COE 571 Digital System Testing

Dr. Aiman El-Maleh

Design for Testability (DFT)

1. **DFT techniques**
2. **Test points**
3. **Enhancing controllability & observability by scan registers**
4. **Boundary scan**
5. **Storage cells for scan design**
6. **LSSD design rules**
7. **Partial scan**

Design For Testability

- Testing cost involves:
  - Cost of test pattern generation
  - Cost of fault simulation
  - Cost of test equipment
  - Testing time
- Increasing the testability of a design should reduce its testing cost
- Testability of a design can be improved by design for testability (DFT) techniques
- Factors impacting test generation complexity
  - Controllability
  - Observability
  - Predictability
- DFT techniques are design effort employed to ensure that a device is testable
- Controllability & observability are important attributes related to testability

- Circuits typically difficult to control:
  - decoders
  - circuits with Feedback
  - oscillators & clock generators
  - Counters

- Less observable circuits:
  - circuits with global feedback
  - embedded RAMs, ROMS, PLAs
  - Concurrent error-checking circuits
  - circuits with redundant nodes

- General observations:
  - Sequential logic more difficult to test than combinational logic
  - Control logic more difficult to test than combinational logic
  - Random logic more difficult to test than structured designs
  - Asynchronous designs much more difficult to test than synchronous designs

Trade - Offs

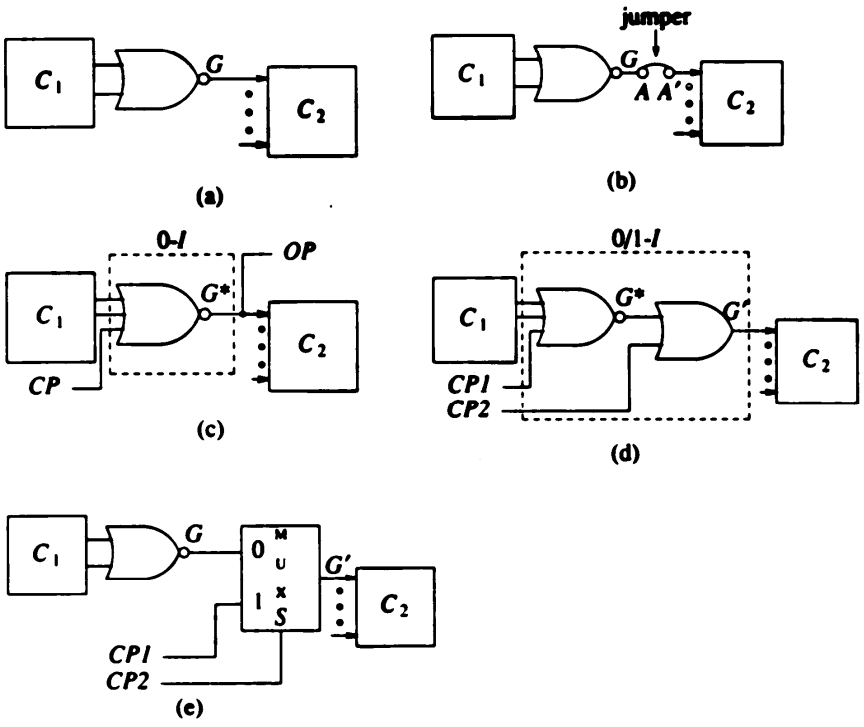
- DFT techniques deal with:
  - resynthesis of an existing design
  - addition of extra hardware to design
- DFT techniques affect:
  - area
  - I/O pins
  - Circuit delay
- Increasing area and/or logic complexity increases power consumption and decreases yield.
- A careful balance must be reached between adding logic for DFT & yield
- If added DFT does not increase fault coverage to certain level, defect level will increase.
- DFT is used to:
  - reduce test generation costs
  - enhace test quality (fault coverage)
  - reduce defect level
  - reduce test length, tester memory and test application time

DFT Techniques

- DFT techniques fall into two classes:
  - Ad Hoc techniques
  - Structured
- Ad Hoc techniques do not deal with a total design methodology & can be used at designer's option where applicable
- Ad Hoc techniques deal with the following issues:
  - test point insertion
  - initialization
  - oscillators & clocks
  - counters / shift registers
  - partitioning large circuits
  - logical redundancy
  - breaking global feedback paths

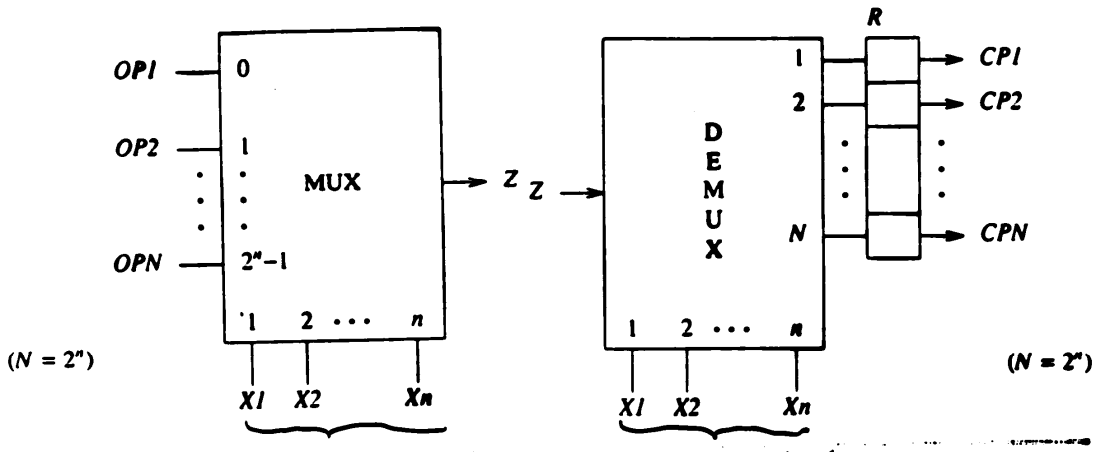
Test Points

- Employ test points to enhance controllability and observability
- Selection of where to insert the test points is done by designer



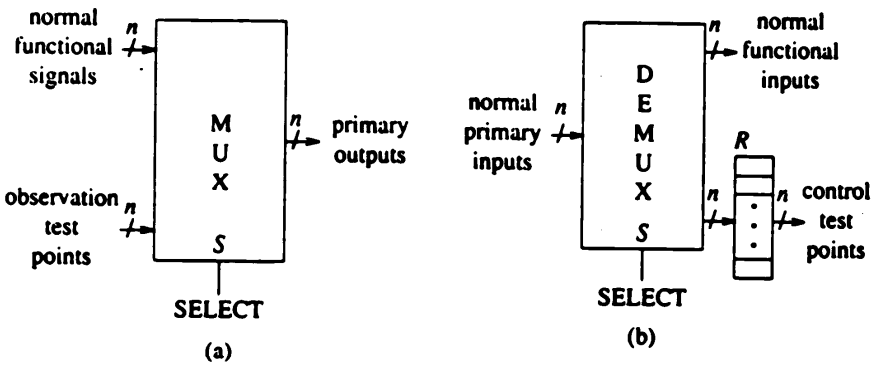
- Major constraint with using test points is large demand on I/O pins.

Reduce Observation points Reduce Control Points



A counter can be used here

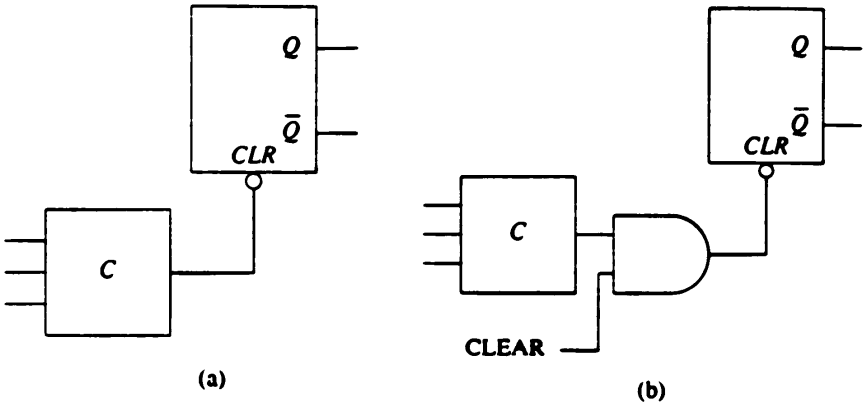
- Time-sharing I/O ports



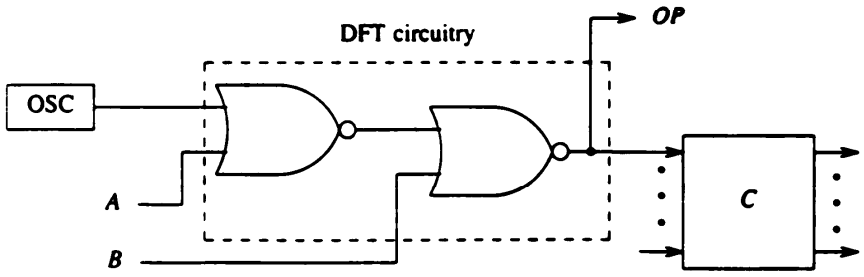
- Good candidates for control points:
  - Control, address, and data bus lines
  - enable / hold inputs to microprocessors
  - enable and read / write inputs to memory
  - clock and preset / clear inputs to memory devices & FFs, counters, shift register
  - data select inputs to multiplexers & demultiplexers
  - Control lines on tristate devices

- Good candidates for observation points:
  - stem lines with high fanout
  - global feedback paths
  - redundant signal lines
  - outputs of devices with many inputs e.g. multiplexers and parity generators
  - outputs of memory devices: FFs, counters, shift registers
  - address, control, and data busses

- Design circuits to be easily initializable

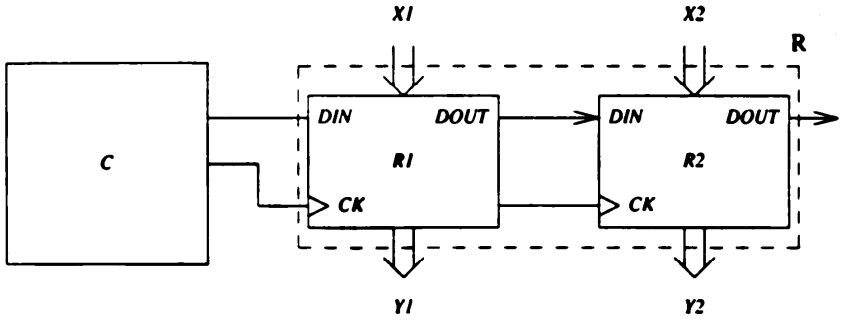


- Disable internal oscillators and clocks during test

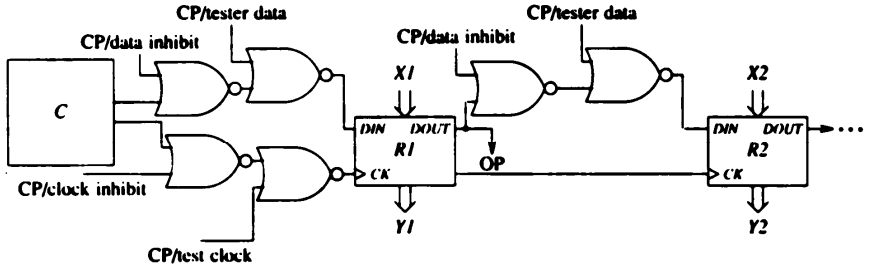


- Partition large counters and shift registers into smaller units

- Example:
  - 16-bit counter could take upto 2^{16} clock cycles to test
  - Partitioned into two 8-bit counters, can be tested with $2 \times 2^8 = 2^9$ clock cycles

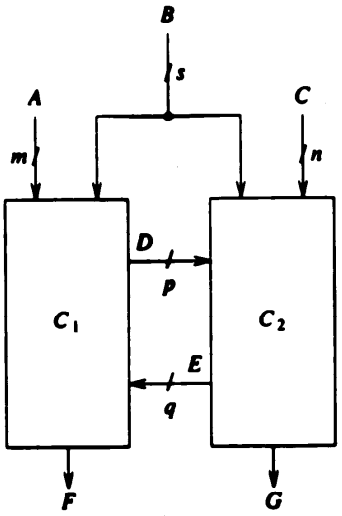


(a)

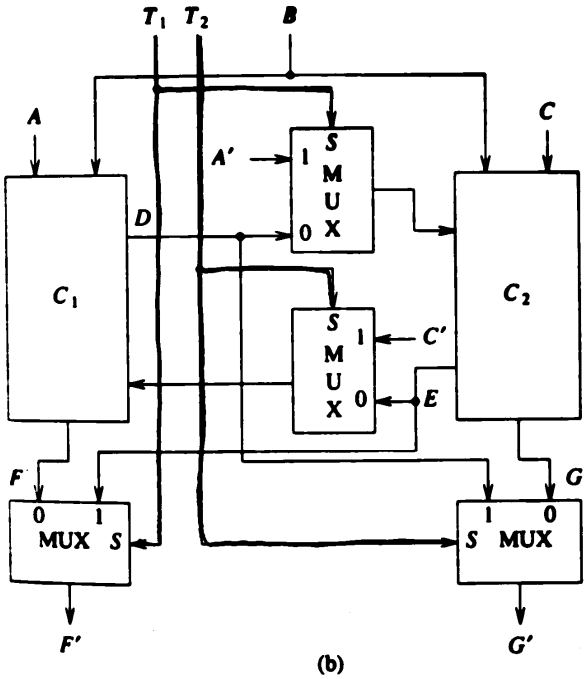


(b)

- Partition large circuits into small subcircuits to reduce test generation cost



(a)

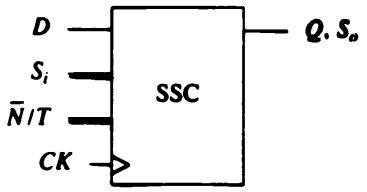


(b)

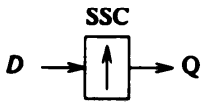
T_1	T_2	Mode
0	0	normal
0	1	test C_1
1	0	test C_2

Enhancing Controllability / Observability by Scan Registers

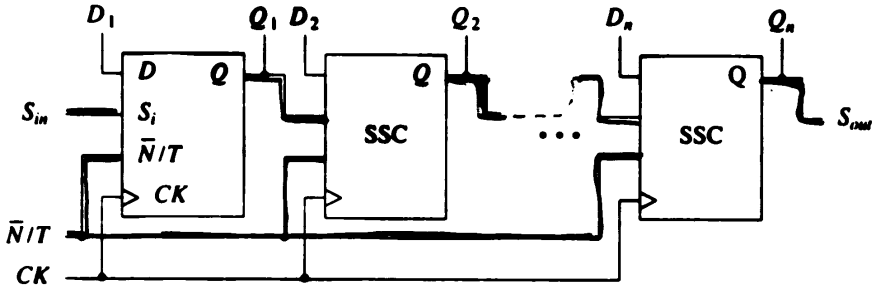
- A scan register (SR) is a register with both shift and parallel-load capability
- Use of scan registers to replace I/O pins is a trade-off between test time, area overhead and I/O pins.



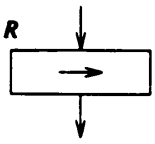
(a)



(b)

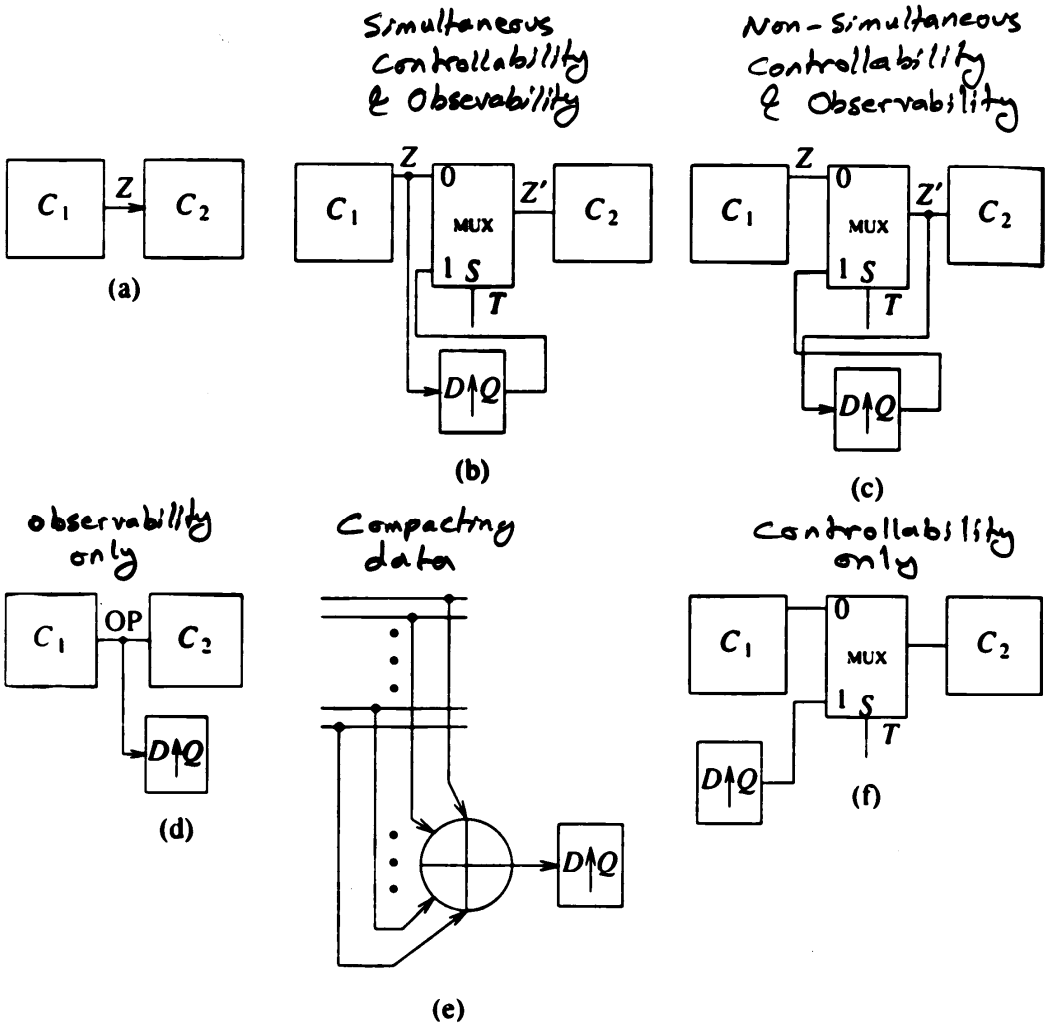


(c)



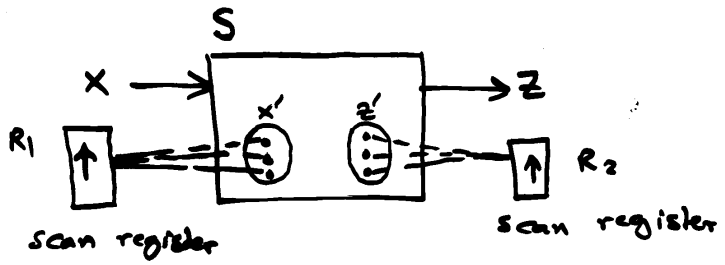
(d)

- Note that T is different from $\bar{N}/T$



Applications

- Consider the circuit S having inputs X and outputs Z.

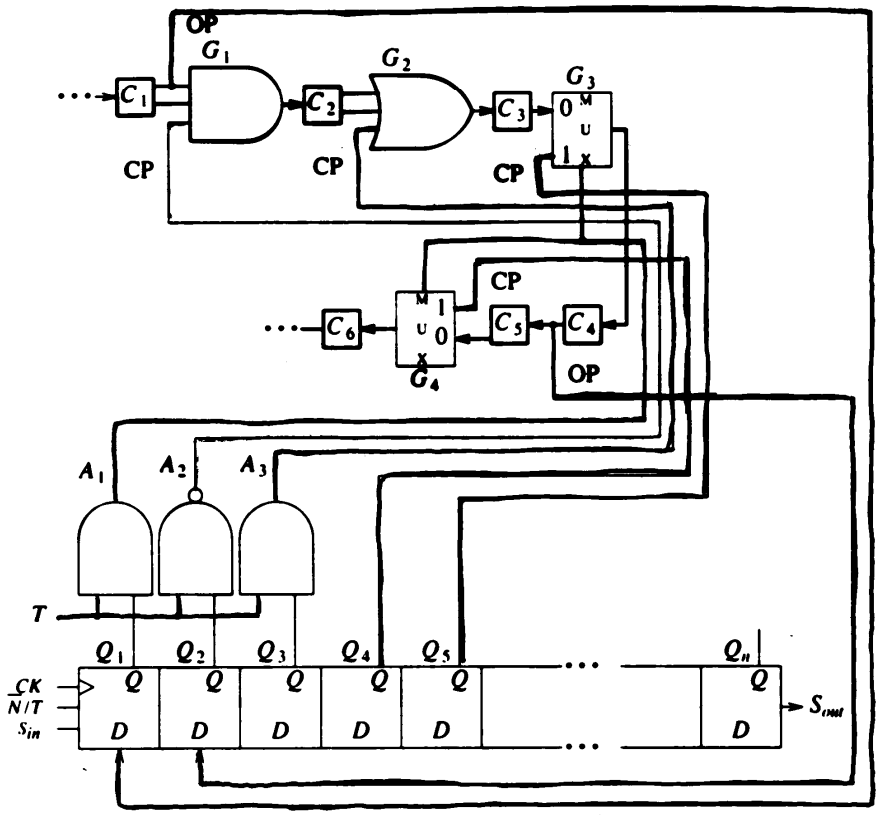


- To enhance controllability, control points added, denoted x'.
- To enhance observability, observation points added, denoted z'.
- Scan register R1 is used to drive x' and scan register R2 is driven by z'.
- x' acts as pseudo-primary inputs and z' as pseudo-primary outputs
- Using x' and z' can significantly simplify ATPG
- Registers R1 & R2 can be combined into a single scan register R where each scan cell in R used as CP and OP.

Adding Controllability and observability

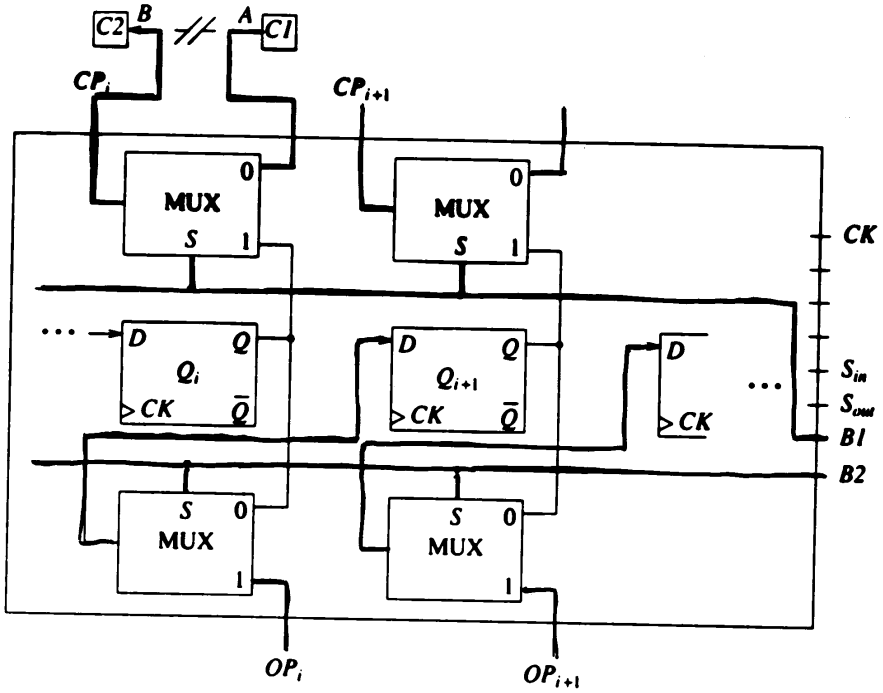
Example

- Circuit modified by inserting 5 CPs & 2 OPs.
- Assume scan register is able to hold its content by disabling its clock
- Five additional I/O pins required : $\bar{N}/T$, Sin, Sout, T, CK



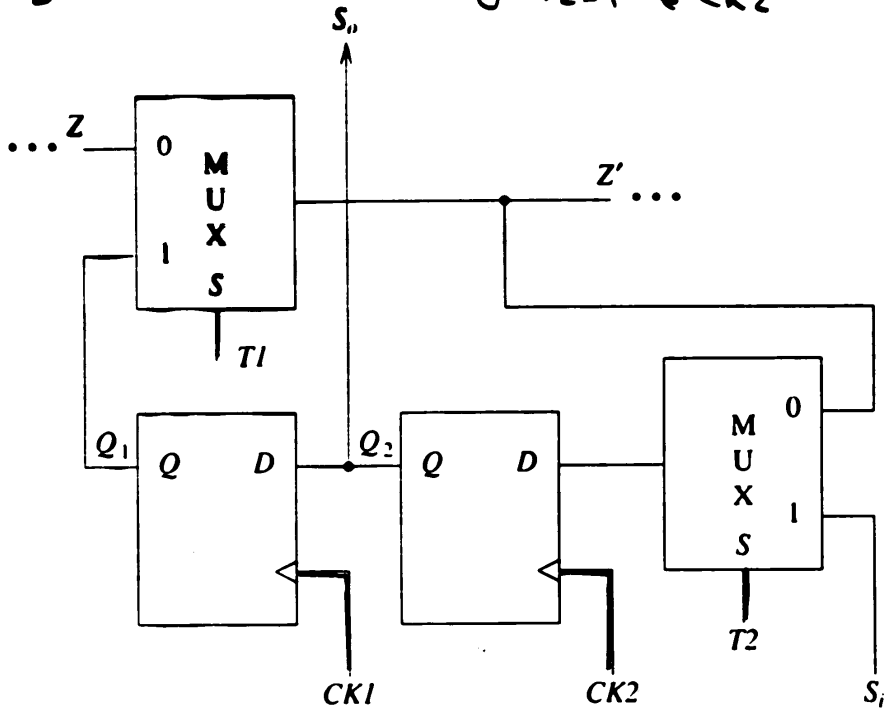
Insertion of CPs & OPs with Scan Chain

- When B1 = 0, normal operation mode
- To scan in data into the scan registers, B2 = 0
- Inject data into CPs from scan register, B1 = 1
- Observe data from OPs to scan register B2 = 1
- Scan out data, B2 = 0

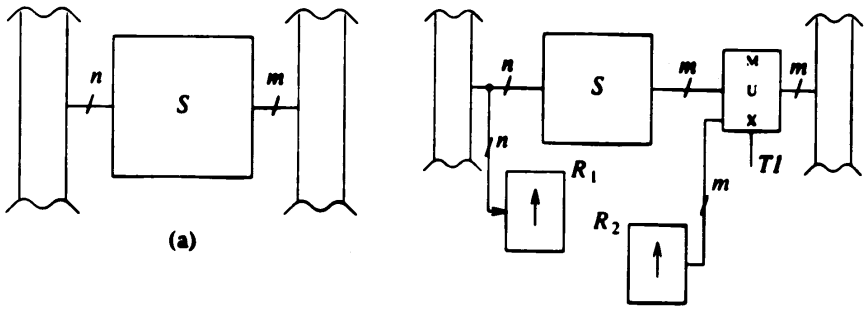


Complex Scan Storage Cell

1. Load scan register with test data by $T2=1$ and clocking CK2
2. Drive UUT to a predefined state with $T1=0$
3. Load Q_2 into Q_1
4. Optional
   - Load Z' into Q_2 by $T_2=0$ & CK2
   - Scan out data in Q_2
5. Inject signals into circuit by $T1=1$
6. Optional : clock out UUT one or more times
7. Observe observation points by $T_2=0$ & CK2
8. Scan out data by $T_2=1$ & CK2



Generic Boundary Scan



- For testing and fault isolation, it is useful to isolate one module from another
- This can be done by boundary scan
  - R1 is used to observe all input data to S
  - R2 used to drive the output lines of S
- All board interconnects can be tested by
  - Scanning in test data into the R2 register of each chip
  - Latching results into the R1 registers
  - Verify results by scanning out R1 registers
- A chip can be tested by
  - loading a test vector into R2 registers driving the chip
  - test results observed by R_1 registers connected at chip output

Boundary Scan Standard

IEEE 1149.1

- A boundary scan cell is inserted in every I/O line
- All scan cells are connected in a scan path

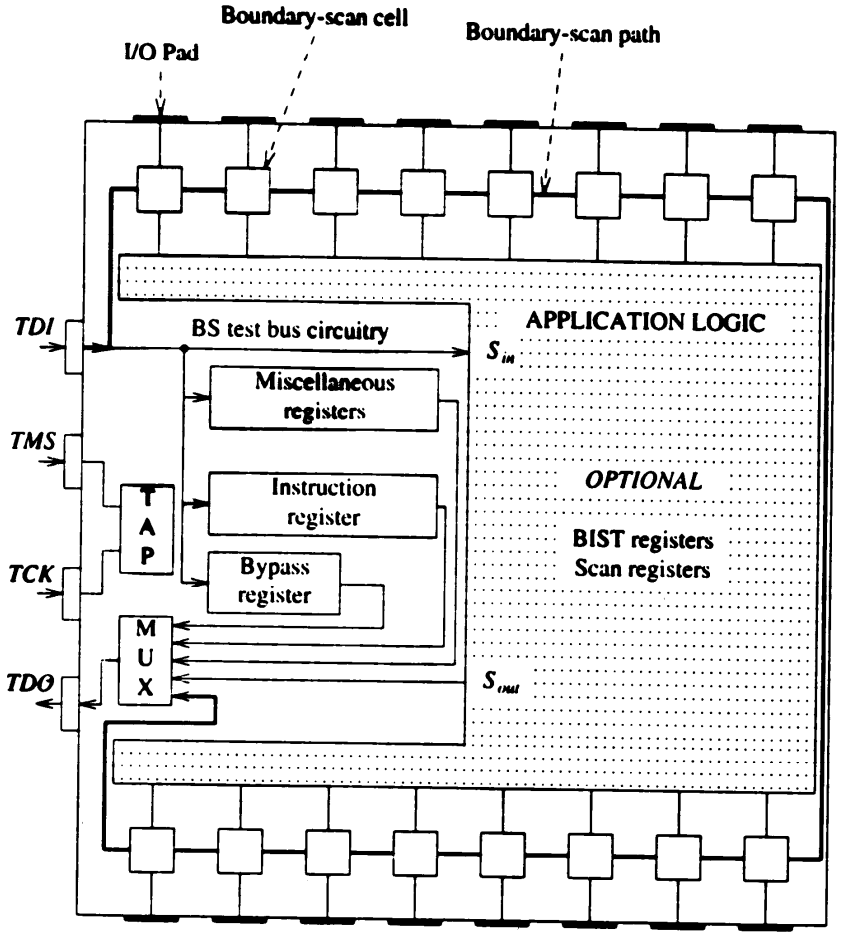
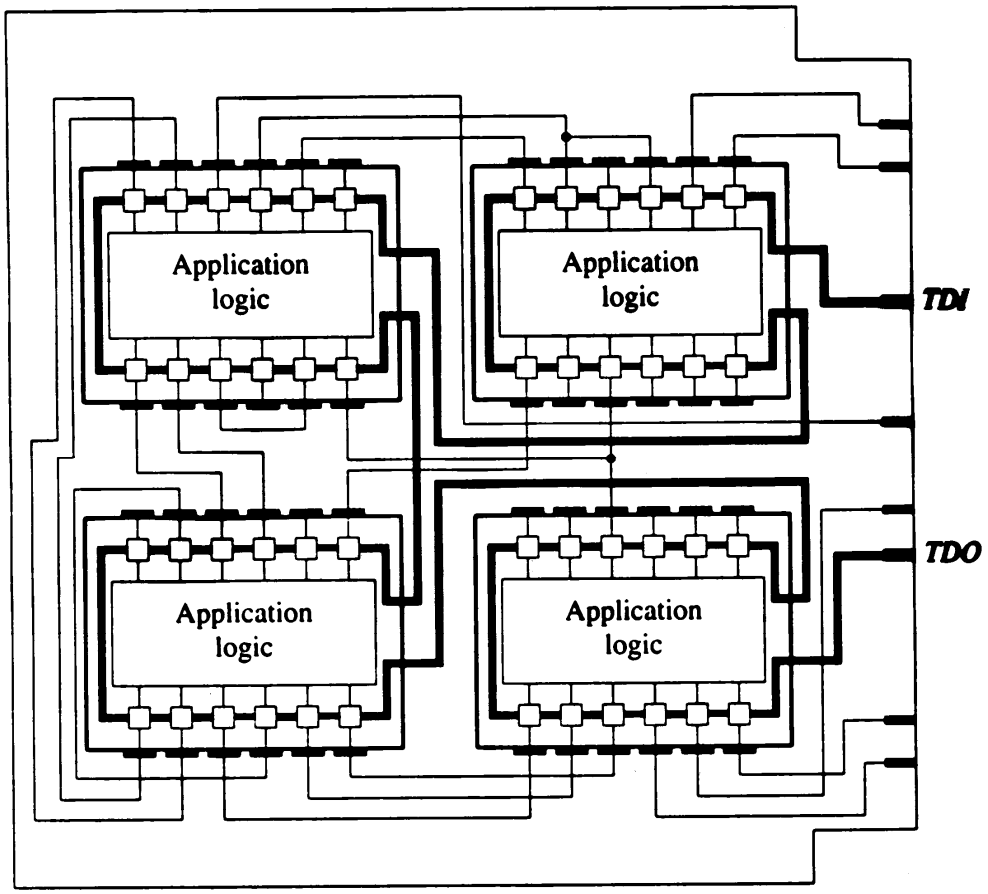


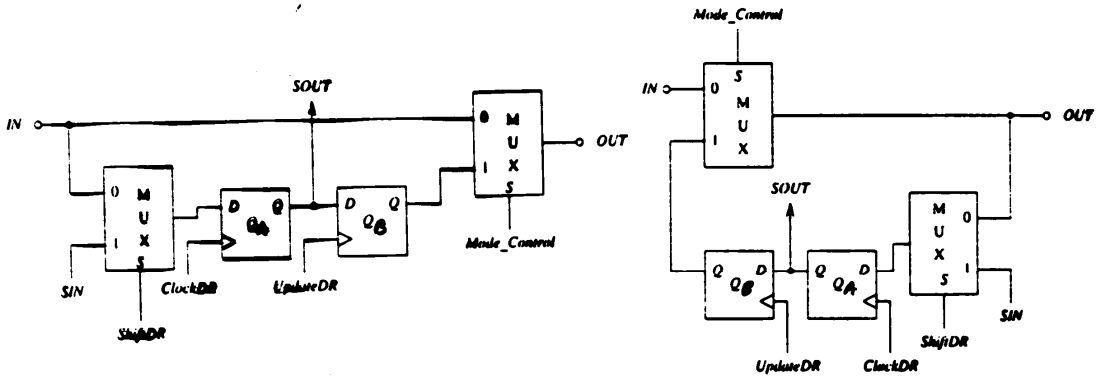
Figure 9.45 Chip architecture for IEEE 1149.1

Printed Circuit Board
with IEEE 1149.1



Boundary Scan Cells

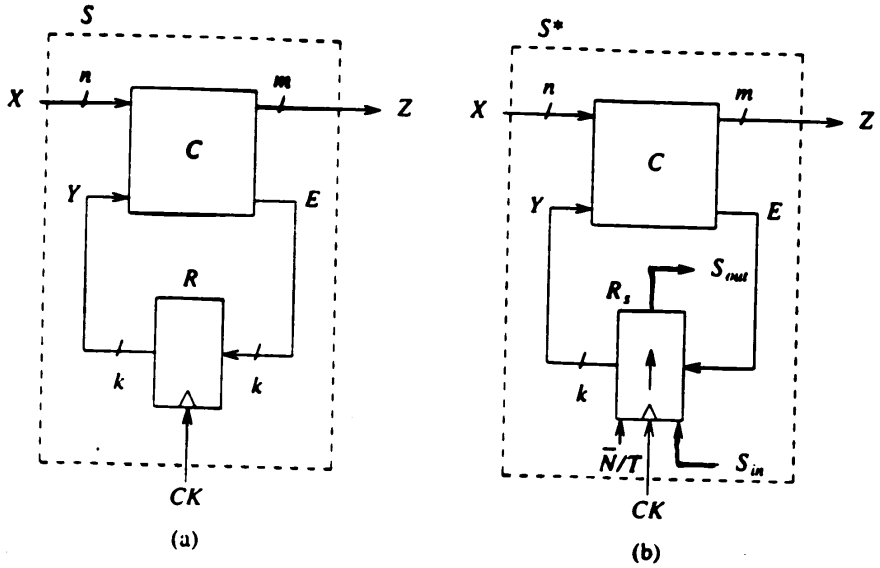
- Can be used as either output or input cells.



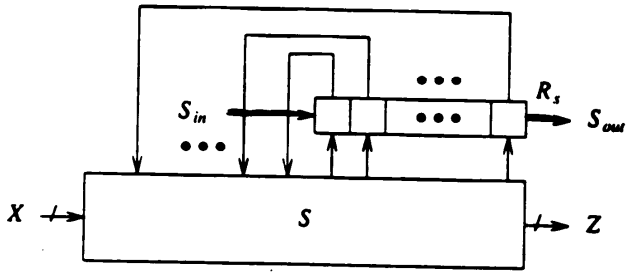
- Normal mode: Mode_control = 0, Idata passes from IN to Out
- Scan mode:
  - Boundary scan cells interconnected into a scan path where SOUT of one cell connected to SIN of next cell in path
  - ShiftDR = 1, and clk pulses applied to clockDR
- Capture mode:
  - Data on line IN can be loaded into scan path by ShiftDR=0 and apply clk pulse to clockDR
- Update mode:
  - Apply data in Q_A to OUT by setting Mode_control=1 and applying clock pulse to UpdateDR

Generic Scan-Based Design

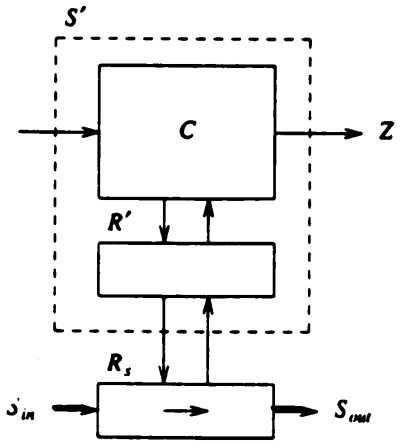
- Full Serial Integrated Scan



- Isolated Serial Scan

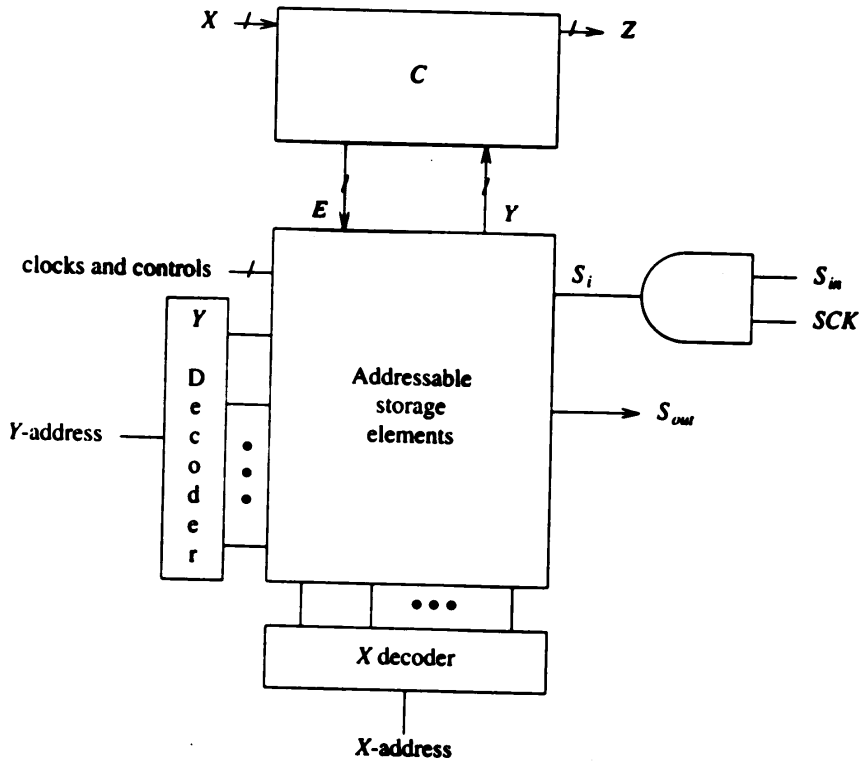


- Full isolated Scan



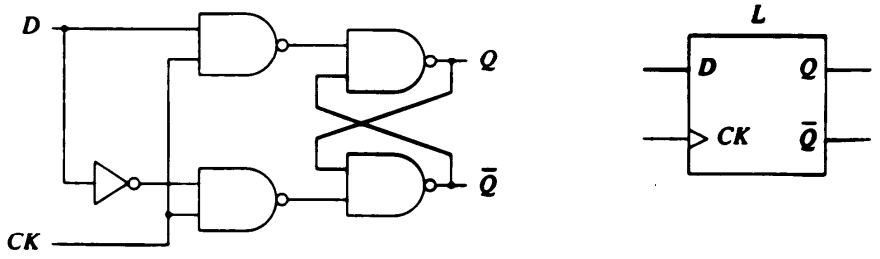
Nonserial Scan

- Storage cells arranged as in a random-access bit addressable memory
- During normal operation, storage cells operate in parallel-load mode
- Major advantage: to scan in a new vector only bits to be changed are addressed & modified
- Considerable overhead for storing cells addresses



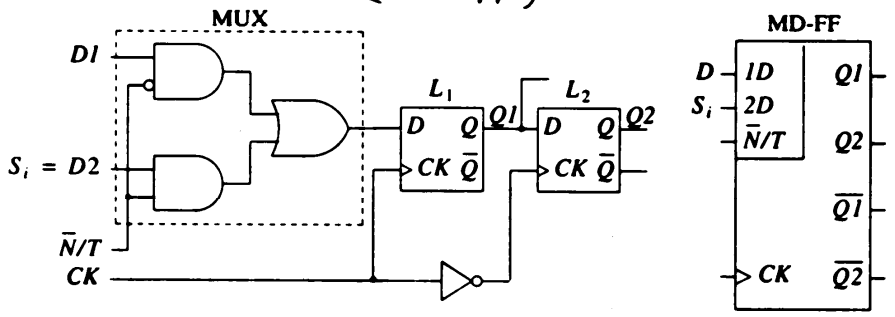
Storage Cells for Scan Design

- Multiplexed data flip-flop (MD-FF)
  - 2-port clocked master-slave FF
  - multiplexor on input
  - When $\bar{N}/T=0$, normal data (D) enter at port 1D
  - When $\bar{N}/T=1$, scan data (SI) enter at port 2D



(a)

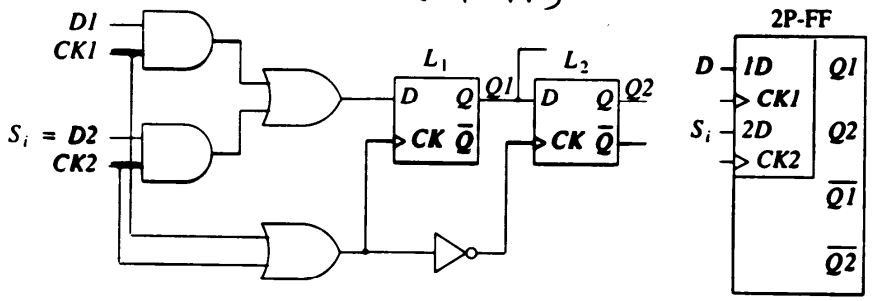
(MD-FF)



(b)

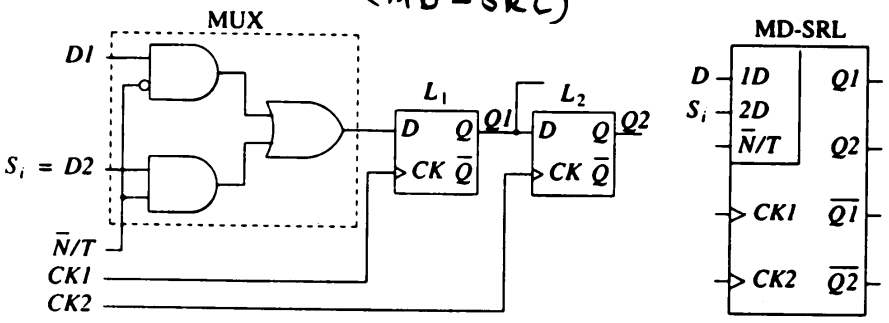
- Two-port clocked flip-flop (2P-FF)
  - Separates system (or normal clock) from scan clock
  - two data ports & two clock ports
- Multiplexed data shift register latch (MD-SRL)
  - Not a FF since each latch has its own clock
  - Avoids races by two-phase nonoverlapping clock

(2P-FF)



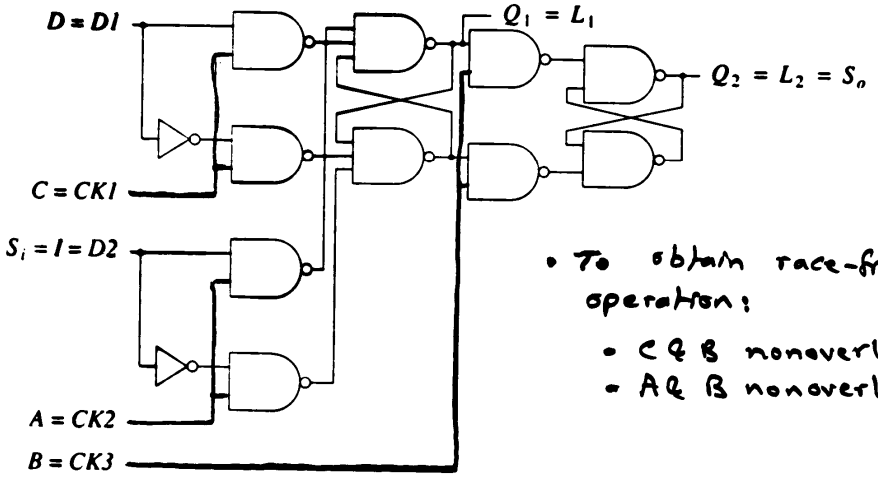
(a)

(MD-SRL)

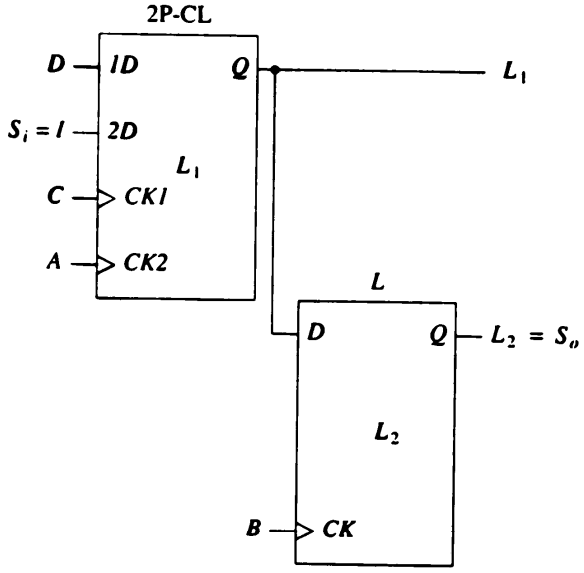


(b)

- **Two-port shift register latch (2P-SRL)**
  - Avoids delay introduced by MUX in MD-SRL
  - Uses three clocks
  - Used in level-sensitive scan design (LSSD) methodology by IBM



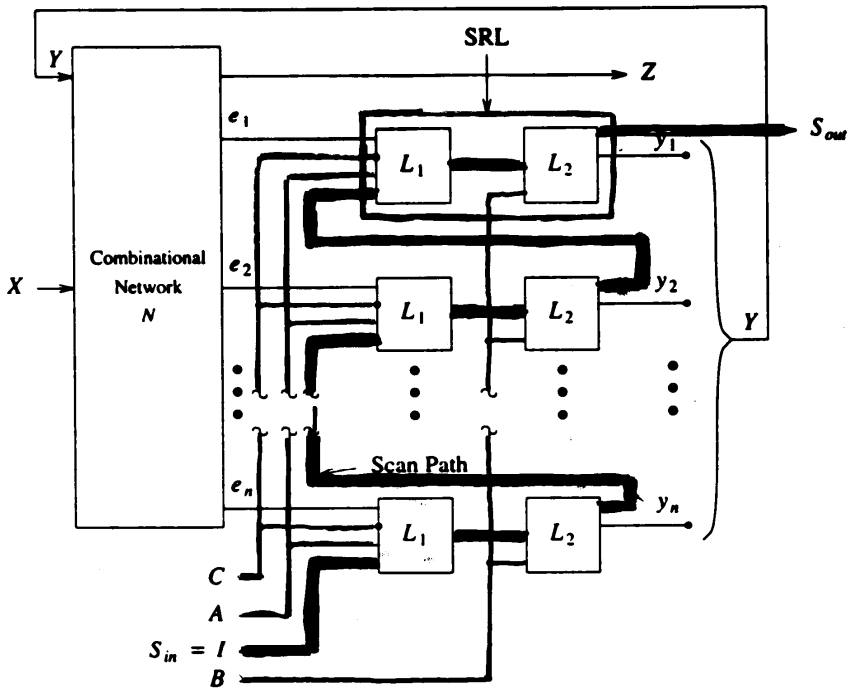
- To obtain race-free operation:
  - C & B nonoverlapping
  - A & B nonoverlapping



Level - Sensitive Scan Design (LSSD)

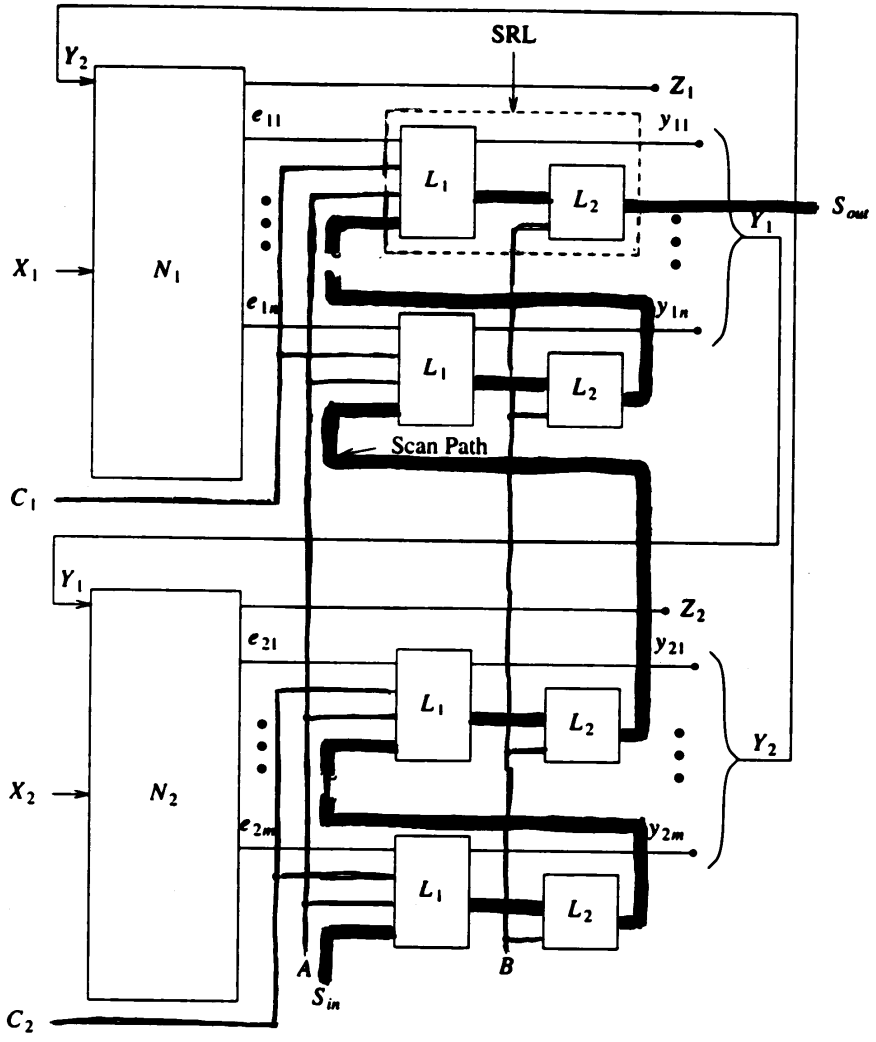
- IBM developed several full serial integrated scan architectures referred to as LSSD
- Normal mode: C and B clocks are used
- Test mode: A and B clocks are used

LSSD Double-Latch Design



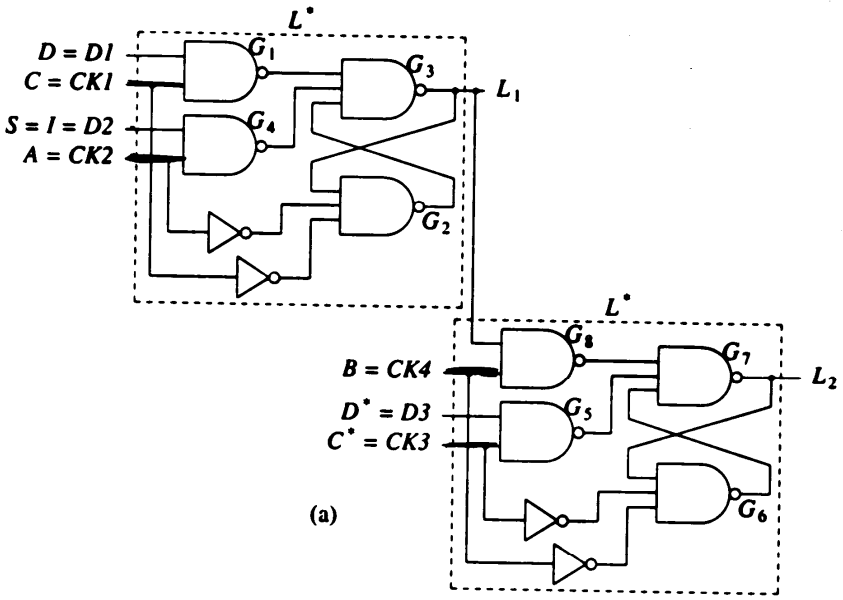
- Design styles with combinational blocks separated by a single latch
- Two system clocks (nonoverlapping) C1 & C2 used

LSSD Single-Latch Design

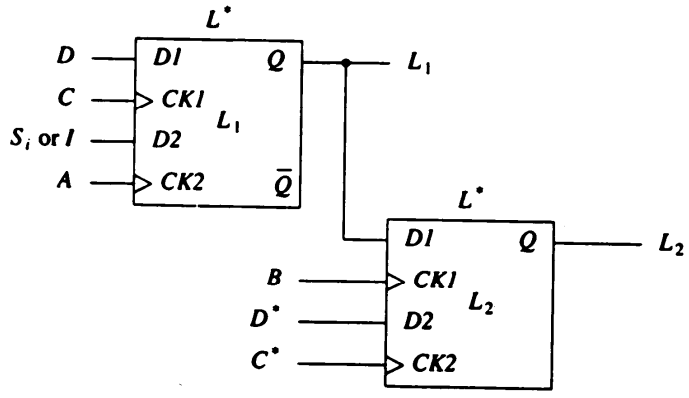


SRL using L_2^* Latch with two data ports

- Employs additional clocked data port D^* and additional clock C^*



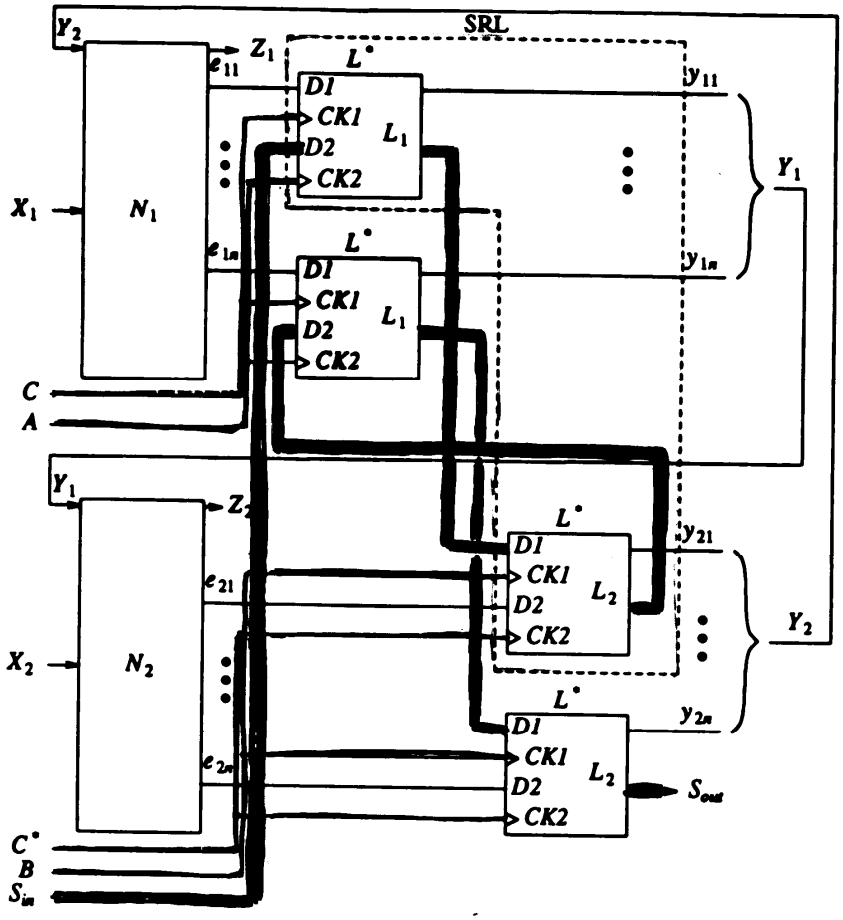
(a)



(b)

- Reduces gate overhead for single latch design
- N1 & N2 cannot be tested simultaneously

LSSD Single-Latch Design
with L_2^* Latch

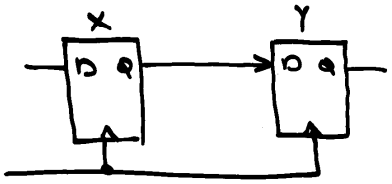


LSSD Design Rules

- Design rules that define LSSD network to insure race-free & hazard-free operation
- A network is level-sensitive iff steady state response to any of input changes is independent of network delays

LSSD Rules

1. All internal storage elements consist of polarity-hold latches
2. Latches controlled by two or more clocks (nonoverlapping) satisfy:
   - A latch X may feed the data port of another latch Y iff the clock for Y does not clock latch X



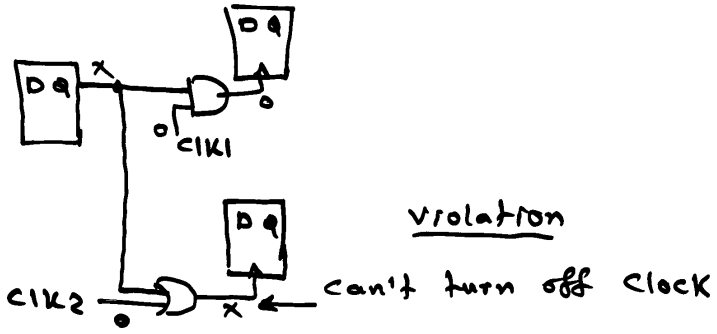
violation

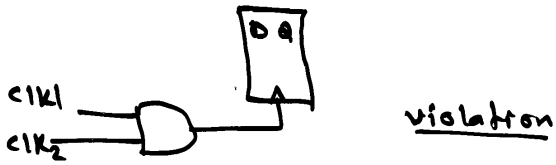
   - A latch X may gate a clock Ci to produce Cig which drives latch Y iff Cig does not clock X

violation

3. There must exist a set of clock PIs from which the clock inputs to all SRLs are controlled such that:

- all clock inputs to SRLs are at "off" state when all clock PIs are at "off" state
- A clock signal at any clock input of SRL must be controlled from one or more clock PIs
- No clock can be anded with either true or complement of another clock





4. Clock PIs cannot feed data inputs of latches directly or through combinat. logic. They can feed primary outputs and clock inputs to latches

The following two rules are used to support scan:

5. Every system latch must be part of SRL. Each SRL must be part of a scan chain.

6. A scan state exists under the following conditions:
   - Each SRL or SO PO is a function of the preceding SRL or SI PI
   - All clocks except the shift clocks are disabled at the SRL inputs
   - Any shift clock to an SRL can be turned on or off by changing corresp. clock PI.

Partial Scan Design

- Partial Scan: subset of storage cells included in scan path
- The maximum sequential depth is largest number of sequential elements from any PI to any PO, where each node is visited at most a single time
- A cycle exists in a circuit when the same node can be revisited starting from that node and traversing circuit forward, not traversing any other node more than once
- Partial Scan Selection
  1. Testability measures (SCOAP):
     - Select the FF that results in the largest reduction in testability cost functions (i.e. controllability & observability)
  2. Structural Cycles or loops:
     - Select the FF that breaks the largest number of loops
     - Select the FF to reduce the maximum length of any cycle in the circuit

3. Sequential depth
   - Select the FF to reduce the maximum sequential depth

4. Consecutive Self-loops
   - Select the FF to reduce the number of consecutive self loops

5. ATPG-based
   - Select a FF that is hard to control
   - Select a FF based on analysis of required states from ATPG
   - Select a FF that improves observability based on fault-simulation & solving a covering problem

- ATPG-based selection can be very time consuming
- A hybrid partial scan selection based on all these techniques might provide the best solution
- Partial Scan Challenges:
  - How much is enough to select to achieve a desired fault coverage

- structural techniques and testability measured based selection are fast but there is no way to find out fault coverage

- Atpg-based selection provides fault coverage information, however:
  - very slow
  - May produce non optimal solutions since produced solution is dependent on fault ordering

- Partial Scan Advantages:
  - Reduce area overhead
  - Avoid scan selection that can affect critical paths ⇒ reduce impact on performance
  - Reduce test application time