Timing Closure Today

Tony Drumm
IBM Corporation
Rochester, MN
drumm@us.ibm.com

Agenda

- Traditional design flows
- Timing analysis
- Transformations for timing correction
- Some of the current problems
- Post-placement optimization
- Summary
Agenda

- Traditional design flows
- Timing analysis
- Transformations for timing correction
- Some of the current problems
- Post-placement optimization
- Summary

Traditional Design Flows

- Design Entry
- Synthesis
- Place
- Route
- Timing

1. Tech independent optimization
2. Tech mapping
3. Rudimentary timing correction
Logic Synthesis Flow

- Technology independent optimization
  - General goal: reduce connections, literals, redundancies, area
- Technology mapping
  - Map logic into technology library
- Timing correction
  - Find and fix critical timing paths
  - Fix electrical violations (load, slew)

Traditional Design Flows

1990's

- Design Entry
- Synthesis w/Timing
- Place w/Timing
- Route
- Timing

Integrate timing with synthesis and placement

1. Tech independent optimization
2. Tech mapping
3. Timing correction
Tools of Timing Correction

Transformations  Drivers

Analysis  Checking

Tools of Timing Correction

Transformations  Drivers

Analysis  Checking
Timing Correction Method

- What to apply?
  - Transformations (or transforms)
    - Change the structure of the design without changing the function of the design

Tools of Timing Correction

Transformations

Drivers

Analysis

Checking
Timing Correction Method

- Where to apply it?
  - Drivers
    - Traverse the design to find the best place to apply transforms

Tools of Timing Correction

- Transformations
- Drivers
- Analysis
- Checking
Timing Correction Method

- Is it beneficial?
  - Analysis
    - Timing: Static Timing Analysis
    - Others: Noise analysis, power analysis, etc

Tools of Timing Correction

- Transformations
- Drivers
- Analysis
- Checking
Timing Correction Method

- Is it legal?
  - Checking
    - Logical correctness
    - Electrical correctness

Agenda

- Traditional design flows
- Timing analysis
- Transformations for timing correction
- Some of the current problems
- Post-placement optimization
- Summary
Timing Analysis

- Give accurate time values on each pin/port of the network
- Has to deal with design changes in optimization toolbox
- *Static* Timing Analysis
  - Simulation far too slow in optimization environment
  - Accuracy is more than enough

Timing Analysis Requirements

- Choose combination of timing analyzer and delay calculator which are appropriate for level of design
  - give the best accuracy
  - for performance that can be tolerated
- Timing Analysis / Delay calculation must be able to cope with logic design changes
  - Incremental
  - Highest performance possible
  - Non-linear delay equations
Timing Analysis Requirements

- Must handle…
  - Difference between rising and falling delays
  - Delay dependent on slew rate
  - Slew and delay dependent on output load
  - Non-linear delay equations

Late Mode Analysis Definitions

- Constraints: assertions at the boundaries
  - Arrival times: $AT_a$, $AT_b$
  - Required arrival time: $RAT_x$
- Delay from $a$ to $x$ is the longest time it takes to propagate a signal from $a$ to $x$
- Slack is required arrival time - arrival time.
Example

\[ SL_a = 0 - 0 = 0 \]
\[ AT_a = 0 \]
\[ AT_b = 1 \]
\[ SL_b = 0 - 1 = -1 \]
\[ AT_c = 0 \]
\[ SL_c = 1 - 0 = 1 \]
\[ SL_y = 1 - 2 = -1 \]
\[ RAT_x = 2 \]
\[ AT_x = 3 \]
\[ SL_x = 2 - 3 = -1 \]

Early mode analysis

- Definitions change as follows
  - longest becomes shortest
  - slack = arrival - required
Delay modeling

Agenda

- Traditional design flows
- Timing analysis
- Transformations for timing correction
- Some of the current problems
- Post-placement optimization
- Summary
Timing Correction

- Fix electrical violations
  - Resize cells
  - Buffer nets
  - Copy (clone) cells
- Fix timing problems
  - Local transforms (bag of tricks)
  - Path-based transforms

Local Transforms

- Resize cells
- Buffer or clone to reduce load on critical nets
- Decompose large cells
- Swap connections on commutative pins or among equivalent nets
- Move critical signals forward
- Pad early paths
- Area recovery
Transform Example

Delay = 4

Delay = 2

Resizing

0.01
0.02
0.03
0.04
0.05

0.2
0.2
0.3

Load

0 0.2 0.4 0.6 0.8 1

A

B

C
Cloning

Buffering
Redesign Fan-in Tree

Redesign Fan-out Tree

Longest Path = 5

Longest Path = 4
Slowdown of buffer due to load
Decomposition

Swap Commutative Pins

Simple Sorting on arrival times and delay works
Move Critical Signals Forward

- Based on ATPG
  - linear in circuit size
  - Detects redundancies efficiently
- Efficiently find wires to be added and remove.
  - Based on mandatory assignments.

Path-based Transforms

- Path-based resizing
- Unmap / remap a path or cone
- Slack stealing
- Retiming
Slack Stealing

- Take advantage of timing behavior of level sensitive registers (latches)

\[
\begin{align*}
\text{C1} & \quad 0 & \quad 1 & \quad 2 \\
\text{C2} & \quad \quad & \quad & \\
\text{Slack} & \quad -1 & \quad +1 & \quad 0
\end{align*}
\]

No change to logic!

Retiming

- Forward
- Backward

\[
\begin{align*}
\text{Delay} & \quad = 3 \\
\text{Delay} & \quad = 2
\end{align*}
\]
Agenda

- Traditional design flows
- Timing analysis
- Transformations for timing correction
- Some of the current problems
- Post-placement optimization
- Summary

The Wall

- Logic designers concentrate on logic and timing
- Design work done in abstract world of gates and wire load models
- Throw design over the wall when complete
- Physical designers concentrate on layout and ability to route
- Effective method for many years
General CMOS Problems

- Small voltages
- Low drive strengths / low power
- Capacitance plays a large role in performance
- Noise
- Variability – range between slowest possible and fastest possible

Additional DSM Problems

- High density / huge designs
- Very thin and resistive wires
- Very high frequencies
- Smaller voltages
- Clock skew and latency
- Electromigration and noise
Clock Distribution Problems

- Most common design approach requires close to zero skew
- CMOS / DSM problems all affect clocks
- Distribution problem increasing
  - Number of latches/flip-flops growing significantly
- Power consumed in clock tree significant
  - $\Delta I$ and noise also of concern

Problem Mitigation

- Carry hierarchical logic design into physical
- Many metal layers
- Copper wires, thick wires to lower R
- Other technology improvements
  - SOI, low k
- More sophisticated clock designs
- Improved analysis
- Custom design
Hierarchy and Physical Design

- Logical hierarchy can be carried over into physical design
- Seems natural top-down approach, using floorplanning as a firm guide to physical design

Pros...

- Run time of P&R tools
- Early *(and valuable)* knowledge of global wires
- Wire delay within macro may be tolerable
- Generally, contain the problem size
- May be the only real method available
**Hierarchy and Physical Design Cons...**

- Placement solution bounded
- Ability to find a routable solution hindered
- Hierarchy usually logically-based, not physically-based
- Boundary conditions explode and must be managed carefully to avoid surprises
- Pin assignment problem for all macros

**Hierarchy Example Plots**
Hierarchy Example Plots
Correlation Pre/Post-P&R

Many of the problems reduce to this:

- How good is the correlation between Pre-P&R estimates and Post-P&R extraction?
- If correlation is good…
  - Problems detected and potentially fixed *early*
- If correlation is bad…
  - Problems detected *late*
  - Not a good situation!

Correlation Pre/Post-P&R

The problem

- Can’t complete layout until logic design is complete
- Can’t complete logic design without timing
- Can’t time without load and net delay data
- Can’t extract load and net delay data until layout is complete
- Can’t complete layout …
Correlation Pre/Post-P&R
Common solution
- Don’t know specific layout data
- But we know something about statistical properties
- Average net load, average net delay
- Further refine using other characteristics
  - Number of sinks
  - Size of design (number of circuits)
  - Physical size

Correlation Pre/Post-P&R
Using averages
- *Wire load models* give synthesis an *estimate* of physical design
- We can correlate averages pre- and post-P&R as accurately as needed
- If specific design has average behavior, its timing, *on average*, can be predicted
- Otherwise, a pass through placement can provide data needed to adjust model
Correlation Pre/Post-P&R Timing and averages

- Statistics are important to timing analysis
- However, cycle time dictated by the worst specific path
- That path is built of individual nets
- One net can determine the speed of an entire design
- Reality: poor correlation for relatively few nets can cause major headaches

Correlation Pre/Post-P&R Averages and Wire Loads

Distribution of C / fan-out

- Number of nets
- pF per fan-out
Correlation Pre/Post-P&R

$C_{wire}$ Data by Logic Design

- How can we use information from one pass through physical design?
- Adjust wire load model coefficients
- *Back annotate* specific net load and delay data to the logic design
- New problem: correlation of logic pre- and post-synthesis
- This correlation may be impossible
Transform Example

Double Inverter Removal

Delay = 4

Delay = 2

Agenda

- Traditional design flows
- Timing analysis
- Transformations for timing correction
- Some of the current problems
- Post-placement optimization
- Summary
Post-Placement Optimization

1. In-place optimizations
2. Minimally disturb placement optimizations

- In-place
  - Resizing (carefully)
  - Pin swapping, some tree rebuilding
  - Wire sizing / typing
- Minimally disruptive
  - Resizing
  - Buffering
  - Cloning
  - Tree rebuilding
  - Cell removal
In-place Optimization

- Not *too* difficult
- Can use extracted electrical data (C, RC) from placement tool
  - Some changes affect pin locations, but may be ignored
  - Tree rebuilding needs incremental extraction
- Can use timing reports for timing data
  - But, accuracy suffers as changes are made
Place-disruptive Optimization

- Nets changing implies...
  - Must be able to recompute C and RC
  - May need to incrementally place new cells
  - Need incremental timing capability

Place-disruptive Optimization

- Placement & extraction
- Placed netlist
- C/RC data
- Opt’d netlist
- Optimization with placer, timer, extractor
- Resize buffer
- Clone
- Cell removal
- Rebuild trees
Post-Placement Example - Buffering long wires

Sample Optimization Results

<table>
<thead>
<tr>
<th>Design</th>
<th>Worst slack / # misses</th>
<th>Cycle time</th>
<th>Tech</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>preplace</td>
<td>placed</td>
<td>opt</td>
</tr>
<tr>
<td>C1</td>
<td>-1 / 2000</td>
<td>-12 / 38k</td>
<td>-2 / 1400</td>
</tr>
<tr>
<td>V1</td>
<td>0 / 0</td>
<td>-12 / 15k</td>
<td>-0.3 / 100</td>
</tr>
<tr>
<td>T1</td>
<td>-0.5 / 2000</td>
<td>-48 / 164k</td>
<td>-6 / 62k</td>
</tr>
<tr>
<td>P1</td>
<td>-0.4 / 100</td>
<td>-97 / 43k</td>
<td>-13 / 20k</td>
</tr>
<tr>
<td>V2</td>
<td>-0.5 / 500</td>
<td>-11 / 2000</td>
<td>-4 / 1000</td>
</tr>
</tbody>
</table>
Fix My Clocks… *Please!*

- Zero skew is not necessary
- CMOS uncertainties are worrisome, but…
- Freedom to adjust clocks arrival times at memory elements provides another powerful timing closure tool
- Similar to retiming but less disruptive

---

**Post-Placement Challenges**

- Getting the timing right
  - Different timers used at different stages
  - Does the optimizer see the same worst paths as the sign-off timer?
  - *Timing analysis is an act of faith* – Drumm’s third law
- Design size / tool capacity
  - Using synthesis technology on flat designs
Post-Placement Challenges

- Incompatible tools, formats
  - Placer, synthesizer, timer may all use different file format, may all be different vendors
  - Basic interoperability issues
- Incremental placer needed for new cells
  - Doesn’t have to be smart
  - But might produce some infeasible solutions
  - Must be integrated with optimizer

Post-Placement Challenges

- Extraction of net data
  - Steiner tree estimation
  - Net C and delay (RC) calculator
  - Do results match other extraction tools?
  - Any optimization which significantly alters net topology needs this ability
    - Insert cells
    - Remove cells
    - Move connections except locally
Good News

- The biggest improvements come from the simplest transformations
  - Resizing easiest – most important
  - Buffering – fixes many placement-induced misses
  - Pin swapping and tree rebuilding – don’t require a placer, don’t affect placement at all

Bad News

- Cycle time and technology advances demand more and more sophisticated optimization techniques
- Disconnect among various tools involved increases turn-around-time and limits optimization
Good News

- The Bad News is commonly recognized
- Many tool vendors, academics, in-house EDA researchers are working to solve these problems
- The problems will be solved

Bad News

- These problems won’t be the last!
- Like frequencies, density, complexity – new problems will be discovered at an ever increasing rate
Agenda

- Traditional design flows
- Timing analysis
- Transformations for timing correction
- Some of the current problems
- Post-placement optimization
- Summary

Summary

- Timing correction methods fairly mature
- Technology and frequency present new problems
- Synthesis techniques can be used successfully post-placement
- Better integration and interoperability are needed to solve newest problems and take us into the future
Acknowledgements

- Leon Stok
- Alex Suess
- José Neves
- Bill Joyner
- IBM Rochester EDA folks