Lecture 10

Circuit Pitfalls

Intel Corporation
jstinson@stanford.edu

Overview

Reading

Lev	Signal and Power Network Integrity
Chandrakasen	Chapter 7 (Logic Families) and Chapter 8 (Dynamic logic)
Gronowski	Dynamic Logic and Latches (talk version of Chapter 8)

Introduction

The basic components on a chip are quite simple, transistors and wires. The key point to remember is that wires are a critical part of your design, and need to planned for and simulated like the transistors. Thus the next level up from transistors should be more than a schematic; it should be a schematic and a floorplan.Designer optimize a circuit to improve its speed, power, or area. Improving these parameters often makes the circuit more complex, which needs to be managed too.

The reason circuit designers then to use simple circuits, is that they are more robust than the 'fancy' designs you read about in journals. This lecture looks at some of the things that can (and have) gone wrong in ICs. Some of quite simple while others are extremely subtle but they all cause the chip to fail.

Cost of Making Mistakes

Design Stage	Approx. Cost	Effort Involved
Initial Design	$10	5 minute fix
Design Review	$100	1 hour re-work
Layout	$1000	10 hours –schematic, layout, simulation
Assembly / Tapeout	$10,000	50 hours rework, validation, re-stream
A-step Silicon	$100,000	200 hours debug/fix, equip costs, new stepping
Sampling	$1,000,000	Delay product launch
Volume Production	$10M to $500M	Product Recall

Complexity

(Why use one transistor, when ten will do)

Why use complex circuits?

- Generally forced into needing them since the simple circuits don't make the spec
- Usually in the critical part of the design. (If it isn't critical, why not use the simple solution)
- Fighting $\Delta t = C\Delta V/i$

Designers must choose their risks carefully

- Therefore, designers become fairly conservative
- Need to validate their design over all corners, and chip environments

Simulation Mismatch

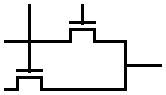
Often stupid stuff but the errors still means the chip fails:

Chip had two node phi1 and Phi1

- There were not connected together on the chip
- Extractor did not complain (different nodes)
- Simulator connected them (case insensitive) (it worked fine)

Now have a tool to check for this

- Use a logic simulator to simulate chip



There was a mux which sometimes had both selects high

- During this cycle the output of the mux did not matter
- Modeled the Mux as a logic element, but the TG is bidirectional

Corrupted one of the inputs, which was needed

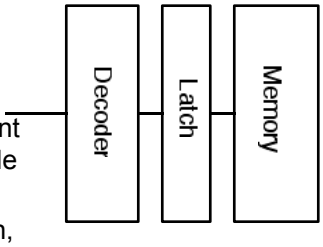
Write checkers for this during logic simulation

Discounting False Paths

In the real world, voltages are analog. This is almost never a problem

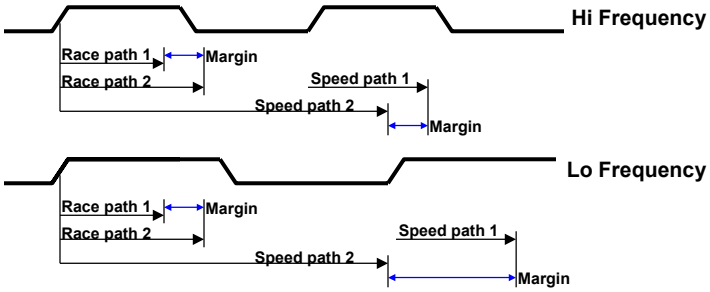
But there once was a chip that had a long false path to the memory

- In this case the decoder output did not settle in time (for the latch)
- The memory was not used in this case, so the address did not matter
- But the decoder outputs settled at different times, so for this long path, it was possible for multiple decoder outputs to be high
- This caused multiple wordlines to go high, which caused one of the locations being read to be written (corrupting real data)



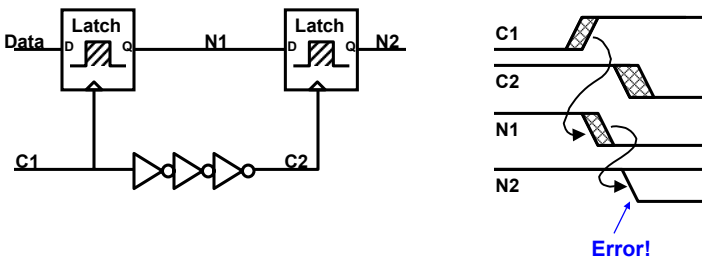
Races

- Definition: signals starting from the same generation point, A, must arrive at a receiving point, B, in a specific temporal order
  - Generation point must be both geographically and temporally the same
    - Temporal component distinguishes a race from a speedpath
  - Need margin in each signal to ensure correct temporal arrival times



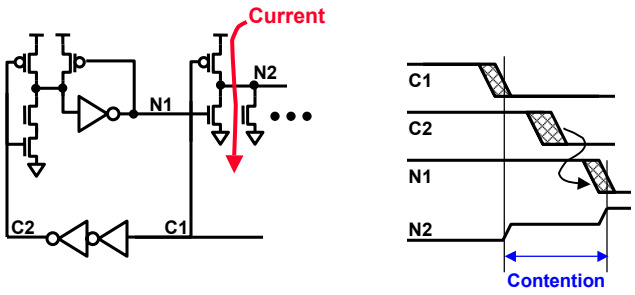
Functional Race

- Causes circuit to produce incorrect results
  - #1 concern with most race conditions
  - $T_{dmin} > T_{hold} + T_{skew} - T_{c\text{-}q}$
- Typically use very conservative estimate for T_{skew}
  - Assume both clock AND delay variance



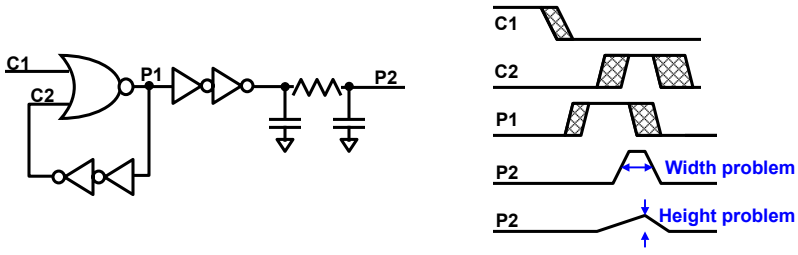
Power Race

- Causes circuit to burn excessive power
  - Does not necessarily cause a functional or circuit failure
  - $T_{dmin} > T_{hold} + T_{skew} - T_{c\text{-}q}$
- Estimation of T_{skew} depends on design constraints
  - Power constrained designs need conservative estimates



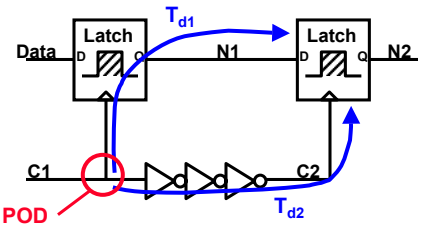
Pulse Evaporation

- Pulse generation is typically a race condition
  - Need to ensure that pulse is wide and tall enough to perform work
  - $T_{pw} > T_{work} + T_{skew}$
  - $T_{pw} > (T_{rise} / 2) + (T_{fall} / 2) + T_{skew}$, assuming T_{pw} is measured at Vdd/2
- Pulse evaporation usually results in a functional failure
  - Requires conservative pulse design
  - Usually need a pulse at least 3 FO4 to survive (4 is better)



Calculating Race Margins ($T_{skew\text{-}margin}$)

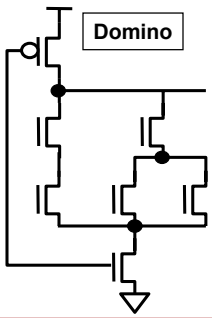
- Skew exists in both the clock path AND the data path
  - $T_{skew\text{-}margin}$ has to take both into account (be careful with T_{skew} terminology)
- Point-of-divergence (POD)
  - Calculate total loop length of racing signals
  - Take percentage of loop length as the $T_{skew\text{-}margin}$
- Statistical
  - More accurate, harder to calculate
  - Random variances add statistically ($T_{skew\text{-}margin}^2 = \sigma t_1^2 + \sigma t_2^2 + \sigma t_3^2 \ldots$)
  - Correlated variances add algebraically ($T_{skew\text{-}margin} = \sigma t_1 + \sigma t_2 + \sigma t_3 \ldots$)

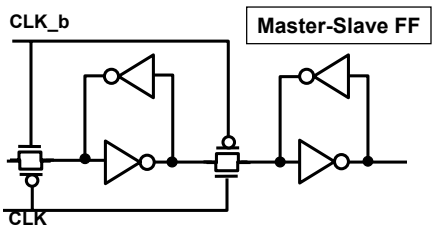


$T_{skew\text{-}margin} = (T_{d1} + T_{d2}) * \alpha$,
where α is usually 0.08-0.15

Charge-Sharing

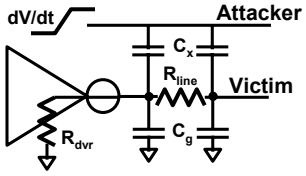
- Domino
  - Classic charge-sharing problem; fix with secondary precharge devices
- Pass-gate
  - Watch out for poorly driven nodes, memory cells, sequentials
- Static CMOS
  - Plain vanilla CMOS is still susceptible to charge-sharing!
  - Glitches can cause downstream failures





Capacitive Wire Coupling

- Can cause both functional and speed failures
- Signals can capacitively couple side-to-side and above/below
- Important elements
  - Victim driver strength (Rdrive and Rline)
  - Attacker slope (dV/dt)
  - Switching cap vs. Quiet cap
  - Attacker switching window relative to Victim
- Use superposition to model multiple attackers
  - Attackers of attackers can have impact on victim nets…..



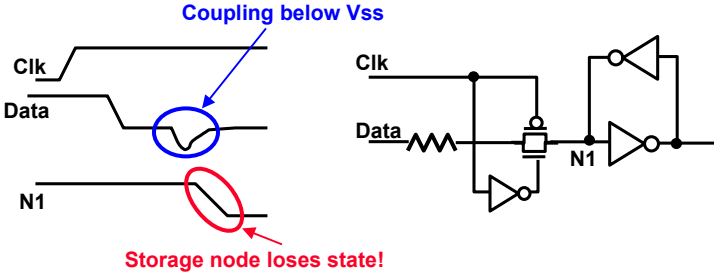
For "quiet" victim:

$$V_{peak} = (R_{tot}) * C_x * \frac{dV}{dt} * (1 - e^{\frac{-dt}{R_{tot}*C_{tot}}})$$

As Rtot ⇨ or dt⇨: $V_{peak} = \frac{C_x}{C_{tot}} * dV$

Wire Coupling and Passgates

- Typically only worry about coupling between Vdd and Vss
  - Noise induced causes downstream circuits to behave poorly
- Passgate inputs susceptible to coupling outside supply rails
  - Coupling above Vdd or below Vss can create Vgs > Vth



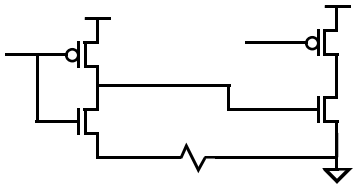
Vss Reference (Gnd Bounce)

Margins are much smaller in a precharge gate

- Switch point is just Vth above Gnd

But Gnd on a chip is not the same

- With Amps of current running through supply can get Gnd drops



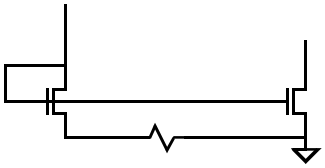
- This can make the margin on a gate even smaller

Gates with keepers have slightly higher margins

- Often require inputs to dynamic logic be locally generated

Current Sources

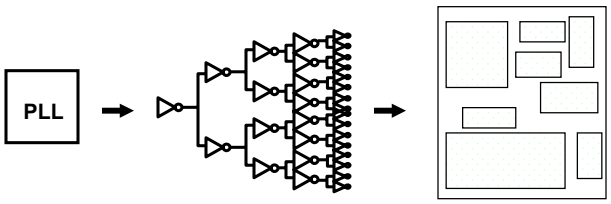
Ground drops area a real problem with making current sources:



- Current sources tend to be used in the analog section of the circuit
  These circuits use DC power
- Tends to want its own (special clean) supply so it does not use grid
  Its supply must be routed
- Gain of current sources tends to be high (30mV can be significant)
- Also watch for Vbb noise changing currents
  - If you need too distribute currents a far distance, pass a current and then go though a mirror

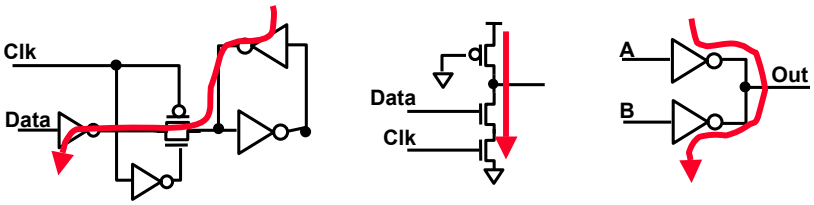
Clock Power Supply

- Clean power supply for PLL
  - Needed to generate low jitter, even duty cycle output
  - Usually separate supply from Vcc to keep it clean
- Power supply choice for clock distribution?
  - Clean power supply provides “perfect” duty cycle, low jitter
  - Core power supply provides better tracking with logic
    - Usually tracking with logic is most important



Contention Circuits

- Rely on sizing and ratios to work correctly
  - Need account for process, voltage and temperature (PVT) variance
  - Correct path must always “win the fight” for correct operation
- Temporary contention circuits
  - Jam latches, memory writes
- Permanent contention circuits
  - Pseudo-NMOS, ratioed logic



Leakage

- Becoming (has become) a major issue in design
  - Accounts for 10-20% of 130nm processor dissipated power
  - Leakage grows 2-4x per generation
- Circuit operation needs to account for leakage
  - Storage nodes are especially susceptible to leakage
    - Domino outputs are treated as storage nodes
  - Sustainer sizes need to be large enough to fight leakage effects
    - Typically results in performance degradation due to contention fights
  - Increasing channel length reduces S-D leakage
    - Exponential reduction in leakage
  - Stacked devices help reduce S-D leakage
    - Strong dependence on Vds

Why do circuits fail?

- Overriding tool warnings or errors
- Poor accounting for parasitics
- Poor accounting for worst case stimulus
- Poor accounting for noise sources
- Poor accounting for variability in process, voltage, temperature
- Tool bugs
- Invalid assumptions

Simulating for Success

- All noise sources accounted for
  - Include model for capacitive coupling
  - Accurate wire model for both attacker and receiver
  - Side-loads
  - Propagated noise from prior stage
  - Supply noise
- Correct simulation stimulus
  - Source worst case initial voltages
  - Active drivers for all switching signals
  - Active loads for circuit under test
- Simulate across worst corner(s)
  - Model changes in PVT