Random Access Scan - II

Virendra Singh

Associate Professor

Computer Architecture and Dependable Systems Lab

Dept. of Electrical Engineering

Indian Institute of Technology Bombay

http://www.ee.iitb.ac.in/~viren

viren@ee.iitb.ac.in

EE 709: Testing & Verification of VLSI Circuits

Lecture – 24 (Feb 28, 2012)

Random Access Scan

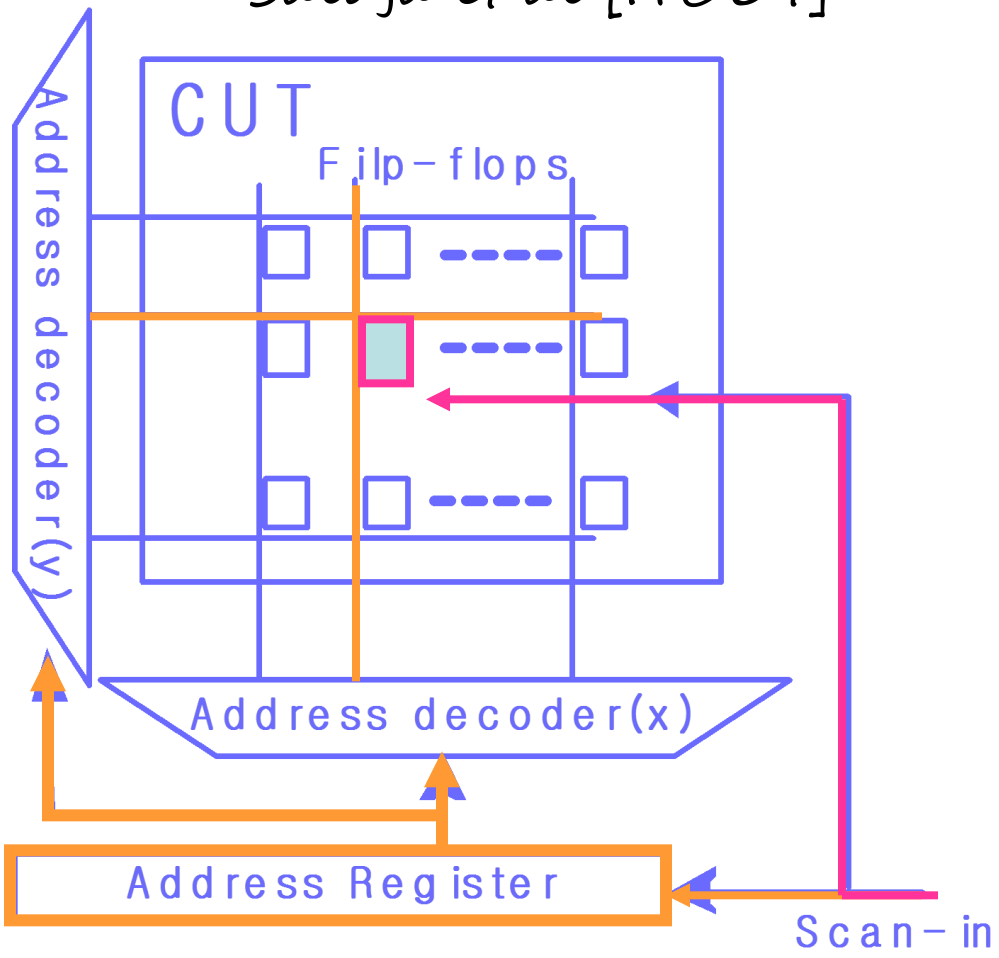
- First proposed by Ando in 1980
- It was considered impractical due to large area overhead.
- Baik at al. revisited it in 2004 [ITC'04]
- Proposed as a simultaneous solution to test power, test time, and test data volume
- Baik,2005, proposed PRAS which showed around 3x speed up, reduction in test data volume with only minor increase in area compared to Serial Scan, thus making RAS practical

Random Access Scan: Architecture

A solution to test power, test time and test data volume

- Architecture
- Each FF has unique address
- Address shift register
- X-Y Decoder
- Select FF to write/read

Saluja et al [ITC'04]

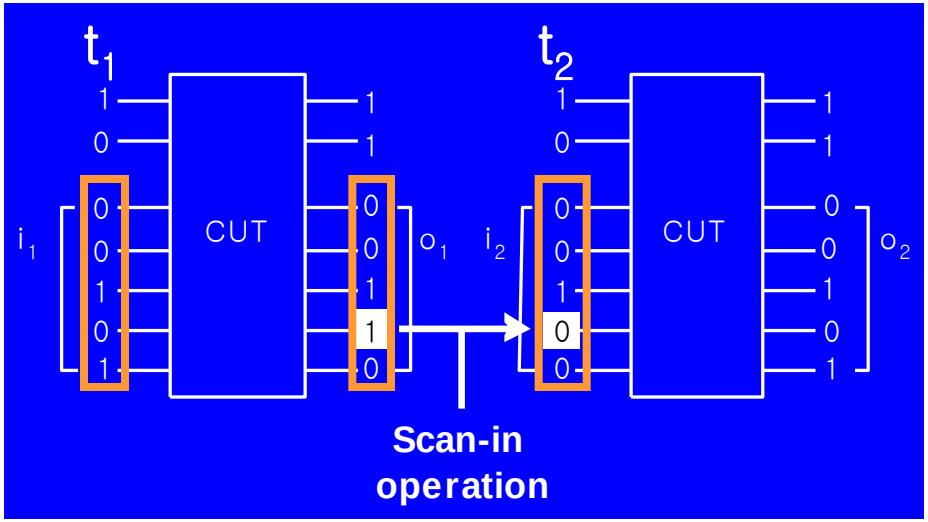


Scan Operation Example

- Test vector

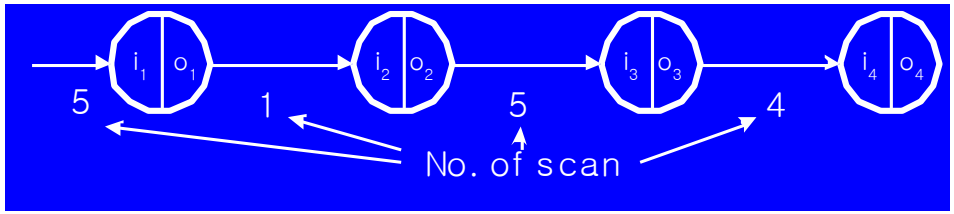
Test	PPI(i_i)	PPO(o_i)
t1	00101	00110
t2	00100	00101
t3	11010	11010
t4	00111	01011

- Scan operation for *t2*



- Complete test application

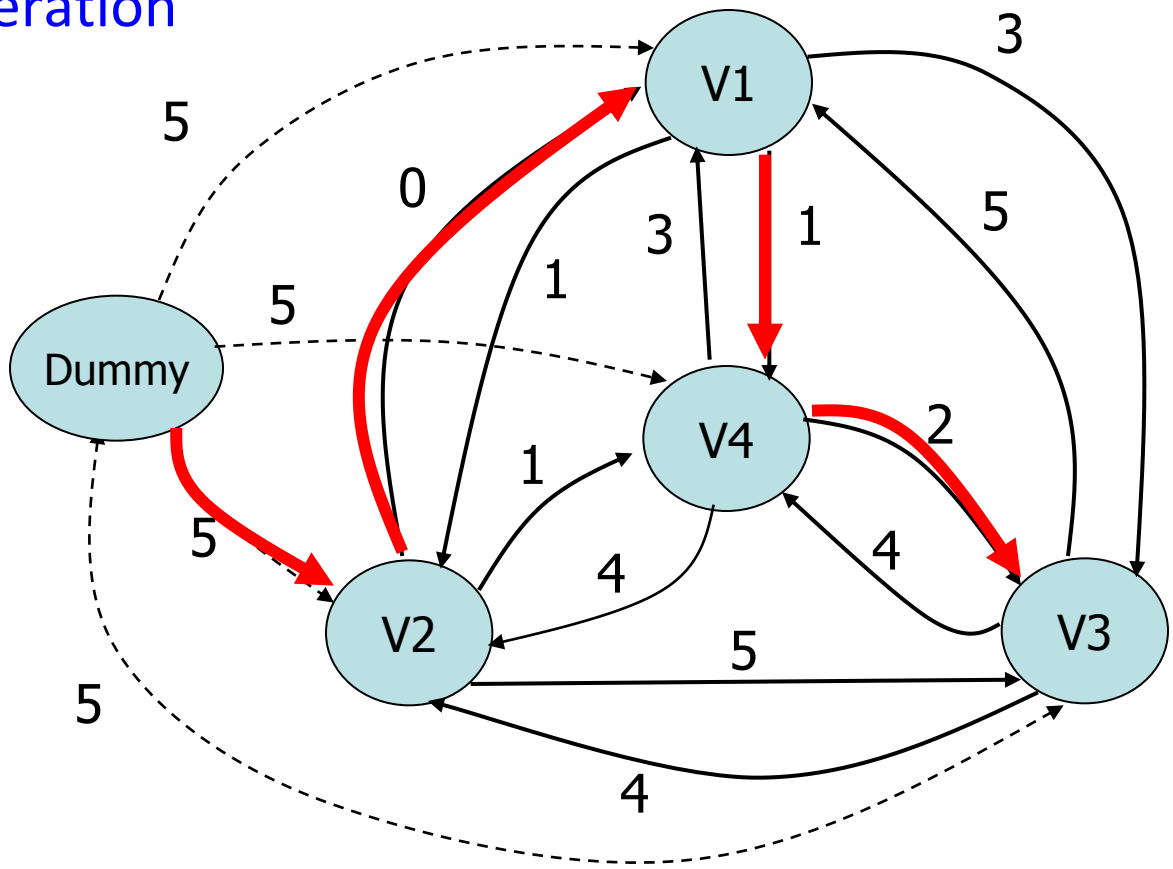
  Total number of scan operation = 15



Test Vector Ordering

- Test data volume and Test application time is proportional to the random access scan operation
- Goal: Reduce # scan operation

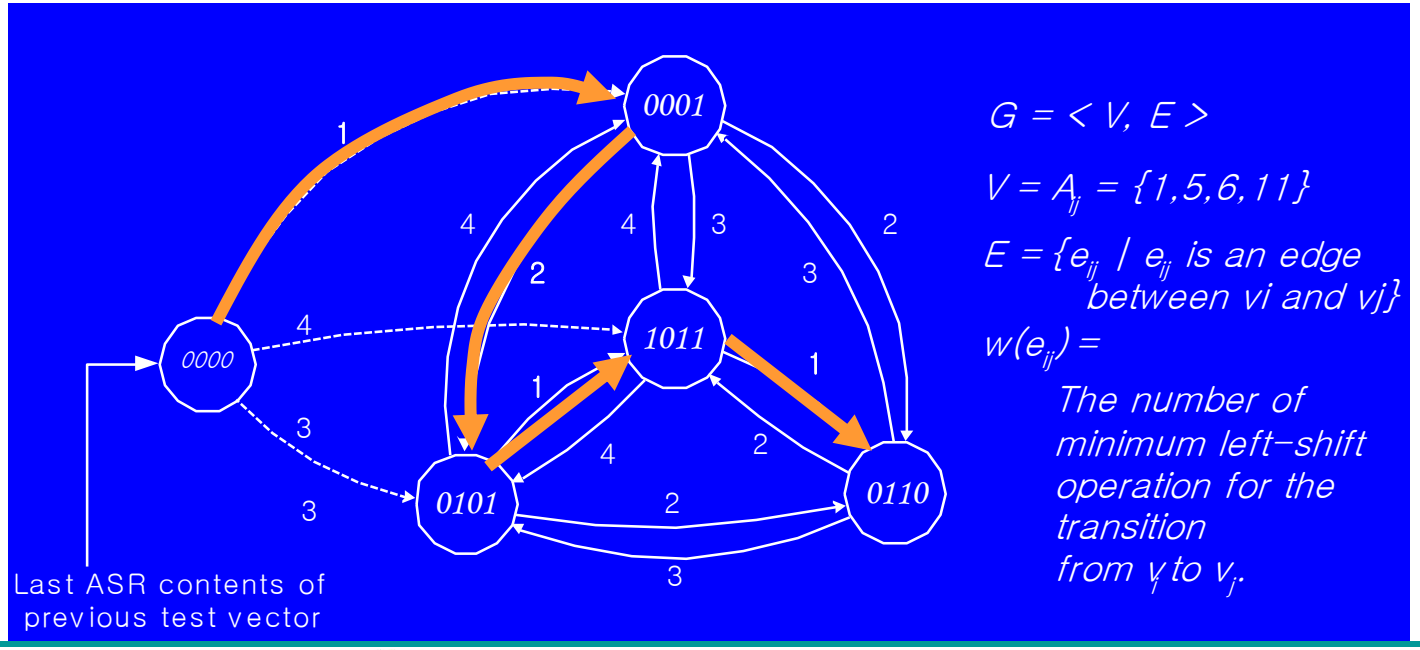
Test	PPI(i_i)	PPO(o_i)
t1	00101	00110
t2	00100	00101
t3	11010	11010
t4	00111	01011



Scan operation = 8

Optimizing Address Scan

- The cost of address shifting
  - # of scan operation x ASR width
  - Example address set = { 1, 5, 6, 11 } for 4-bit ASR
  - 4 X 4 = 16
- Proper ordering of address can minimize shifting cost
  - Apply 11(1011) after 5(0101) → needs only 1 left-shift
- Minimizing address shifting cost
  - Construct Address Shifting Distance Graph (ASD-graph)
  - Find min-cost Hamiltonian path using ATSP algorithm (Result : 5 shifts)



Hamming Distance Reduction

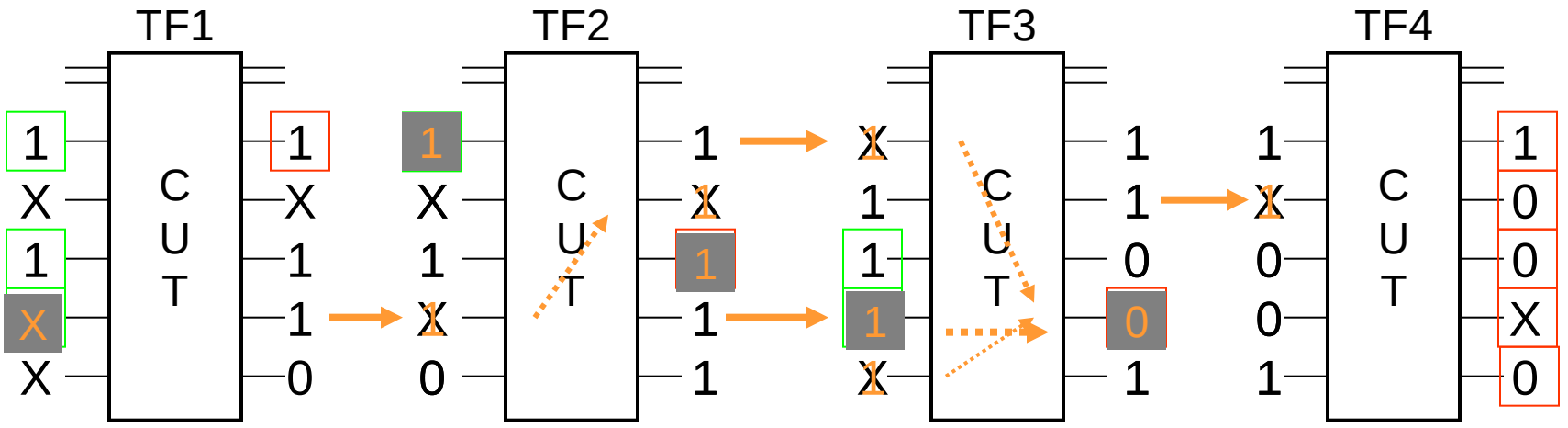
- Don't care values in PPI do not need scan operation
  - Use Don't care identification method

    Fully specified test vector → Vectors w/ X values on targeted bit positions without loss of fault coverage

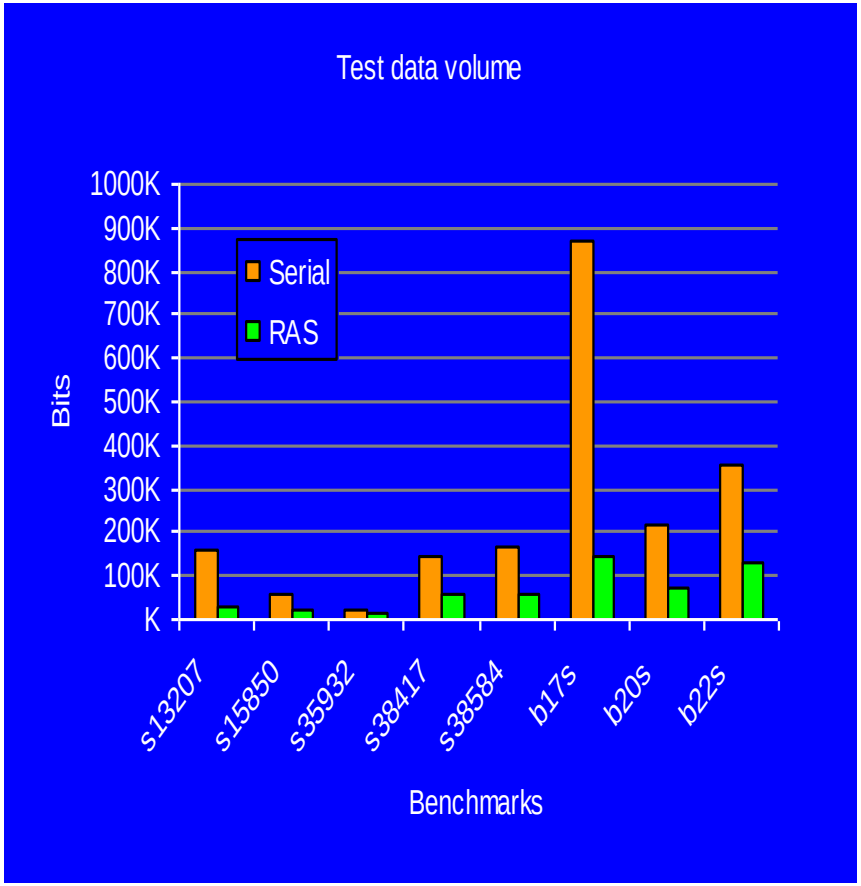
1. Before vector ordering: Identify don't cares in PPI
2. Vector ordering
3. Simulate test vector in order / Fill X's with previous vectors PPO
4. Identify more X's on targeted bit in PPI
   - odd vector
   - even vector
5. Repeat 3,4 until no more X's are identified

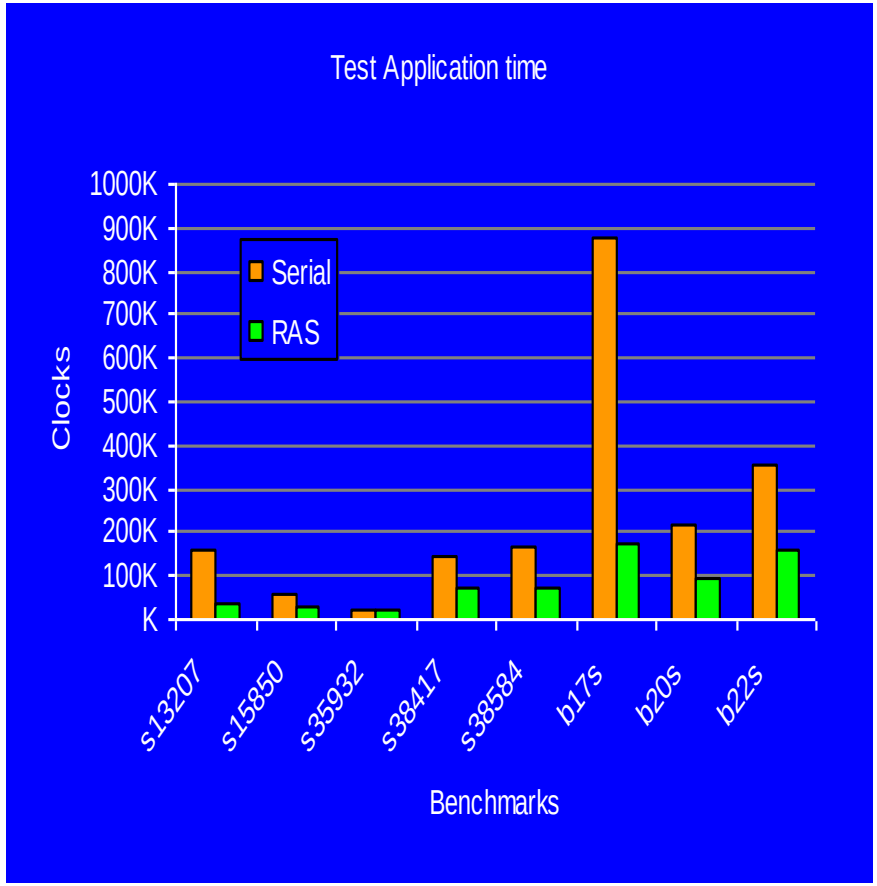
Change allowed

Targeted



Result (Test Time/Data)

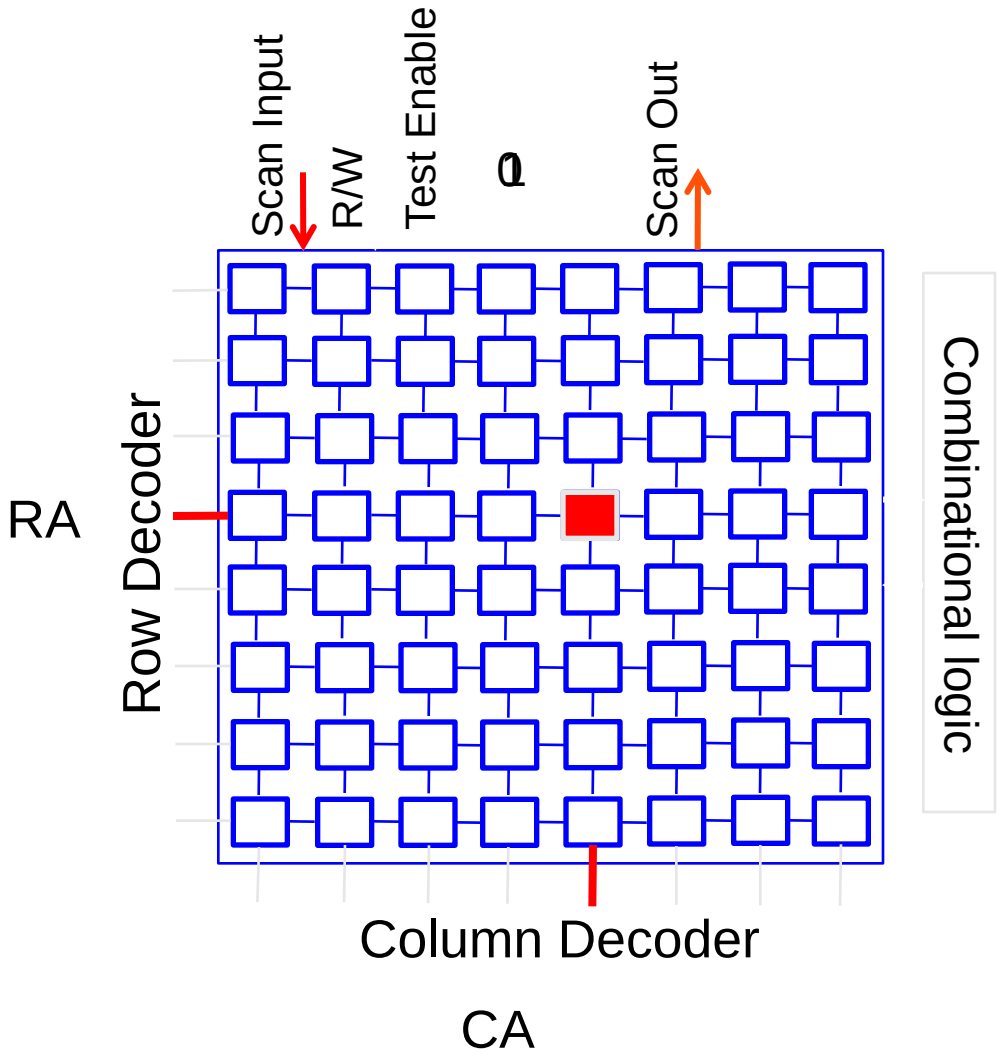




Can test time be further reduced?

Random Access Scan

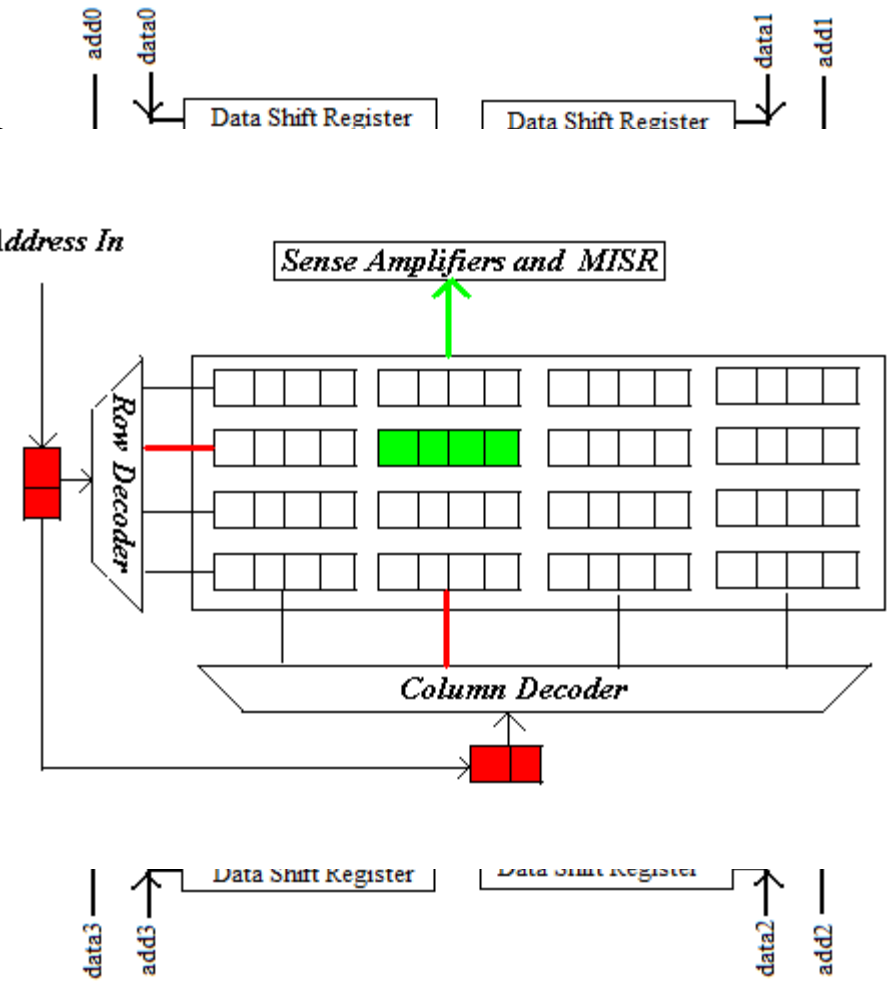
- Generic Architecture of RAS
- RAS in normal mode
- Application of test vector using RAS
- Reading of test response



Serial Input Random Access Scan

Adiga, VLSID'10

- Architectur
- SIRAS in fu
- SRAM type
- Test Applic
- Test Respo

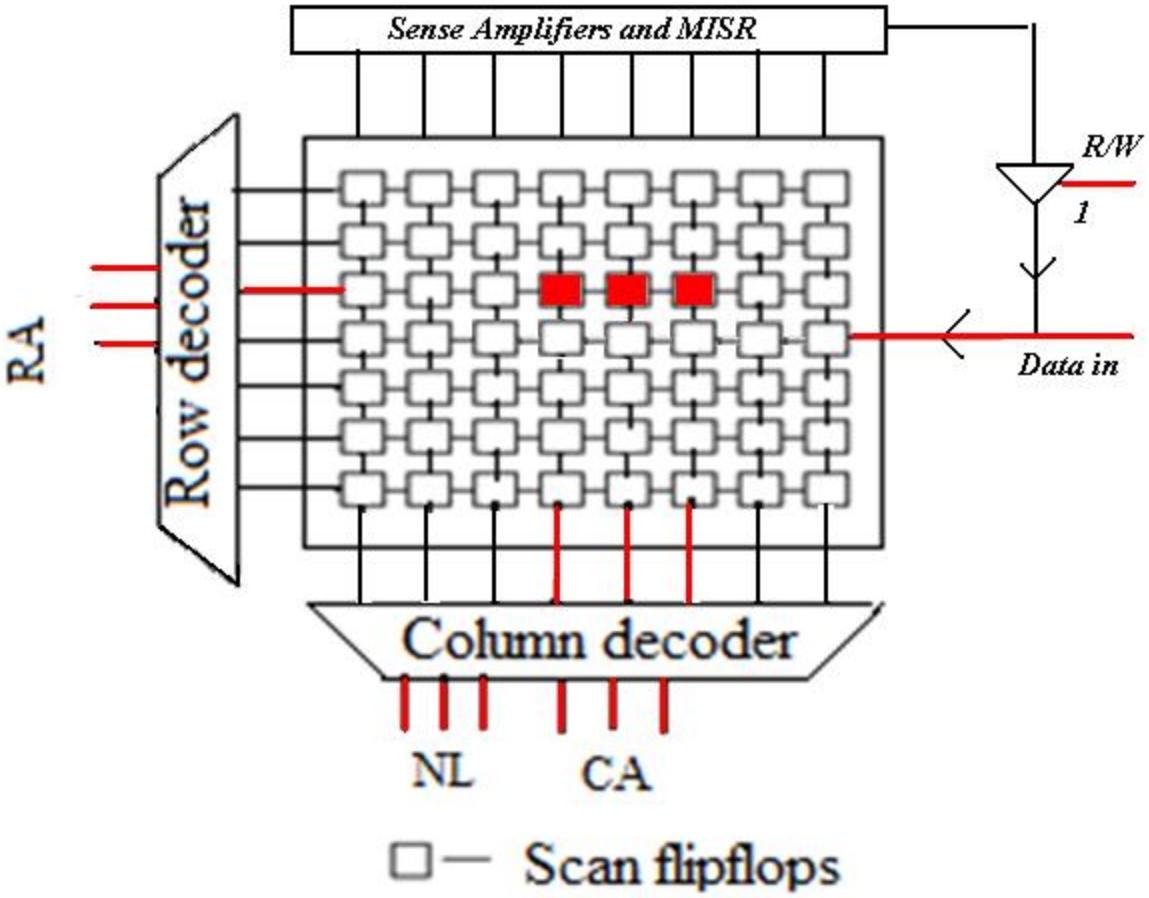


RAS

Variable Word Length Random Access Scan

Adiga, VLSID'10

- **Architecture o**
- **In functional n**
- **Test Applicatio**
- **Test Respons**



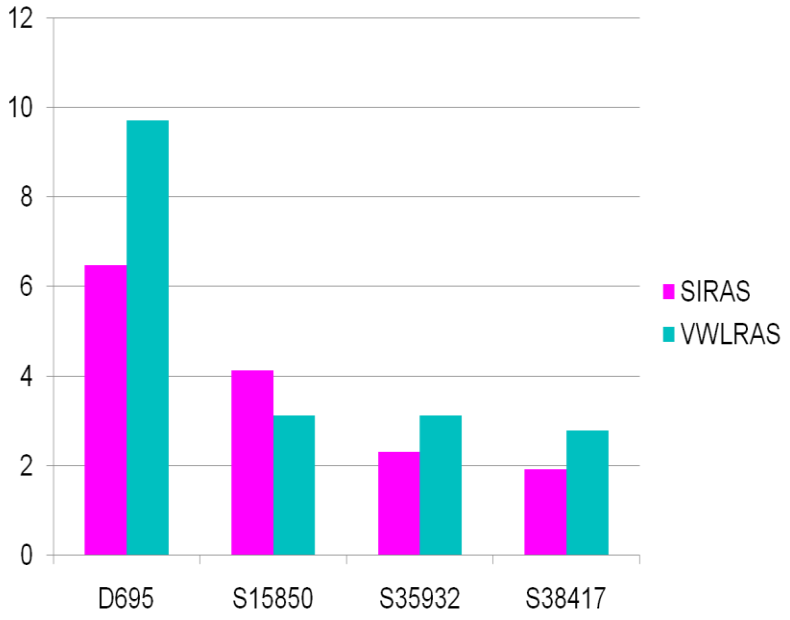
Flip Flop Grouping

- Both VWLRAS and SIRAS require Flip Flops to be grouped in order to reduce the test application time

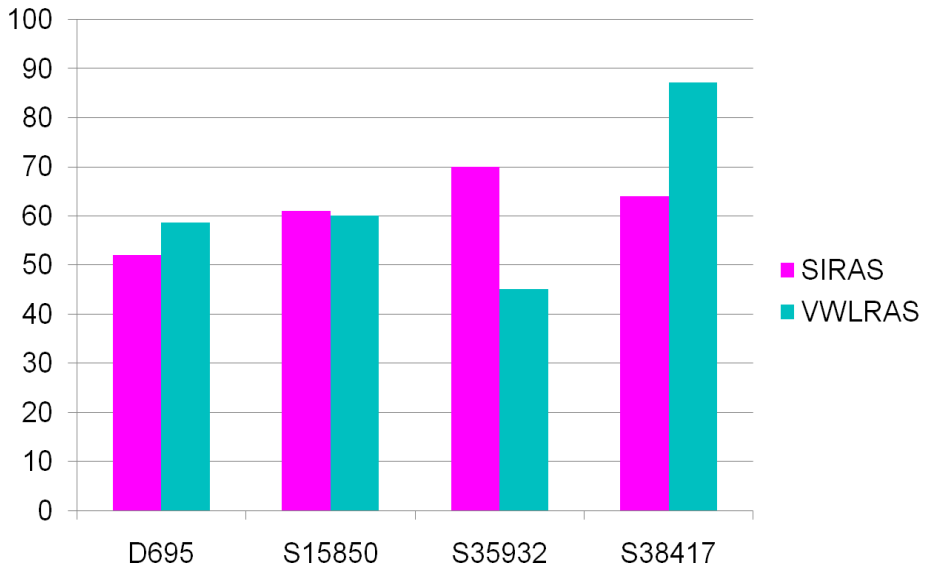
- Flip Flops are grouped by constructing the graph using Flip Flops as nodes and edges weights are calculated using the hamming distance like calculation between test vectors using weight matrix.

Comparison of SIRAS, VWLRAS and RAS

TT Reduction

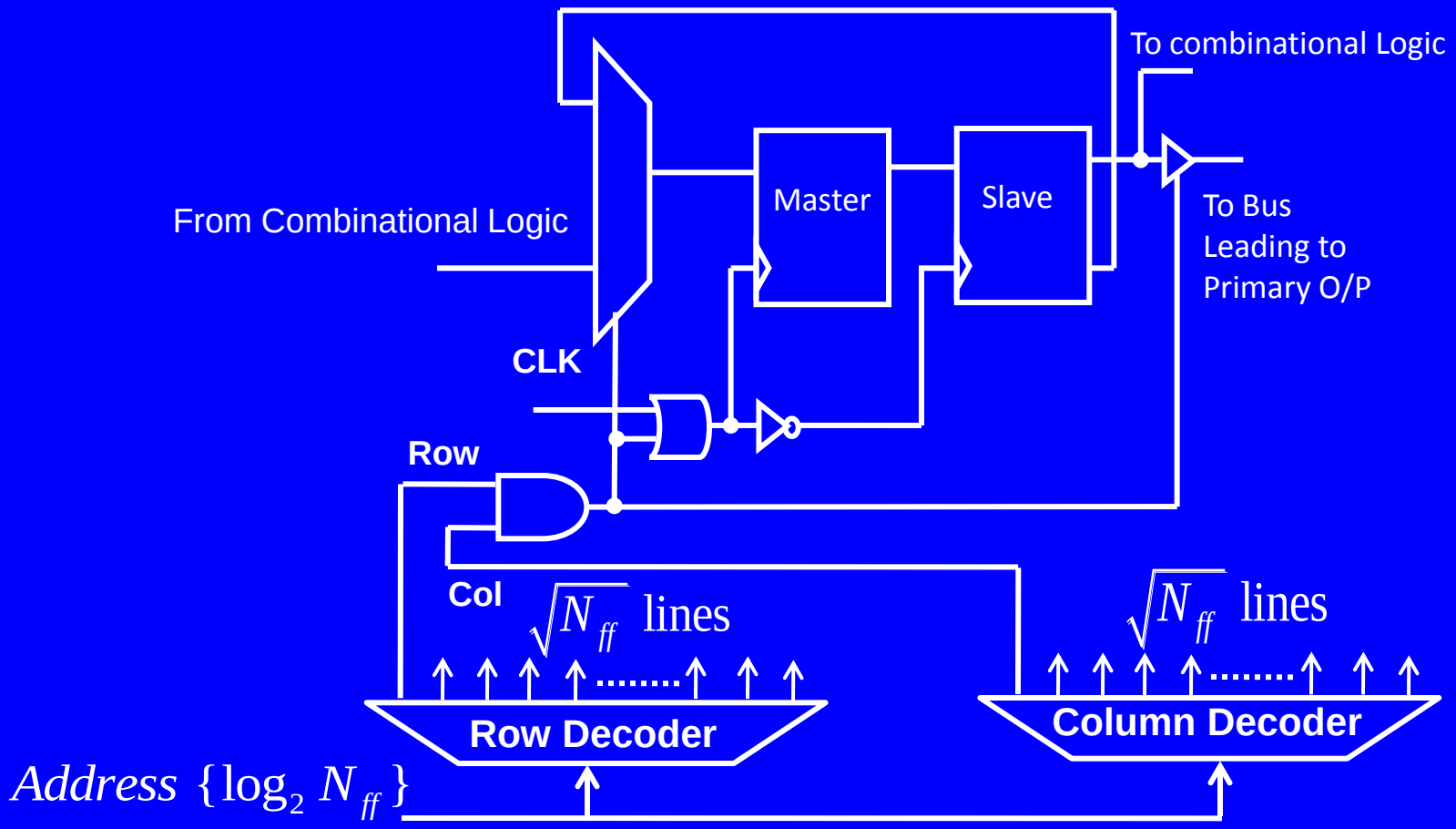


TDV Reduction



T-Flip-Flop based Scan cell for RAS

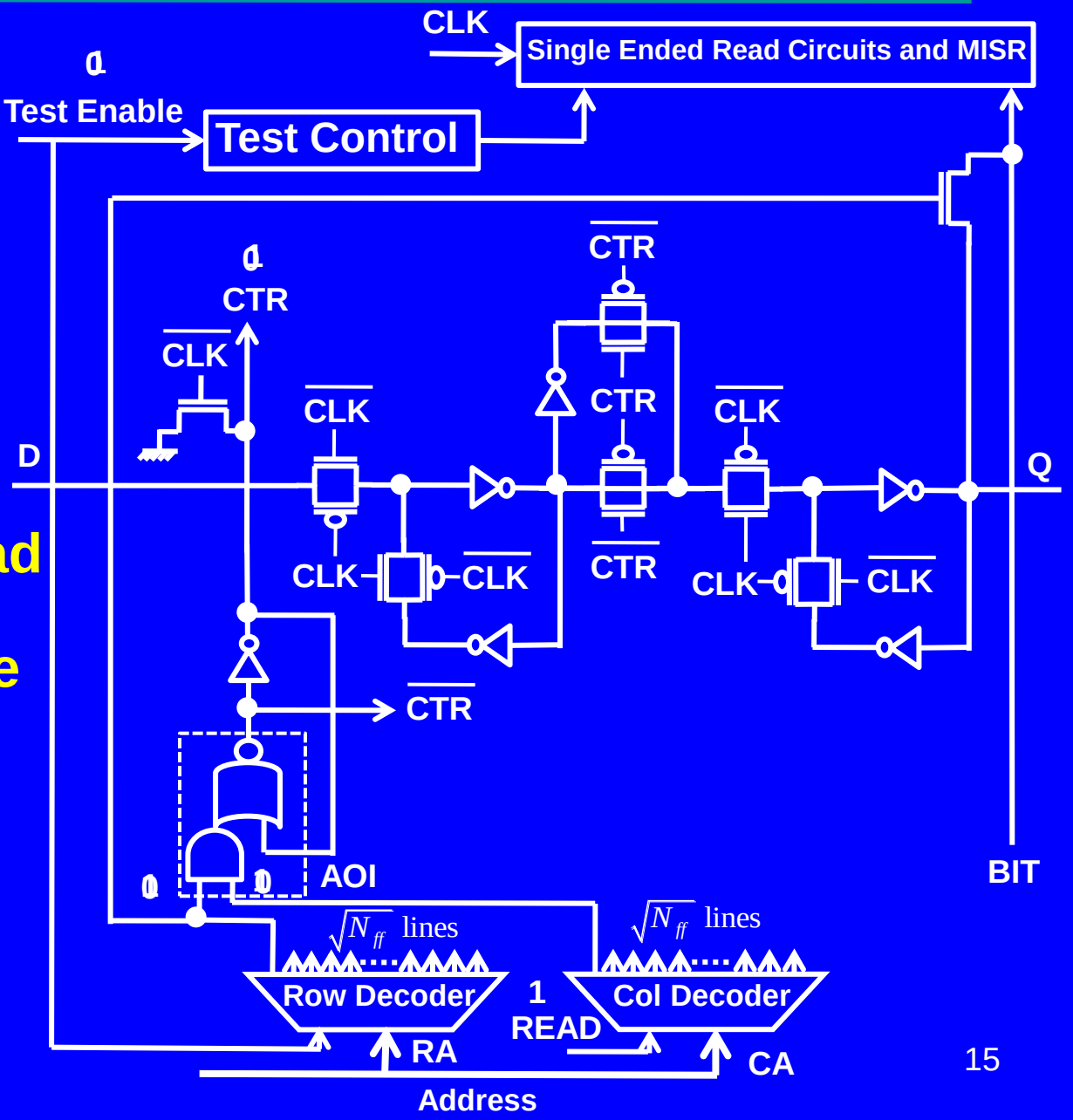
Mudlapur, VDAT'05



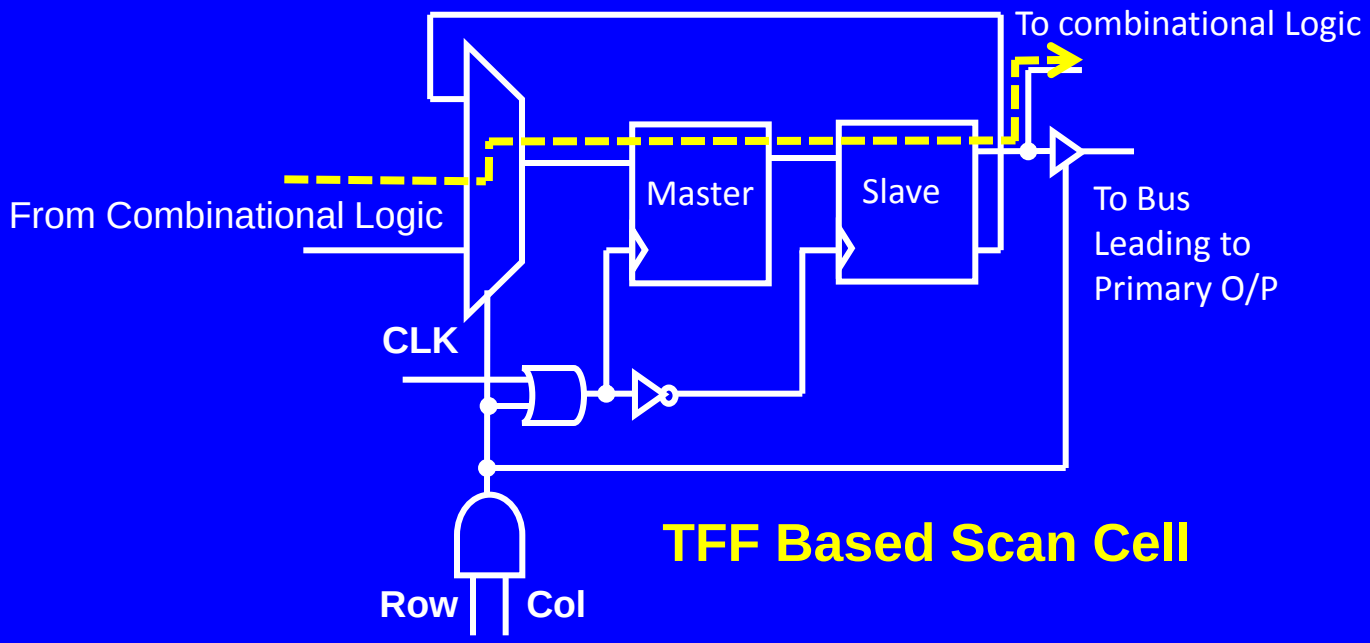
Modified T-Flip-Flip based Cell

Gandhi, ETS'10

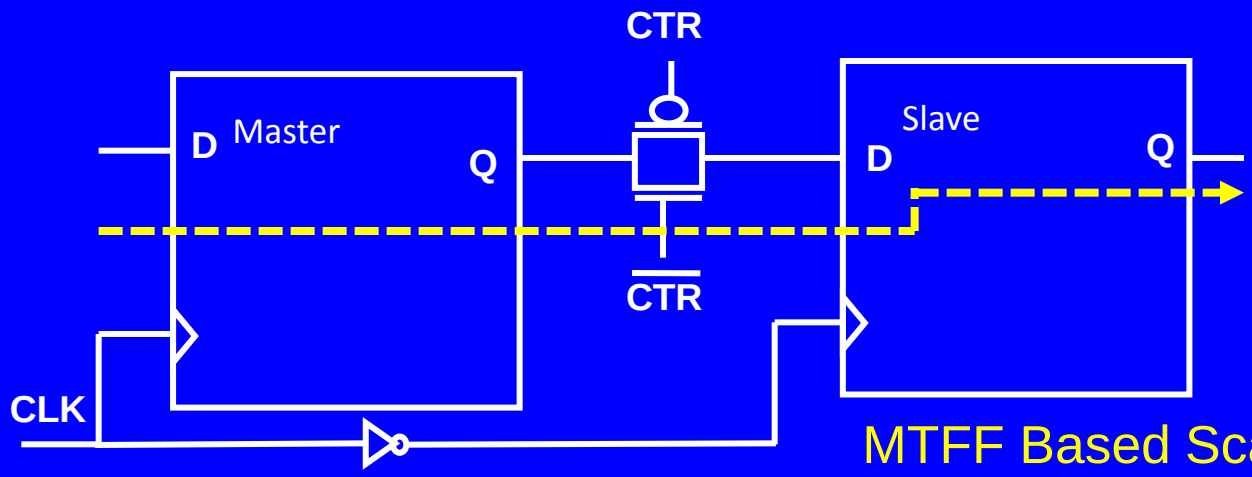
- **Architecture**
- **Functional Mode**
- **Test Application & Simultaneous Read**
- **Reading Response only**



Critical Path Analysis



TFF Based Scan Cell



MTFF Based Scan Cell

Gate Overhead

Circuit	#FF	#Gates	GOV_SS	GOVTFF_RAS	GOVMTFF_RAS
S5378	179	2779	17.47%	25.3%	20.3%
S9234	228	5597	12.04%	17.44%	13.99%
S13207	669	7951	21.36%	30.75%	24.6%
S15850	597	9772	16.7%	24.15%	19.36%
S35932	1728	16065	25.36%	36.4%	29.1%
S38417	1636	22179	19.3%	27.8%	22.2%
S38584	1452	19253	19.7%	28.3%	22.6%

Efficient Decoder Design

Abhishek, ISCAS'10

- Optimize the column decoder
- Uses concept of basis vectors in linear algebra
- Minimizes the number of linear combinations of basis vectors to generate the transition vectors
- Directly related to the no. of clock cycles
- 2-3 times speed up compare to standard RAS

Thank You