Design for Testability

- Testability Measurement
- DFT Basics
- DFT Techniques
  - ad hoc
  - Scan design
  - Boundary scan

Testability

- Controllability: The ability to set some circuit nodes to a certain states or logic values.
- Observability: The ability to observe the state or logic values of internal nodes.

Usage of Testability Measures

- Speed up test generation
- Improve the design testability
- Guide the DFT insertion

Testability Measurement

- TMEAS [Stephenson & Grason, 1976]
- SCOAP [Goldstein, 1979]
- TESTSCREEN [Kovijanic 1979]
- CAMELOT [Bennetts *et al.*, 1980]
- VICTOR [Ratiu *et al.*, 1982]

SCOAP

- Sandia Controllability Observability Analysis Program.
- Using integers to reflect the difficulty of controlling and observing the internal nodes.
- Higher numbers indicate more difficult to control or observe.
- Applicable to both combinational & sequential circuits.

Measurement in SCOAP

- For a node A:

  $CC^0(A)$: Combinational 0-controllability

  $CC^1(A)$: Combinational 1-controllability

  $SC^0(A)$: Sequential 0-controllability

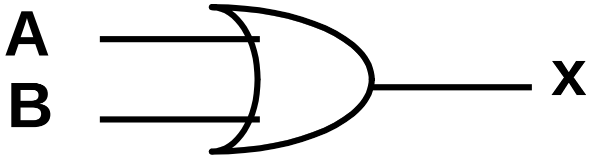
  $SC^1(A)$: Sequential 1-controllability

  $CO(A)$: Combinational observability

  $SO(A)$: Sequential observability

Combinational Components in SCOAP

Ex:



$CC^0(x) = CC^0(A) + CC^0(B) + 1;$

$CC^1(x) = \min\{CC^1(A), CC^1(B)\} + 1.$

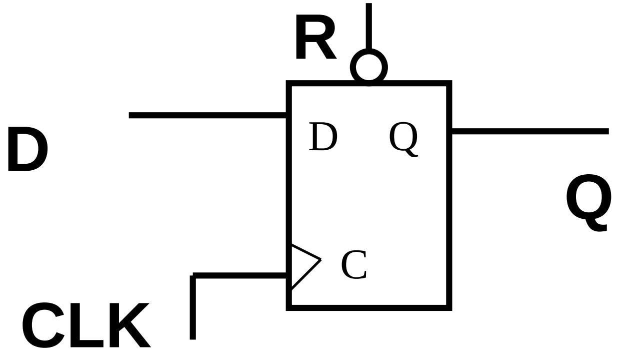
$SC^0(x) = SC^0(A) + SC^0(B)\ ;$

$SC^1(x) = \min\{SC^1(A), SC^1(B)\}.$

- CC implies the distance from PI
- SC implies the number of time frames needed to provide a 0 or 1.

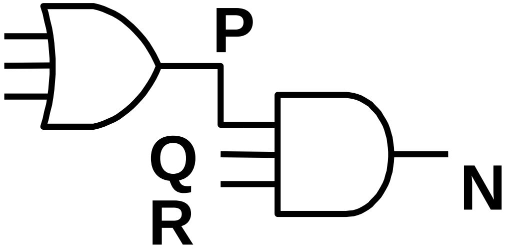
Sequential Components in SCOAP

Ex:

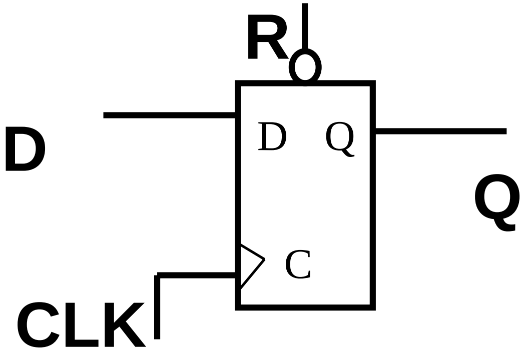


- $CC^0(Q)$ = min{$CC^0(R)$, $CC1(R) + CC^0(D) + CC^0(C) + CC^1(C)$}
- $CC^1(Q) = CC^1(R) + CC^1(D) + CC^0(C) + CC^1(C)$
- $SC^0(Q)$ = min{$SC^0(R)$, $SC^1(R) + SC^0(D) + SC^0(C) + SC^1(C)$} + 1
- $SC^1(Q) = SC^1(R) + SC^1(D) + SC^0(C) + SC^1(C) + 1$

Observalibity in SCOAP



- $CO(P) = CO(N) + CC^1(Q) + CC^1(R) + 1$
- $SO(P) = SO(N) + SC^1(Q) + SC^1(R)$



- $CO(R) = CO(Q) + CC^1(Q) + CC^0(R)$
- $SO(R) = SO(Q) + SC1(Q) + SC^0(R) +1$

Initial States

- PI: $CC^0 = CC^1 = SC^0 = SC^1 = 1$
- PO: $CO = SO = 0$
- All other numbers are initially set to ∞

Importance of Testability Measures

- They can guide the designers to improve the testability of their circuits.
- Test generation algorithms using heuristics usually apply some kind of testability measures to their heuristic operations (e.g., in making search decisions), which greatly speed up the test generation process.

Design for Testability (DFT)

- Test Costs:
  - Test Generation
  - Fault Simulation
  - Fault Location
  - Test Equipment
  - Test Application Time
- Test Difficulties:
  - Sequential > Combinational
  - Control Logic > Data Path
  - Random Logic > Structured Logic
  - Asynchronous > Synchronous
- Testability
  - Controllability
  - Observability

Design for Testability (DFT)

- DFT techniques are design efforts specifically employed to ensure that a device in testable.
- In general, DFT is achieved by employing extra H/W.

  ⇒Conflict between design engineers and test engineers.

  ⇒ Balanced between amount of DFT and gain achieved.
- Examples:
  - DFT

    ⇒Area ↑ & Logic complexity ↑

    ⇒Yield ↓

    ⇒For fixed fault coverage, defect level ↑
  - Therefore, DFT must guarantee to increase fault coverage.

Benefits of DFT

- In general, DFT has the following benefits:
  - Fault coverage ↑
  - Test generation (development) time ↓
  - Test length
  - Test Memory } hope ↓
  - Test application time
  - Support a test hierarchy {
    - Chips
    - Boards
    - Subsystems
    - Systems
  - Concurrent engineering
  - Reduce life-cycle costs

⇒ Pay less now and pay more later without DFT!

Costs Associated with DFTs

- Pin Overhead
- Area / Yield
- Performance degradation
- Design Time

⇒ There is no free lunch !

DFT Techniques

- ad hoc DFT technology
- Scan-based design
- Boundary Scan

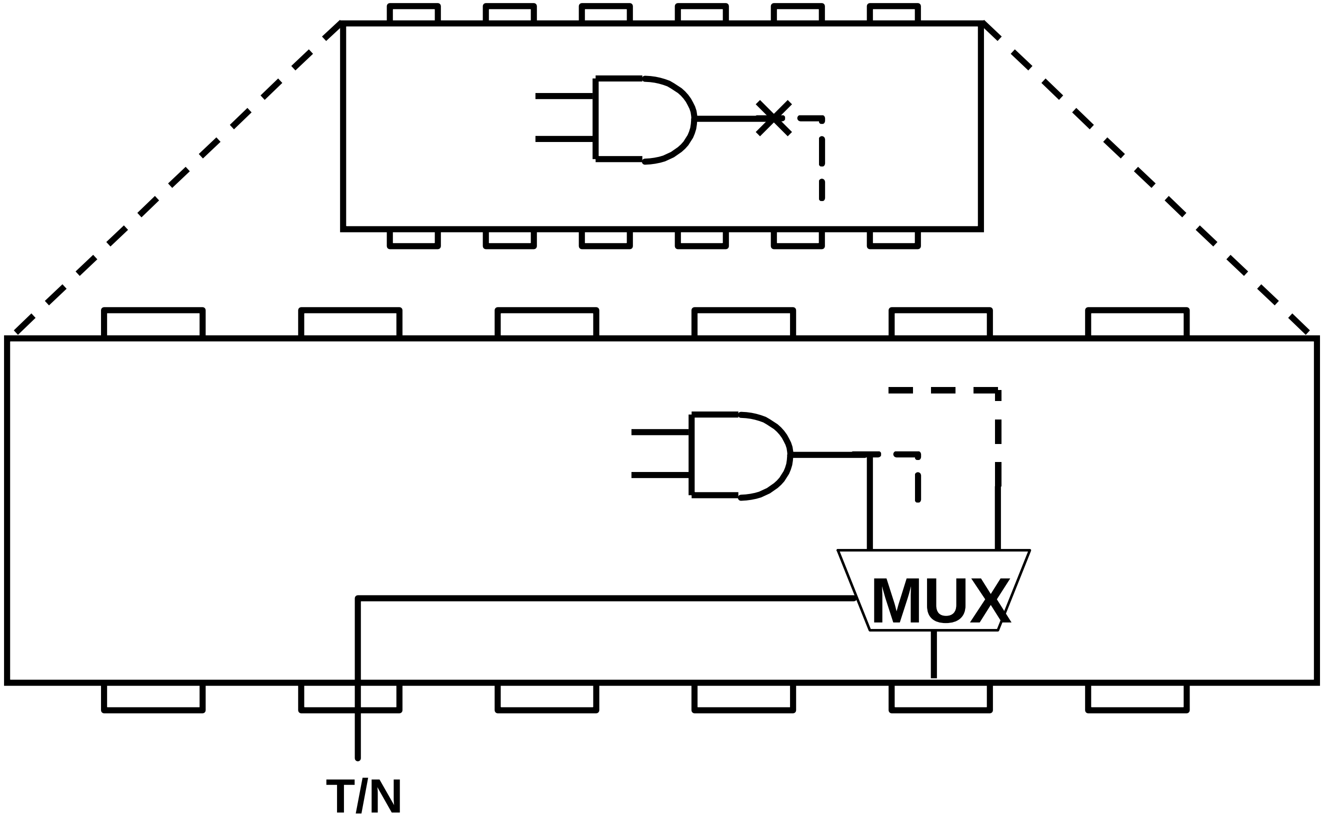
Ad hoc Techniques

- Test points
- Initialization
- Monostable multivibrators (one shot)
- Oscillators and clocks
- Counter / Shift registers
- Partitioning large circuits
- Logic redundancy
- Break global feedback paths

⋮

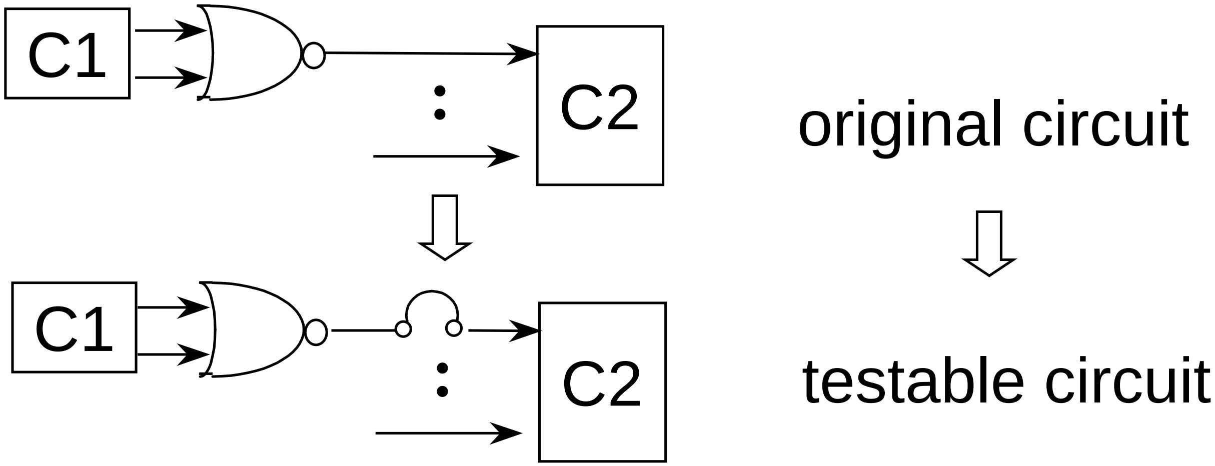
Example of *ad hoc* Techniques

- Insert test point



Test Points

- **Rule : to enhance controllability and observability by inserting control points (cp) and observation points (op), respectively.**
- **Ex:**



original circuit

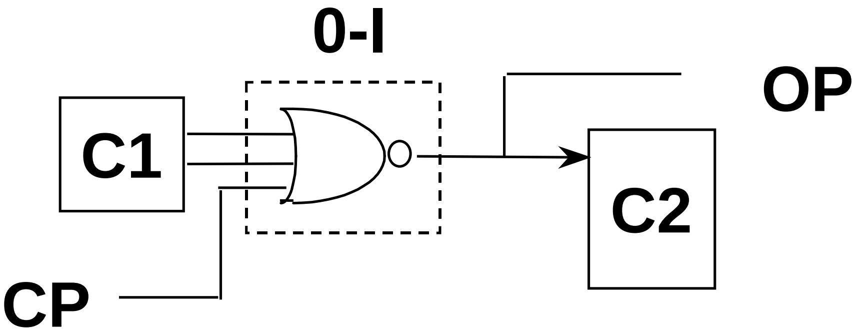
⇩

testable circuit

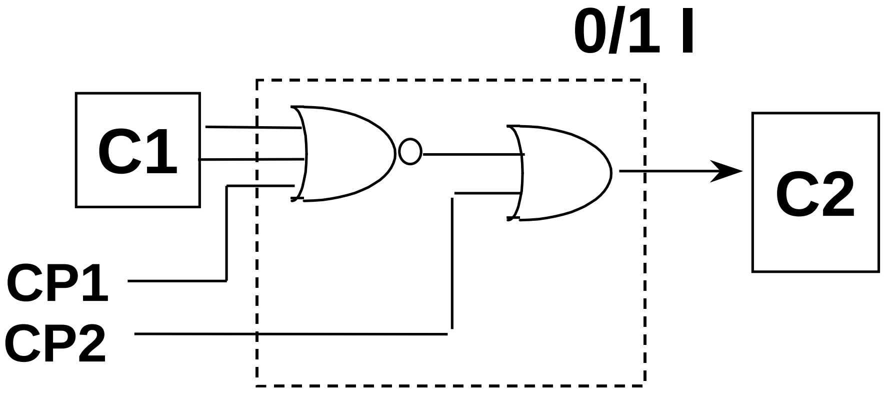
⇨ **can be done only for board**

Test Points (Cont.)

- Using a CP for 0-injection and an OP for observability:

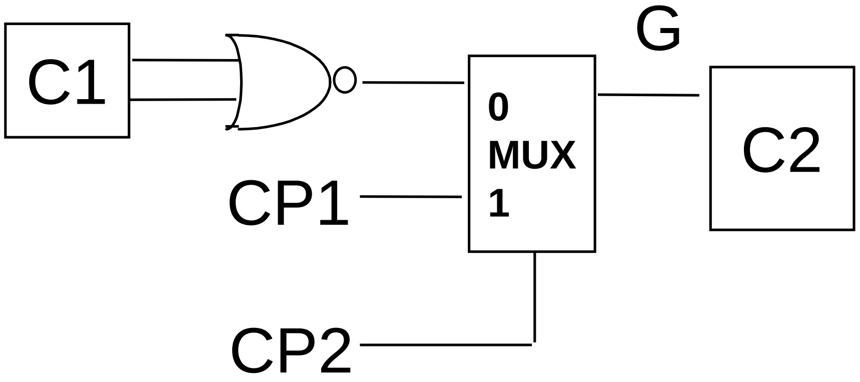


- 0/1 Injection:



Test Points (Cont.)

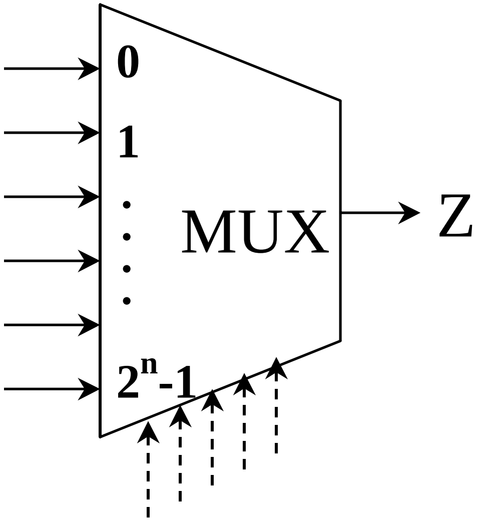
- **Using a MUX**



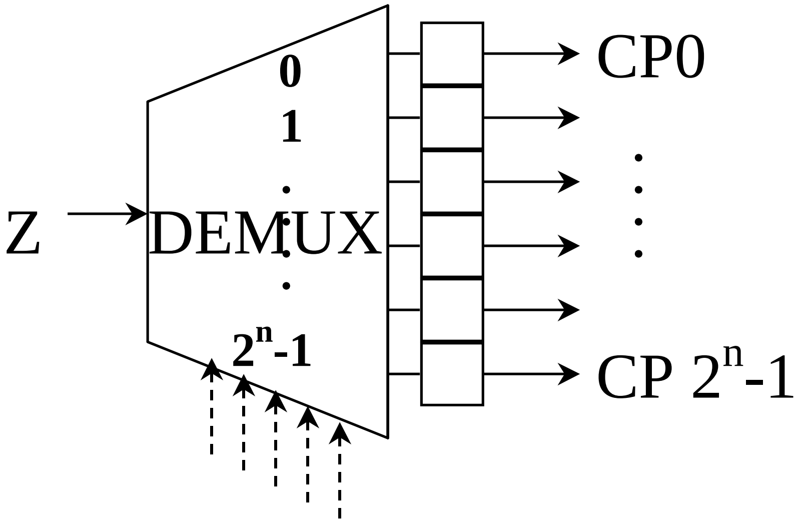
CP2=0 : normal
CP2=1: G=1 if CP1=1
G=0 if CP2=0

Test Points (Cont.)

- **Multiplexing Observation Points:**



- **Demultiplexing and Latching Control Points:**



Selection of CP

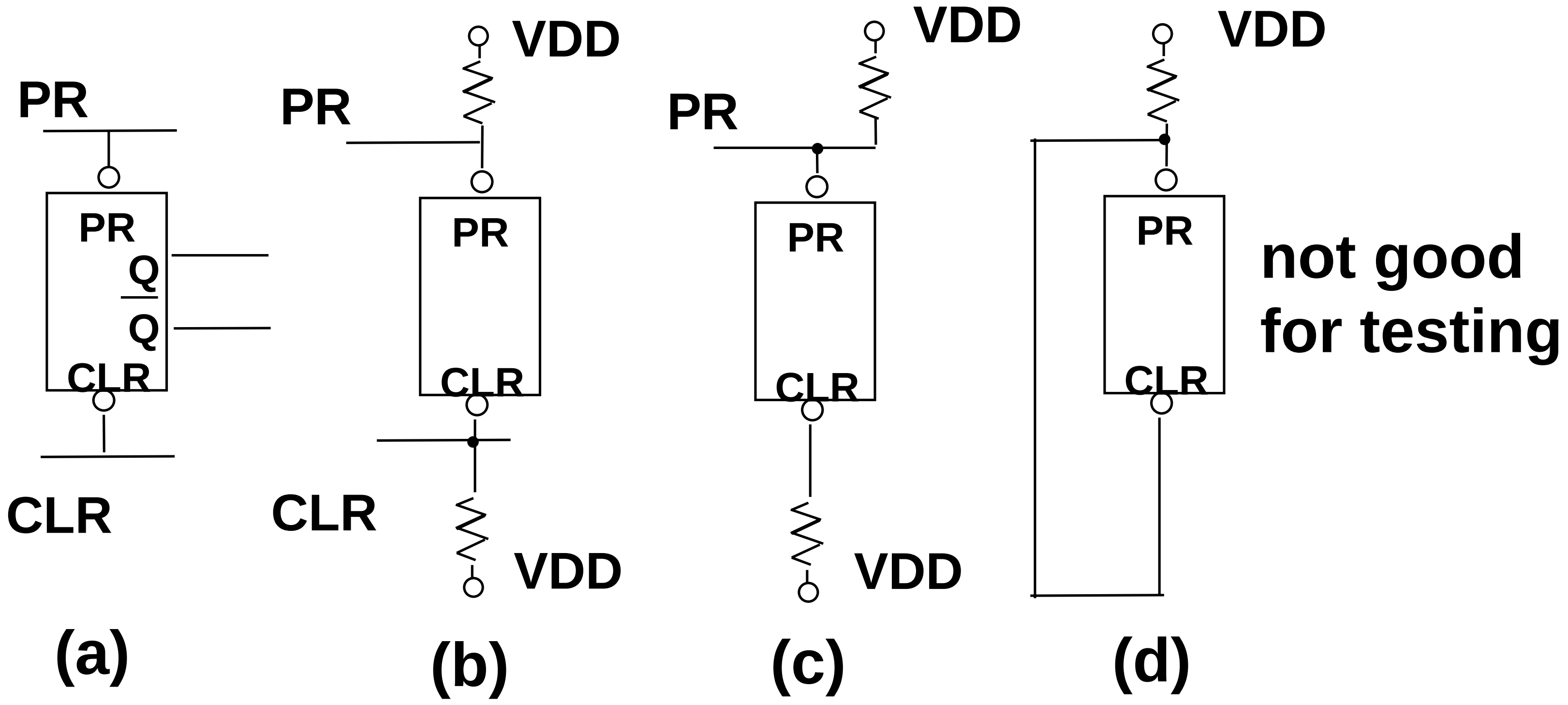
- Control, address and data bus lines on bus-structured designs.
- Enable/hold inputs to microprocessors.
- Enable and Read/write inputs to memory.
- Clock and preset/reset inputs to F/Fs, counters, shift registers, etc.
- Data select inputs to multiplexers and demultiplexers.
- Control lines on tri-state devices.

Selection of OP

- Stem lines with high fanout.
- Global feedback path
- Redundant signal lines
- Outputs of devices with many inputs, e.g., multiplexers and parity generators.
- Outputs from state devices.
- Address, control, data buses

Initialization

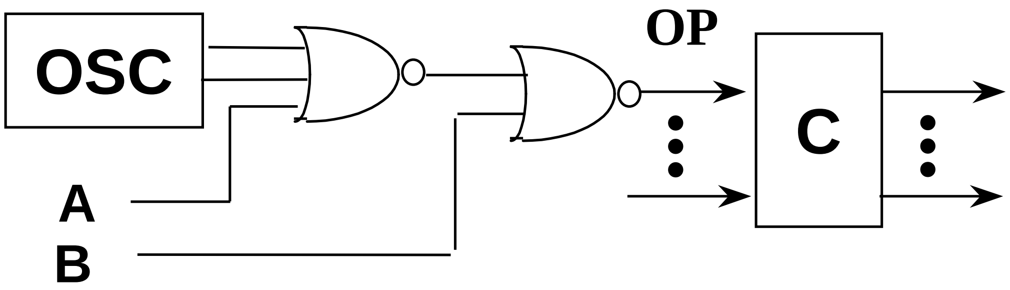
- Rule: Design circuits to be easily initialized
  - Don't disable preset and clear lines



not good for testing

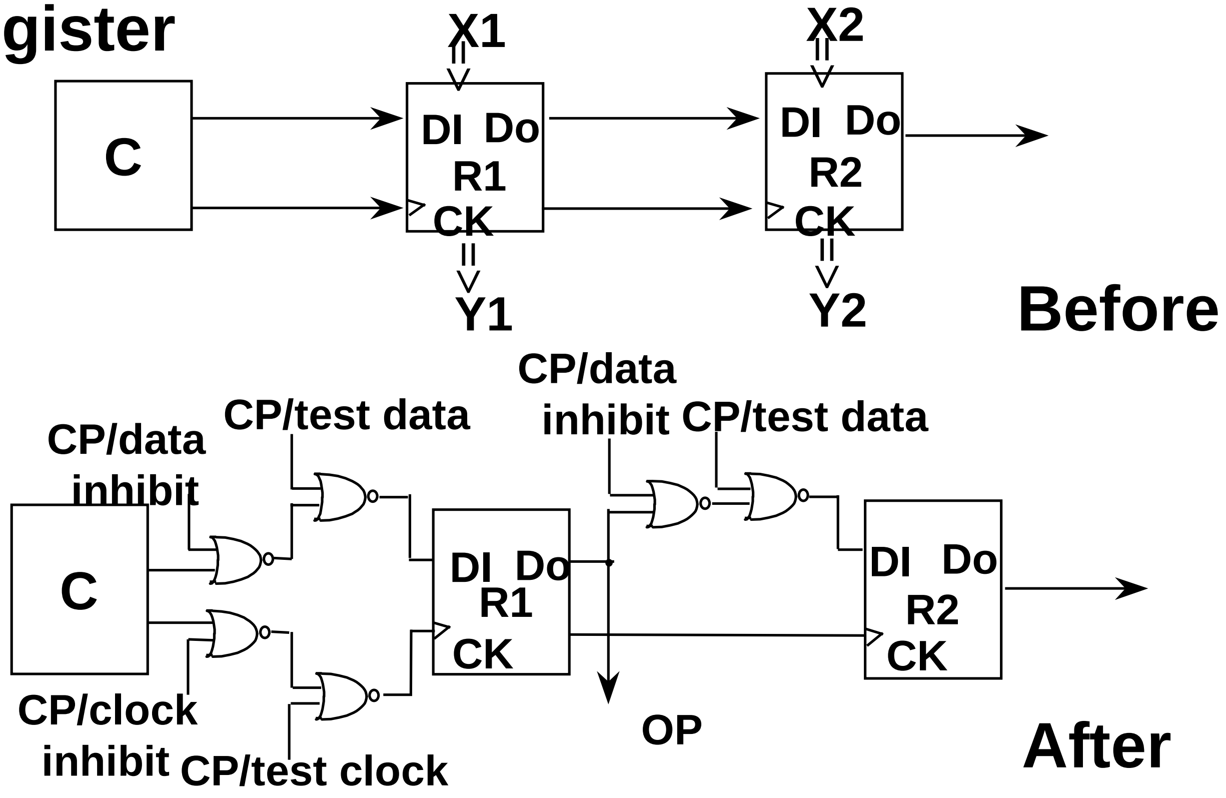
Monostable Multivibrator, Oscillators and Clocks

- Rule: Disable internal one shot, OSC and clocks
  - inserting CP and/or OP while disabling these devices
- Example:



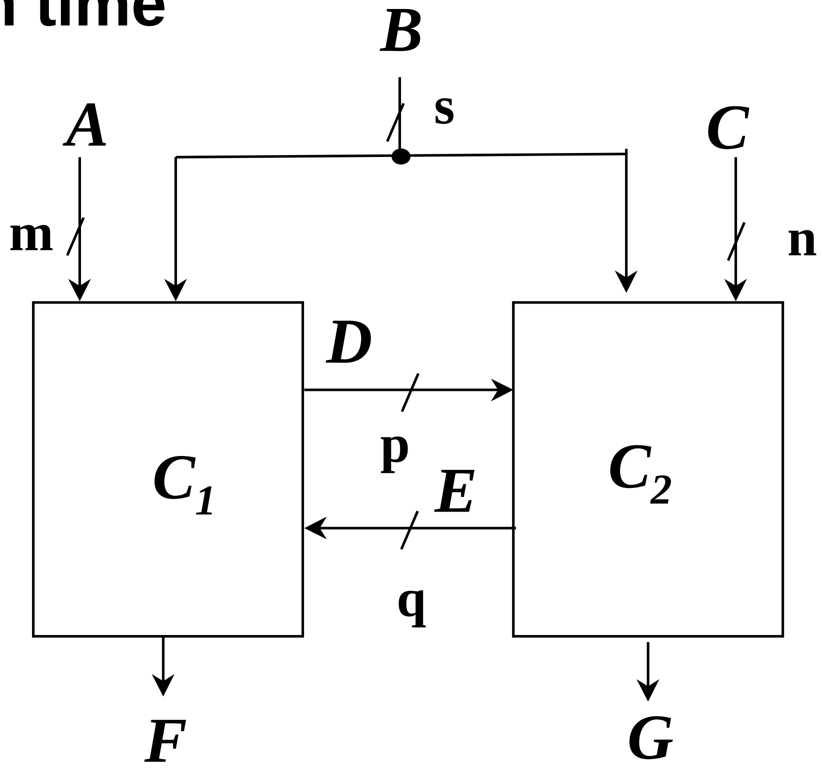
Partitioning Counters and Shift Registers

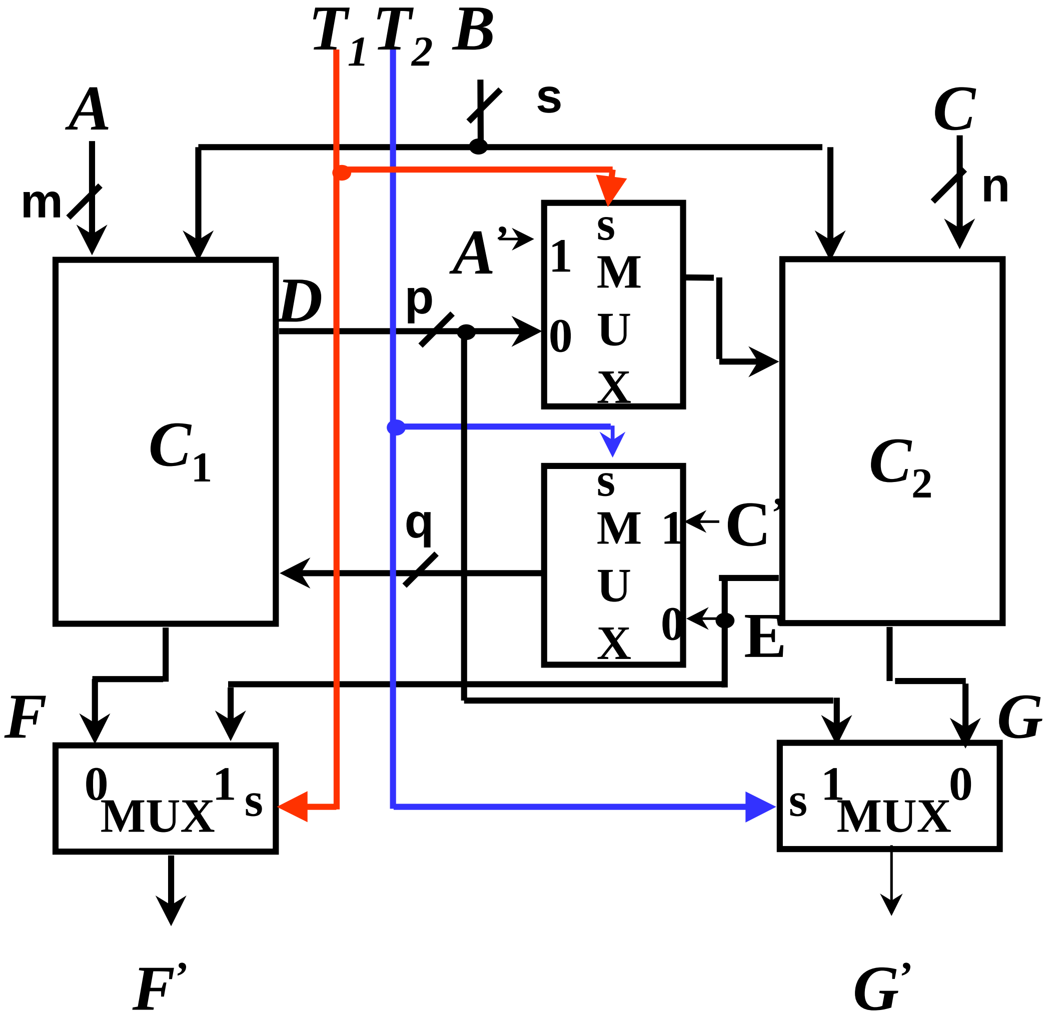
- Rule: Partition into small units
- Ex: Register



Partition of Large Combinational Circuits

- **Rule : To reduce test generation costs and/or test application time**





T_1	T_2	Mode
0	0	normal
0	1	test C_1
1	0	test C_2

If $2^{p+n} + 2^{q+m} < 2^{m+n}$ then test time can be reduced

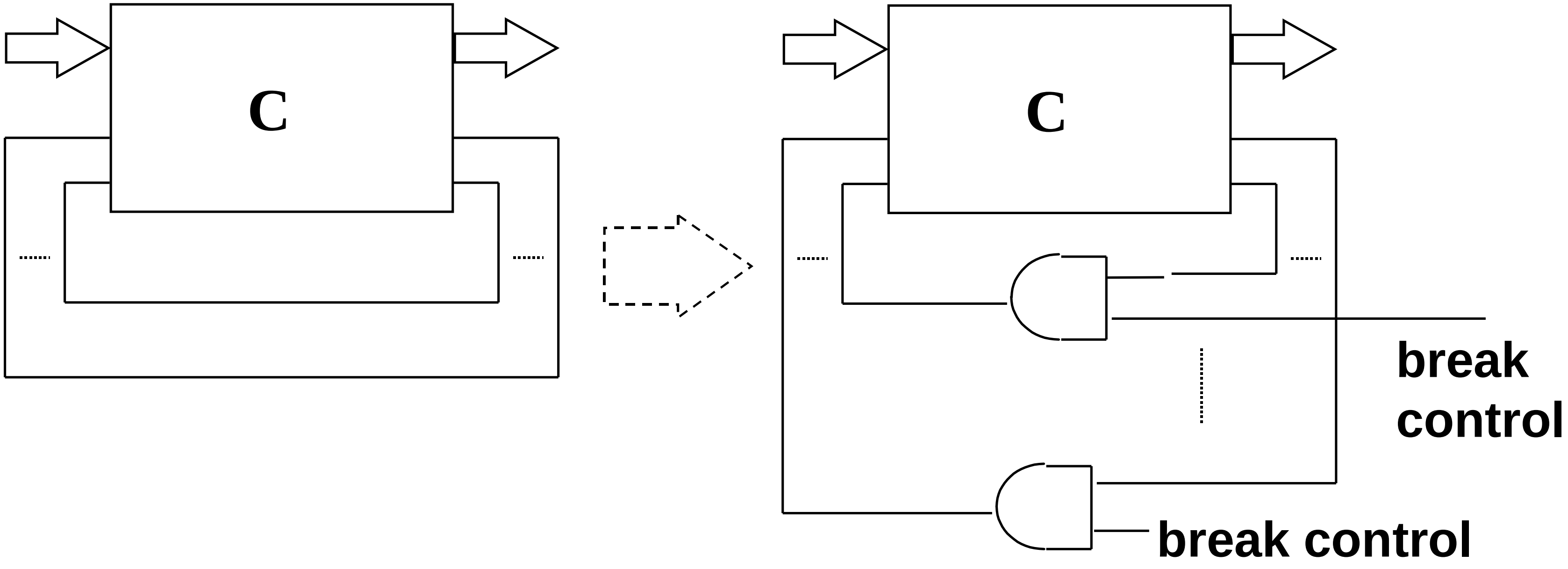
Logic Redundancy

Rule: Avoid or eliminate redundancy ckt.

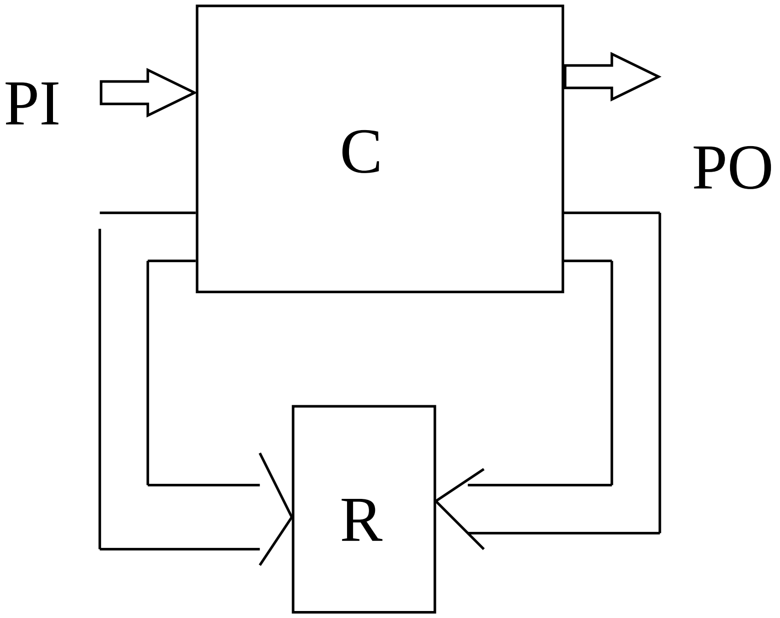
- Design errors
- Undetectable faults
- Invalidation of some tests
- Bias fault coverage

Global Feedback Paths

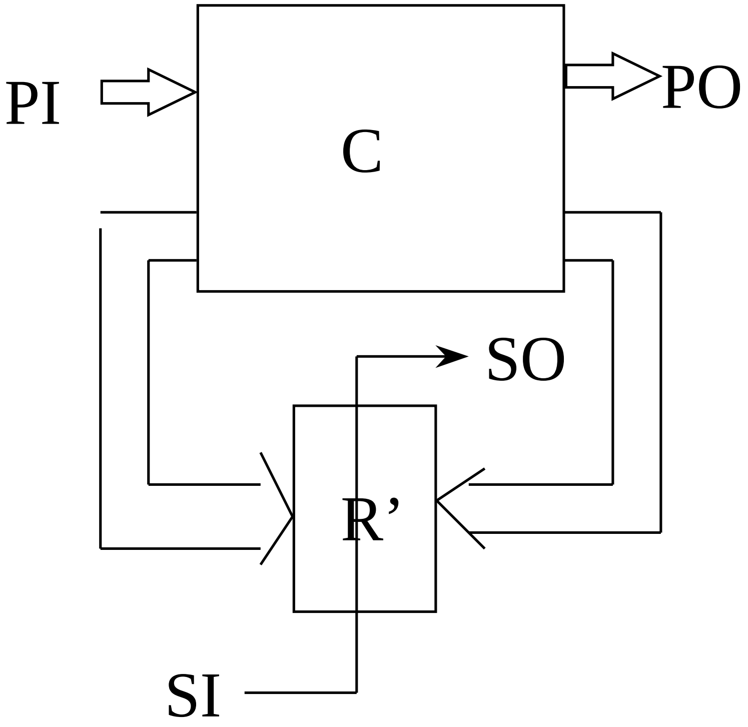
Rule: break global feedback



Scan System

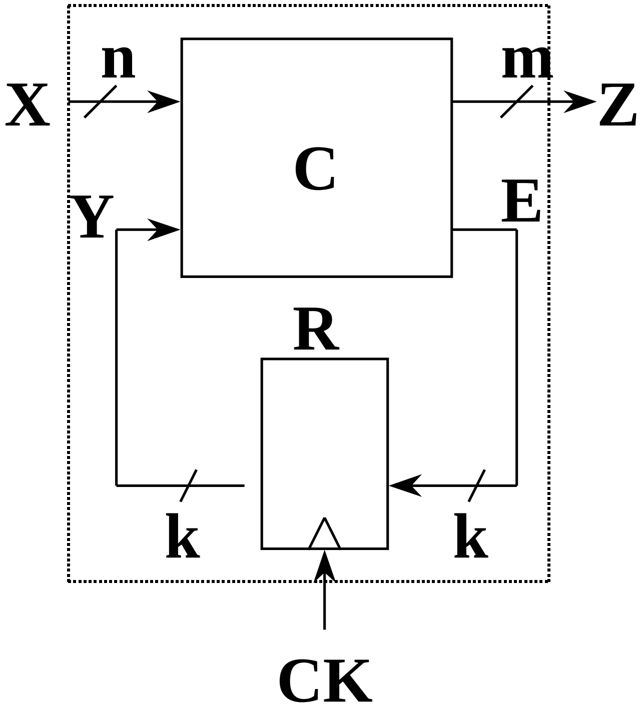


Original design

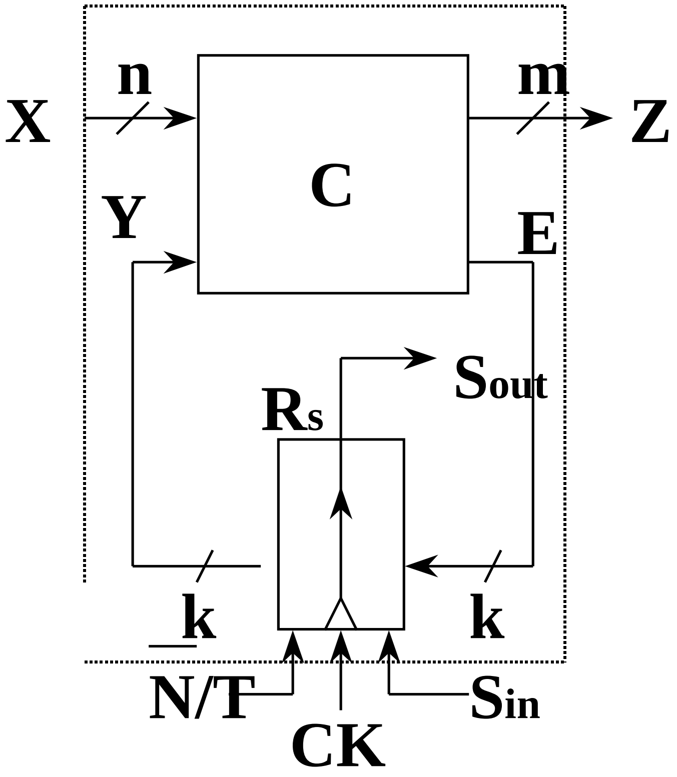


Modified circuit

Full Serial Integrated Scan



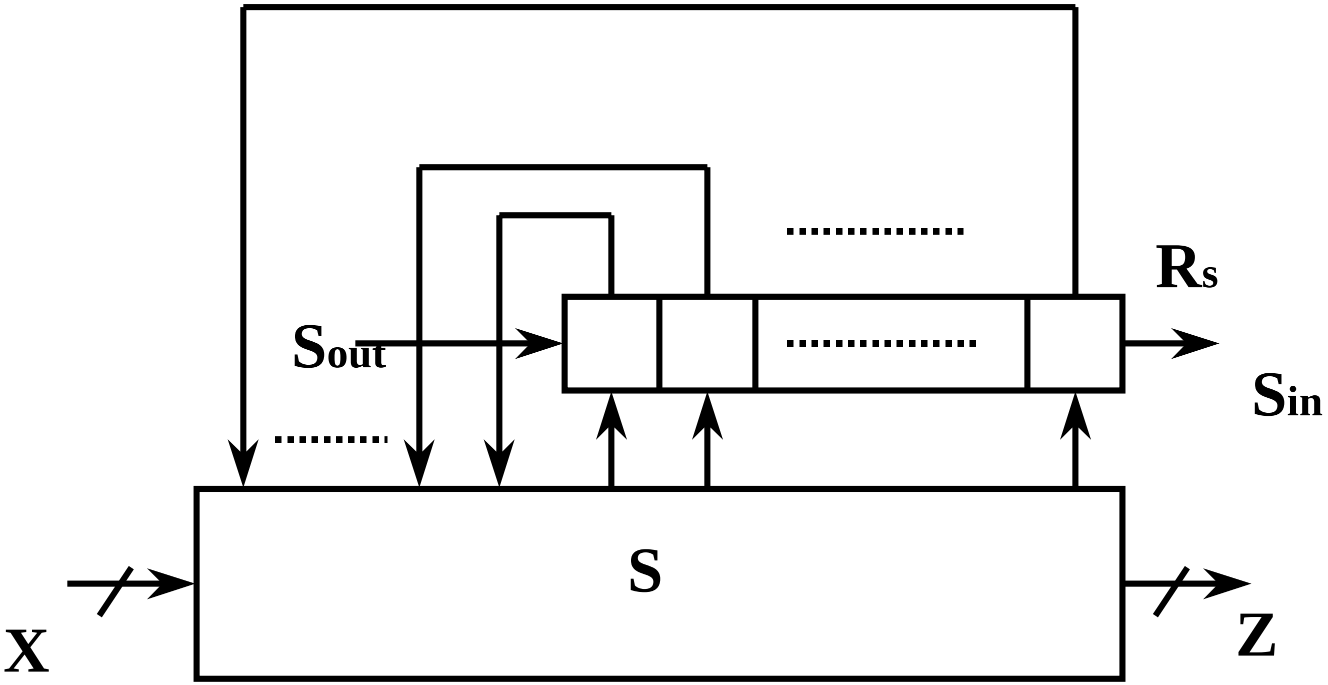
Normal



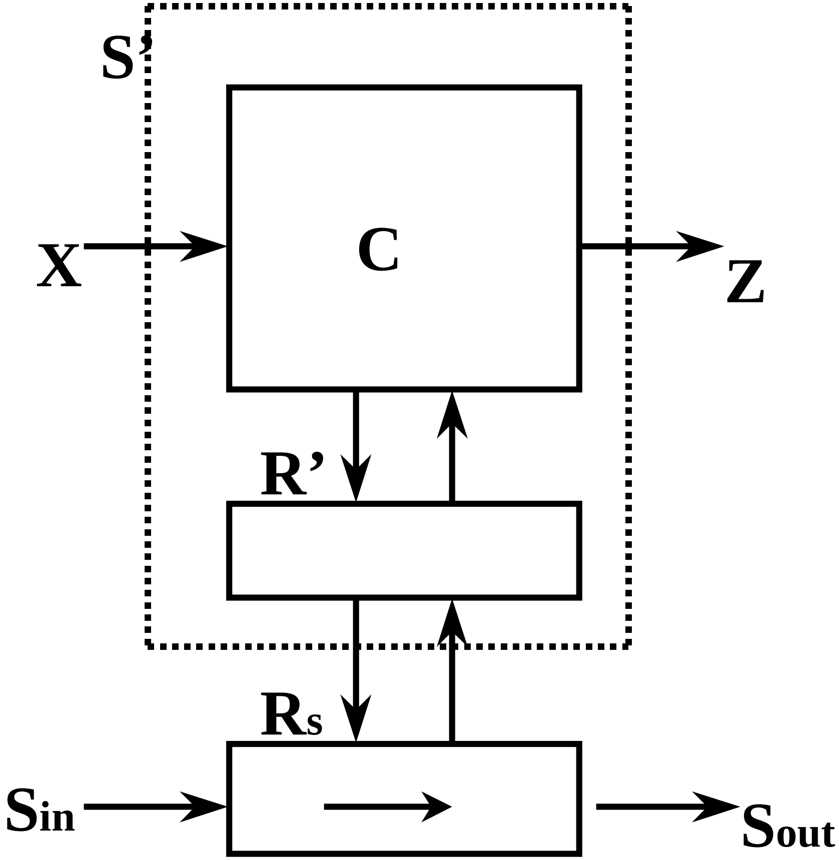
Scanned

Sequential ATPG → Combinational ATPG

Isolated Serial Scan (Scan/set)

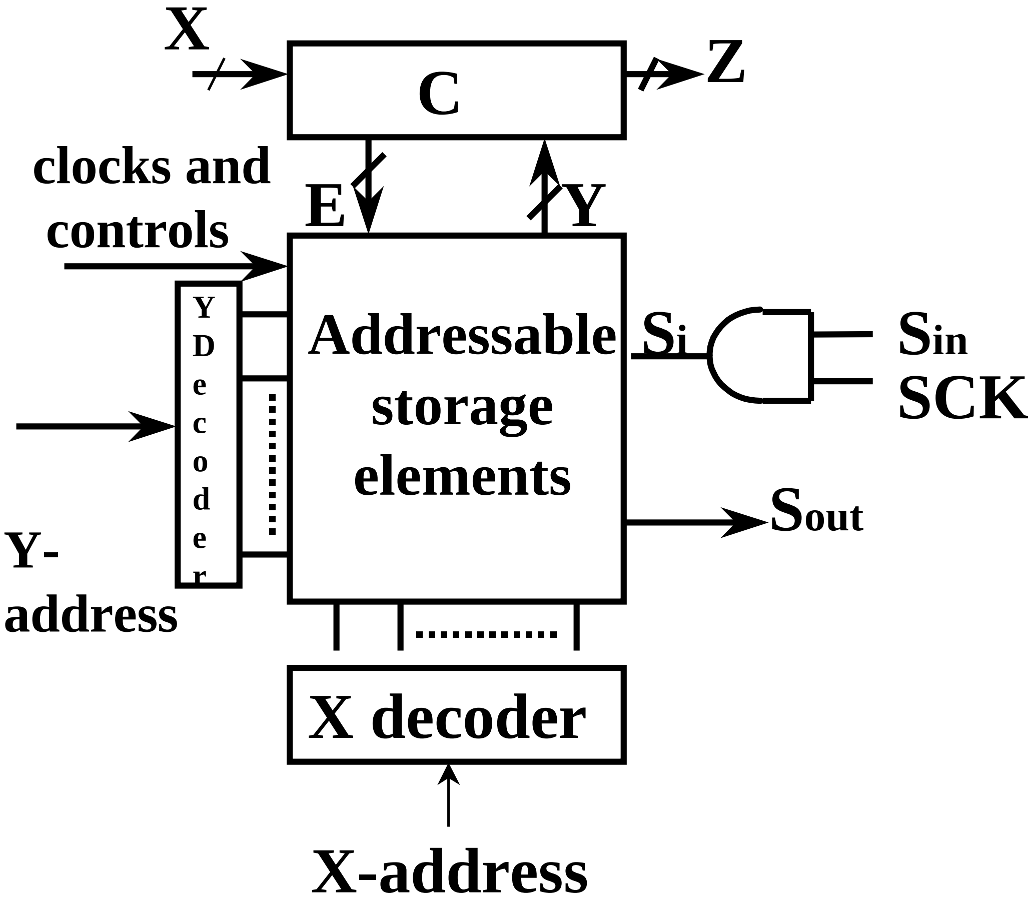


Full Isolated Scan (Structured)



- Shadow register
- Real-time test support
- snapshot

Random-Access Scan (Non serial-structured)



- High area overhead
- Faster test application: only bit change
- Concept of crosscheck

Scan Cell Design

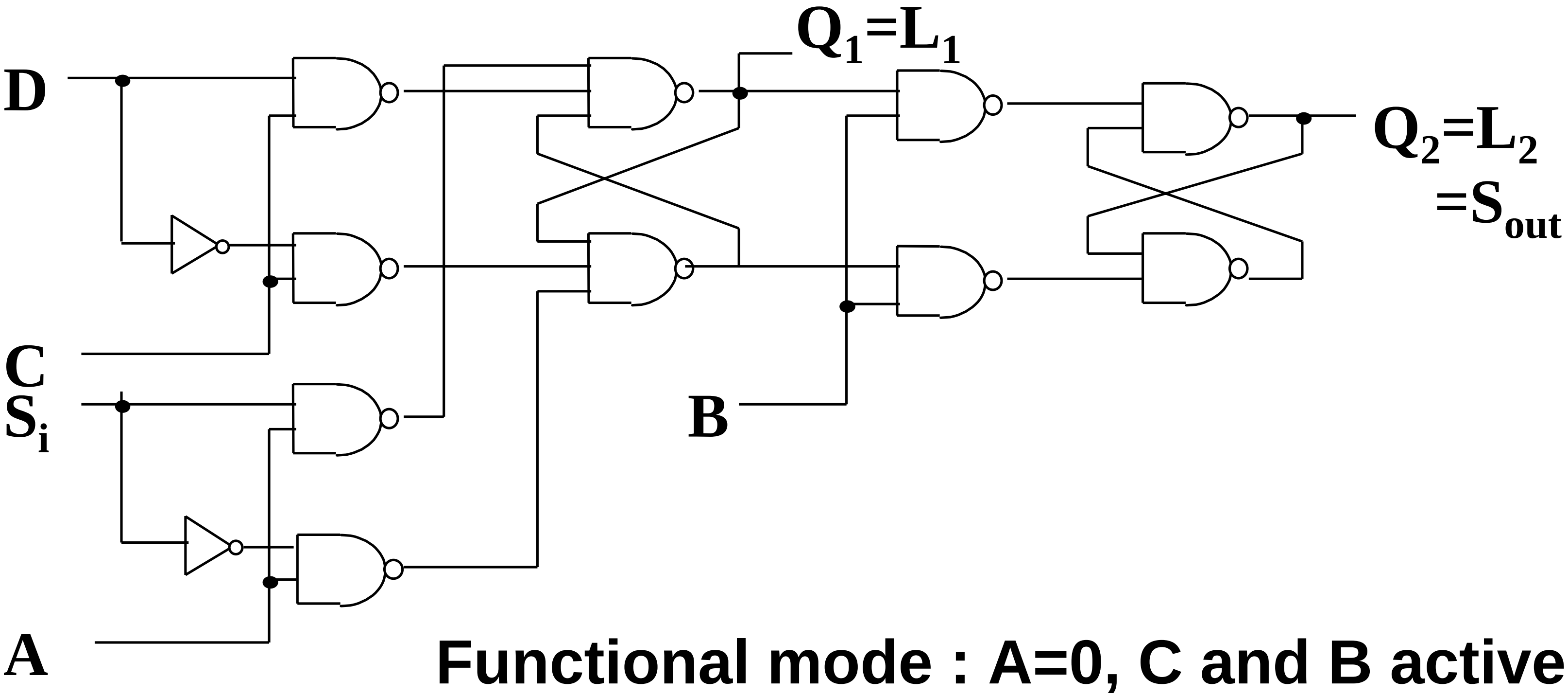
- Static / Dynamic
- Single / Double stages
- Latch / Flip-flop (Clocking Scheme)

Usually Two Operation Modes

- Functional mode
- Shift mode

IBM LSSD Scan Cell

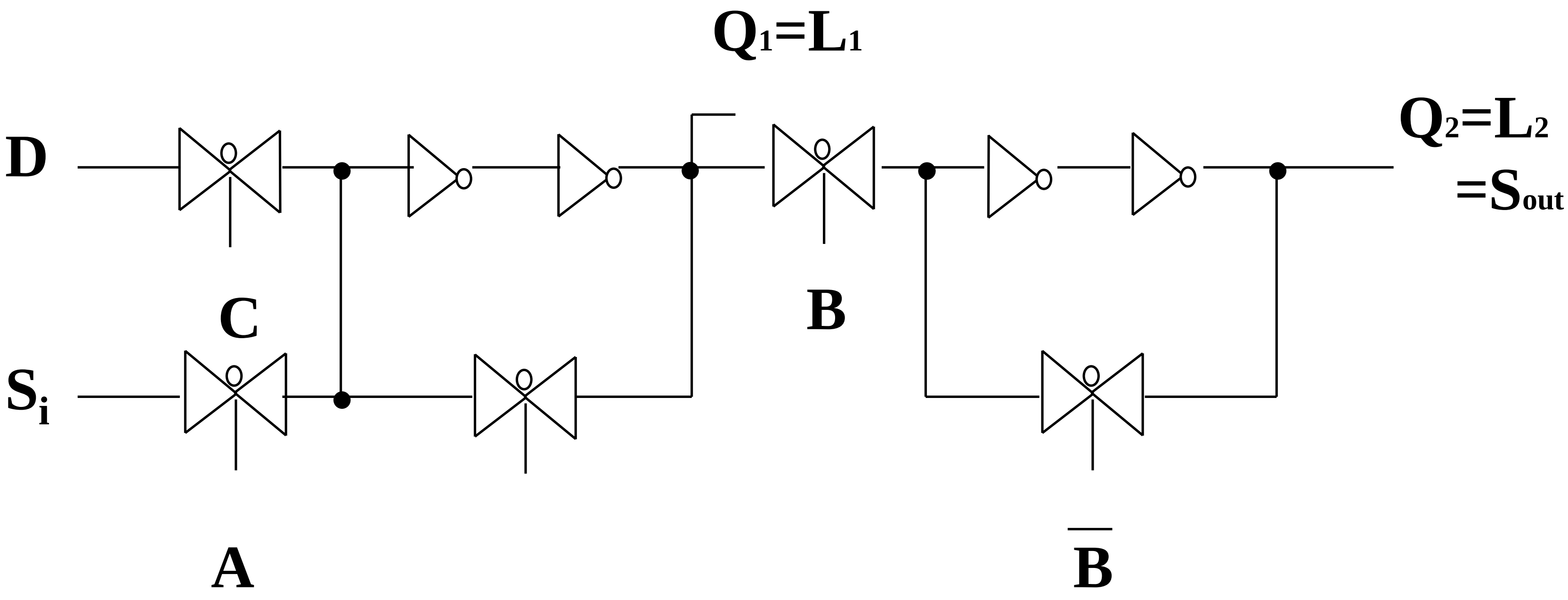
- Gate Level



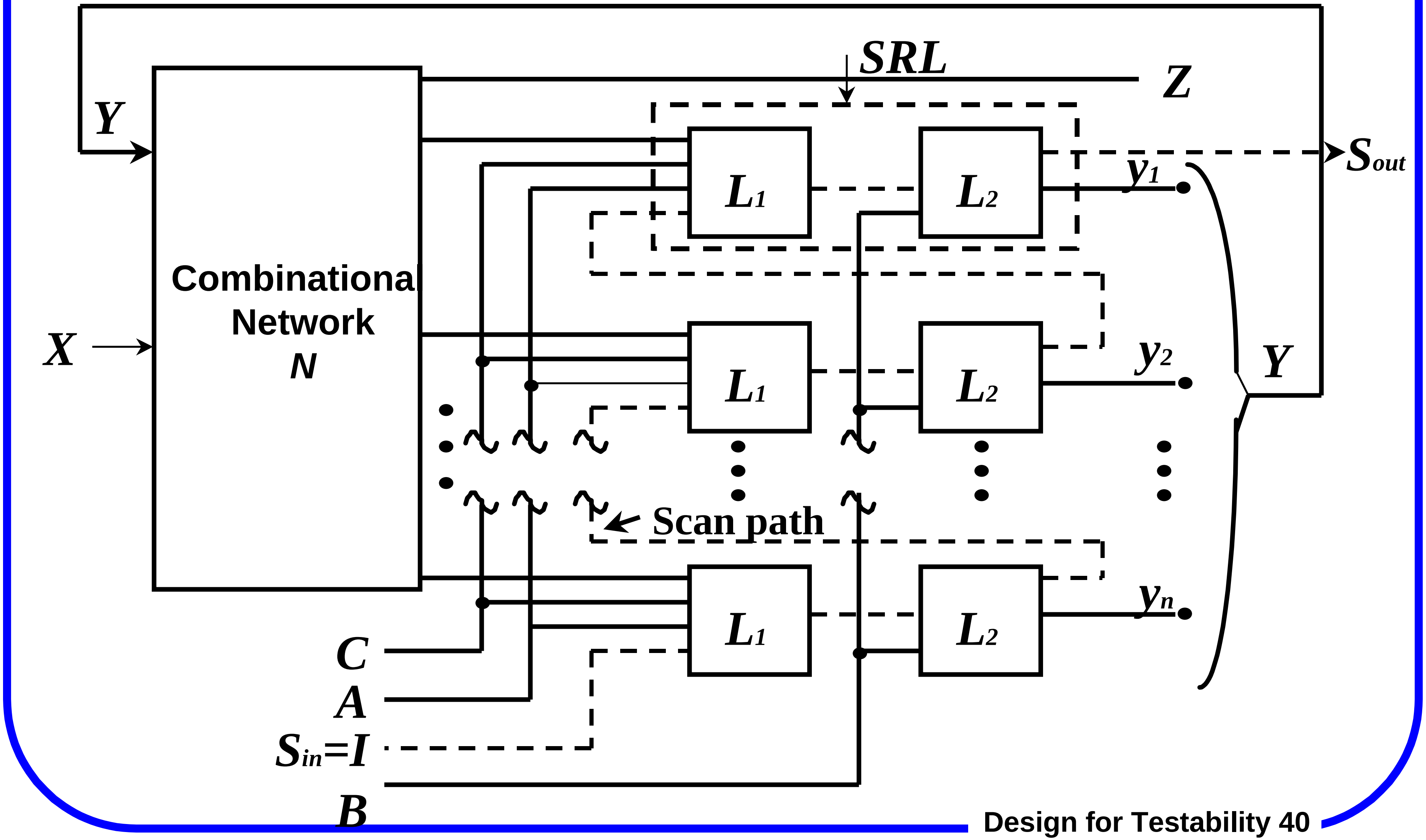
Functional mode : A=0, C and B active
Test(Shift) mode : C=0, A and B active

IBM LSSD Scan Cell (Cont.)

- **Switch / Inverter level**

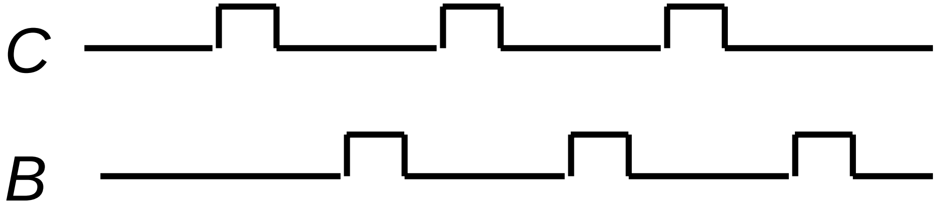


LSSD Double-Latch Design

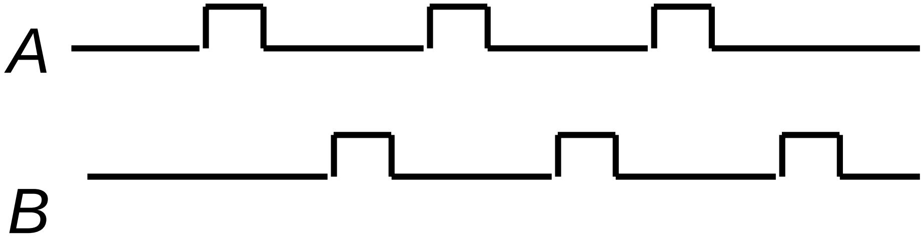


Clocking Scheme of LSSD Double-Latch

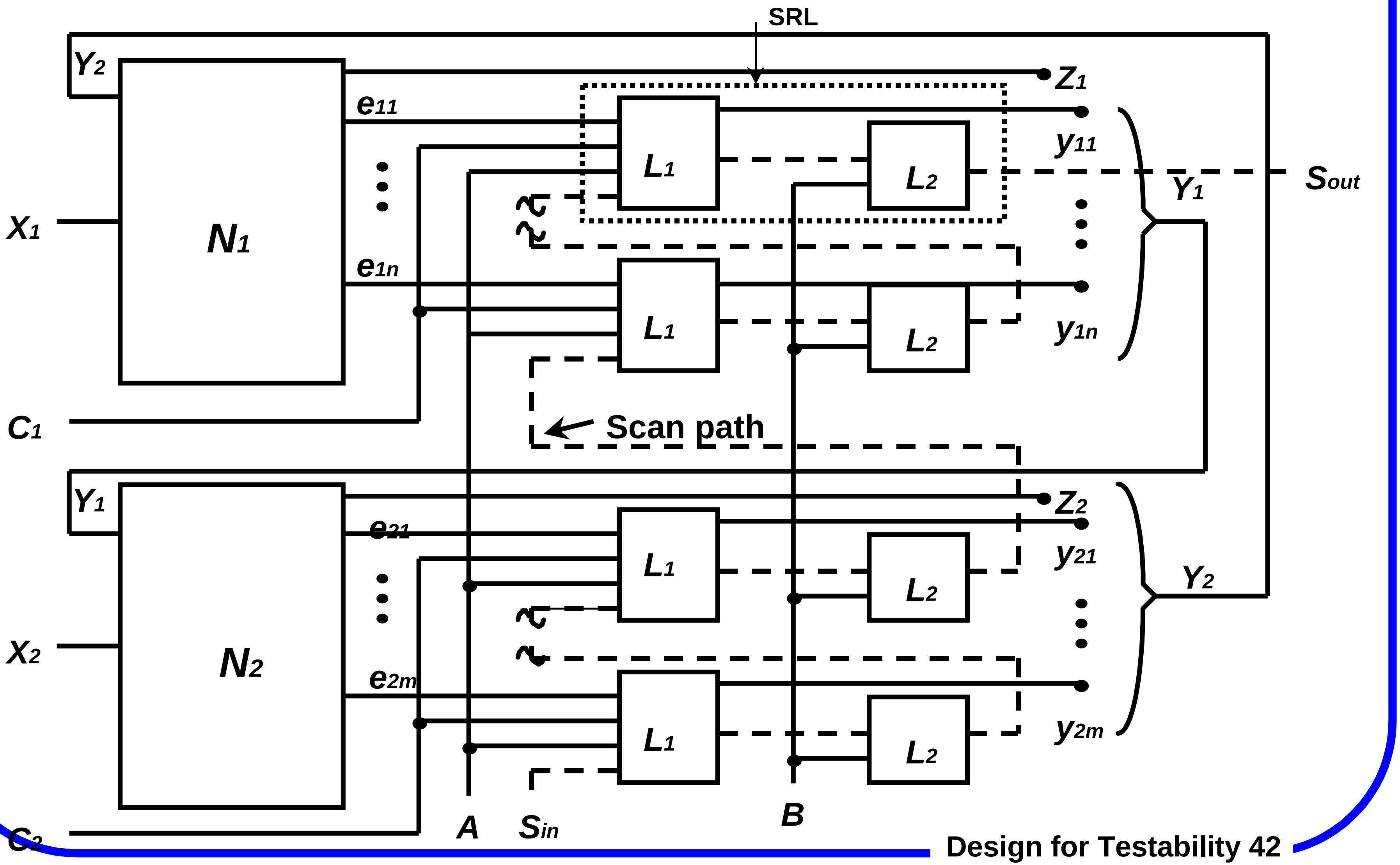
Normal mode:



Test mode:



LSSD Single-Latch Design



Scan Design Costs

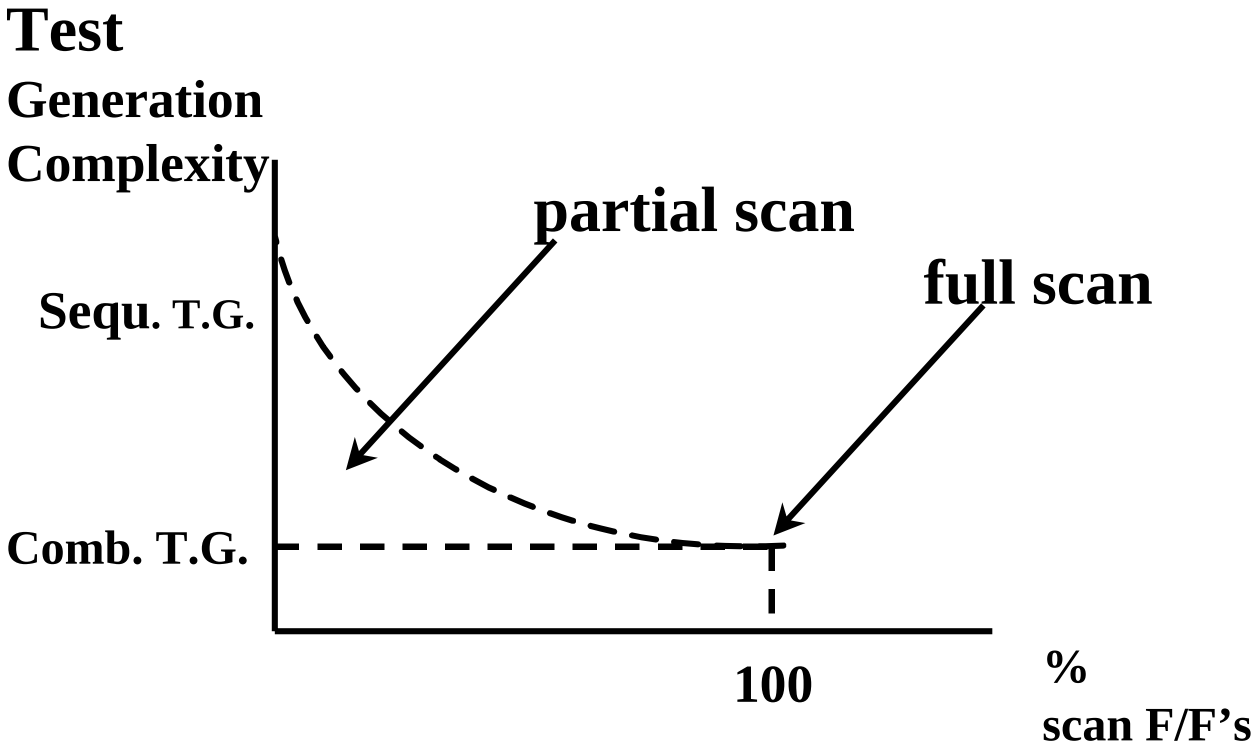
- Area overhead
- Possible performance degradation
- Extra pins
- High test time
- Extra clock control

Advanced Scan Concepts

- Partial scan (P.S.)
- Multiple test session (M.T.S.)
- Multiple scan chains (M.S.C.)
- Broadcast scan chains (B.S.C)

Method	P.S.	M.T.S.	M.S.C.	B.S.C.
Area overhead	↓	same	same or ↑	same
Performance Degradation	↓	same	same	same
Extra pins	same	same	same or ↑	↓
Extra clock control	same	same	same	same
Test application time	↓ or ↑	↓	↓	↓

Partial Scan: Only a subset of all flip-flops are scanned



- Trade-off between
  - –Area overhead
  - –Test generation complexity

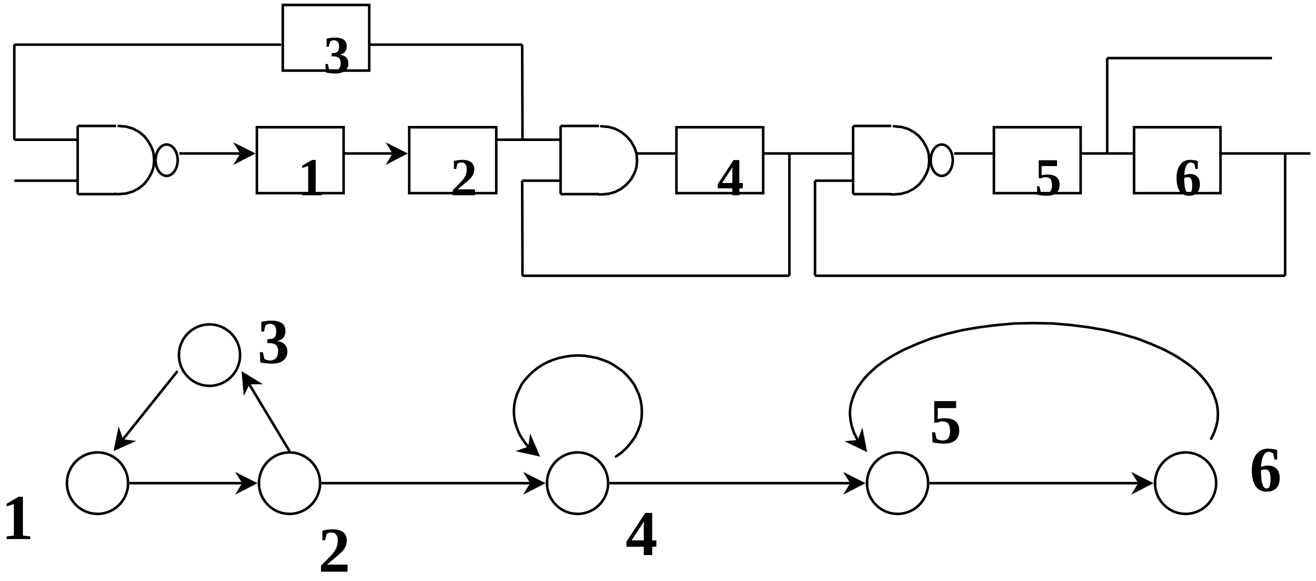
Partial Scan by Cheng & Agrawal
(pp. 544-548, IEEE Trans. Computers, Apr. '90)

- Basic idea:
  - Representing a circuit as a directed graph G=(v,E)
  - Trying to break cycles and reducing sequential depth

Graph Representation

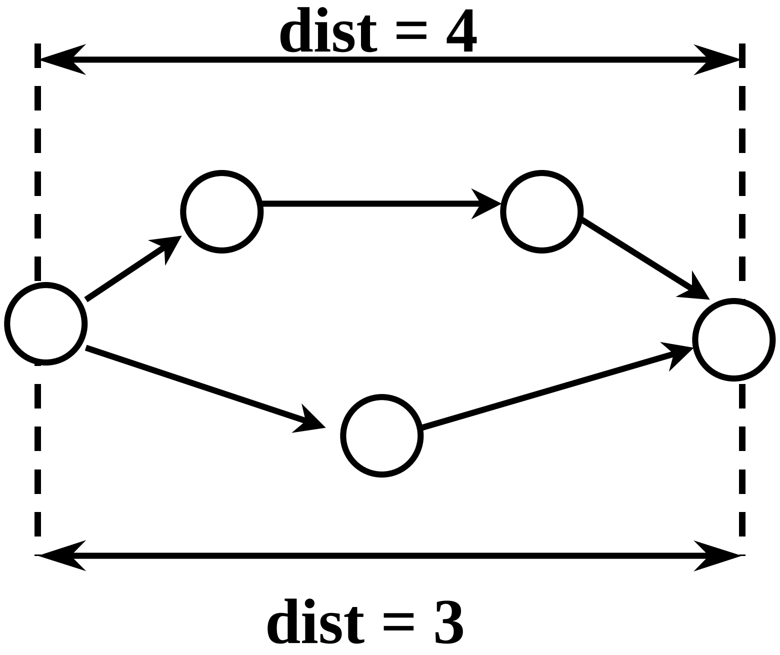
- Each flip-flop i => a vertex V_i
- Each combinational path from FF_i to FF_j
  - ⇨ an edge form V_i to V_j

Ex:



Graph Representation

Def: Distance between two vertices on a path = # of vertices on that path

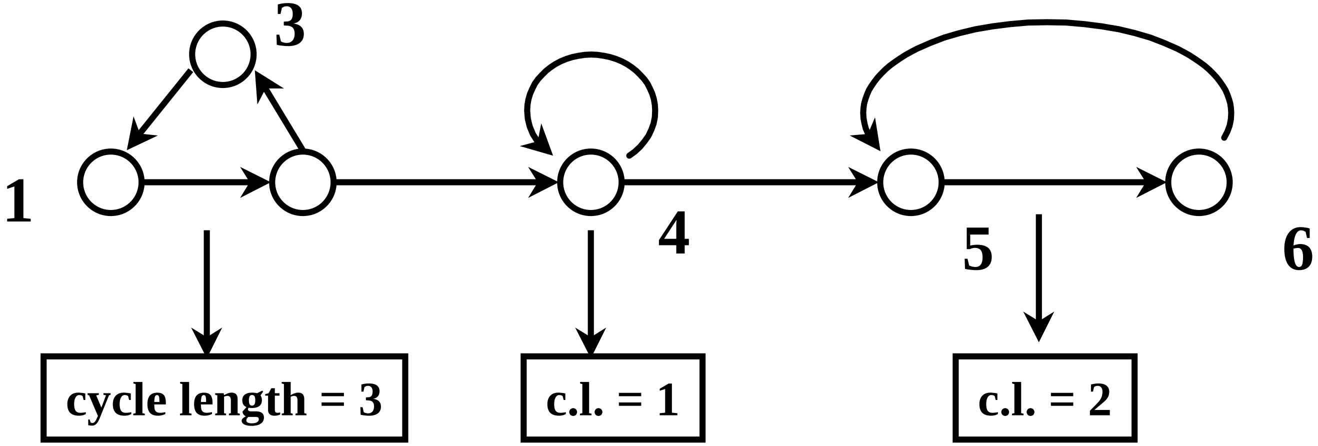


Graph Representation

Def: sequential depth of a circuit = the distance of the longest path

Def: Cycle length = maximum # of vertices in a cycle

EX: dist = 6



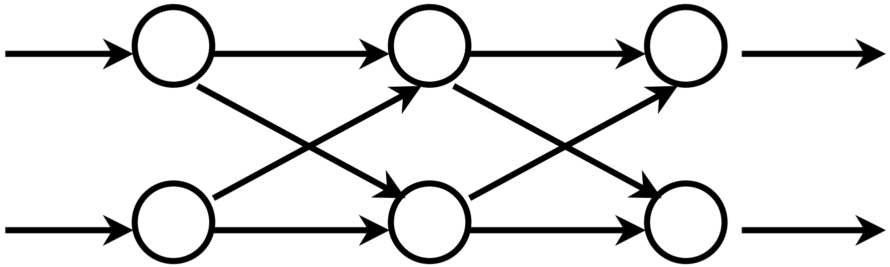
Analysis of Sequential Circuits

- Any sequential circuit can be divided into 3 classes of subcircuits based on the directed graph representation

  1. acyclic directed
  2. directed with only self loops
  3. directed with cycles of two or more vertices

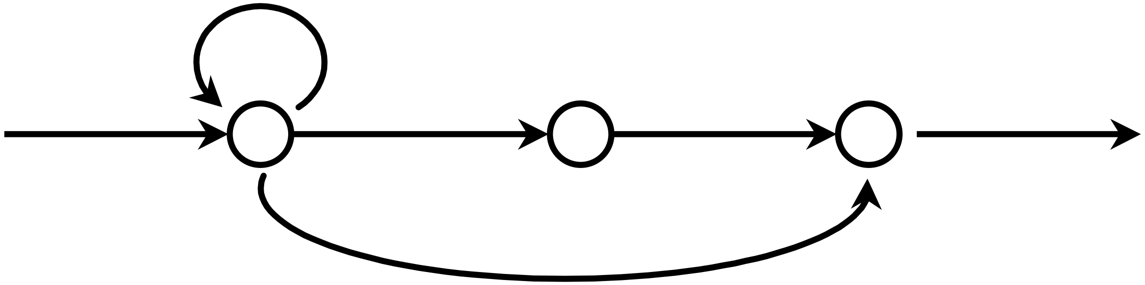
Ex:

1.

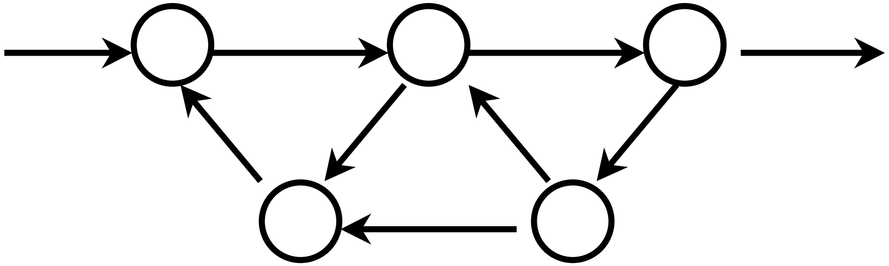


Analysis of Sequential Circuits (Cont.)

2.



3.



Experimental Results

- Experimental results show that
  - # of gates or # FF's is not the dominant factor for test generation complexity
  - Cycle length is the dominant factor
  - Sequential depth is minor

⇨ To reduce test generation complexity, cycles of length >= 2 should be eliminated

Flip-Flop Selection Algorithm

- Identify all cycles
- Repeat

```
for each vertex
    count the frequency of appearance in the cycle list
select the most frequently used vertex
    remove all cycles containing the remove  (selected) vertex
until (cycle list is empty)
```

⇨ This is a feedback vertex set problem, a well-known NP-complete problem, hence heuristic is used.

Experimental Results (Cheng & Agrawal ‘90)

PARTICAl SCAN FOR MULT4 (382 GATES, 15 FLIP-FLOPS)

No. Of scan FFs	Max cycle length	Depth	CPU sec. Test gen.	CPU sec. Fault sim.	Fault cov.	No. Of test	Total vector
0	4	13	75	5	98.01%	115	115
5	1	6	8	2	99.68%	69	345
6	1	4	8	2	99.68%	72	432

PARTICAl SCAN FOR CHIP-A (1112 GATES, 39 FLIP-FLOPS)

No. Of scan FFs	Max cycle length	Depth	CPU sec. Test gen.	CPU sec. Fault sim.	Fault cov.	No. Of test	Total vector
0	1	14	269	274	98.80%	868	868
8	1	10	85	56	99.60%	529	4132
16	1	6	49	33	99.80%	387	6192

Experimental Results

(Cheng & Agrawal ‘90)

PARTICAl SCAN FOR CHIP-B (5294 GATES, 318 FLIP-FLOPS)

No. Of	Max cycle	Depth	CPU sec.		Fault	No. Of	Total
scan FFs	length		Test gen.	Fault sim.	cov.	test	vector
0	40	43	11018*	2256	82.60%	948	948
14	1	19	2946*	2986	97.90%	2607	39498
36	1	10	2041*	2765	98.30%	2494	89784
44	1	6	1207*	2526	97.80%	1741	76604
87	1	4	643*	862	98.20%	842	73254
87	1	4	2294	7961	98.43%	2536	220632

*20% sample of total faults used for test gen. and fault sim.

Experimental Results

(Cheng & Agrawal ‘90)

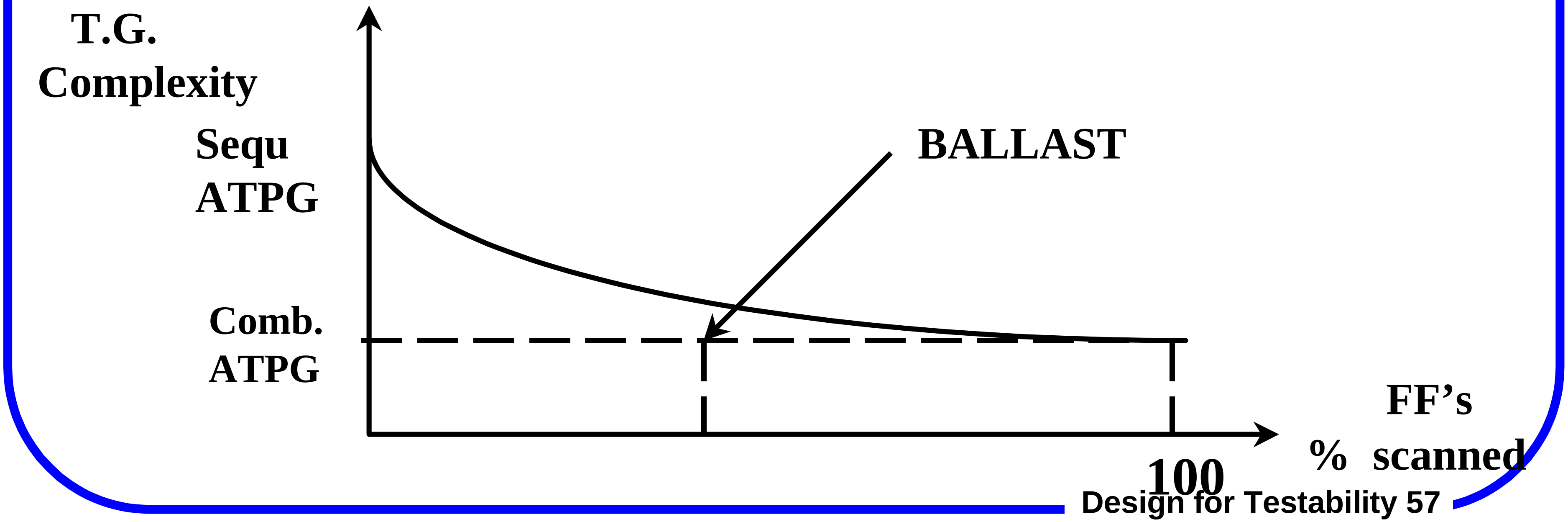
TEST GENERATION FOR SEQUENTIAL BENCHMARK CIRCUITS WITH PARTIAL SCAN

Circuit Name	Total No.Of FFs	Scan FFs		No. of test Vector	Fault Coverage(%)			Tgen + Fsim Sec (VAX 8650)
		No.	%		Tested	Redundant	Total	
s400	21	9	42.86	107	99.81	1.89	100.00	7
s713	19	7	36.84	83	90.71	9.29	100.00	18
s5378	179	32	17.89	2612	93.38	6.32	99.70	1253
s9234	228	53	23.25	3458	43.23	55.80	99.04	6208

The BALLAST Methodology

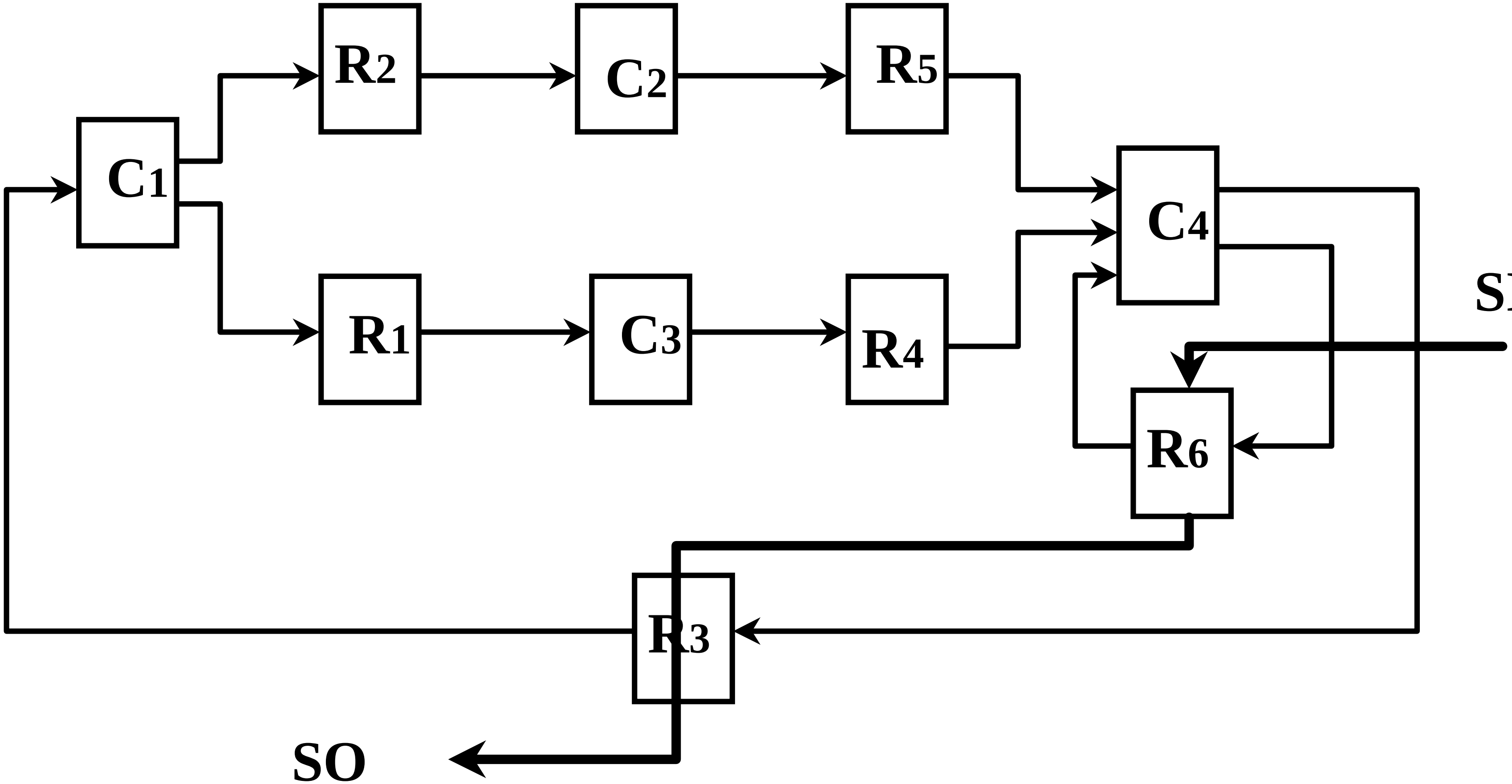
(Rajesh Gupta, Rajiv Gupta, M.A.Breuer, IEEE T-Computers, Apr.'90)

- Scan storage elements are selected such that remainder of the circuit has some desirable structure.
- A complete test set can be obtained using combinational ATPG.



Example

Fig.1



Example (Cont.)

- Test procedure:
  - Scan in a test pattern to R3 and R6 .
  - Hold test pattern in R3, R6 for two clock cycles such that test response appears in R5 and R4.
  - Load data (from R5, R4) to R6, R3 and shift out

Circuit Model

- Register:
  - Collection of one or more FF's driven by the same clock signal and (if any) mode control signal.
- Two types of registers:
  - Load set L - the set of registers whose FF's have no explicit load enable control => always operate in LOAD mode.
  - Hold set H - two modes of operations : LOAD and HOLD.
- In the previous example, R1, R2, R4, R5 belong to LOAD set, and R3, R6 belong to Hold set H.

Circuit Model (Cont.)

- Clouds
  - The combinational logic logic in a circuit S can be partitioned into clouds, where each cloud is a maximum region of connected combinational logic such that its inputs are either PIS or outputs of FF's and its outputs are either POS or inputs to FF'S.
- Ex
  - In Fig.1 each block of C1, C2 , C3 and C4 represents a cloud.

⇨ No two clouds can be directly connected together.

⇨ Each FF (in any register) must receive data from exactly one cloud and must feed exactly one cloud.

⇨ FF's can be grouped into registers such that each register receivers data from exactly one cloud and feeds exactly one cloud.

Topology graph G=(V, A, H, W)

V: set of clouds.

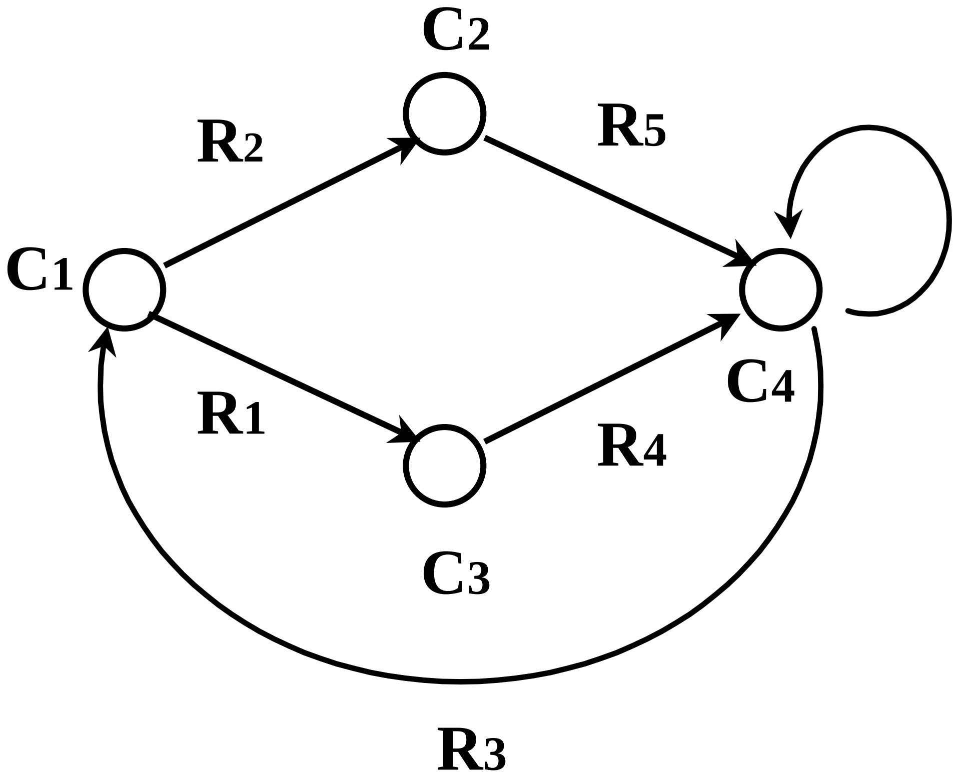
A: connections between clouds through registers.

H: arcs in $H \subset A$ represents HOLD registers.

w:A → Z+ (positive integers) defines the number of FF'S in each register.

⇨ **w(a) represents the cost of converting register a into a scan path register.**

Example



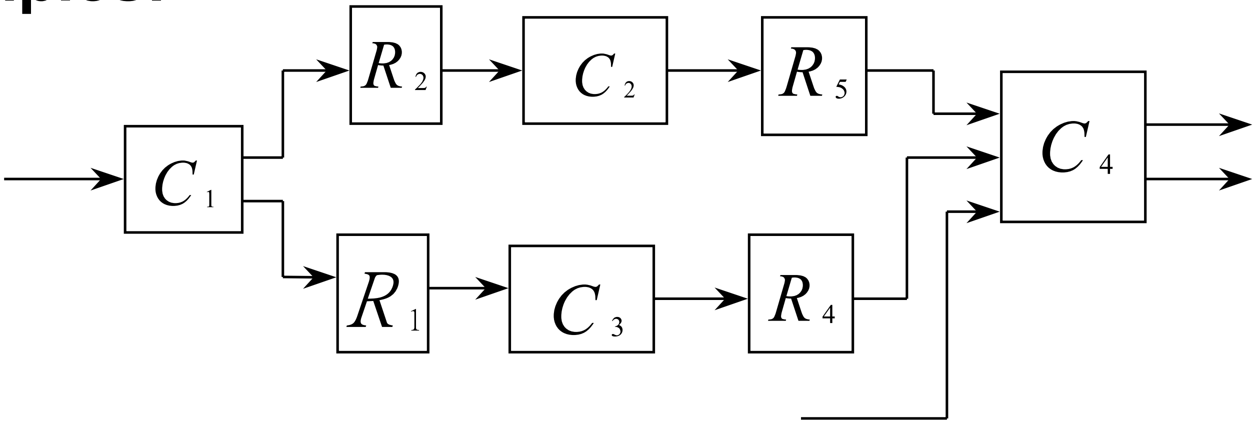
B-structure

- S: a synchronous sequential circuit with topology graph G=(V, A, H, W).
- S is said to be to a balanced sequential structure (B-structure) if
  - G is acyclic.
  - ∀ v1,v2 ⊂ V, all directed paths from v1 to v2 are of equal length.
  - ∀ h ⊂ H, if h is removed from G, the resulting graph is disconnected.
- When considering whether a circuit with scan registers is a B-structure, the arcs corresponding to scan registers must be removed.
- Condition 3 means that the removed of in the scan path will disconnect the graph.

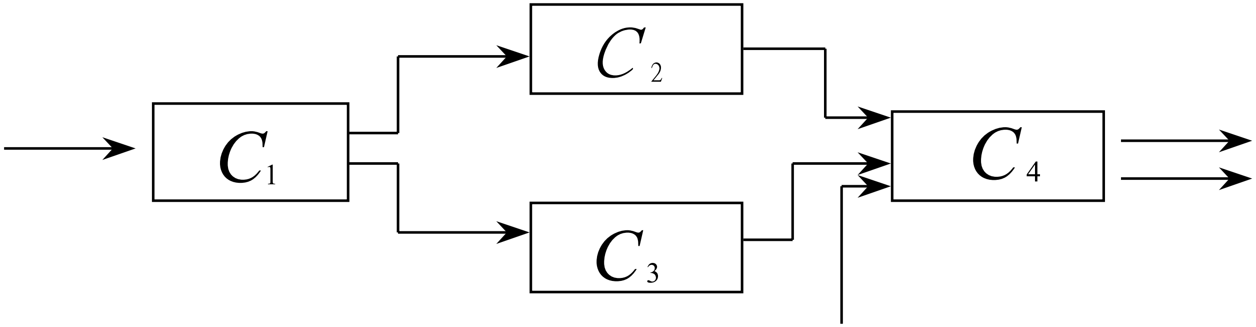
Kernel :The circuit excluding the scan path

- Combinational equivalent of C^B of a B-structure S^B : the combinational circuit formed by replacing each FF in every register in S^B by a wire.
- Depth d of S^B:the longest directed path in the topology graph

Examples:



Kernel of Fig.1



Combinational
equivalent

- Given an input pattern I applied to S^B,the single -pattern output of $S^B$for I is defined as the steady -state output of S^B when I is held constant and all registers are operated in LOAD mode for at least d clock cycles
- Given some fault f in S^B,if the single-pattern outputs for I of the good and faulty circuits are different , then I is a single -pattern test for f

⇨ B - structures : (1) single -pattern testable (2) complete single - pattern test set can be derived using combinational test generation techniques

Outline of BALLAST

(1) Construct the topology graph G of the circuit

(2) Select a minimal cost set of arcs R to be removed from G such that the remaining topology graph is balanced. let S^B be the B - structure corresponding to the resulting topology graph

(3) Determine C^B of S^B. Using a combination ATPG to determine a complete test set T for C^B

(4) Construct a scan path by appropriately ordering the registers in R and connecting them so that they can both "shift" and "hold"

⇨ Later some "hold"can be released

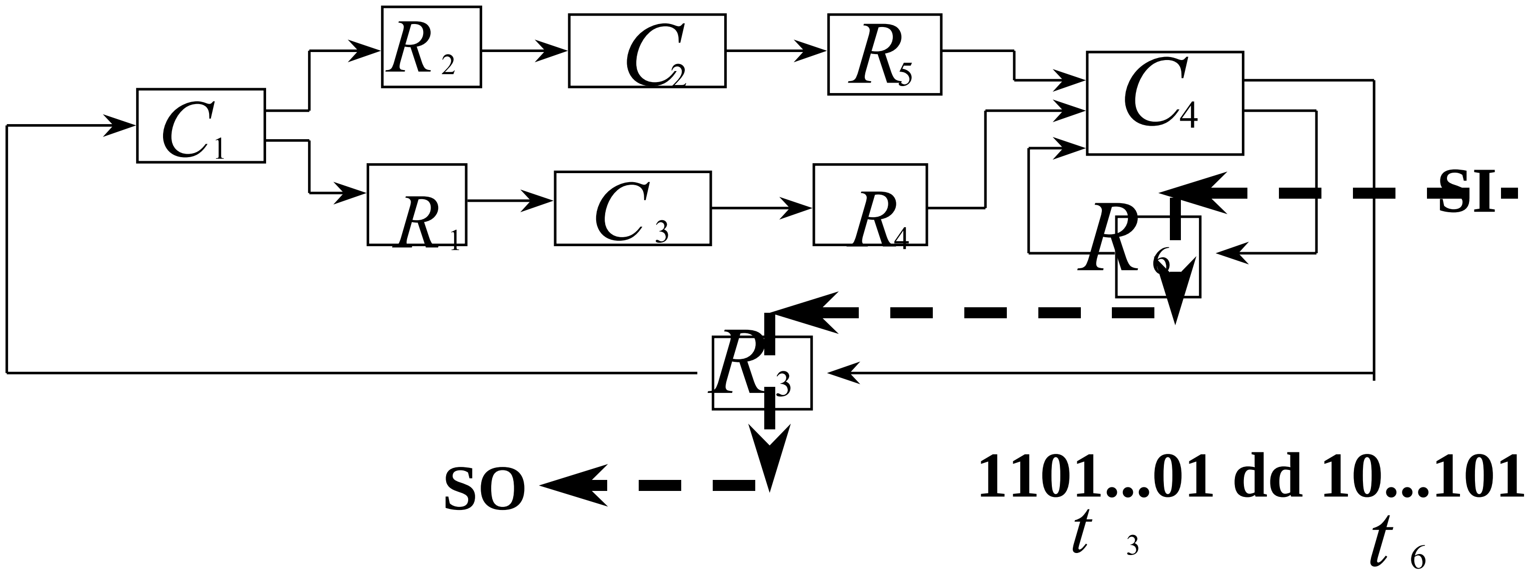
Selection of Scan Registers

(1) Transform G=(V,A,H,W) into an a cyclic topology graph G_A by removing a set of “feedback” arcs R_A such that $\sum_{a \in R_A} w(a)$ is minimized

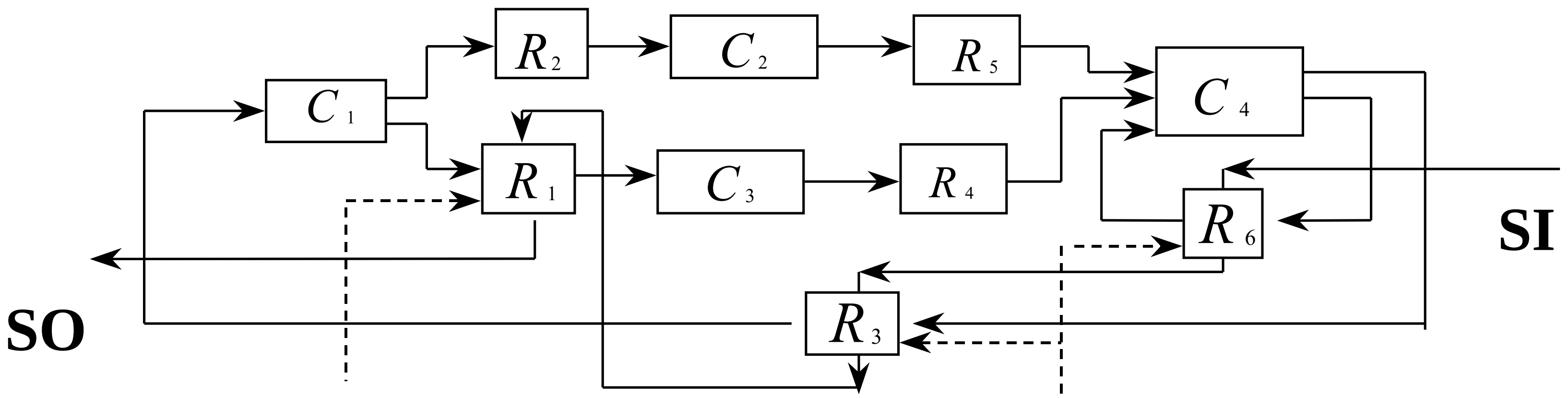
(2) Transform G_A into a balanced topology graph G_B by removing a set of arcs R_B such that $\sum_{a \in R_B} w(a)$ is minimized R= $R_A \cup R_B$ is the desired set of registers

- Both(1) , (2) are NP-complete. Refer to the paper for a heuristic for (2)

Eliminate on HOLD mode of Scan Registers



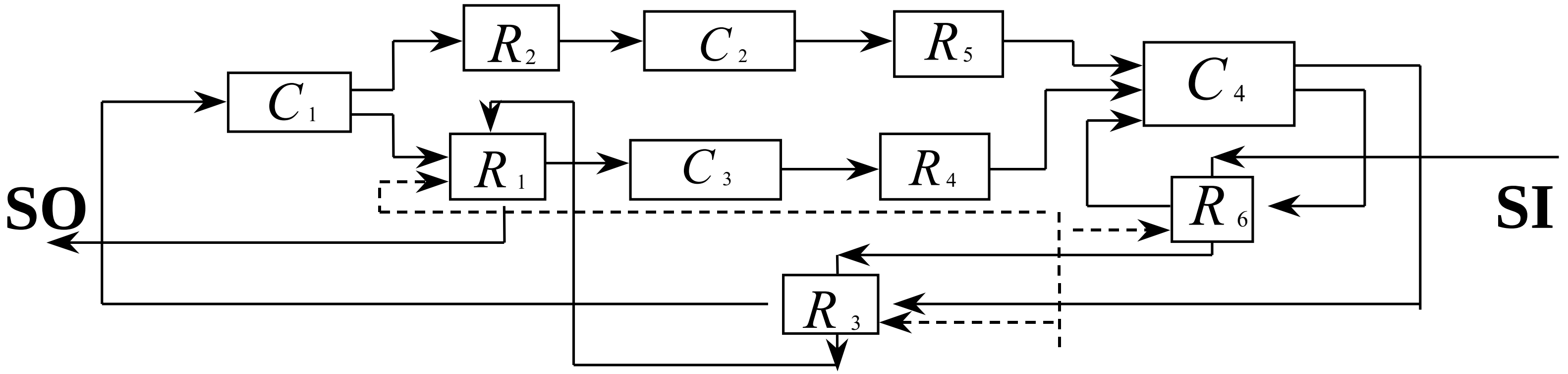
- By adding two dummy bits between the patterns to be scanned to R_3 and R_6 ,the HOLD mode can be eliminated



HOLD control(in original circuit)

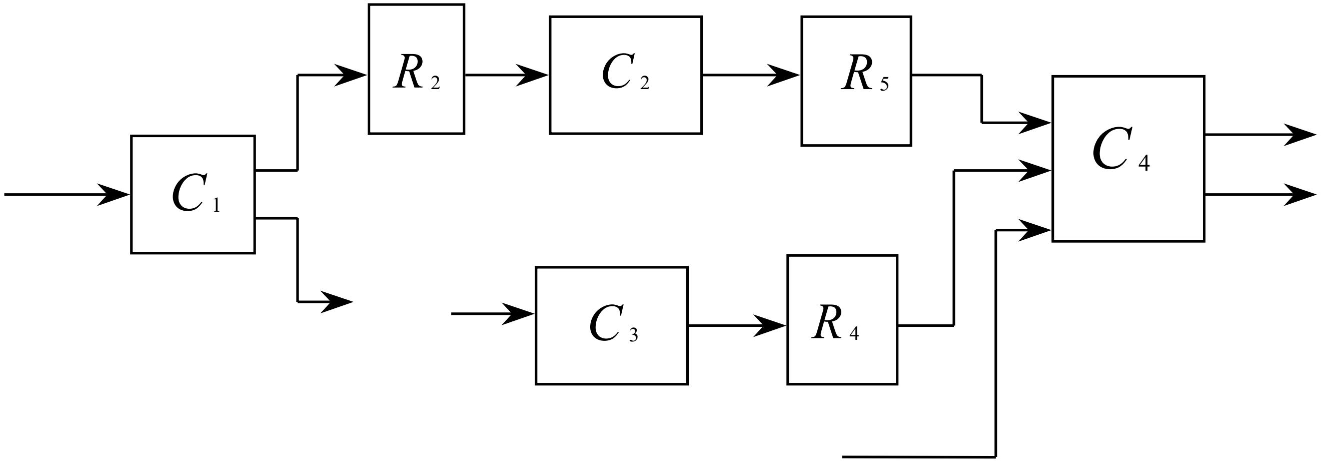
HOLD control (for test)

(a)

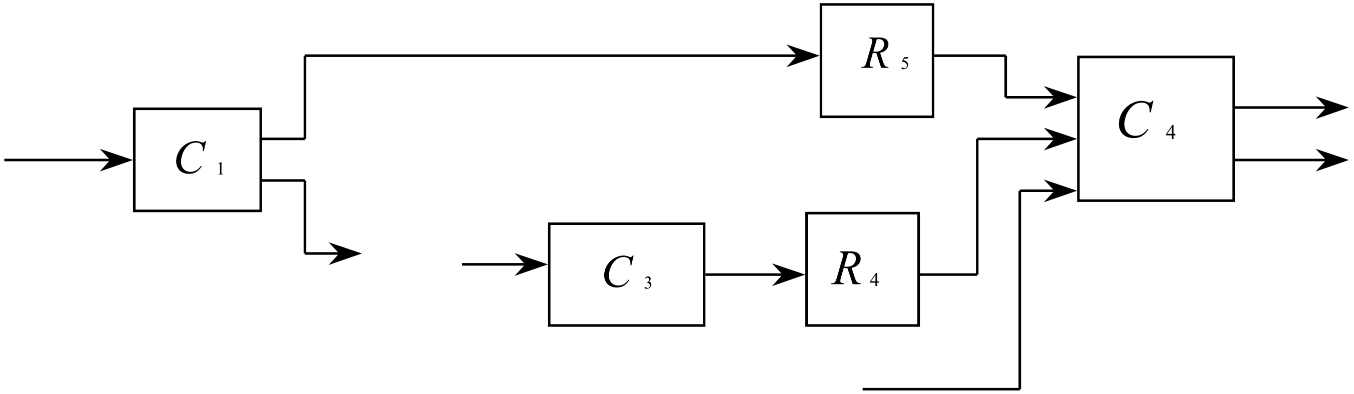


HOLD control (for test)

(b)



(c)



(d)

Test Procedure

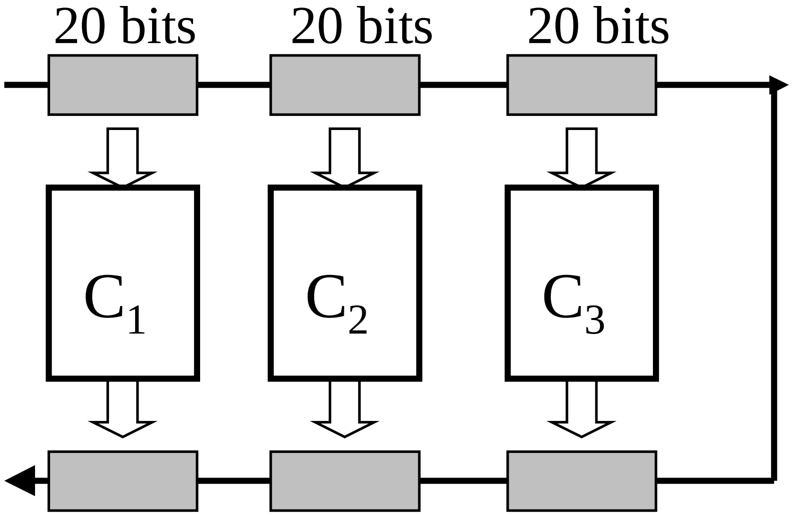
(1) all scan registers in SHIFT mode for l clock cycle

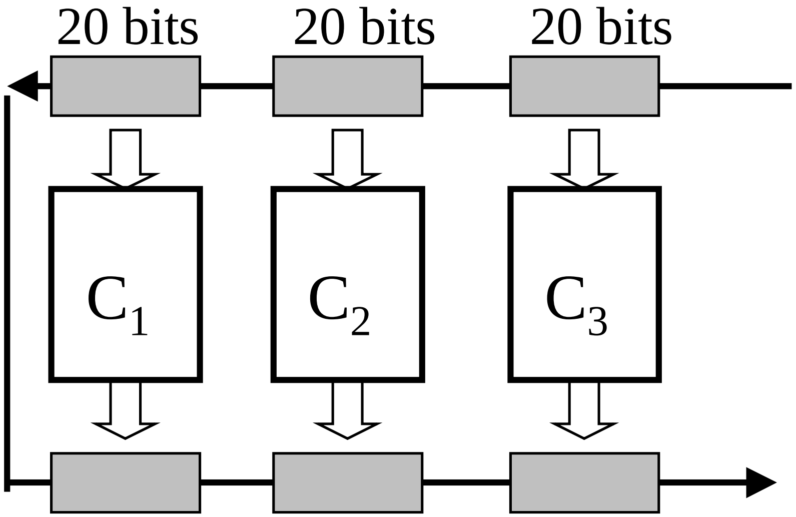
(2) Repeat N times

- a. HOLD all scan registers , LOAD all other for d cycles
- b. LOAD all scan registers for 1 clock cycle
- c. SHIFT out scan data

Multiple Test Session

- # patterns: C_1 :100, C_2: 200 and C_3: 300

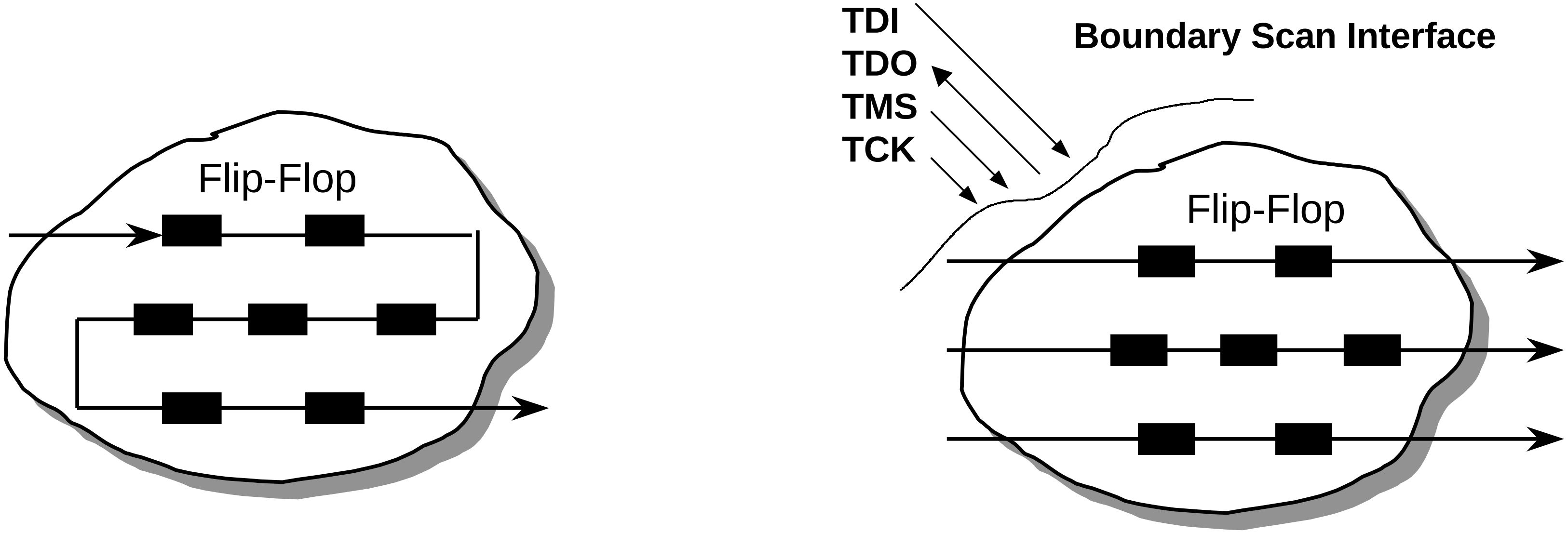




Test Time
= 60 *300
=18,000 (cycles)

Test Time
= 60 *100+40*100+20*100
=12,000 (cycles)

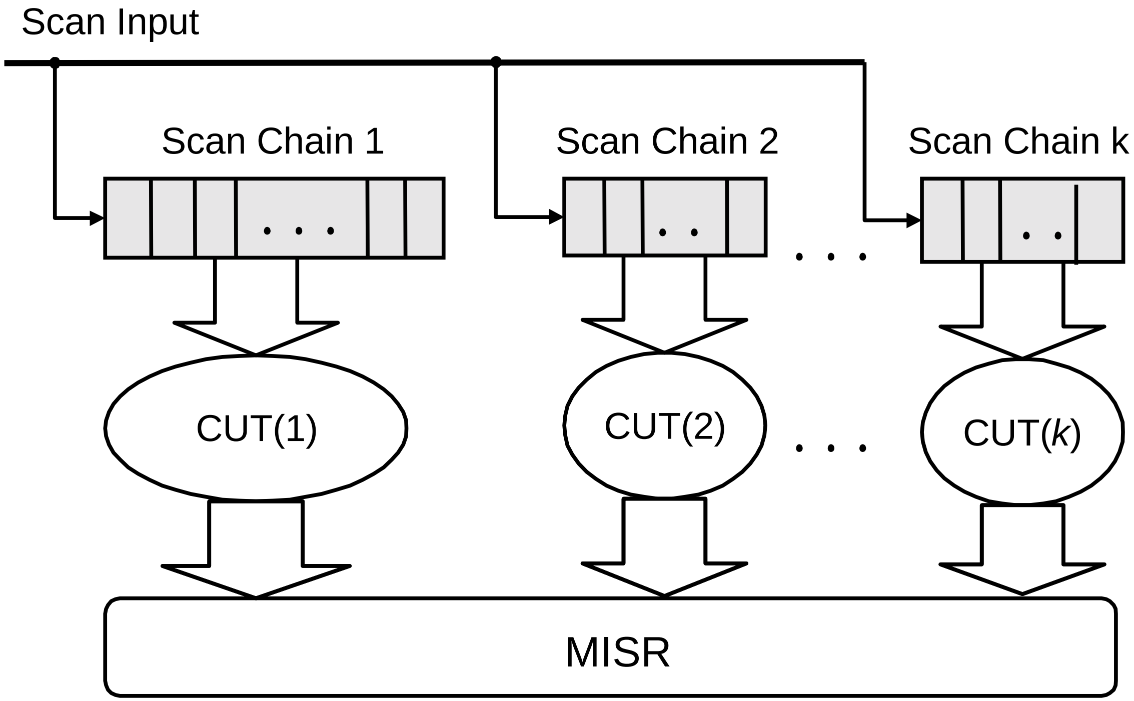
Multiple Scan Chains



- Reduce test application time
- Usually test I/O will share the normal I/O

Broadcast Scan Chains-
General Hardware Architecture

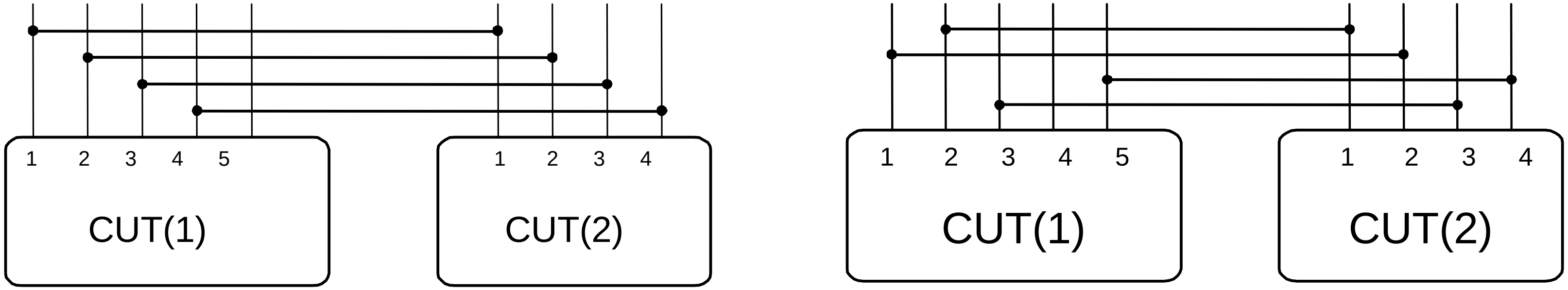
- **Using a single data input to support multiple scan chains**



Virtual Circuits

- The inputs of CUTs are connected in a 1-to-1 manner.

  Example :



(a) i-to-i connection

(b) random connection

➡ **The whole virtual circuit is considered as one circuit during ATPG.**

➡ **The resulting test patterns can be shared by all CUTs.**

Experimental Results

Experimental results for ISCAS’85

ISCAS'85 Experiment	Single	Multiple	Method 1	Method 2
Test Efficiency (%)	100	100	100	100
# Test Patterns	130	130	195	177
Scan Chain Length	834	206	206	206
Test Generation Time(secs)	163.2	163.2	122.2	130.3
Test Application Cycles	108420	26780	40170	36462
Normalized Test Application Cycles	4.05	1	1.50	1.36
	1	0.25	0.37	0.34

Method 1: Combine all input 1’s, input 2’s, etc.
Method 2: Distributed.

Experimental Results (Cont.)

Experimental results for ISCAS'89

ISCAS'89 Experiment	Single	Multiple	Method 1		Method 2	
			FFs	FFs & PIs	FFs	FFs & PIs
Test Efficiency (%)	100	100	100	100	100	100
# Test Patterns	281	281	287	294	280	285
Scan Chain Length	6587	1728	1728	1728	1728	1728
Test Generation Time(secs)	1293.9	1293.9	1802.0	1820.1	1893.7	1869.2
Test Application Cycles	1850947	485568	495936	508032	483840	492480
Normalized Test Application Cycles	3.81	1	1.02	1.05	0.99	1.01
	1	0.26	0.27	0.27	0.26	0.27

FFs : Only FFs are combined.

FFs & PIs : Both FFs and PIs are combined.