An Interconnect-Centric Design Flow for Nanometer Technologies

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
Interconnect Delays in Nanometer Technologies

<table>
<thead>
<tr>
<th>Technology (um)</th>
<th>0.25</th>
<th>0.18</th>
<th>0.15</th>
<th>0.13</th>
<th>0.10</th>
<th>0.07</th>
</tr>
</thead>
<tbody>
<tr>
<td>1mm (ps)</td>
<td>59</td>
<td>49</td>
<td>51</td>
<td>44</td>
<td>52</td>
<td>42</td>
</tr>
<tr>
<td>2cm no-opt (ps)</td>
<td>2600</td>
<td>2500</td>
<td>2700</td>
<td>2600</td>
<td>3700</td>
<td>4700</td>
</tr>
<tr>
<td>2cm best-opt (ps)</td>
<td>900</td>
<td>800</td>
<td>770</td>
<td>700</td>
<td>770</td>
<td>670</td>
</tr>
</tbody>
</table>

- Best-opt uses simultaneous buffer insertion, driver/buffer sizing, and wiresizing
- Based on NTRS'97 data, with consideration of use copper & low-K materials

Fact: Global interconnect is 15x – 20x slower than logic gates

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 → interconnect/communication centric

- Analogy

  Programs → Data/Objects
  Interconnect → Device

Jason Cong 10/17/01
Interconnect-Centric IC Design Flow
Under Development at UCLA

Architecture/Conceptual-level Design

Design Specification

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

Synthesis and Placement under Physical Hierarchy

Interconnect Synthesis
- Topology generation & wire sizing for delay
- Wire ordering & spacing for noise control

Interconnect Layout

Interconnect Performance Estimation Models (IPEM)

Interconnect Optimization
- 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 Layout

Route Planning

Point-to-Point Gridless Routing

Final Layout

Interconnect Planning

- Current approach:
  - RT-level floorplanning based on logic hierarchy
  - Delay budgeting + block by block synthesis + physical design
Example of Logic Hierarchy

```
module cpu;
  input ...;
  output ...;
  iu fpair(iu rs2_e), fpbin(iu rs1_e), fpout(fpout),
       fpzyn(fp dyn_e), fpall(iu Ibid, fp),
       fpout(fpu data e), clk(clk), ...);
  pcsu( ...);
  smu( ...);
  dtag_shell(data_in), ...);
  dcram_shell(data_in[31], ...);
  icram_shell(icu data), ...);
  icu(icu Ibid).
endmodule
```

Interconnect Planning

- **Current approach:**
  - RT-level floorplanning based on logic hierarchy
  - Delay budgeting + block by block synthesis + physical design
- **Problem:** may loss much optimality
  - Logic hierarchy may not embed well on a 2D silicon surface, resulting poor global interconnect
Example of Logic Hierarchy in Final Layout

By courtesy of IBM (Tony Drumm)
Interconnect Planning

- Current approach:
  - RT-level floorplanning based on logic hierarchy
  - Delay budgeting + block by block synthesis + physical design
- Problem: may loss much optimality
  - Logic hierarchy may not embed well on a 2D silicon surface, resulting in poor global interconnect

- Our conclusion:
  - RT-level floorplanning of logic blocks may be a bad idea

- Our proposal:
  - synthesis under physical hierarchy

Physical Hierarchy Generation

Problem Formulation

- Logical Hierarchy
- Assign modules to physical hierarchy with interconnect estimation and optimization

- Hard IP  Soft module
- Same color for modules of the same logic hierarchy

Assign modules to physical hierarchy with interconnect estimation and optimization
Impact of Physical Hierarchy Generation
Define the Global Interconnects
Example: Global interconnects defined by two different physical hierarchies

Synthesis under Physical Hierarchy

Latch Critical path

Alternative Architecture Block Selection Re-Synthesis and Retiming

Jason Cong 10/17/01 15

Jason Cong 10/17/01 16
Difficulties in Physical Hierarchy Generation

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

- How to handle the high complexity of “almost flattened” designs
  
  Use the multi-level optimization technique

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, \( d(v) = 5.0 \)

After retiming, \( d(v) = 3.0 \)

Placement 2

Before retiming, \( d(v) = 4.0 \)

Better Initial Placement!!
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, \( t = 4.0 \)
After retiming, \( t = 4.0 \)

Better Initial Placement!!

Sequential Arrival Time (SAT)

- Definition [Pan et al, TCAD98]
  \( l(v) = \max_{u,v} \{ l(u) - f \cdot w(u,v) + d(u,v) + d(v) \} \)

- Relation to retiming: \( r(v) = \frac{\max_{u,v} \{ l(u) - f \cdot w(u,v) + d(u,v) + d(v) \}}{f} - 1 \)

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

\( l(u) = 7, l(w) = 3, d(v) = 1, d(e) = 2, f = 5 \)
\( l(v) = \max\{ 7 - 5 \cdot 1 + 2 + 1, 3 + 2 + 1 \} = 6 \)
Sequential Arrival Time (SAT) Computation

- **Difficulty**
  - Need to work on the entire circuit, with many cycles
  - Topological order does not exist!

- **Basic approach:**
  - Start with min \( l \)-value for each node and iteratively improve it

- **Will the computation converge?**
  - YES, if the the circuit can be retimed to the target cycle time
  - Theorem: Convergence is guaranteed in \( O(n) \) iterations if the circuit can be retimed to the target cycle time

- **Practical experience**
  - Converge in constant iterations with a good DFS order

---

**Example: SAT Computation**

\[ d(v) = 1, \quad d(e) = 2 \]

Is \( t = 4.5 \) 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 \( l(g) = 4.5 \)?
Example: SAT Computation

d(v)=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 \( l(g) = 4.5 \).

Simultaneous (Coarse) Placement with Retiming on Interconnects

Our solution
- Compute SATs of all nodes for a given placement solution
- Minimize SATs of POs by improving the placement solution

Alternative solution [Brayton, et al]
- Enforcing all loop constraints during placement
Difficulties in Physical Hierarchy Generation

- How to consider retiming/pipelining over global interconnects

Use of the concepts of sequential arrival/required times

- How to handle the high complexity of “almost flattened” designs

Use the multi-level optimization technique

Multi-Level Framework

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]
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**

For Altera APEX FPGAs with 2-level hierarchy (LABs & MegaLABs)

- Quartus + TLC (full ND)
- Quartus + TLC (partial ND)
- Quartus + TLC (no ND)
- Quartus

**Normalized Delay**

Some Experimental Result

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

Close to 40% delay reduction!
Experiment on an IBM Design

Physical Hierarchy Generation

Detailed Placement by IBM tools

270k cells, 300k nets
Technology: ibm_sa27e (0.11um copper)

Interconnect-Centric IC Design Flow Under Development at UCLA

Architecture/Conceptual-level Design
Design Specification

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

Synthesis and Placement under Physical Hierarchy
- Topology generation & routing for delay
- Wire ordering & spacing for noise control

Interconnect Layout
- Timing

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

Jason Cong 10/17/01
35 36
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
  - Select different behavior and logic synthesis solutions for each block for global optimization
  - Scheduling for hiding interconnect latency
- …

Ongoing work – Micro-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
Concluding Remarks

* Interconnects determine system performance
* An interconnect-centric design flow is needed
  * Interconnect planning
  * Synthesis/layout under physical hierarchy
  * 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, David Pan, 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)