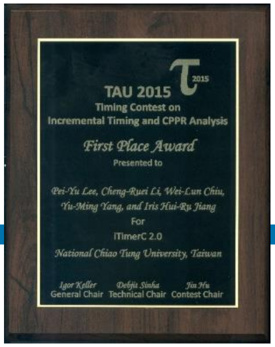
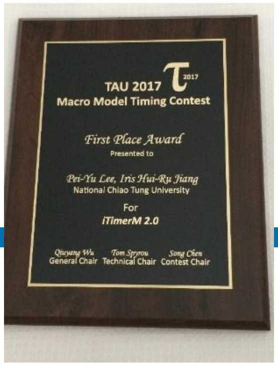
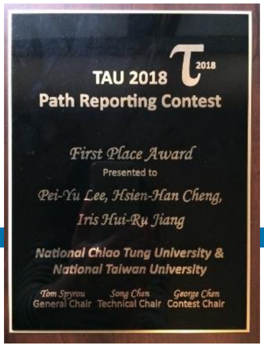
# About Iris~~

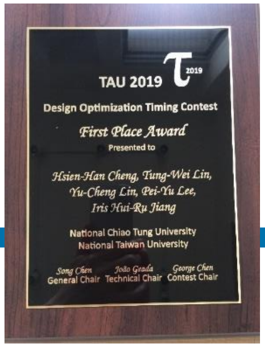
- Education
  - B.S & ph.D., Electronics, NCTU
- Experiences
  - Visiting scholar, IBM Austin Research Lab, 2013~14
  - Assistant/Associate/Full Professor at NCTU EE (02/2005 ~ 07/2017)
  - Professor at NTU EE (08/2017 ~ now)
- DATC chair
  - DATC: Design Automation Technical Committee
  - IEEE CEDA技術委員會首位女性主席/來自亞洲之主席，2016~21
- IEEE TCAD AE
  - 擔任EDA領域頂尖期刊副編輯
- DAC/ICCAD/ISPD General Chair/Program Chair/TPC
  - 擔任EDA領域頂尖會議議程委員
  - ISPD 2024 General Chair / ISPD 2023 Program Chair
  - ASP-DAC 2024 Program Chair / ASP-DAC 2023 Vice Program Chair
- 中國電機工程學會: 優秀青年電機工程師; 傑出電機工程教授

# IRIS Lab


TAU 2015
Timing Contest on
Incremental Timing and CPPR Analysis
*First Place Award*
Presented to
*Pei-Yu Lee, Cheng-Ruei Li, Wei-Lun Chiu, Yu-Ming Yang, and Iris Hui-Ru Jiang*
For
iTimerC 2.0
*National Chiao Tung University, Taiwan*
*Igor Keller* General Chair · *Debjit Sinha* Technical Chair · *Jin Hu* Contest Chair


TAU 2017
Macro Model Timing Contest
*First Place Award*
Presented to
*Pei-Yu Lee, Iris Hui-Ru Jiang*
National Chiao Tung University
For
*iTimerM 2.0*
*Qiuyang Wu* General Chair · *Tom Spyrou* Technical Chair · *Song Chen* Contest Chair


TAU 2018
Path Reporting Contest
*First Place Award*
Presented to
*Pei-Yu Lee, Hsien-Han Cheng, Iris Hui-Ru Jiang*
*National Chiao Tung University & National Taiwan University*
*Tom Spyrou* General Chair · *Song Chen* Technical Chair · *George Chen* Contest Chair


TAU 2019
Design Optimization Timing Contest
*First Place Award*
Presented to
*Hsien-Han Cheng, Tung-Wei Lin, Yu-Cheng Lin, Pei-Yu Lee, Iris Hui-Ru Jiang*
National Chiao Tung University
National Taiwan University
*Song Chen* General Chair · *João Geada* Technical Chair · *George Chen* Contest Chair

- Research interests:
  - 1) Timing analysis & optimization 積體電路時序分析與最佳化
  - 2) Design for manufacturability 積體電路製造可行性最佳化
  - 3) EDA for emerging technologies 新興技術之設計自動化
- 連續在EDA領域最頂尖之會議 DAC，ICCAD發表論文
  - DAC連續十二年 (2011~now), ICCAD連續十二年 (2011~now)
- 獲DAC best paper candidate, ISPD best paper nominee, ICCAD best-in-track paper, ASP-DAC best paper award
  - DAC '16 best paper nominee: 876篇投稿, 16篇提名, 提名率1.8%
  - ASP-DAC '20 best paper award: 263篇投稿, 2篇獲獎 (最高分)
- 連續在國際EDA研發競賽得獎
  - TAU Timing Analysis Contest連續九年(2013~2021)
  - 4次冠軍，冠軍紀錄保持團隊

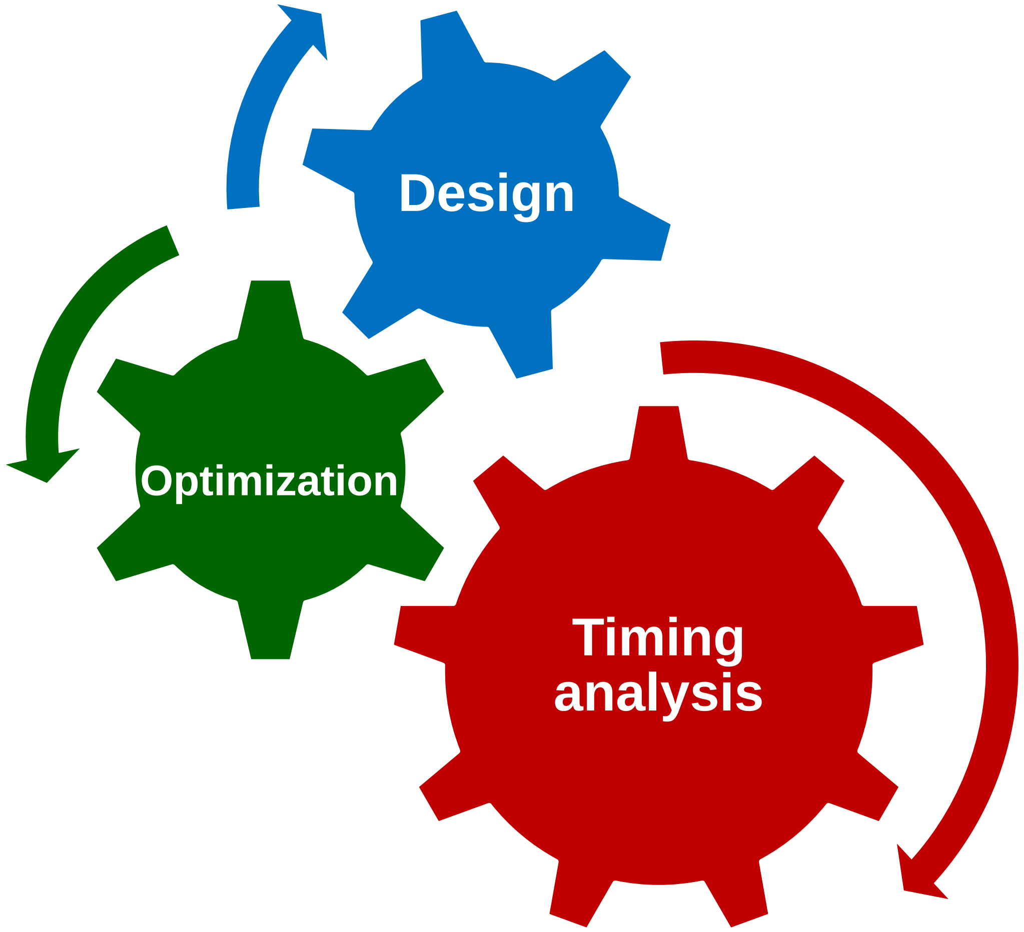
SYNOPSYS Predictable Success · IBM · tsmc · GUC · MEDIATEK

# A Quick Journey through Timing Analysis

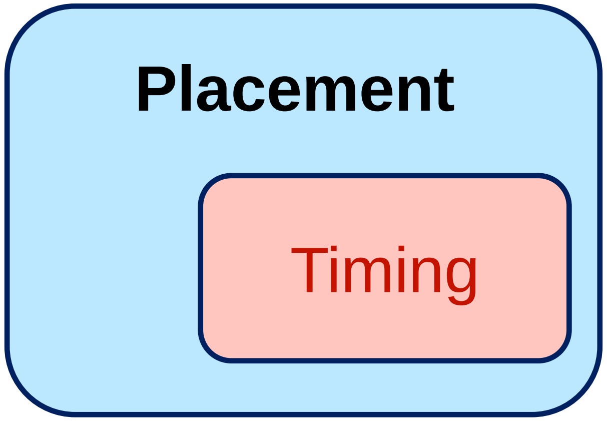
Iris Hui-Ru Jiang
Department of Electrical Engineering
Graduate Institute of Electronics Engineering
National Taiwan University

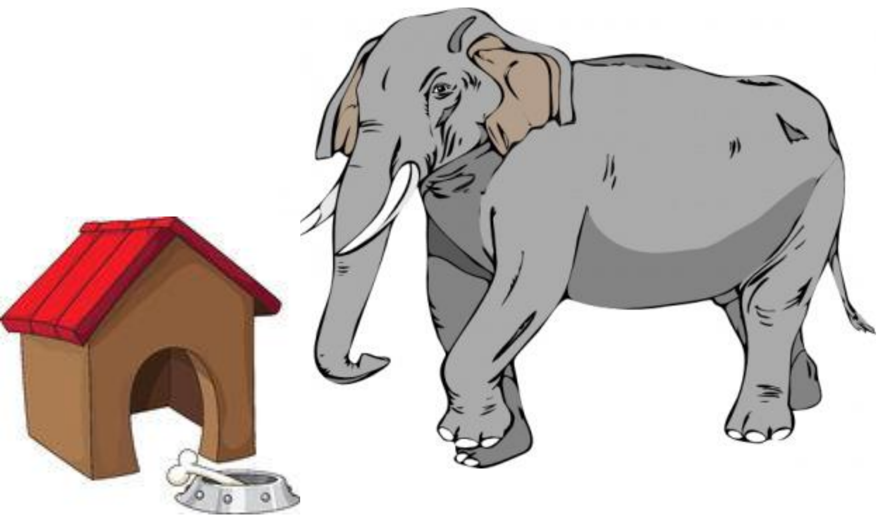
## Timing is Everything…

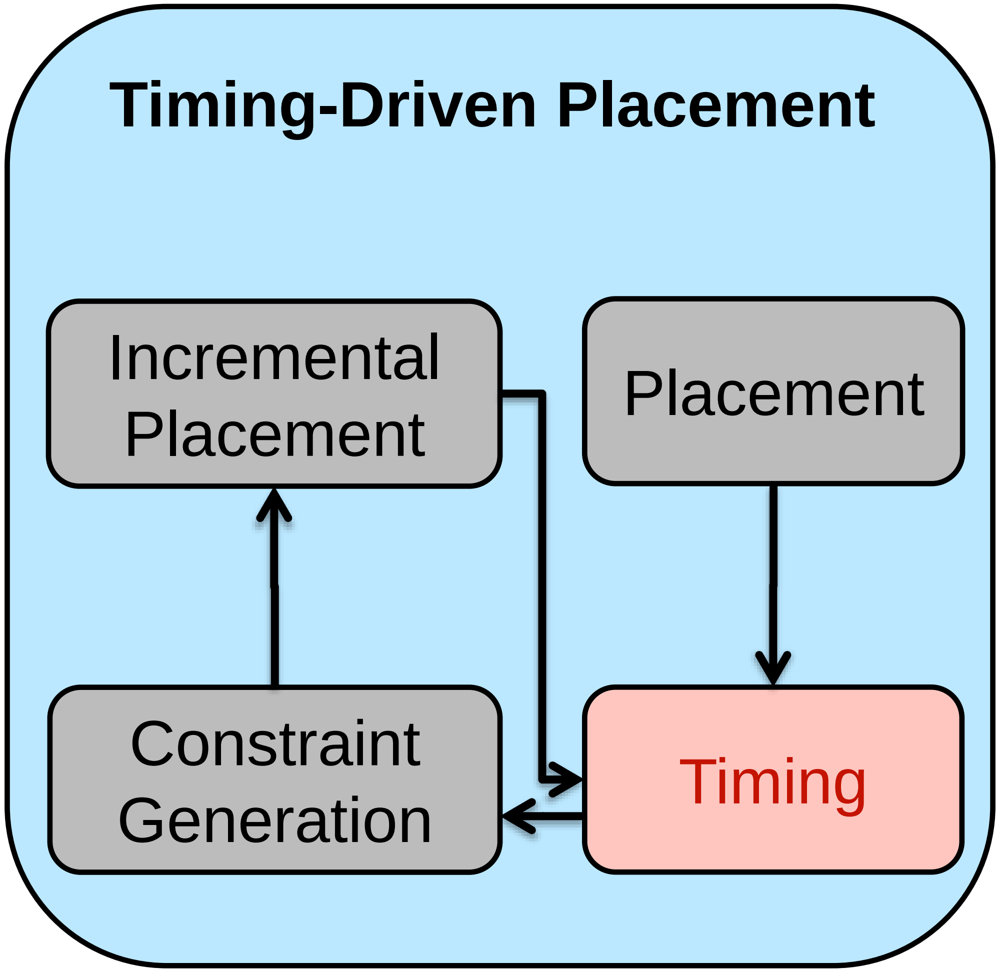
- Signals should arrive at the right place at the right time
- Timing analysis is essential in the modern IC design flow



# Don't Put Timer into Other Engines!







Courtesy of C. Alpert

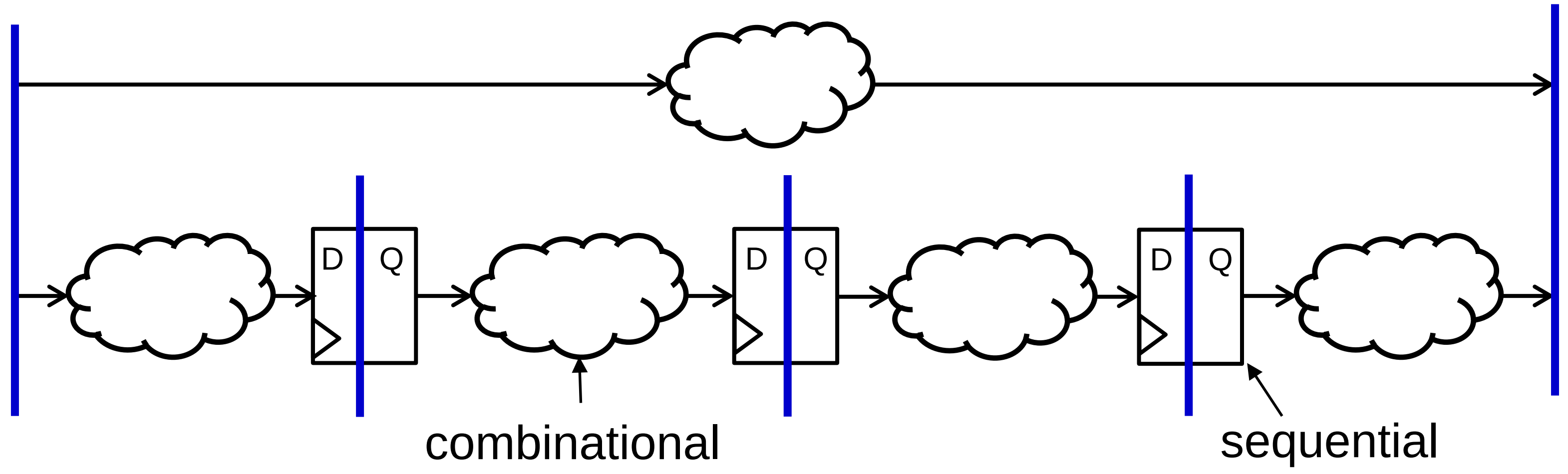
## Timing Analysis

- Categories
  - Functional timing analysis (FTA): concurrently consider input vectors and path sensitization to verify timing performance
    - Accurate but slow to achieve 100% coverage
  - Static timing analysis (STA): separate timing from functionality, consider the **worst-case** performance, calculate delay based on pure structural analysis
    - Fast but pessimistic
    - Need false path information

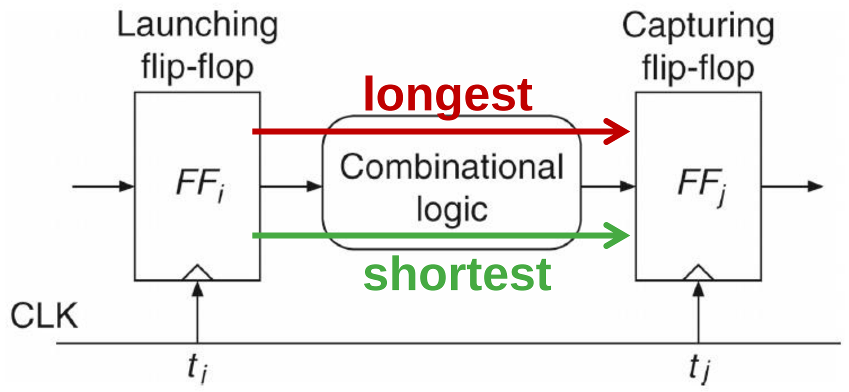
- Because of its high scalability for large scale designs, STA is widely adopted in the modern IC design flow (industry standard)

## STA SOP

1. Divide a design into timing windows based on flip-flops/latches and I/Os
2. Construct a timing graph (directed acyclic graph) for each window
3. Calculate the circuit delay
4. Check timing constraints

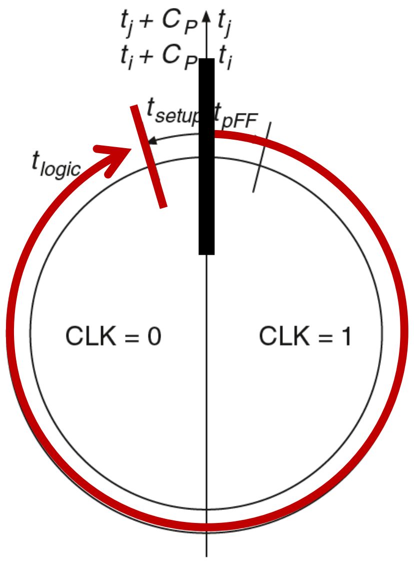


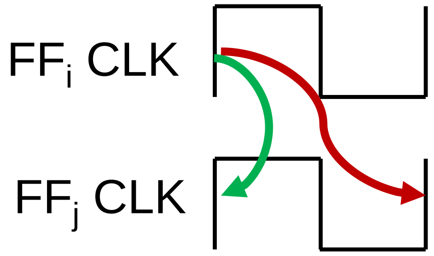
# Timing Check

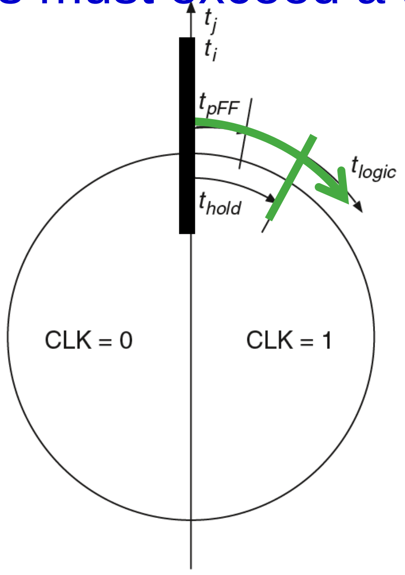


- **Setup time**: Delay between 2 flops should be less than 1 cycle

- **Hold time**: Delay between 2 flops must exceed a threshold







## How to Perform STA?

- Path-based STA (PBA)
  - Test one path at one time
  - Provide detailed critical path information
  - Number of paths grows exponentially with circuit size
  - Fast path search algorithms are desired

- Graph-based STA (a.k.a. block-based STA) (GBA)
  - Propagate only the worst-case timing information
  - Most efficient, fastest and lowest memory
  - Hard to retrieve other timing-critical paths
  - Overly pessimistic due to propagating worst slew/delay

- Tradeoffs for all timing analysis tasks: Accuracy vs. Efficiency (runtime & memory)

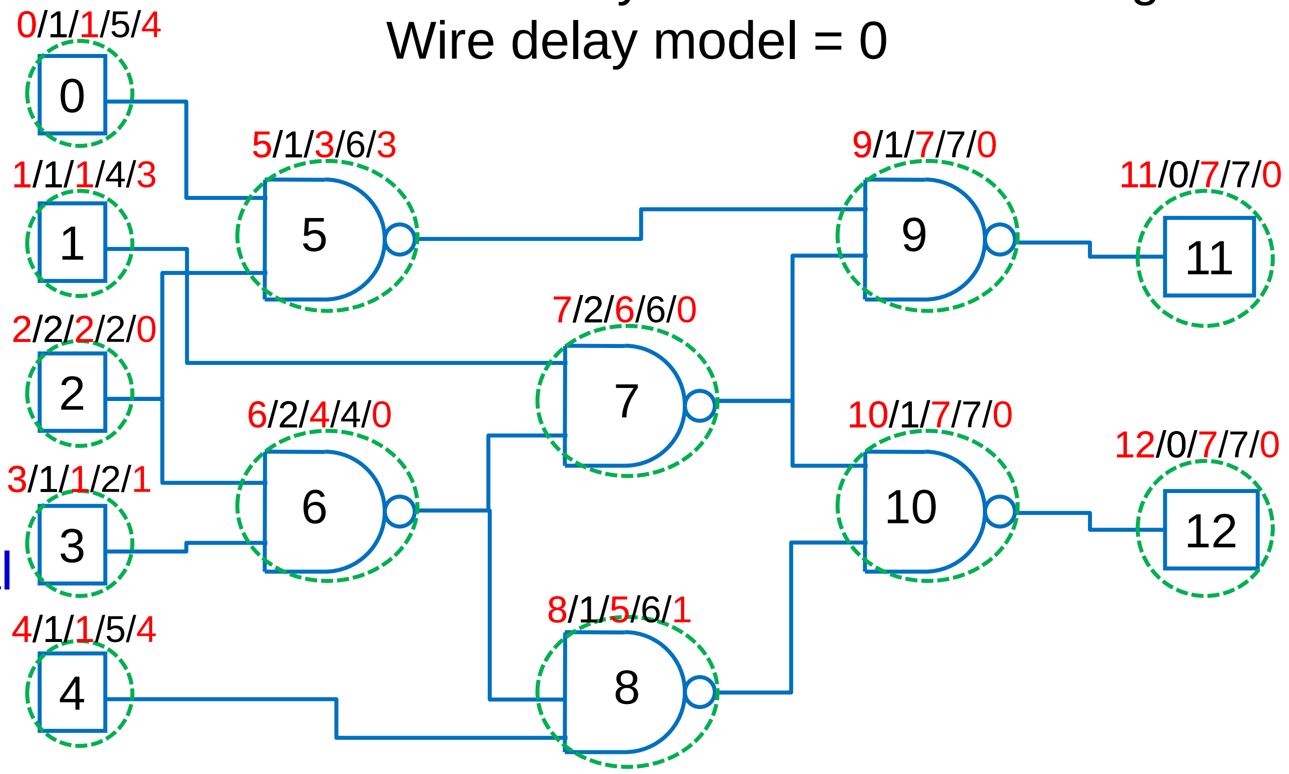
# The Simplest Timing Analysis Engine

***Setup time***

- Graph-based: Topological ordering + longest/shortest path
  - Cycles in combinational are not allowed
- Timing graph:
  - Vertex: gate/IO
  - Edge: wire
- Arrival time
  - Forward
- Required time
  - Backward
- Slack
  - Required-arrival

Gate delay model = # of fanout gates

Wire delay model = 0



Order/delay/arrival/required/slack

# Other STA Ingredients

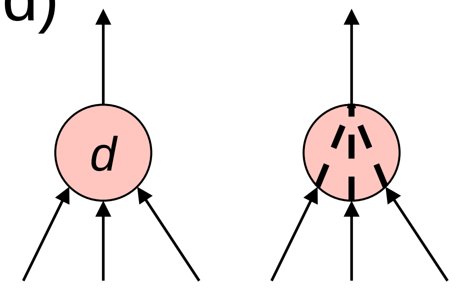
- Delay models
  - Gate/cell delay
  - Wire/interconnect
- Environment constraints
  - Operating conditions
  - Wire load model
  - Design rule constraints
- Timing constraints
  - Input
  - Output
  - Clock waveform
- Timing exceptions
  - Multicycle paths
  - False paths

.lib
Liberty
(cell timing library)

.sdc
Synopsys Design Constraint
(timing constraints)

# Gate Delay Models

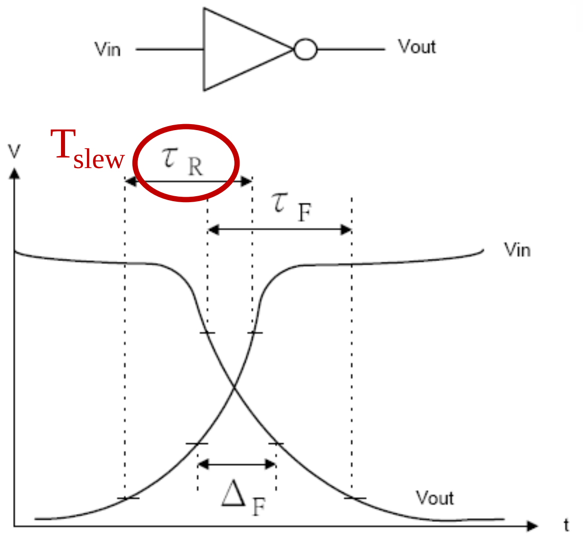
- Constant delay model (popular for front-end)
  - Constant gate delay, or pin-to-pin gate delay
  - Not accurate
- Fanout delay model (popular for front-end)
  - Gate delay considering fanout load (#fanouts)
  - Slightly more accurate
- Linear delay model (abstract model used in academia)
  - More accurate than fanout delay model

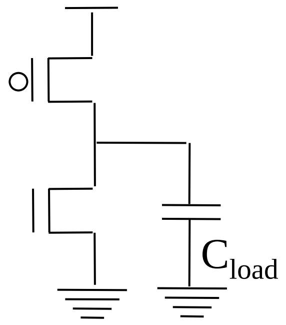


- Library delay model (industry)
  - Tabular delay data given in the cell library (SPICE)
    - Pin-to-pin
    - Determine delay from input slew and output load
    - Table look-up + interpolation/extrapolation
  - Accurate

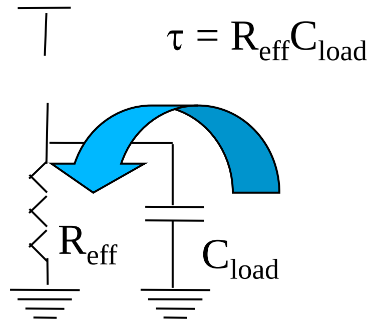
# Gate Delay

- The delay of a gate depends on:
- 1. Input slew
  - Slew = transition time
  - Slower transistor switching $\Rightarrow$ longer delay and longer output slew
- 2. Output load
  - Capacitive loading $\propto$ charge needed to swing the output voltage
  - Due to interconnect and logic fanout
- Propagation delay: 50%-50%



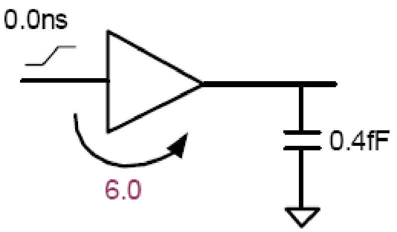


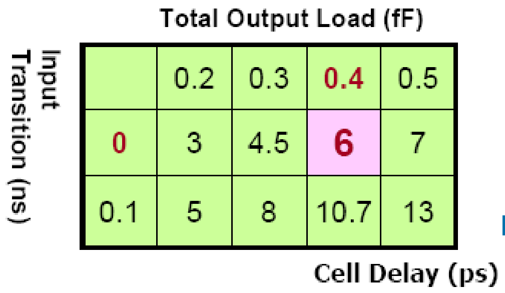
An inverter



*e.g. output* 1→0

# Timing Library





Total Output Load (fF)

Input Transition (ns)

| | 0.2 | 0.3 | 0.4 | 0.5 |
|---|---|---|---|---|
| 0 | 3 | 4.5 | 6 | 7 |
| 0.1 | 5 | 8 | 10.7 | 13 |

Cell Delay (ps)

- Timing library contains all relevant information about each standard cell
  - E.g., pin direction, clock, pin capacitance, etc.

- Delay (fastest, slowest, and often typical) and output slew are encoded for each input-to-output path and each pair of transition directions (rise/fall)

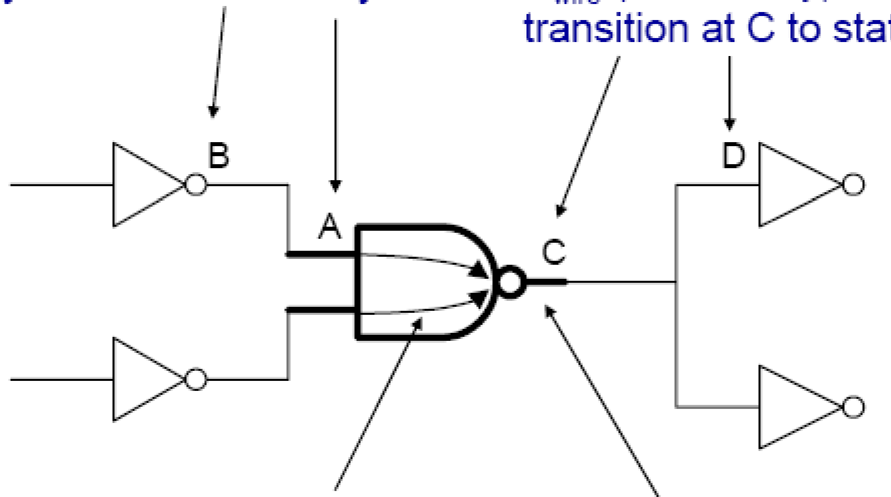
- Values are typically represented as 2D look-up tables (of input slew and output load) based on SPICE simulation
  - (Linear) interpolation/extrapolation is used

# Linear Delay Model (1/2)

$$\boxed{\text{Delay} = \overbrace{\text{Dslope} + \text{Dintrinsic} + \text{Dtransition}}^{\text{Gate delay}} + \overset{\text{Wire delay}}{\text{Dwire}}}$$

$D_{slope}$ (Slope delay): delay at input A caused by the transition delay at B

$D_{wire}$ (Wire delay): time from state transition at C to state transition at D



$D_{intrinsic}$ (Intrinsic delay): incurred from cell input to cell output

$D_{transition}$ (Transition delay): output pin loading, output pin drive

# Linear Delay Model (2/2)

- Delay = $D_{slope} + D_{intrinsic} + D_{transition} + D_{wire}$
- Slope delay
  - The delay due to a slow logic transition at the input pin
  - $D_{slope} = D_{Tprevious} * S$ ($D_{Tprevious}$: previous-stage transition; S: slope sensitivity)
- Intrinsic delay
  - The built-in delay, fixed
- Transition delay
  - Output resistance times load
  - $D_{transition} = R_{drive} * (C_{pin} + C_{wire})$
- Wire delay
  - The time to propagate a logic transition through an interconnect network
  - $D_{wire} = R_{wire} * \boxed{(C_{pin} + C_{wire})}$

Downstream capacitance depends on interconnect model

## Wire/Interconnect Delay

- Wire load model (front-end)
  - Unit fanout delay model
    - Incorporate an additional delay for each fanout
  - RC model (best-case, worst-case, balanced-case)
    - Calculate delay according to physical information (distance, loading, etc.)

- Wire model (back-end)
  - RC network extracted from routing
    - Global routing: Steiner tree
    - Detailed routing: RC extraction
  - SPEF (Standard Parasitic Exchange Format, .spef) (Industry!)

## Wire/Interconnect Delay

- Wire load model (front-end)
  - Unit fanout delay model
    - Incorporate an additional delay for each fanout
  - RC model (best-case, worst-case, balanced-case)
    - Calculate delay according to physical information (distance, loading, etc.)

- Wire model (back-end)
  - RC network extracted from routing
    - Global routing: Steiner tree
    - Detailed routing: RC extraction
  - SPEF (Standard Parasitic Exchange Format, .spef) (Industry!)

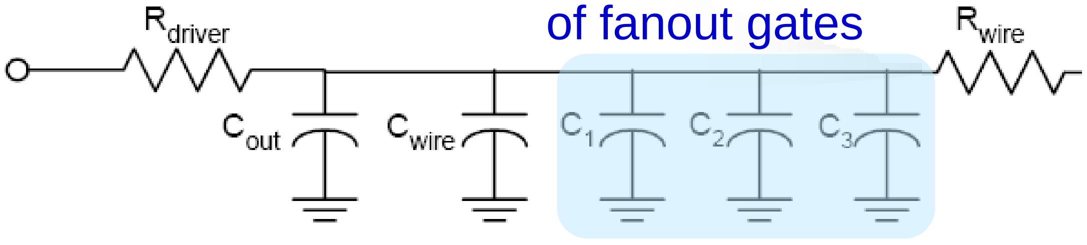
# Front-End RC Models

Downstream capacitance depends on RC network

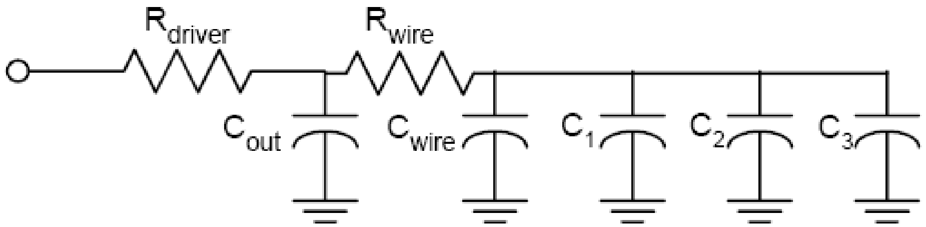
- $D_{wire} = R_{wire} * \boxed{(C_{pin} + C_{wire})}$

Input capacitance of fanout gates

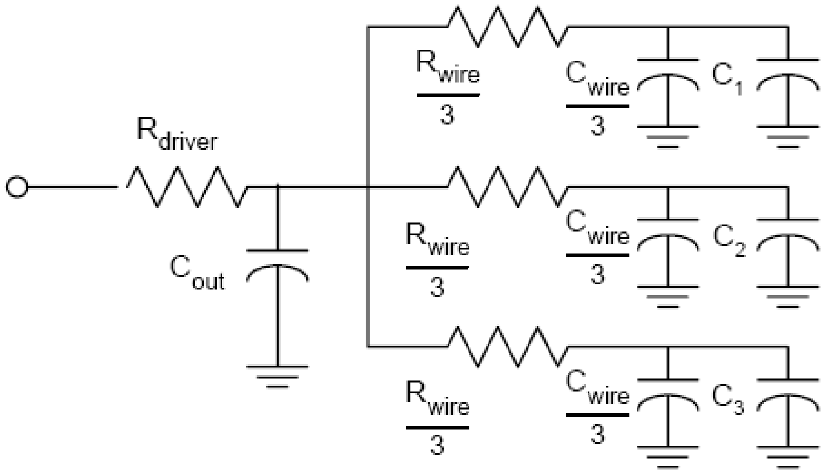
- Best-case RC tree
  - Wire delay = 0



- Worst-case RC tree



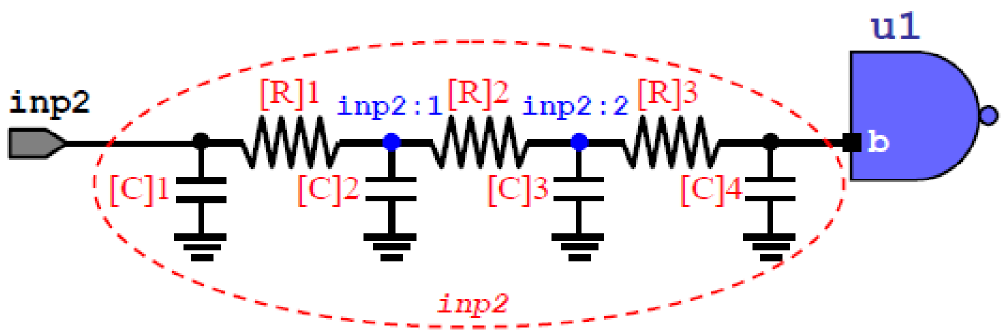
- Balanced-case RC tree



## Back-End Wire Model

- Extract an RC network from routing
- SPEF (Standard Parasitic Exchange Format) (Industry!)

```
*D_NET inp2 2.0
*CONN
*P inp2 I
*I u1:b I
*CAP
1 inp2 0.2
2 inp2:1 0.5
3 inp2:2 0.4
4 u1:b 0.9
*RES
1 inp2 inp2:1 1.4
2 inp2:1 inp2:2 1.5
3 inp2:2 u1:b 1.6
*END
```



## Wire Delay Calculation

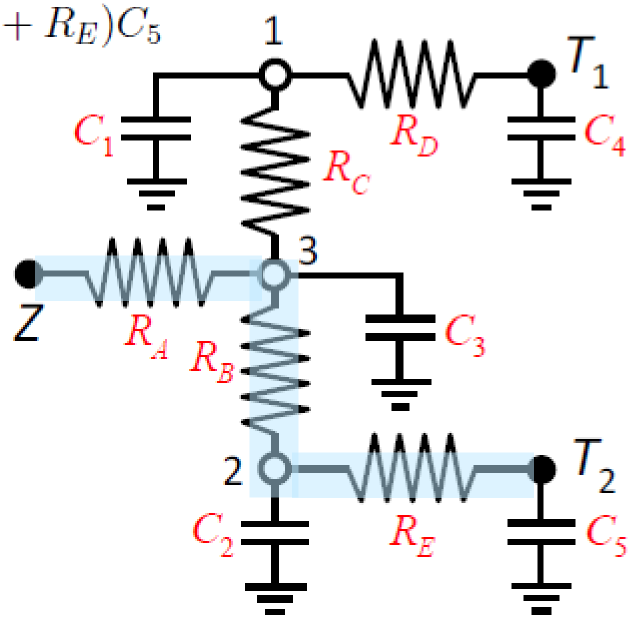
- Elmore delay model
- Delay

$$\begin{aligned} d_{T_2} &= R_A C_1 + R_A C_3 + R_A C_4 + (R_A + R_B) C_2 + (R_A + R_B + R_E) C_5 \\ &= R_A (C_1 + C_3 + C_4) + (R_A + R_B) C_2 + (R_A + R_B + R_E) C_5 \end{aligned}$$

- Slew

$$S_{oT} \approx \sqrt{{s_i}^2 + 2\beta_T - {d_T}^2}$$

  - $s_i$: input slew
  - $\beta_T$: second moment of the input response
  - $d_T$: Elmore delay



W.C. Elmore, “The Transient Analysis of Damped Linear Networks with Particular Regard to Wideband Amplifiers,” J. Applied Physics, vol. 19(1), 1948.

C. V. Kashyap, C. J. Alpert, F. Liu and A. Devgan,“Closed-form Expressions for Extending Step Delay and Slew Metrics to Ramp Inputs for RC Trees”, IEEE Transactions on Computer-aided Design of Integrated Circuits and Systems, 23(4)(2004), pp. 509-516.

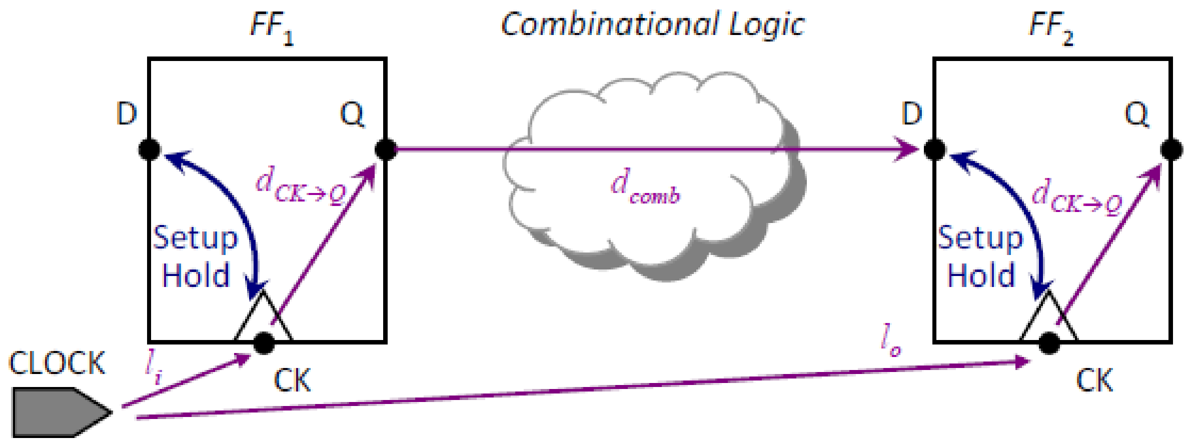
# Timing Propagation (Setup Time)

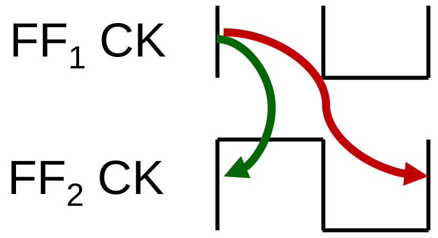
- Arrival time can be computed in the topological order from inputs to outputs
  - Arrival time of a primary input is given
  - When a node is visited, its output arrival time is:
    the max of its fanin arrival times + its own gate delay

- Required time can be computed in the reverse topological order from outputs to inputs
  - Required time of a primary output is given or specified due to clock cycle constraint
  - When a node is visited, its input required time is:
    the min of its fanout required times – its own gate delay

- Slack = required time – arrival time
  - Timing flexibility margin (positive: good; negative: bad)

# Setup and Hold Constraints





**Setup**

$$at_D^{late} = l_i^{late} + d_{CK\rightarrow Q} + d_{comb}^{late}$$

$$rat_{setup} = rat_D^{late} = \boxed{T} + l_o^{early} - t_{setup}$$

$$slack^{late} = rat^{late} - at^{late}$$

**Hold**

$$at_D^{early} = l_i^{early} + d_{CK\rightarrow Q} + d_{comb}^{early}$$

$$rat_{hold} = rat_D^{early} = l_o^{late} + t_{hold}$$

$$slack^{early} = at^{early} - rat^{early}$$

## Environment Constraints

- Operating conditions
- Wire load model
- Design rule constraints

## Operating Conditions

- Cases: best, typical, worst

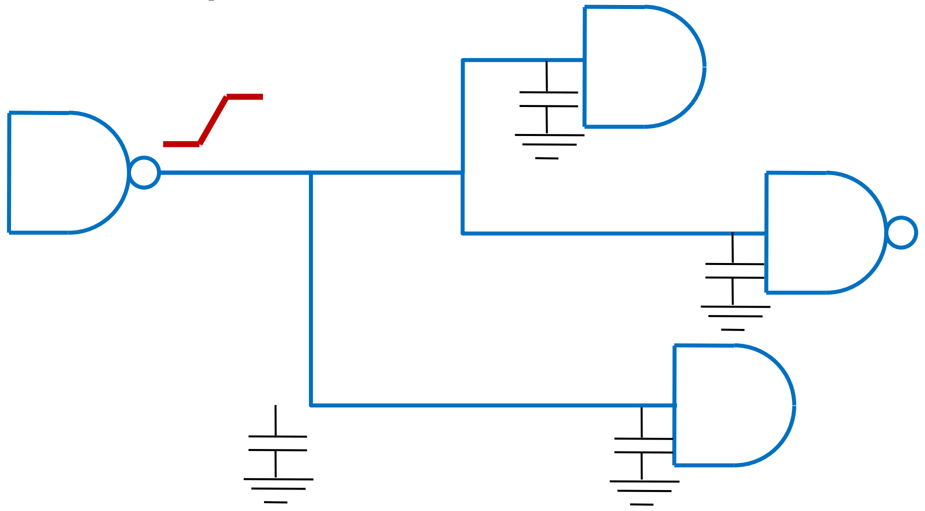
- Process variation
  - Treated as a straight percentage variation in any performance calculation
- Voltage variation
  - Speed increased with higher voltage
- Temperature variation
  - Delay increased with higher temperature

- Implemented by setting different derating values

## Design Rule Constraints

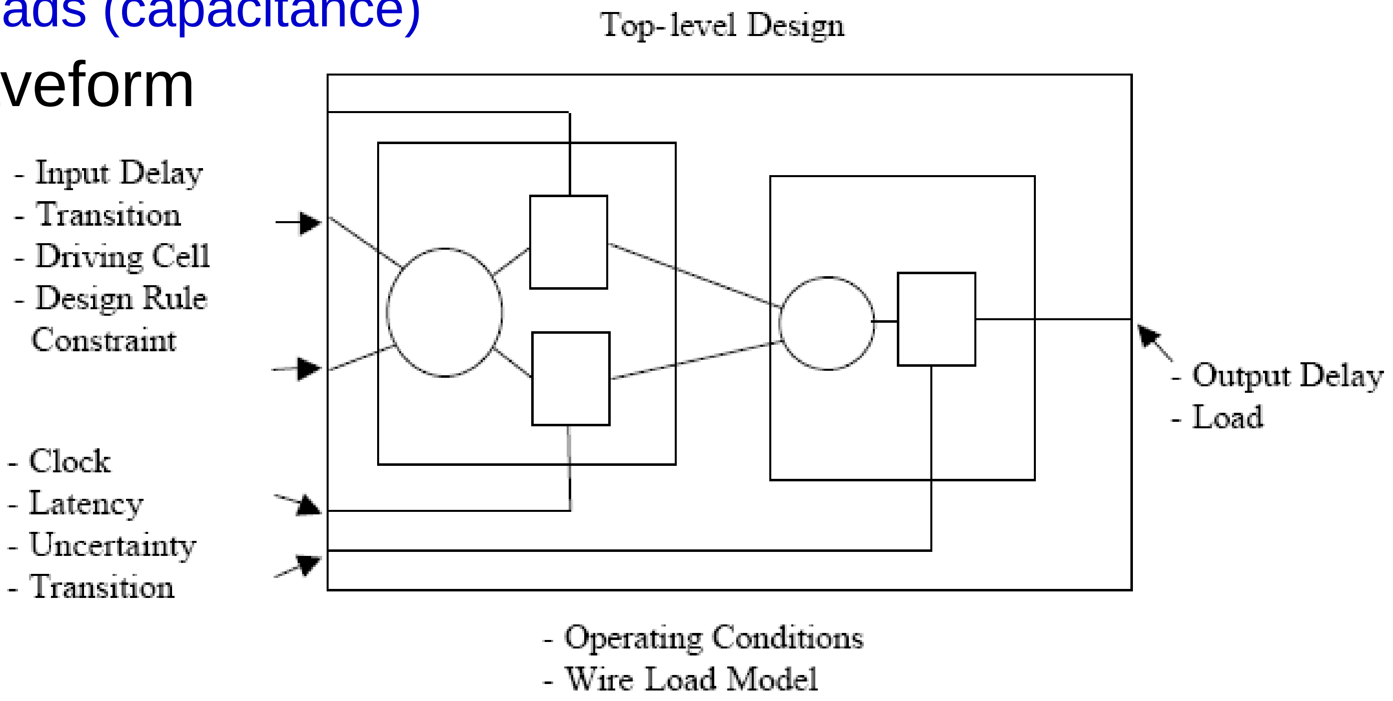
- ≠ design rules for layout editing or mask generation
  - Width, spacing, enclosure

- Max transition time: max transition time of an output pin
- Max capacitance: max capacitance of an output pin
- Max fanout: max number of fanout allowed for an output pin (soft constraint)



## Timing Constraints

- Inputs:
  - Arrival times (input delays)
  - Input drives (resistance)
- Output
  - Required times (output delays)
  - Output loads (capacitance)
- Clock waveform



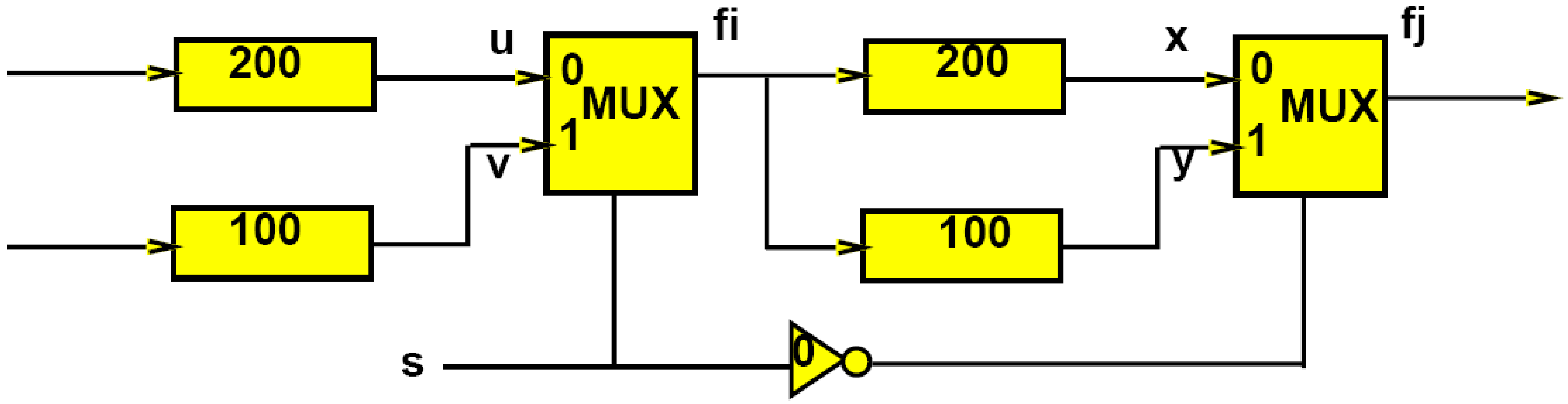
## Timing Exceptions

- Override default single-cycle timing behavior
  - Multicycle path
    - Override the default setup & hold relations
  - False path
    - Prune timing paths which are never be exercised functionally

## False Paths

- STA maybe inaccurate because of false paths
- Designers specify false paths
- Example:

Static analysis is fast but leads to **false paths**
Path of length 400 is never “exercised”

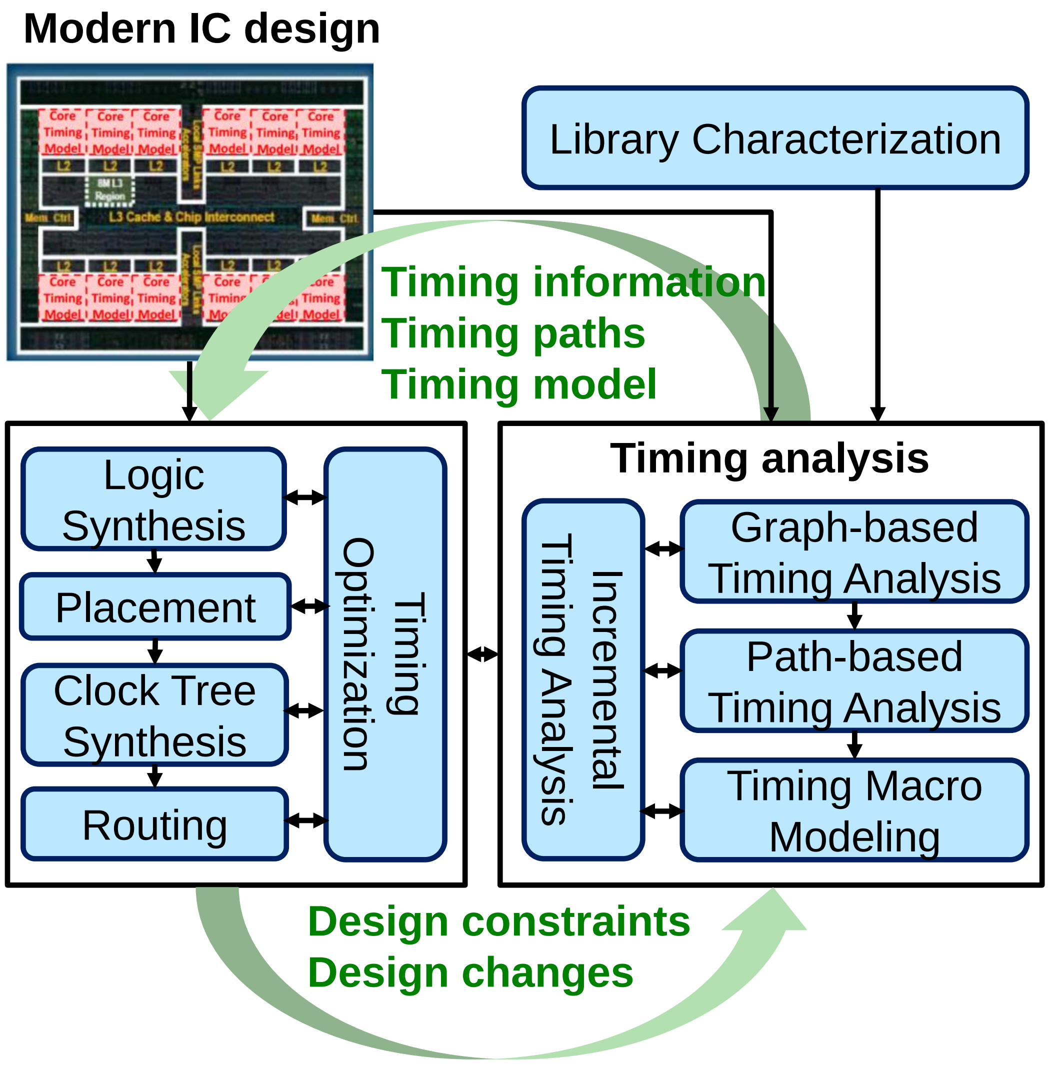


– Other examples: carry look ahead adder vs. ripple adder

## Welcome to the real world!

# A Generic Modern Timer

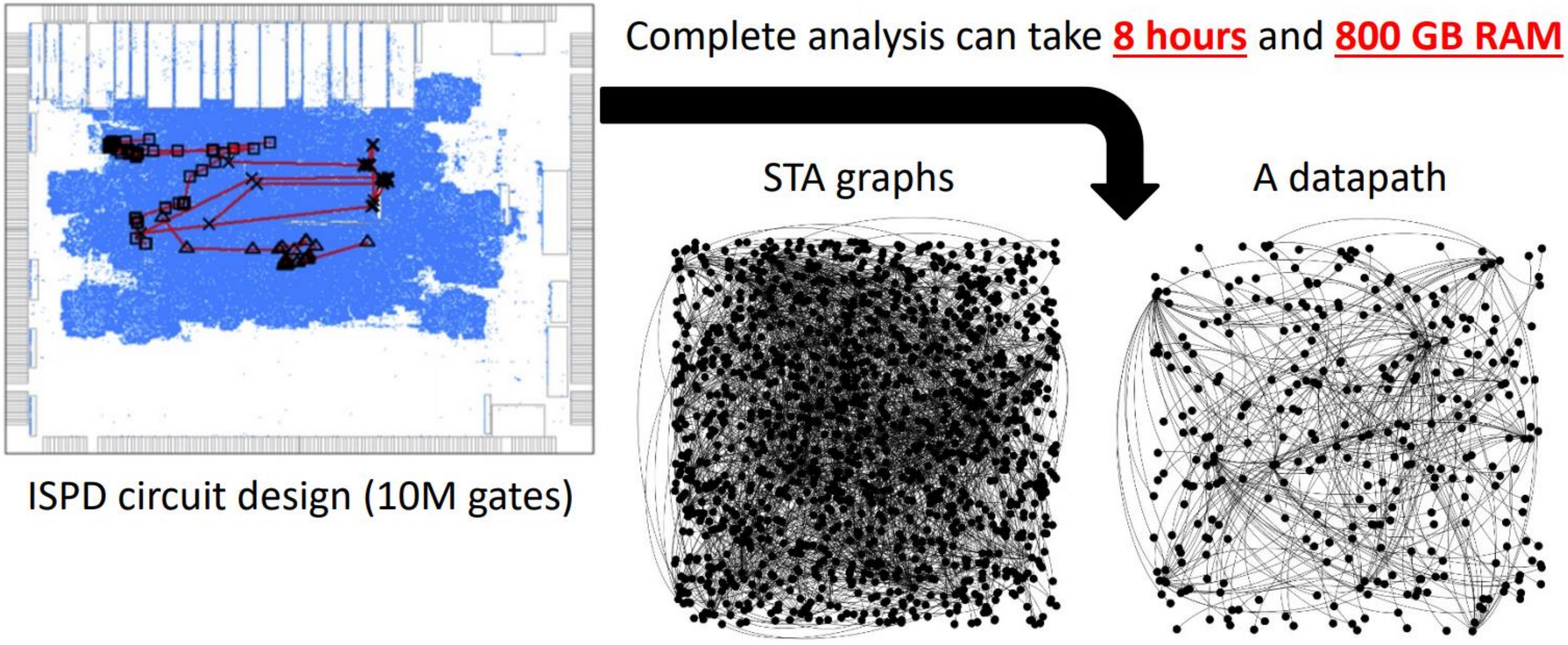
- Hybrid!
  - Require a good bounding scheme
- Graph-based analysis
  - How to update timing fast when design changes occur?
- Path-based analysis
  - How to quickly provide timing paths under complex design constraints?
- Timing macro modeling
  - How to extract accurate and compact timing model fast?



# Emerging Challenge #1: Analysis Efficiency and Scalability

- Timing graph becomes extremely large and irregular

- Solution #1: parallel computing by CPU, GPU, CPU-GPU
- Solution #2: machine learning


Complete analysis can take **8 hours** and **800 GB RAM**
STA graphs
A datapath
ISPD circuit design (10M gates)

STA graphs are extremely large and irregular

Source: T.-W. Huang

## Emerging Challenge #2: Advanced Process Effect

- **Local layout effect:** The timing and power of a gate is strongly affected by its neighbors in the placement
  - Stress LDEs (diffusion dimension, oxide spacing, gate pitch), gate cut stress LDE, metal boundary effect, density gradient effect
- **Corner explosion:** Considering design robustness under variations, verify over the possible range of PVT before signoff
  - Multiple chiplets in 2.5D/3D ICs
  - Voltage variations under dynamic IR drop
  - Temperature variations
- **Aging effect:** Workload-dependent; the aging rates of different paths are non-uniform
  - Have a huge impact on automotive and IoT products with long product life and exposure to harsh temperatures
- **Design technology co-optimization:** Improve performance, power efficiency, transistor density, and cost (PPAC)

# Thank You!