Topics

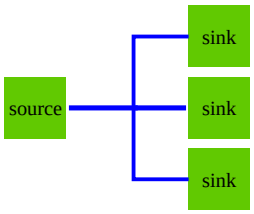
- Combinational network delay
- Logic optimization

Sources of delay

- Gate delay
  - Drive
  - Load
- Wire delay

Fanout

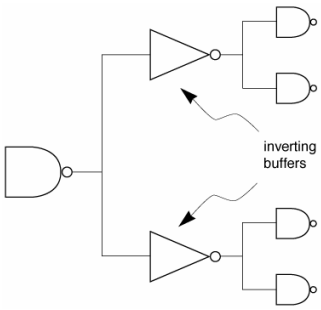
- Fanout adds capacitance



Ways to drive large fanout

- Increase sizes of driver transistors
  - Must take into account rules for driving large loads
- Add intermediate buffers
  - This may require/allow restructuring of the logic

Buffers



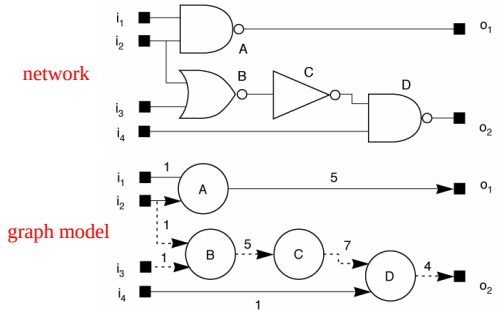
Announcements!

- HW5 posted!
  - Due 2/19
  - Try to do all HSPICE HWs
    - » HWs will ease the course project
- 0.25u and 0.13u libraries posted in Resources
  - A sample 0.25u inverter posted in Resources
- Midterm on 2/19
- Final
  - 1/1 or 1/3?

Path delay

- Combinational network delay is measured over paths through network
  - Can trace a maximum delay from inputs to worst-case output

Path delay example



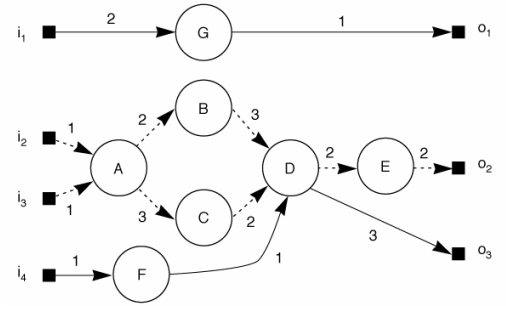
Critical path

- Critical path = path which creates longest delay
- Can trace transitions which cause delays that are elements of the critical delay path

Delay model

- Nodes represent gates
- Assign delays to edges
  - Signal may have different delays to different sinks
- Lump gate and wire delay into a single value

Critical path through delay graph



How to extract the critical path?

- Sort the nodes based on topological sort algorithm
- Traverse the list one by one
- Calculate delay of each node based on previous nodes on the list
- Extract the largest delay number
- What is the complexity?
  - $O(N^2)$, O(NlogN), or …?

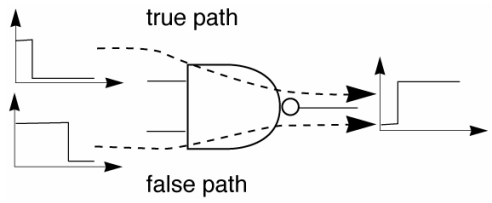
Reducing critical path length

- To reduce circuit delay, must speed up the critical path
  - Reducing delay off the path doesn't help
- There may be more than one path of the same delay
  - Must speed up all equivalent paths to speed up circuit
- Must speed up cutset through critical path

False paths

- Logic gates are not simple nodes
  - Some input changes don't cause output changes
- A false path is a path which cannot be exercised due to Boolean gate conditions
- False paths cause pessimistic delay estimates

False path example



Logic transformations

- Can rewrite by using subexpressions
- Flattening logic increases gate fanin
- Logic rewrites may affect gate placement

Logic optimization

- Logic synthesis programs transform Boolean expressions into logic gate networks in a particular library
- Optimization goals
  - Minimize area, meet delay constraint

Technology-independent optimizations

- Works on Boolean expression equivalent
- Estimates size based on number of literals
- Uses factorization, resubstitution, minimization, etc. to optimize logic
- Technology-independent phase uses simple delay models

Technology-dependent optimizations

- Maps Boolean expressions into a particular cell library
- Mapping may take into account area, delay
- May perform some optimizations on addition to simple mapping
- Allows more accurate delay models

Logic rewrites