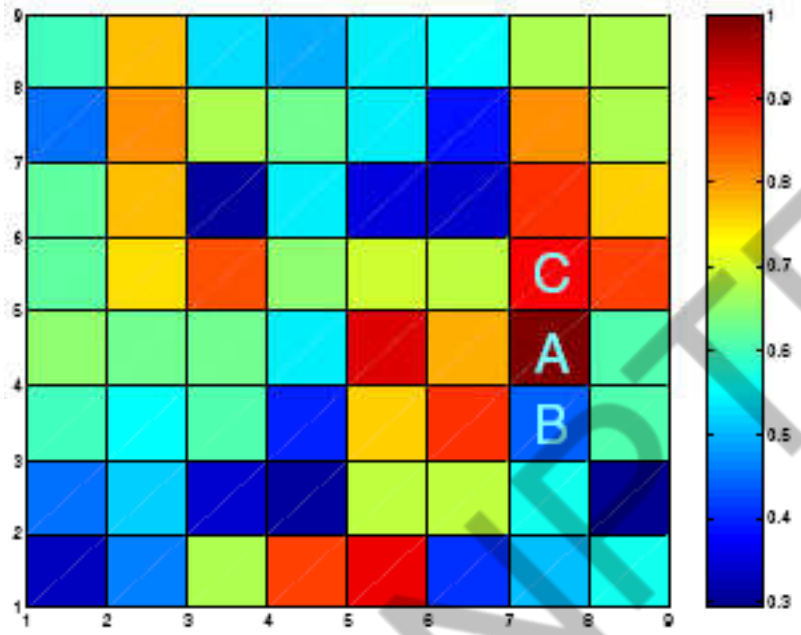
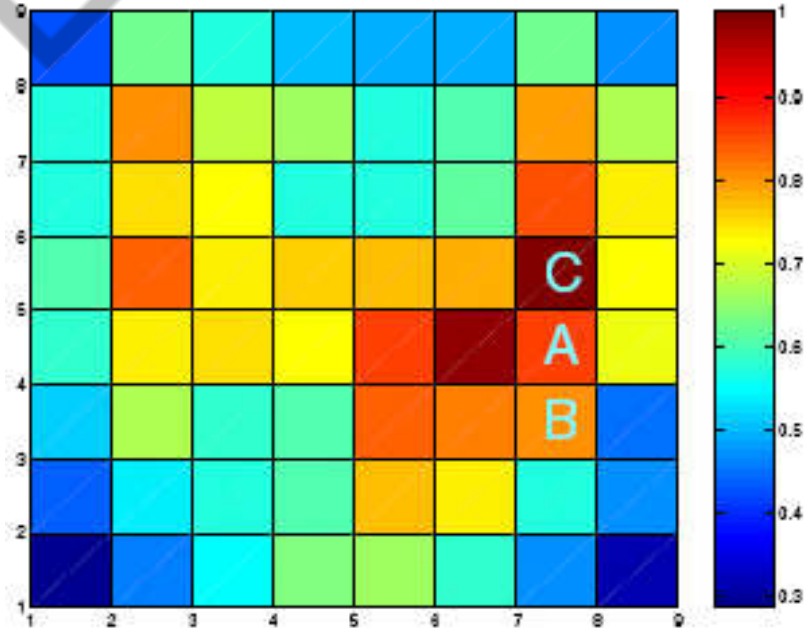
Lecture 37

Thermal Aware Testing

Why Thermal Aware Testing

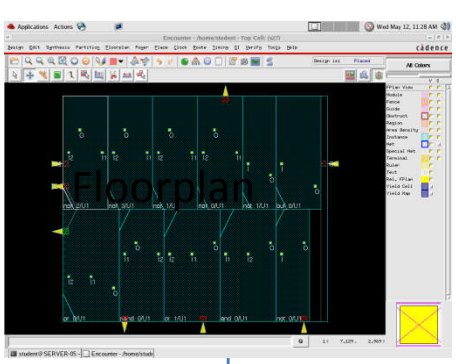


Power Profile



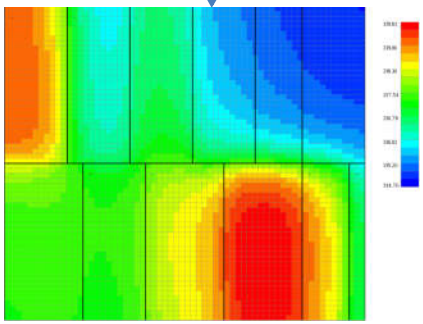
Temperature Profile

HotSpot - Thermal Simulator



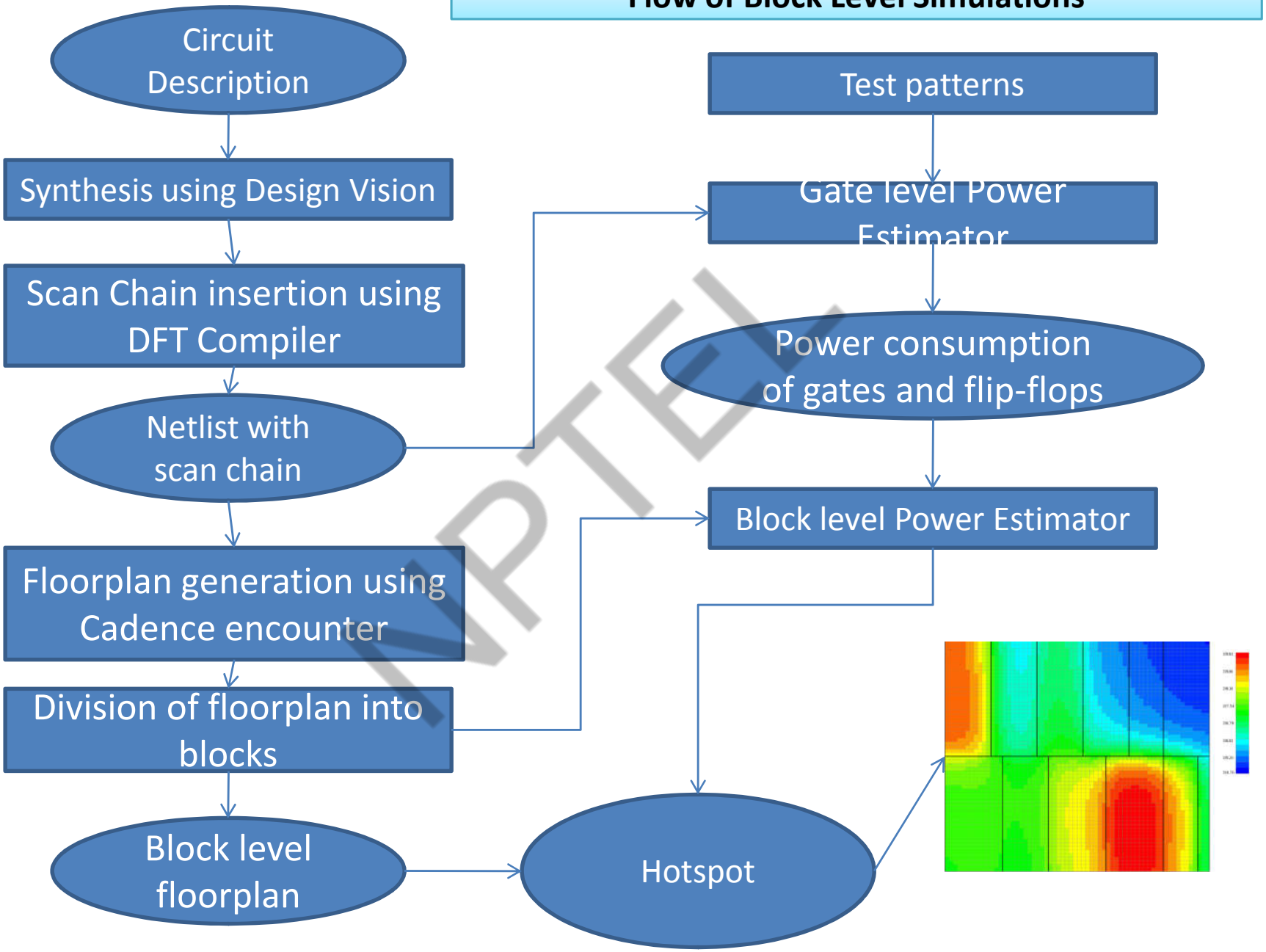
Power Trace

Thermal Estimator



HotSpot: lava.cs.virginia.edu/HotSpot/

Flow of Block Level Simulations



External Testing

THERMAL-AWARE TEST VECTOR REORDERING

Two test vector reordering techniques: one based on hamming distance, another using Particle Swarm Optimization

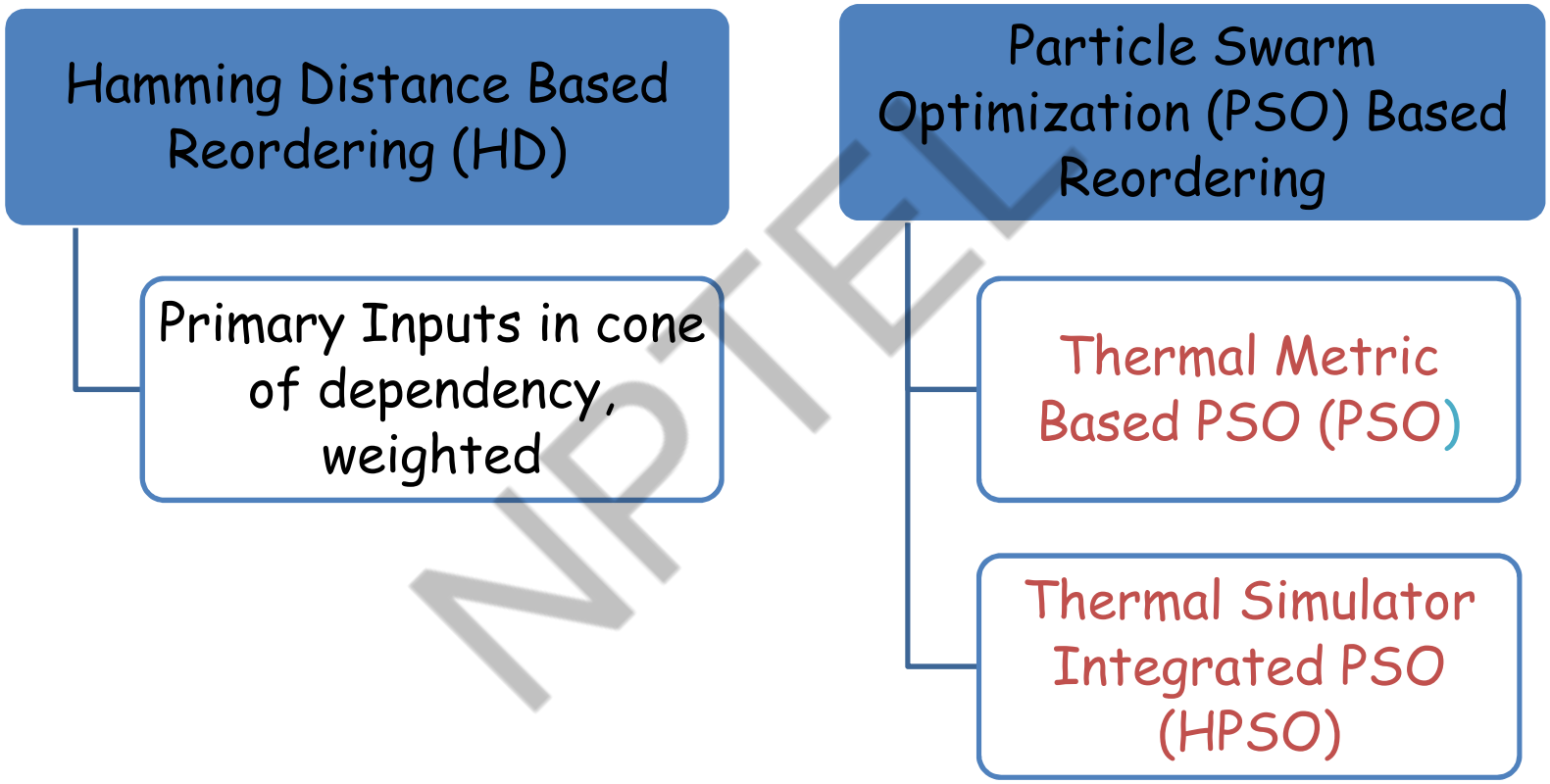
- PEAKASO algorithm* tries to minimize peak temperature to some extent by overheat pre-compensation.

Drawbacks:

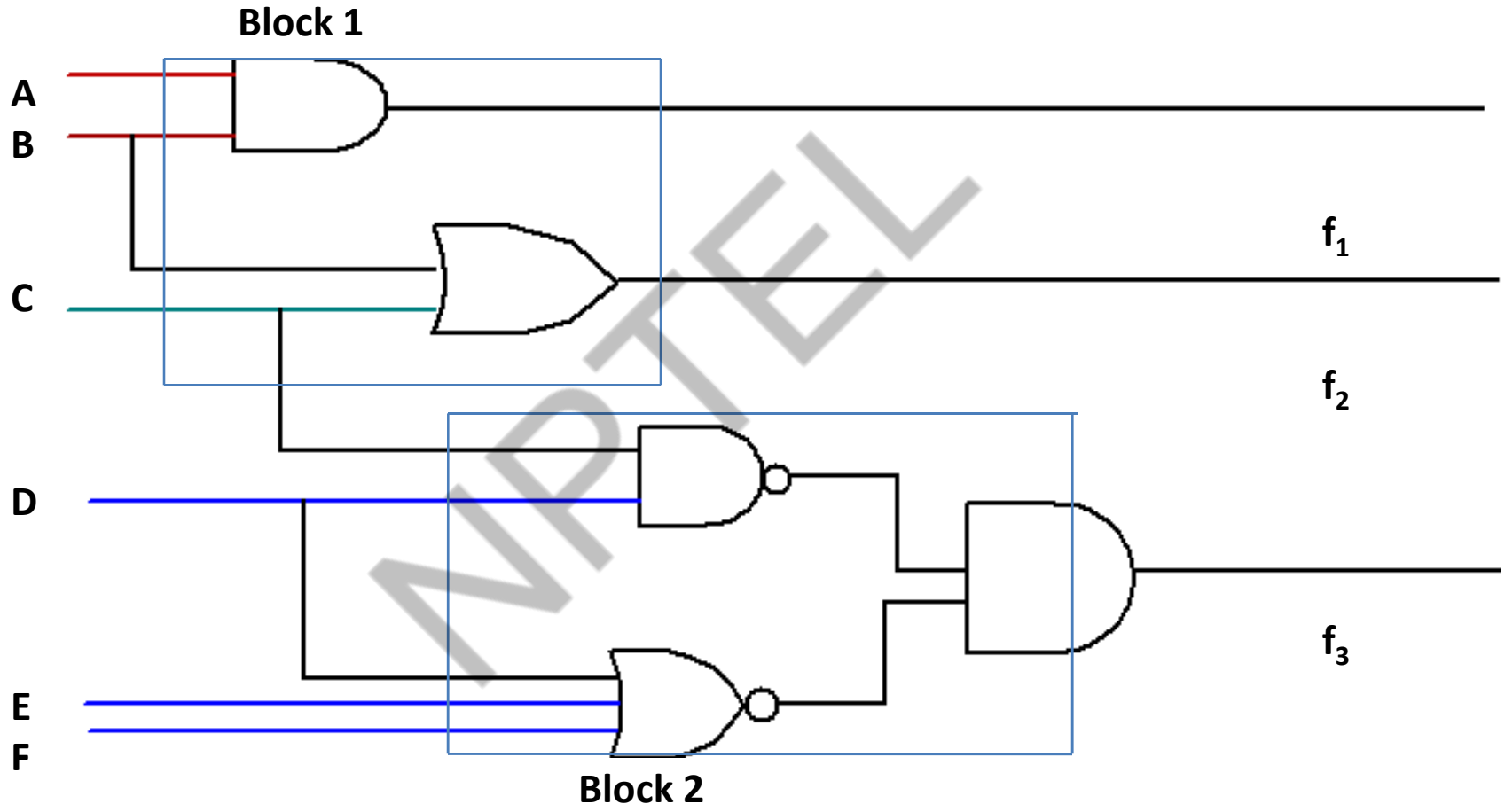
- concentrate on minimizing circuit transitions rather than peak temperature of CUT
- the average temperature of the CUT increase.

* Minsik Cho and David Z. Pan. ,"PEAKASO: Peak-temperature aware scan-vector optimization," Proceedings 24th IEEE VLSI Test Symp., pp. 52-57, 2006.

Vector Reordering Approaches



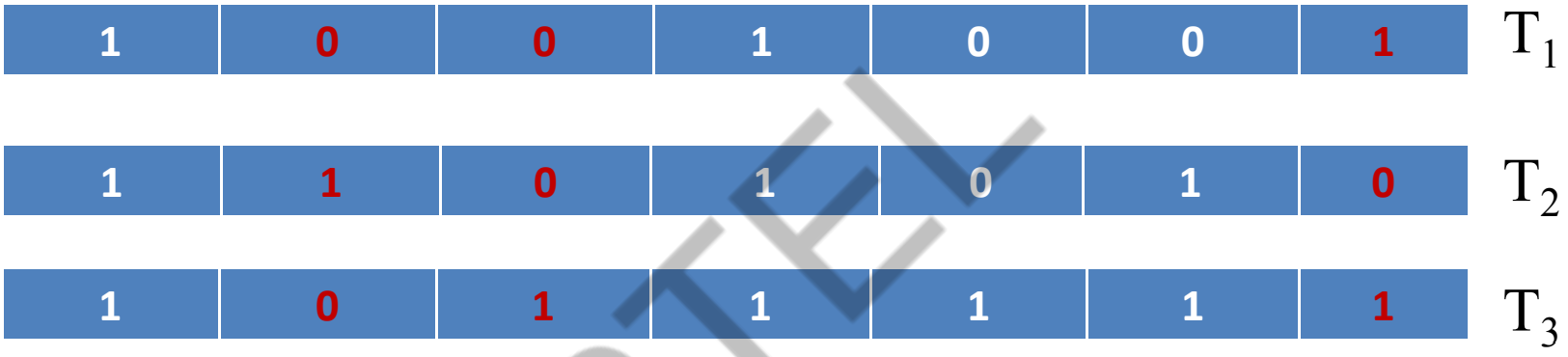
Input Cone of Dependency of Blocks



Block 1: A,B,C

Block 2: C,D,E,F

Algorithm 1 – Hamming Distance Based Vector Reordering(HD)



Weight of a Block W_{bi} = Temperature of the Block B_i (T_{bi})/Maximum Temperature at any Block (T_{max})

Cost Function:

Σ W_{bi}* Hamming Distance at the Primary Inputs in the cone of dependency of Block B_i

Drawbacks of Hamming Distance Based Approach

- May lead to an increase in the transitions at primary inputs other than in the cone of hottest blocks
- Average temperature of the chip may increase

Particle Swarm Optimization

- *Particle Swarm Optimization (PSO)* is a population based stochastic technique developed by Eberhart and Kennedy in 1995.
- Inspired by social behaviour of bird flocking or fish schooling.
- Potential solutions, called *particles, fly through* the problem space by following the current optimum particles.
- PSO has been successfully applied in many areas and has been found to outperform GA in cost function and execution time.

PSO

- Consider the following scenario: a group of birds are randomly searching for food in an area.
- There is only one piece of food in the area being searched. None of the birds knows exactly where the food is.
- During each iteration, they learn via their inter-communications, how far the food is.
- So the best strategy to find the food is to follow the bird which is nearest to the food.
- PSO is learnt from this bird-flocking scenario, and used to solve the optimization problems.

PSO

- Each single solution (particle) is perceived as a bird in the search space.
- Each particle has
  - a fitness value which is evaluated by the fitness function (the cost function to be optimized), and
  - a velocity which directs its flight.
- The particles fly through the problem space by following the current optimum particles.
- Initialized with a group of random particles (solutions) and then searches for optima by updating through generations.

PSO

- During every generation (iteration), each particle is updated by following two *best values.*
  - *pbest*, the position vector of the best solution (fitness) this particle has achieved so far. This fitness value is also stored along with the particle. This
  - *gbest*, the global best position, obtained so far, by any particle in the population.

Lecture 38

Particle updation

$$v_{k+1}^i = wv_k^i + c_1 r_1 (pbest^i - x_k^i) + c_2 r_2 (gbest_k - x_k^i)$$

$$x_{k+1}^i = x_k^i + v_{k+1}^i$$

where, v_{k+1}^i is the velocity of particle i at $(k+1)^{th}$ iteration, x_k^i is the current particle solution (position), r_1 and r_2 are random numbers between 0 and 1, c_1 is the self-confidence (cognitive) factor, c_2 is the swarm confidence (social) factor, w is the inertia factor.

- First term in the first equation represents the effect of inertia of the particle
- Second term represents the particle memory influence
- Third term represents the swarm influence.
- The velocities of the particles on each dimension may be clamped to a maximum velocity Vmax.
- A comparison with GA shows that PSO is better than GA as it has lesser number of tuning parameters, and is faster than GA due to the linear complexity of its main loop.

PSO Formulation for Vector Reordering

- Number of test vectors= *n*

- Particle
  - ❖Ordered test vector set
  - ❖Permutation of numbers from 0 to n-1

Swap Operator & Swap Sequence

Swap Operator (SO)

Particle P = { *t1, t3, t5, t7, t4, t2, t6* }

Swap Operator SO(2, 3)

P_{new} = { *t1, t3, t7, t5, t4, t2, t6* }

Swap Sequence (SS)

Particle P = { *t1, t3, t5, t7, t4, t2, t6* }

Swap Sequence = { SO(2,3), SO(0,4) }

P_{new} = { *t4, t3, t7, t5, t1, t2, t6* }

Algorithm 2 – Peak Power Minimization (PPM)

- To check the benefit of the thermal-aware optimization over a simple power-aware approach, a power-aware PSO has been constructed

Cost Function:

Fitness = Minimize the Peak block power

Algorithm 2 – Thermal Metric Based PSO(PSO)

Neighboring gates of Block B_i = $ne(B_i)$

Average Weight W_{avegi} of a Block W_{bi} = (1/N) Σ W_b
where, $b \varepsilon ne(B_i)$

Criticality $C_i = (1 + (W_b - W_{avegi}))^{-1}$

Cost Function:

Fitness = Σ $W_{bi} * C_i * (T_i - T_{bi\_initial})$

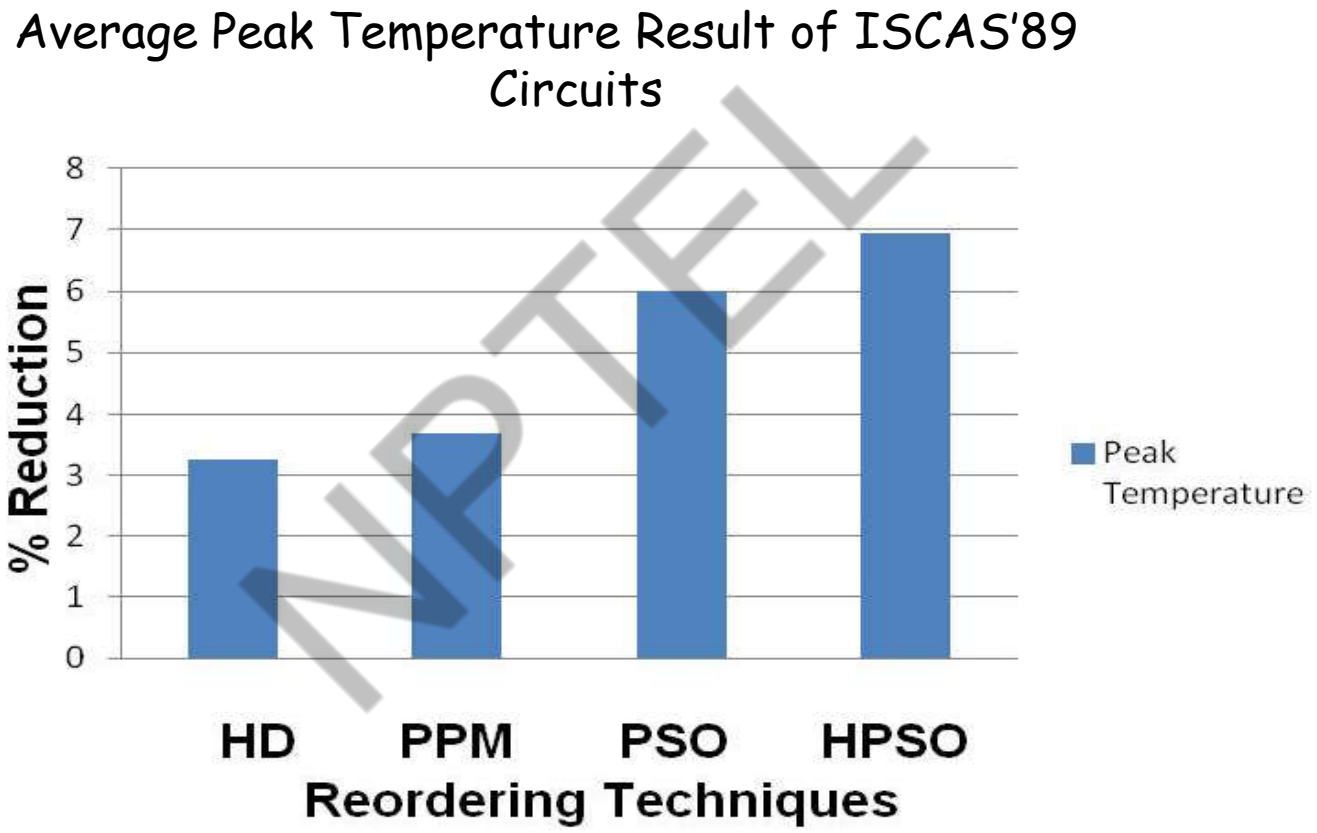
Algorithm 3 – Thermal Simulator Integrated PSO (HPSO)

- Hotspot is called within PSO
- For a particular ordering, block level power trace is generated and fed to Hotspot to get the actual temperature trace
- Peak block temperature found out

Cost Function:

Fitness = Minimize the Peak block temperature

% Reduction in Peak Temperature

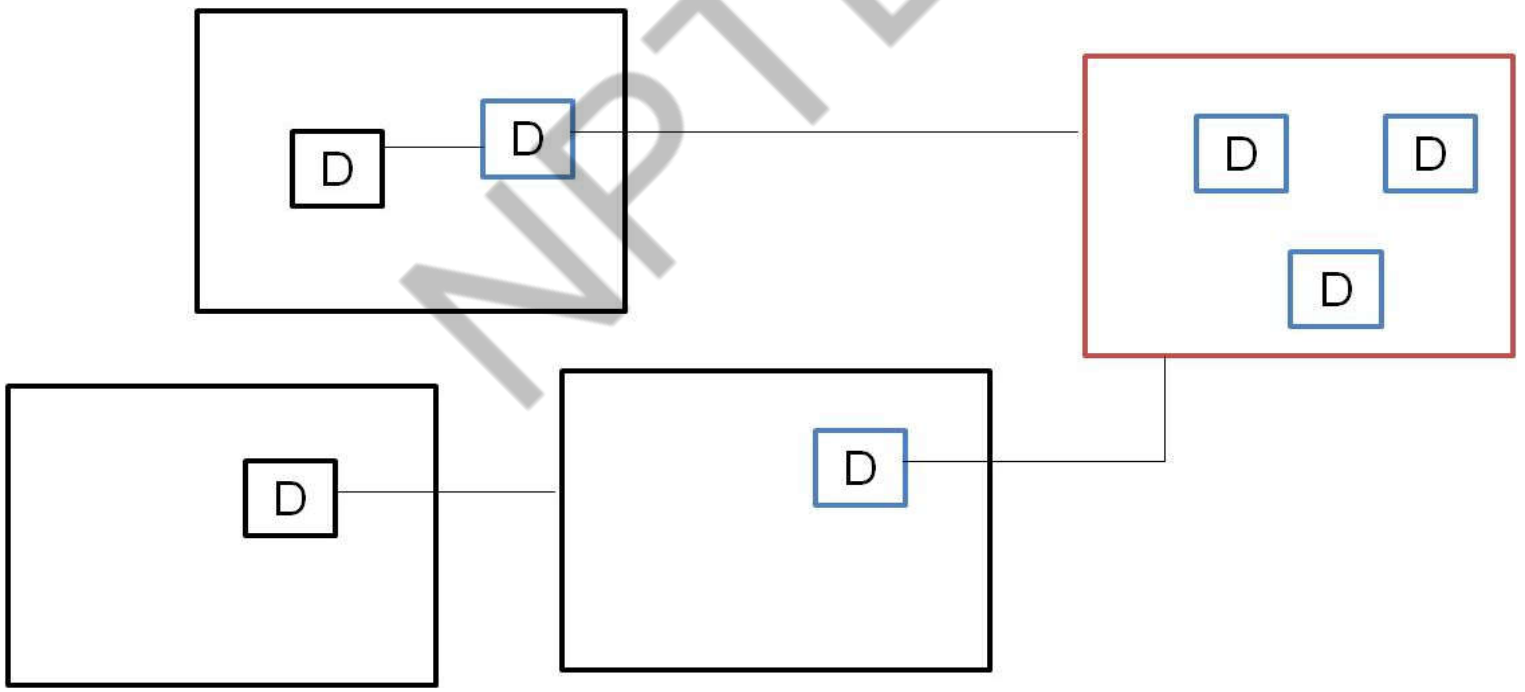


Though HPSO puts more emphasis on reduction temperature than PSO, the CPU time needed for HPSO is almost 4 times greater than PSO. So, among the proposed approaches , PSO is the feasible solution.

Thermal Aware Don’t Care Filling

Critical Flip-flops

- The FFs which are either within the block or in the near fan-in cone of a gate within the block



Weight of a Block

- 10,000 random patterns are applied

Weight of a Block (W_{bi}) = Total power seen at the Block B_i (P_{bi_random}) / Power at the Critical FFs of Block B_i(C_{bi})

- A higher block weight represents that the block will consume more power when transitions occur at the flip-flops

The Power Estimator Metric

- Used for accurately measure of the power behavior of the CUT
- Defined with respect to a particular set of fully specified test vectors

Metric for estimating power (P) = Σ Weight of Block B_i(W_{bi}) *Power of critical flip-flops for Block B_i after application of when a fully specified test set to the CUT (T_{bi})

Estimation of Thermal Behavior

1. A new term, *Criticality* is introduced
2. It is an index of the thermal gradient between the block and its neighboring ones
3. Blocks with higher CR_i will get better attention compared to the ones with lower values of CR_i

"Trust Thy Neighbors"

- *Criticality* in both 2D and 3D, depends upon the power dissipation of the block and its neighbors
- Except in 3D, along with the neighbors, it depends upon the layer number residing the block

Criticality of a Block B_i (CR_{bi}) *in 2D* = (P_{bi} + $P_{ne(bi)}$) /(N+1)

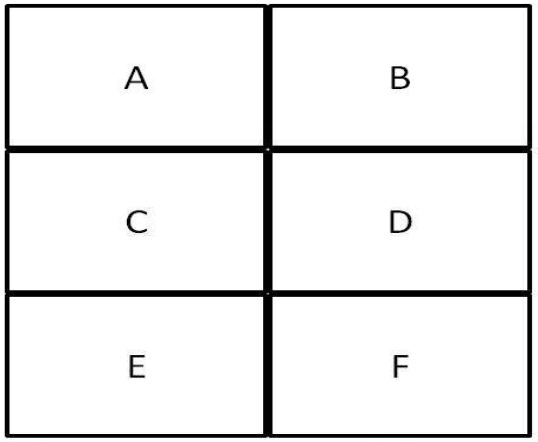
Criticality of a Block B_i (CR_{bi}) *in 3D* = (P_{bi} + $P_{ne(bi)}$) /((N+1) *(l_{bi} +1))

Where

$P_{ne(bi)}$ *is* total Power of the neighboring blocks of B_i

l_{bi} is the layer number of B_i

N is the number of neighboring blocks of B_i



Block	Neighbors
A	B,C
B	A,D
C	A,D,E
D	B,C,F
E	C,F
F	D,E

Fitness Function Used to Minimize Temperature and Variance

- Aim is to fill the don't care bits in test patterns in such a way that it will minimize fitness function

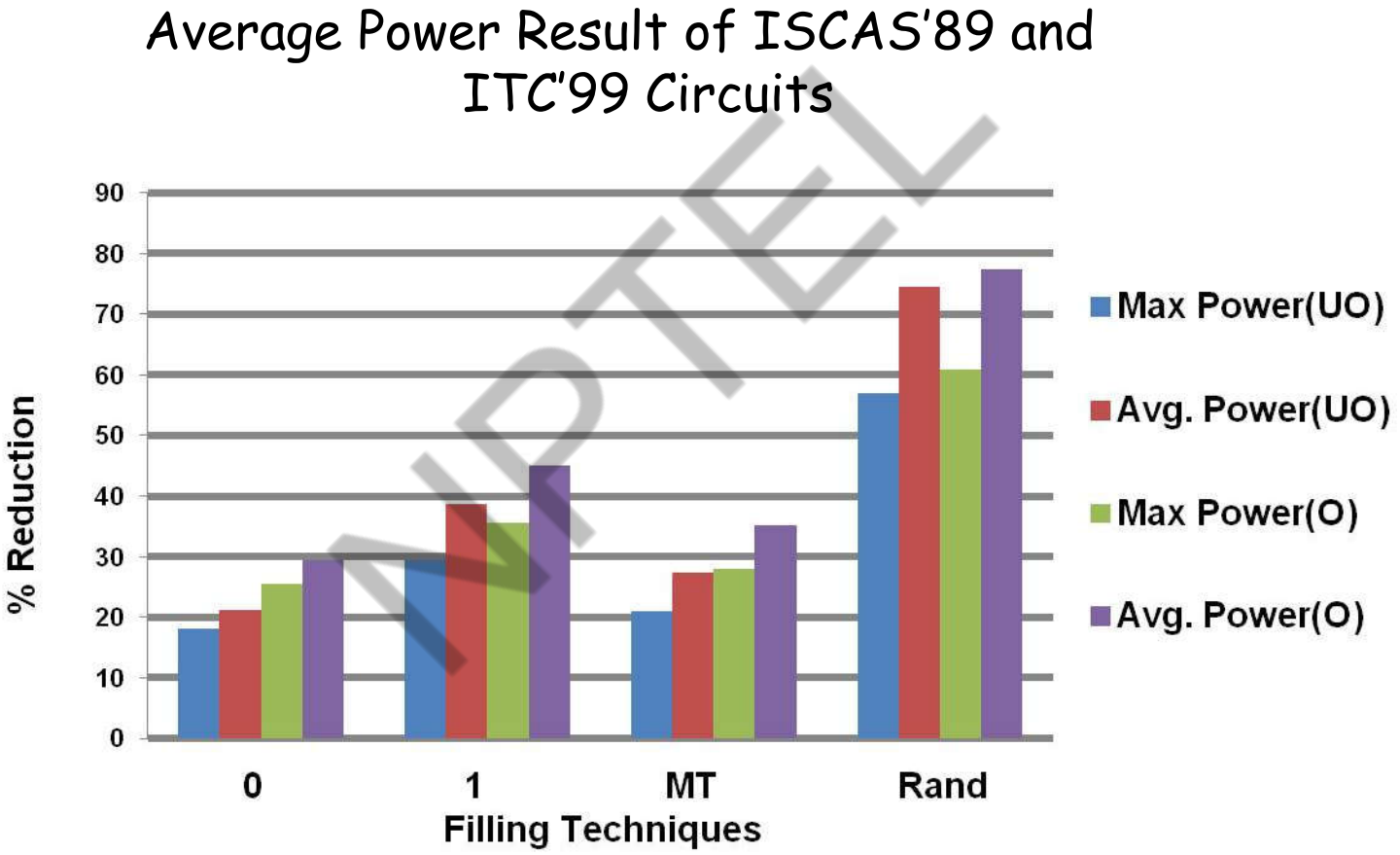
$$\text{Fitness } (C) = MAX(CR_{bi})$$

Proposed Algorithm

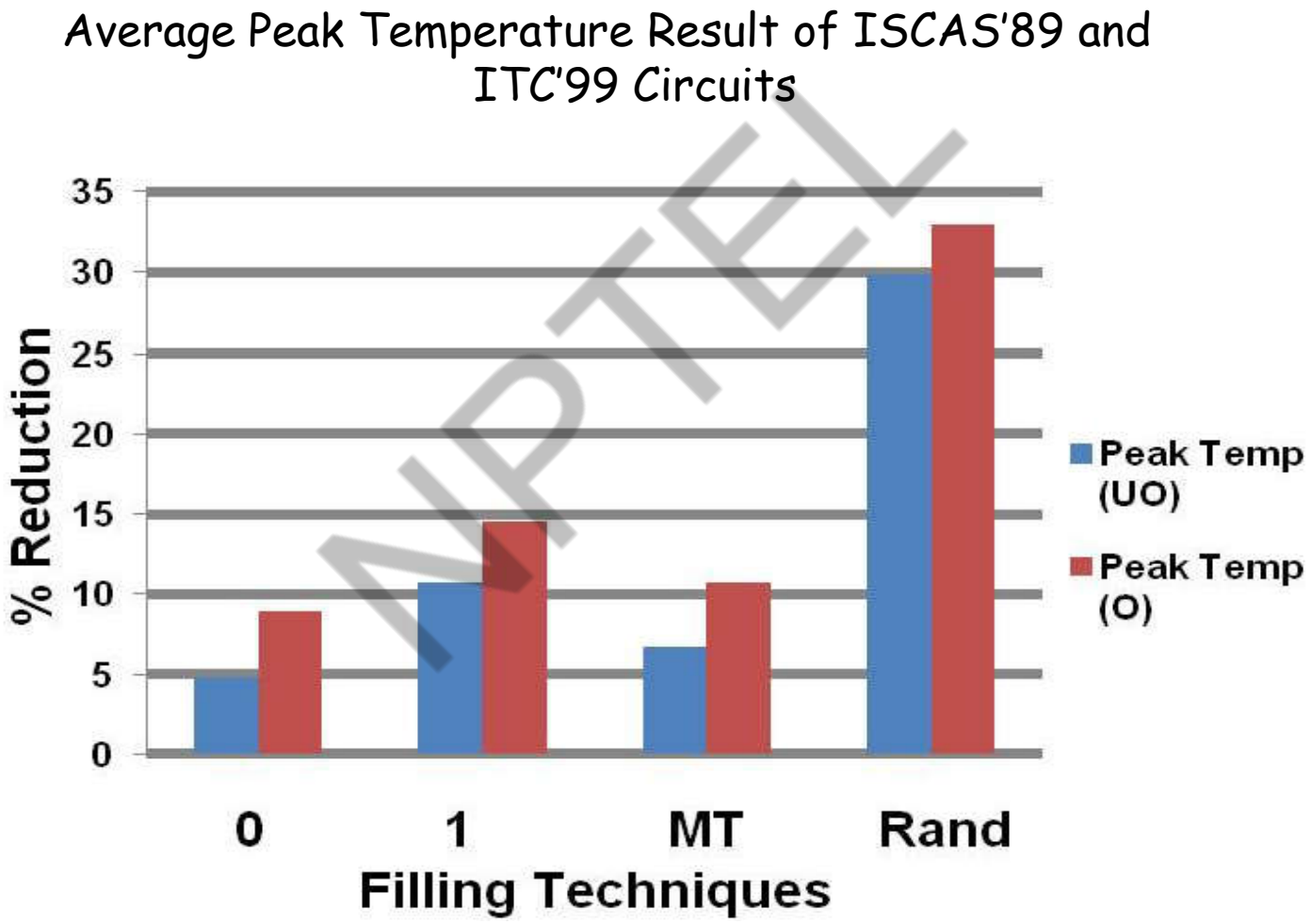
- Fill the test set using the different filling techniques(0-fill, 1-fill, rand-fill, MT-fill) and select the best test set to be used as initial test set and calculate *C*
- For all the patterns
  - For every 'x' bit initially present in the test set
    - Flip the bit from its originally filled value
    - Calculate *C* and store it in *temp_C*
    - If (*temp_C* < *C*),
      - retain the flip
    - Otherwise, change the bit to its initial filled value

Lecture 39

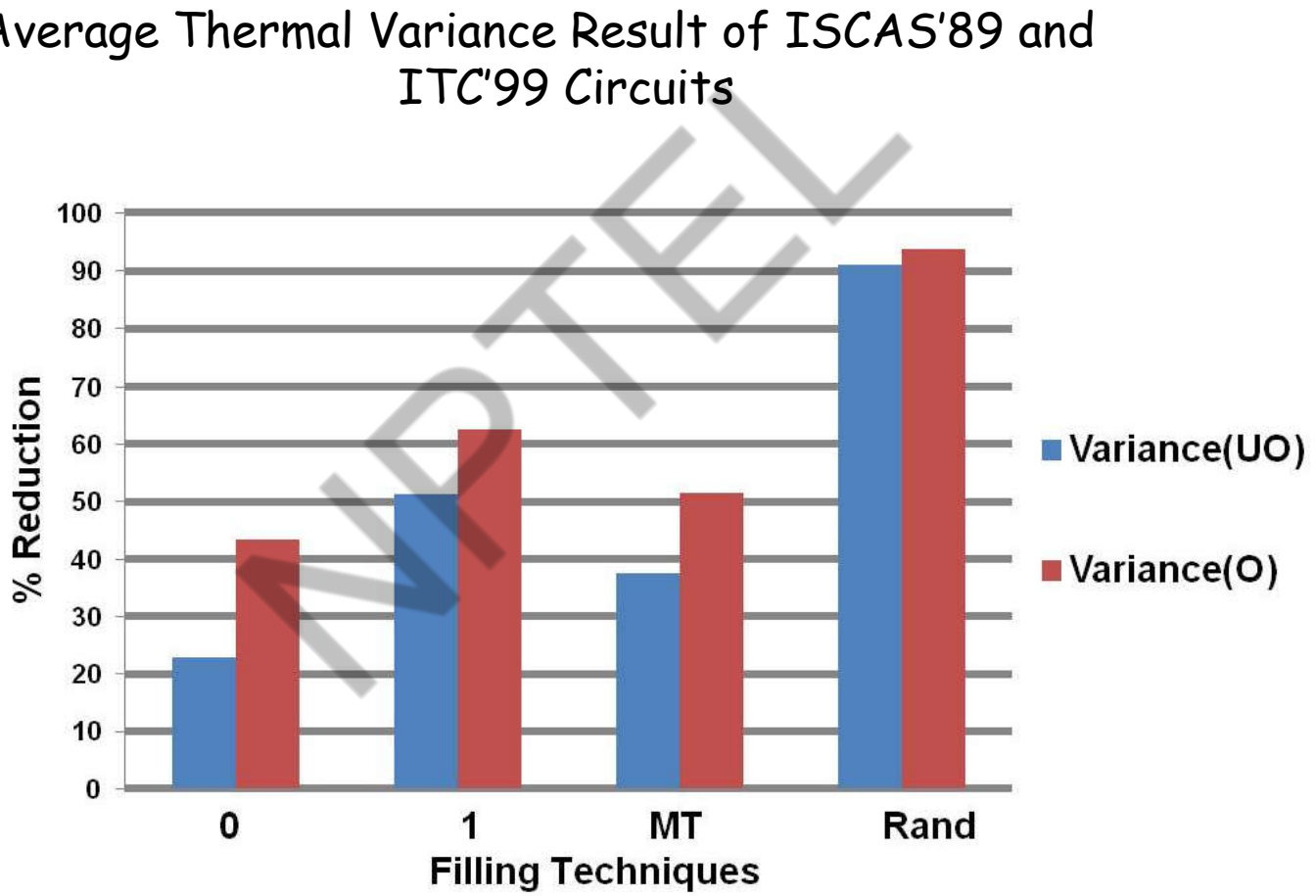
% Reduction in Power in 2D



% Reduction in Peak Temperature in 2D

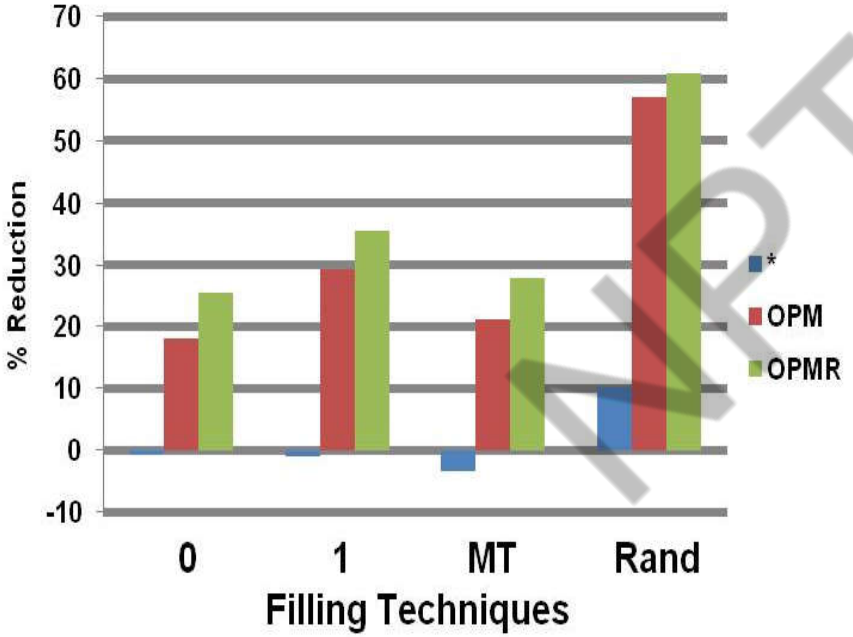


% Reduction in Thermal Variance in 2D

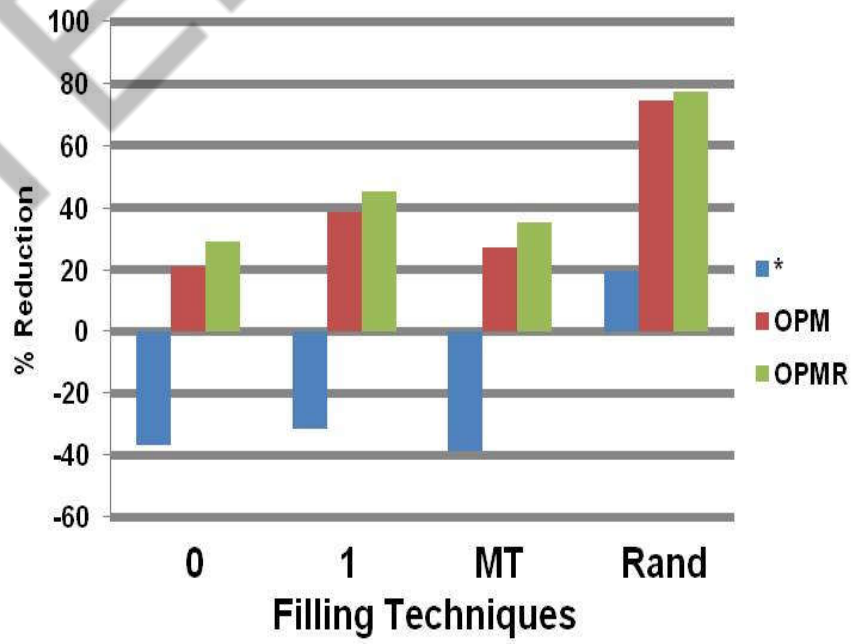


Comparison With Yoneda et al. 2011

% Reduction in Peak Power



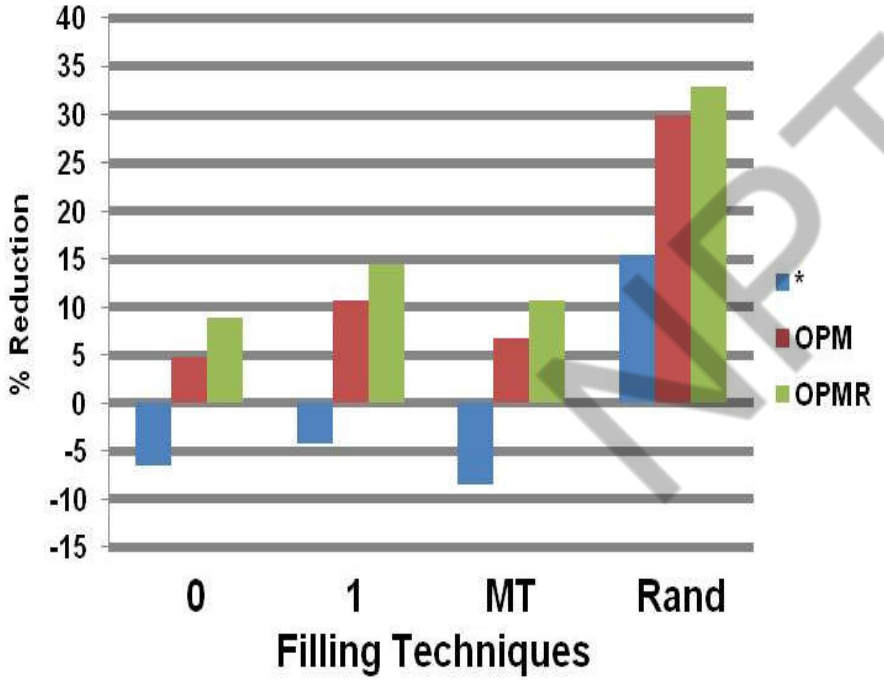
% Reduction in Avg. Power



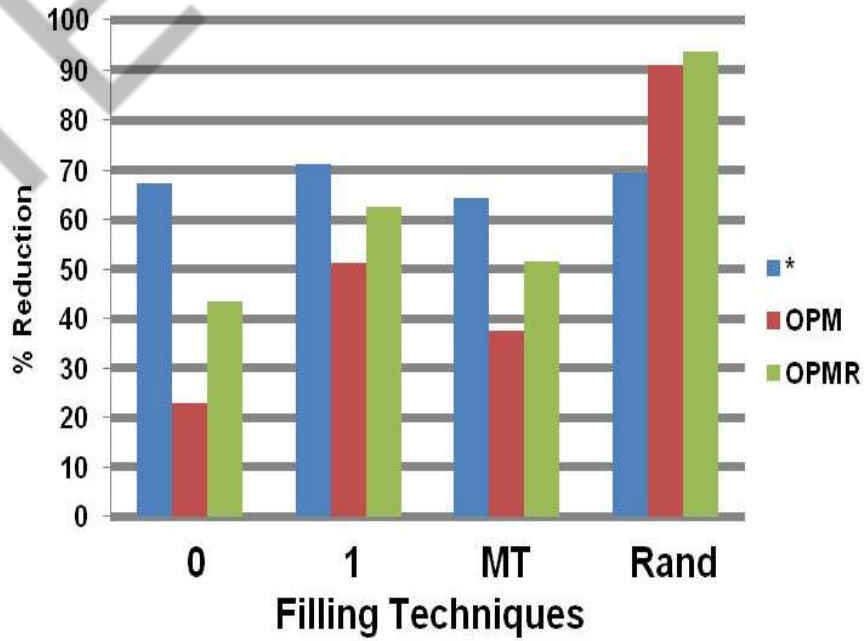
*Yoneda, T., Nakao, M., Inoue, I., Sato, Y. and Fujiwara, H., "Temperature-Variation-Aware Test Pattern Optimization" in Proc. European Test Symposium, pp.214, 2011.

Contd.

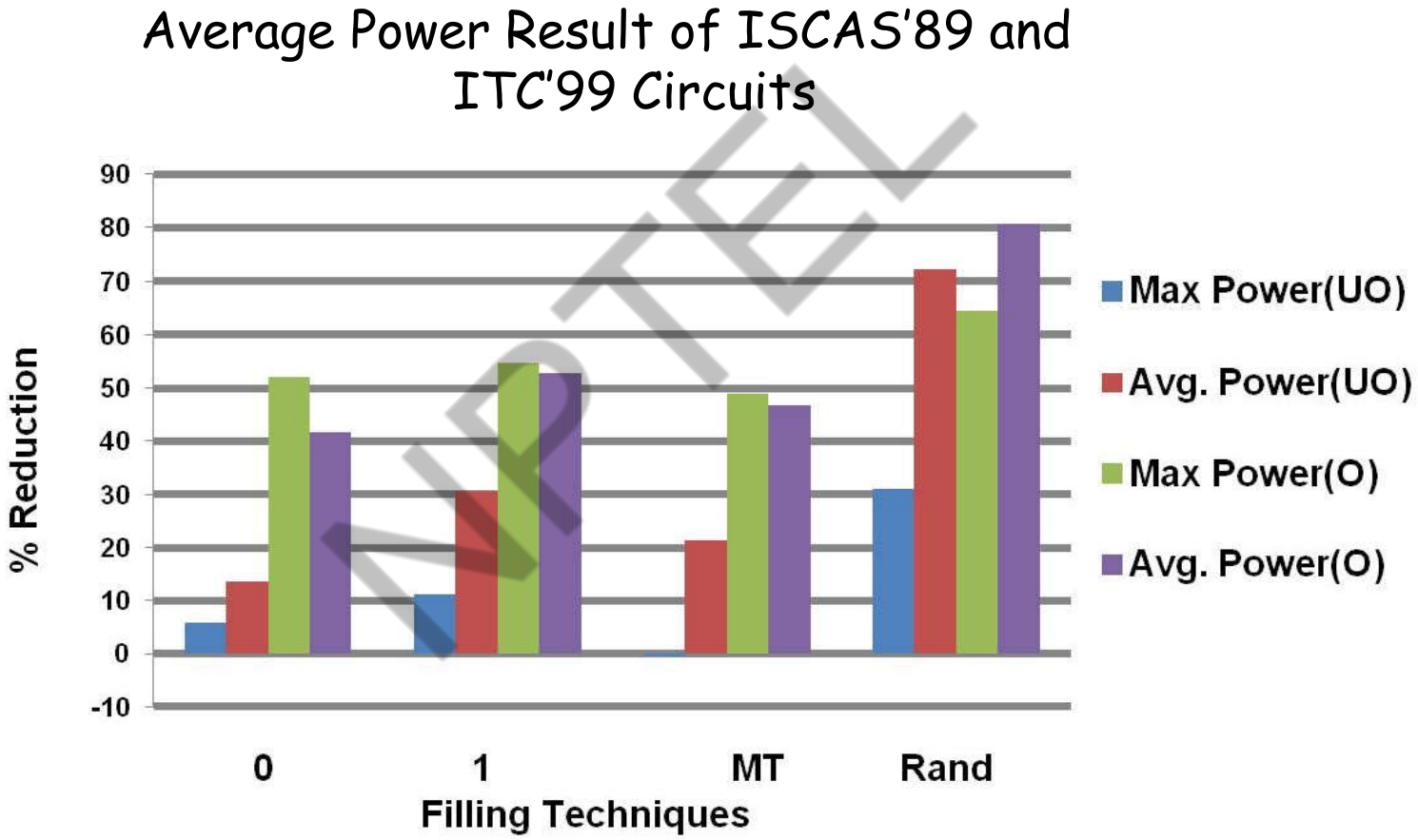
% Reduction in Peak Temperature



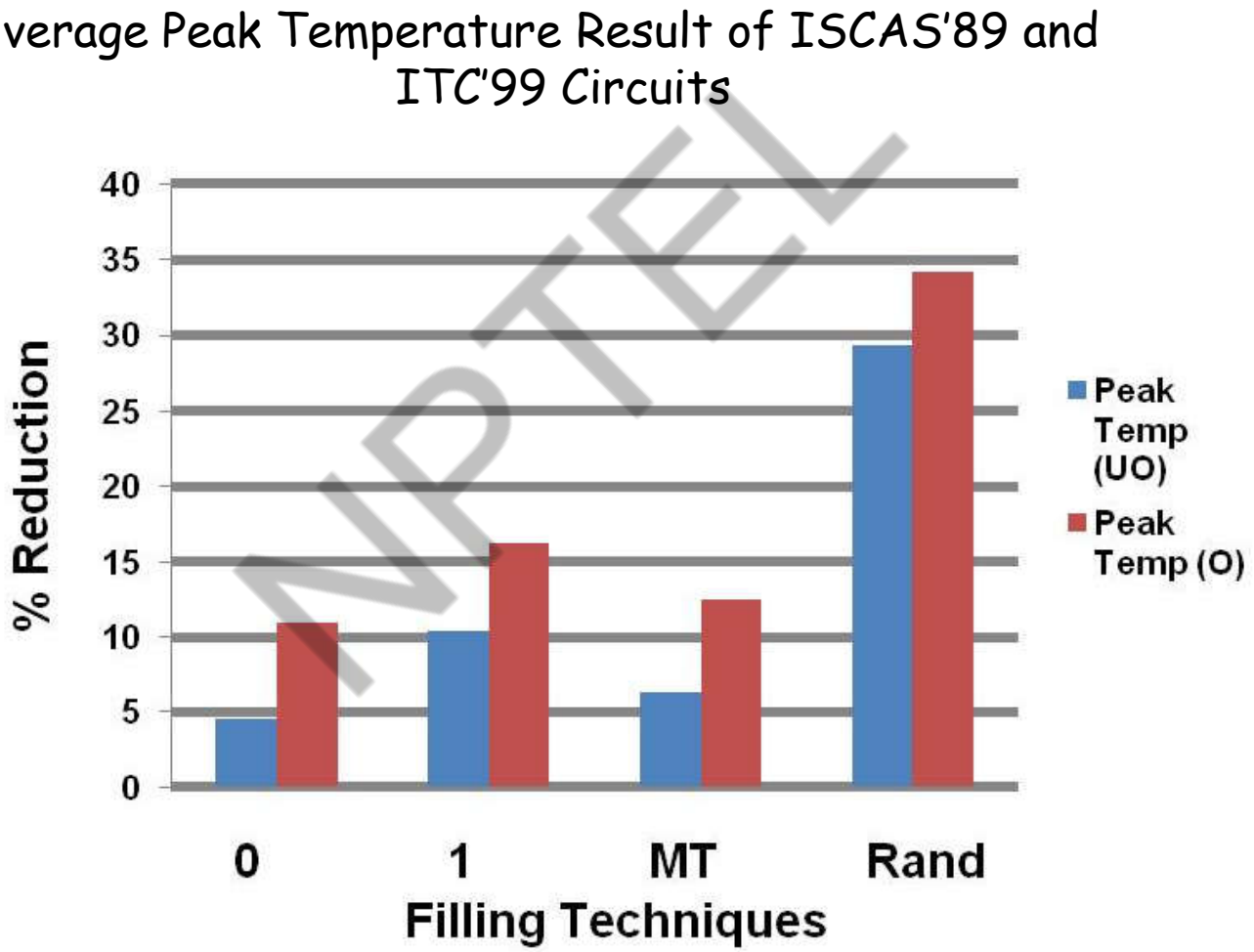
% Reduction in Thermal Variance



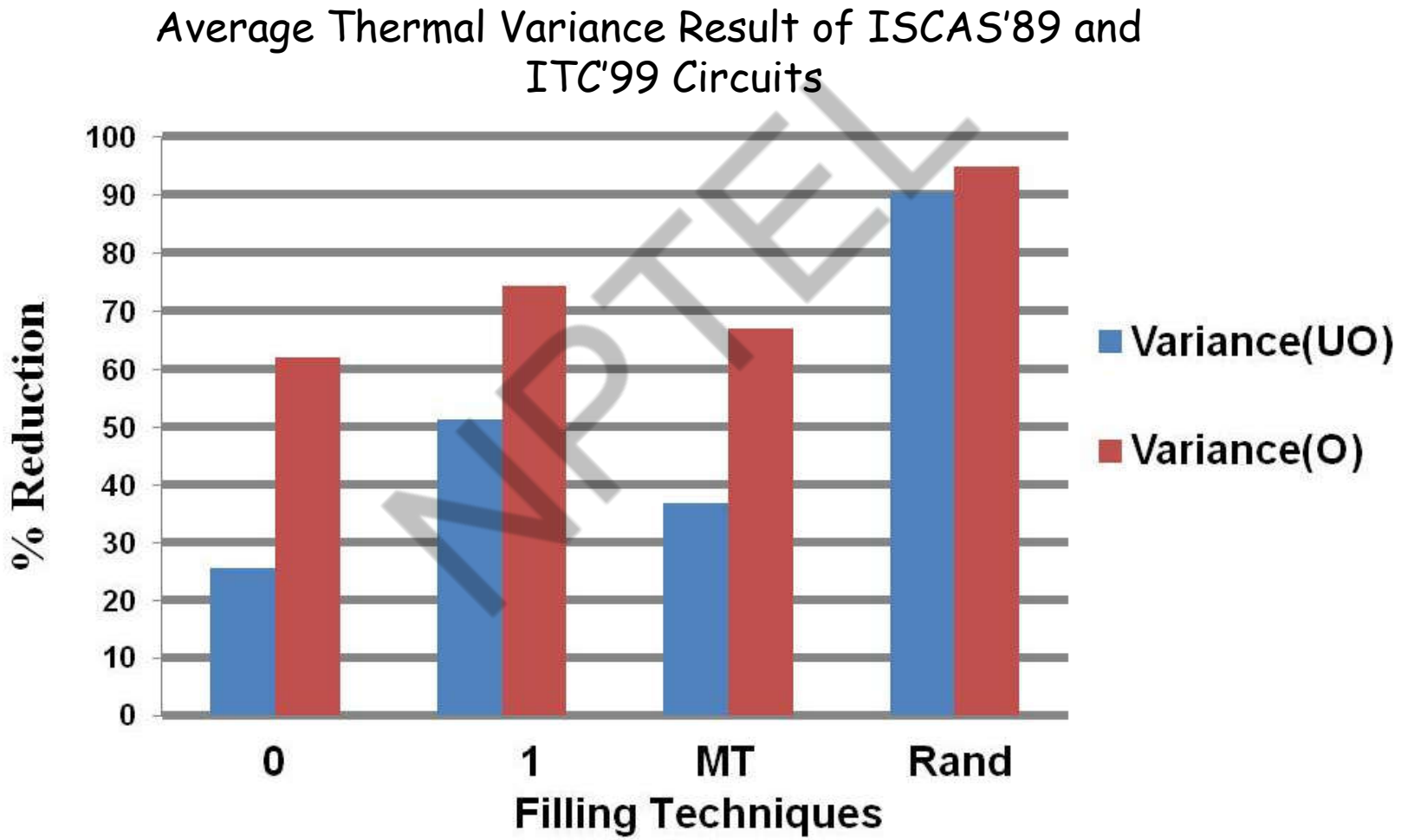
% Reduction in Power in 3D



% Reduction in Peak Temperature in 3D



% Reduction in Thermal Variance in 3D



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

Initial pattern

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

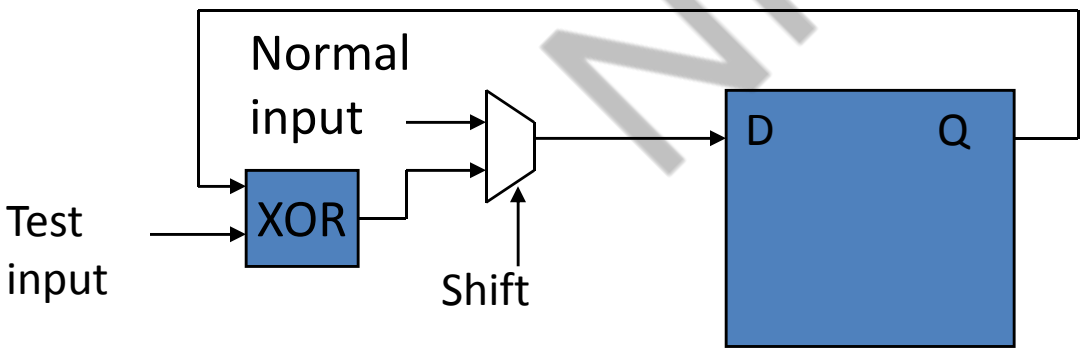
No. of transitions: 4

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

Scan Chain Modification Algorithm

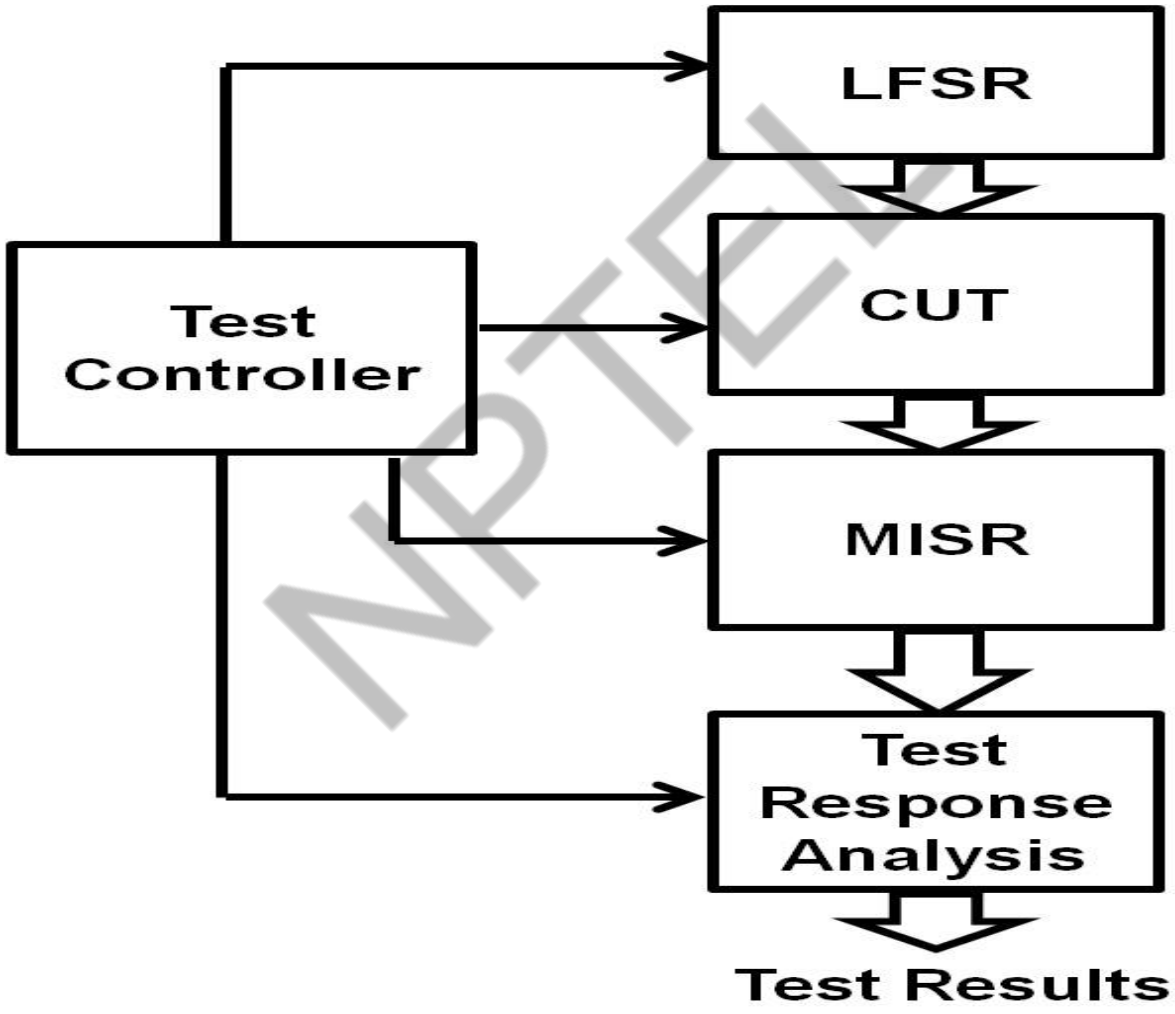
- Calculate C using the filled test set using the x filling algorithm proposed and considering all the flip-flops as D flip-flops.
- For all iterations
  - For every scan cell bit in the scan chain
    - Convert the cell from D flip flop to T flip flop and vice versa.
    - Modify the original test set and their responses by shifting in/out through the modified scan chain.
    - Calculate C and store it in *temp_C*
    - If (*temp_C* < C),
      - retain the configuration
    - Otherwise, change the cell to its initial configuration.
  - If C of two successive iterations match,
    - End the procedure.

% Reduction in Peak Temperature

Circuit	% Reduction in Peak Temperature w.r.t.							
	Unordered filled test set				Ordered filled test set			
	0 - Filled	1- Filled	MT- Filled	Random Filled	0 - Filled	1- Filled	MT- Filled	Random Filled
s5378	10.81	12.89	4.34	34.28	11.27	13.33	4.82	34.62
	11.25	13.31	4.80	34.60	11.57	13.63	5.15	34.84
s9234	5.83	17.33	11.46	28.34	12.40	23.11	17.65	33.34
	7.64	18.93	13.17	29.72	13.32	23.91	18.51	34.04
s13207	3.43	12.55	10.49	31.67	8.39	17.04	15.09	35.18
	4.26	13.31	11.27	32.26	9.60	18.14	16.21	36.04
s15850	4.76	14.17	8.95	28.83	12.97	21.57	16.80	34.96
	5.51	14.83	9.66	29.38	14.03	22.52	17.81	35.75
s38417	6.93	10.45	6.10	31.36	15.84	19.02	15.09	37.93
	8.41	11.87	7.59	32.45	16.52	19.68	15.78	38.44
s38584	8.34	18.16	13.70	32.01,	16.77	25.69	21.63	38.26
	14.70	23.84	19.68	36.72	17.54	26.38	22.36	38.83
Avg.	**6.68**	**14.26**	**9.17**	**31.08**	**12.94**	**19.96**	**15.18**	**35.72**
	8.63	**16.02**	**11.03**	**32.52**	**13.76**	**20.71**	**15.97**	**36.32**

Internal Testing

Basic BIST Architecture

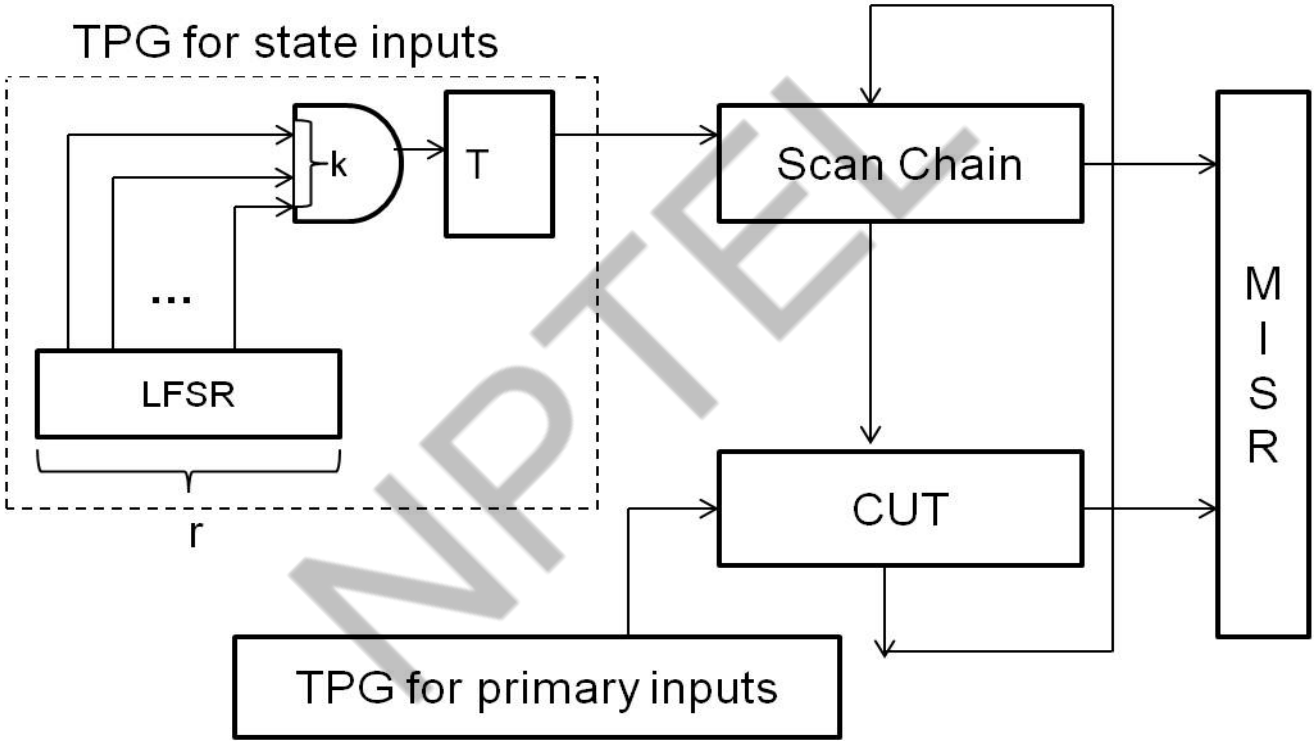


- Major approaches proposed in literature
  - **LFSR Tuning**
    - analyzed the impact of an LFSR's polynomial and seed selection
  - **Vector Filtering BIST**
    - the patterns which don't detect any new faults are filtered out from reaching to the circuit
  - **Low Power Test Pattern Generator**
    - feed highly correlated test stimulus bits among adjacent scan cells in every scan chain or modify the clock scheme

Drawbacks:

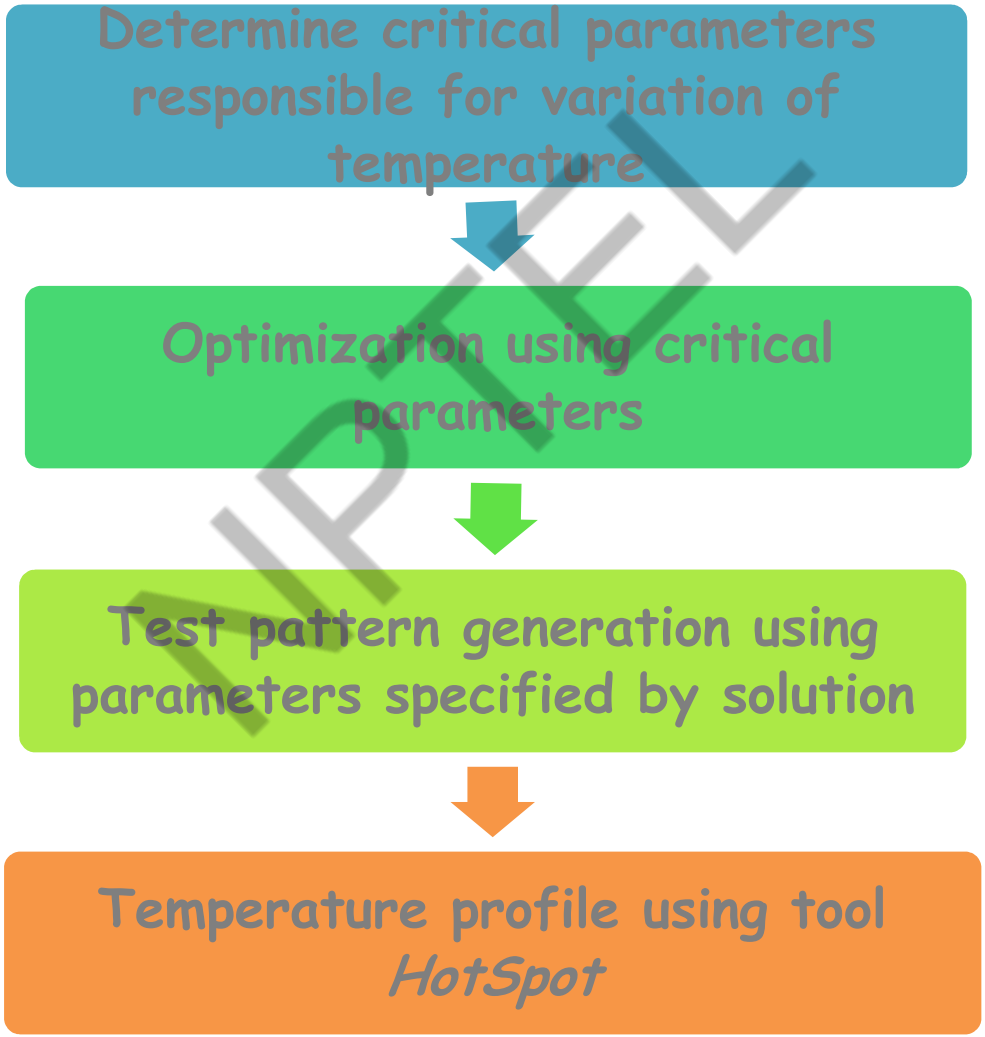
- Do not concentrate on minimizing overall peak temperature and thermal variance of CUT

LT-RTPG Architecture*



* S. Wang and S. K. Gupta, "LT-RTPG: a new test-per-scan BIST TPG for low switching activity," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems , vol.25, no.8, pp.1565,1574, Aug. 2006.

Flow of Work



Results to Determine Critical Parameters for Pattern Generation

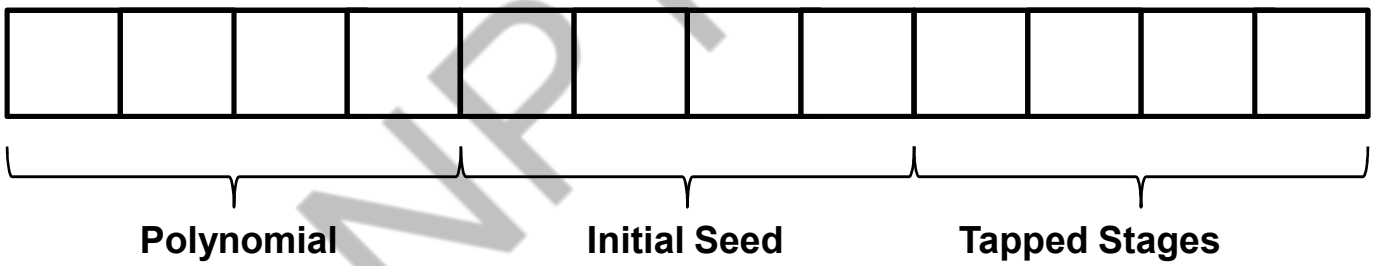
Circuit	Vary polynomial with seed and tapped stages unchanged		Vary seed with polynomial and tapped stages unchanged		Vary tapped stages with polynomial and seed unchanged	
	Max Temperature (K)	Min Temperature (K)	Max Temperature (K)	Min Temperature (K)	Max Temperature (K)	Min Temperature (K)
s5378	414.34	410.07	415.70	405.24	416.45	412.12
s9234	440.98	436.64	440.54	431.23	439.89	432.67
s13207	471.41	467.89	470.91	458.43	469.47	464.09
s15850	449.25	445.44	450.35	445.67	450.92	447.05
s38417	467.34	465.19	467.78	460.56	468.89	464.22
s38584	489.13	484.82	489.45	479.32	486.89	478.51
b14	435.38	431.37	437.18	428.48	438.07	433.60
b15	442.54	436.81	441.76	439.67	443.96	438.03
b20	473.92	469.55	475.63	471.88	476.53	468.24
b21	476.49	472.41	473.94	470.27	478.59	473.75

Critical Parameters

1. Primitive polynomial used by LFSR
2. The initial seed fed to LFSR
3. The tapped stages of the LFSR in LT-RTPG

PSO Formulation

- Particle
  - Composed of three critical parameters
  - Initial particle generated randomly
  - Every bit of the particle assigned to either 0 or 1



- Fitness Function
  - Used fitness function in the first work(X-filling)

Peak Temperature, Fault Coverage (%) and % Reduction in Peak Temperature and Thermal Variance

Circuit	Normal BIST scheme		PSO based LT-RTPG Scheme		% Reduction w.r.t Normal BIST Scheme in	
	Peak Temperature (K)	Fault Coverage (%)	Peak Temperature (K)	Fault Coverage (%)	Peak Temperature (K)	Thermal Variance
s5378	432.14	97.32	360.30	96.41	16.62	92.38
s9234	458.18	87.56	363.15	87.32	20.74	91.62
s13207	476.89	91.78	368.36	89.91	22.76	88.81
s15850	477.37	93.41	369.23	92.78	22.65	88.90
s38417	482.97	88.38	400.43	88.24	17.09	77.28
b14	461.03	91.23	330.40	90.78	28.33	98.63
b15	455.67	97.67	326.38	97.12	28.37	99.05
b20	480.73	95.49	335.26	94.26	30.26	97.53
b21	484.89	92.21	350.49	91.37	27.72	93.63

Lecture 40

Boundary Scan and Core-Based Testing

Outline

- Introduction
- Digital Boundary Scan (1149.1)
- Boundary Scan for Advanced Networks (1149.6)
- Embedded Core Test Standard (1500)
- Comparison between 1149.1 and 1500

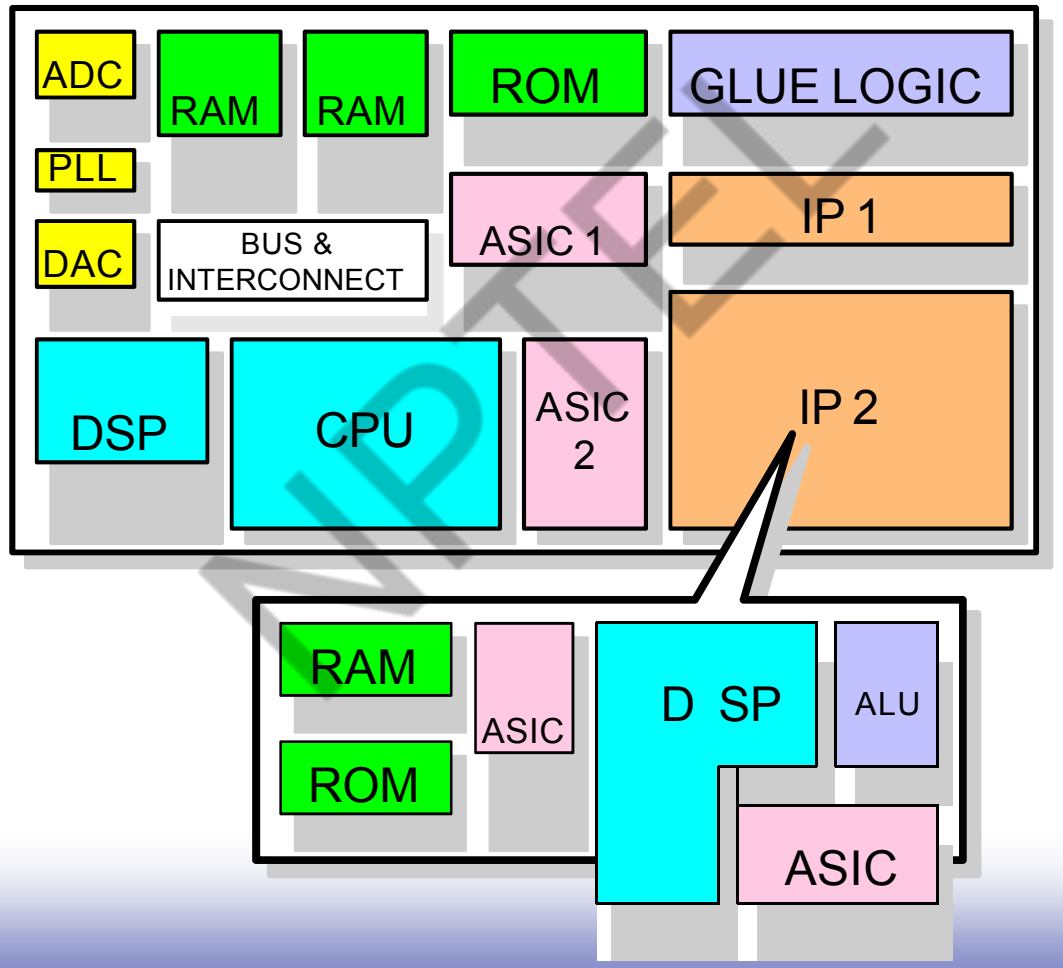
Boundary Scan

- Original objective: board-level digital testing
- Now also apply to:
  - MCM and FPGA
  - Analog circuits and high-speed networks
  - Verification, debugging, clock control, power management, chip reconfiguration, etc.
- History:
  - Mid-1980: JETAG
  - 1988: JTAG
  - 1990: First boundary scan standard – 1149.1

Boundary Scan Family

No.	Main target	Status
1149.1	Digital chips and interconnects among chips	Std. 1149.1-2001
1149.2	Extended digital serial interface	Discontinue
1149.3	Direct access testability interface	Discontinue
1149.4	Mixed-signal test bus	Std. 1149.4-1999
1149.5	Standard module test and maintenance (MTM) bus	Std. 1149.5-1995 (not endorsed by IEEE since 2003)
1149.6	High-speed network interface	Std. 1149.6-2003

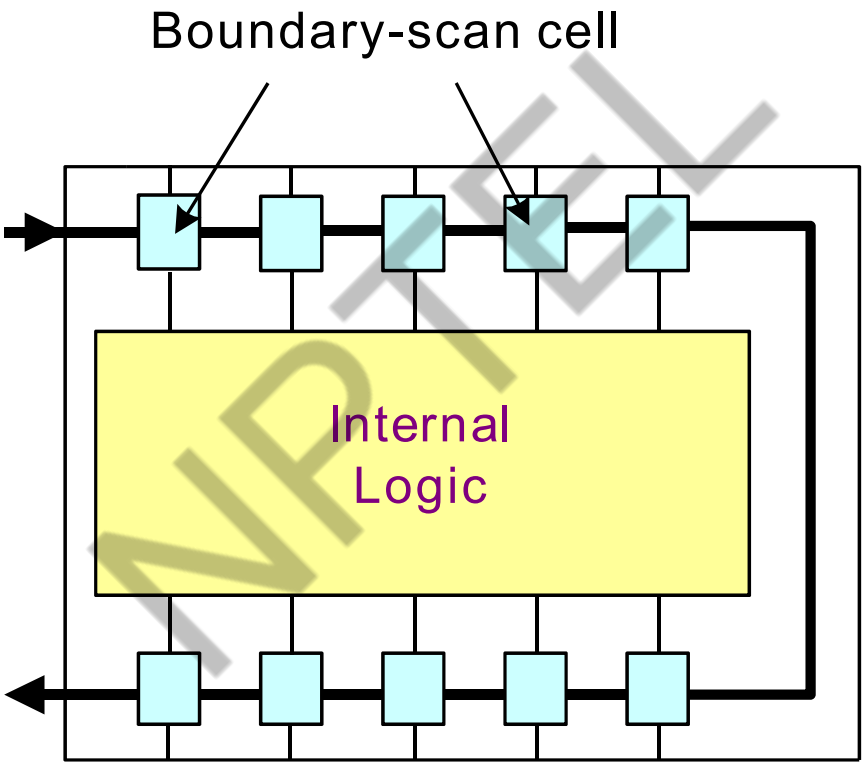
Core-Based SOC Design



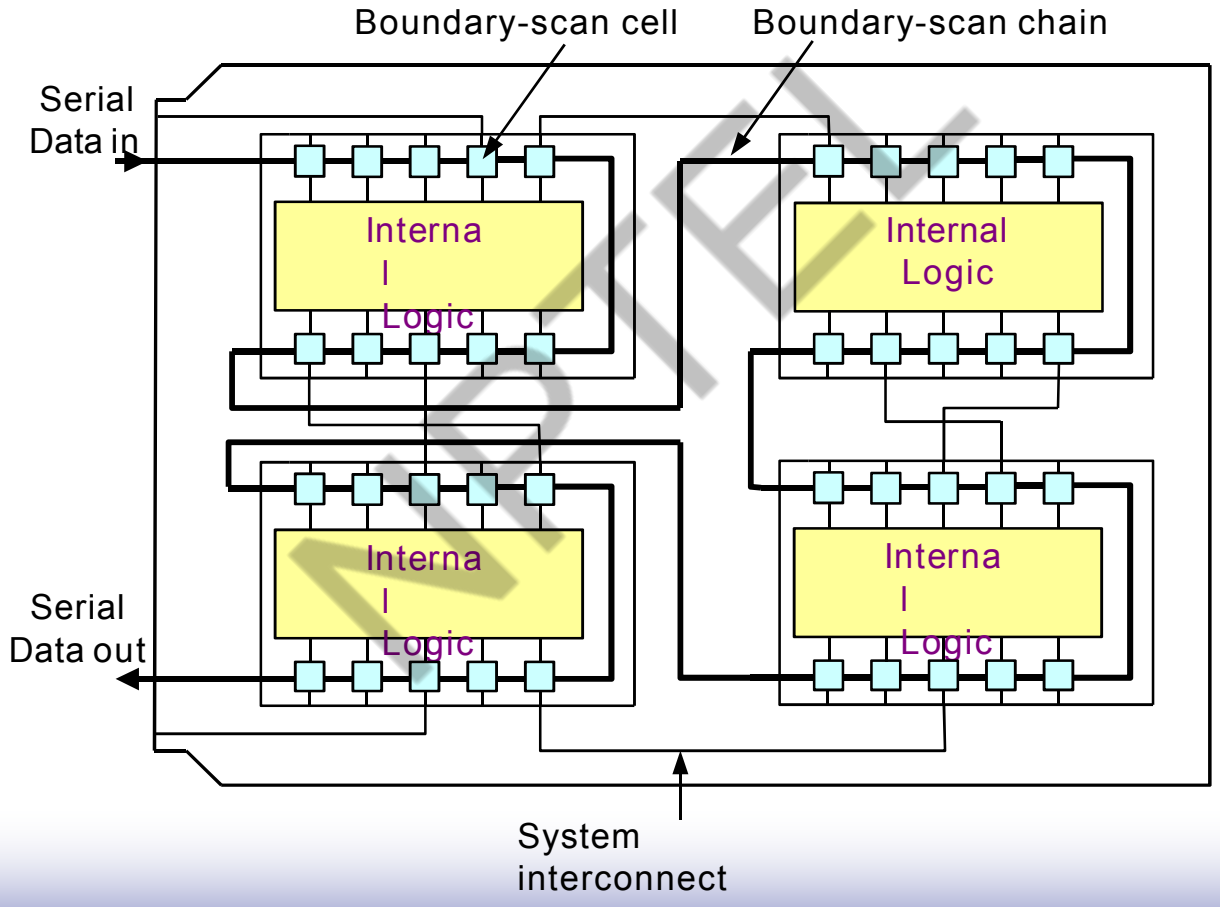
Digital Boundary Scan – 1149.1

- Basic concepts
- Overall test architecture & operations
- Hardware components
- Instruction register & instruction set
- Boundary scan description language
- On-chip test support
- Board/system-level control architectures

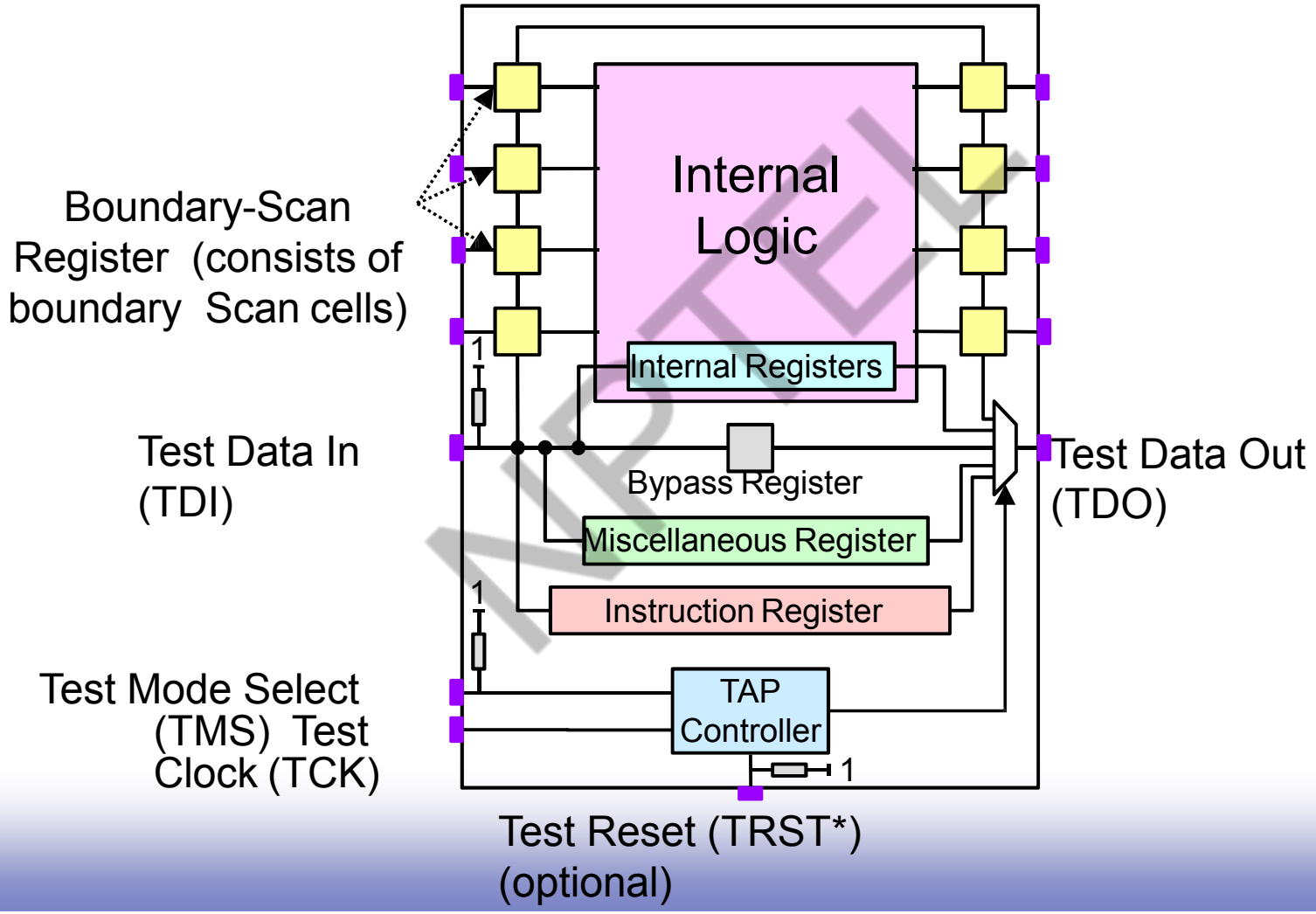
Basic Idea of Boundary Scan



A Board Containing 4 IC's with Boundary Scan



1149.1 Boundary-Scan Architecture



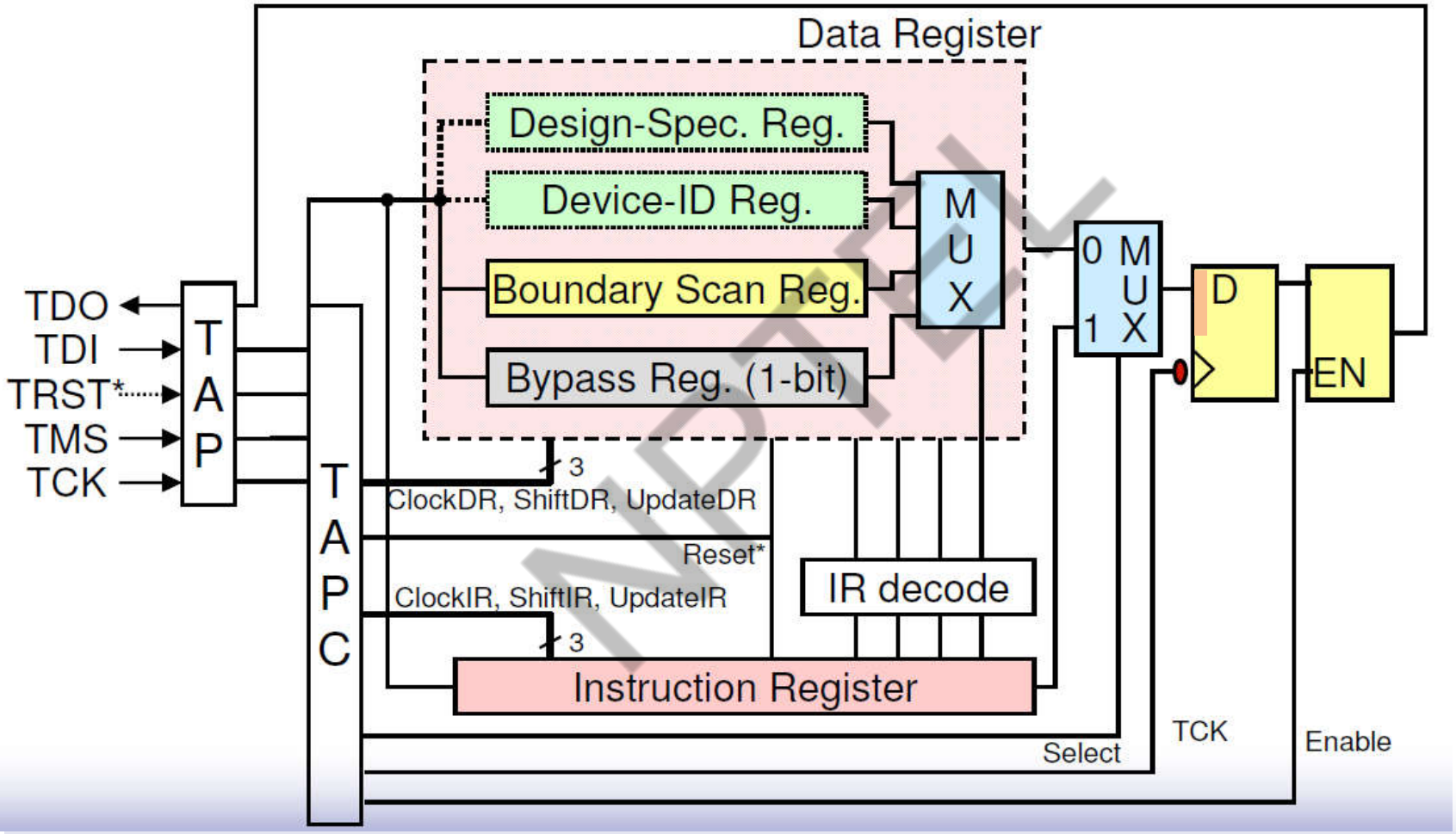
Hardware Components of 1149.1

- A test access port (TAP) consisting of :
  - 4 mandatory pins: Test data in (TDI), Test data out (TDO), Test mode select (TMS), Test clock (TCK), and
  - 1 optional pin: Test reset (TRST)
- A test access port controller (TAPC)
- An instruction register (IR)
- Several test data registers
  - A boundary scan register (BSR) consisting of boundary scan cells (BSCs)
  - A bypass register (BR)
  - Some optional registers (Device-ID register, design-specified registers such as scan registers, LFSRs for BIST, etc.)

Basic Operations

1. Instruction sent (serially) through TDI into instruction register.
2. Selected test circuitry configured to respond to the instruction.
3. Test pattern shifted into selected data register and applied to logic to be tested
4. Test response captured into some data register
5. Captured response shifted out; new test pattern shifted in simultaneously
6. Steps 3-5 repeated until all test patterns are applied.

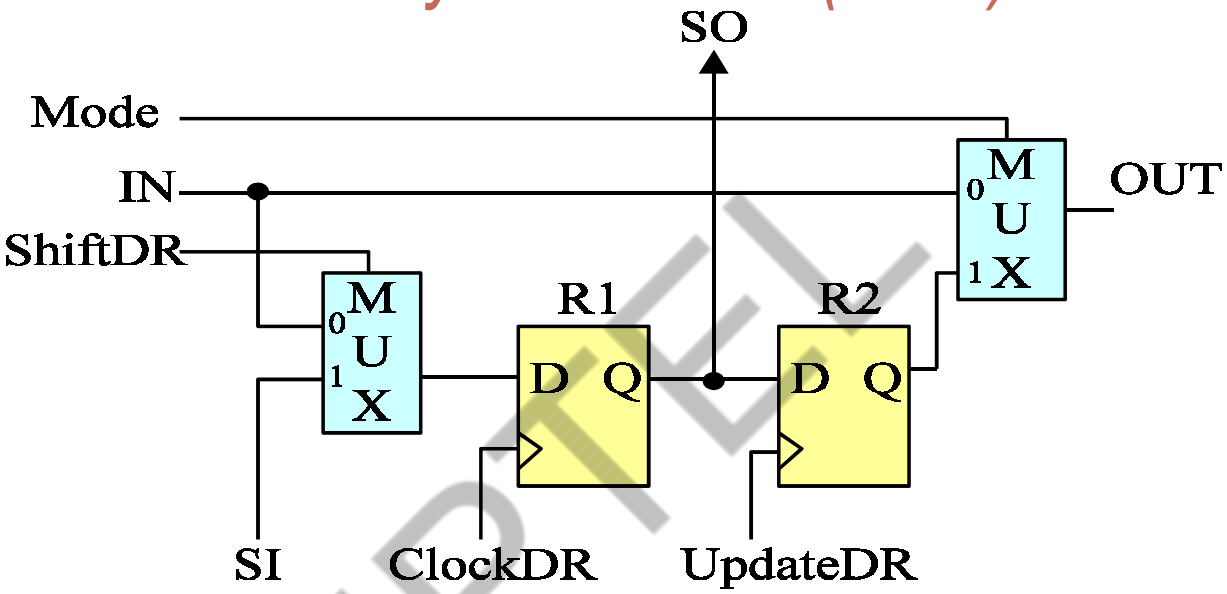
Boundary-Scan Circuitry in A Chip



Data registers

- Boundary scan register: consists of boundary scan cells
- Bypass register: a one-bit register used to pass test signal from a chip when it is not involved in current test operation
- Device-ID register: for the loading of product information (manufacturer, part number, version number, etc.)
- Other user-specified data registers (scan chains, LFSR for BIST, etc.)

A Typical Boundary-Scan Cell (BSC)



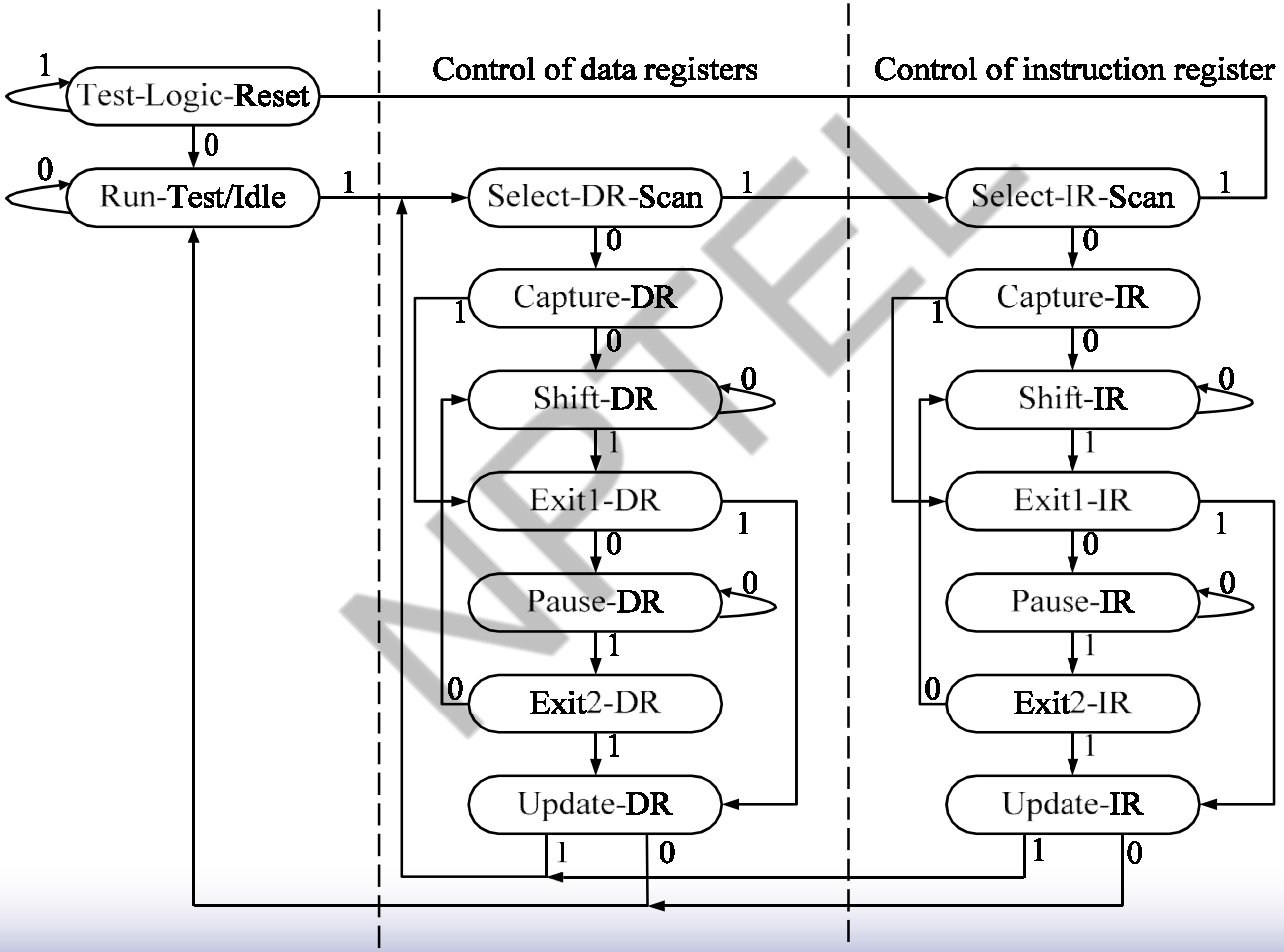
- Operation modes
  - Normal: IN → OUT (Mode = 0)
  - Shift: TDI → ... → IN → OUT → ... → TDO (ShiftDR = 1, ClockDR)
  - Capture: IN → R1, OUT driven by IN or R2 (ShiftDR = 0, ClcokDR)
  - Update: R1 → OUT (Mode_Control = 1, UpdateDR)

Lecture 41

TAP Controller

- A finite state machine with 16 states
- Input: TCK, TMS
- Output: 9 or 10 signals included ClockDR, UpdateDR, ShiftDR, ClockIR, UpdateIR, ShiftIR, Select, Enable, TCK and TRST* (optional).

State Diagram of TAP Controller



Main functions of TAP controller

- Providing control signals to
  - Reset BS circuitry
  - Load instructions into instruction register
  - Perform test capture operation
  - Perform test update operation
  - Shift test data in and out

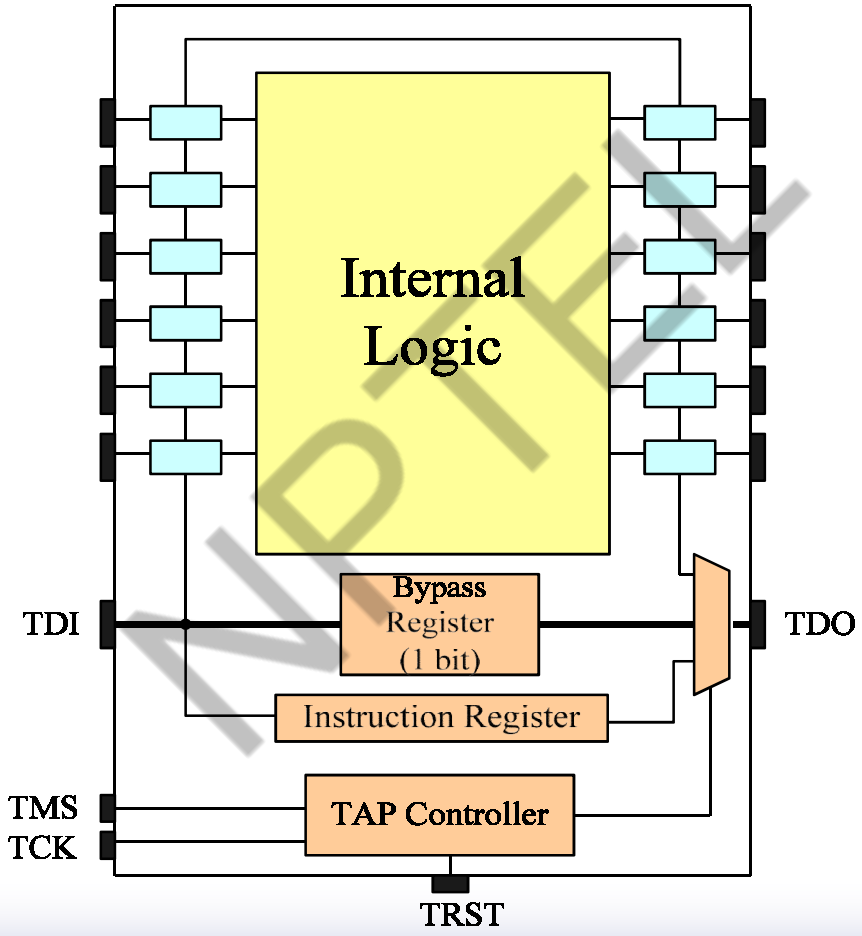
States of TAP Controller

- Test-Logic-Reset: normal mode
- Run-Test/Idle: wait for internal test such as BIST
- Select-DR-Scan: initiate a data-scan sequence
- Capture-DR: load test data in parallel
- Shift-DR: load test data in series
- Exit1-DR: finish phase-1 shifting of data
- Pause-DR: temporarily hold the scan operation (e.g., allow the bus master to reload data)
- Exit2-DR: finish phase-2 shifting of data
- Update-DR: parallel load from associated shift registers Note: Controls for IR are similar to those for DR.

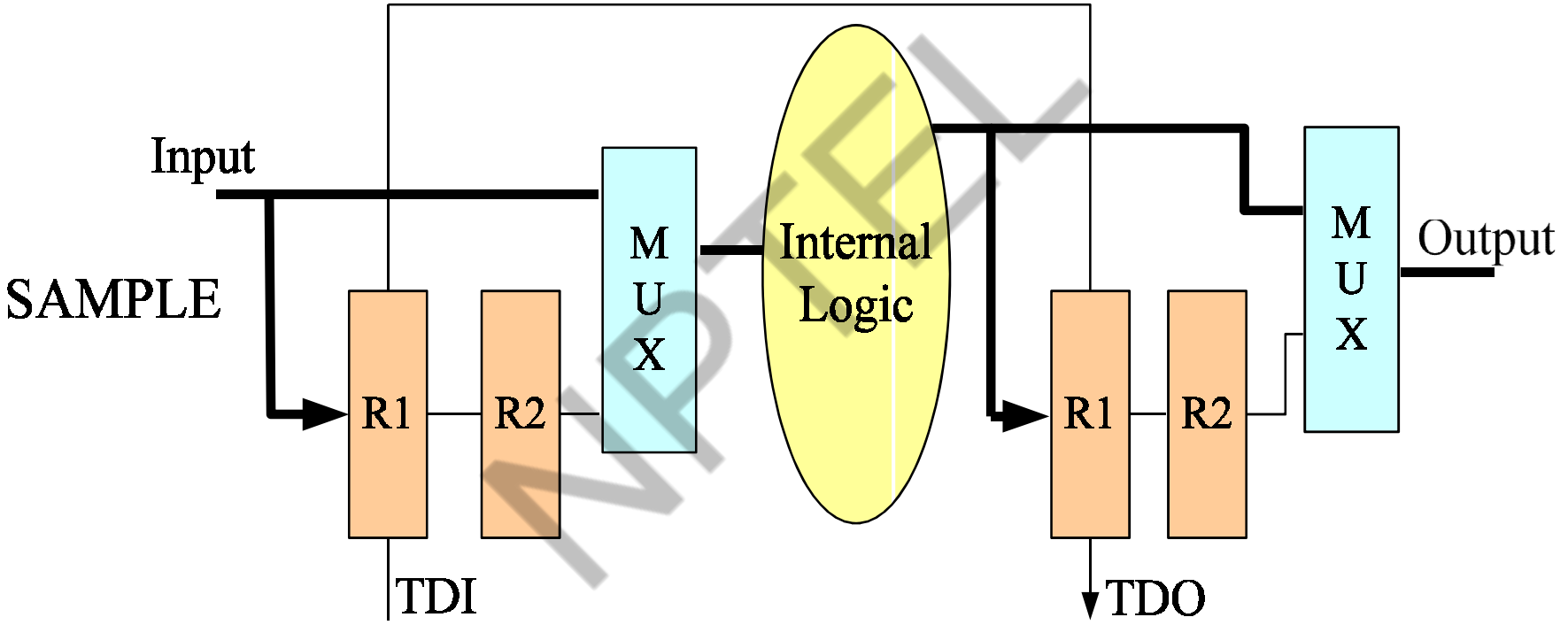
Instruction Set

- BYPASS
  - Bypass data through a chip
- SAMPLE
  - Sample (capture) test data into BSR
- PRELOAD
  - Shift-in test data and update BSR
- EXTEST
  - Test interconnection between chips of board
- Optional
  - INTEST, RUNBIST, CLAMP, IDCODE, USERCODE, HIGH-Z, etc.

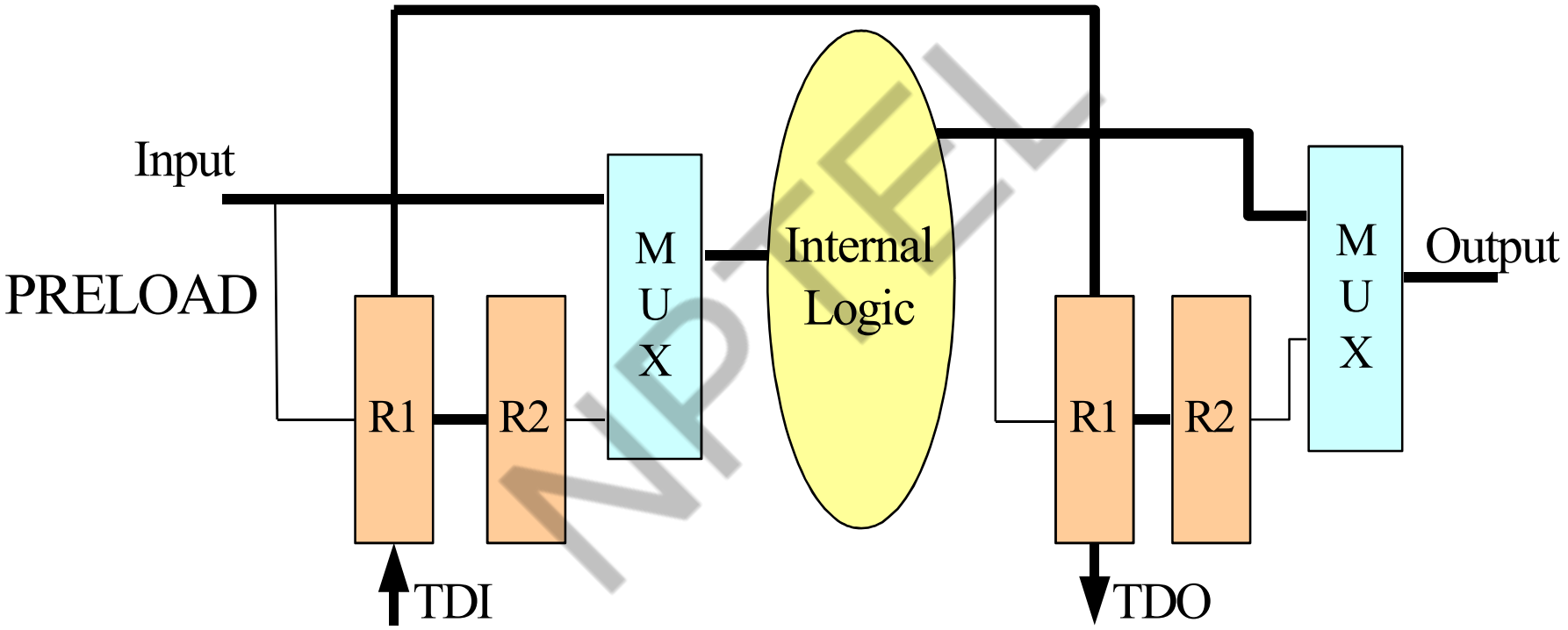
Execution of BYPASS Instruction



Execution of SAMPLE Instruction

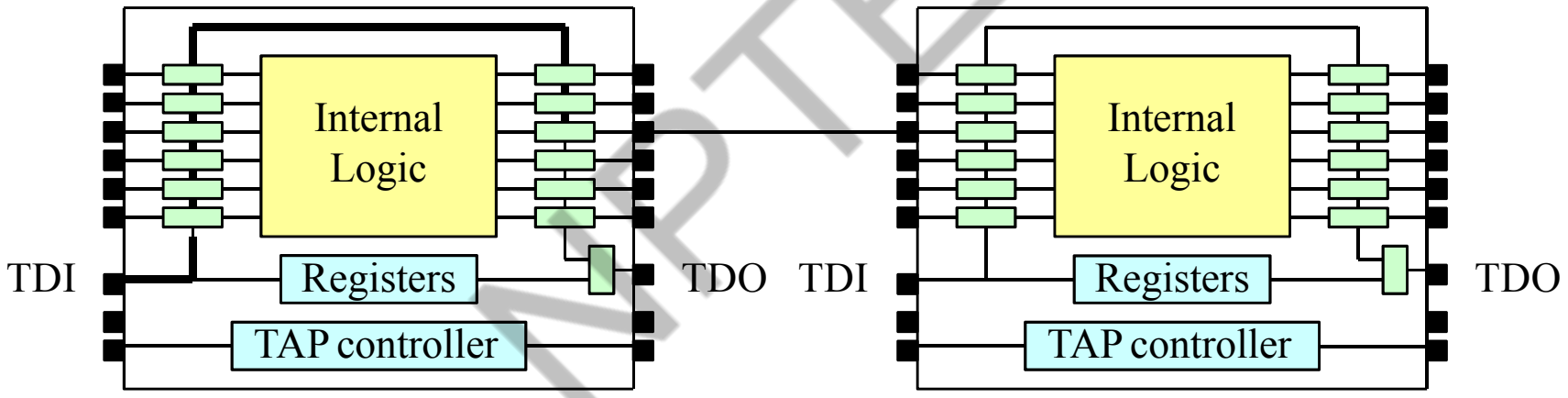


Execution of PRELOAD Instruction



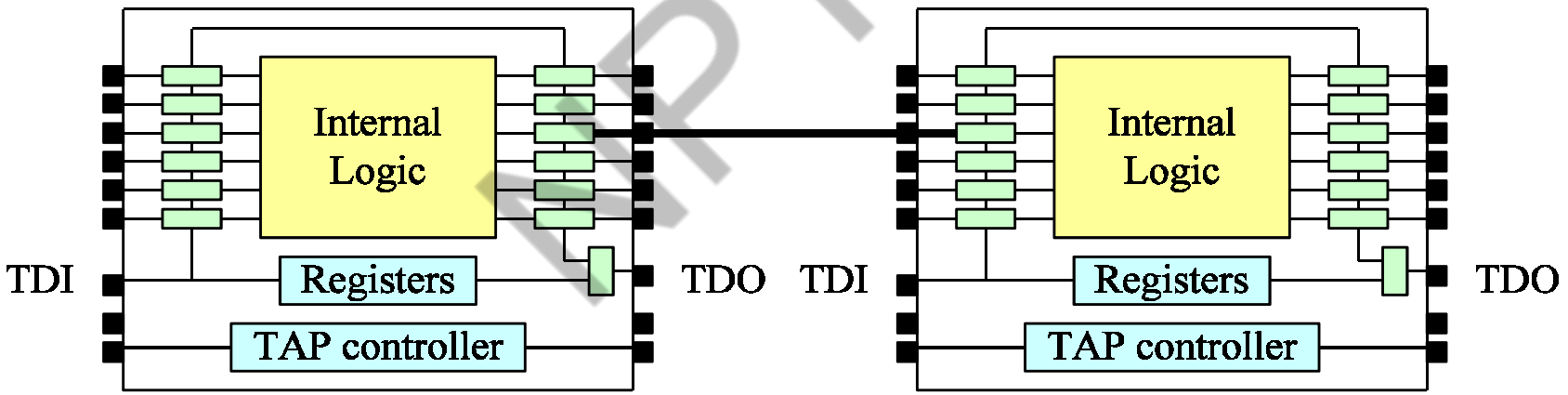
Execution of EXTEST Instruction (1/3)

- Shift-DR(Chip1)



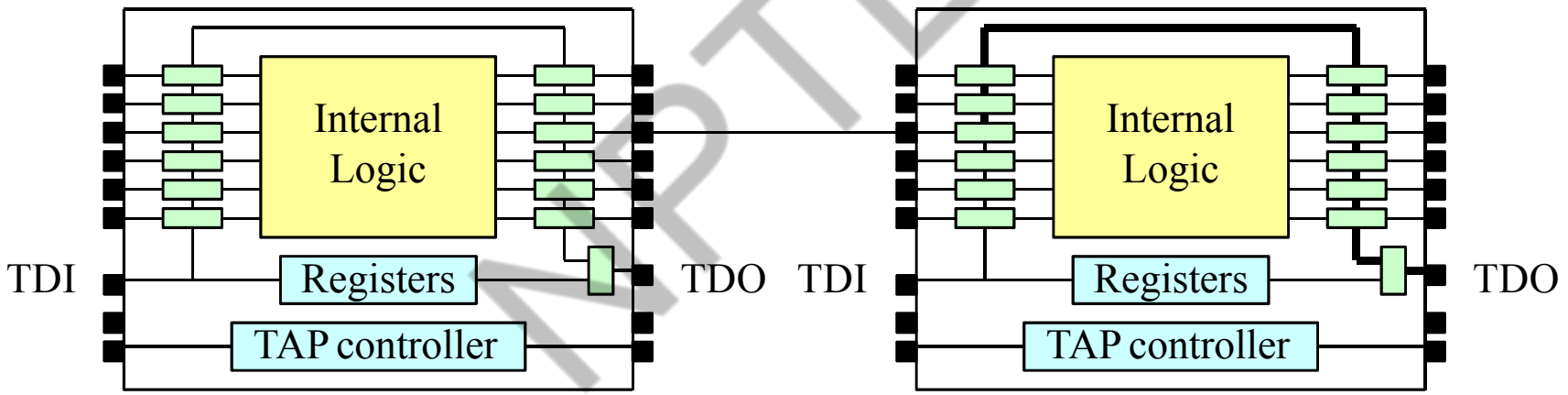
Execution of EXTEST Instruction (2/3)

- Update-DR(Chip1)
- Capture-DR (Chip2)



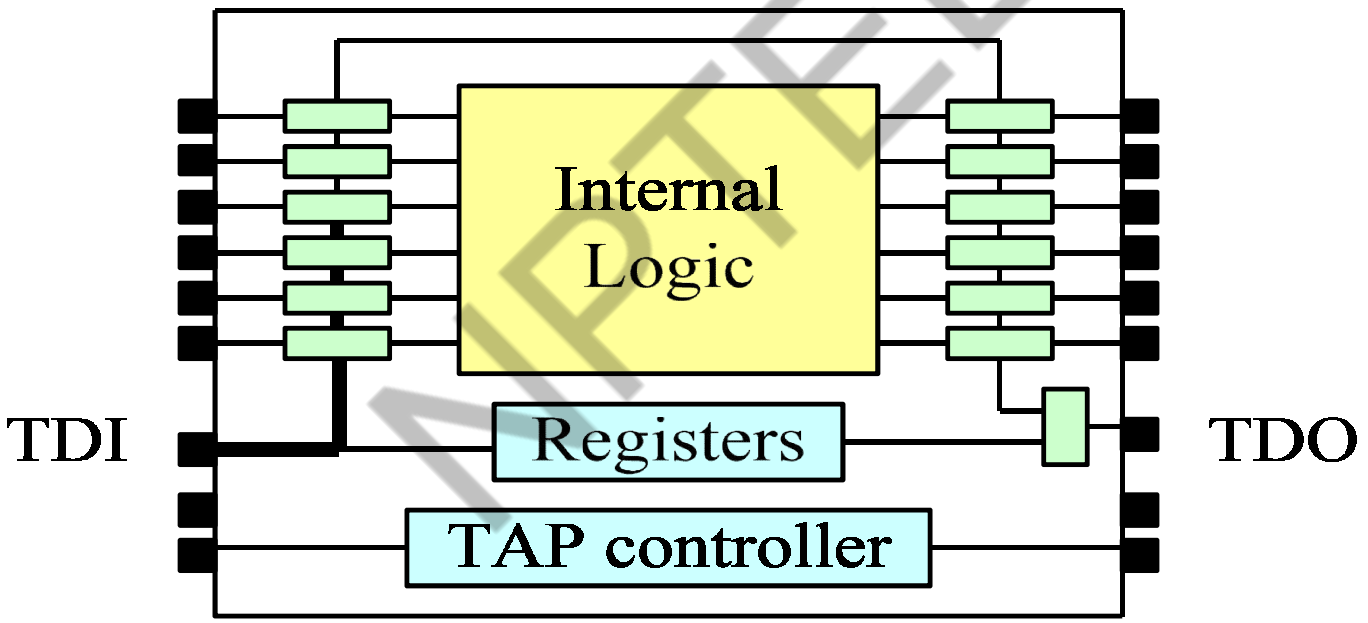
Execution of EXTEST Instruction (3/3)

- Shift-DR(Chip2)



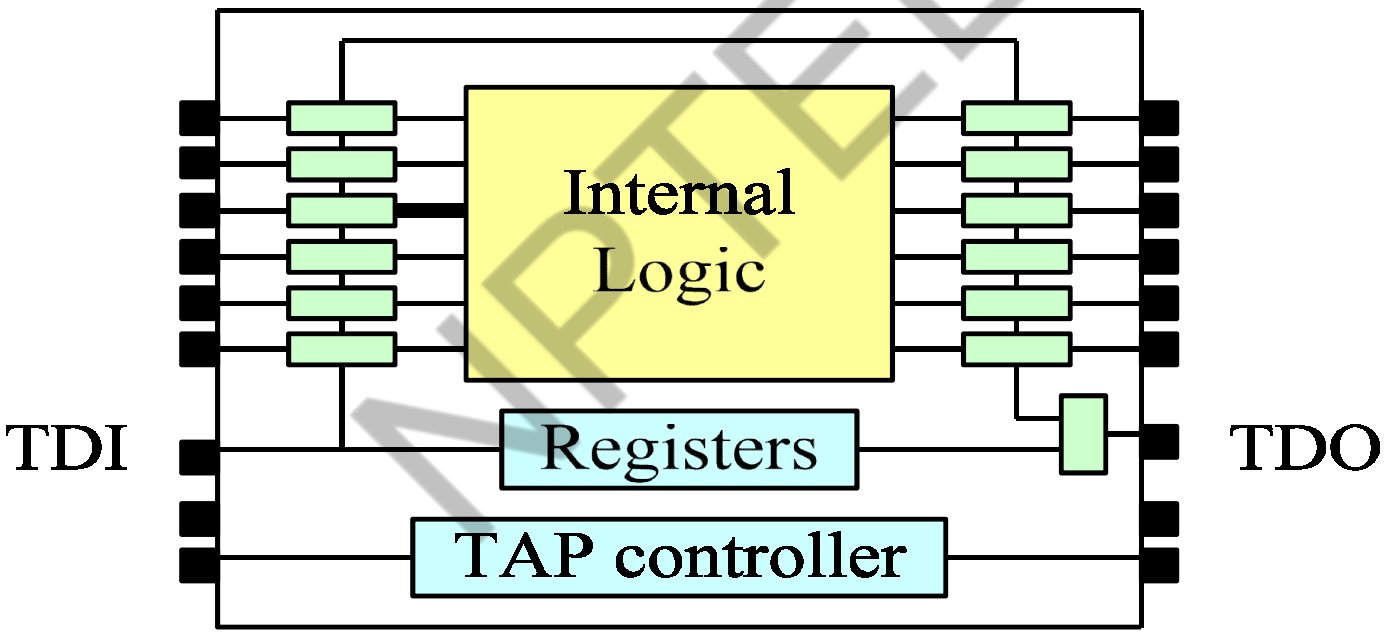
Execution of INTEST Instruction (1/4)

- Shift-DR



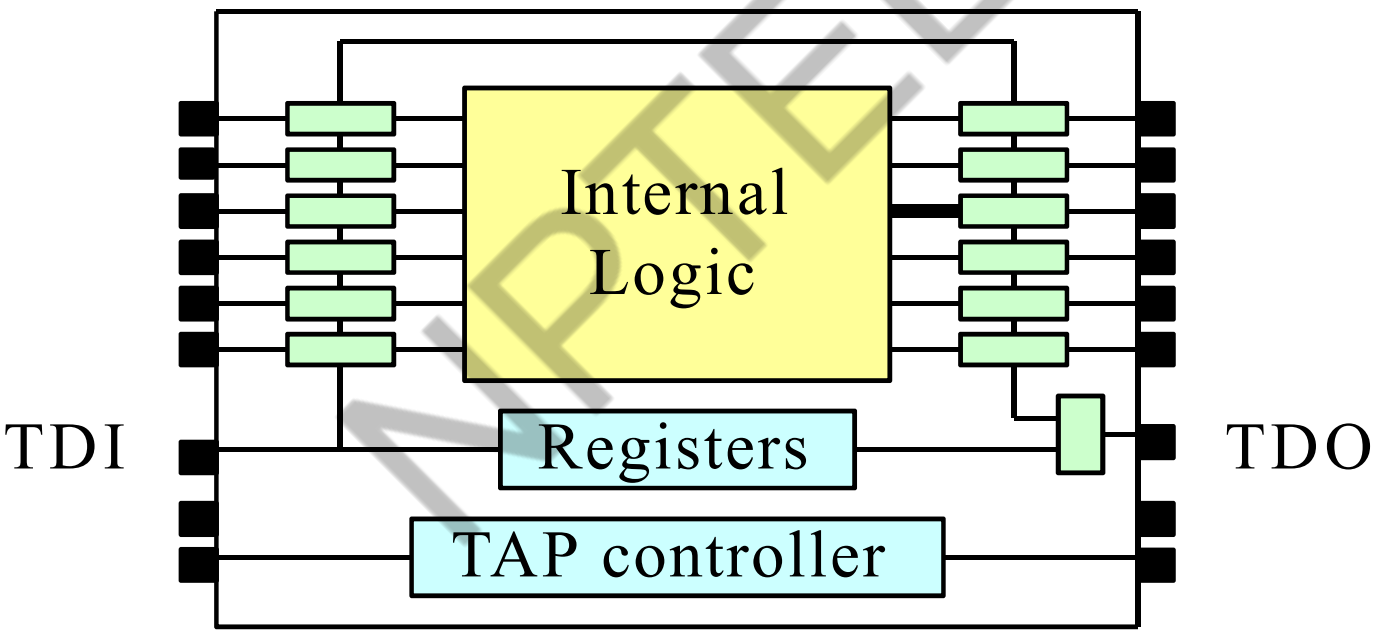
Execution of INTEST Instruction (2/4)

- Update-DR



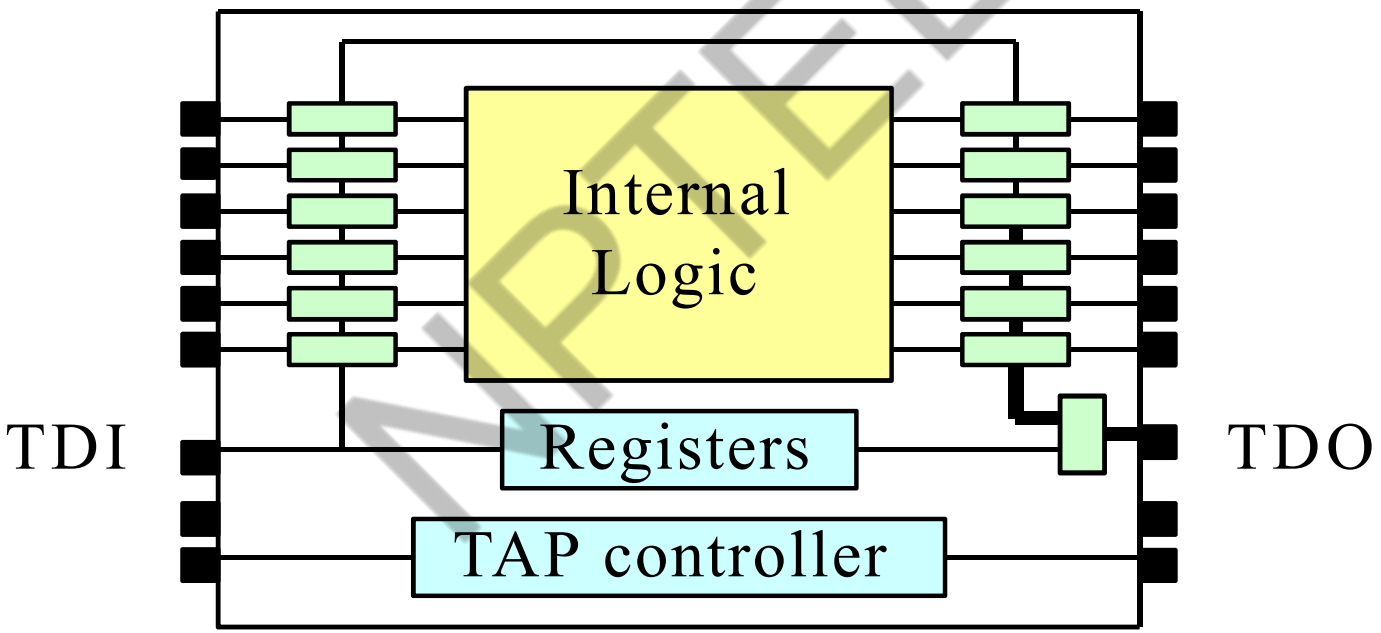
Execution of INTEST Instruction (3/4)

- Capture-DR



Execution of INTEST Instruction (4/4)

- Shift-DR



Boundary Scan Description Language (BSDL)

- Now a part of IEEE 1149.1-2001
- Purposes:
  - Provide standard description language for BS devices.
  - Simplify design work for BS – automated synthesis is possible.
  - Promote consistency throughout ASIC designers, device manufacturers, foundries, test developers and ATE manufacturers.
  - Make it easy to incorporation BS into software tools for test generation, analysis and failure diagnosis.
  - Reduce possibility of human error when employing boundary scan in a design.

Features of BSDL

- Describes the testability features of BS devices that are compatible with 1149.1.
- S subset of VHDL.
- System-logic and the 1149.1 elements that are absolutely mandatory need not be specified.
  - Examples: BYPASS register, TAP controller, etc.
- Commercial tools to synthesize BSDL exist.

Scan and BIST Support with Boundary Scan