

ECE 3060
VLSI and Advanced Digital Design

Lecture 19

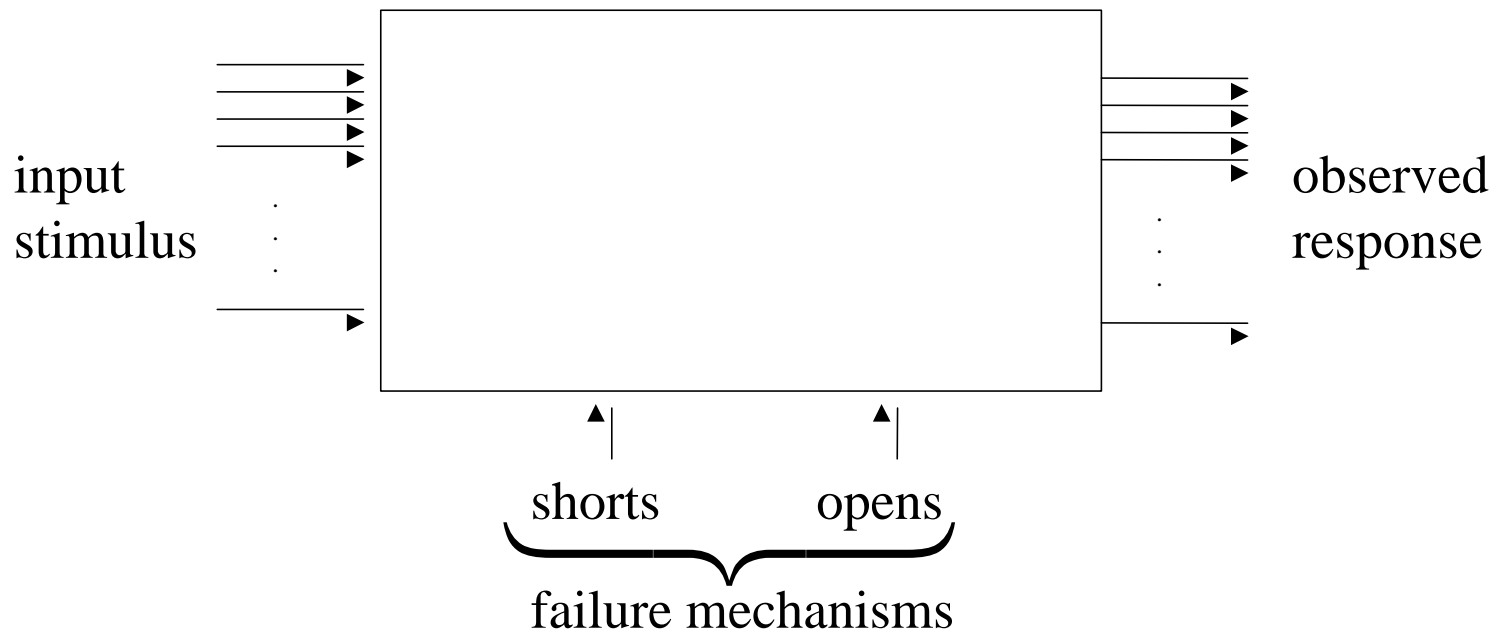
Testing

Test Cost


- A large percentage (5-40%) of the manufacturing cost of a VLSI chip is due to test
- Cost is larger for larger circuits
 - more functionality in a "System-on-a-Chip" means more potential faults
 - brute-force "stimulate all possible input vectors" can take centuries or more
- Mixed-technology test
 - analog, digital on same chip

Testing: Definition

- Testing = experiment in which a system is stimulated and its response analyzed in order to determine if the system will behave correctly under arbitrary input stimulus and prescribed failure mechanisms

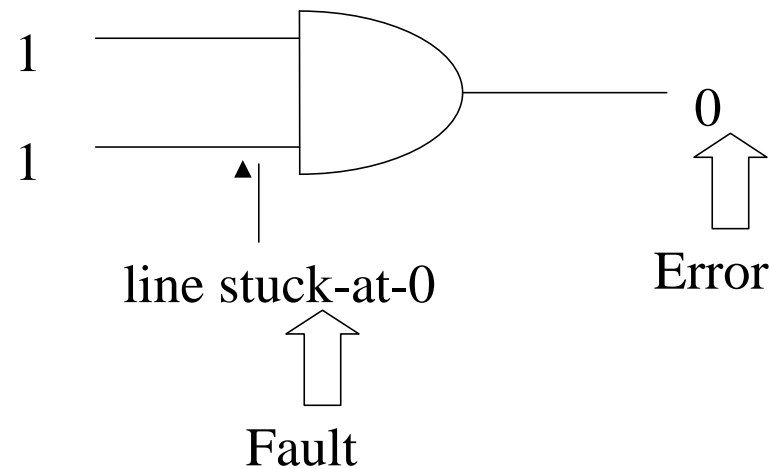


Failure mechanisms

- Failure mechanisms can be analyzed by looking at how they affect system behaviour at different levels of abstraction
 - messages → system level
 - programs/data structures → processor level
 - instructions/words → instruction set level
 - logic values/words → register level
 - logic values → logic level
- Control | Data
- 
- A diagram consisting of two vertical lines, one on the left and one on the right, with a vertical bar between them. The word "Control" is written to the left of the left vertical line, and the word "Data" is written to the right of the right vertical line. Above each vertical line, there is a small upward-pointing arrow.

Definitions

- Fault = Defect = Physical failure
 - e.g., shorts, opens, near-open, near-short
- Error = effect of fault, i.e., incorrect logic value due to fault



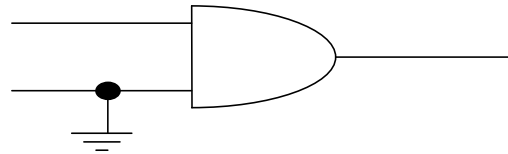
Source of Errors

- Note that errors may be due to design errors or fabrication errors
- Design errors:
 - incomplete or inconsistent specification
 - incorrect mappings between different levels of design
 - violations of design rules
- Fabrication errors:
 - wrong components
 - incorrect wiring
 - shorts caused by improper soldering

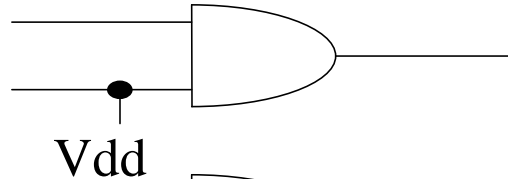
Fault Model

- Model describes nature of the fault

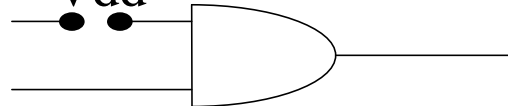
- stuck at 0



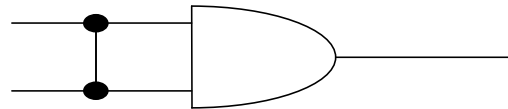
- stuck at 1



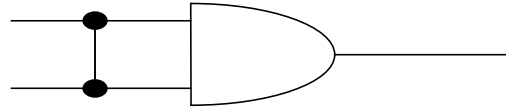
- stuck open



- stuck closed



Stuck Closed: Wired-AND or Wired-OR



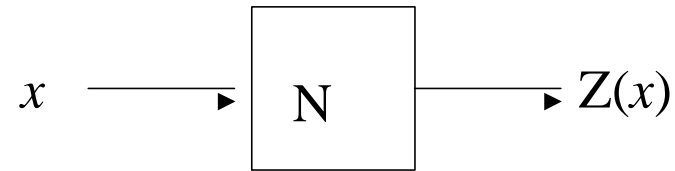
- **Wired-AND**

a	b	c	d	output
0	0	0	0	0
0	1	0	0	0
1	0	0	0	0
1	1	1	1	1

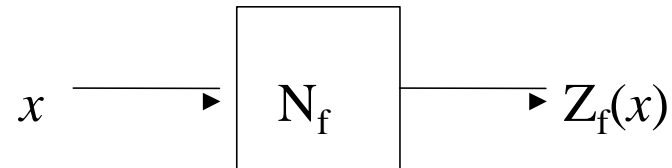
- **Wired-OR**

a	b	c	d	output
0	0	0	0	0
0	1	1	1	1
1	0	1	1	1
1	1	1	1	1

Fault Detection

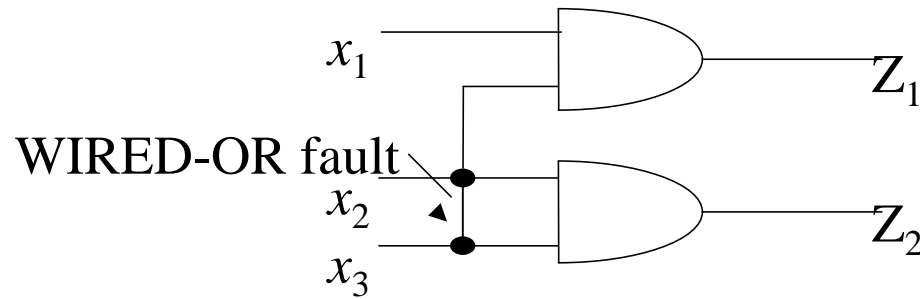


$x = \text{input}$ $Z(t) = \text{response}$



- Test set = $\{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_m\}$
 - $Z(\mathbf{t}_1), Z(\mathbf{t}_2), \dots, Z(\mathbf{t}_m)$
 - $Z_f(\mathbf{t}_1), Z_f(\mathbf{t}_2), \dots, Z_f(\mathbf{t}_m)$
- Definition:
 - \mathbf{t} detects f iff $Z_f(\mathbf{t}) \neq Z(\mathbf{t})$

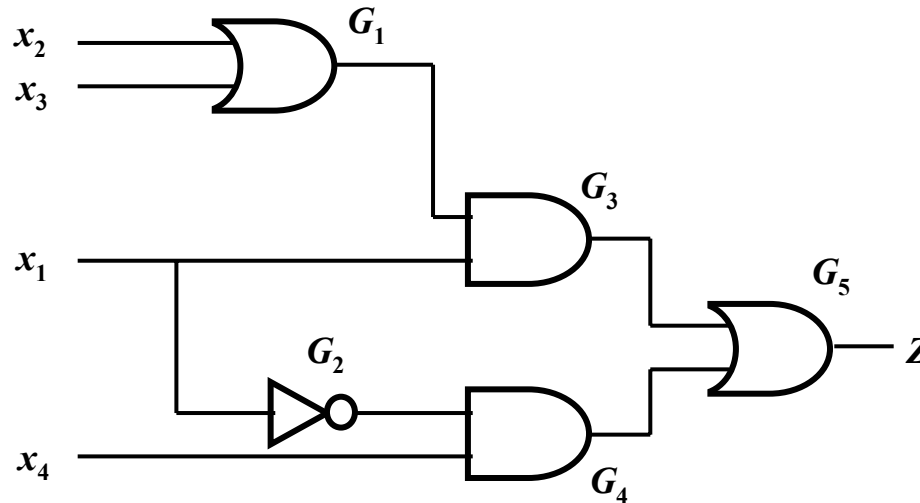
Test Set for Fault f



- $Z_1 = x_1 \cdot x_2$, $Z_2 = x_2 \cdot x_3$
- $Z_{1f} = x_1 \cdot (x_2 + x_3)$, $Z_{2f} = x_2 + x_3$
- $\mathbf{t} = \{0, 1, 1\}$ detects f
- Set of all tests that detect f (single out) is given by
 - $Z(\mathbf{x}) \oplus Z_f(\mathbf{x}) = 1$

Examples

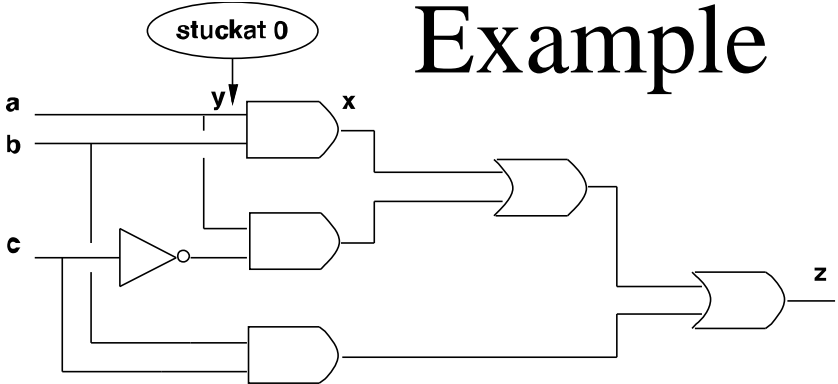
- $f_1 = x_4$ s-a-0 $\rightarrow x_1'x_4=1$
- Simulate 1001 with $f_2 = G_2$ s-a-1: path sensitization



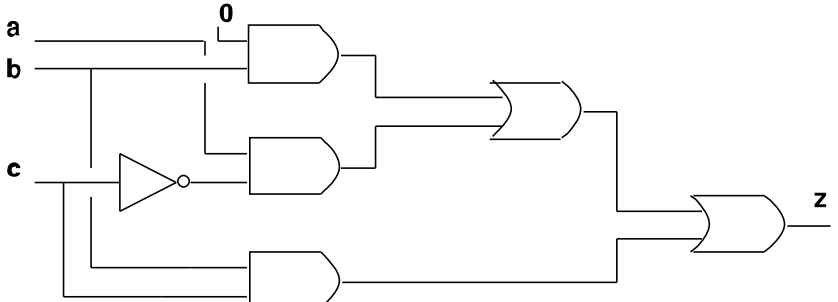
Synthesis and Testability

- Assumptions
 - combinational circuit
 - single or multiple *stuck-at* faults
- Redundancy removal
 - use tests to search for untestable faults and reduce circuit
- Algorithm:
 - if stuck-at-0 on net y is untestable
 - set $y = 0$
 - propagate constant, reducing logic
 - if stuck-at-1 on net y is untestable
 - set $y = 1$
 - propagate constant, reducing logic

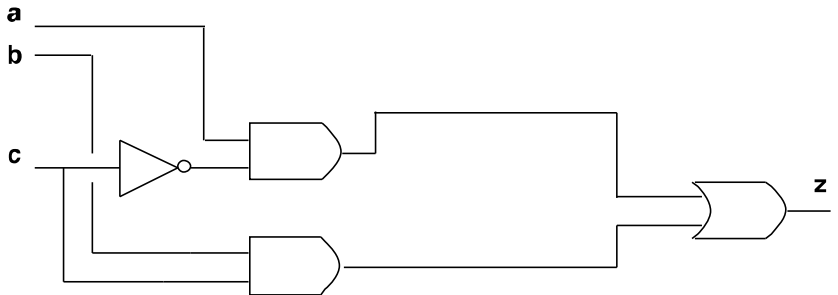
Example



(a)



(b)

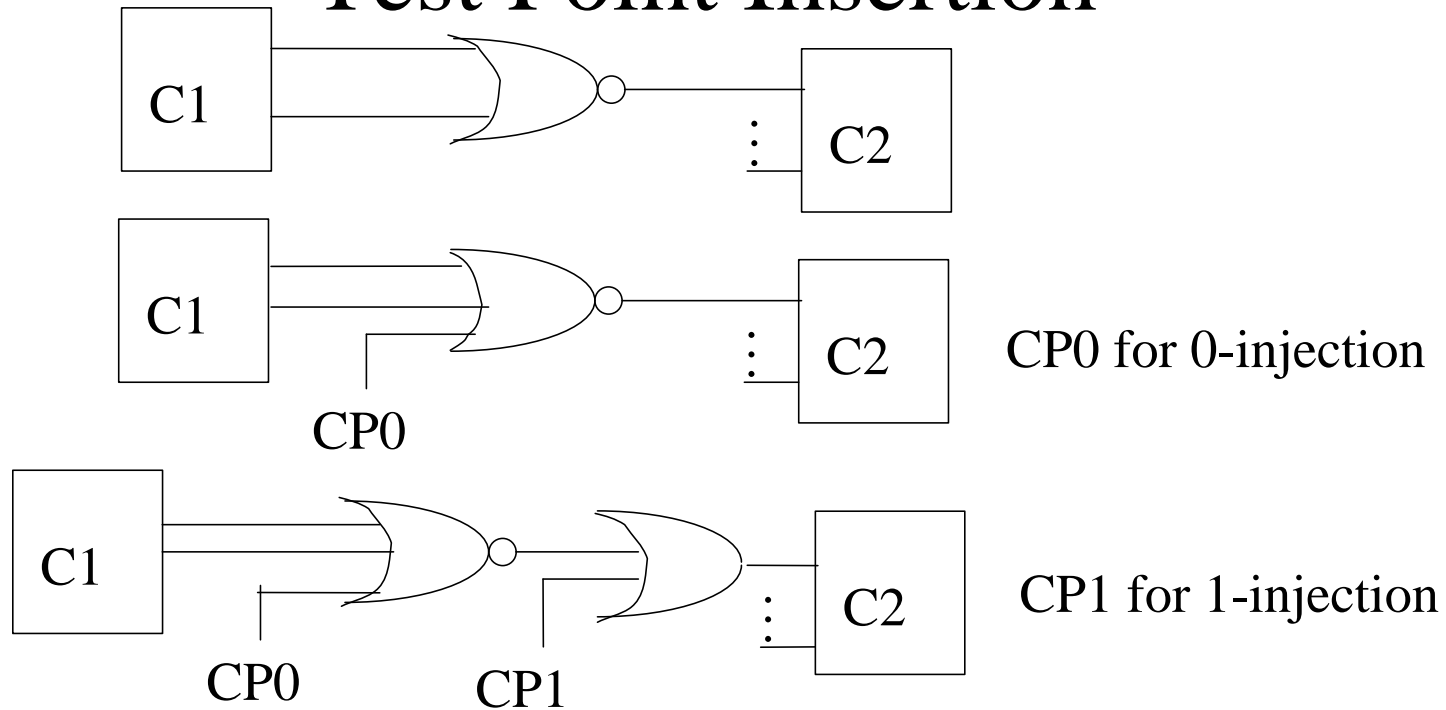


(c)

Design for Testability

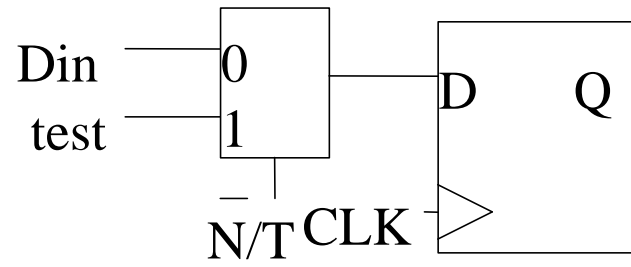
- Status (normal, inoperable, degraded) of a device to be determined
- DFT allows cost-effective development of tests to determine status
- Controllability
 - setting net y to a 1 (1-controllability)
 - setting net y to a 0 (0-controllability)
- Observability
 - ability to drive an error from net y to a primary output

Test Point Insertion

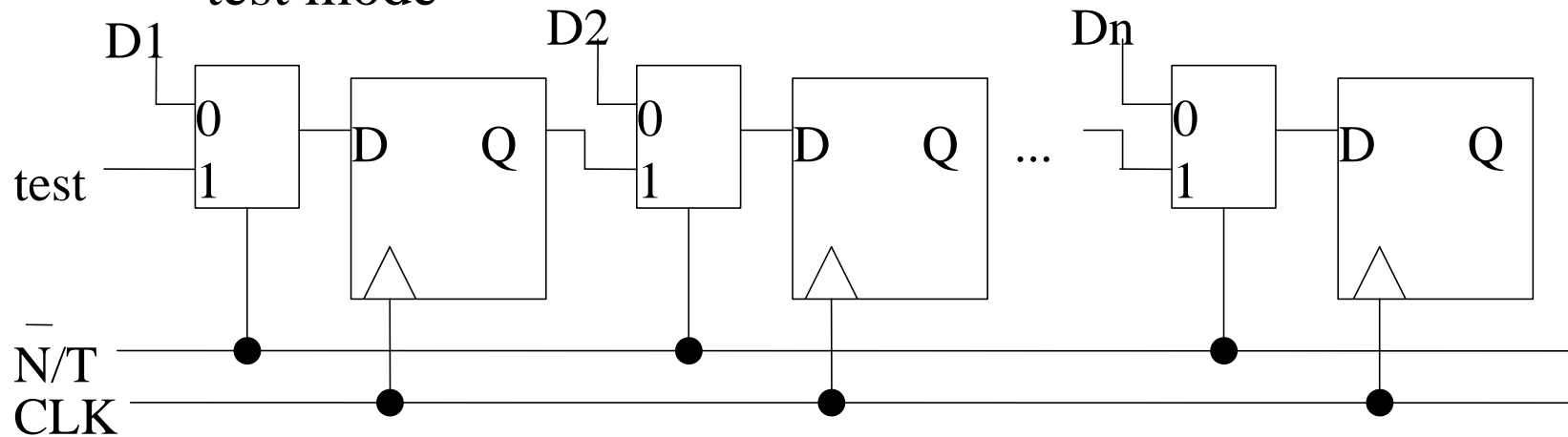


- Design circuits to be initializable
- Partition large circuits into smaller subcircuits in order to reduce test cost

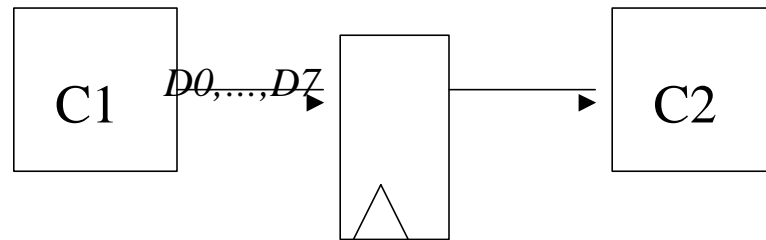
Scan Registers



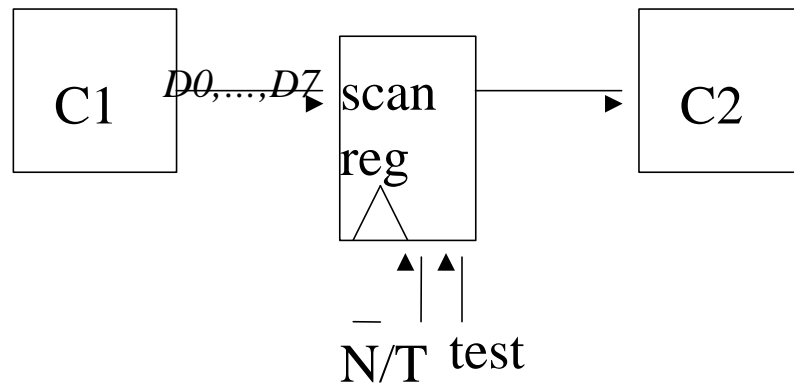
- $\overline{N/T} = 0 \Rightarrow$ data loaded from data line Din
– normal mode
- $\overline{N/T} = 1 \Rightarrow$ data loaded from test input
– test mode



Scan Register Usage

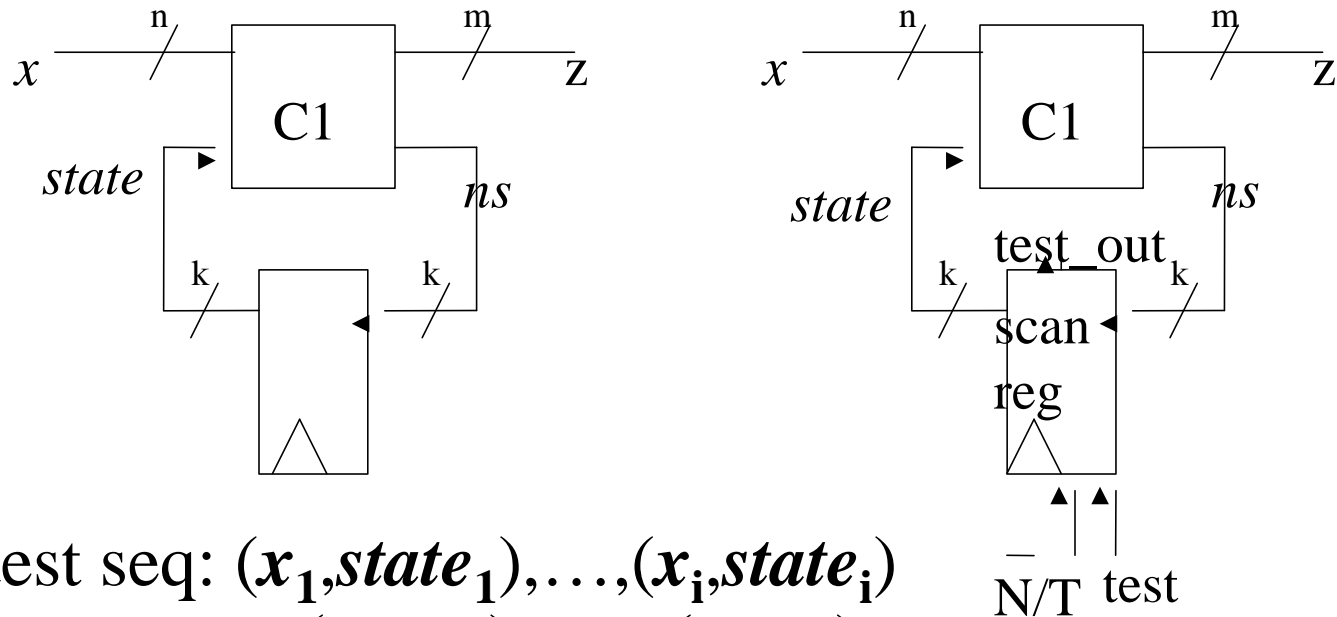


replace with



- $\overline{N/T} = 0 \Rightarrow$ normal mode
- $\overline{N/T} = 1 \Rightarrow$ data loaded serially from test input

Fully Integrated Serial Scan



- test seq: $(x_1, state_1), \dots, (x_i, state_i)$
- responses: $(z_1, ns_1), \dots, (z_i, ns_i)$
- set $N/T=1$, scan $state_1$ into scan reg.
- Apply x_1 to input, $N/T = 0$
- next clock edge latches ns_1
- set $N/T = 1$, scan out ns_1 while scanning in x_2
- repeat i times