ESE534: Computer Organization

Day 16: March 19, 2012
Interconnect 3: Richness

Last Time

• Rent’s Rule
  – And its implications
• Superlinear growth rate of interconnect
  \( p > 0.5 \)
  \( \implies \text{Area growth } \Omega(N^{2p}) \)

Today

• How rich should interconnect be?
  – specifics of understanding interconnect
  – methodology for attacking these kinds of questions

Now What?

• There is structure (locality)
• Rent characterizes locality
• How rich should interconnect be?
  – Allow full utilization of compute units?
  – What is most area efficient?
  \( \implies \text{Need to model requirements and area impact} \)

Preclass 1 and 2

• Wire count?
Preclass 1 and 2

- Wire count?

Step 1: Build Architecture Model

- Assume geometric growth
- Pick parameters: Build architecture can tune
  - C
  - p

Tree of Meshes

- Natural model is hierarchical
- Restricted internal bandwidth
- Can match to model

Parameterize C

Parameterize Growth by p

- What are IO schedules? (preclass 3)

Preclass 5

- What are IO schedules according to Rent for particular p's?
  - p=1/2
  - p=2/3
  - p=3/4

\[ IO = c N^p \]
Parameterize $p$

- What is $p$ for each network? (preclass 6)

---

Parameterize Growth

Step 2: Area Model

- Need to know effect of architecture parameters on area (costs)
  - focus on dominant components
    - wires
    - switches
    - logic blocks(?)

---

Area Parameters

- $A_{\text{logic}} = 10K \text{ } F^2$
- $A_{\text{sw}} = 625 \text{ } F^2$
- Wire Pitch = 4 $F$

---

Switchbox Population

- Full population is excessive (next lecture)
- Hypothesis: linear population adequate
  - still to be (dis)proven

---

“Cartoon” VLSI Area Model

(Example artificially small for clarity)
Step 3: Characterize Application Requirements

- Identify representative applications.
  - Today: IWLS93 logic benchmarks
- How much structure there?
- How much variation among applications?

Application Requirements

Max: C=7, P=0.68  Avg: C=5, P=0.72
Complication

• Interconnect requirements vary among applications
• Interconnect richness has large effect on area
• What is effect of architecture/application mismatch?
  – Interconnect too rich?
  – Interconnect too poor?

Interconnect Mismatch in Theory

Step 4: Assess Resource Impact

• Map designs to parameterized architecture
• Identify architectural resource required

Compare: mapping to k-LUTs; LUT count vs. k.

Mapping to Fixed Wire Schedule

• Easy if need fewer wires than Net
• If need more wires than net, must depopulate to meet interconnect limitations.

Preclass 4

• Smallest Network that Top graph fits on?
Mapping to Fixed-WS

- Better results if "reassociate" rather than keeping original subtrees.

Preclass 4

- Smallest Network that Middle graph fits on?

Middle vs. Top

- Smallest Network that Bottom graph fits on?

Observation

- Don't really want a "bisection" of LUTs
  - subtree filled to capacity by *either* of
    - LUTs
    - root bandwidth
  - May be profitable to cut at some place other than midpoint
    - not require "balance" condition
  - "Bisection" should account for both LUT and wiring limitations
Challenge

• Not know where to cut design
  – not knowing when wires will limit subtree capacity

Brute Force Solution

• Explore all cuts
  – start with all LUTs in group
  – consider “all” balances
  – try cut
  – Recurse

• Viable?

Brute Force

• Too expensive
• Exponential work

• …viable if solving same subproblems

Simplification

• Single linear ordering
• Partitions = pick split point on ordering
• Reduce to finding cost of [start,end] ranges (subtrees) within linear ordering
• Only $n^2$ such subproblems
• Can solve with dynamic programming

Dynamic Programming

• Just one possible “heuristic” solution to this problem
  – not optimal
  – dependent on ordering
  – sacrifices ability to reorder on splits to avoid exponential problem size

• Opportunity to find a better solution here...

Ordering LUTs

• Another problem
  – lay out gates in 1D line
  – minimize sum of squared wire length
    • tend to cluster connected gates together
  – Is solvable mathematically for optimal
    • Eigenvector of connectivity matrix

• Use this 1D ordering for our linear ordering
Step 5: Apply Area Model

- Assess impact of resource results

Resources × Area Model ⇒ Area

Net Area

Picking Network Design Point

<table>
<thead>
<tr>
<th>Objective</th>
<th>params</th>
<th>Sigma</th>
<th>LUT</th>
<th>relative area</th>
<th>rel area</th>
<th>Util.</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>P</td>
<td></td>
<td></td>
<td>6</td>
<td>0.6</td>
<td>1.23</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.87</td>
<td></td>
<td></td>
</tr>
<tr>
<td>area with</td>
<td></td>
<td></td>
<td></td>
<td>full util</td>
<td>10</td>
<td>0.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.98</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.00</td>
</tr>
</tbody>
</table>

Don’t optimize for 100% compute util. (100% yield)
...also don’t optimize for highest peak.

What about a single design?
LUT Utilization predict Area?

Methodology
1. Architecture model (parameterized)
2. Cost model
3. Important task characteristics
4. Mapping Algorithm
   – Map to determine resources
5. Apply cost model
6. Digest results
   – find optimum (multiple?)
   – understand conflicts (avoidable?)

Admin
• Reading for Wednesday online
• HW8 out

Big Ideas
[MSB Ideas]
• Interconnect area dominates logic area
• Interconnect requirements vary
  – among designs
  – within a single design
• To minimize area
  – focus on using dominant resource (interconnect)
  – may underuse non-dominant resources (LUTs)

Big Ideas
[MSB Ideas]
• Two different resources here
  – compute, interconnect
• Balance of resources required varies among designs (even within designs)
• Cannot expect full utilization of every resource
• Most area-efficient designs may waste some compute resources (cheaper resource)