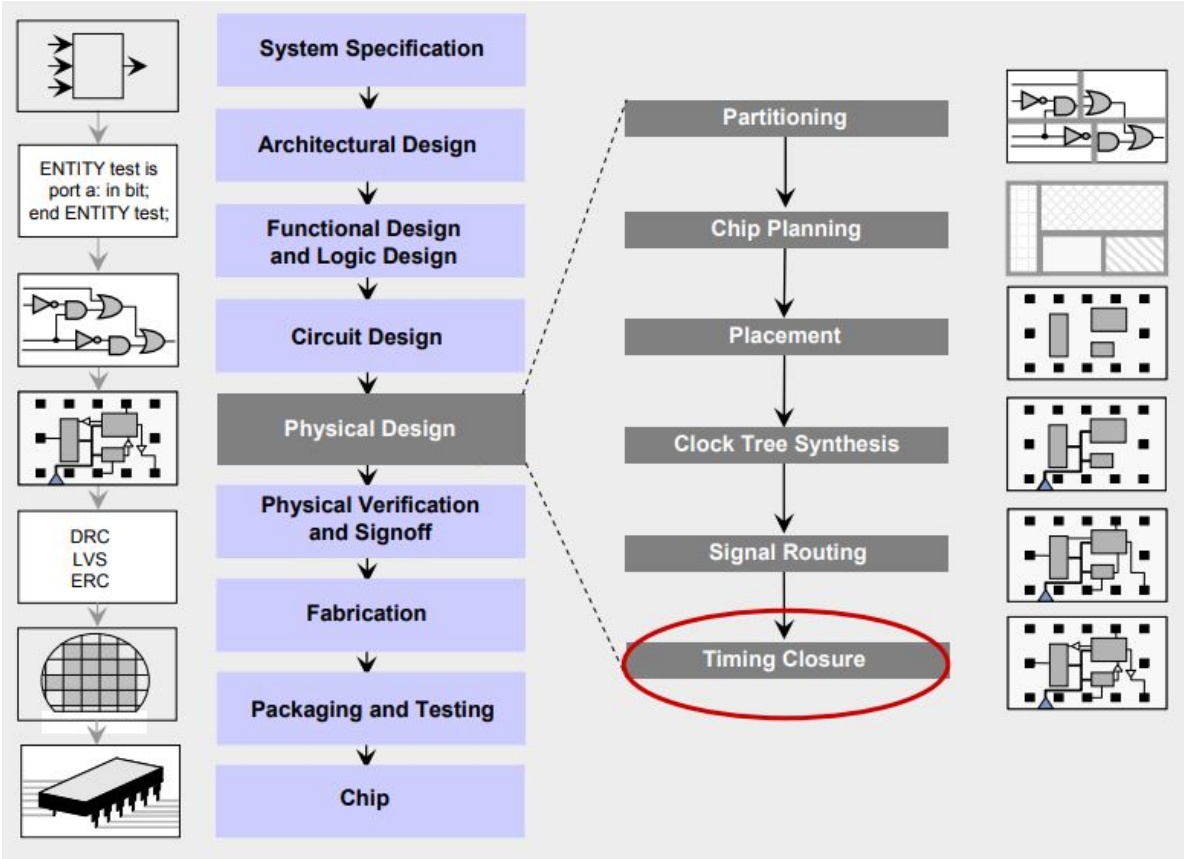
# Intro
# To
# Static Timing Analysis

# Introduction

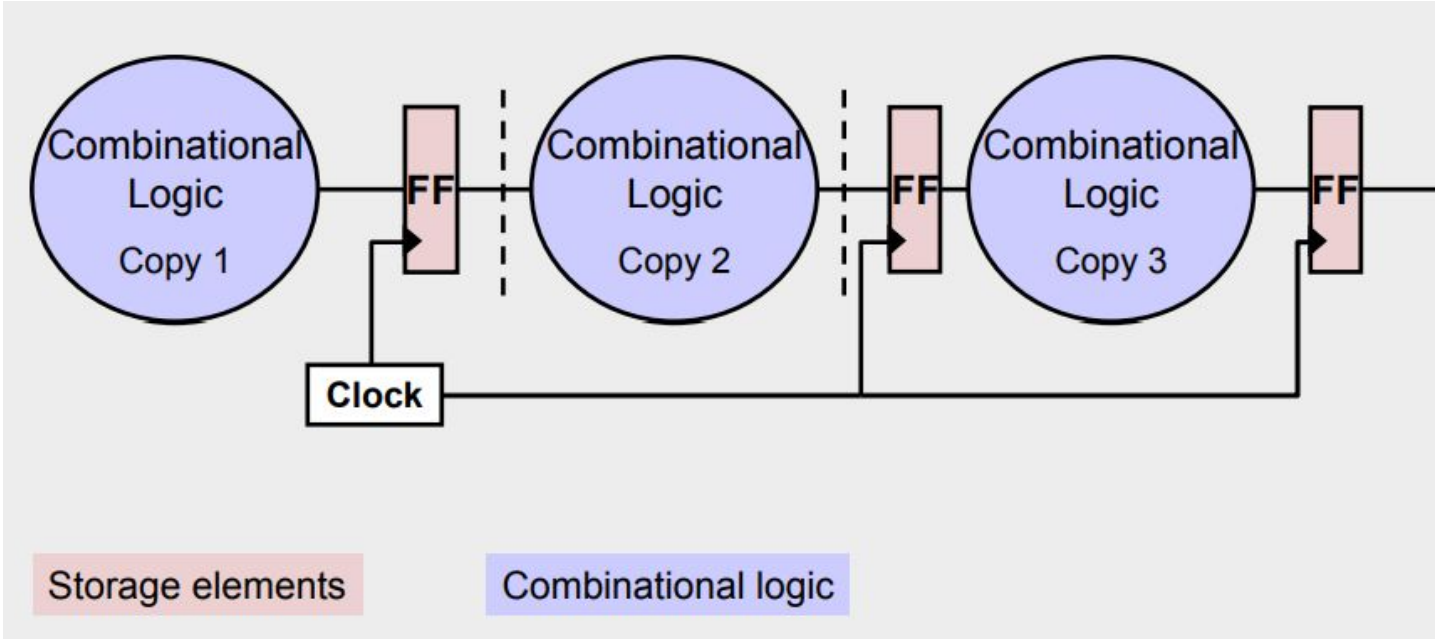
- FPGA Flow and ASIC Design Flow share a lot of similar processes



# Timing Closure

- Timing closure is the process of satisfying timing constraints through design and netlist modifications
- Designers should achieve Timing Closure
  - Style of Digital System Design
  - Optimization process that meet timing constraints (This is automated, so designer doesn't have control of this)
    - This is where design automation engineers play their role
  - Optimizations occur at Partitioning, Placement, and Routing
    - Timing-Driven Implementations **minimizes** Signal Delays

# Example

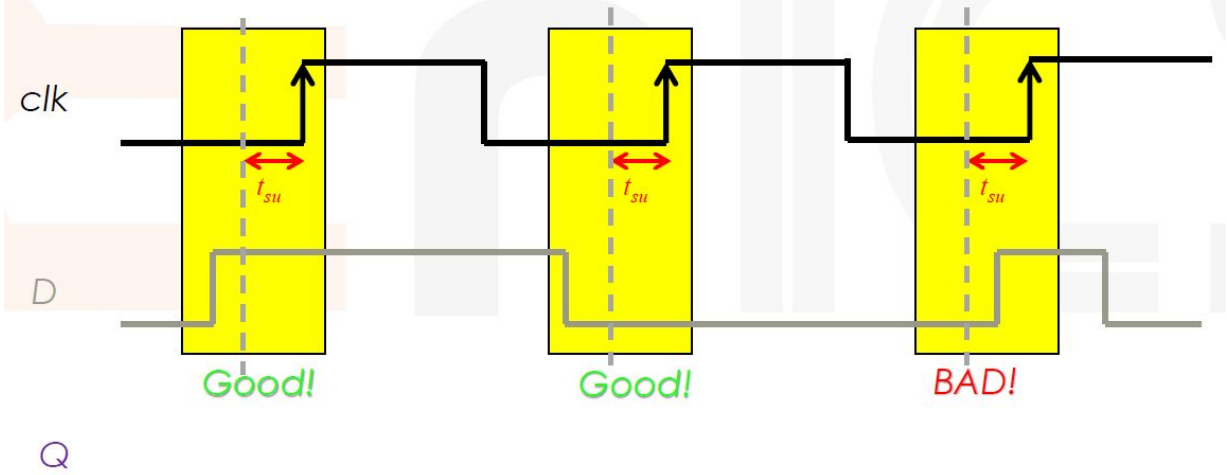

Combinational Logic
Copy 1
FF
Combinational Logic
Copy 2
FF
Combinational Logic
Copy 3
FF
Clock
Storage elements
Combinational logic

# Static Timing Analysis

- Main delay concerns in sequential circuits
  - **Gate delays** are due to gate transitions
  - **Wire delays** are due to signal propagation along wires
  - **Clock skew** is due to the difference in time the sequential elements activate

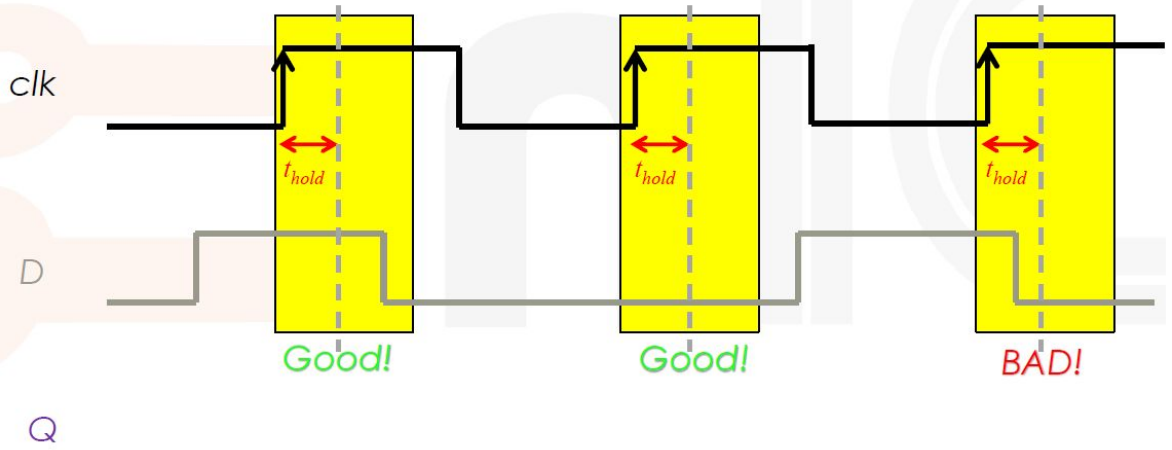
# Terminologies

- **Setup Time** - the time the data has to arrive before the clock to ensure correct sampling.



# Terminologies

- **Hold Time** -the time the data has to be stable after the clock to ensure correct sampling.

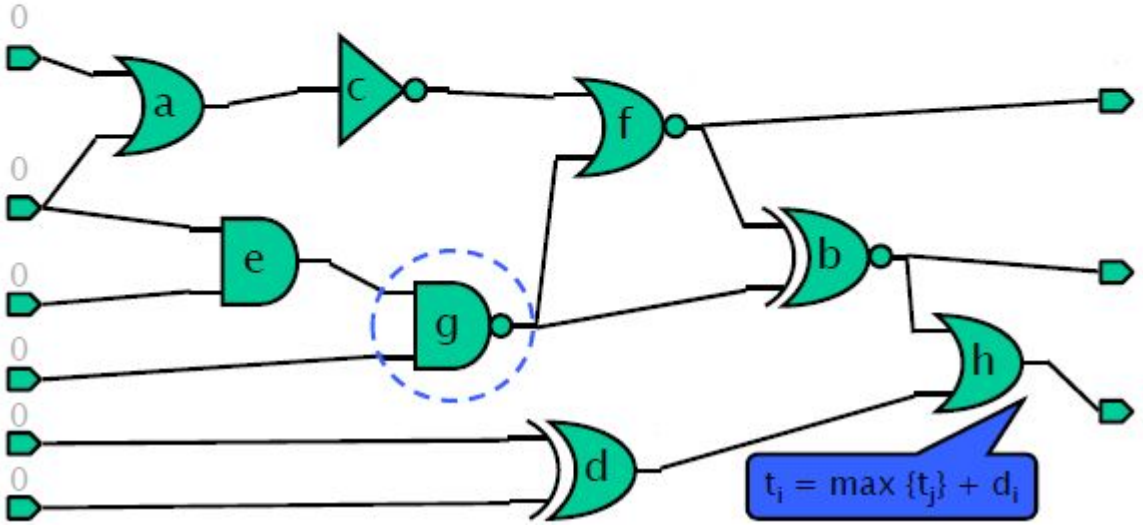


# Terminologies

- **Required Time** ($R_t$) - represents the time by which data has to be present at that node
  - Determined by the Clock Frequency
- **Arrival Time** ($A_t$) - represents the time at which the data arrives at the
  - Determined by Wire Delay and Gate Delay
- **Setup Slack = $R_t$ - $A_t$**
  - Commonly called as Slack
- **Hold Slack = $A_t$ - $R_t$**

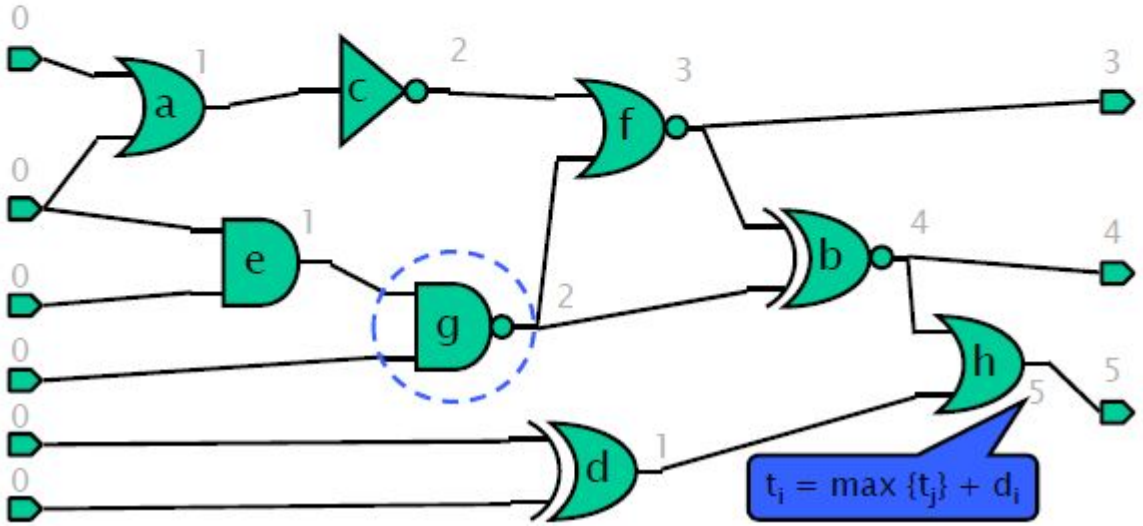
# Example - Arrival Time

- Assumptions:
  - All inputs arrive at time 0
  - All gate delays = 1
  - All wire delays = 0
- Question: Arrival time of each gate? Circuit delay?



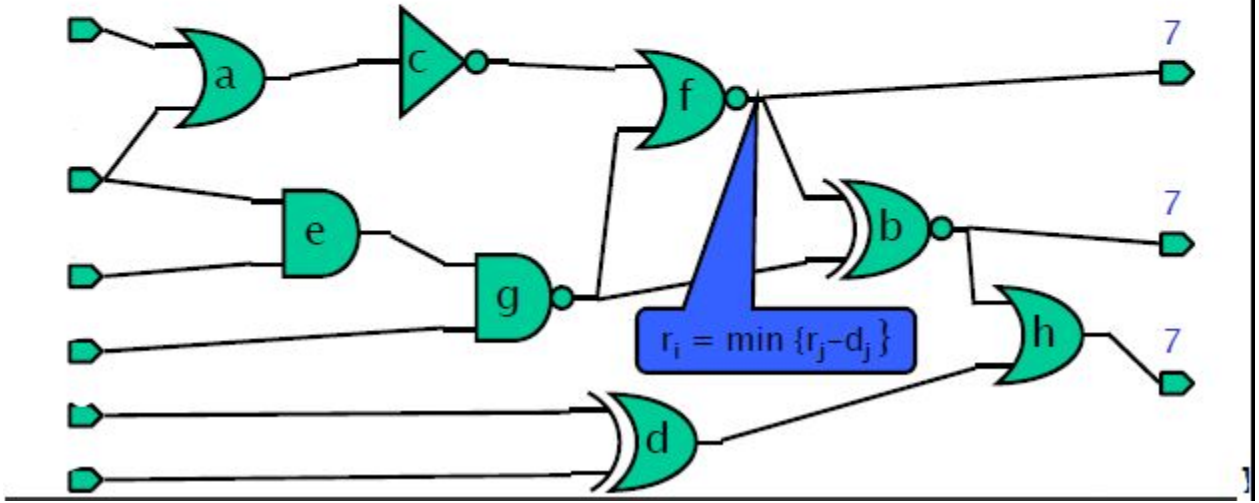
# Example - Arrival Time

- Assumptions:
  - All inputs arrive at time 0
  - All gate delays = 1
  - All wire delays = 0
- Question: Arrival time of each gate? Circuit delay?



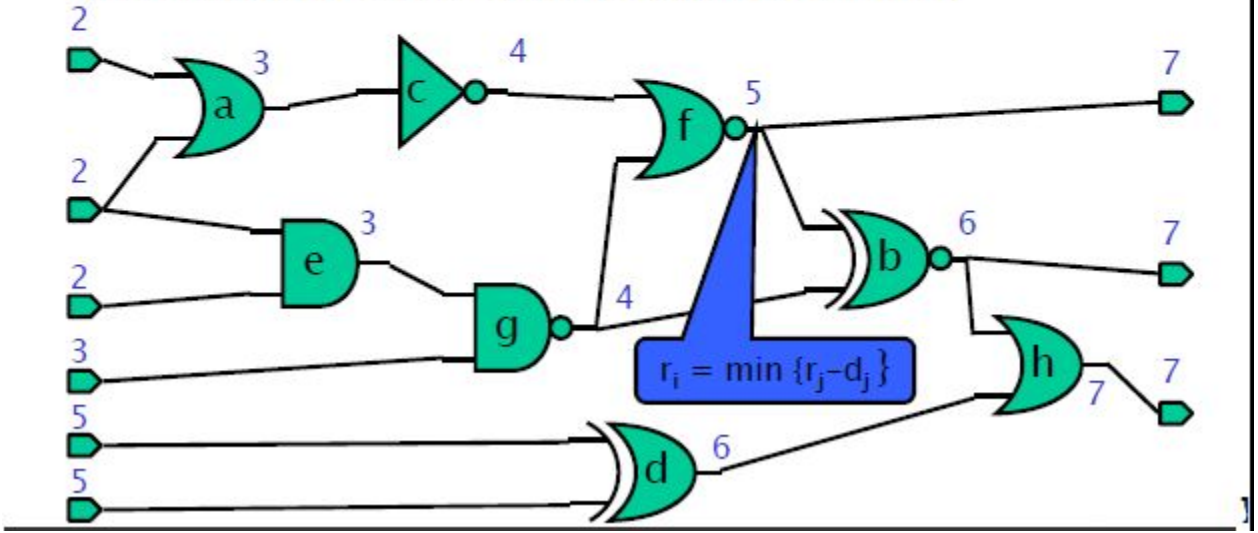
# Example - Required Time

- Assumptions:
  - All inputs arrive at time 0
  - All gate delays = 1, wire delay = 0
  - Clock period = 7
- Question: maximum required time (RT) of each gate? (i.e., if the gate output is generated later than RT, clk period is violated)



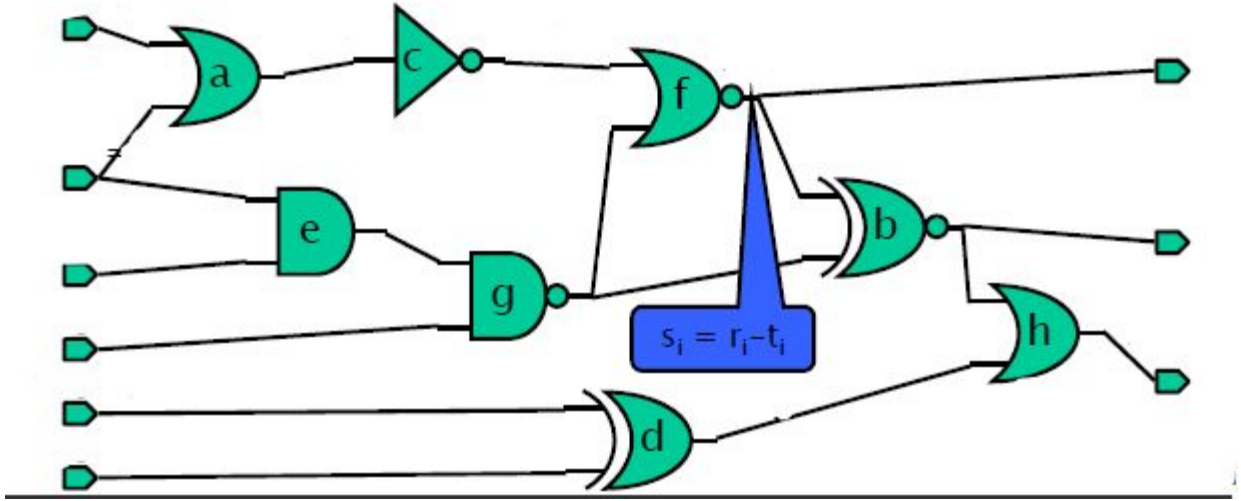
# Example - Required Time

- Assumptions:
  - All inputs arrive at time 0
  - All gate delays = 1, wire delay = 0
  - Clock period = 7
- Question: maximum required time (RT) of each gate? (i.e., if the gate output is generated later than RT, clk period is violated)



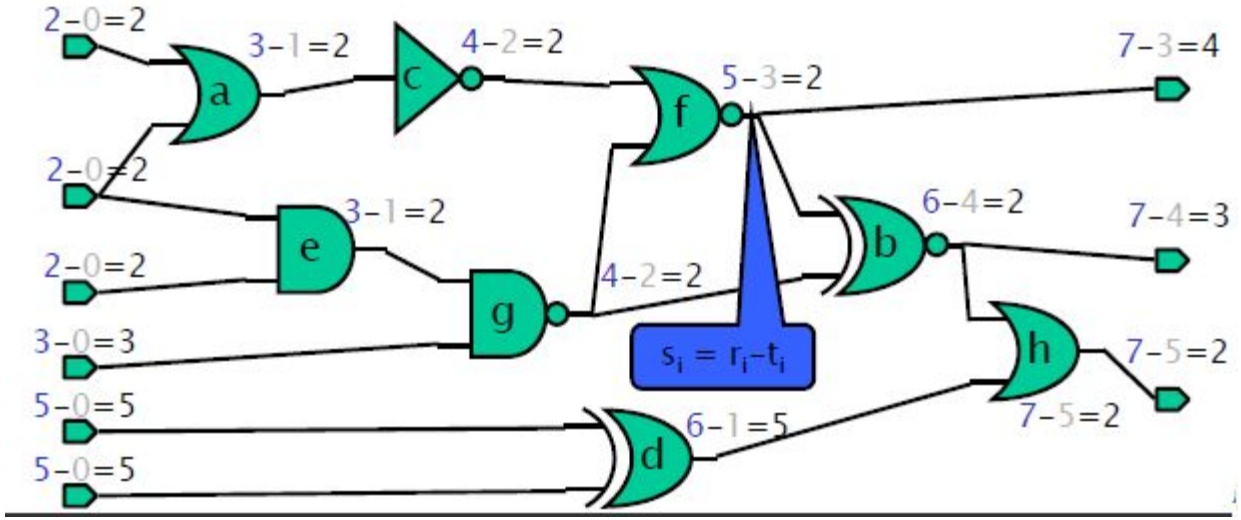
# Example - Slack

- Assumptions:
  - All inputs arrive at time 0
  - All gate delays = 1, wire delay = 0
  - Clock period = 7
- Question: What is the maximum amount of delay each gate can be slower not to violate timing?



# Example - Slack

- Assumptions:
  - All inputs arrive at time 0
  - All gate delays = 1, wire delay = 0
  - Clock period = 7
- Question: What is the maximum amount of delay each gate can be slower not to violate timing?



# Implications

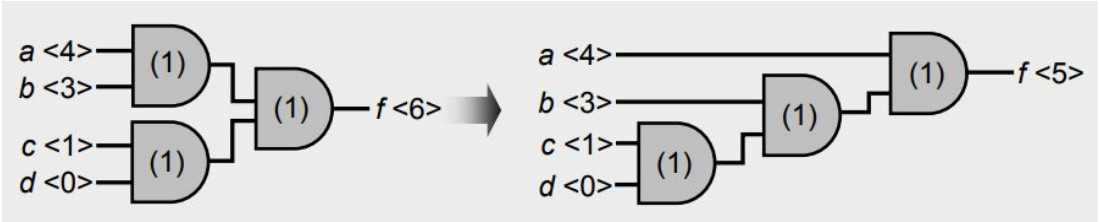
- **Negative slack** at any output means the circuit does not meet timing
- **Positive slack** at all outputs means the circuit meets timing

| Setup | | Hold | | Pulse Width | |
|---|---|---|---|---|---|
| Worst Negative Slack (WNS): | 5.631 ns | Worst Hold Slack (WHS): | 0.116 ns | Worst Pulse Width Slack (WPWS): | 4.500 ns |
| Total Negative Slack (TNS): | 0.000 ns | Total Hold Slack (THS): | 0.000 ns | Total Pulse Width Negative Slack (TPWS): | 0.000 ns |
| Number of Failing Endpoints: | 0 | Number of Failing Endpoints: | 0 | Number of Failing Endpoints: | 0 |
| Total Number of Endpoints: | 43 | Total Number of Endpoints: | 43 | Total Number of Endpoints: | 23 |

**All user specified timing constraints are met.**

# How to Achieve Timing Closure?

- Timing Corrections Include
  - Gate Sizing
  - Buffer Insertion
  - Netlist Re-Construction
    - Cloning: duplicating gates
    - **Redesign** of fanin or fanout tree: changing the topology of gates
    - Swapping communicative pins: changing the connections
    - Gate decomposition: e.g., changing AND-OR to NAND-NAND
    - Boolean restructuring: e.g., applying Boolean laws to change circuit gates



# END

Any Questions?