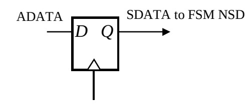
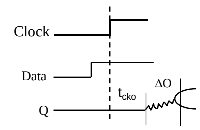
0 Admin
1 Recap
2 Testability
3 Error detection and correction

CS150 Fall 1999 Lecture 25 Copyright © 1999 The Regents of the University of California R. Fearing

1 RECAP

Metastability





ΔO metastable time

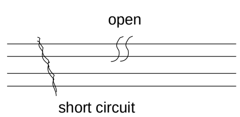
$$\text{Failure Rate} = \frac{1}{\text{MTBF}} = f_1 \cdot f_2 \cdot k_1 \cdot e^{-k_2 \Delta O}$$

Testability – Boundary Scan (long shift register)

CS150 Fall 1999 Lecture 25 Copyright © 1999 The Regents of the University of California R. Fearing

2 STATIC FAULT TESTING

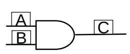
Metallization



Model: Single fault for combinational logic

Goal: Avoid exhaustive testing → 16 inputs, 2^{16} tests ...

Example: Test AND gate

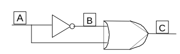


CS150 Fall 1999 Lecture 25 Copyright © 1999 The Regents of the University of California R. Fearing

2 STATIC FAULT TESTING (cont.)

Node	Test vector A B	Expected output
C Sa 0	1 1	1
C Sa 1	00, 01, or 10	0
A Sa 0	1 1	1
A Sa 1	0 1	0
B Sa 0	1 1	1
B Sa 1	1 0	0

Unminimized logic is not completely testable!



Node to test	Test vector	Expected output
A Sa 0	?	
A Sa 1	?	
B Sa 0	0	1 ← only testable combination
B Sa 1	?	

CS150 Fall 1999 Lecture 25 Copyright © 1999 The Regents of the University of California R. Fearing

2 STATIC FAULT TESTING (cont.)

Summary comparison

8 input AND gate: exhaustive $2^8 = 256$

Single Fault Model: (SA0, SA1) × 8 pairs = 16 tests

Big open research problem: Verification
big problem: pattern sensitivity
complexity explodes

Computer in class: MAR, MBR, IR, PC, AC

$5 \text{ Regs} \cdot 16 \text{ bits}$ $\quad 2^{80}$ states + Control FSM

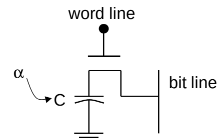
exhaustive test 1.2×10^{24} hopeless $\quad \sim 2^{100}$ states

CS150 Fall 1999 Lecture 25 Copyright © 1999 The Regents of the University of California R. Fearing

3 ERROR CORRECTION

Correct for random errors (not design errors)

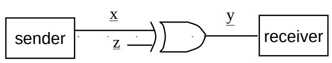
Where: 1) magnetic and optical disks – bit rot from unstable polymers
2) dynamic RAM getting nuked by α particle or cosmic ray



$C \approx 1\,\text{pf}$

3 ERROR CORRECTION (cont.)

3.1 Model of Noisy Channel



$\underline{y} = \underline{x} \oplus \underline{z}$ ← noise (bitwise)

Example: $\underline{x} = 1001$
$\underline{z} = 0100$
$\underline{y} = 1101$

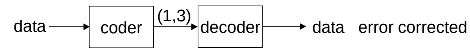
single bit error (not good if your bank account!)

3 ERROR CORRECTION (cont.)

3.2 How to improve reliability?

First look at brute force: send multiple copies, use majority rule, e.g., (1,3) code

data bits — total code bits



CODER

IN	OUT
0	000
1	111

DECODER

0 0 0	0
0 0 1	0
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	1
1 1 0	1
1 1 1	1

Assume no more than single bit error:
- ☞ 2 errors? Can't detect
- ☞ 3 errors? Looks perfect

3 ERROR CORRECTION (cont.)

3.2 How to improve reliability? (cont.)

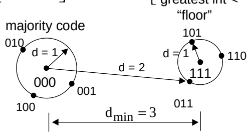
General principle involved – "distance between code words" recall

$\text{distance}\{000, 011\} = 2$

Theorem (without proof)

A code is capable of correcting all error patterns of distance $\leq e$ iff $e \leq \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor$

⌊ greatest int < "floor"



$e \leq \left\lfloor \frac{3-1}{2} \right\rfloor = 1$

To correct 2 bit errors, what would d_{min} be? $\rightarrow$ $\left\lfloor \frac{5-1}{2} \right\rfloor = 2$

3 ERROR CORRECTION (cont.)

3.3 Hamming Code – $\underline{x} = (7,4)$ $4 \text{ data bits} + 3 \text{ parity bits}$

$\underline{x} = [\underbrace{a\ b\ c\ d}_{\text{data}}\ e\ f\ g]$

$$\left.\begin{aligned} e &= a \oplus b \oplus c \rightarrow (a+b+c+e) = 0 \\ f &= a \oplus b \oplus d \quad (a+b+d+f) = 0 \\ g &= a \oplus c \oplus d \quad (a+c+d+g) = 0 \end{aligned}\right\} \text{3 simultaneous equations}$$

In matrix form:

$$\begin{array}{ccccccc} a & b & c & d & e & f & g \end{array}$$
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \end{bmatrix} = 0$$

7 unique columns

$H\underline{x}^T = 0$

Total valid code words? (a,b,c,d)
16

3 ERROR CORRECTION (cont.)

3.3 Hamming Code (cont.) – Matrix:

Let $\underline{y}^T = \underline{x}^T \oplus \underline{z}^T$ (errors) or plus

$H\underline{y}^T = H(\underline{x}^T + \underline{z}^T) = Hz^T \quad \text{if} \neq 0 \text{ must be error}$

Example: $Hz^T = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$ What is z? single bit assumed z = [0 0 1 0 0 0 0] error in c bit

Can correct single bit error.

Fancy convolution codes – Galileo, Reed Solomon