ECE260B – CSE241A
Winter 2004
Parasitic Estimation

Website:  http://vlsicad.ucsd.edu/courses/ece260b-w04
Parasitic Extraction Accuracy

- Above 0.5µm feature size, wire cross-section was rectangular
- Interconnect modeled as parallel plate over ground plane
  - Parallel plate capacitance
  - Fringe capacitance
- 2-D extraction accurate enough: Area + Fringe
Layout Parasitic Extraction

- Necessary step after routing
- Account for non-ideal nature of interconnect
  - Wire capacitance
  - Wire and via resistance
- Parasitic information is used in post-layout verification
  - Timing verification of synchronous circuits
  - Functional verification of asynchronous circuits
- Design performance is ultimately limited by parasitics
Layout Parasitic Extraction

- **GOAL:** Generate “real” RC model of interconnect

- Extraction can generate various types of data:
  - RC (dspf/rspf/set_load)
  - Custom Wire Load Model
  - LEF capacitance coefficients

- Many extraction tools provide connectivity analysis:
  - 0’th order LVS

- **2-D extraction**
  - Simple parallel plate model

- **3-D extraction**
  - Solve for real 3-D geometries of wiring

- **2.5-D extraction**
  - Use look-up table for topologies
2.5-D Extraction

- Compromise between speed and accuracy
- Generates coefficients through solving the 3-D equations for “representative” sample of topologies
  - Really, cross-sections through “tunnel” that contains a section of the victim net
  - Creates look-up table
  - Time consuming, but only done once
  - Each layer of interconnect added roughly doubles time for coefficient generation
- Extraction matches topologies to entries in look-up table
Extraction for Cross Talk Analysis

- Cross talk analysis requires knowledge of capacitive coupling between nets
  - 2.5-D signal extraction models capacitance to ground plane
  - Net to net capacitance required
Extraction to To Floating Metal

- Extraction to floating metal similar to extraction for cross talk analysis
  - Net to net capacitance required
  - Effective capacitance to floating metal dependent on potential of floating metal
  - E.g., Cadence HyperExtract models floating metal as grounded
  - If we model floating metal as grounded, this is pessimistic

- Below 0.18µm with “local fill” requirements, fill metal can impact timing
2.5D Extraction Data

- **Input data**
  - Technology data
    - Metal and via resistances
    - Capacitance coefficients
  - Library data
    - Input pin capacitances
  - Design data
    - Routing
    - Boundary conditions (load and drive information)

- **Output data**
  - Parasitic information:
    - DSPF
    - RSPF
    - Set_load
  - Interpreted parasitic information
    - Custom WLM
    - LEF coefficients
Set_load

- **Set_load** is a report of lumped C on per net basis.

- **Set_load** is used by Synopsys tools:
  - In-Place Optimization
  - Custom WLM generation

- **Net based vs. instance based parasitic conventions**:
  - Cadence is instance based
  - Synopsys is net based
Custom WLM

- Custom WLM provides a more accurate WLM for design than generic library WLM
- Physical design tools generate WLM for entire physical block
- Synthesis tools can read parasitic data and build custom logical WLM for logical blocks inside design, not just physical blocks
- Custom WLM is better able to account for, e.g., congestion
  - Congested block: more neighbor coupling, more detouring \rightarrow larger capacitances
LEF Coefficient Correlation

- LEF capacitance values are 2-D
  - CPERSQDIST
  - EDGECAPACITANCE

- Capacitance coefficients are statistical in DSM
  - Effective area and edge capacitance dependent on surrounding routing
  - Congested blocks have higher effective capacitance

- Need to route design during floorplanning to generate LEF coefficient data

- Need to modify the LEF coefficients on block by block basis
Why WLMs?

- Synthesis needs placement parasitics
- Placement needs synthesized netlist
- Chicken vs. egg
- WLMs estimate parastics pre-placement
- Wire cap = f(net fanout) (but WLM has to pick ONE value)

Are WLMs “no longer accurate”? (and, is this a problem?)
Kinds of WLMs

- **Custom**
  - multiple iterations

- **Statistical**
  - from library

- **Structural**
  - look at adjacent nets
Historical Design Flows

- No WLM
  - interconnect insignificant

- Custom WLMs
  - requires iterations

- Statistical WLMs
  - fewer iterations

- Add post-placement optimization
  - IPO, PB-Opt

- Post-placement only (integrated tool)
  - Cadence PKS, Synopsys Physical Compiler, Magma Blast Fusion
Motivating Questions and Background

- How inaccurate are WLMs?
- Are WLMs good for optimization?
- Are WLMs good for estimation?
- How to use WLMs in optimization flow?

Related thinking

- “Bad” WLMs motivate post-placement synthesis

- Need to constructively estimate
  - Scheffer and Nequist, Proc. ACM SLIP 2000, pp. 139-144
  - “Law of small numbers”: worst-case is an extreme order statistic, and cannot be accurately predicted

- Accuracy vs. fidelity
  - Boese et al., Proc. IEEE ICCD 1993, pp. 81-84
  - Inaccurate estimates can still enable correct choices between options A and B
Example Custom WLM Construction

1. Write out net and module data
2. Cluster modules
3. Group net fanouts within module clusters
4. Construct WLMs for each cluster using multi-variable linear regression

Details
- Combine modules into clusters so that each module cluster contains at least as many nets as some given target number of nets, e.g., 2000. Each module cluster will have its own WLM calculated for it.
- To cluster the modules, sort them by cell area. Restrict the clusters to each contain a contiguous set of modules in the sorted area. Use dynamic programming to cluster the modules based on maximum difference between the number of nets in each cluster and the target cluster size.
- Break ties by average squared difference between the number of nets in a cluster and the target cluster size.
How Accurate Are WLMs?

- High Deviations $\Rightarrow$ WLMs Inaccurate
- Normalized to Average Net Capacitance

![Wire Cap Variation (Fanout 1)](image)
WLM Accuracy: Skew Measures

- Bad for short nets (not critical, but overestimated)
- Bad for long nets (critical, but underestimated)
- What if long nets not critical?
  - timing-driven placement
  - buffer insertion
- Must test WLMs in actual flows
Are WLMs Good for Optimization?

- Sufficient?
- Necessary?
- Compare different flows
  - real tool (PKS)
  - real designs
Default Cadence PKS Flow

- RTL
- Generic Netlist
- Mapped Netlist
- Placed Netlist
- Routed Netlist

- Area opt
- Technology mapping
- Timing opt
- Placement
- Post-placement timing opt
- Global routing
Flow Variations

- Structural WLM
- Library WLM
- Custom WLM
- No WLM
- No Pre Pl Opt
- No Post Pl Logic Opt
- No Post Pl Opt

1. Area opt
2. Technology mapping
3. Timing opt
   - WLM or No WLM
4. Placement
5. Post-pl timing opt
   - Opt or No Opt
6. Global routing
Flow Comparisons

**Normalized Run Time**

- Struct WLM
- Library WLM
- Custom WLM
- No preOpt
- No WLM
- No postLogicOpt
- No postOpt

**Average Slack**

- Struct WLM
- Library WLM
- Custom WLM
- No preOpt
- No WLM
- No postLogicOpt
- No postOpt
Implications

- WLMs not sufficient to achieve good slacks
  - post-place opt needed

- WLMs not necessary for good slacks (?)

- But, WLMs reduce running time

- Some improvement from post-placement logic optimization

Caveats

- Specific tool, version, designs
- Global routing (not detailed or extracted)
- No clock tree
- Routability not verified
- Die size fixed
- Some designs pre-optimized
Do WLMs Give Good Estimates?

- WLM slack vs. GRoute slack
- Custom WLM
How To Use WLMs?

- How much optimism?

- How much pessimism?
  - E.g., “standard trick”: overclock the design in synthesis if you want to make timing
  - Two “generic” knobs for timing-driven synthesis: (1) play with the percentiles that determine the WLM, or (2) play with the frequency target.

  What do you think will happen to the P&R result as you change the percentiles that determine the WLM used in synthesis (and, should these percentiles differ for each number of pins), and as you change the frequency target used in synthesis?

- Which WLM type is best?
  - Structural?
  - Custom?

- Multi-start?
  - Meta-heuristic: Run many heuristics, take best result
  - Different WLMs / optimism levels
  - Could randomness save us?