Today

- Topological Worst Case
  - not adequate (too conservative)
- Sensitization Conditions
- Timed Calculus
- Delay-justified paths
  - Timed-PODEM
- Speedup

Topological Worst-Case Delay

- Compute ASAP schedule
  - Take max of arrival times
  - Apply node Delay

Conservative

- Topological Worst-Case Delay can be conservative

[Fig/Examples from Logic Synthesis by Devadas, Gosh, Keutzer 1994]
Example

• Assume each gate 1:

6 delays in longest path,
(5 if assume c0 latest arriving)

Example

• Is this path possible?

• Out from mux 0 input
  • and10 = 0
  • p0=0 or p1=0
  • p1=0 1→6→7 not matter
  • p0=0 c0 not matter,
  • This path not feasible

False Paths

• Once consider logic for nodes
  – There are logical constraints on data values
• There are paths which cannot logically occur
  – Call them false paths

What can we do?

• Need to assess what paths are real
• Brute force
  – for every pair of inputs
  – compute delay in outputs from in1→in2
  – take worst case
• Expensive:
  – $2^n$ delay traces

Alternately

• Look at single vector and determine what controls delay of circuit
  – I.e. look at values on path and determine path sensitized to change with input

Controlled Inputs

• Controlled input to a gate:
  – input whose value will determine gate output
  – e.g.
  • 0 on a AND gate
  • 1 on a OR gate
**Static Sensitization**

- A path is statically sensitized
  - if all the side (non-path) inputs are non-controlling
  - i.e. this path value flips with the input

**Sufficiency**

- Static Sensitization is **sufficient** for a path to be a **true** path in circuit

**Static Co-sensitization**

- Each output with a controlled value
  - has a controlling value as input on path
  - (and vice-versa for non-controlled)

**Necessary**

- Static Co-sensitization is a **necessary** condition for a path to be true
Cosensitize path of length 6.
Real delay is 5.

Combining

• Combine these ideas into a timed-calculus for computing delays for an input vector

Computing Delays

AND
Timing Calculus

Rules

• If gate output is at a controlling value, pick the minimum input and add gate delay
• If gate output is at a non-controlling value, pick the maximum input and add gate delay

Example (1)

Example (2)
Now...

• We know how to get the delay of a single input condition
• Could:
  – find critical path
  – search for an input vector to sensitize
  – if fail, find next path
  – …until find longest true path
• May be $O(2^n)$

Better Approach

• Ask if can justify a delay greater than $T$
• Search for satisfying vector
  – …or demonstration that none exists
• Binary search to find tightest delay

Delay Computation

• Modification of a testing routine
  – used to justify an output value for a circuit
• PODEM
  – backtracking search to find a suitable input vector associated with some target output
  – Simple a branching search with implication pruning
    • Heuristic for smart variable ordering

Search1

• Takes in list of nodes to satisfy
• If all satisfied $\Rightarrow$ done
• Backtrace to set next PI
• if inconsistent PI value
  – try inverting this PI call Search2
• else
  – search to set next PI
  – if fail
    • try inverting and Search2

Search2

• ;; same idea, but this one not flip bit
• ;; because already tried inverted value
• If no conflict
  – search to set next PI
• otherwise
  – pass back failure

Backtrace

• Follow back gates w/ unknown values
  – sometimes output dictate input must be
    • (AND needing 1 output; with one input already assigned 1)
  – sometimes have to guess what to follow
    • (OR with 1 output and no inputs set)
    • Uses heuristics to decide what to follow
Example

Try justify g=1

For Timed Justification

• Also want to compute delay
  – on incompletely specified values
• Compute bounds on timing
  – upper bound, lower bound
  – Again, use our timed calculus
    • expanded to unknowns

Delay Calculation

AND rules

Timed PODEM

• Input: value to justify and delay T
• Goal: find input vector which produces value and exceeds delay T
• Algorithm
  – similar
  – implications check timing as well as logic

Example

Justify 1(3)
Example

Search

- Less than $2^n$
  - pruning due to implications
  - here saw a must be 0
    - no need to search 1xx subtree

Speed Up

- Start with area optimized network
- Know target arrival times
  - Know delay from static analysis
- Want to reduce delay of node

Basic Idea

- Improve speed by:
  - Collapsing node(s)
  - Refactoring collapsed subgraph to reduce height

Speed Up

- While (delay decreasing, timing not met)
  - Compute delay (slack)
    - Static timing analysis
  - Generate network close to critical path
  - Weight nodes in network
  - Compute $\text{mincut}$ of nodes on weighted network
  - Partial collapse and timing redecompose on cut nodes
Weighted Cut

- Want to minimize area expansion
- Want to maximize likely benefit
  - Prefer nodes with varying input times
  - Prefer nodes with critical path on longer paths

Timing Decomposition

- Extract area saving kernels that do not include critical inputs to node
- When decompose (e.g. into nand2’s) similarly balance with critical inputs closest to output

Example

Example

Example

Admin
Big Ideas

• Topological Worst-case delays are conservative
  – Once consider logical constraints
  – may have false paths
• Necessary and sufficient conditions on true paths
• Search for paths by delay
  – or demonstrate non existence
• Search with implications
• Iterative improvement