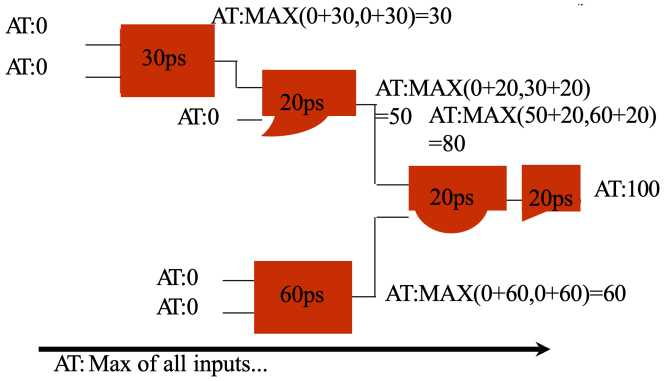
laalcham@ucsc.edu for CSE 125, UCSC
Static Timing Analysis and Variation

Why is it called Static Timing Analysis?

Timing analysis is static because to make something execute fast, the functionality of a circuit must be abstracted away. Every gate in a design is assigned a propagation delay. These propagation delays are used to compute the latency of the longest/critical path between registers in a design. The latency of the critical path determines the maximum clock frequency.

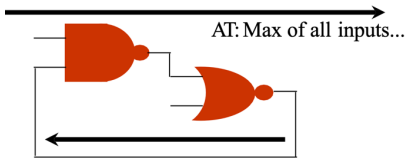
Arrival Time (AT)



Common Error: Timing Loops

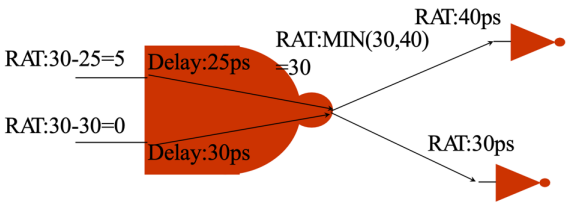
When doing STA, you assume there are no timing loops. The timing graph is assumed to be **acrylic**. A maximum latency cannot be calculated if there is a cycle in the design.

- timing loops are usually due to inferred latches
- flip-flops break up loops.
- the **report_timing -loop** command can be used to find loops in the design compiler



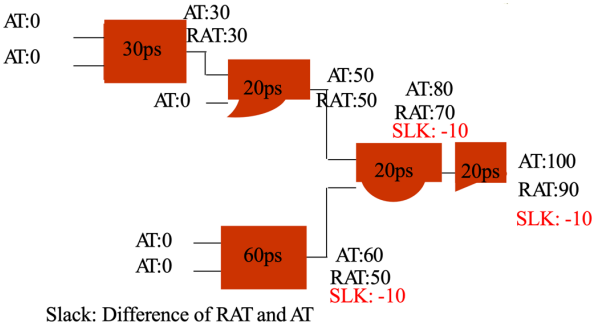
Required Arrival Time (RAT)

The **required arrival time** of a design can be calculated as well. Useful to determine if a design can meeting certain timing constraints. The worst-case assumption of design elements are taken and a minimum latency is calculated by working from output(s) to input(s).



The required arrival time is typically computed for (rise, fall) x (early, late)

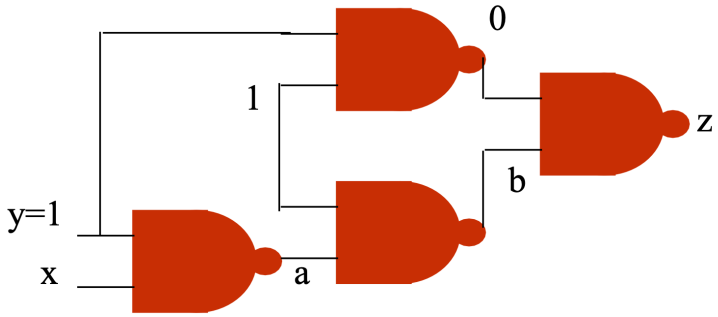
In the example below, timing is violated because the AT exceeds the RAT. There is a negative slack: RAT - AT = SLK = 90 – 100 = -10. To fix the design, pipelining can be introduced.



False Paths

Runtime complexity of STA is linear. If each input to a black box was simulated, the runtime complexity would exponentially scale. There is a tradeoff to ignoring all the different combinations of inputs to a design: **false paths**. A **false path** is a path identified by STA that doesn't actually affect the design (it can never be logically sensitized). False paths often occur with pass gates if all bidirectional cases are not valid.
STA is pessimistic: it calculates an AT that is higher than actually occurs if there are false paths.

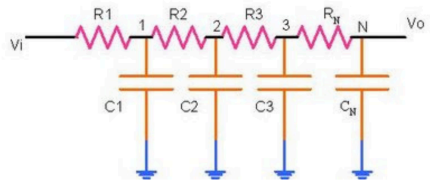
[Example: false path]
The 3-gate path from x to z is a false path. y=1 prevents b from getting to z, and y=1 is needed to get x->a.



Wires in STA

- about 50% of delay in a design is due to wires – and STA ignores wires!
- from a synthesis perspective, it's very hard to predict wire delay. Spacial arrangement of gates are not chosen by the tools
- **place and route (P&R)** is done by placing every gate and then connecting them by routing wires in between them.
  - In a modern ASIC there are 20 layers of wires.
- **parasitics extraction** is done by looking at every wire between 2 gates (length, width, layer, VS) and computing its parasitic capacitance.

- **Net Length**
- **Net cross-sectional area**
- **Resistively of material used for metal layers (Aluminum vs. copper)**
- **Number of vias traversed by the net**
- **Proximity to other nets (crosstalk)**

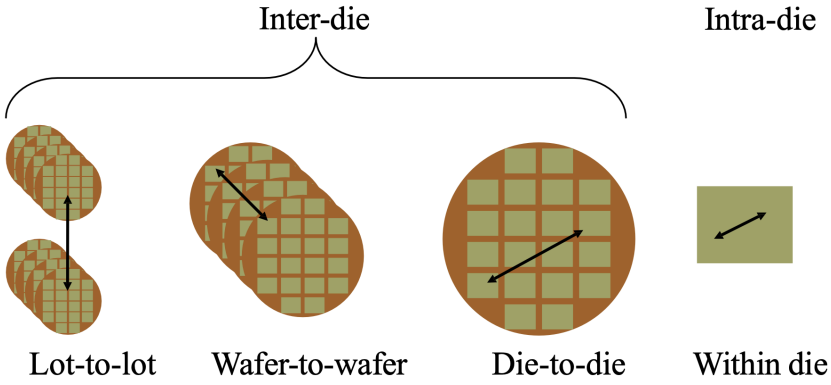


Above is a model of parasitic extraction and how it calculates wire delay. Ideally, there is an infinite # of resistors.

Types of Variation

Recall: chips are made from **wafers**. It is a substrate that chips are printed on. Silicon is made from very hot sand. Pillars up to 2m high are made and then cooled very quickly to create a crystalline structure. They are sliced into wafers, very thin. Chips are printed on the wafers. AS many chips as possible are put on a wafer.

There are many factors that affect the chip making process and many reasons why they can go wrong. Pictured below is how chips can end up different from each other:

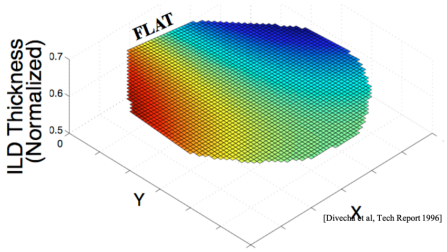


causes include:

- environmental
  - temperature
  - voltage
- physical
  - lithography
  - materials
- fatigue
  - negative-bias temp. instability NBTI (chips degrade over time)

ILD Variation

When talking about variation across a wafer we talk about ILD variation: inter level dielectrics (ILD) of a wafer:



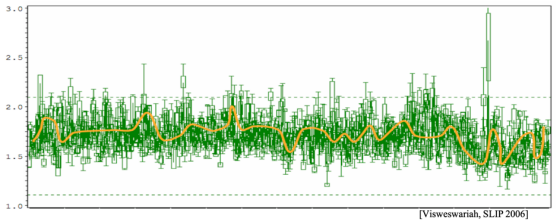
Above, the wire delays in chips in the red portion may be 20% higher than wire delays in the blue portion. Notice the graceful degradation of chips.

- ILD variation is a very significant portion of variability (about 20%)
- high spatial correlation coefficient in a single die.

Normalized Metal Resistance

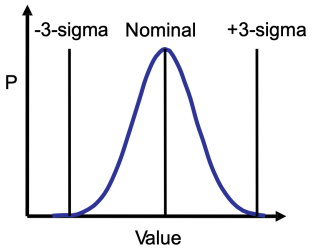
The graph below shows an experiment. Over a 3 month period, metal resistance was observed in the same fab.

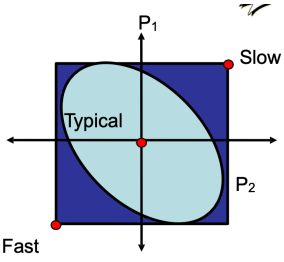
- expected resistance ranges varied from 1.5 to 1.8 (from varied widths of wires)
- range if different wafers at any given time was found to be 30%.



Timing Corners

No longer assume a wire/gate has a fixed propagation delay. Instead, use a normal (Gaussian) distribution and measure standard deviation.

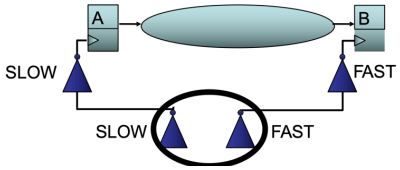




- There are 3 timing corners that STA is run for, P1=+3-sigma, P2=-3-sigma, P3=Nominal.
  - P2: “worst corner” that is good for setup checks
  - P1: “fast corner” that is good for hold checks
  - P3: “typical corner” that is good for estimating how most of the chips will perform.

Common-Path Pessimism

Problem: the shared inverter in the clock tree cannot be both SLOW and FAST. Neither can the wires and gates on the clock path.
Solution: use slow corner delay at the launching clock pin and fastest delay at the capture clock pin.



Statistical STA

Statistical STA tries to consider the normal distribution of chip performance. A distribution is used to evaluate delays instead of a single value.

- research started 20 years ago for analog circuits (they are very sensitive to device mismatch)

Issues with statistical STA:

- computationally expensive
- correlation models are complicated
- who develops the models?

2 main approaches:

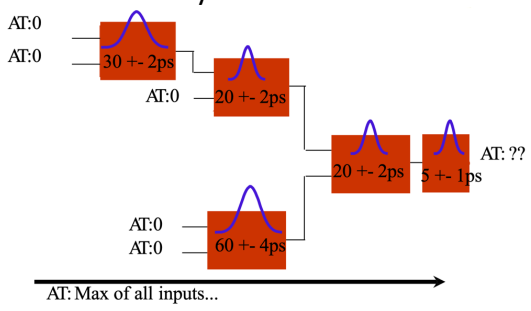
1. path based
2. block-based

Path-Based STA

Path-based STA enumerates the critical paths, gets the delay distribution of each, and tries to find the maximum distribution of the critical paths.
Given:

- set of path delays and their variation
- correlation of all paths

Calculate the maximum of all such paths using either Monte Carlo (randomness) or integration.
Values for each block are picked randomly from the normal distribution.



Monte Carlo
A very flexible mathematical technique. A "stochastic algorithm" • performs random sampling of result to determine properties • useful when a closed-form formula is too complex or doesn't exist • example: integration to compute pi

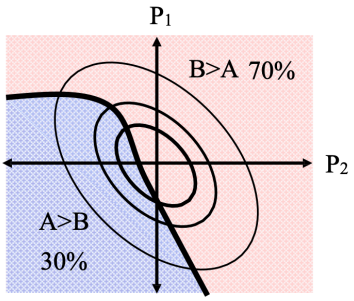
Block-based STA

Block-based STA tries to mimic STA but with a PDF instead of a single AT/RAT

Joint Probability of Two Multivariates

Maximum of two multivariate normal distributions
- First and second moments calculated analytically [Clark 1961,Cain 1994]
- Sensitivities approximated by proportional weight [Visweswariah et al., DAC 2004]

Monte Carlo maximum is very quick
- Can be used as a "check" but is considerably slower.
- Linear regression provides accurate sensitivities.



$$\text{Max}(A,B) = 70\%\ B + 30\%\ A$$

Yield Tradeoff

A tradeoff: Yield versus Chip Performance
- increase yield by lowering performance
- decrease yield to improve performance
- bin chips into categories based on performance
  - fast and expensive, slow and cheap

Yield versus MTBF
- Monte Carlo provides likely path delay by computing the joint probability of different probabilities
- increase yield by allowing chips to sometimes fail
- what MTBF are customers willing to accept?

Summary: STA makes our life both easy and hard. Timing without simulating all possible inputs is much easier but results are pessimistic. Statistical STA models are still uncertain, and high accuracy of the STA requires more runtime. It's an active area of research. ☺