Digital Integrated Circuit Design

CMPE 530/630

Testing and Verification

Dr. Cory Merkel

Learning Objectives

- Identify common fault mechanisms in ICs
- Describe and implement basic testing strategies

Testing

- Testing is one of the most expensive parts of chips
  - Logic verification accounts for > 50% of design effort for many chips
  - Debug time after fabrication has enormous opportunity cost
  - Shipping defective parts can sink a company

- Example: Intel FDIV bug (1994)
  - Logic error not caught until > 1M units shipped
  - Recall cost $450M (!!!)

Logic Verification

- Does the chip simulate correctly?
  - Usually done at HDL level
  - Verification engineers write test bench for HDL
    - Can't test all cases
    - Look for corner cases
    - Try to break logic design
- Ex: 32-bit adder
  - Test all combinations of corner cases as inputs:
    - 0, 1, 2, 2^{31}-1, -1, -2^{31}, a few random numbers
- Good tests require ingenuity

Silicon Debug

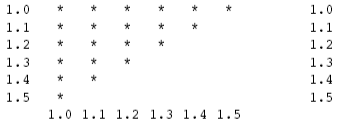
- Test the first chips back from fabrication
  - If you are lucky, they work the first time
  - If not…
- Logic bugs vs. electrical failures
  - Most chip failures are logic bugs from inadequate simulation
  - Some are electrical failures
    - Crosstalk
    - Dynamic nodes: leakage, charge sharing
    - Ratio failures
  - A few are tool or methodology failures (e.g. DRC)
- Fix the bugs and fabricate a corrected chip

Shmoo Plots

- How to diagnose failures?
  - Hard to access chips
    - Picoprobes
    - Electron beam
    - Laser voltage probing
    - Built-in self-test
- Shmoo plots
  - Vary voltage, frequency
  - Look for cause of electrical failures

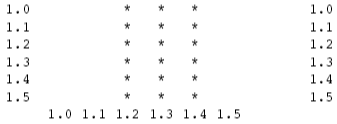
Clock period in ns on the left, frequency increases going up
Voltage on the bottom, increase left to right

** indicates a failure*



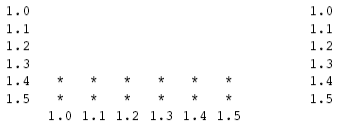
Normal
Well-behaved shmoo
Typical Speedpath

"Brick Wall"
Bistable
Initialization

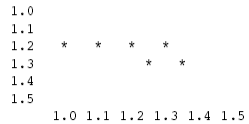


"Wall"
Fails at a certain voltage
Coupling, charge share, races

"Reverse speedpath"
Increase in voltage reduces frequency
Speedpath, leakage



"Floor"
Works at high but not low frequency
Leakage



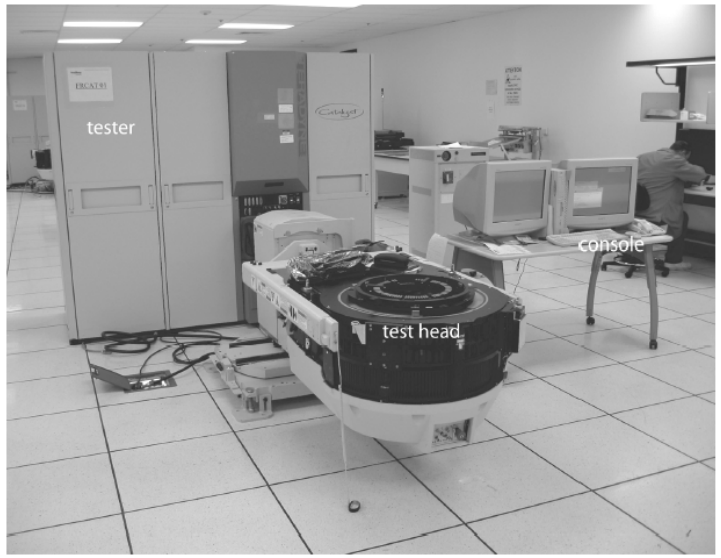
"Finger"
Fails at a specific point in the shmoo
Coupling

Manufacturing Test

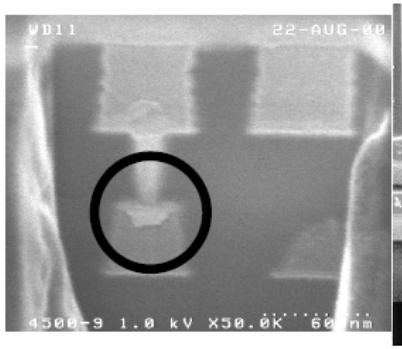
- A speck of dust on a wafer is sufficient to kill chip
- *Yield* of any chip is < 100%
  - Must test chips after manufacturing before delivery to customers to only ship good parts
- Manufacturing testers are

  very expensive
  - Minimize time on tester
  - Careful selection of

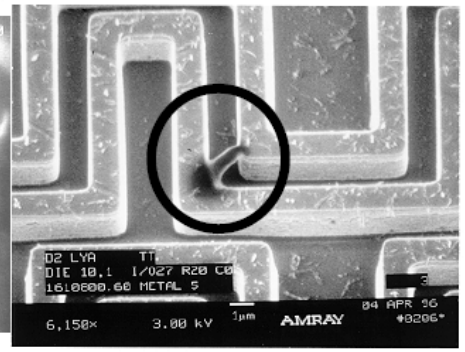
    test vectors



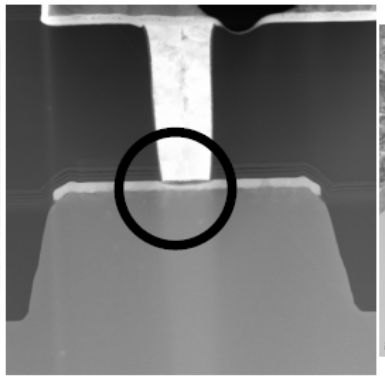
Manufacturing Failures



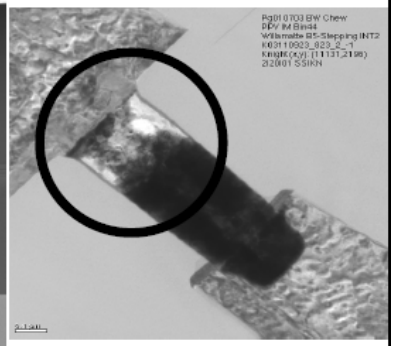
Metal 1 Shelving



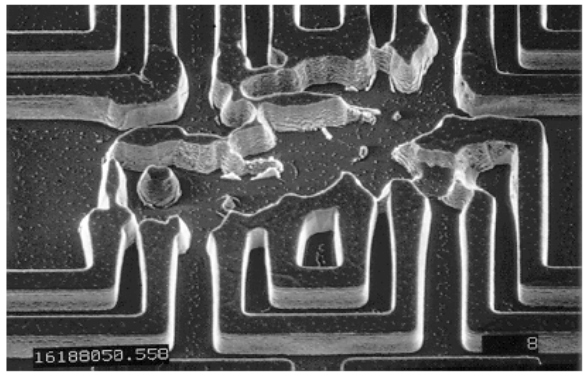
Metal 5 film particle
(bridging defect)



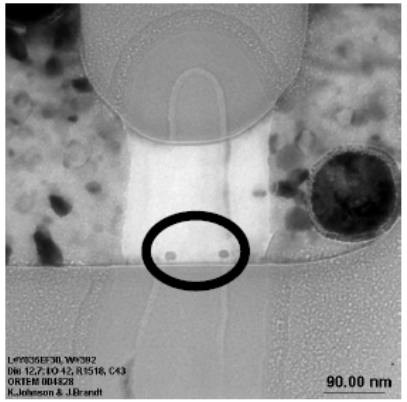
Open defect



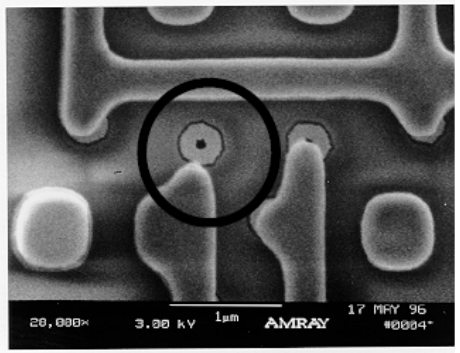
Spongy Via2
(Infant mortality)



Metal 5 blocked etch
(patterning defect)



Spot defects
"Co" Defect under Gate



Metal 1 missing pattern
(open at contact)

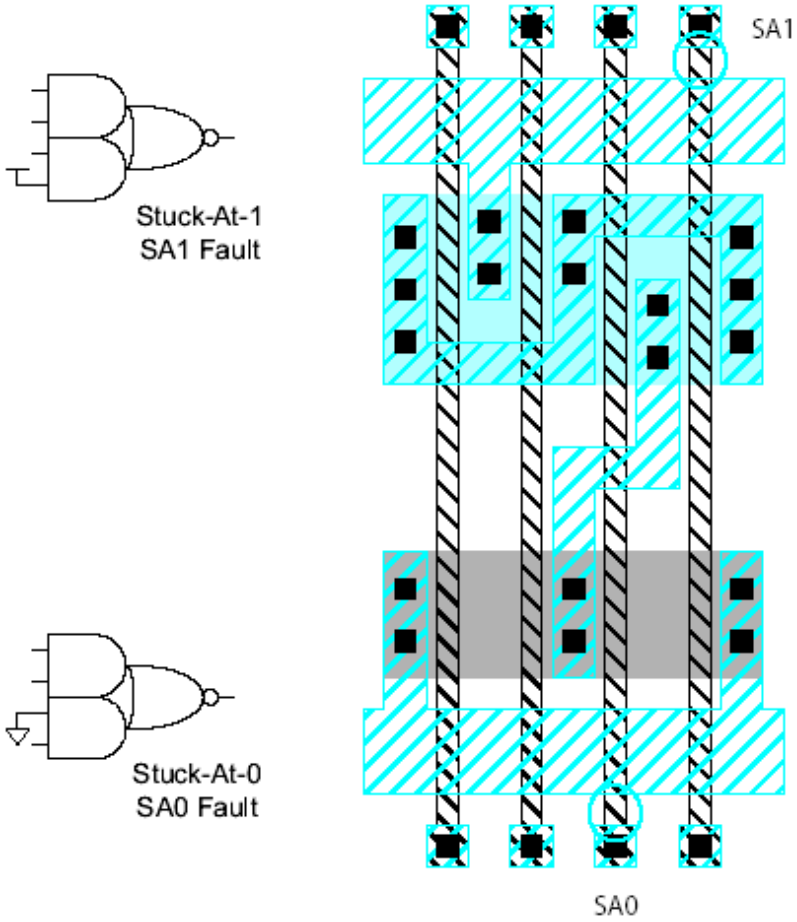
PVT Corners

- ICs need to be tested at corner cases of the process, voltage, and temperature (PVT)
  - Process refers to parameters like threshold voltage that vary during fabrication
    - Corners are Slow N, Slow P (SS), Slow N, Fast P (SF), Fast N, Slow P (FS), Fast S, Fast P (FF), and Normal N, Normal P (NN)
      - Specific types of failures occur at specific corners
  - Voltage and temperature variations also need to be tested
    - Leakage increases with temperature
    - Drive reduces with reduced voltage
    - Etc.

Stuck-At Faults

- How does a chip fail?
  - Usually failures are shorts between two conductors or opens in a conductor
  - This can cause very complicated behavior
- A simpler model: *Stuck-At*
  - Assume all failures cause nodes to be "stuck-at" 0 or 1, i.e. shorted to GND or V_{DD}
  - Not quite true, but works well in practice

Examples



Observability & Controllability

- *Observability*: ease of observing a node by watching external output pins of the chip
- *Controllability*: ease of forcing a node to 0 or 1 by driving input pins of the chip

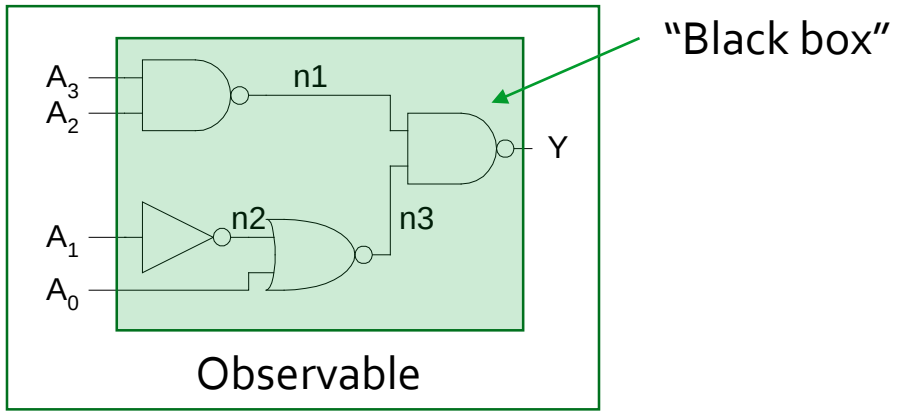
- Combinational logic is usually easy to observe and control
- Finite state machines can be very difficult, requiring many cycles to enter desired state
  - Especially if state transition diagram is not known to the test engineer

Test Pattern Generation

- Manufacturing test ideally would check every node in the circuit to prove it is not stuck.
  - For a circuit with N inputs and M state flip flops, would require at least 2^{N+M} tests. – Not feasible!
- Apply the smallest sequence of test vectors necessary to prove each node is not stuck.
- Good observability and controllability reduces number of test vectors required for manufacturing test.
  - Reduces the cost of testing
  - Motivates design-for-test

Path Sensitization

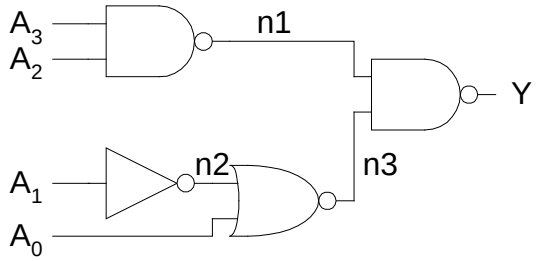
» Path sensitization ensures that an observable output is dependent on a particular node:



» Example: Path from A_3 to Y is sensitized by $n_3 = A_2 = 1$

Finding Minimum Test Pattern Set for 1 SA Fault

	SA1	SA0
» A_3	{0110}	{1110}
» A_2	{1010}	{1110}
» A_1	{0100}	{0110}
» A_0	{0110}	{0111}
» n1	{1110}	{0110}
» n2	{0110}	{0100}
» n3	{0101}	{0110}
» Y	{0110}	{1110}



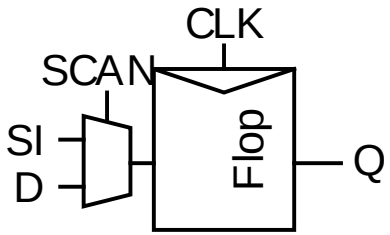
» Minimum set: {0100, 0101, 0110, 0111, 1010, 1110}

Design for Test

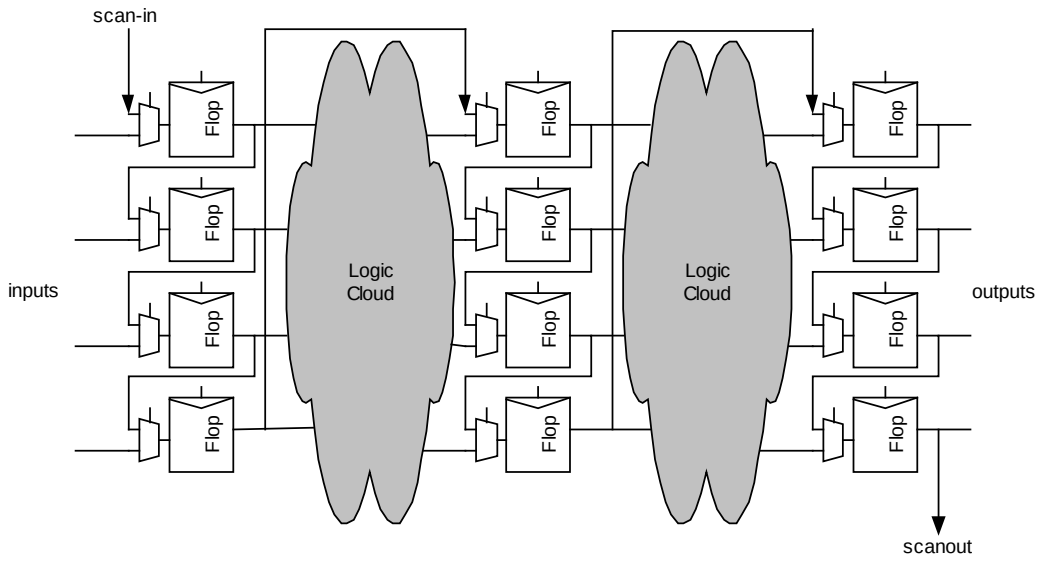
- Design the chip to increase observability and controllability
- If each register could be observed and controlled, test problem reduces to testing combinational logic between registers.
- Better yet, logic blocks could enter test mode where they generate test patterns and report the results automatically.

Scan

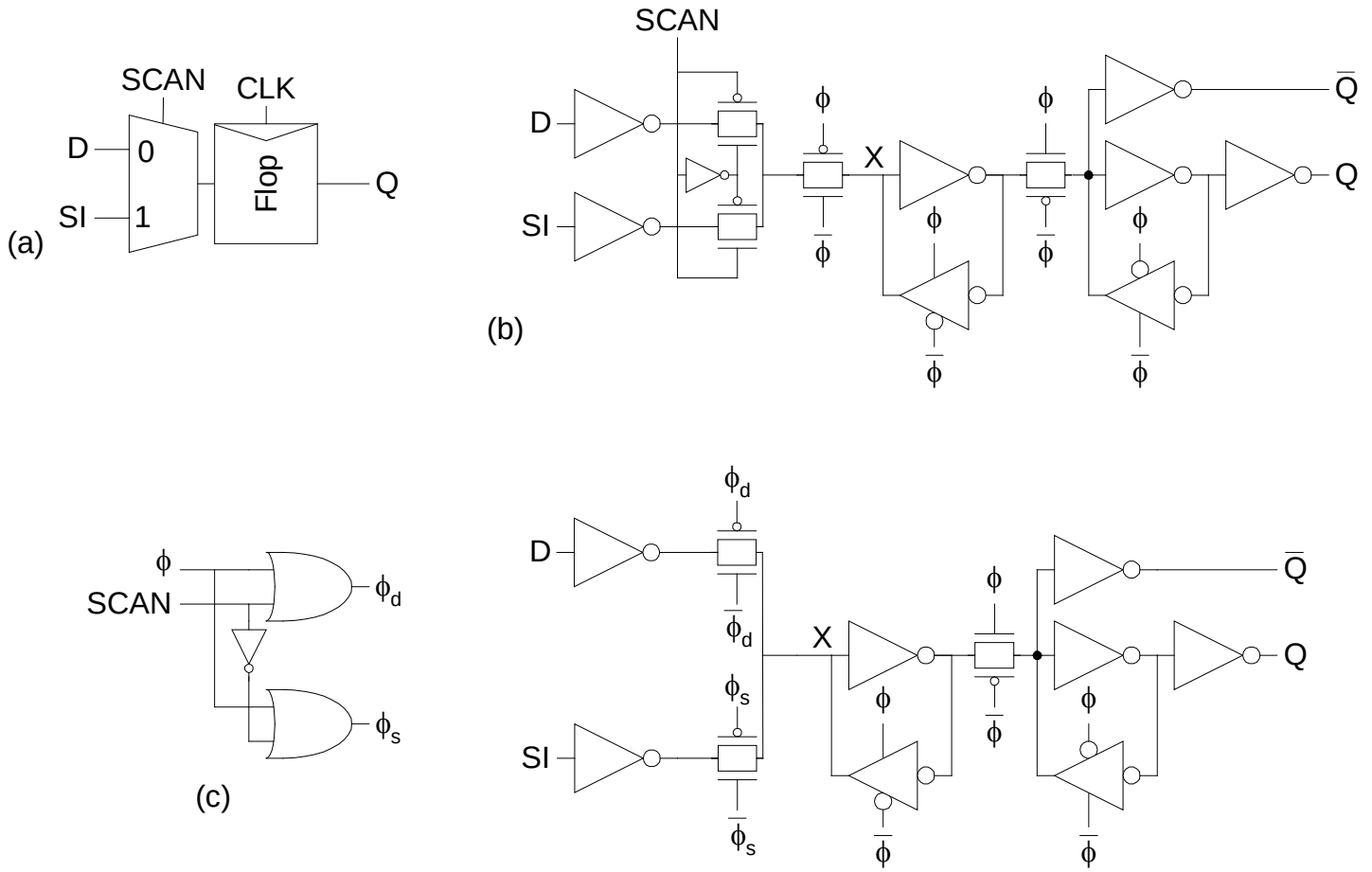
- Convert each flip-flop to a scan register
  - Only costs one extra multiplexer
- Normal mode: flip-flops behave as usual
- Scan mode: flip-flops behave as shift register



- Contents of flops can be scanned out and new values scanned in



Scannable Flip-flops



ATPG

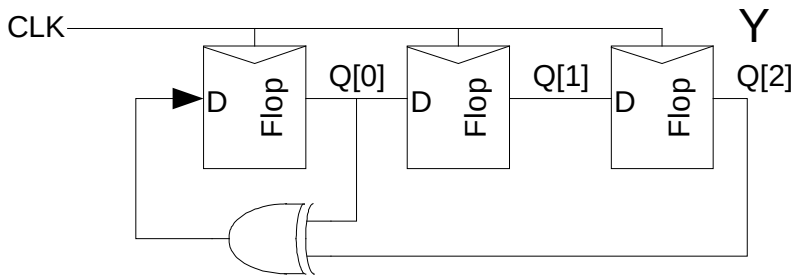
- Test pattern generation is tedious
- Automatic Test Pattern Generation (ATPG) tools produce a good set of vectors for each block of combinational logic
- Scan chains are used to control and observe the blocks
- Complete coverage requires a large number of vectors, raising the cost of test
- Most products settle for covering 90+% of potential stuck-at faults

Built-in Self-test

- Built-in self-test lets blocks test themselves
  - Generate pseudo-random inputs to comb. logic
  - Combine outputs into a *syndrome*
  - With high probability, block is fault-free if it produces the expected syndrome

PRSG

» *Linear Feedback Shift Register*
- Shift register with input taken from XOR of state
- *Pseudo-Random Sequence Generator*

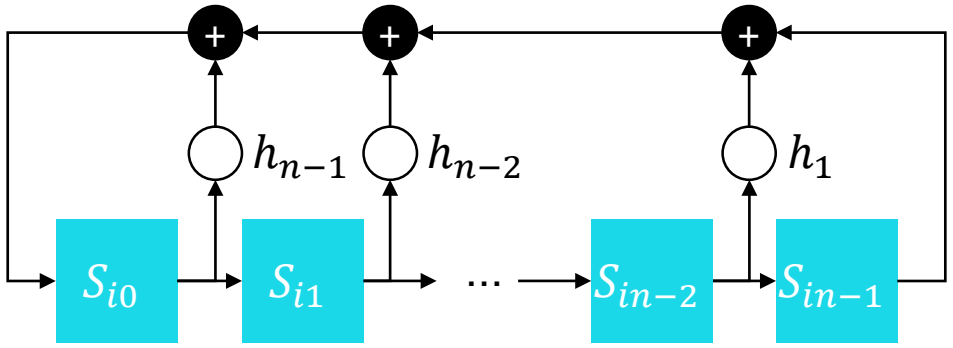


Flops reset to 111

Step	Y
0	111
1	110
2	101
3	010
4	100
5	001
6	011
7	111 (repeats)

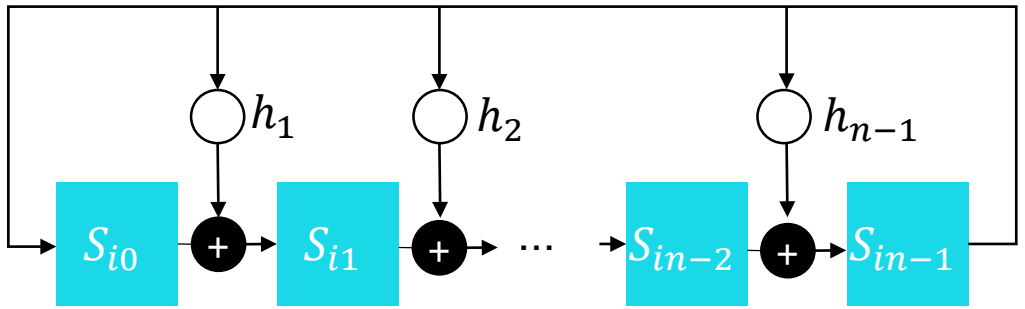
Linear Feedback Shift Registers

» Standard (a.k.a. external XOR, Fibonacci) LFSR



» Modular (a.k.a. internal XOR, Galois) LFSR

- Faster, since there is at most one XOR delay at each stage



A Tiny Bit of Algebra

- **Field**: Set on which addition, subtraction, multiplication, and division are defined, with other properties such as associativity, commutativity, etc. (e.g. rationals, reals)

- **Finite (Galois) Field (GF)**: Field with finite number elements (e.g. integers mod p).

- We are interested in GF(2)—Galois Field with 0s and 1s

Linear Feedback Shift Registers

» Structure of the LFSR is defined by a **characteristic polynomial**:

$$f(x) = 1 + h_1 x + h_2 x^2 + \cdots + h_{n-1} x^{n-1} + x^n$$

» After i shifts of the LFSR seed, the LFSR has polynomial representation:

$$S_i(x) = S_{i0} + S_{i1} x + S_{i2} x^2 + \cdots + S_{in-1} x^{n-1}$$

» If f is a polynomial with coefficients ∈ GF(2) and $f(0) = 1$, then ∃ T such that $f(x)$ divides $1 + x^T$

- T is the **period** of the LFSR

Long Division of Polynomials

- Just like regular long division
- Example:

$$
\begin{array}{r}
x^2 - 3x + 11 \\
x + 3 \overline{\big)\, x^3 + 2x + 4} \\
\underline{x^3 + 3x^2} \\
-3x^2 + 2x + 4 \\
\underline{-3x^2 - 9x} \\
11x + 4 \\
\underline{11x + 33} \\
-29
\end{array}
$$

- When we do division over GF(2), everything becomes mod 2!

Linear Feedback Shift Registers

- If a polynomial with coefficients in GF(2) is **irreducible**, then its period divides $2^n - 1$
  - An irreducible polynomial with period $2^n - 1$ is said to be **primitive**
- Example: What is the period of the LFSR with characteristic polynomial $f(x) = x^4 + x^2 + 1$?
  - $f(x)$ is reducible: $f(x) = (x^2 + x + 1)^2 \bmod 2$
  - $T > 4$

$$\begin{array}{r|l} x^4 + x^2 + 1 & x \ \cdots \\ \hline x^5 + 1 & \\ x^5 + x & \\ \vdots & \end{array}$$

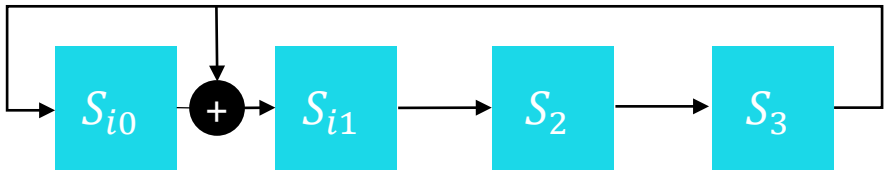
We get a remainder here

$$\begin{array}{r|l} x^4 + x^2 + 1 & x^2 + 1 \\ \hline x^6 + 1 & \\ x^6 + x^4 + x^2 & \\ \hline x^4 + x^2 + 1 & \\ x^4 + x^2 + 1 & \\ \hline 0 & \end{array}$$

No remainder, so $T = 6$

Linear Feedback Shift Register

» Example: Show that the LFSR has a maximum length:

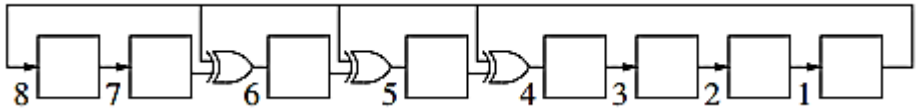


- Show that
- $f(x) = x^4 + x + 1$
- divides $x^{15} + 1$

$$
\begin{array}{r|l}
 & x^{11} + x^8 + x^7 + x^5 + x^3 + x^2 + x + 1 \\
\hline
x^4 + x + 1 & x^{15} + 1 \\
 & x^{15} + x^{12} + x^{11} \\
\hline
 & x^{12} + x^{11} + 1 \\
 & x^{12} + x^9 + x^8 \\
\hline
 & x^{11} + x^9 + x^8 + 1 \\
 & x^{11} + x^8 + x^7 + 1 \\
\hline
 & x^9 + x^7 + 1 \\
 & x^9 + x^6 + x^5 \\
\hline
 & x^7 + x^6 + x^5 + 1 \\
 & x^7 + x^4 + x^3 \\
\hline
 & x^6 + x^4 + x^5 + x^3 + 1 \\
 & x^6 + x^3 + x^2 \\
\hline
 & x^5 + x^4 + x^2 + 1 \\
 & x^5 + x^2 + x \\
\hline
 & x^4 + x + 1
\end{array}
$$

Some Galois LFSR Tap Positions

» Maximum length (i.e. $2^N - 1$ outputs) LFSR with "taps" at positions 8, 6, 5, 4:



» Maximum length LFSRs with 2 and 4 taps:

n	LFSR-2	LFSR-4	n	LFSR-2	LFSR-4	n	LFSR-2	LFSR-4
2	2, 1		24		24, 23, 21, 20	46		46, 40, 39, 38
3	3, 2		25	25, 22	25, 24, 23, 22	47	47, 42	47, 46, 43, 42
4	4, 3		26		26, 25, 24, 20	48		48, 44, 41, 39
5	5, 3	5, 4, 3, 2	27		27, 26, 25, 22	49	49, 40	49, 45, 44, 43
6	6, 5	6, 5, 3, 2	28	28, 25	28, 27, 24, 22	50		50, 48, 47, 46
7	7, 6	7, 6, 5, 4	29	29, 27	29, 28, 27, 25	51		51, 50, 48, 45
8		8, 6, 5, 4	30		30, 29, 26, 24	52	52, 49	52, 51, 49, 46
9	9, 5	9, 8, 6, 5	31	31, 28	31, 30, 29, 28	53		53, 52, 51, 47
10	10, 7	10, 9, 7, 6	32		32, 30, 26, 25	54		54, 51, 48, 46
11	11, 9	11, 10, 9, 7	33	33, 20	33, 32, 29, 27	55	55, 31	55, 54, 53, 49
12		12, 11, 8, 6	34		34, 31, 30, 26	56		56, 54, 52, 49
13		13, 12, 10, 9	35	35, 33	35, 34, 28, 27	57	57, 50	57, 55, 54, 52
14		14, 13, 11, 9	36	36, 25	36, 35, 29, 28	58	58, 39	58, 57, 53, 52
15	15, 14	15, 14, 13, 11	37		37, 36, 33, 31	59		59, 57, 55, 52
16		16, 14, 13, 11	38		38, 37, 33, 32	60	60, 59	60, 58, 56, 55
17	17, 14	17, 16, 15, 14	39	39, 35	39, 38, 35, 32	61		61, 60, 59, 56
18	18, 11	18, 17, 16, 13	40		40, 37, 36, 35	62		62, 59, 57, 56
19		19, 18, 17, 14	41	41, 38	41, 40, 39, 38	63	63, 62	63, 62, 59, 58
20	20, 17	20, 19, 16, 14	42		42, 40, 37, 35	64		64, 63, 61, 60
21	21, 19	21, 20, 19, 16	43		43, 42, 38, 37	65	65, 47	65, 64, 62, 61
22	22, 21	22, 19, 18, 17	44		44, 42, 39, 38	66		66, 60, 58, 57
23	23, 18	23, 22, 20, 18	45		45, 44, 42, 41	67		67, 66, 65, 62

Output Response Analysis

» Ones counting
  - If a circuit has 1 output, we can apply L test patterns and count the number of '1's, then compare to the number of expected '1's, defined as m
  - The probability of *aliasing* (saying we passed all tests when we didn't) will be related to number of ways of having m ones in L bits:

$$P_{oc}(m) = \frac{\binom{L}{m} - 1}{2^L - 1}$$

» Example: Exhaustive testing of a 9-input circuit with $m = 256 \rightarrow P_{oc} \approx 3.5\%$

Output Response Analysis

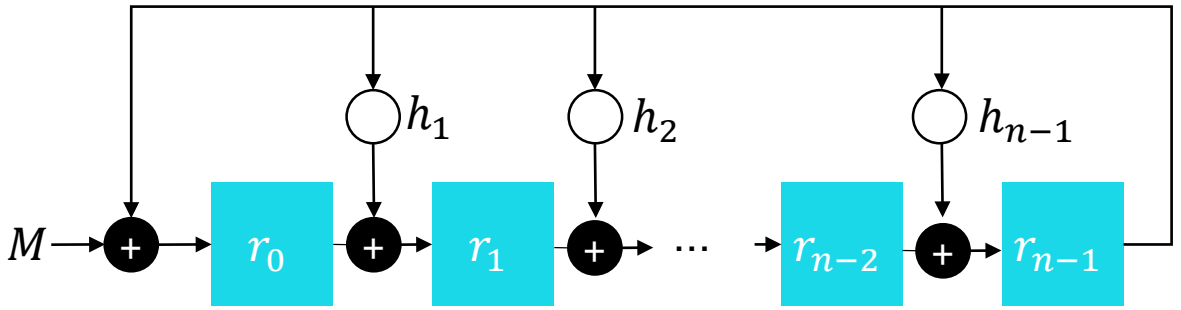
» Transition counting
  - Instead of counting ones, count the number of transitions between 0 and 1, or 1 and 0, and compare to expected number m
  - Probability of aliasing is number of ways to have m transitions in L bits:

$$P_{tc}(m) = \frac{2\binom{L-1}{m} - 1}{2^L - 1}$$

» Example: Exhaustive testing of a 9-input circuit with $m = 256$ → $P_{tc} \approx 3.5\%$

Output Response Analysis

- Signature analysis
  - Create a unique signature that depends on the outputs of the circuit and compare with a known-good or "golden" signature
- Single-input signature register (SISR):



$$M(x) = M_{L-1}x^{L-1} + \cdots + M_1 x + M_0$$

- After shifting M (L shifts) into the SISR

$$r(x) = r_{n-1}x^{n-1} + \cdots + r_1 x + r_0$$

Output Response Analysis

» The SISR performs polynomial division:

$$M(x) = q(x)f(x) + r(x)$$

where $q(\mathrm{x})$ is the quotient and $r(x)$ is the remainder, which will be the final value in the register.

» Note that the SISR's starting value, or seed, is all zeros

Output Response Analysis

» Let us define the error polynomial $E(x)$ as

$$E(x) = M(x) + M'(x)$$

where M' is some faulty sequence

» Expanding this:

$$E(x) = q(x)f(x) + r(x) + q'(x)f(x) + r'(x)$$

» Also, note that

$$\frac{E(x)}{f(x)} = q(x) + q'(x) + \frac{r(x) + r'(x)}{f(x)}$$

» If $r(x) = r'(x)$, then the last term will be 0 after mod 2

» Therefore, if f divides E, then the faults that created M' will not be detected. Otherwise, they will.

Output Response Analysis

- The aliasing probability is related to the number of ways to produce a correct signature with incorrect outputs

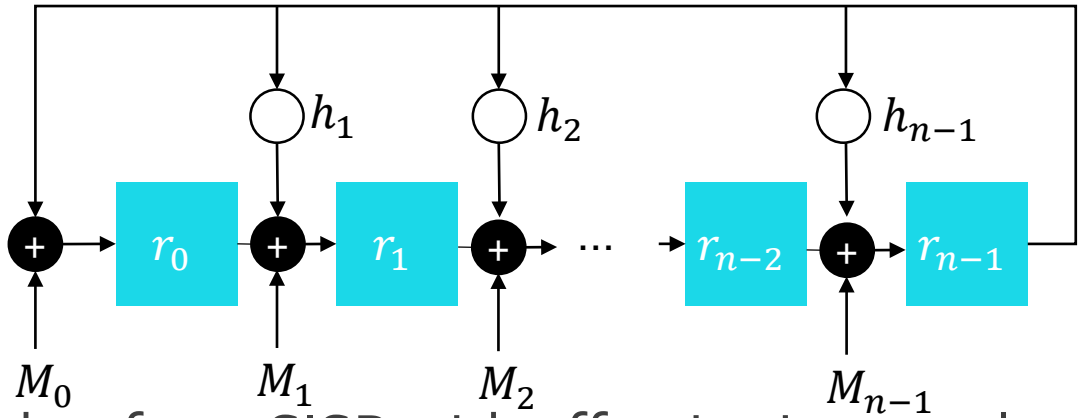
$$M'(x) = q(x)f(x) + r'(x)$$

- If we make $r'(x) = r(x)$, where r is n bits, then we have $L - n$ bits leftover to determine q. Therefore, there are 2^{L-n} possible ways of making an M' that produces $r(x)$.
- So, the aliasing probability is

$$P_{SISR}(n) = \frac{2^{L-n} - 1}{2^L - 1} \approx 2^{-n} \text{ for } L \gg n$$

Output Response Analysis

» Multiple-input signature register (MISR):



» The MISR can be thought of as a SISR with effective input polynomial:

$$M(x) = x^{n-1}M_{n-1}(x) + \cdots + xM_1(x) + M_0(x)$$

and error polynomial

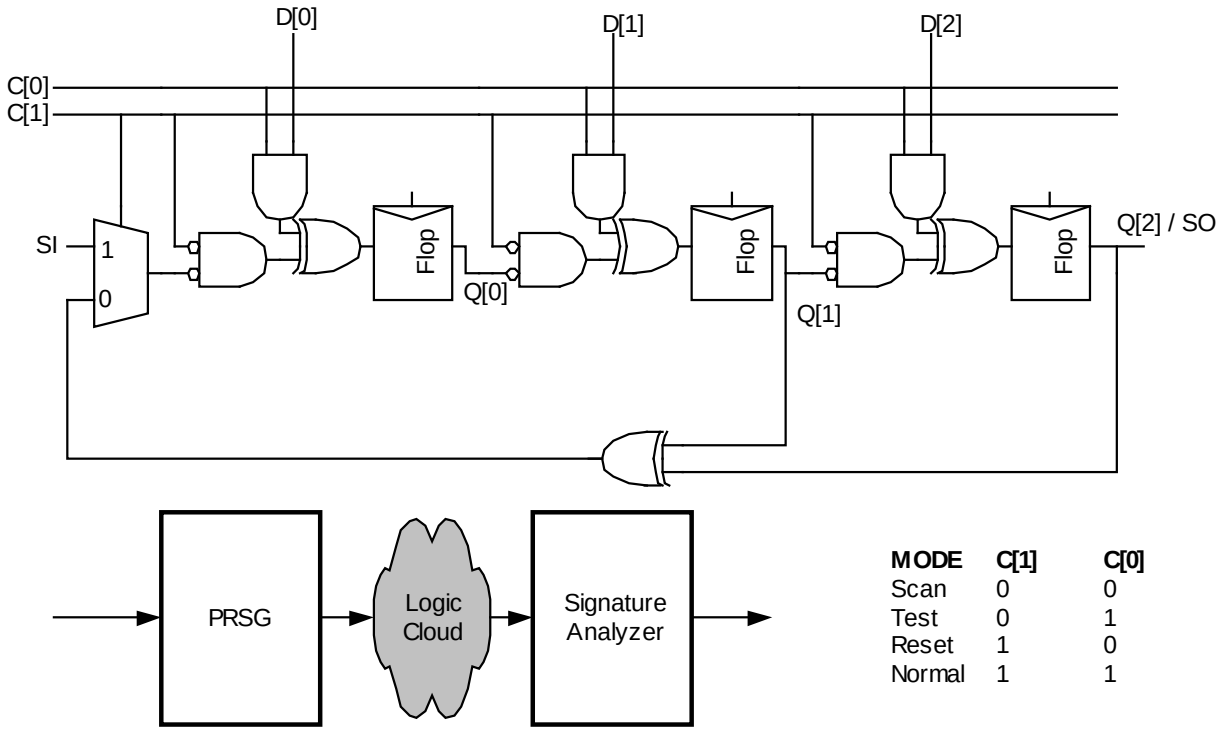
$$E(x) = x^{n-1}E_{n-1}(x) + xE_1(x) + E_0(x)$$

» If there are m circuit outputs connected to the MISR, then the aliasing probability is

$$P_{MISR}(n) = (2^{mL-n} - 1)/(2^{mL} - 1) \approx 2^{-n} \text{ for } L \gg n$$

BILBO

- Built-in Logic Block Observer
  - Combine scan with PRSG & signature analysis

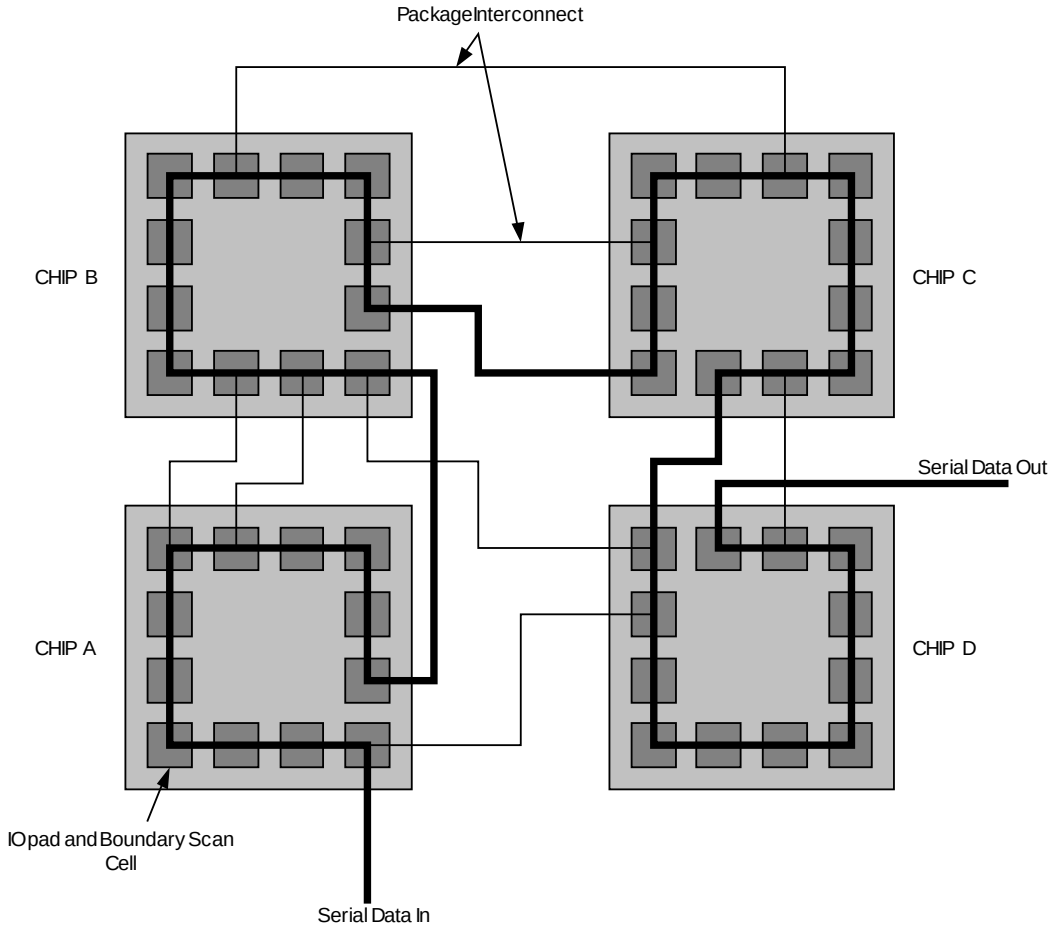


MODE	C[1]	C[0]
Scan	0	0
Test	0	1
Reset	1	0
Normal	1	1

Boundary Scan

- Testing boards is also difficult
  - Need to verify solder joints are good
    - Drive a pin to 0, then to 1
    - Check that all connected pins get the values
- Through-hold boards used "bed of nails"
- SMT and BGA boards cannot easily contact pins
- Build capability of observing and controlling pins into each chip to make board test easier

Boundary Scan Example



Boundary Scan Interface

- Boundary scan is accessed through five pins
  - TCK: test clock
  - TMS: test mode select
  - TDI: test data in
  - TDO: test data out
  - TRST*: test reset (optional)

- Chips with internal scan chains can access the chains through boundary scan for unified test strategy.

Summary

- Think about testing from the beginning
  - Simulate as you go
  - Plan for test after fabrication

- "If you don't test it, it won't work! (Guaranteed)"

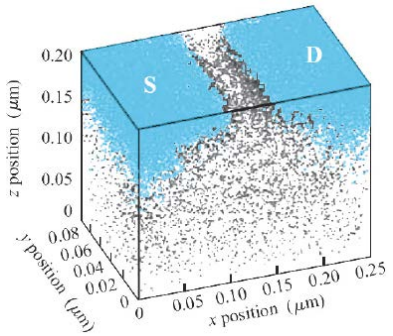
Variation

- Process
  - Threshold
  - Channel length
  - Interconnect dimensions
- Environment
  - Voltage
  - Temperature
- Aging / Wearout

Process Variation

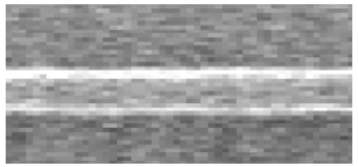
- Threshold Voltage
  - Depends on placement of dopants in channel
  - Standard deviation inversely proportional to channel area

$$\sigma_{V_t} = \frac{t_{ox}}{\varepsilon_{ox}} \frac{\sqrt[4]{q^3 \varepsilon_{si} \phi_b N_a}}{\sqrt{2LW}} = \frac{A_{V_t}}{\sqrt{LW}}$$

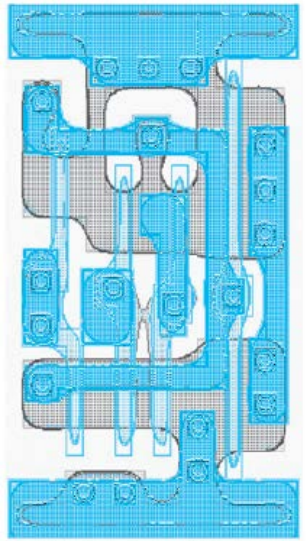


[Bernstein06]

- Channel Length
  - Systematic *across-chip linewidth variation* (ACLV)
  - Random line edge roughness (LER)



Courtesy Texas Instruments

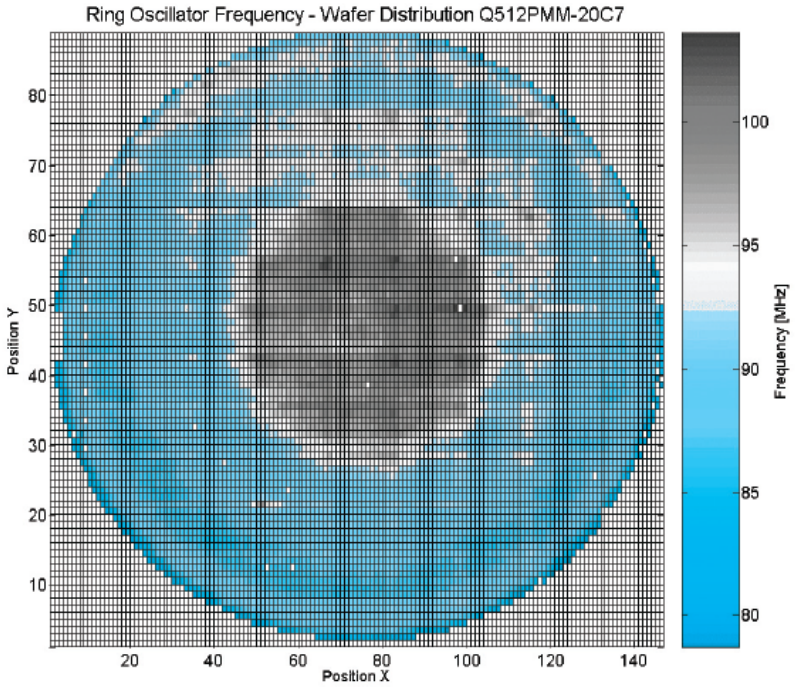


Courtesy Larry Pileggi

- Interconnect
  - Etching variations affect w, s, h

Spatial Distribution

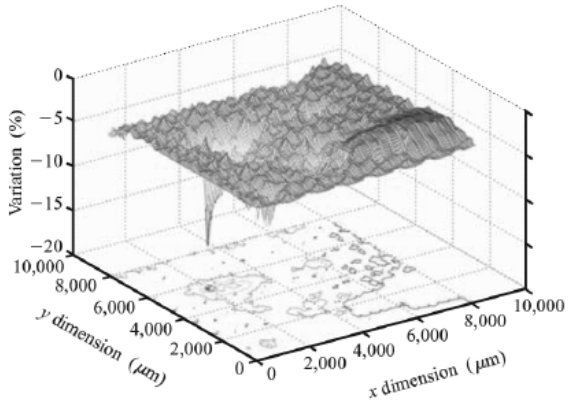
- Variations show spatial correlation
  - *Lot-to-lot* (L2L)
  - *Wafer-to-wafer* (W2W)
  - *Die-to-die* (D2D) / *inter-die*
  - *Within-die* (WID) / *intradie*
- Closer transistors match better



Courtesy M. Pelgrom

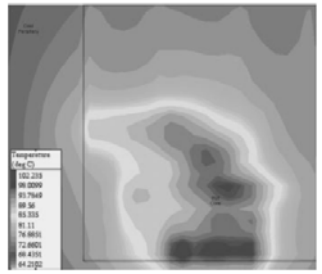
Environmental Variation

- Voltage
  - V_{DD} is usually designed +/- 10%
  - Regulator error
  - On-chip droop from switching activity
- Temperature
  - Ambient temperature ranges
  - On-die temperature elevated by chip power consumption



Courtesy IBM

Standard	Minimum	Maximum
Commercial	0 °C	70 °C
Industrial	−40 °C	85 °C
Military	−55 °C	125 °C



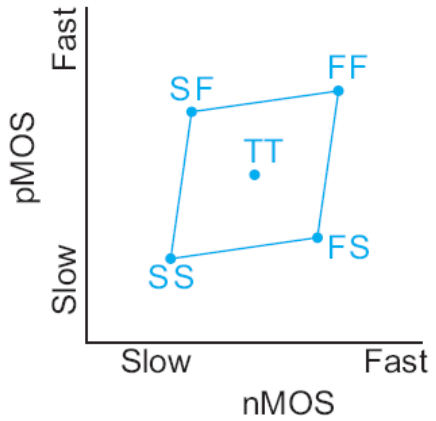
[Harris01b]

Aging

- Transistors change over time as they wear out
  - Hot carriers
  - Negative bias temperature instability
  - Time-dependent dielectric breakdown
- Causes threshold voltage changes
- More on this later…

Process Corners

- Model extremes of process variations in simulation
- Corners
  - Typical (T)
  - Fast (F)
  - Slow (S)
- Factors
  - nMOS speed
  - pMOS speed
  - Wire
  - Voltage
  - Temperature



Corner	Voltage	Temperature
F	1.98	0 °C
T	1.8	70 °C
S	1.62	125 °C

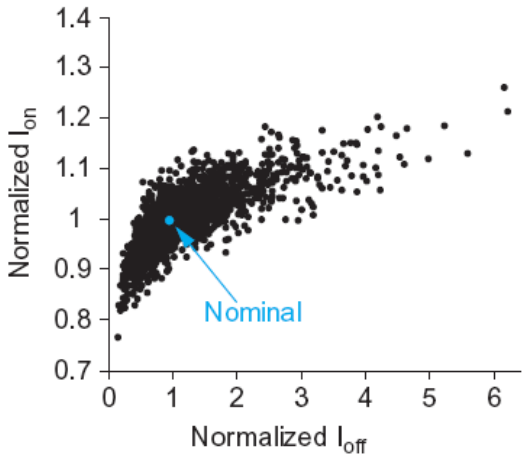
Corner Checks

» Circuits are simulated in different corners to verify different performance and correctness specifications

Corner					Purpose
nMOS	pMOS	Wire	V_{DD}	Temp	
T	T	T	S	S	Timing specifications (binned parts)
S	S	S	S	S	Timing specifications (conservative)
F	F	F	F	F	Race conditions, hold time constraints, pulse collapse, noise
S	S	?	F	S	Dynamic power
F	F	F	F	S	Subthreshold leakage noise and power, overall noise analysis
S	S	F	S	S	Races of gates against wires
F	F	S	F	F	Races of wires against gates
S	F	T	F	F	Pseudo-nMOS and ratioed circuits noise margins, memory read/write, race of pMOS against nMOS
F	S	T	F	F	Ratioed circuits, memory read/write, race of nMOS against pMOS

Monte Carlo Simulation

- As process variation increases, the worst-case corners become too pessimistic for practical design
- Monte Carlo: repeated simulations with parameters randomly varied each time
- Look at scatter plot of results to predict yield
- Ex: impact of V_t variation
  - ON-current
  - leakage

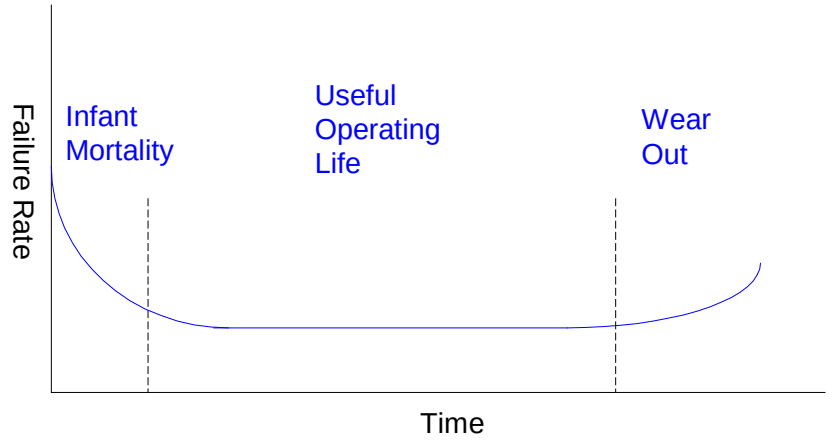


Noise

- Sources
  - Power supply noise / ground bounce
  - Capacitive coupling
  - Charge sharing
  - Leakage
  - Noise feedthrough
- Consequences
  - Increased delay (for noise to settle out)
  - Or incorrect computations

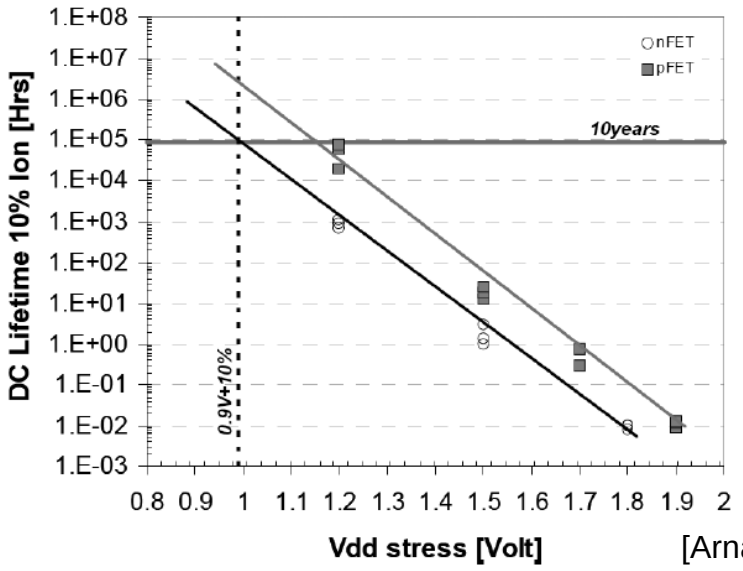
Reliability

- Hard Errors
  - Oxide wearout
  - Interconnect wearout
  - Overvoltage failure
  - Latchup
- Soft Errors
- Characterizing reliability
  - Mean time between failures (MTBF)
    - # of devices x hours of operation / number of failures
  - Failures in time (FIT)
    - # of failures / thousand hours / million devices



Accelerated Lifetime Testing

- Expected reliability typically exceeds 10 years
- But products come to market in 1-2 years
- Accelerated lifetime testing required to predict adequate long-term reliability



[Arnaud08]

Hot Carriers

- Electric fields across channel impart high energies to some carriers
  - These "hot" carriers may be blasted into the gate oxide where they become trapped
  - Accumulation of charge in oxide causes shift in V_t over time
  - Eventually V_t shifts too far for devices to operate correctly
- Choose V_{DD} to achieve reasonable product lifetime
  - Worst problems for inverters and NORs with slow input risetime and long propagation delays

NBTI

- *Negative bias temperature instability*
- Electric field applied across oxide forms dangling bonds called traps at Si-SiO_2 interface
- Accumulation of traps causes V_t shift
- Most pronounced for pMOS transistors with strong negative bias ($V_g = 0$, $V_s = V_{DD}$) at high temperature

$$\Delta V_t = k e^{\frac{E_{ox}}{E_0}} t^{0.25} \qquad E_{ox} = V_{DD}/t_{ox}$$

TDDB

- *Time-dependent dielectric breakdown*
  - Gradual increase in gate leakage when an electric field is applied across an oxide
  - a.k.a *stress-induced leakage current*
- For 10-year life at 125 C, keep E_{ox} below ~0.7 V/nm

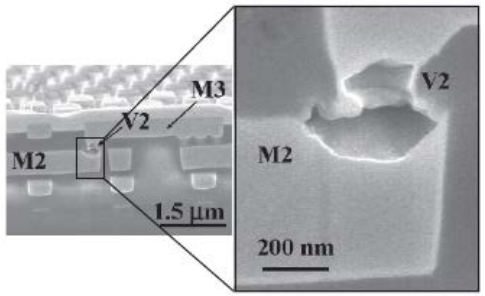
Electromigration

- "Electron wind" causes movement of metal atoms along wires
- Excessive electromigration leads to open circuits
- Most significant for unidirectional (DC) current
  - Depends on current density J_{dc} (current / area)
  - Exponential dependence on temperature

  - Black's Equation:

  - Typical limits: $J_{dc} < 1 - 2$ mA / μm²

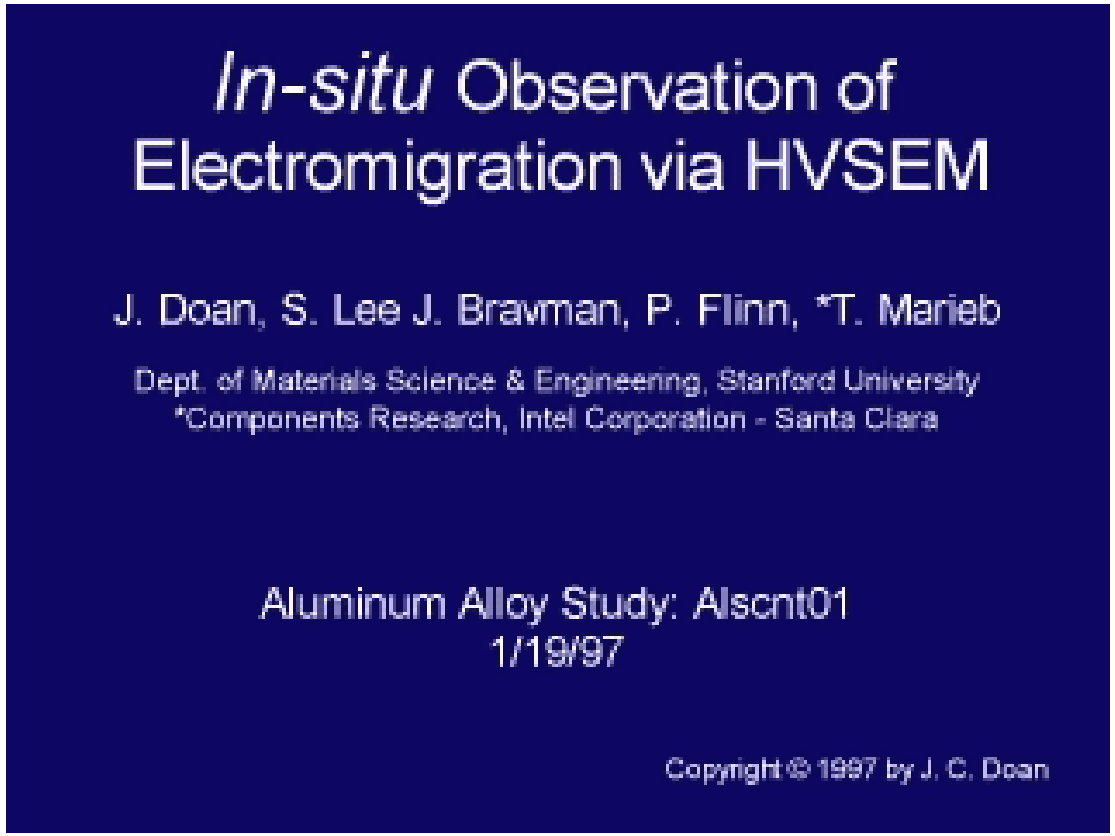
$$MTTF \propto \frac{e^{\frac{E_a}{kT}}}{J_{dc}^{\ n}}$$



[Christiansen06]

Electromigration Video

Electromigration Video 2



Self-Heating

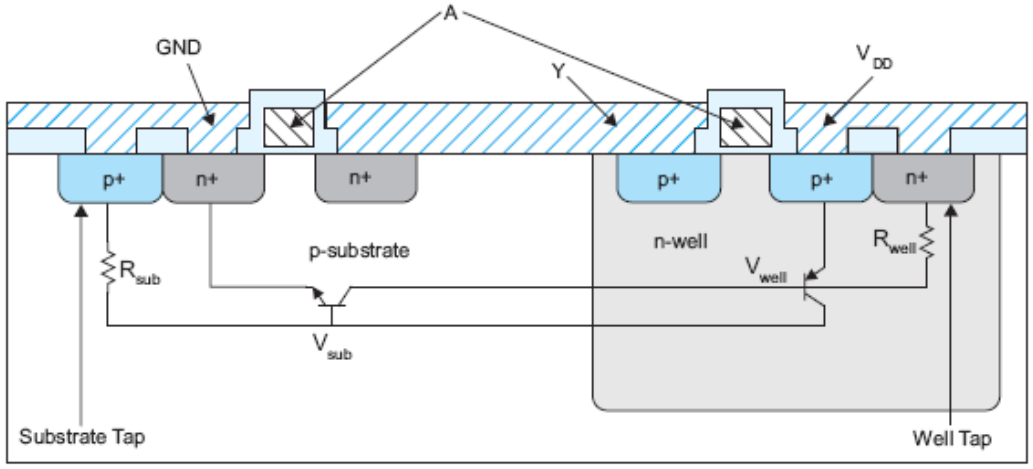
- Current through wire resistance generates heat
  - Oxide surrounding wires is a thermal insulator
  - Heat tends to build up in wires
  - Hotter wires are more resistive, slower
- Self-heating limits AC current densities for reliability
  - Typical limits: $J_{rms} < 1.15 \sqrt{\frac{\int_T I(t)^2 dt}{T}}$ mA / μm²

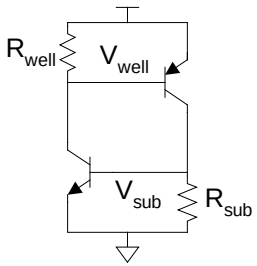
Overvoltage Failure

- High voltages can blow out tiny transistors
- *Electrostatic discharge* (ESD)
  - kilovolts from static electricity when the package pins are handled
- *Oxide breakdown*
  - In a 65 nm process, $V_g \approx 3$ V causes *arcing* through thin gate oxides
- *Punchthrough*
  - High V_{ds} causes depletion region between source and drain to touch, leading to high current flow and destructive overheating

Latchup

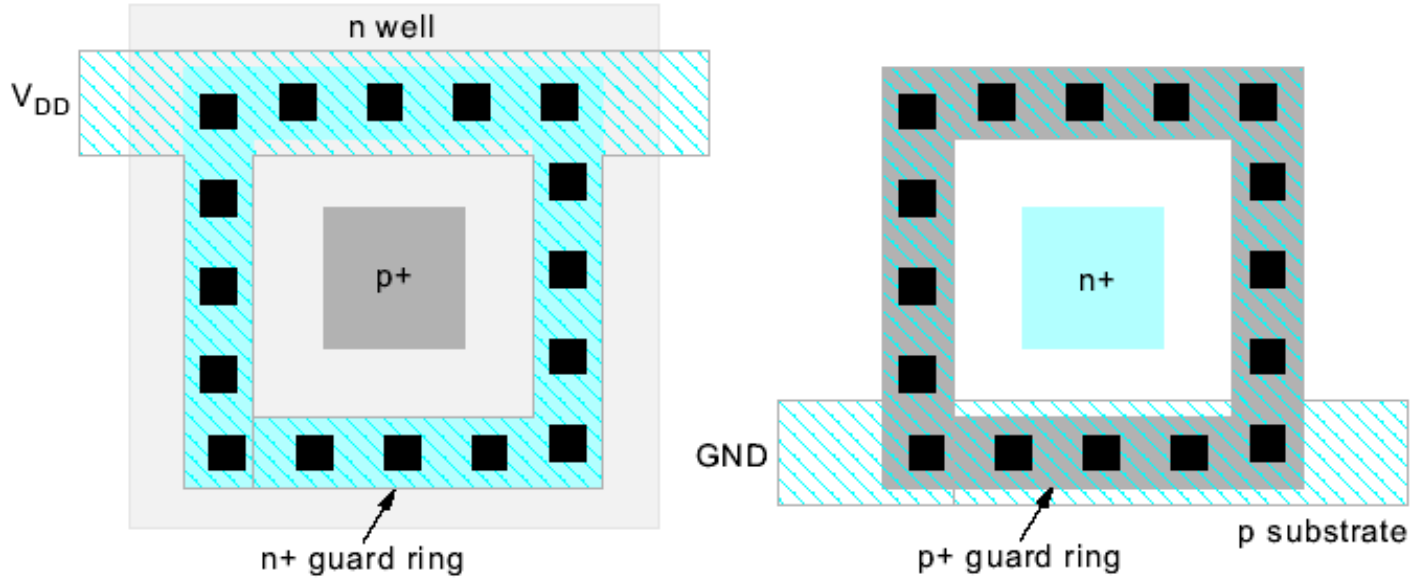
- Latchup: positive feedback leading to V_{DD} – GND short
  - Major problem for 1970's CMOS processes before it was well understood
- Avoid by minimizing resistance of body to GND / V_{DD}
  - Use plenty of substrate and well taps





Guard Rings

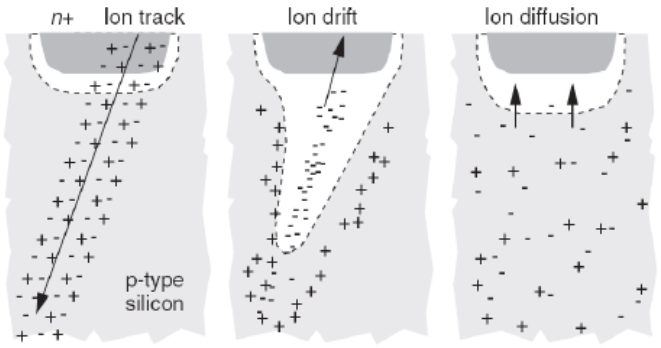
- Latchup risk greatest when diffusion-to-substrate diodes could become forward-biased
- Surround sensitive region with guard ring to collect injected charge



Soft Errors

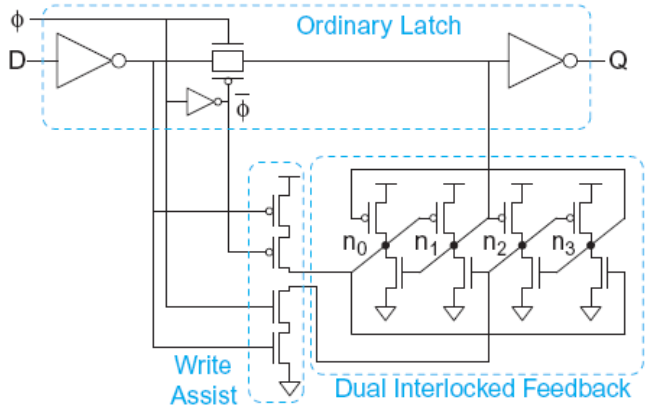
- In 1970's, DRAMs were observed to randomly flip bits
  - Ultimately linked to alpha particles and cosmic ray neutrons
- Collisions with atoms create electron-hole pairs in substrate
  - These carriers are collected on p-n junctions, disturbing the voltage

[Baumann05]



Radiation Hardening

» Radiation hardening reduces soft errors
  - Increase node capacitance to minimize impact of collected charge
  - Or use redundancy
  - E.g. *dual-interlocked cell*



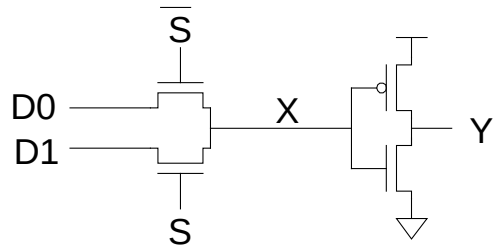
» Error-correcting codes
  - Correct for soft errors that do occur

Circuit Pitfalls

» Detective puzzle
  - Given circuit and symptom, diagnose cause and recommend solution
  - All these pitfalls have caused failures in real chips

Bad Circuit 1

» Circuit
  • 2:1 multiplexer

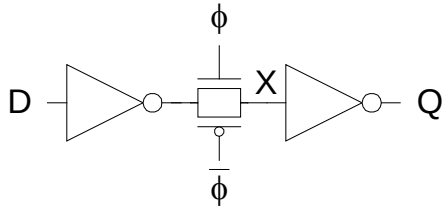


» Symptom
  • Mux works when selected D is 0 but not 1.
  • Or fails at low V_{DD}.
  • Or fails in SFSF corner.

❑ Principle: Threshold drop
  – X never rises above V_{DD}-V_t
  – V_t is raised by the body effect
  – The threshold drop is most serious as V_t becomes a greater fraction of V_{DD}.
❑ Solution: Use transmission gates, not pass transistors

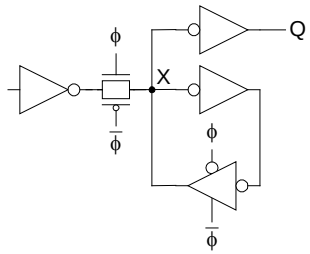
Bad Circuit 2

» Circuit
  • Latch



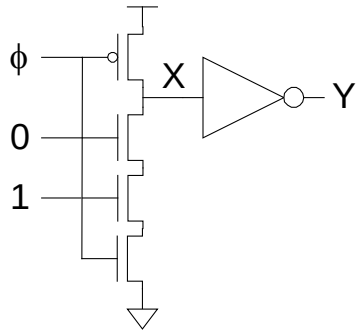
» Symptom
  • Load a 0 into Q
  • Set ϕ = 0
  • Eventually Q spontaneously flips to 1

❑ Principle: Leakage
  – X is a dynamic node holding value as charge on the node
  – Eventually subthreshold leakage may disturb charge
❑ Solution: Staticize node with feedback
  – Or periodically refresh node (requires fast clock, not practical processes with big leakage)



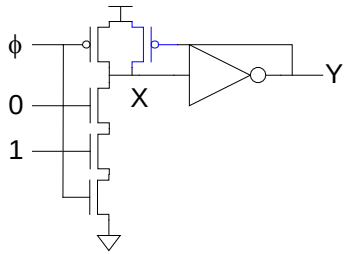
Bad Circuit 3

- Circuit
  - Domino AND gate



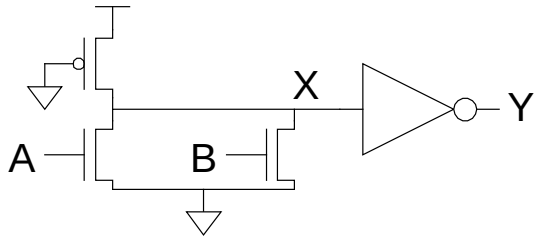
- Symptom
  - Precharge gate (Y=0)
  - Then evaluate
  - Eventually Y spontaneously flips to 1

- Principle: Leakage
  - X is a dynamic node holding value as charge on the node
  - Eventually subthreshold leakage may disturb charge
- Solution: Keeper



Bad Circuit 4

» Circuit
  • Pseudo-nMOS OR

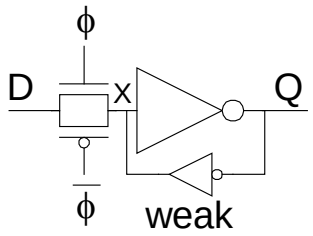


» Symptom
  • When only one input is true, Y = 0.
  • Perhaps only happens in SF corner.

❑ Principle: Ratio Failure
  – nMOS and pMOS fight each other.
  – If the pMOS is too strong, nMOS cannot pull X low enough.
❑ Solution:

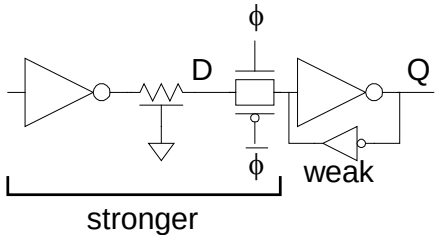
Bad Circuit 5

» Circuit
  • Latch



» Symptom
  • Q stuck at 1.
  • May only happen for certain latches where input is driven by a small gate located far away.

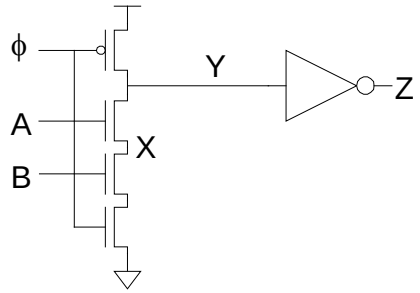
❑ Principle: Ratio Failure (again)
  – Series resistance of D driver, wire resistance, and tgate must be much less than weak feedback inverter.



❑ Solutions:

Bad Circuit 6

» Circuit
  • Domino AND gate



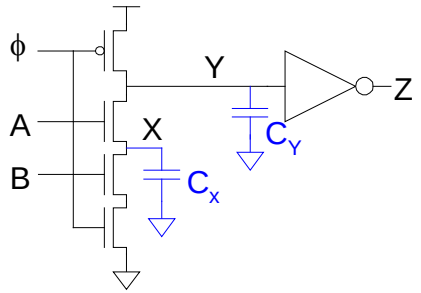
» Symptom
  • Precharge gate while
    A = B = 0, so Z = 0
  • Set ϕ = 1
  • A rises
  • Z is observed to sometimes rise

❑ Principle: Charge Sharing
  – If X was low, it shares charge with Y

❑ Solutions: Limit charge sharing

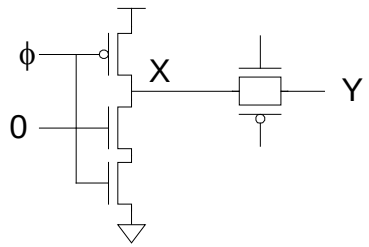
$$V_x = V_Y = \frac{C_Y}{C_x + C_Y} V_{DD}$$

  – Safe if $C_Y >> C_X$
  – Or precharge node X too



Bad Circuit 7

» Circuit
  • Dynamic gate + latch

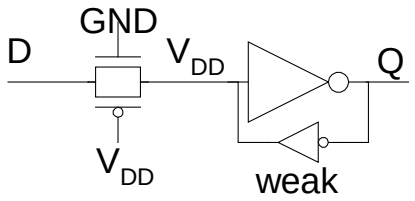


» Symptom
  • Precharge gate while transmission gate latch is opaque
  • Evaluate
  • When latch becomes transparent, X falls

❑ Principle: Charge Sharing
  – If Y was low, it shares charge with X
❑ Solution: Buffer dynamic nodes before driving transmission gate

Bad Circuit 8

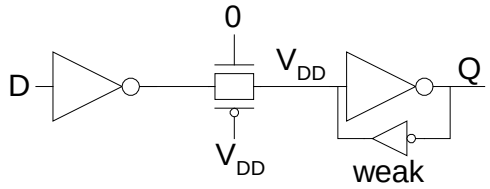
» Circuit
- Latch



» Symptom
- Q changes while latch is opaque
- Especially if D comes from a far-away driver

❑ Principle: Diffusion Input Noise Sensitivity
  – If D < -V_t, transmission gate turns on
  – Most likely because of power supply noise or coupling on D

❑ Solution: Buffer D locally



Summary

- Static CMOS gates are very robust
  - Will settle to correct value if you wait long enough
- Other circuits suffer from a variety of pitfalls
  - Tradeoff between performance & robustness
- Essential to check circuits for pitfalls
  - For large chips, you need an automatic checker.
  - Design rules aren't worth the paper they are printed on unless you back them up with a tool.