Lecture 32

Low Power Testing

Introduction

- Revolution in design paradigm challenges both design and test engineers.
- A circuit or system consumes more power in test mode than in normal mode.
- Can give rise to severe hazard in circuit reliability, or instant circuit damage.
- Increases product cost, difficulty in performance verification, reduced autonomy of portable systems, and decrease in overall yield.

Test challenges

- Escalating transistor counts, increasing chip's complexity, while maintaining its size.
- Testing is one of the most expensive and problematic aspects in a circuit design cycle.
- Traditionally test engineers evaluated test techniques according to area, fault coverage, test application time etc.
- The new class of low power systems make power management a critical parameter that cannot be ignored in test development.

Why test power is higher

- Test efficiency correlates with toggle rate.
- In test mode, switching activity of all nodes is often several times higher than during normal operation.
- Often parallel testing is used in SOCs to reduce test application time.
- DFT circuitry designed to reduce test complexity is often idle during normal operation, but might be intensively used in test mode.
- Correlation between successive functional inputs may be significant, however, for test patterns it is generally kept low.

Ad hoc Solutions

- Oversizing power supply, package, and cooling to withstand increased current.
- Insert breaks into the test process to avoid hot spots.
- Testing with reduced operating frequency.
- System-under-test partitioning.
- Appropriate test planning.

Lecture 33

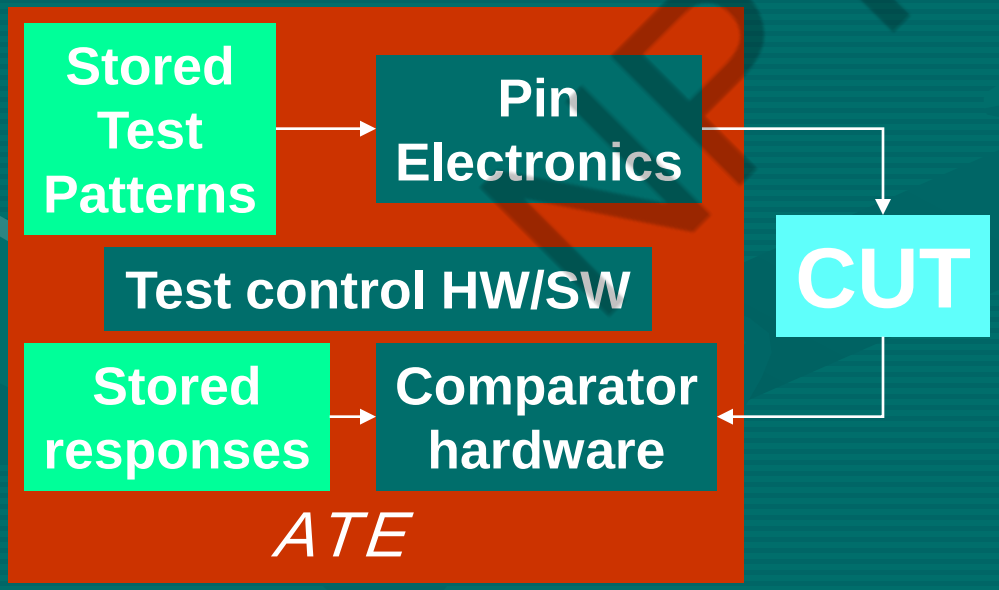
Problems induced by excessive test power

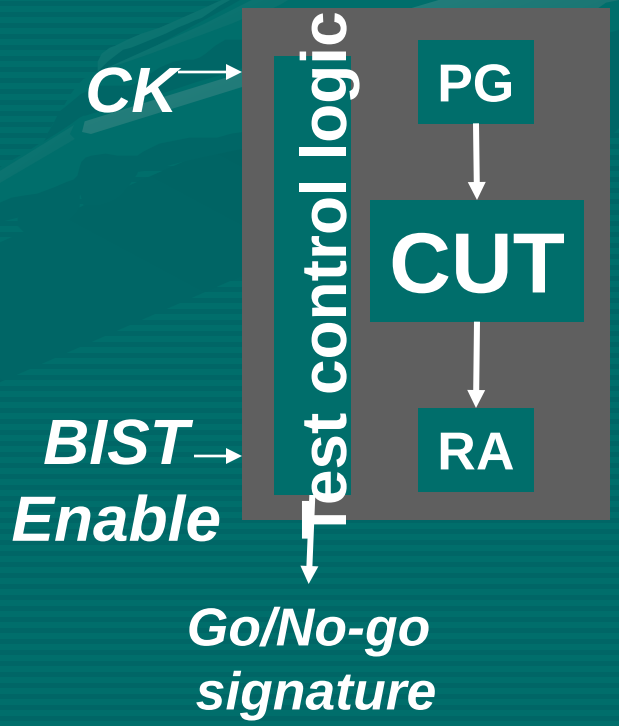
- Test needs to be nondestructive.
- Cost constraints typically require plastic packages.
- Autonomy of battery-powered remote and portable systems suffers from increased activity.
- Aggressive timing makes it essential for tests to identify slow chips through delay testing.
- Just after wafer etching, the unpackaged bare die has very little provision for power or heat dissipation.
- BIST using LFSR-generated test patterns takes longer to reach acceptable level of fault coverage.
- Scan-based testing requires a power-consuming shift operation to provide test patterns and evaluate response.

Internal Testing

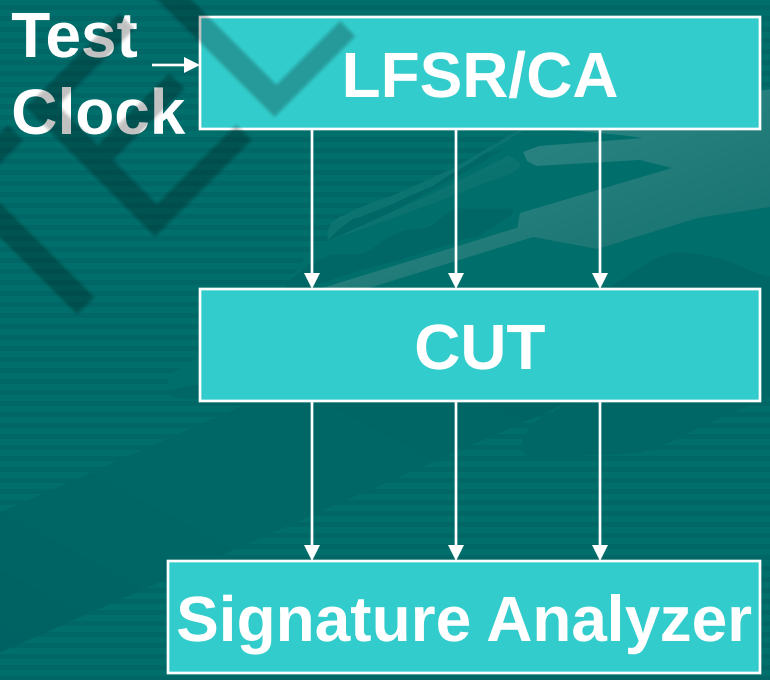
Built-In Self-Test

- Implement the function of automatic test equipment (ATE) on circuit under test (CUT).
- Hardware added to CUT:
  - Pattern generation (PG)
  - Response analysis (RA)
  - Test controller





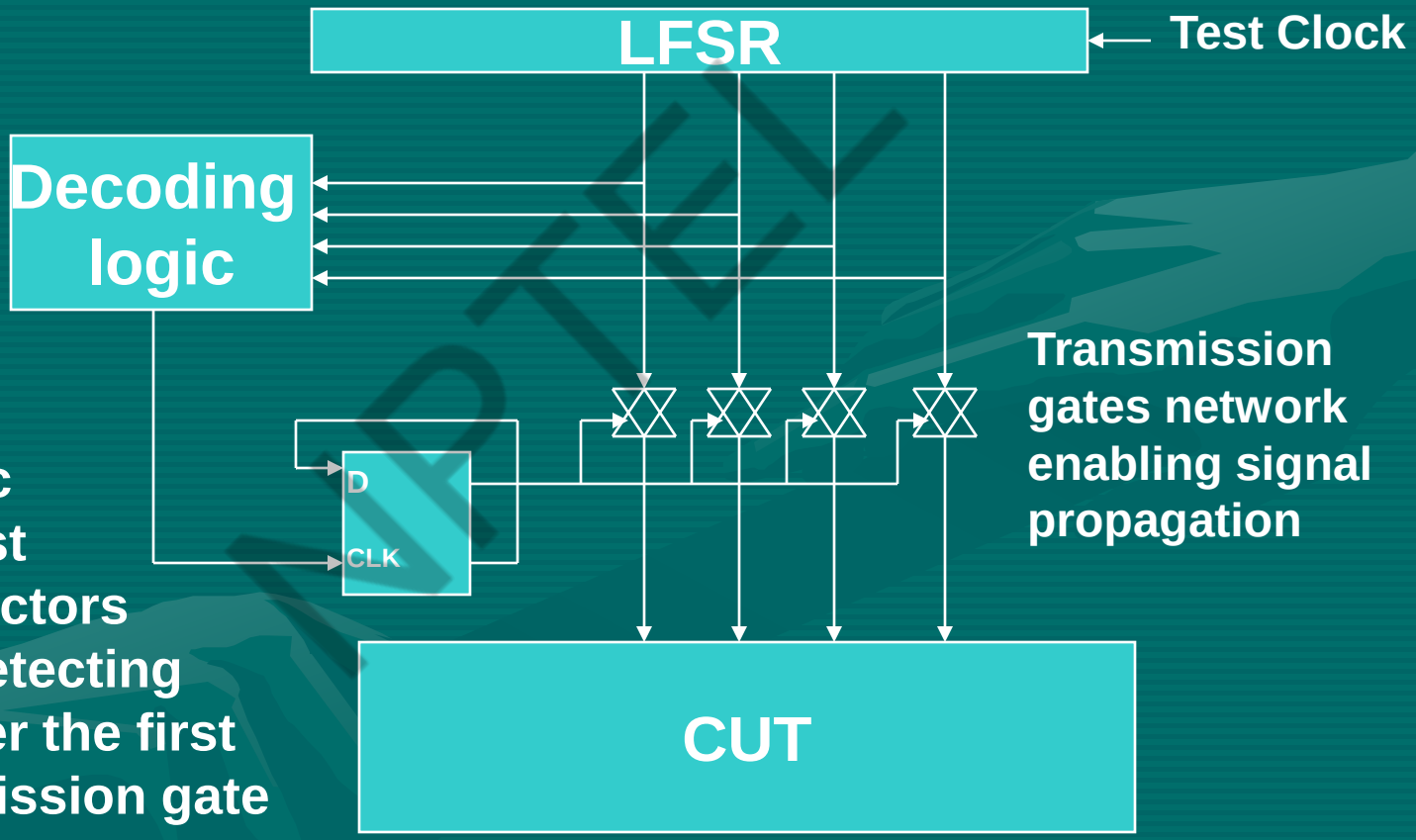
- A BIST environment consists of
  - a test pattern generator using LFSR/CA
  - A signature analyzer
- Once initiated, the generator produces a pseudo-random sequence of patterns applied to the circuit, signature analyzer analyzes the response
- A large number of non-detecting patterns are generated, consuming lot of power



BIST Techniques

- Vector filtering
- Reseeding the LFSR
- Precomputation based schemes
- Dual speed LFSR
- LFSR tuning
- Augmenting LFSR structure with another LFSR or shift register
- Circuit partitioning

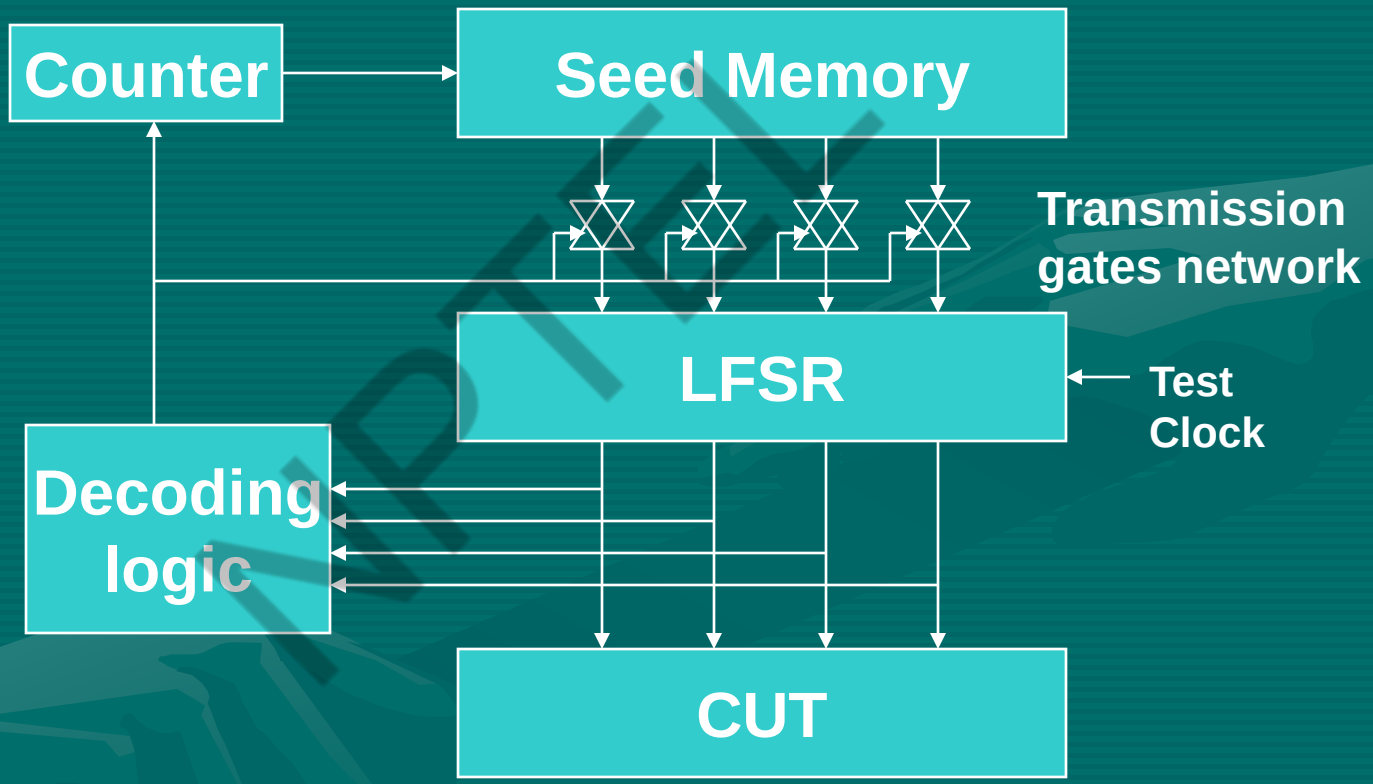
BIST: Filtering vectors



Transmission gates network enabling signal propagation

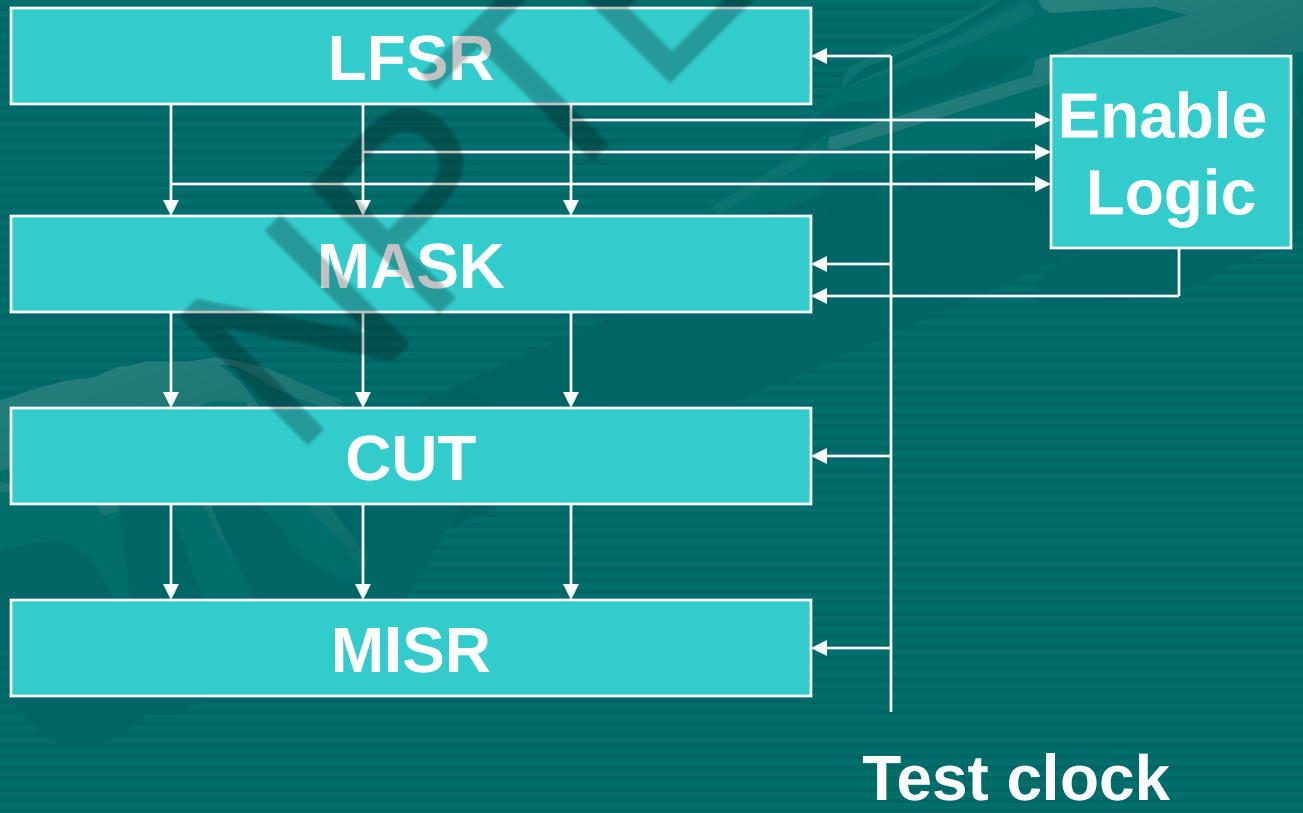
Decoding logic detects the first and the last vectors of each non-detecting sequence. After the first vector, transmission gate network inhibits signal propagation

BIST: Re-seeding LFSR



BIST : Precomputation based

- MASK may be latch based or AND-based
- Enable logic implements an incompletely specified Boolean function whose on-set is the set of unaltered vectors and off-set is the set of eliminated vectors



BIST: Dual Speed LFSR

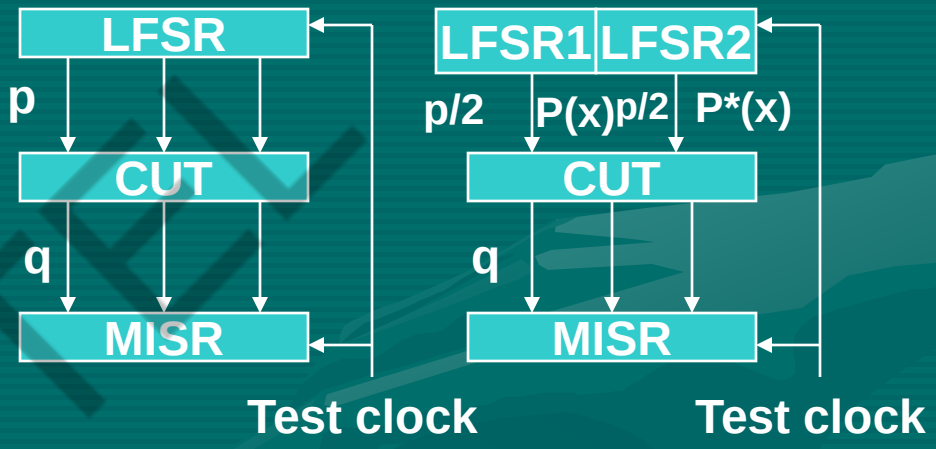
- Uses two different speed LFSRs
- Connect the inputs with elevated transition densities to slow-speed LFSR
- Test efficiency of DS-LFSR is higher than normal LFSR

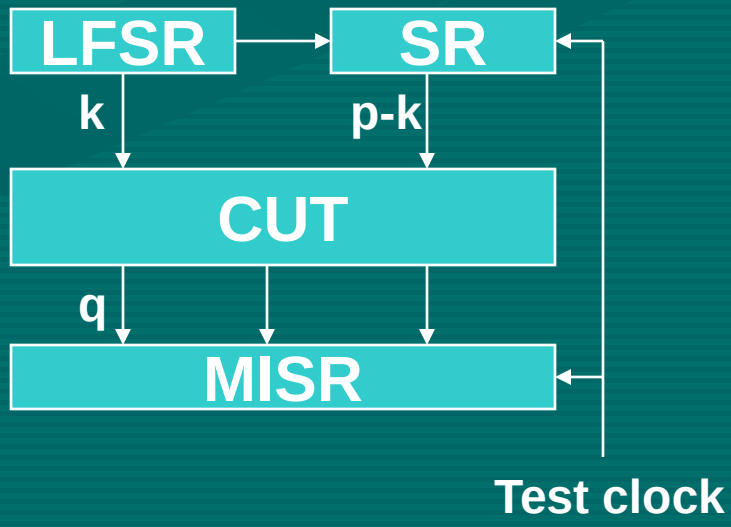
BIST: LFSR Tuning

- Parameters of LFSR:
  - Polynomial
  - Seed value
- Polynomial does not affect the power consumption significantly
- Seed selection plays a vital role in determining the power consumption

Different single source/single sink BIST architectures

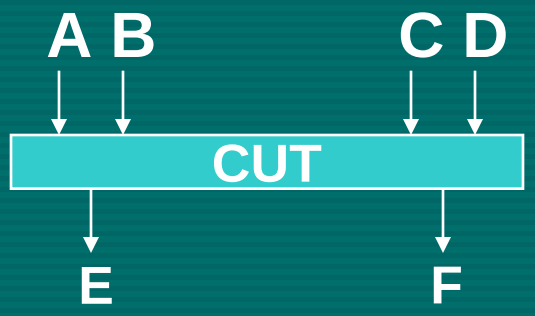
- An LFSR with a shift register produces lower power dissipation, BIST area overhead and test application time when compared to a single LFSR and two LFSRs with reciprocal characteristic polynomials
- Lower fault coverage



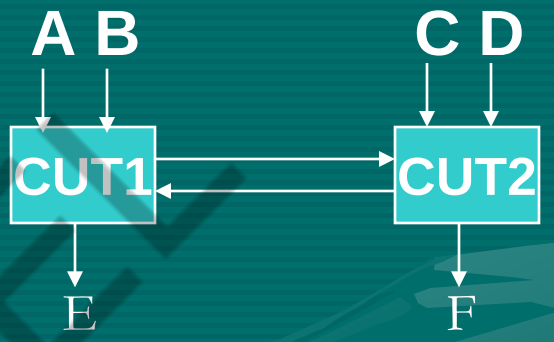


Lecture 34

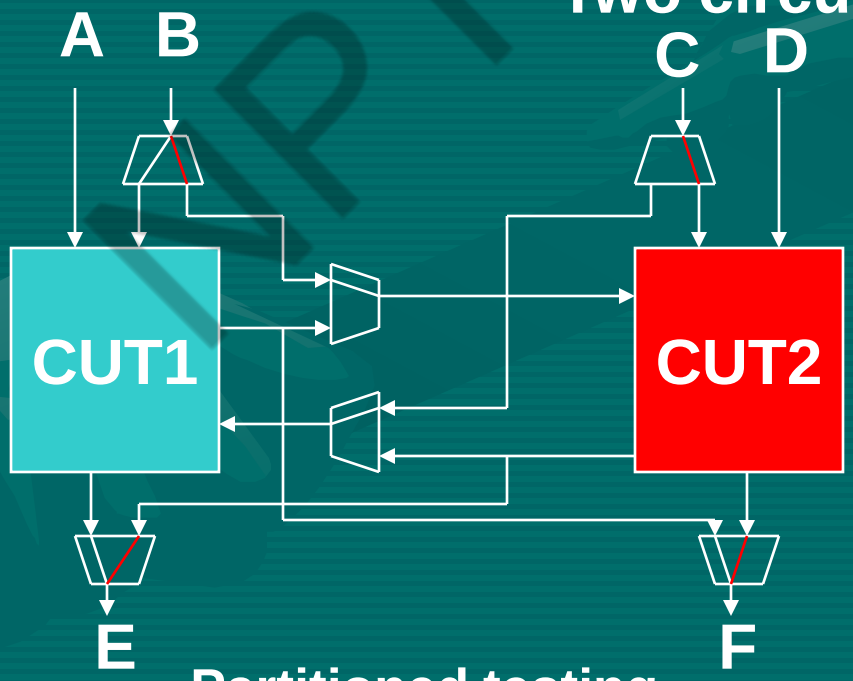
BIST: Circuit partitioning



Initial Circuit



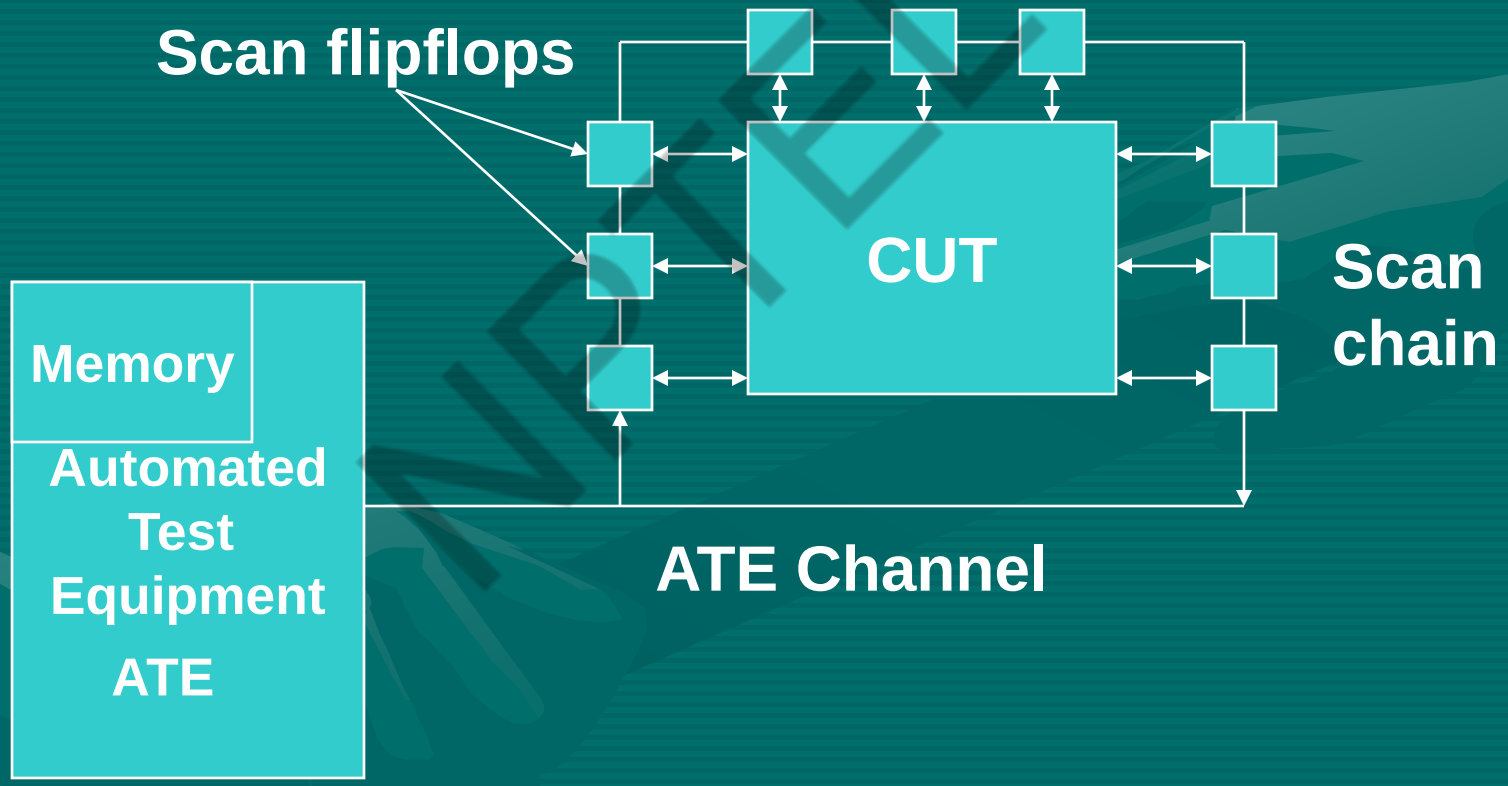
Two circuit partitions



Partitioned testing

External Testing

External Testing



Low Power External Testing Techniques

- Low power ATPG algorithms
- Ordering techniques
- Scan power minimization
  - Test vector ordering
  - Scan flip-flop ordering
- Input control
- Vector compaction
- Test data compression
- Best primary input change time
- Scan chain transformation
  - Scan flip-flop ordering
  - Using both true and complemented outputs of scan flip-flops
  - Using T/D flip-flops
  - Inserting XOR gates at selected positions
  - Multiple scan chain

Low Power ATPG

- Modify ATPG algorithm (like PODEM) to fill-up the don't cares judiciously so that the number of transitions between two successive test vectors is minimized.
- Generating redundant set of patterns
  - A number of patterns are generated for each fault
  - For sequential circuit testing, patterns are to be applied in a sequence only.
  - Each such sequence may have different power requirements
  - Select those sequences requiring lesser power, without sacrificing fault coverage.

Low Power ATPG (contd.)

- For combinational circuits
  - Generate a set of redundant patterns
  - Identify faults detected by each of them
  - Compute the number of gate transitions for each pair of test vectors
  - Perform a search for a pattern set with low power and high fault coverage with a proper re-ordering of the set
  - Genetic algorithm may be used to explore the search space efficiently

Vector Reordering

A	B	C	X	Y
0	0	0	0	0
0	0	1	0	1
0	1	0	0	0
0	1	1	0	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

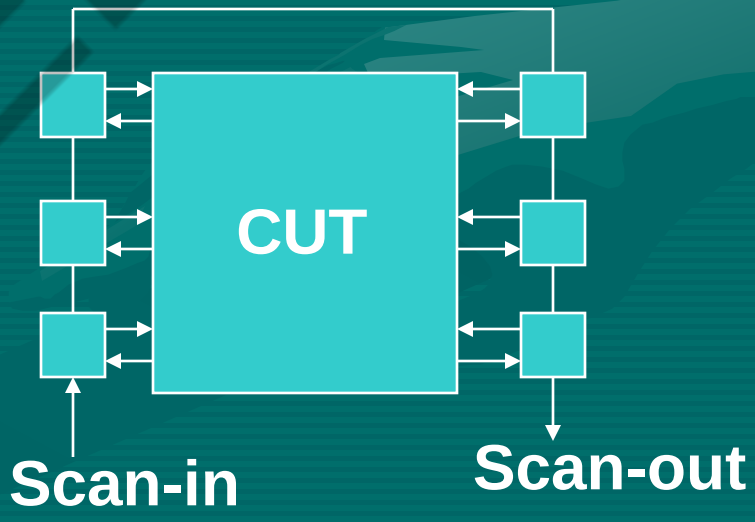
No. of transitions = 6

A	B	C	X	Y
0	0	0	0	0
0	1	0	0	0
1	0	0	0	0
0	0	1	0	1
0	1	1	0	1
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

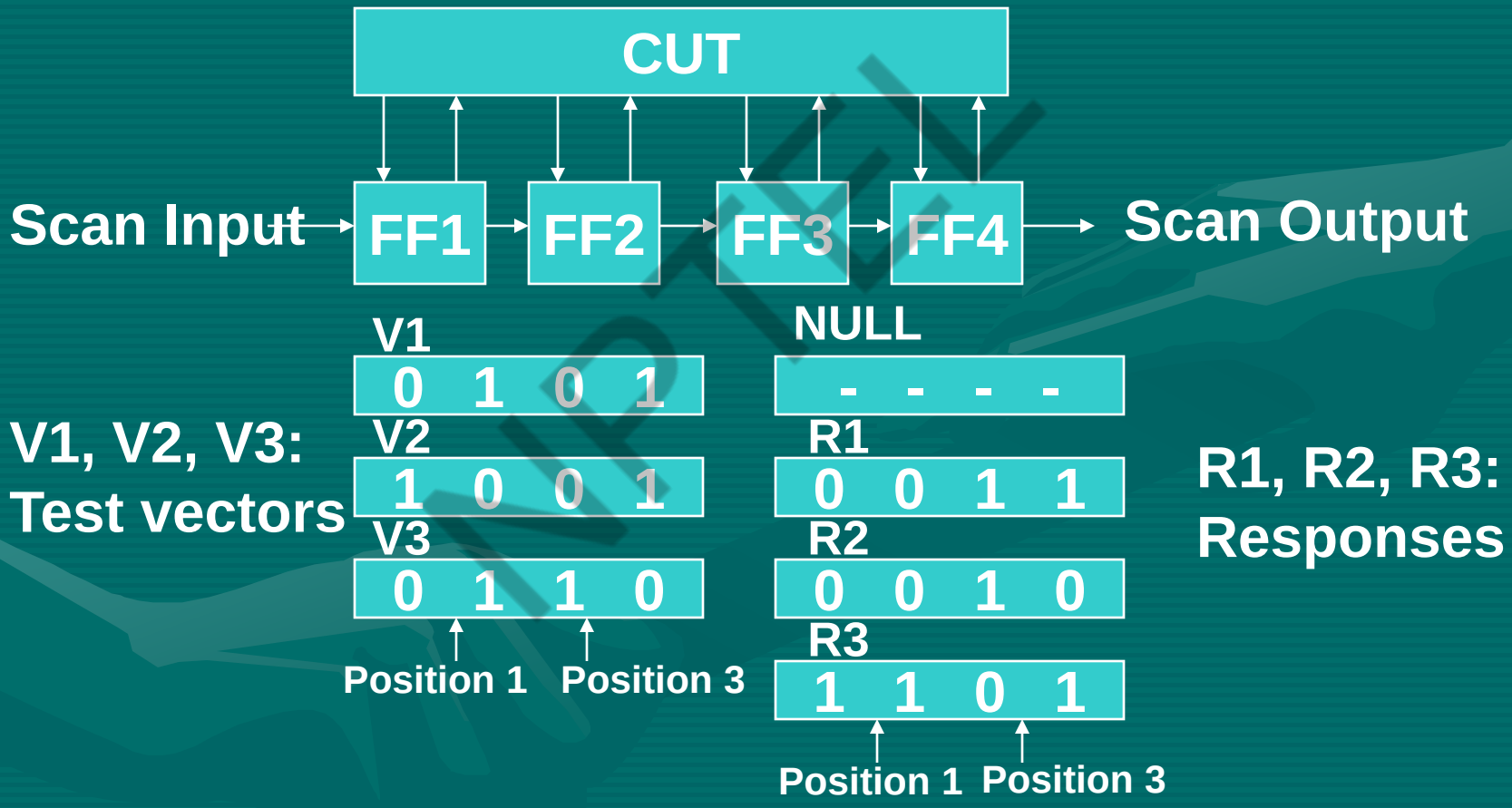
No. of transitions = 2

Scan Power Minimization

- A good amount of power is consumed when test patterns and responses are shifted through the scan chain
- Ripples cause lot of transitions in the circuit inputs which in turn creates transitions in the circuit gates
- Two solutions:
  - Test vector ordering
  - Scan flip-flop ordering



Weighted Transition Metric

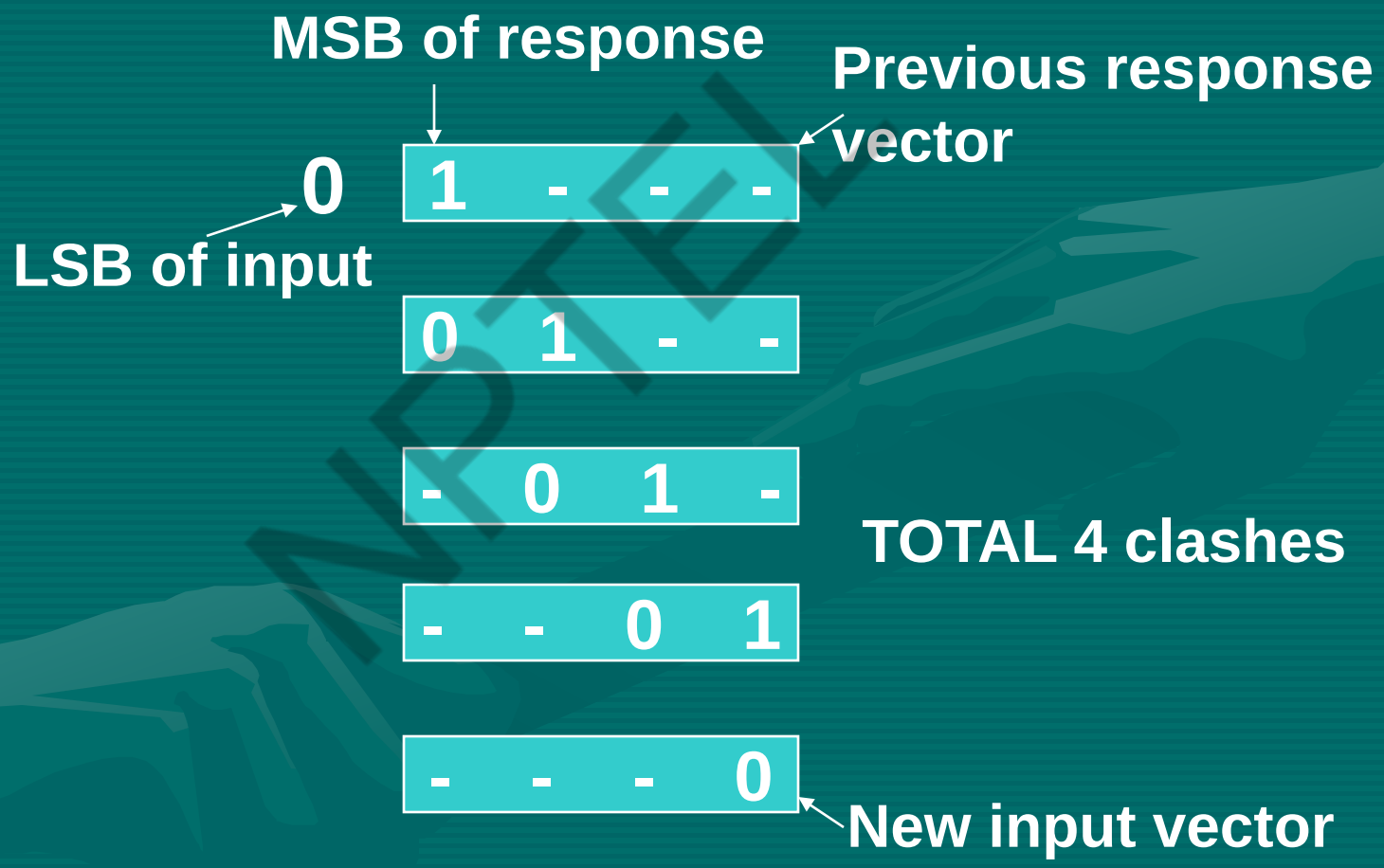


Weighted Transition Metric (contd.)

- Weighted transitions = Σ (Size_of_scan_chain – Position_of_transition)
- For scan vector “0001”, weighted transitions = (4 – 1) = 3
- For output response “0001”, weighted transitions = (4 – 3) = 1

Lecture 35

Clash


MSB of response
Previous response vector
0
LSB of input
1 - - -
0 1 - -
- 0 1 -
- - 0 1
- - - 0
TOTAL 4 clashes
New input vector

Clash (contd.)

- A clash occurs when the MSB of the last response differs from the LSB of next test vector
- Number of transitions created by a clash is equal to the size of the scan chain
- Thus, total number of transitions =

(Number of weighted transitions)
+
(Number of clashes) X (Size of scan chain)

Test Vector Ordering

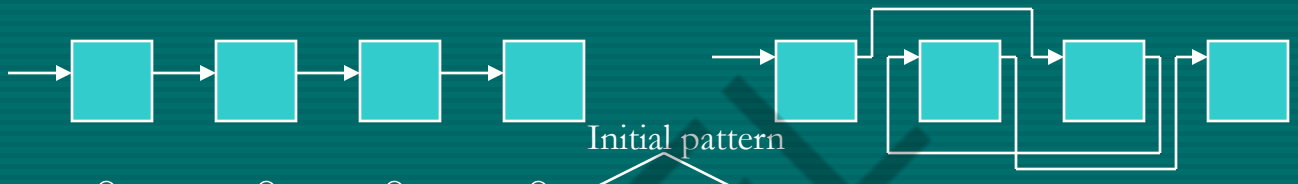
- Can be applied to reduce inter-vector transitions, that is clashes
- Test vectors are grouped into four distinct categories based on first test vector bit and the last response bit

Group	Test vector bit	Response bit
I	0	0
II	0	1
III	1	0
IV	1	1

Test Vector Ordering (contd)

- The sequence of patterns should be as follows.
  - Apply patterns belonging to Group I (00)
  - Apply patterns belonging to Group IV (11)
  - Repeat until any group exhausts
    - Apply a pattern from Group II
    - Apply a pattern from Group III
  - Apply remaining patterns

Scan Cell Reordering



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

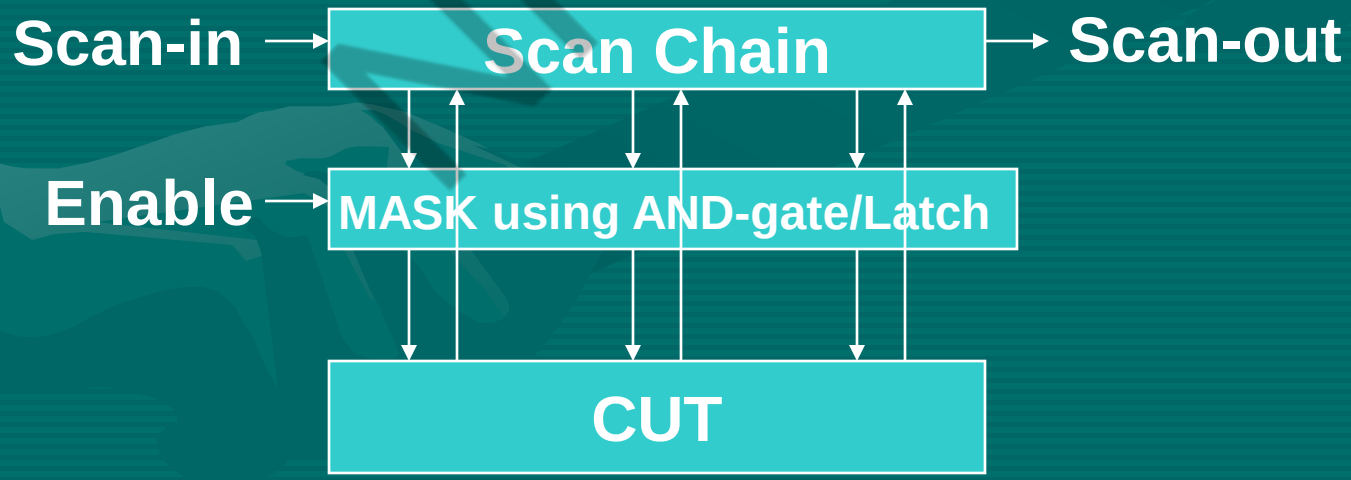
No. of transitions: 6

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

No. of transitions: 2

Input control

- Identify an input control pattern for a full-scan circuit.
- The pattern is applied to the circuit's primary inputs during scan.
- This minimizes or eliminates switching activity in the combinational part of the circuit.
- Can be used in conjunction with vector- and/or scan-reordering.



Vector Compaction

- Initial set of test vectors generated by an ATPG tool is huge
- To reduce the test application time, the tools use a compaction of test vectors
- In conventional static compaction, test cubes are merged randomly
- Selecting the merging order of the test cube pairs can reduce both the average and peak power

Test Data Compression

- Primarily used to reduce
  - Volume of test data to be stored in the ATE memory
  - Transfer time of test patterns from ATE to the chip
- Test vectors are kept in a compressed form
- Special codes like Golomb codes, Huffman codes etc. are used to compress the test vectors
- Coding strategy often fills up the don't cares in the patterns in such a way that the occurrence of similar code words increases, increasing the compression ratio
- On-chip decoder decompresses the code words to get back the patterns to be sent via the scan chain
- Can be traded off to reduce the scan power simultaneously

Golomb Coding

Group	Run-length	Group prefix	Tail	Code word
A1	0	0	00	000
	1		01	001
	2		10	010
	3		11	011
A2	4	10	00	1000
	5		01	1001
	6		10	1010
	7		11	1011
A3	8	110	00	11000
	9		01	11001
	10		10	11010
	11		11	11011
...	...	...	...	...

An example:

0 1 — 001

1 — 000

1 — 000

1 — 000

1 — 000

0 0 0 0 0 1 — 1001

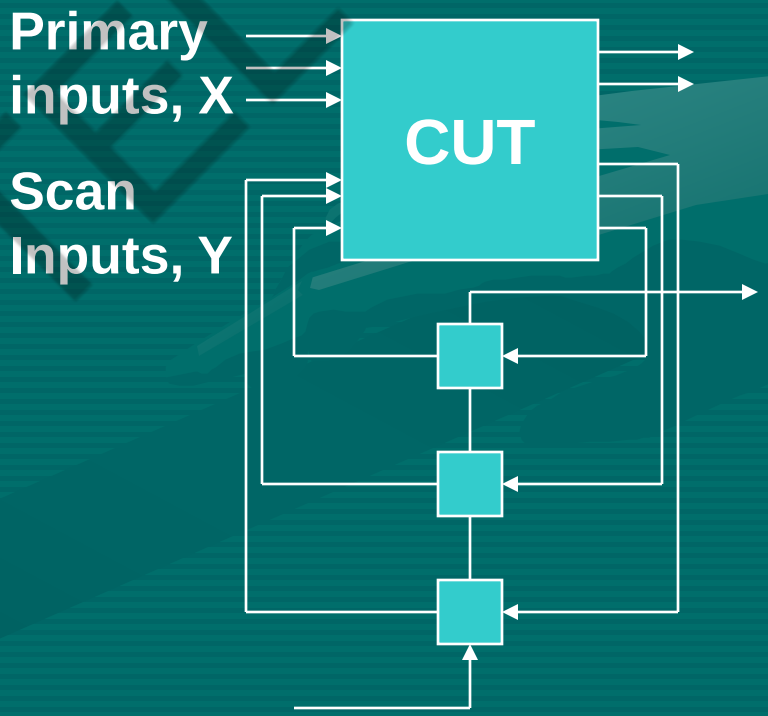
Total: 19 bits

Power Compression Trade-off in Golomb Coding

Partially-specified scan vector	Fully-specified vector (Minimim WTM)	Fully-specified vector (Don't cares mapped to 0s)
01XXX10XXX01	011111000001 Golomb code length: 19 bits (m=4) WTM = 18	010001000001 Golomb code length: 10 bits (m=4) WTM = 25
01X1010XXXX1	011101011111 Golomb code length: 27 bits (m=4) WTM = 23	010101000001 Golomb code length: 13 bits (m=4) WTM = 32

Best Primary Input Change Time

- Each test vector $V_i = <X_i, Y_i>$, where X_i is the primary input part and Y_i is the pseudo input fed via scan chain
- For *m* scan cells, Y_i is shifted in *m* clock cycles
- Primary input can be changed at any of these *m* cycles without affecting test efficiency
- When should the primary input be changed to minimize number of transitions in the circuit



Lecture 36

Drawbacks of BPIC

- Computation of BPIC time is dependent on the size and the value of the test vectors
- May result in high computation time limiting applicability of BPIC to reasonably large circuit
- Test set independent strategies are preferred e.g. multiple scan chain

Scan Chain Transformation

- Several alternatives proposed:
  - Scan flip-flop reordering
  - Using TRUE and COMPLEMENTED outputs
  - Using D/T type flip-flops
  - Inserting XOR gates in the scan chain
  - Multiple scan chain based design

Using Both Outputs

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

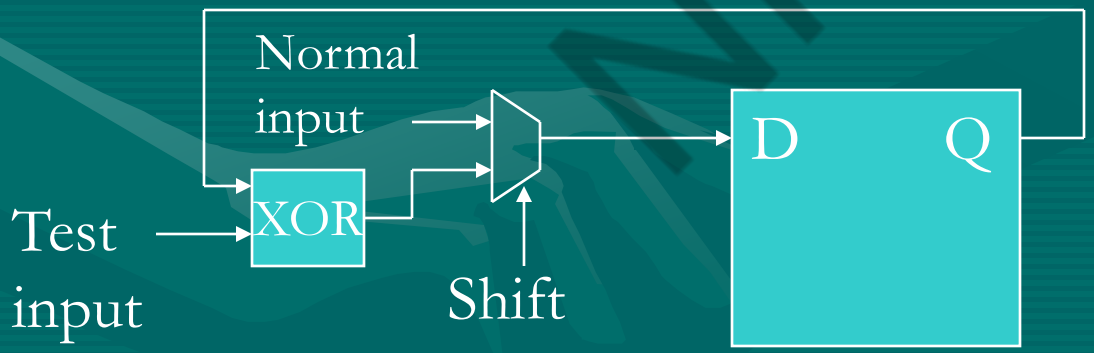
No. of transitions: 4

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

No. of transitions: 3

Using T and D Flip-flops

- D flip-flop can be changed to T flip-flop during scan-in operation
- To capture the response, it remains D flip-flop
- Response part will get modified while being transmitted through the scan chain



D-T Flip-flop - example

D	D	D	D
0	0	0	0
1	0	0	0
0	1	0	0
1	0	1	0
0	1	0	1

No. of transitions: 9

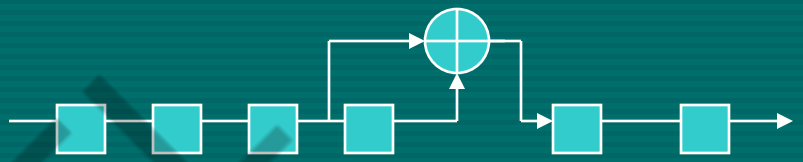
D	T	T	T
0	0	0	0
1	0	0	0
0	1	0	0
0	1	1	0
0	1	0	1

No. of transitions: 6

Inserting Multiple XOR Gates

Test stimulus transformation matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

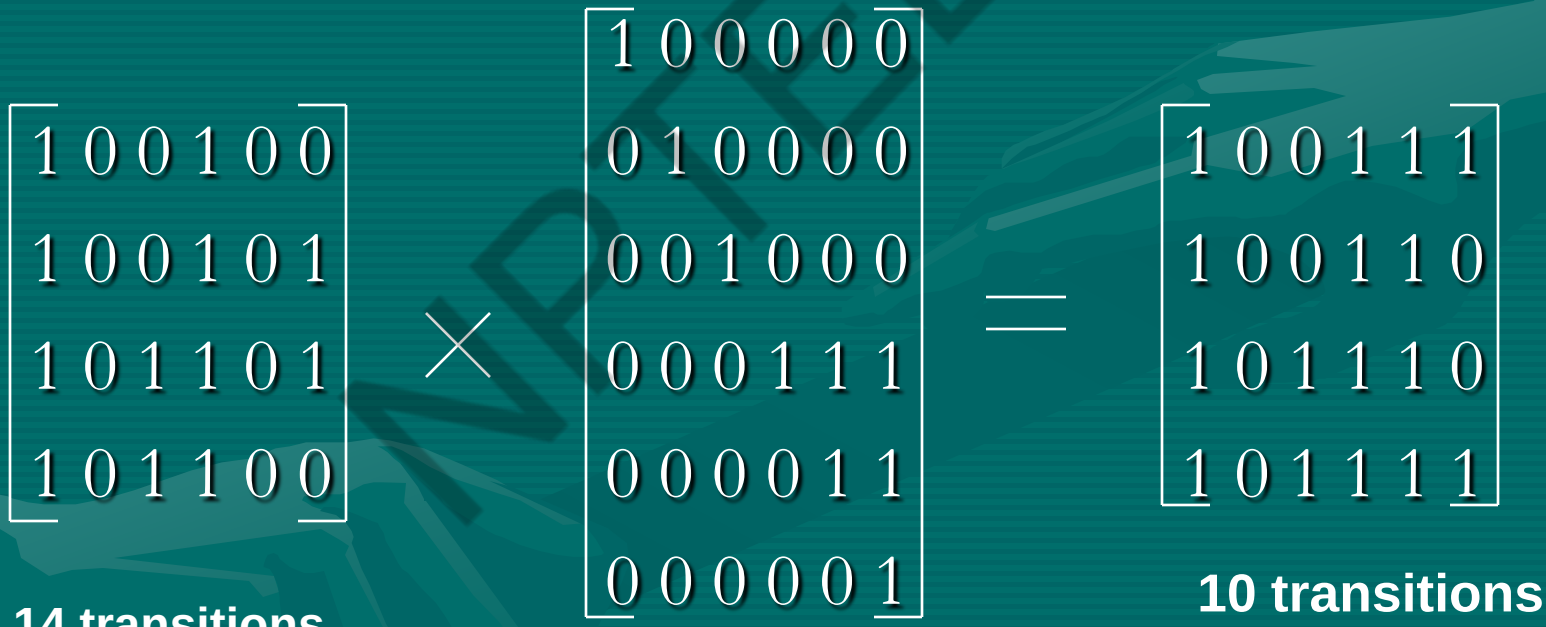


Test stimulus transformation matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiple XORs (contd.)

- Reduces number of transitions in the overall test set.
- For example,



14 transitions

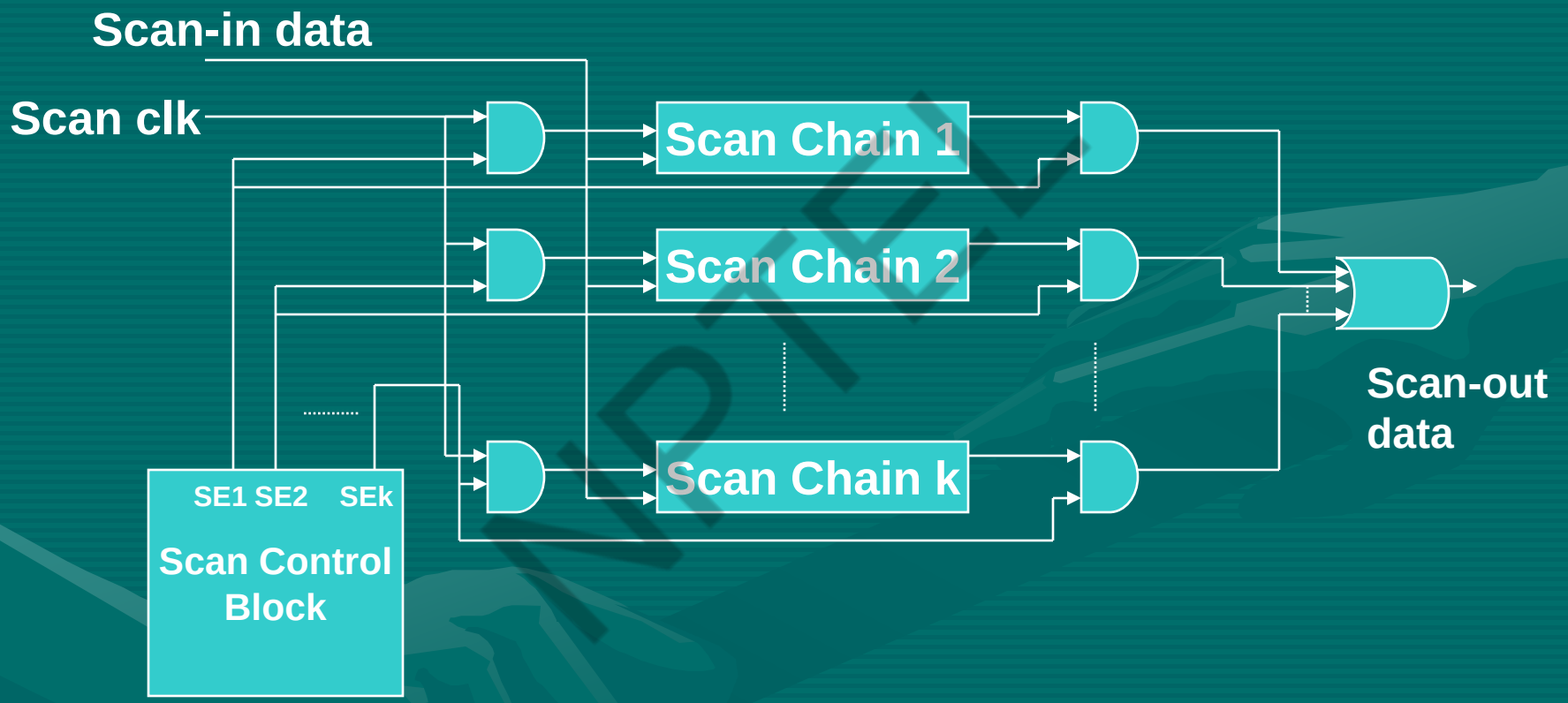
10 transitions

Caution: These are not weighted transitions

Use of Multiple Scan Chains

- Low test area and test data volume overhead
- No penalty in test application time, test efficiency, or performance
- Extra hardware required can be specified at logic level and synthesized with the rest of the circuit
- Multi-scan chain design easily embeddable in the existing VLSI design flow

Multi Scan Architecture

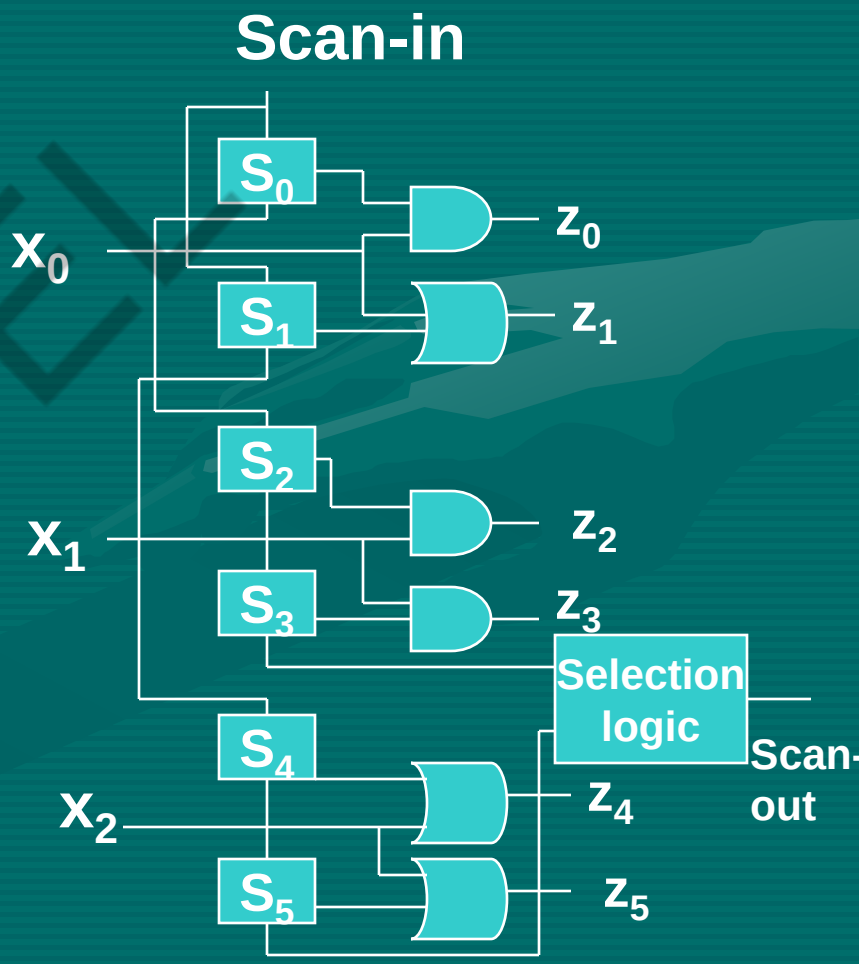


Compatible, Incompatible, Independent scan cells

- To partition the scan cells into multiple ones, they are classified as:
  - Compatible
  - Incompatible
  - Self-incompatible
  - Independent

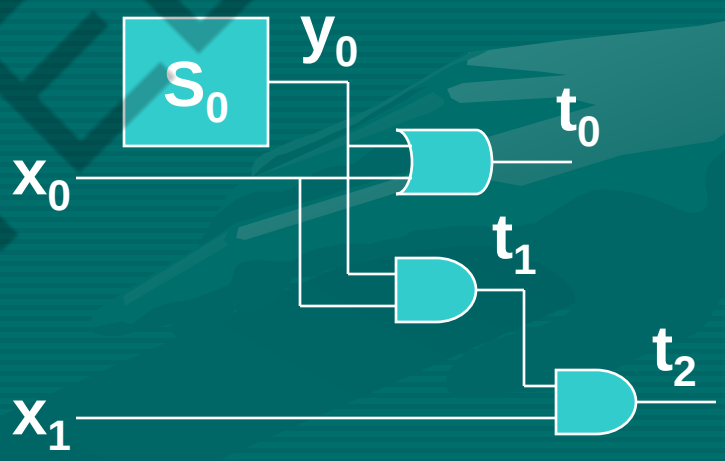
Example

- S_0 and S_1 are *incompatible* as different values are needed at x_0 to suppress spurious transitions originating from them
- S_2 and S_3 are *compatible* as x_1*=0* suppresses spurious transitions from both of them
- Similarly, S_4 and S_5 are *compatible*
- Scan chains: *{S_0, S_2, S_3}, {S_1, S_4, S_5}*



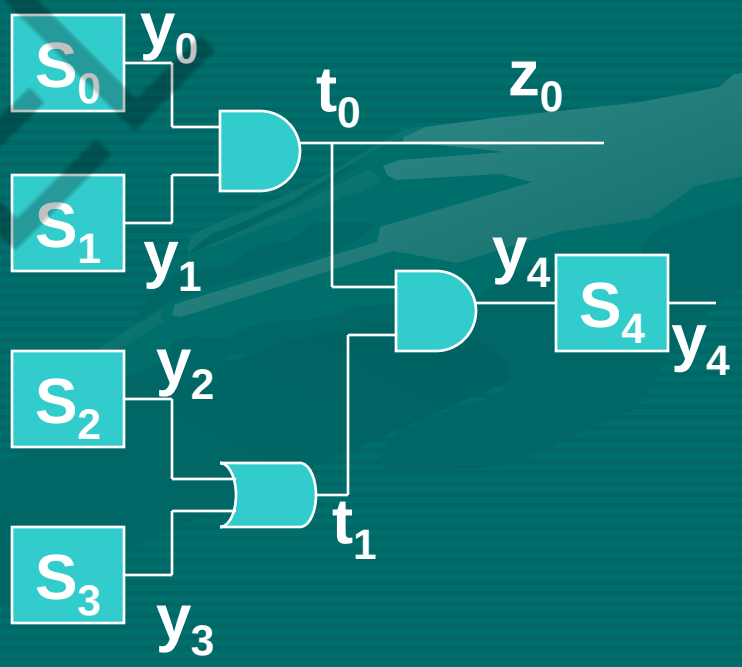
Example: Self-incompatible

- S_0 is self-incompatible
- For any setting of x_0, the transition proceeds via either t_0 or t_1



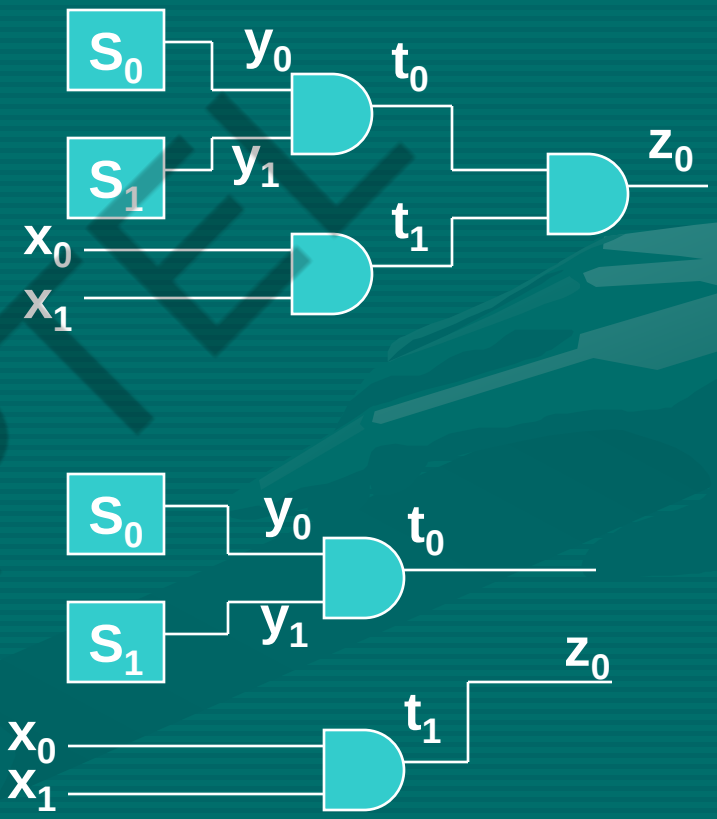
Example: Independent

- S_0, S_1, S_2, S_3 do not have side inputs for gates t_0 and t_1. Thus, these are independent



Computing Primary Inputs to Eliminate Spurious Transitions

- t_1 removes spurious transitions originating at S_0 and S_1
- S_0, S_1, and t_0 are excluded from the reduced circuit
- z_0 is modified to a buffer
- Targeted fault in the reduced circuit is t_1 *s-a-1*
- Extra test vectors *{0X, X0}*



Clock scheme modification

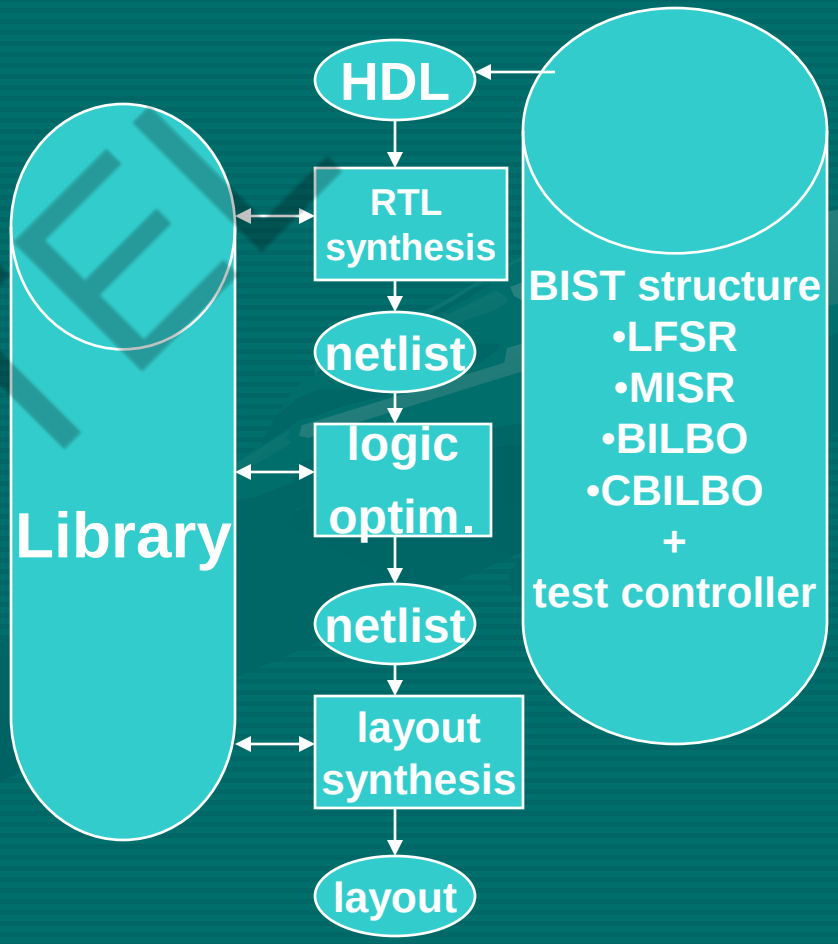
- Test power's major contributor is the clock tree.
- Generate and order test sets so that some scan chains can have their clocks disabled for portions of the test set. This prevents flip-flops from transitioning and the reduces test power.
- Gated clock for scan path and the clock tree feeding the scan path. Lowers transitions and thus minimizes average and peak power, and energy consumption.

Test scheduling algorithms

- A distributed BIST control scheme can be used that can schedule the execution of each BIST element to keep power dissipation under specified limit.
- Reduces average power, however, increases test time.
- Several SoC test scheduling algorithms have been proposed based on test bus partitioning, rectangle packing, simulated annealing, genetic algorithm etc.
- The schemes take care of power and precedence as constraints.

Power Conscious Test Synthesis

- Address testability at higher levels of abstraction during early stages of VLSI design
- BIST hardware is inserted at RTL
- Power dissipation during test application in BIST data paths need to be accounted for

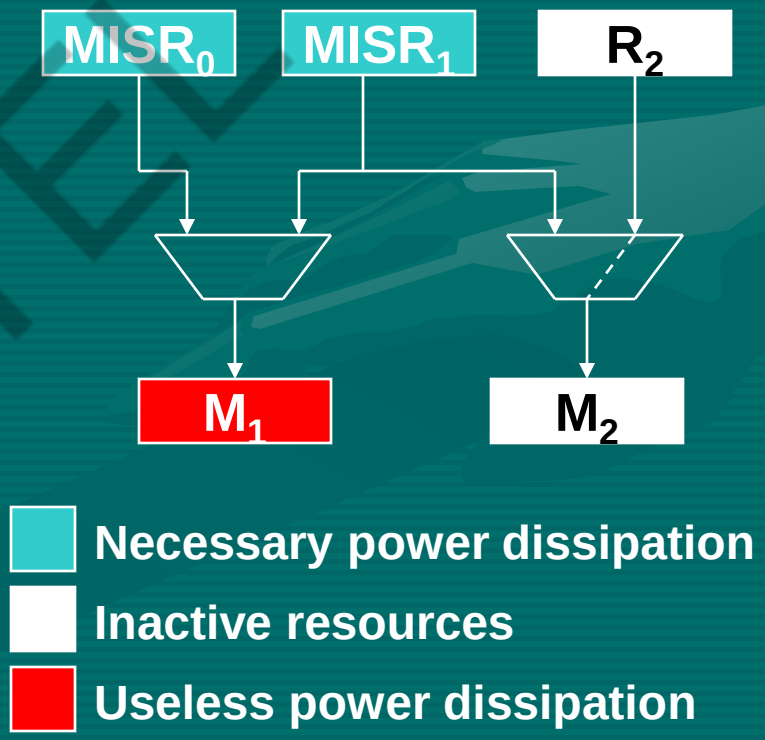


Power Dissipation in BIST Data Paths

- **Existing power constrained test scheduling approaches are optimistic for BIST data paths, since,**
  - **Test scheduling assumes fixed amount of power dissipation associated with each test which is not the case for BIST data paths – useless power dissipation in untested modules**
  - **Test scheduling is performed on a fixed test resource allocation without considering the strong interrelation between test synthesis and test scheduling**

Example

- $MISR_0$ and $MISR_1$ are active simultaneously analyzing outputs of some circuit components
- Transitions in $MISR_1$ is not affecting M_2 as inactive register R_2 can be selected as it's input
- M_1 consumes useless power



Selecting effective low power testing strategy

- Implementation context: Whether the technique is for external testing, scan, scan BIST etc.
- Way to address test power minimization: Act on test sequence or on the test architecture.
- Relax classical test constraints:
  - Fault coverage, test time must remain unaltered.
  - Area overhead from hardware modification must be acceptably low.
  - Must maintain circuit performance.
  - Effect on the design flow, and hence on the design time must be small enough for the solution to be acceptable.
  - Designers must make the clock tree as small as possible or disable clock signals as often as possible.