

---

# EECS 427

## Lecture 21: Design for Test (DFT)

### Readings: Insert H.3, CBF Ch 25

EECS 427 F09

Lecture 21

1

## Reminders

---

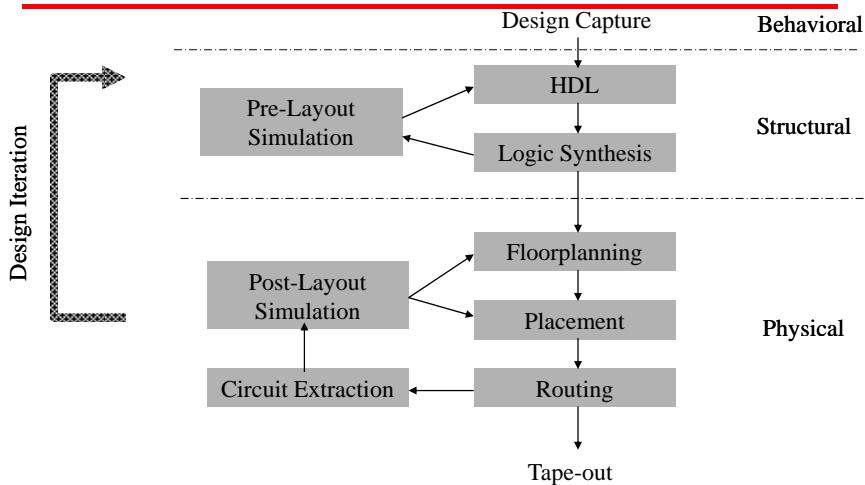
- One more deadline
  - Finish your project by Dec. 14
  - Schematic, layout, simulations, and final assembly (CAD9)
  - Final report and project presentation (HW5)
- Remaining office hours this week
  - Sun 3 – 6 pm
- Remaining lectures
  - 12/2 Wednesday: Design for test
  - 12/7 Monday: Zhengya's research
  - 12/9 Wednesday: Clock and power distribution
  - 12/14 Monday: Project presentation

EECS 427 F09

Lecture 21

2

# Semicustom Design Flow



EECS 427 F09

Lecture 21

3

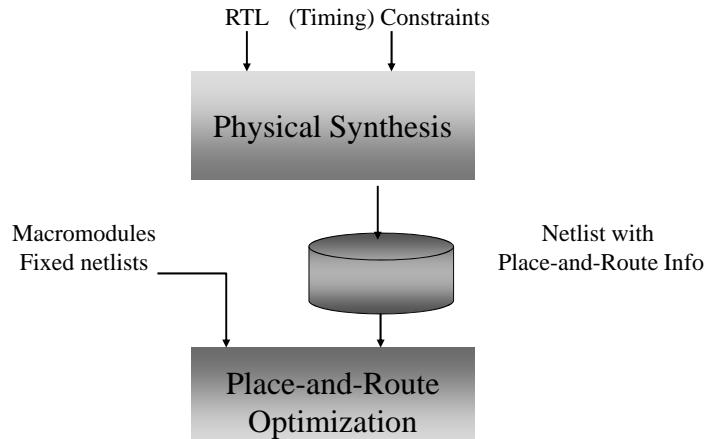
## The “Design Closure” Problem



Iterative Removal of Timing Violations (white lines)

Courtesy Synopsys

# Integrating Synthesis with Physical Design

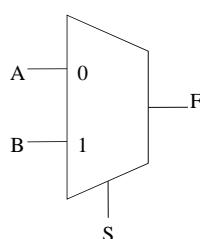


EECS 427 F09

Lecture 21 Artwork

5

## 2-input mux as programmable logic block



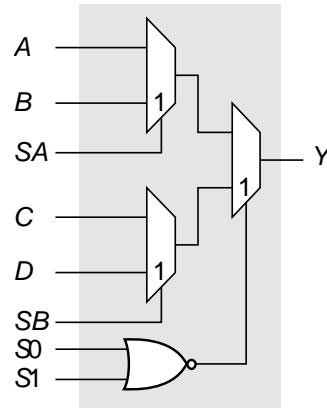
Configuration			
A	B	S	F =
0	0	0	0
0	X	1	X
0	Y	1	Y
0	Y	X	XY
X	0	Y	X̄Y
Y	0	X	X̄Y
Y	1	X	X 1 Y
1	0	X	X̄
1	0	Y	Ȳ
1	1	1	1

EECS 427 F09

Lecture 21

6

## Logic Cell of Actel Fuse-Based FPGA

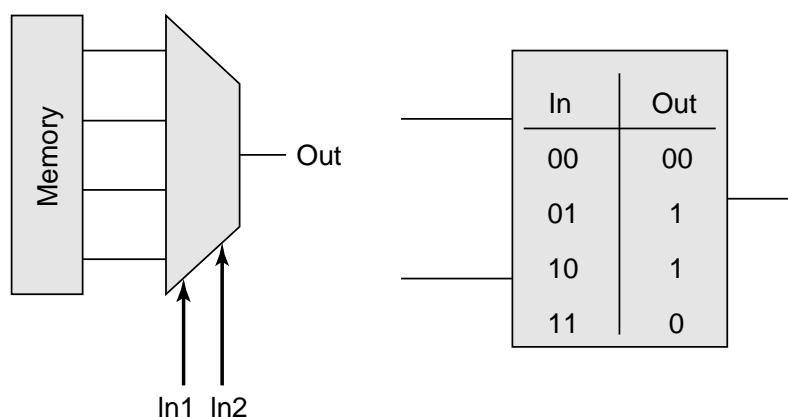


EECS 427 F09

Lecture 21

7

## Look-up Table Based Logic Cell

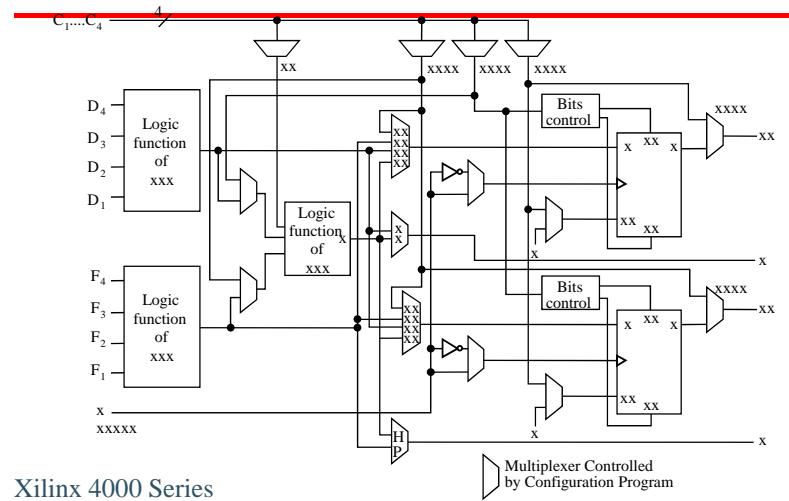


EECS 427 F09

Lecture 21

8

## LUT-Based Logic Cell

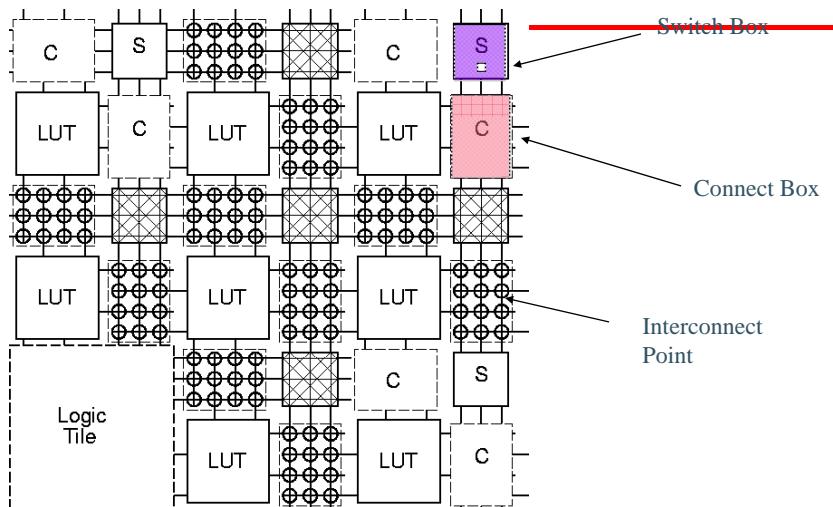


EECS 427 F09

Lecture 21  
Courtesy Xilinx

9

## Mesh-based Interconnect Network

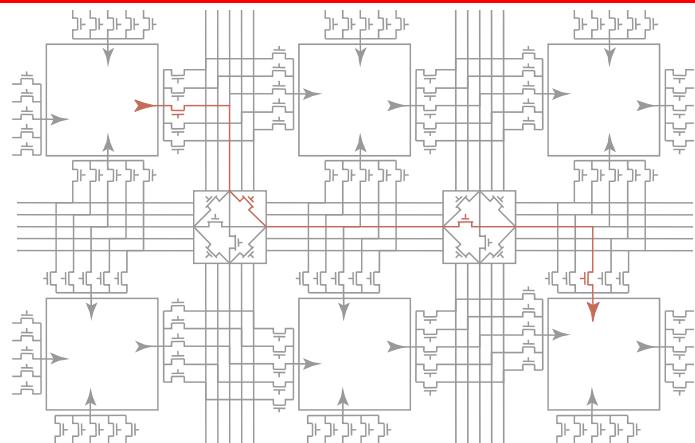


EECS 427 F09

Lecture 21  
Courtesy Dehon and Wawrzyniak

10

## Transistor Implementation of Mesh

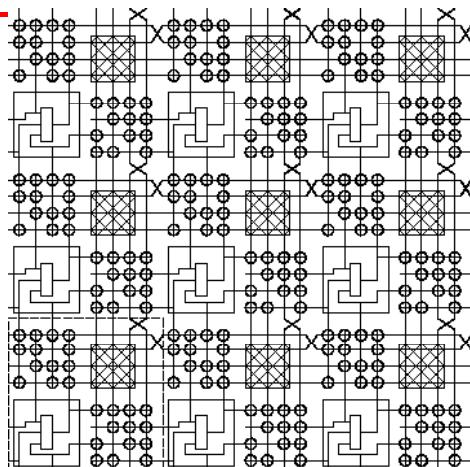


EECS 427 F09

Lecture 21  
Courtesy Dehon and Wawrzyniek

11

## Hierarchical Mesh Network



Use overlayed mesh  
to support longer connections

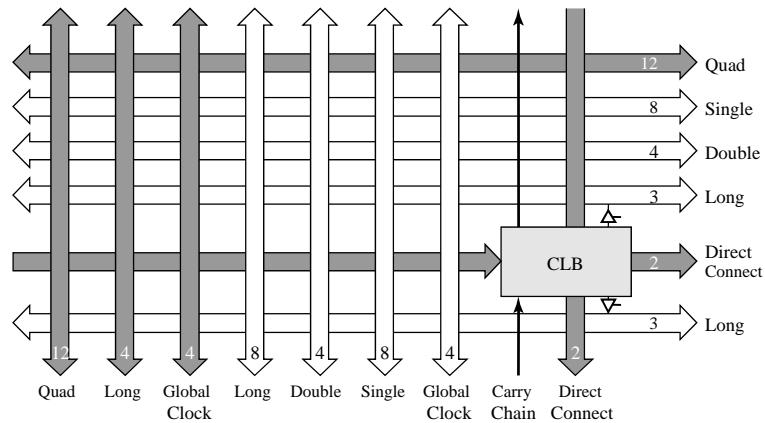
Reduced fanout and reduced  
resistance

EECS 427 F09

Lecture 21  
Courtesy Dehon and Wawrzyniek

12

## Xilinx 4000 Interconnect Architecture



EECS 427 F09

Lecture 21  
Courtesy Xilinx

13

## Testing is Expensive

- VLSI testers cost ~ \$5M
- Volume manufacturing requires large number of testers, maintenance
- Tester time costs are in ¢/sec
- Test cost contributes 20-30% to total chip cost
- The alternative:
  - \$1 to find a bad chip and toss it
  - \$10 to find a bad IC on circuit board
  - \$100 to find bad PC board in a system
  - \$1000 to find a bad component in a field system
  - \$1000000s to find a recurring bad part in a high-volume system (e.g., Intel floating point divide bug but this was a functional problem and not a manufacturing problem)

EECS 427 F09

Lecture 21

14

# Validation and Test of Manufactured Circuits

## Goals of Design-for-Test (DFT)

Make testing of manufactured part swift and comprehensive

## DFT Mantra

Provide controllability and observability

## Components of DFT strategy

- Provide circuitry to enable test
- Provide test patterns that guarantee reasonable coverage

EECS 427 F09

Lecture 21

15

# Test Classification

- Diagnostic test
  - Used in chip/board debugging, seeks to find location of faults
- “go/no go” or production test
  - Used in chip production
- Burn-in test
- Parametric test
  - Looks at continuous parameters, rather than discrete
  - Check parameters such as NM,  $T_{clk}$
  - Frequency binning (Intel, AMD) fits here

EECS 427 F09

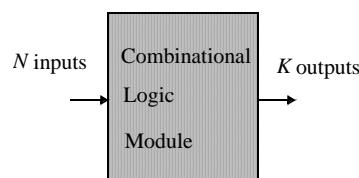
Lecture 21

16

## Burn-in or Stress test

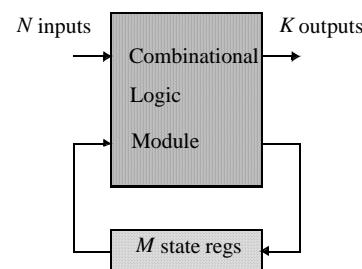
- Subject chips to high temperature and increased Vdd while running production tests
- Aimed to catch:
  - ‘Infant mortality’ cases; chips that would have failed quickly after shipping due to major defects

## Design for Testability



(a) Combinational function

$2^N$  patterns



(b) Sequential engine

$2^{N+M}$  patterns

**Exhaustive test is impossible or unpractical**

## Goals: Controllability/Observability

- Controllable: Can a specific node be brought to any desired state with only a single input vector?
- Observable: Can you monitor the node directly on output pins? Or do you have to wait many clock cycles?

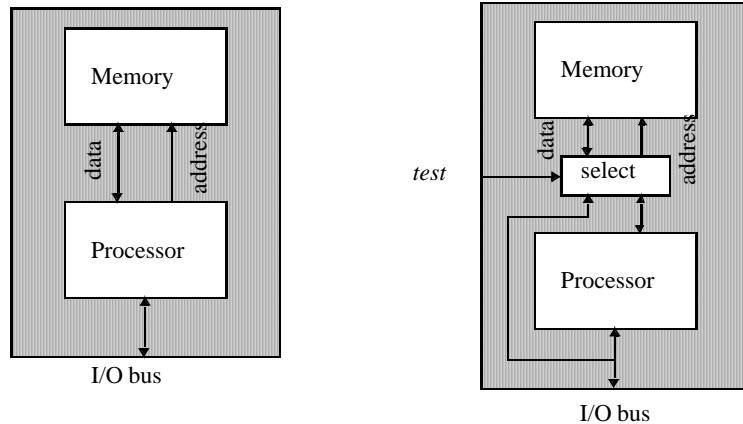
## Test Approaches

- Ad-hoc testing
- Scan-based Test
- Self-Test

Problem is getting harder

- Increasing complexity and heterogeneous combination of modules in system-on-a-chip
- Larger designs with more inputs mean that less of the design space can be searched

## Ad-hoc Test Example



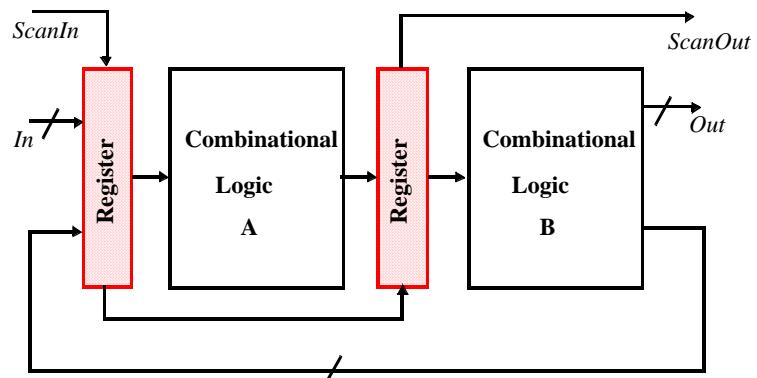
Inserting multiplexers improves testability at expense of additional hardware (and delay during normal operation)

## Scan-based Test

Make all registers externally loadable and readable

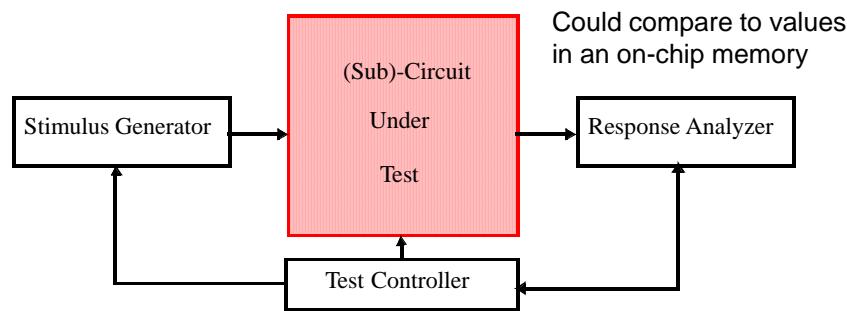
When testing, register 1 reads from ScanIn. Then block A executes and Register 2 outputs to ScanOut for comparison

At the same time, Register 1 reads in next test vector



22

## Self-test



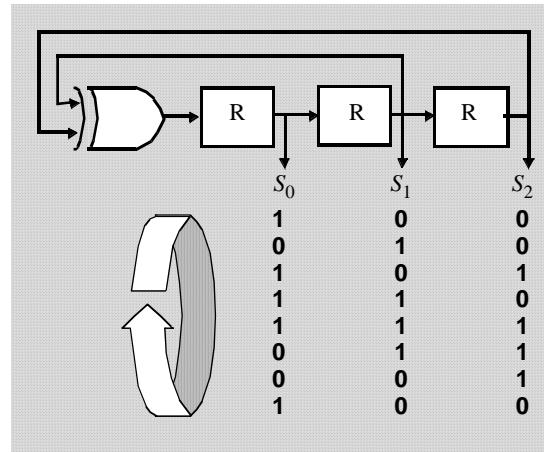
Rapidly becoming more important with increasing chip-complexity and larger modules

EECS 427 F09

Lecture 21

23

## Linear-Feedback Shift Register (LFSR)



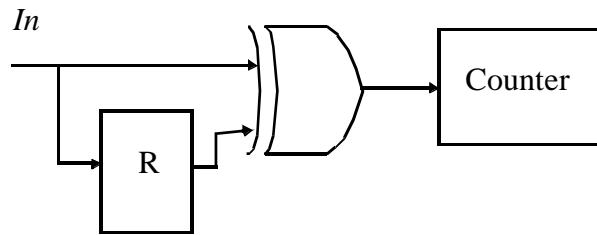
Pseudo-Random Pattern Generator

EECS 427 F09

Lecture 21

24

# Signature Analysis



Counts transitions on single-bit stream  
≡ Compression in time

Sort of a parity check – does not guarantee correctness

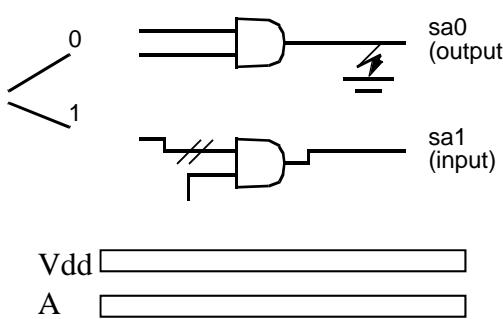
EECS 427 F09

Lecture 21

25

# Fault Models (H.4.1)

Most Popular – Single “Stuck-at” model



- Fault is permanent
- Effect of fault is that the faulty node is tied to either Vdd or ground
- Gate now functions improperly (which allows for observability)

EECS 427 F09

Lecture 21

26

## Pros/Cons of Stuck-at Fault Model

- Advantages:
  - Reasonable # of faults:  $2n$  where  $n$  is # of circuit nodes
  - Well-studied
  - ~90% of possible manufacturing defects are covered by this model
    - Source/drain shorts (see next slide), oxide pinholes, missing features, metallization shorts
- Disadvantages:
  - Does not cover all defects found in CMOS circuits

EECS 427 F09

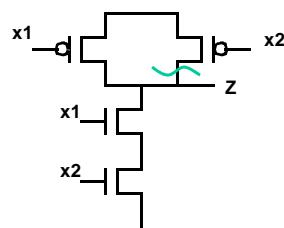
Lecture 21

27

## Problem with stuck-at model: CMOS open fault



$x_1$	$x_2$	$z$
0	x	1
1	1	0
1	0	$z_{n-1}$



**Sequential effect**

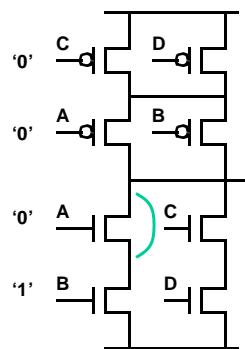
**Needs two vectors to ensure detection!**  
**Less controllable**

EECS 427 F09

Lecture 21

28

## Problem with stuck-at model: CMOS short fault

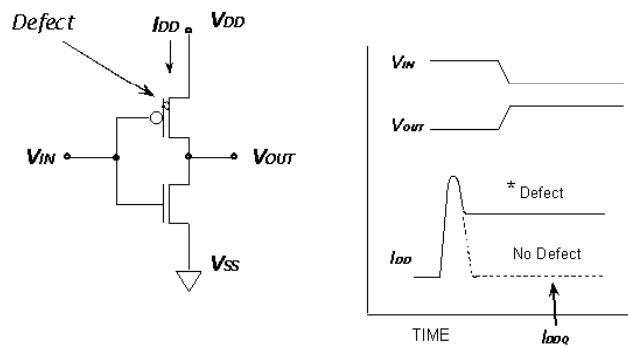


Causes short circuit between  
Vdd and GND for A=C=0, B=1  
  
Possible approach:  
Supply Current Measurement  
(IDDQ)

## IDDQ testing

- Physical defects often lead to large currents flowing, even when the circuit is supposedly in a quiescent state
  - Normally a quiet CMOS circuit will have very little current draw
    - Note this is becoming much less valid with rising leakage currents today, jeopardizing IDDQ testing
- By measuring the quiescent current from the supply voltage, we can assume that a very large value means an error/fault/defect

## Example, IDDQ testing



Gate-source short – when  $V_{in}$  is high, there is no current flow but when  $V_{in}$  goes low, current jumps

Gate still functions properly but you wouldn't want this to be shipped

EECS 427 F09

Lecture 21

Soden92 31

## Generating and Validating Test-Vectors

- Automatic test-pattern generation (ATPG)
  - for given fault, determine excitation vector (called **test vector**) that will propagate error to primary (observable) output
  - majority of available tools: combinational networks only
  - sequential ATPG available from academic research
- Fault simulation
  - determines **test coverage** of proposed test-vector set
  - simulates correct network in parallel with faulty networks
- Both require adequate models of faults in CMOS integrated circuits

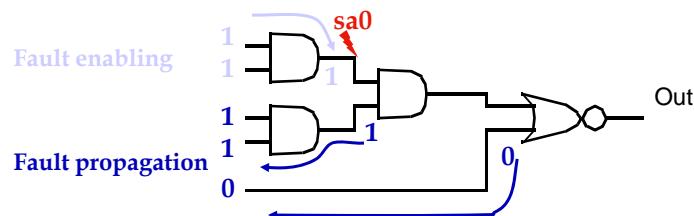
EECS 427 F09

Lecture 21

32

# Path Sensitization

Goals: Determine input pattern that makes a fault controllable  
(triggers the fault, and makes its impact visible at the output nodes)



Techniques Used: D-algorithm, Podem

# Summary

- Testing is an important part of designing integrated circuits
- Many engineers specialize in DFT techniques and are always in demand
- Fault models are abstractions of physical defects and are used to assess their impact on circuit behavior
  - Stuck-at 0/1 are most common
  - Test vectors can be created to determine whether a node is actually stuck at 0 or 1
- Key design for test techniques include:
  - Scan: load data into registers, run through logic, then scan out to compare to expected result
  - Self-test (or built-in self-test BIST): Incorporate everything on-chip which eases testing equipment requirements but requires lots of design effort