臺大: EDA頂尖會議領頭15年

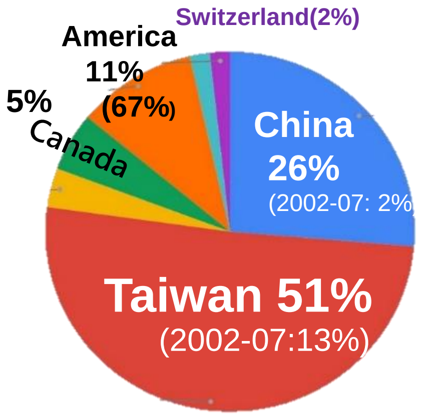
- 過去16年臺大在EDA頂尖會議DAC+ICCAD論文發表數15年世界第一
- ACM CADathlon競賽冠軍數: 臺大NTU: 9 ; UC-Berkeley: 3; MIT: 1
- EDA 研發競賽: 12 次冠軍，世界第一

EE Times

C. Johnson: **"The Best and Brightest Worldwide"** (4/6/2015): "The best engineering minds on the planet compete each year in the ACM's ISPD design contest, which was won this year by the **National Taiwan University**."

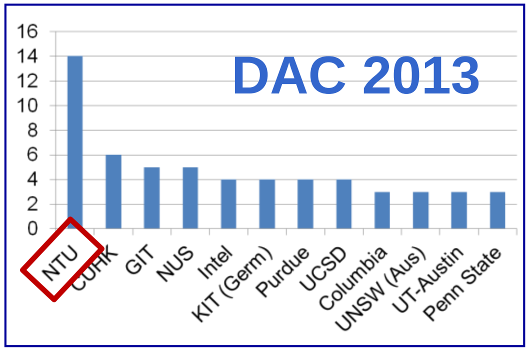
EE Times

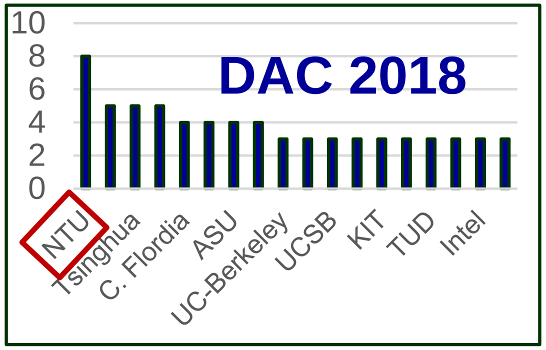
Taiwan: Microelectronics expertise widens' (5/15/2013) : "**Taiwan's success so far has been in large part due to electronic design automation (EDA) expertise**, where it **has only been outperformed by the U.S.** for the last five years."

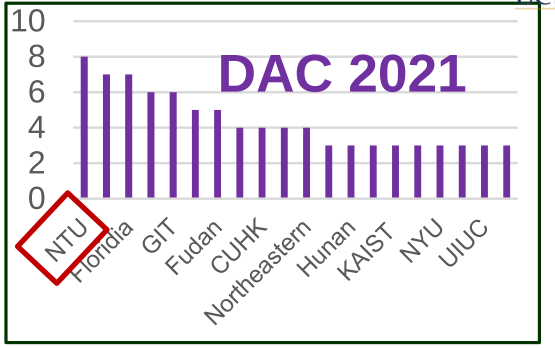


2014-2018 International EDA Contest Statistics

Source







EDA Research Areas

Front-end

江介宏教授 郭斯彥教授 郭大維教授

System-level verification & simulation

H/S co-design & multi-core

楊佳玲教授 黃鐘揚教授

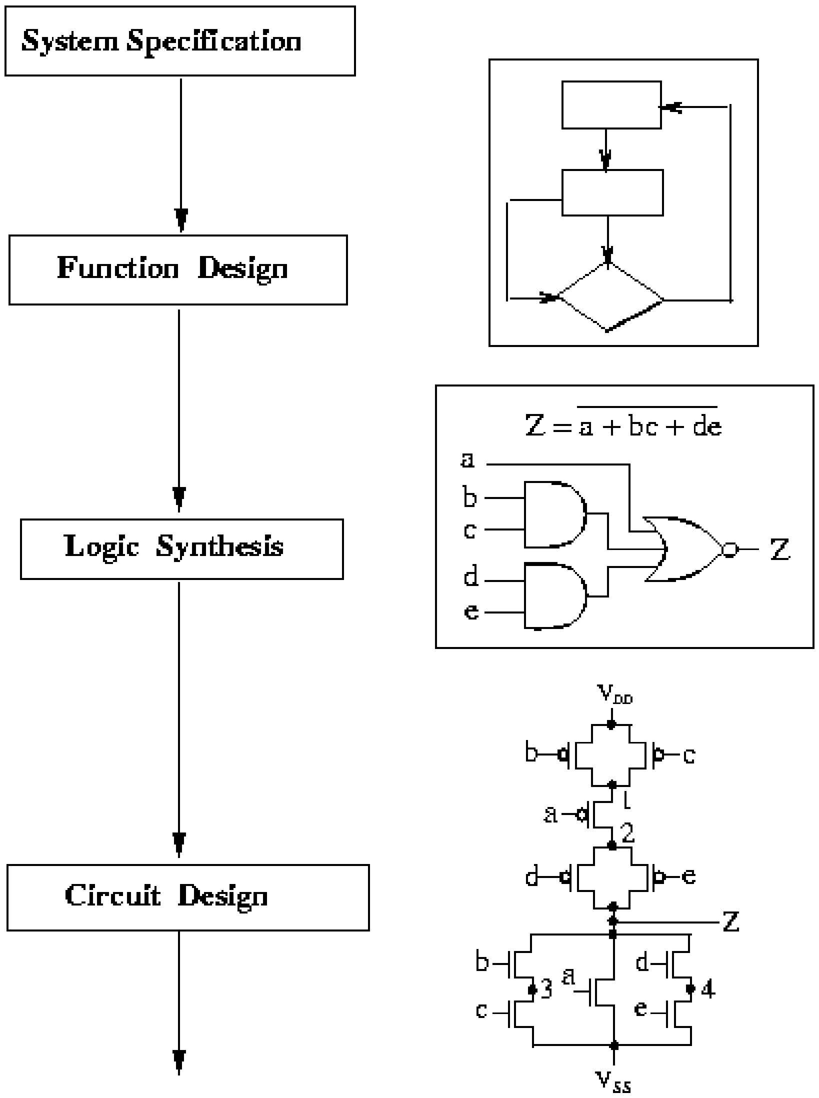
江介宏教授 黃鐘揚教授 江蕙如教授

鄭振牟教授 黃筱鈞教授

Formal verification & logic synthesis

Device/Circuit Simulation & modeling

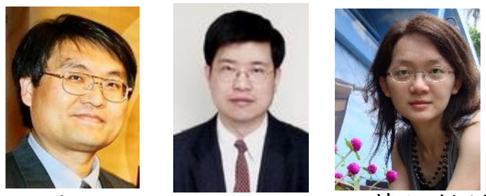
陳中平教授 蔡坤諭教授 李建模教授



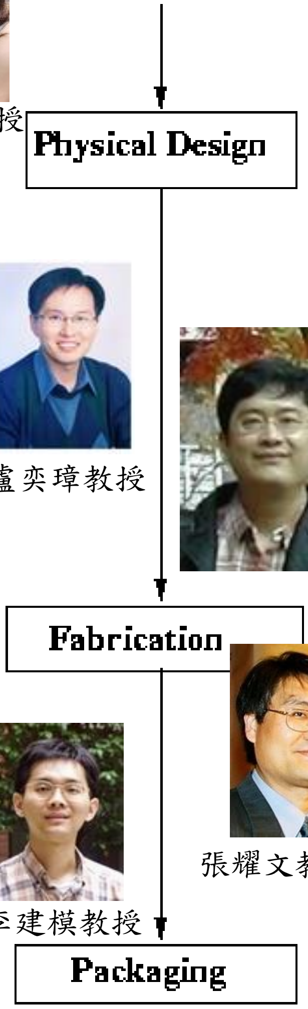
EDA Research Areas

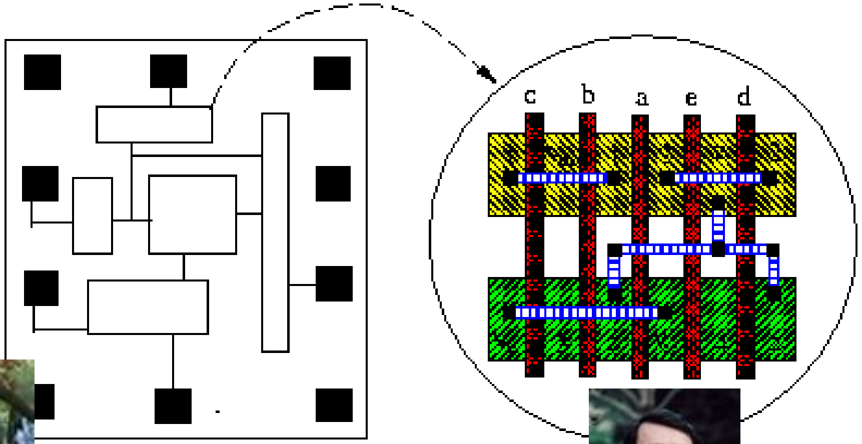
Back-end

Physical design



張耀文教授 陳中平教授 江蕙如教授

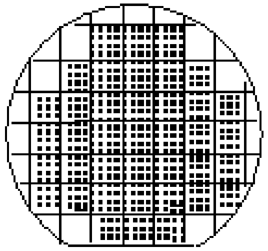




張耀文教授 陳中平教授 蔡坤諭教授 江蕙如教授 盧奕璋教授

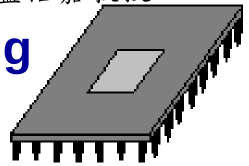
Design for Manufacturability

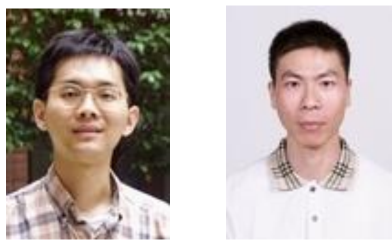
郭斯彥教授



張耀文教授 盧奕璋教授 盧信嘉教授

Packaging





李建模教授 黃俊郎教授

Testing

江介宏教授 黃俊郎教授 陳中平教授 江蕙如教授李建模教授

Emerging Technologies

Synthesis & Verification (Jie-Hong R. Jiang)

- Research highlights
  - Scalable logic synthesis (**ICCAD’07, DAC’10 best paper nominees**)
    - Adopted in UC Berkeley ABC logic synthesis & verification system
  - Hardware and software verification (**DAC’16 best paper nominee**)
  - Decision procedures and proof systems (**citation# 342 till 2020.4**)
    - Unify QBF syntactic and semantic certification (**CAV’11 distinct paper**)
    - Develop new QBF resolution proof systems and solvers
  - Design automation for biology
    - Digital and analog bio-circuit synthesis
- International **contest winners**
  - CAD Contest @ ICCAD
  - ICCAD CADathlon
  - IWLS Programming Contest
- International joint projects and collaboration
  - UC Berkeley, CNRS, INRIA, Tomsk State Univ., Higher School of Economics, Russian Academy of Sciences, Univ. of Freiburg, Graz Univ. of Technology, Univ. of Tokyo

Physical Design (Yao-Wen Chang)

- Sole 6-time 1st-place winner:
  placement, global routing, gate sizing, clock network synthesis, timing analysis

NTUplace4 (2015) => MaxPlace (Maxeda)

C. Johnson: "The Best and Brightest Worldwide" (4/6/2015):"**The best engineering minds on the planet compete each year in the ACM's ISPD design contest, which was won this year by the National Taiwan University**."

ACM Student Research Contest @ DAC

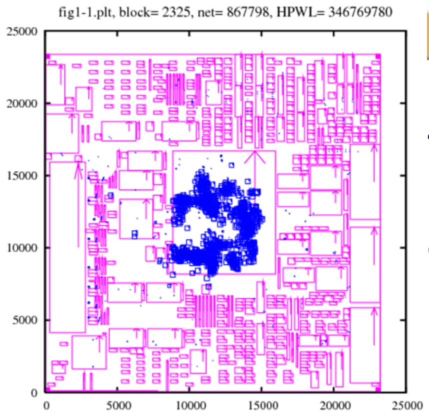
Graduate Category

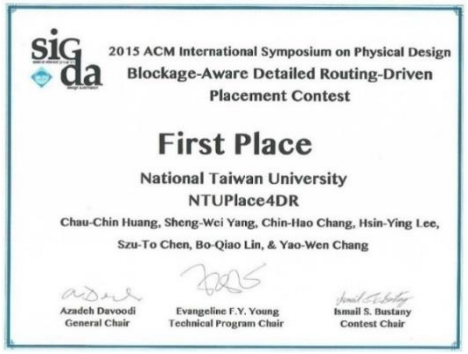
1st Place (Gold):

Student: Shao-Yun Fang, EE, NTU (Advisor: Prof. Yao-Wen Chang)

Paper Title: Lithography Optimization for Sub-22 Nanometer Technologies

Primary Metric: 1st Place Winner
Team name: NTUgr
Affiliation: National Taiwan University
49
DESIGN AUTOMATION CONFERENCE





2015 ACM International Symposium on Physical Design
Blockage-Aware Detailed Routing-Driven Placement Contest

First Place

National Taiwan University
NTUPlace4DR

Chau-Chin Huang, Sheng-Wei Yang, Chin-Hao Chang, Hsin-Ying Lee, Szu-To Chen, Bo-Qiao Lin, & Yao-Wen Chang

Azadeh Davoodi	Evangeline F.Y. Young	Ismail S. Bustany
General Chair	Technical Program Chair	Contest Chair

4/17/2008 EE Times:

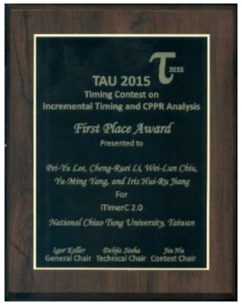
"**how the Taiwanese beat both the US and Europeans** in the ISPD Global Routing Contest…"

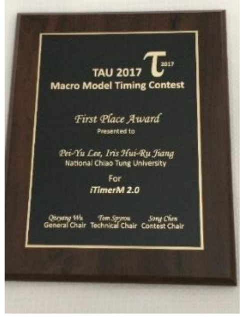
DAC 2017 Best Paper Award

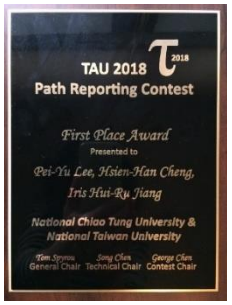
http://www2.dac.com/events/videoarchive.aspx?confid=139&filter=keynote&id=139-120--0&#video

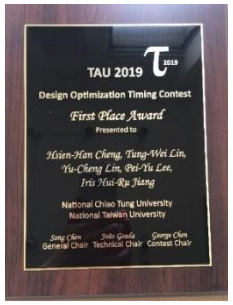
Timing & DFM (Iris Hui-Ru Jiang)

- 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) Emerging technology using EDA techniques 新興技術之應用
- 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~now)
  - 4次冠軍，冠軍紀錄保持團隊


TAU 2015 τ 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 τ 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 τ 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

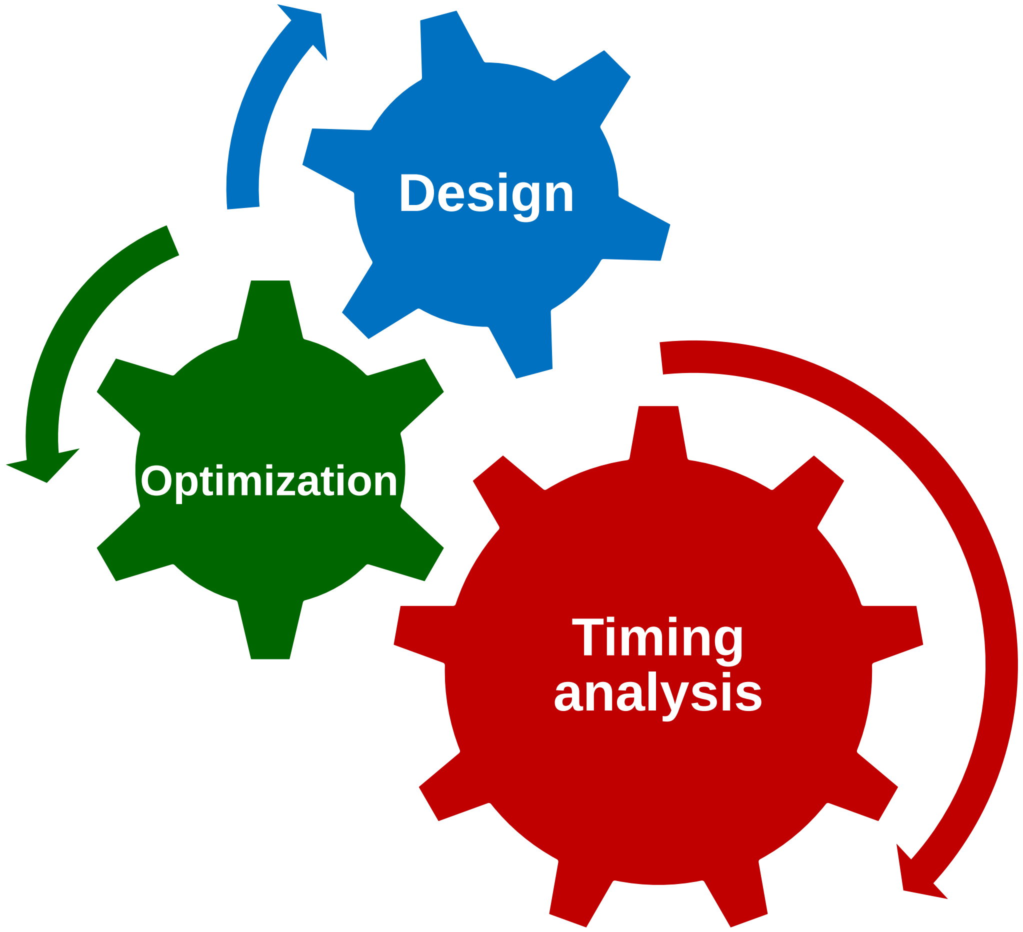

TAU 2019 τ 2019
Design Optimization Timing Contest
First Place Award
Presented to
Hsien-Han Cheng, Tung-Wei Lin,
Yu-Cheng Lin, Pei-Yu Lee,
Iris Hui-Ru Jiang
National Chiao Tung University
National Taiwan University
Song Chen General Chair · João Geada Technical Chair · George Chen Contest Chair

A Quick Journey through Timing Analysis

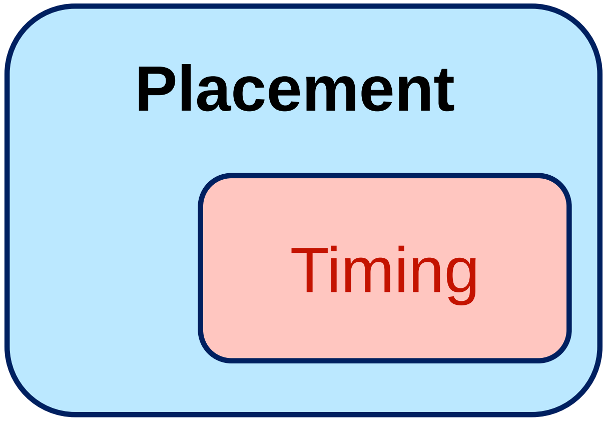
Iris Hui-Ru Jiang
Department of Electrical Engineering
Graduate Institute of Electronics Engineering
National Taiwan University

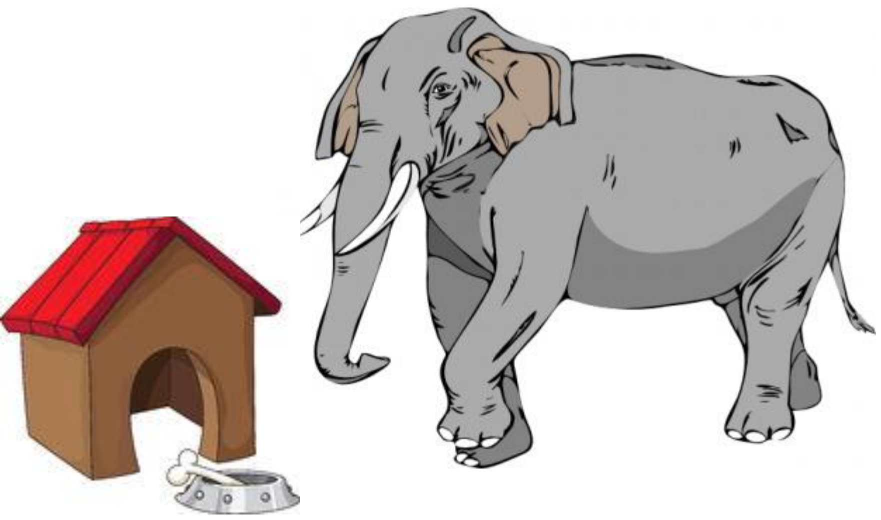
Timing is Everything…

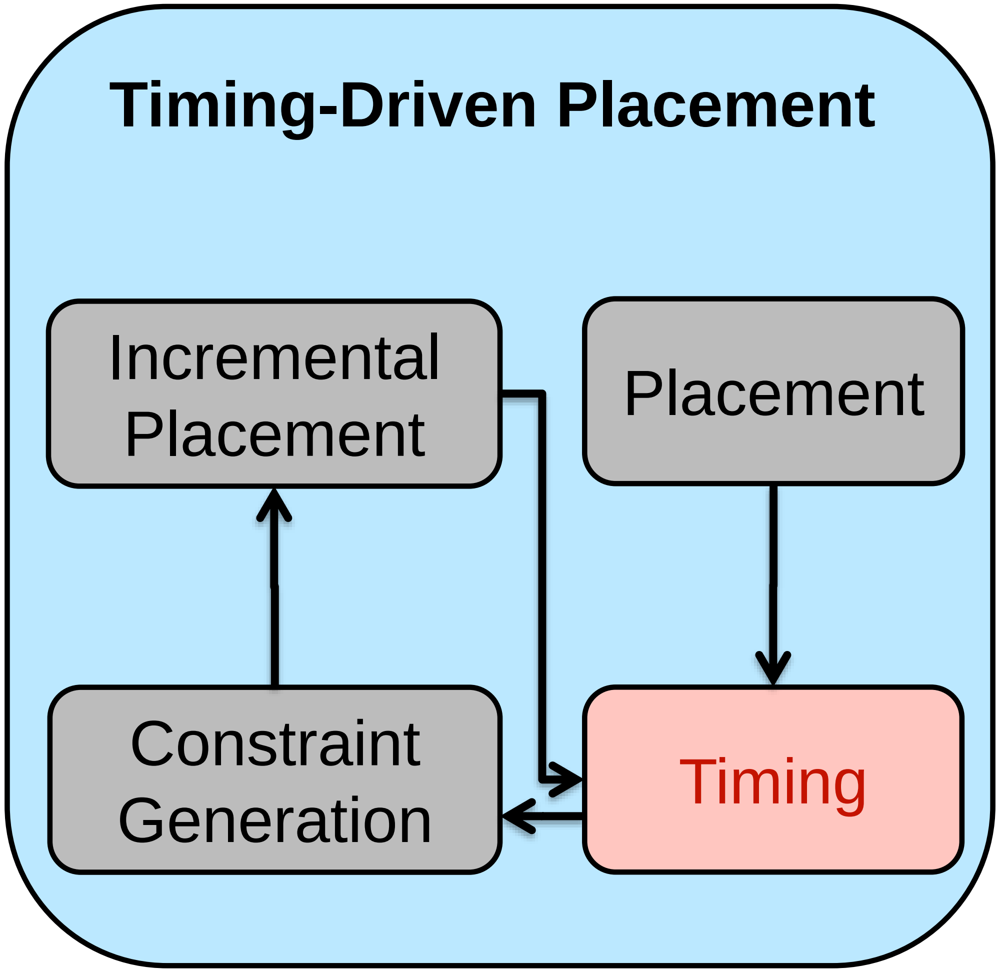
- Signals should arrive at the right place at the right time
- Timing analysis is essential in the modern IC design flow



Don’t Put Timer into Other Engines!







Courtesy of C. Alpert

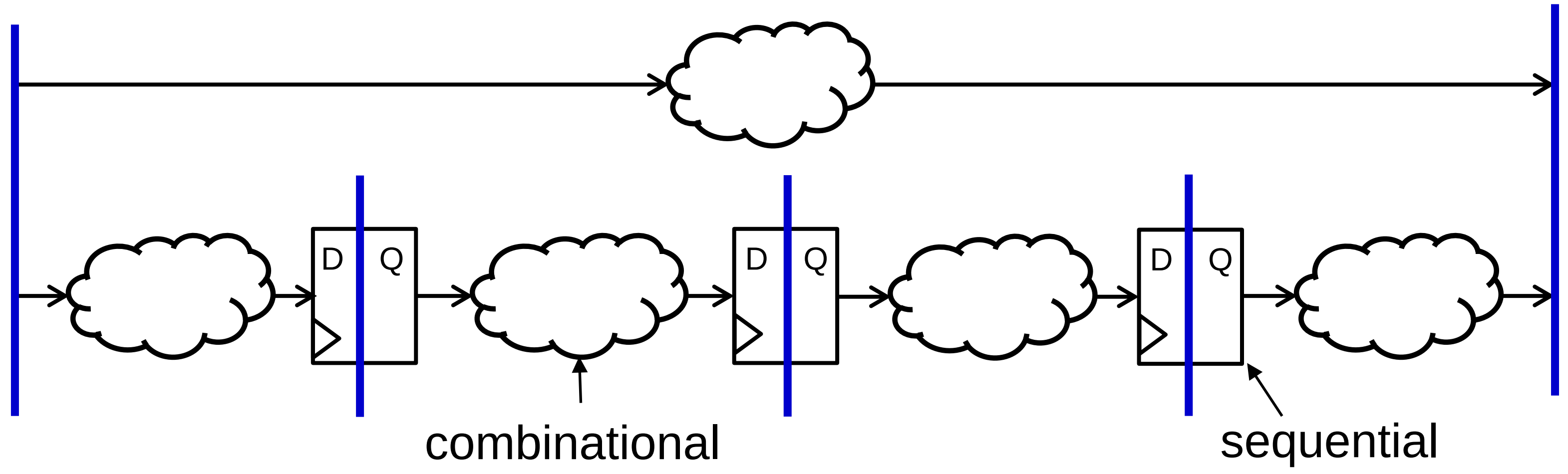
Timing Analysis

- 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

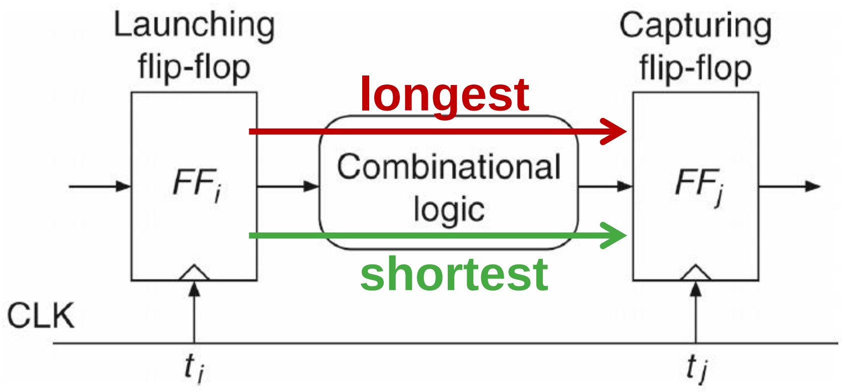
- Because of its high scalability for large scale designs, STA is widely adopted in the modern IC design flow (industry standard)

STA SOP

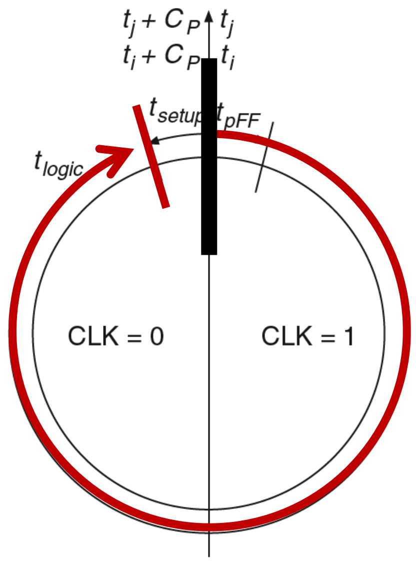
1. Divide a design into timing windows based on flip-flops/latches and I/Os
2. Construct a timing graph (directed acyclic graph) for each window
3. Calculate the circuit delay
4. Check timing constraints

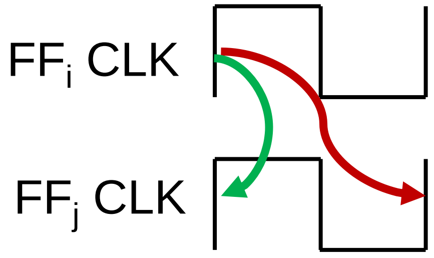


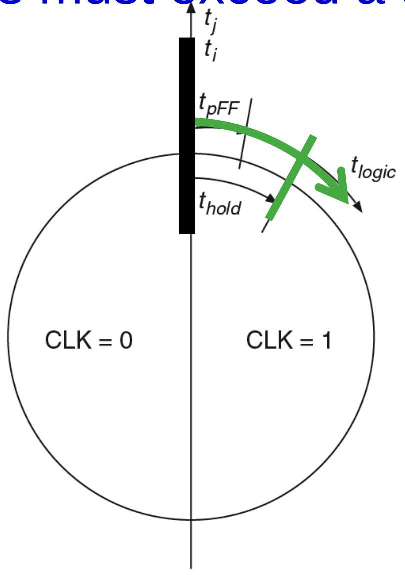
Timing Check



- **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?

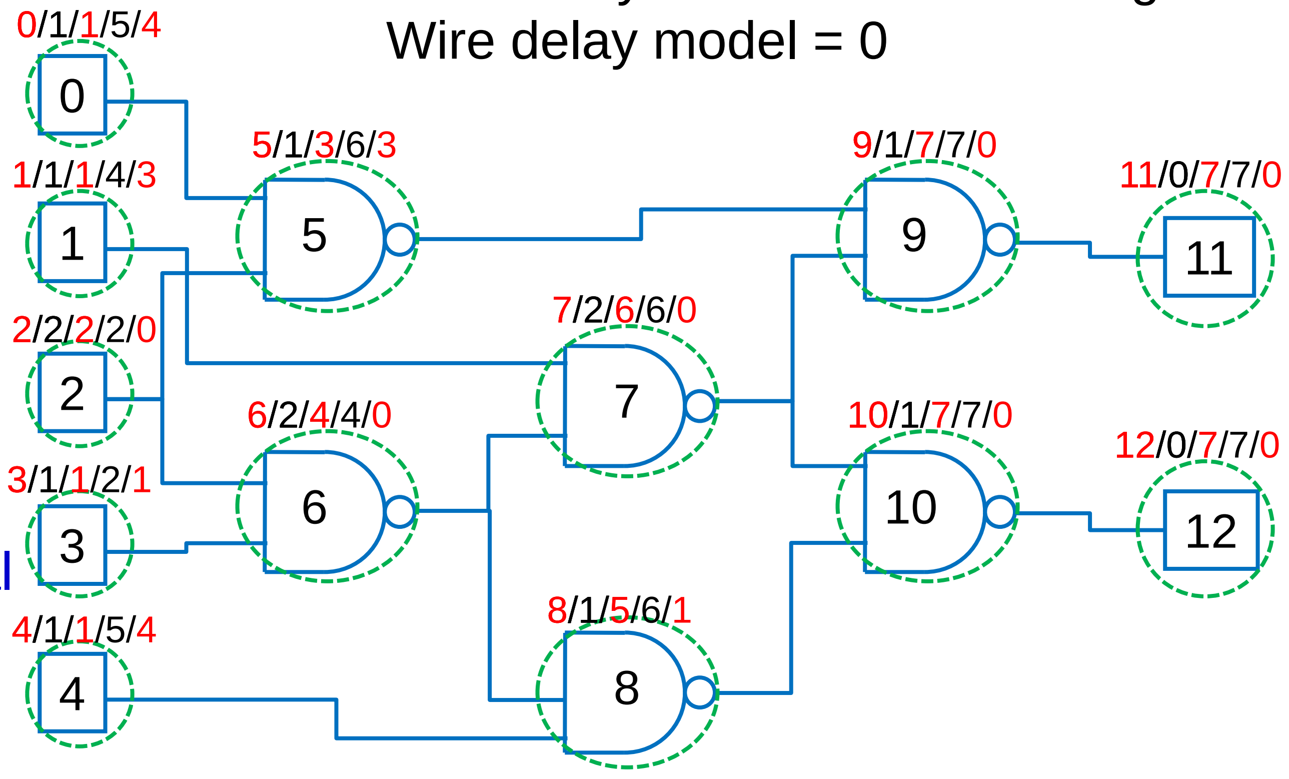
- Path-based STA (PBA)
  - Test one path at one time
  - Provide detailed critical path information
  - Number of paths grows exponentially with circuit size
- Block-based STA (a.k.a. graph-based STA, GBA)
  - Propagate only the worst-case timing information
  - Most efficient, fastest and lowest memory
  - Hard to retrieve other timing-critical paths

The Simplest Timing Analysis Engine

Setup time

- Block-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

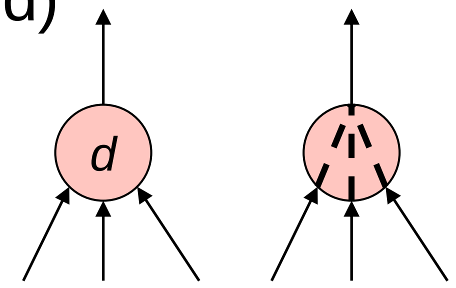
- Delay models
  - Gate/cell delay
  - Wire/interconnect
- Environment constraints
  - Operating conditions
  - Wire load model
  - Design rule constraints
- Timing constraints
  - Input
  - Output
  - Clock waveform
- Timing exceptions
  - Multicycle paths
  - False paths

.lib
Liberty
(cell timing library)

.sdc
Synopsys Design Constraint
(timing constraints)

Gate Delay Models

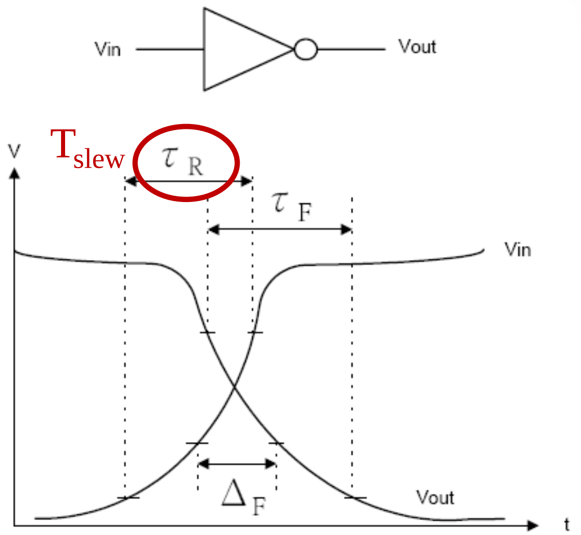
- Constant delay model (popular for front-end)
  - Constant gate delay, or pin-to-pin gate delay
  - Not accurate
- Fanout delay model (popular for front-end)
  - Gate delay considering fanout load (#fanouts)
  - Slightly more accurate
- Linear delay model (abstract model used in academia)
  - More accurate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

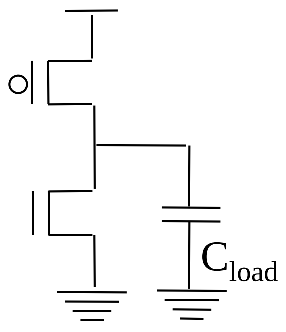


- Library delay model (industry)
  - Tabular delay data given in the cell library (SPICE)
    - Pin-to-pin
    - Determine delay from input slew and output load
    - Table look-up + interpolation/extrapolation
  - Accurate

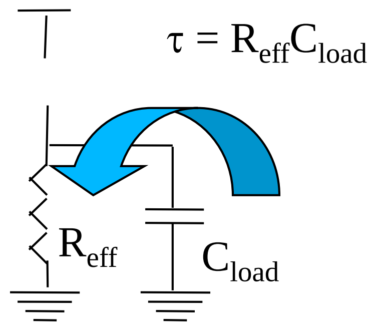
Gate Delay

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



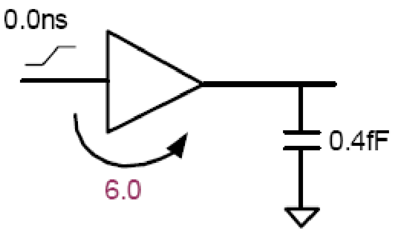


An inverter



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)

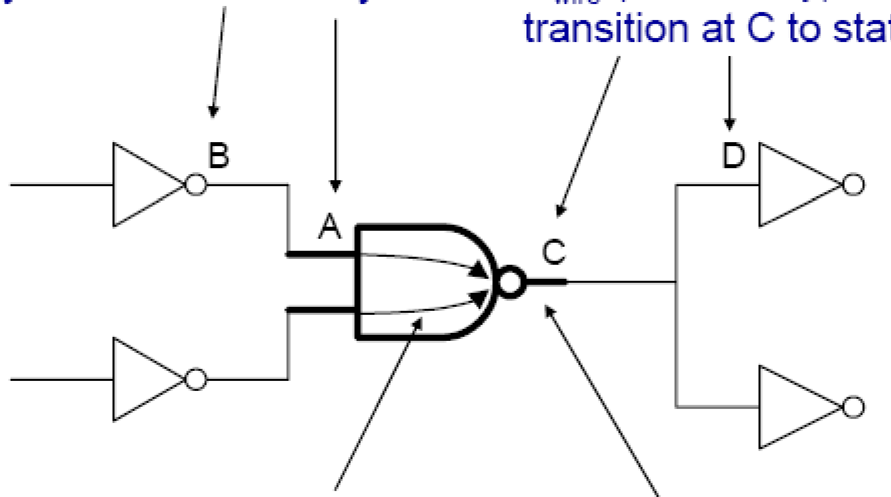
- Values are typically represented as 2D look-up tables (of input slew and output load) based on SPICE simulation
  - (Linear) interpolation/extrapolation is used

Linear Delay Model (1/2)

$$\underbrace{\text{Delay} = D_{slope} + D_{intrinsic} + D_{transition}}_{\text{Gate delay}} + \underbrace{D_{wire}}_{\text{Wire delay}}$$

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!)

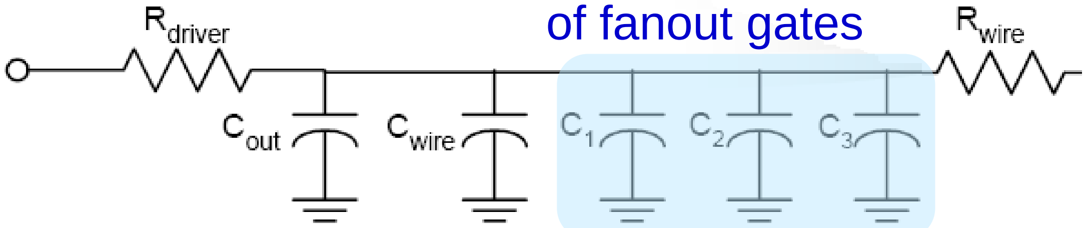
Front-End RC Models

Downstream capacitance depends on RC network

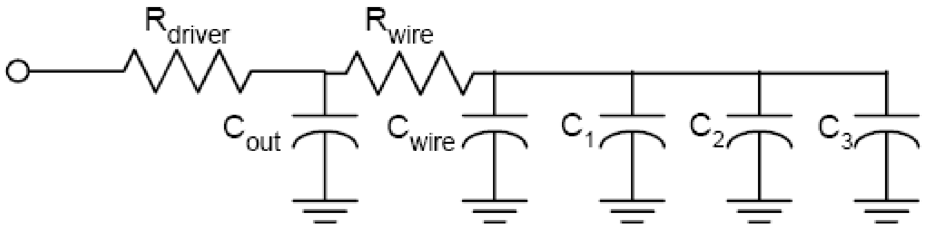
- $D_{wire} = R_{wire} * \boxed{(C_{pin} + C_{wire})}$

Input capacitance of fanout gates

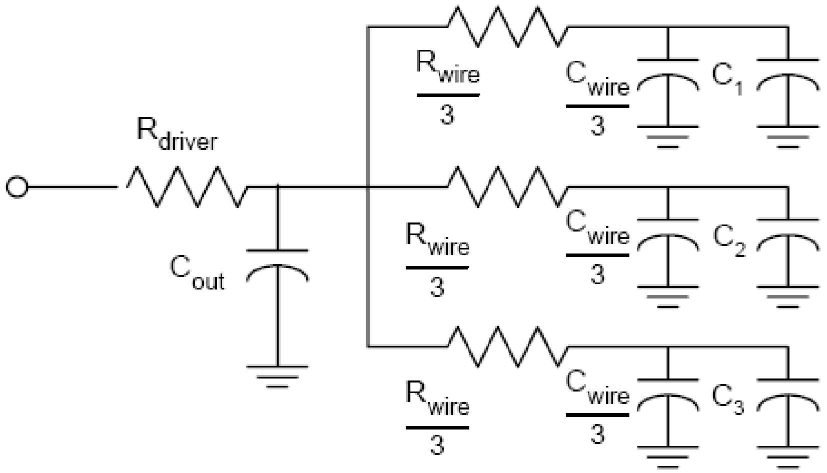
- Best-case RC tree
  - Wire delay = 0



- Worst-case RC tree



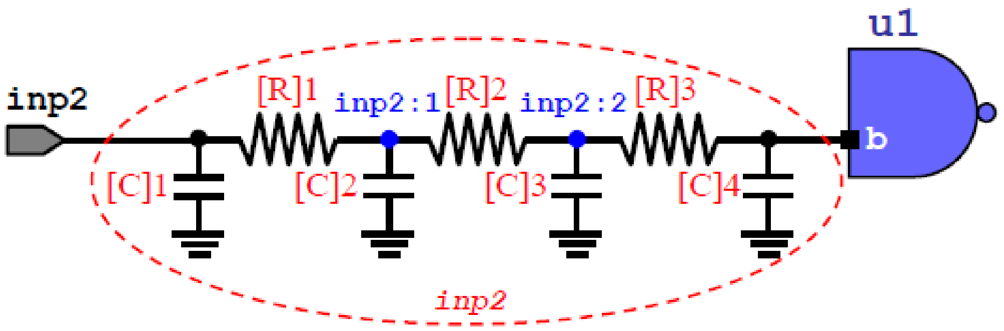
- Balanced-case RC tree



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

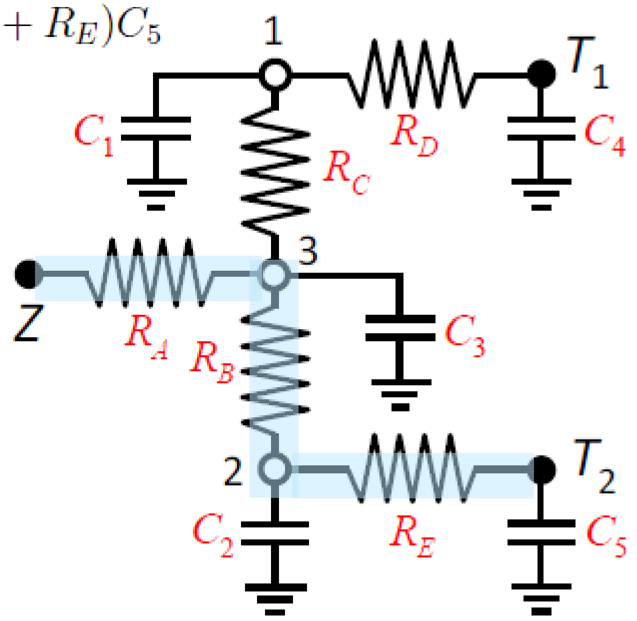
- Elmore delay model
- Delay

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

- Slew

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

  - s_i: input slew
  - β_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.

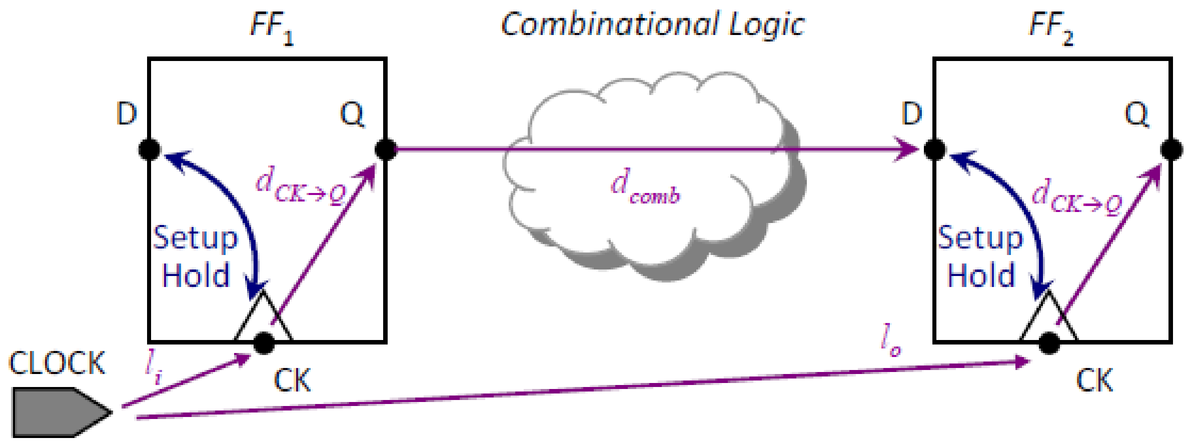
Timing Propagation (Setup Time)

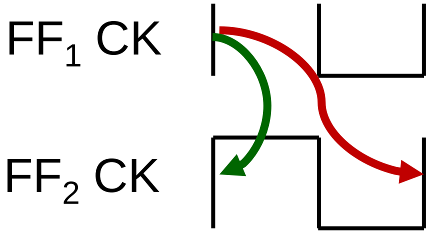
- Arrival time can be computed in the topological order from inputs to outputs
  - Arrival time of a primary input is given
  - When a node is visited, its output arrival time is:
    the max of its fanin arrival times + its own gate delay

- 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

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

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

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

Hold

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

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

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

Environment Constraints

- Operating conditions
- Wire load model
- Design rule constraints

Operating Conditions

- Cases: best, typical, worst

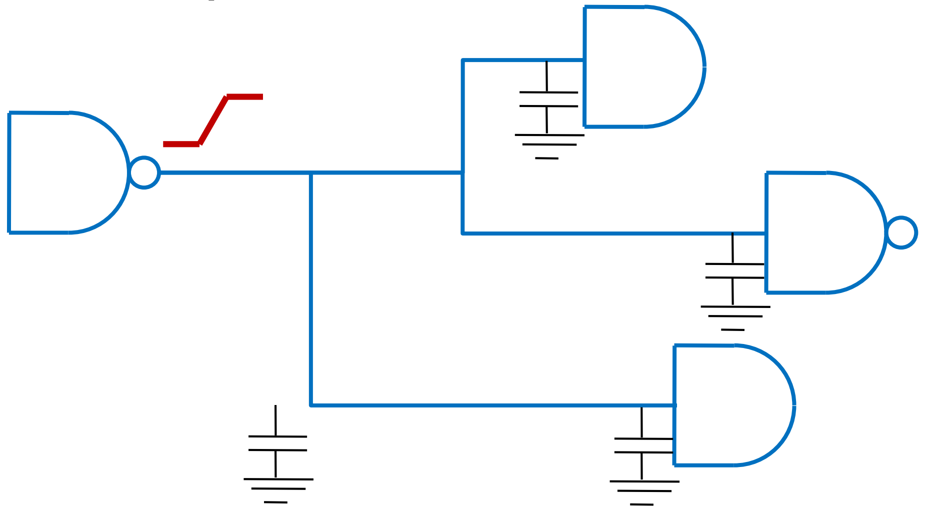
- Process variation
  - Treated as a straight percentage variation in any performance calculation
- Voltage variation
  - Speed increased with higher voltage
- Temperature variation
  - Delay increased with higher temperature

- Implemented by setting different derating values

Design Rule Constraints

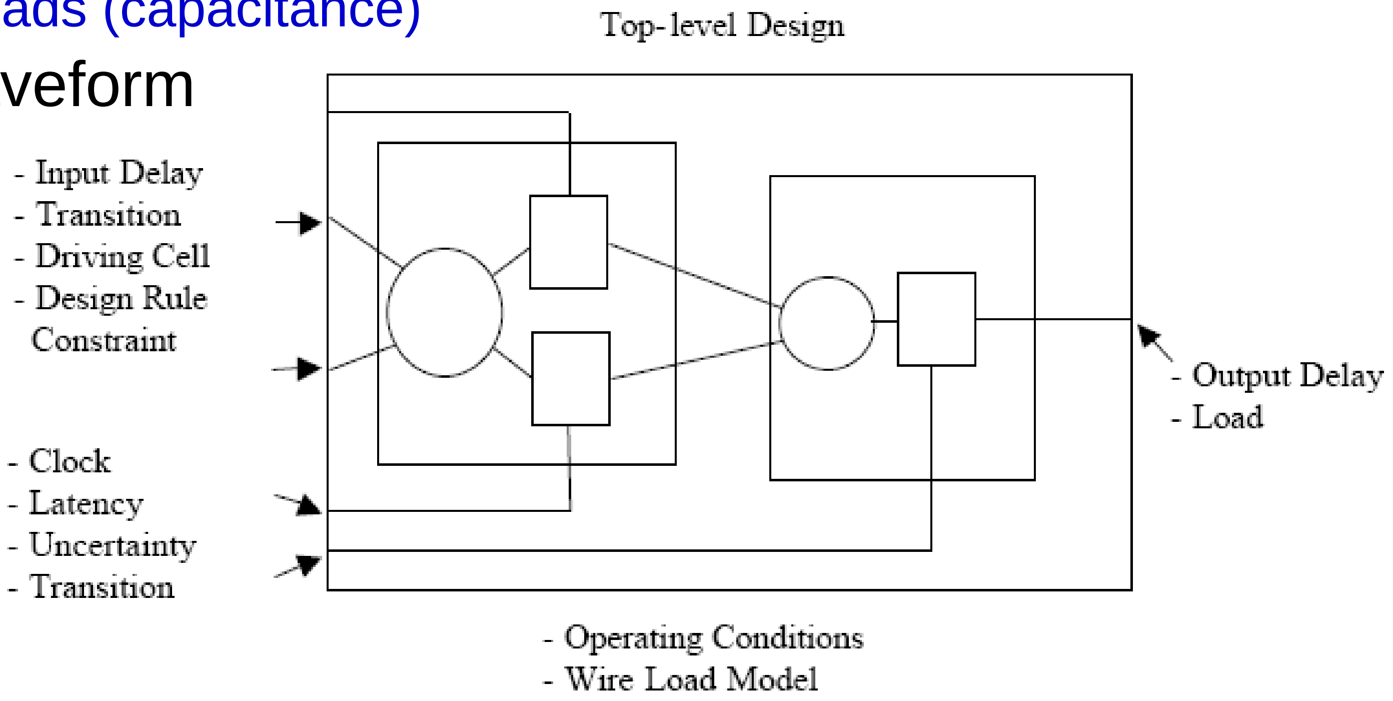
- ≠ design rules for layout editing or mask generation
  - Width, spacing, enclosure

- 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



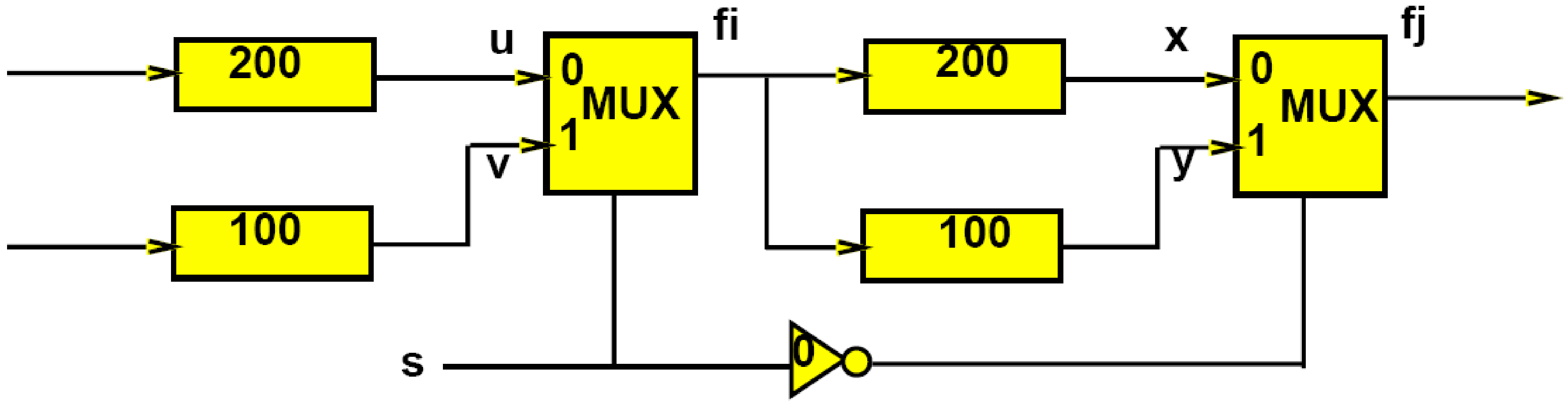
Timing Exceptions

- 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!

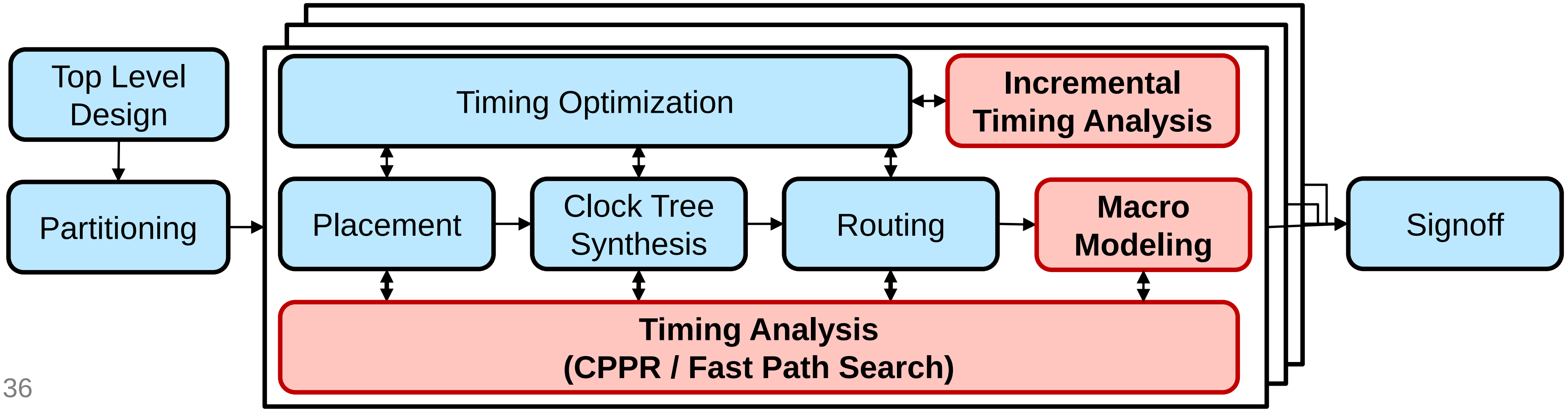
Challenges

- On chip variations: Process, Voltage, Temperature
  - Remove artificial pessimism
- Wire delay/slew calculation
- New cell delay model
- All critical path information is important!
  - K-most vs. the most critical path
- Repeated timing optimization operations
- Hierarchical design for large-scale integration
  - Hierarchical STA

- Key concerns: Accuracy, runtime/memory and scalability
  - Golden: SPICE

Key Techniques for Modern Timers

- On chip variations: Process, Voltage, Temperature
  - Common path pessimism removal (CPPR)
- All critical path information is important!
  - Fast path search
- Repeated timing optimization operations
  - Incremental timing analysis
- Hierarchical design paradigm for large-scale integration
  - Compact and accurate timing macro modeling for hierarchical STA



What We Have Covered Today…

- We have reviewed STA ingredients

- Recent research endeavors
  - Common path pessimism removal
  - Wire timing prediction
  - Fast path search
  - Fast incremental timing analysis
  - Compact and accurate timing macro modeling
  - Current source delay model

- Future directions:
  - Multi-corner multi-mode
  - Signal integrity
  - Voltage awareness
  - Statistical timing analysis

Thank you!