalized by DOMINO for wire length comparison

- 1k-10k test cases: s9224, s5378, s13207, s15850, bigkey
- 10k-50k test cases: s38417, s38584, clma, big1, big3, big4
- 50k-200k test cases: big2, big5, big6
Ongoing work – Architecture Evaluation

- Architecture blocks with different implementations with
  - Different areas
  - Different delays
  - Different pipeline stages
- ...
- Parameterized Buses with different bus widths
- Interconnect planning extracts area, delay, etc. for architecture evaluation
- Interconnect planning uses architecture evaluation functions to explore alternative architecture blocks and buses for system performance optimization

Ongoing work – Synthesis under Physical Hierarchy

- Consider interconnect information during behavior and logic level synthesis
- Explore various synthesis solutions to tradeoff long global wires with short local wires
- Generalized technology mapping: choosing different behavior and logic synthesis solutions for each block
- Revisit and extend various re-wiring techniques
Concluding Remarks

- Interconnects determine system performance
- Interconnect-centric design is needed
  - Interconnect planning
  - Interconnect synthesis
  - Interconnect layout
- Physical hierarchy generation is crucial for interconnect planning
- A good combination of partitioning/placement and retiming can hide global interconnect delays, and lead to good physical hierarchy
- Multi-level method is an effective way to cope with complexity

Acknowledgements

- Thanks for current and former students contributed to this project: Chin-Chih Chang, Ashok Jagannathan, Sung Lim, Michail Romesis, Chang Wu, and Xin Yuan
- Thanks supports from GSRC, SRC, Fujitsu, IBM, and Intel

More details: [http://cadlab.cs.ucla.edu](http://cadlab.cs.ucla.edu)