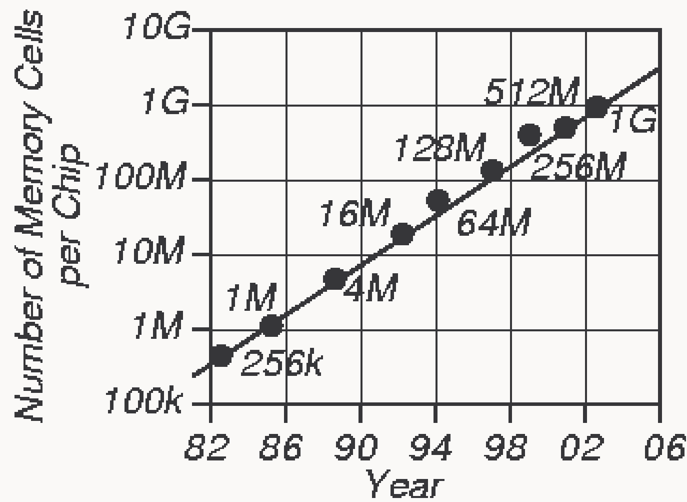
DIGITAL SYSTEM TESTING
COE -545

Lecture – 20

Memory Testing

Dr. Alaaeldin Amin
COE , KFUPM

Memory Cells Per Chip



Test Time in Seconds (Memory Size *n Bits*)

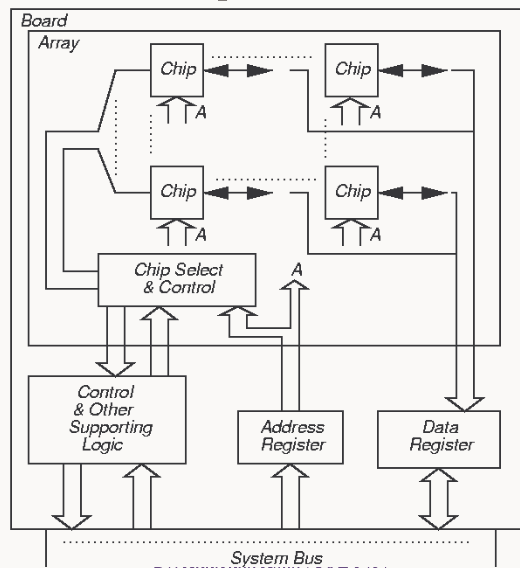
Size	Number of Test Algorithm Operations			
n	n	$n \times \log_2 n$	$n^{3/2}$	n^2
1 Mb	0.06	1.26	64.5	18.3 hr
4 Mb	0.25	5.54	515.4	293.2 hr
16 Mb	1.01	24.16	1.2 hr	4691.3 hr
64 Mb	4.03	104.7	9.2 hr	75060.0 hr
256 Mb	16.11	451.0	73.3 hr	1200959.9 hr
1 Gb	64.43	1932.8	586.4 hr	19215358.4 hr
2 Gb	128.9	3994.4	1658.6 hr	76861433.7 hr

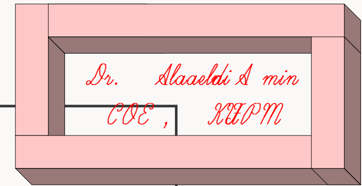
Memory cycle time = 60ns

Dr. Alaaeldin Amin
COE , KFUPM

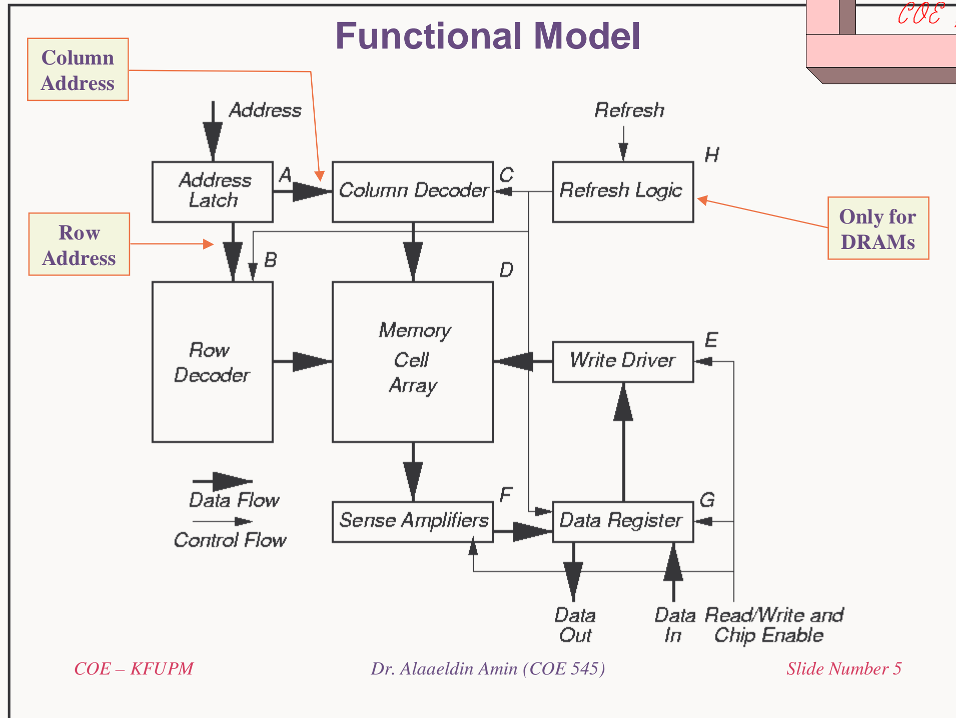
Memory Test Levels

Chip, Array, & Board



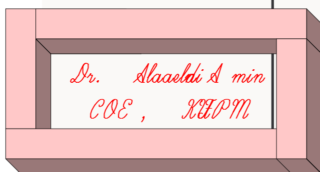


Functional Model

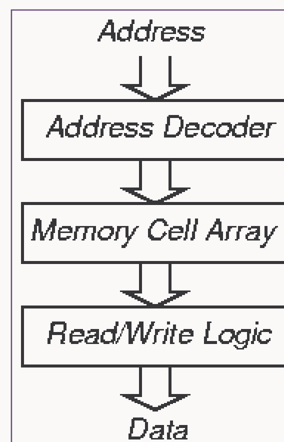


COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 5

Simplified Functional Model



- **Need to Consider Faults in these 3 Blocks Only**
- **Faults in the Read/Write Logic will be Manifested as Array Faults**
- **Need to Consider Faults in 2 Blocks Only**
  - **Memory Array**
  - **Address Decoder**
- **Need to Adopt Suitable Fault Models**



COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 6

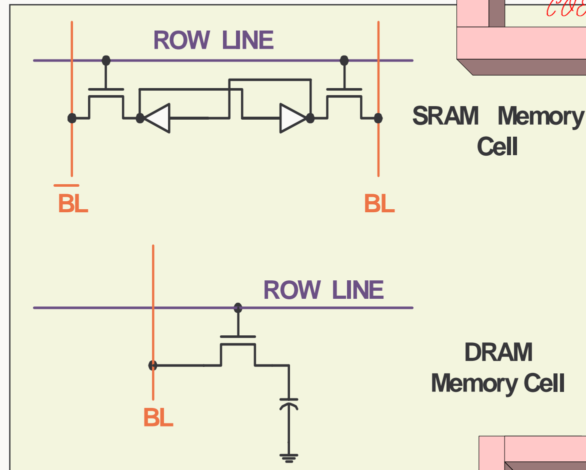
Memory Cells

SRAM Cell

- Cross Coupled Inverters with 2 Access Transistors (6T-Cell)
- One Row (Word) Line and 2 Bit (Column) Lines
- Static Storage of Data → No Need For Refresh

DRAM Cell

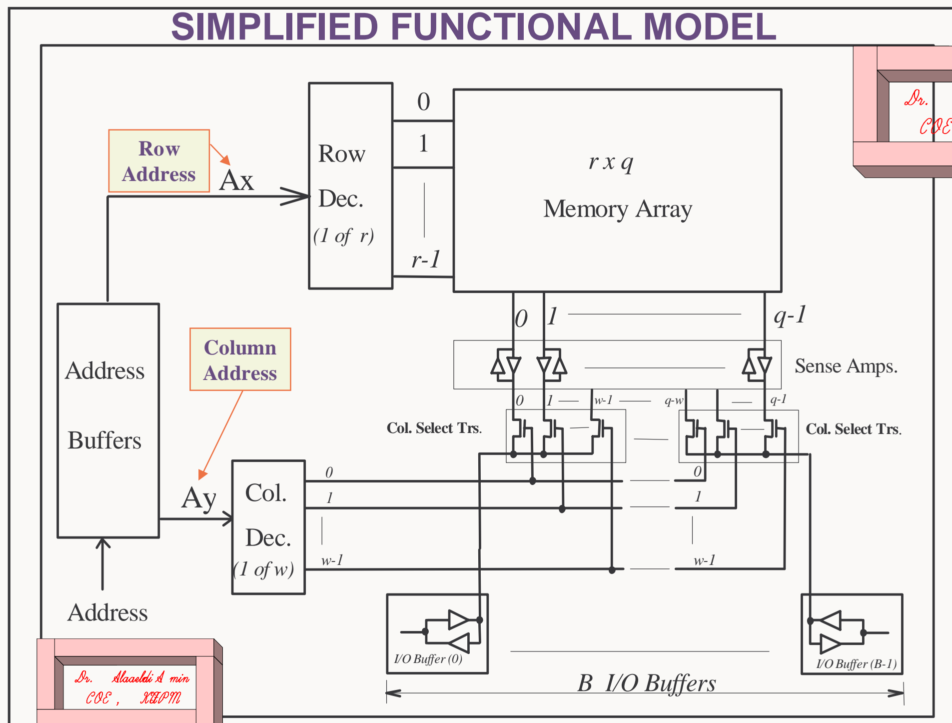
- A Capacitor and An Access Transistor
- One Row (Word) Line and 1 Bit (Column) Lines
- Dynamic Storage of Data (Charge on a Capacitor) → Stored Data Refresh Required



Dr. Alaaeldin Amin (COE 545)   Slide Number 7

Dr. Alaaeldin Amin COE, KFUPM

SIMPLIFIED FUNCTIONAL MODEL



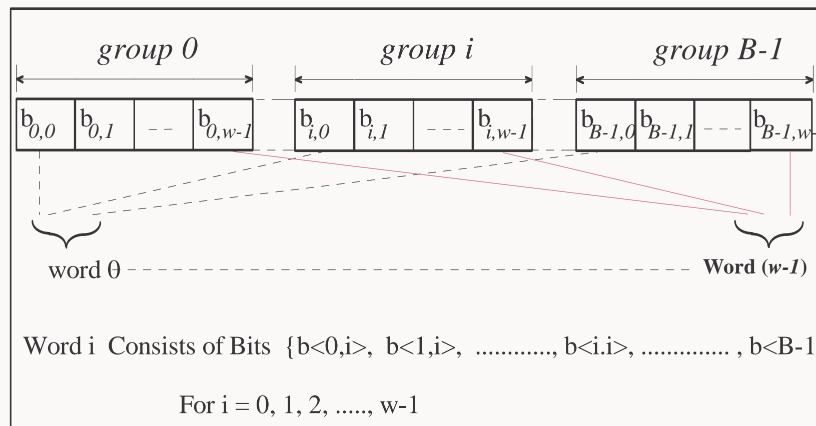
Dr. Alaaeldin Amin COE, KFUPM

SIMPLIFIED FUNCTIONAL MODEL

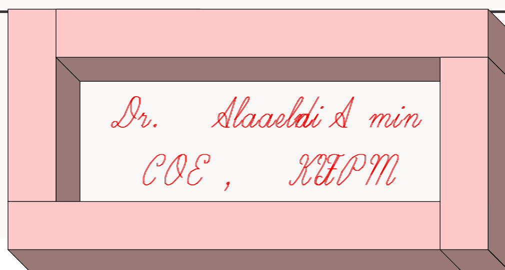
- Memory Array *r* Rows & *q* Columns
- Addresses ⇒ Row Address (Ax) + Column Address (Ay)
- Ax Selects A Single ROW in the Array (Row Decoder)
- Data OF All Cells On the Selected Row Are Passed to The Array Columns Which Are Sensed By Sense Amplifiers.
- Only Data of the Selected Word Are Passed to The Output Buffers By the Column-Select Transistor Which Are Accessed By The Column Address Ay (Through Column Decoder)
- Each Row contains ***w*** words of ***B*** bits each ($q = w$ x B)
- Each Row is Partitioned into B ***groups*** (0, 1, 2, ... , B-1)
- Each Group Consists of ***w*** Memory Cells.
- Each word on a row contributes one cell to each *Group* (*Group i* Consists of Memory Cells Corresponding to the *ith* Bit of ALL the *w* Words on this Row).
- The *qxr* array consists of *r* rows of *q* cells each.

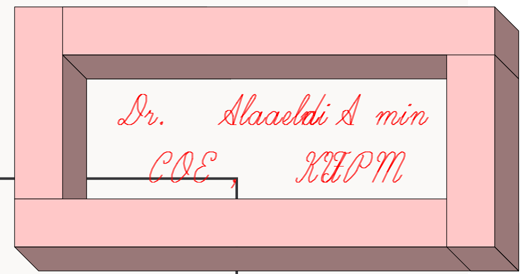
COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 9

SIMPLIFIED FUNCTIONAL MODEL



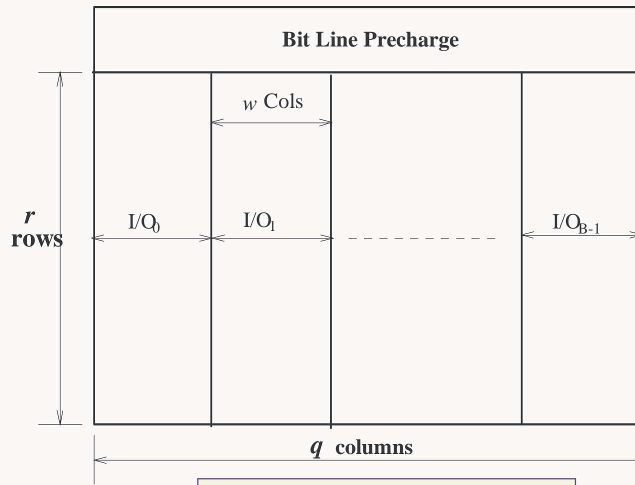
- Each Row contains ***w*** words of ***B*** bits each ($q = w$ x B)
- Each Row is Partitioned into B ***groups*** (0, 1, 2, ... , B-1)
- Each Group Consists of ***w*** Memory Cells.
- Each word on a row contributes one cell to each *Group* (*Group i* Consists of Memory Cells Corresponding to the *ith* Bit of ALL the *w* Words on this Row).
- The *qxr* array consists of *r* rows of *q* cells each.





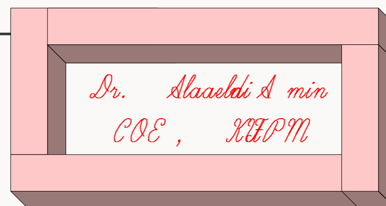
Memory Array Architecture

- The Memory Array Is modeled as a set of B blocks,
- Each Block Interfaces One I/O buffer.
- Each I/O Block Consists of *w* Columns
- Each Column Has *r* Memory Cells
- The *i*th I/O Block Consists of All the *i*th *Groups* on All *r* rows.



General Memory Array Layout

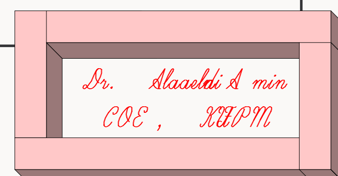
COE – KFUPM | Dr. Alaaeldin Amin (COE 545) | Slide Number 11



Memory Array Fault Models

- **Fault Models Have to Be *Realistic* Representation of Physical Defects or Fault Effects**
- **The Fault Model Should Be *Efficient*:**
  - **Covers Large Class of Physical Failures**
  - **Has Low Complexity**
- **Memory Fault Models Are Classified Into 3 Categories**
  - **Single Cell Fault Model (Stuck & Transition Faults)**
  - **Two-Cell Fault Model (Coupling Faults)**
  - **K-Cell Fault Model**
    - **K-Cells Clustered in a Physical Neighborhood (Pattern Sensitive Faults)**
    - **K-Cells May Be Located Anywhere (K-Coupling, Bridging & State Coupling Faults).**

COE – KFUPM | Dr. Alaaeldin Amin (COE 545) | Slide Number 12

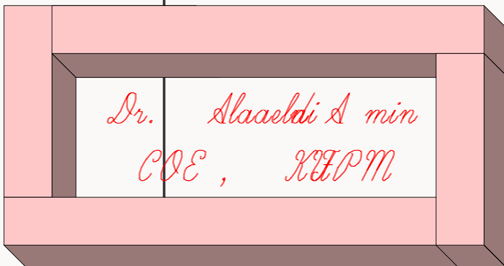


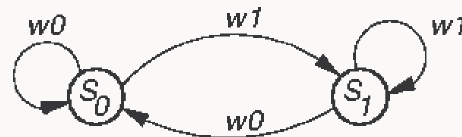
Stuck-at Faults

- ***Condition***: **For each cell, must read a 0 and a 1.**

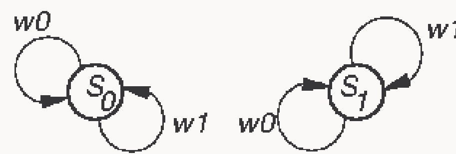
Notation

- **SA0 ⇒ < ∀ / 0>**
- **SA1 ⇒ < ∀ / 1>**





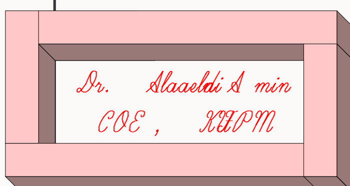
(a) State diagram of a good cell.

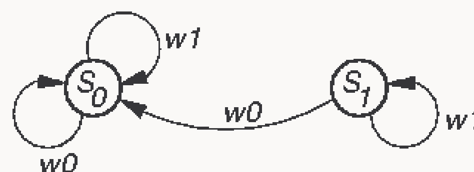


(b) SA0 fault. (c) SA1 fault.

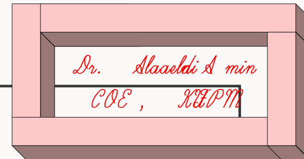
Transition Faults (TF)

- **Considered As a Special Case of *Stuck-At Faults.***
- **Cell Fails to Perform a 0 → 1 Transition (UP-TF)** ↑
- **Notation ⇒ < ↑ / 0>.**
- **Cell Fails to Perform a 1 → 0 Transition (Down-TF)** ↓
- **Notation ⇒ < ↓ / 1>**





< ↑ / 0> *transition fault*

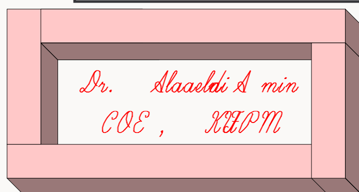


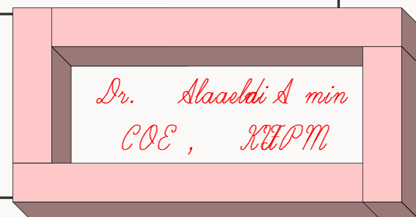
Coupling Faults

Coupling Fault (CF): Transition in bit *j* causes unwanted change in bit *i*

- ***2-Coupling Fault*: Involves 2 cells, special case of *k-Coupling Fault***
  - **Must restrict *k* cells to be practical**
- ***Inversion* and *Idempotent* CFs -- special cases of *2-Coupling Faults***
- ***Inversion Coupling Fault* (CFin) A Transition Write into the Coupling Cell INVERTS Contents of The Coupled Cell.**
  - ***Notation*** $\Rightarrow < \uparrow ; \updownarrow >$ ***OR*** $< \downarrow ; \updownarrow >$
- **Idempotent Coupling Fault (CFid) A Transition Write into the Coupling Cell *Forces* Contents of The Coupled Cell A Prticular State**
  - ***Notation*** $\Rightarrow < \uparrow; 1>, < \uparrow; 0>$ ***OR*** $< \downarrow; 1>, < \downarrow; 0>$

***K- Coupling Fault*: Same As 2-Coupling Cells BUT, Fault Occurs ONLY IFF Remaining (K-2) Cells Have a Particular State (Pattern)**





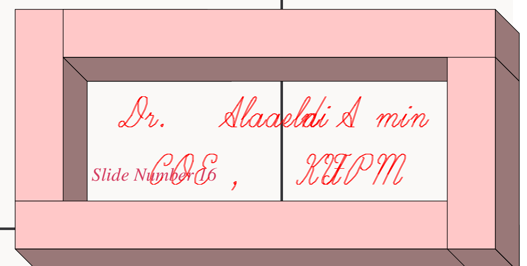
Bridging Faults

Bridging Faults:

- ***Bridging Fault (BF) is a SHORT Between Two Lines, e.g, Row-Lines, Bit-Lines, etc.)***
- ***Two Types of BFs (AND-Bridging) and (OR-Bridging)***
- ***AND-Bridging* Notation** $\Rightarrow$ **< 0, 0 / 0 , 0> , < 0, 1 / 0 , 0> < 1, 0 / 0 , 0> < 1, 1 / 1 , 1>**
- ***OR-Bridging* Notation** $\Rightarrow$ **< 0, 0 / 0 , 0> , < 0, 1 / 1 , 1> < 1, 0 / 1 , 1> < 1, 1 / 1 , 1>**

***State- Coupling Faults*:**

- ***Coupling Is NOT Due to a Transition Write But Rather Depends On the State of the Coupling Cell.***
- ***Notation*** $\Rightarrow$ ***< 0 ; 0> , < 0 ; 1> < 1 ; 0>< 1 ; 1>***



Pattern Sensitive Faults (PSF)

- *A Memory Cell has a PSF iff its Content Is Affected By Contents of Other Memory Cells in its Physical Neighborhood.*
- *Affected Cell is Referred To as the Base Cell*
- The Cells Affecting the Base Cell → *Deleted Neighborhood*
- *The Base Cell and its Deleted Neighborhood is Referred to as the NEIGHBORHOOD*

Classes OF Neighborhoods of Size n :

- *Unrestricted PSF Neighborhood ≡ K-Coupling (K = n) ⇒ Impractical Tests O(n2^n).*
- *Restricted Neighborhood (Neighborhood Pattern Sensitive Faults, or NPSF)*
  - *Restrictions on Neighborhood Size, (5-Cell or 9-Cell)*
  - *Restrictions on The physical Position of the Deleted Neighborhood Cells wrt the Base Cell*

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 17

Dr. Alaaeldi Amin
COE , KFUPM

Pattern Sensitive Faults (PSF)

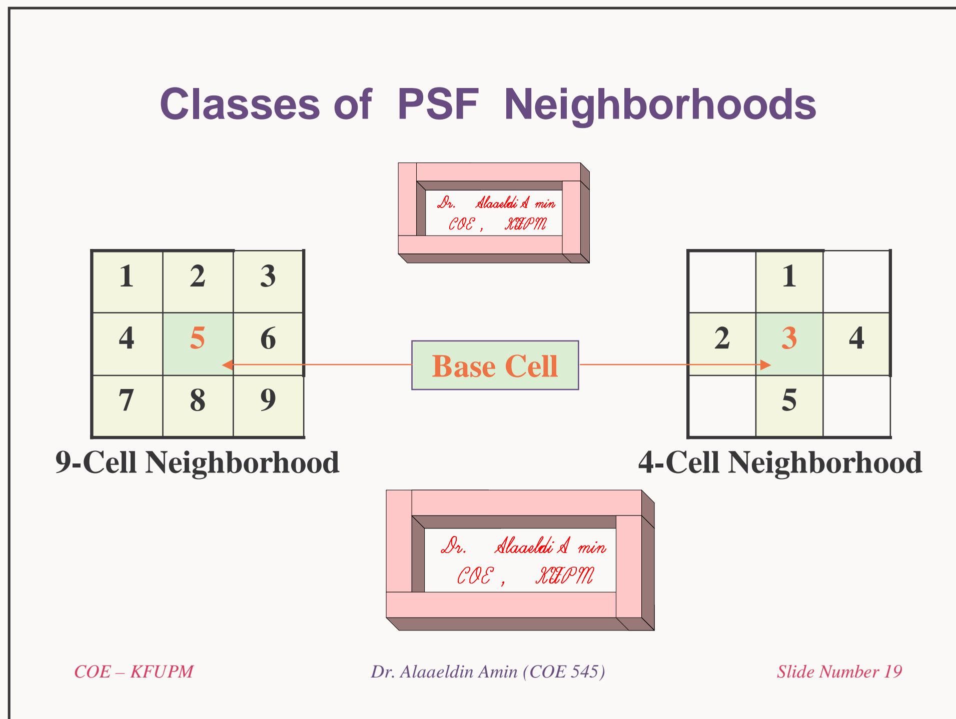
- *A Memory Cell has a PSF iff its Content Is Affected By Contents of Other Memory Cells in its Physical Neighborhood.*
- *Affected Cell is Referred To as the Base Cell*
- The Cells Affecting the Base Cell → *Deleted Neighborhood*
- *The Base Cell and its Deleted Neighborhood is Referred to as the NEIGHBORHOOD*

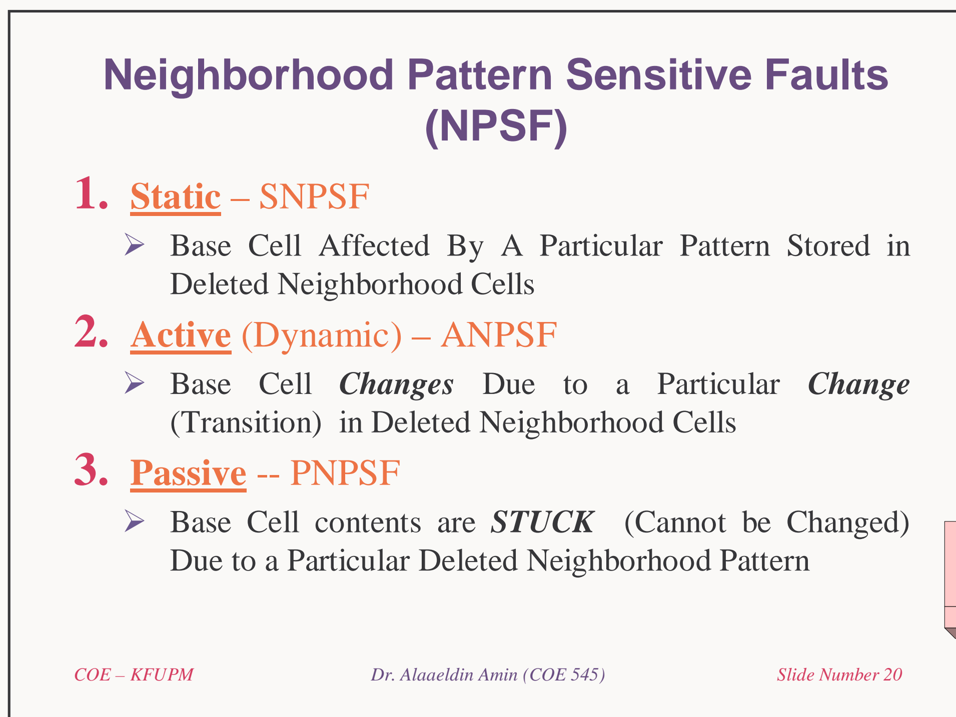
Classes OF Neighborhoods of Size n :

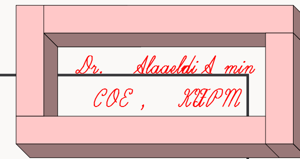
- *Unrestricted PSF Neighborhood ≡ K-Coupling (K = n) ⇒ Impractical Tests O(n2^n).*
- *Restricted Neighborhood (Neighborhood Pattern Sensitive Faults, or NPSF)*
  - *Restrictions on Neighborhood Size, (5-Cell or 9-Cell)*
  - *Restrictions on The physical Position of the Deleted Neighborhood Cells wrt the Base Cell*

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 18

Dr. Alaaeldi Amin
COE , KFUPM


Classes of PSF Neighborhoods
Dr. Alaaeldin Amin
COE, KFUPM
1 2 3
4 5 6
7 8 9
Base Cell
1
2 3 4
5
9-Cell Neighborhood
4-Cell Neighborhood
Dr. Alaaeldin Amin
COE, KFUPM
COE – KFUPM
Dr. Alaaeldin Amin (COE 545)
Slide Number 19


Neighborhood Pattern Sensitive Faults (NPSF)
1. Static – SNPSF
Base Cell Affected By A Particular Pattern Stored in Deleted Neighborhood Cells
2. Active (Dynamic) – ANPSF
Base Cell Changes Due to a Particular Change (Transition) in Deleted Neighborhood Cells
3. Passive -- PNPSF
Base Cell contents are STUCK (Cannot be Changed) Due to a Particular Deleted Neighborhood Pattern
Dr. Alaaeldin Amin
COE, KFUPM
COE – KFUPM
Dr. Alaaeldin Amin (COE 545)
Slide Number 20



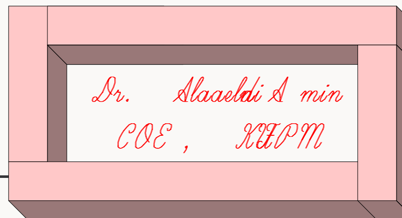
NPSF Justification

- Memory Cells Have Been Shrinking in Size & Dimensions
- Memory Cells Getting Physically Closer ⇒ Electrically Interact With One Another Through:
  - Signal Coupling
  - Leakage From One Cell to Another
- Most Other Fault Classes (e.g. Stuck and Coupling Faults) Can Be Considered as Special Types of PSFs.

Faults in Read / Write Logic

- These Are Mapped into Faults in the Memory Array
- **Example** A Stuck-Fault in R/W Logic Shows As SAF in Large Clusters of Array Cells

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 21



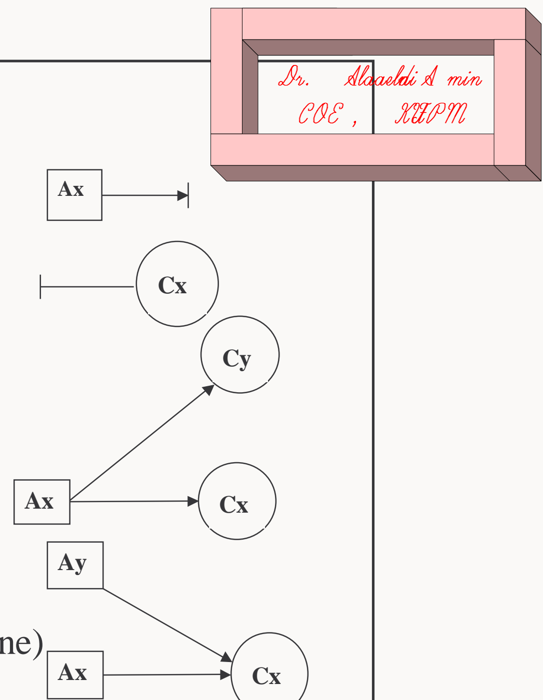
Address Decoder Faults (ADFs)

- **Address decoding Fault assumptions:**
  - **Decoder does not become sequential**
  - **Same behavior during both read & write**
- **Multiple ADFs must be tested for**
- **CMOS stuck-open faults**

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 22

Address Decoder Faults (Functional Fault Model)

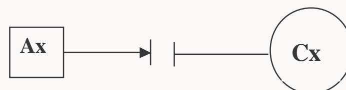
- Address Accessing No Cell
- Inaccessible Cell
- One Address Accessing Many Cells (One-To-Many)
- More Than One Address Accessing One Cell (Many -To- One)

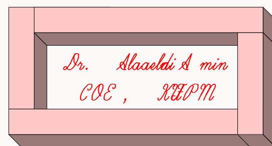


COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 23

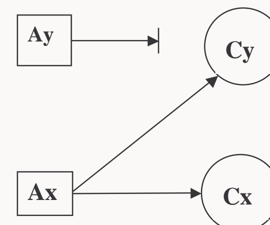
Address Decoder Fault Combinations

- Some Address (Ax) Does Not Access its Own Cell (Cx)





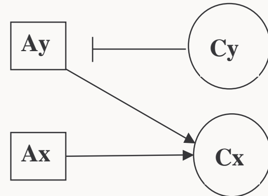
- One-To-Many && Singular Address



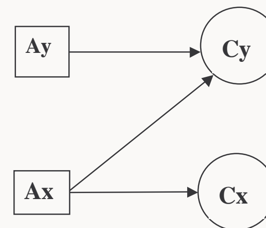
COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 24

Address Decoder Fault Combinations

- Many-To-One && Inaccessible Cell



- One-To-Many && Many-To-One



- CMOS Stuck_Open Faults

COE – KFUPM   Dr. Alaaeldin Amin (COE 545)   Slide Number 25

Test Algorithms

1. AD-HOC Algorithms (Traditional Tests), e.g. M-Scan, Checker-Board, GALPAT, etc.
2. Fault Model Based Algorithms
3. Random Tests

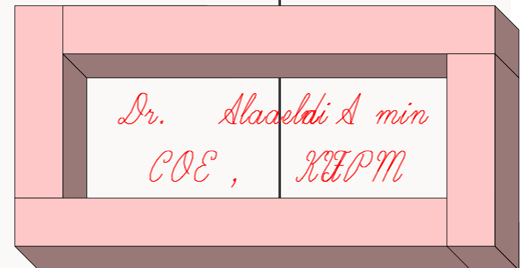
NOTATION:

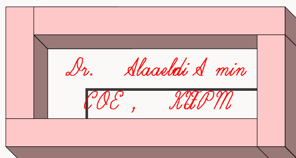
- B The number of bits (cells) in a memory word
- N The Number of Address bits (total of 2N addresses)
- n The total number of bits (cells) in the memory
- K The size of the neighborhood
- A An address
- C A cell
- M A set of cells, words or addresses
- R A read operation
- W A write operation

COE – KFUPM   Dr. Alaaeldin Amin (COE 545)   Slide Number 26

March Test Notation

- r0 -- Read a 0 from a memory location
- r1 -- Read a 1 from a memory location
- w0 -- Write a 0 to a memory location
- w1 -- Write a 1 to a memory location
- ↑ -- Write a 1 to a cell containing 0
- ↓ -- Write a 0 to a cell containing 1
- ↕ -- Complement the cell contents
- ⇑ -- Increasing memory addressing
- ⇓ -- Decreasing memory addressing





Traditional Tests

1- MSCAN Test

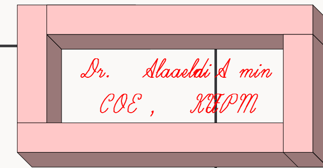
- Writing 0s and 1s to the memory.

For i=0 to n-1
{ W(i) = 0
R(i) (=0)
W(i) = 1
R(i) (=1)
}

{⇑ (W0, R0, W1, R1) }

Evaluation

- Low Fault Coverage
- No Decoder Faults detected (Pass if all Addresses Map to a Single Cell)
- O(n) Complexity



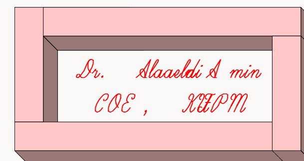
Checkerboard Tests

Step 1.
W(i, j) = 0 ∀ i+j =even
W(i, j) = 1 ∀ i+j =odd
Step 2.
R(i, j) = 0 ∀ i+j =even
R(i, j) = 1 ∀ i+j =odd
Step 3.
Repeat steps 1 & 2
Interchanging 0's and 1's.

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

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

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 29



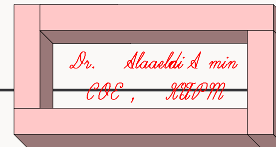
Checkerboard Tests

Step 1.
W(i, j) = 0 ∀ i+j =even
W(i, j) = 1 ∀ i+j =odd
Step 2.
R(i, j) = 0 ∀ i+j =even
R(i, j) = 1 ∀ i+j =odd
Step 3.
Repeat steps 1 & 2
Interchanging 0's and 1's.

Evaluation
- Low Deterministic Fault Coverage
- No Decoder Faults detected (Pass if all Addresses Map to a 4 Cells)
- O(n) Complexity

0	1
1	0

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 30



Marching Test

Step 1. *Initialize*
W(i) = 0 ∀ i=0, n-1

Step 2. For *i=0 to n-1*
{ R(i) (=0);
W(i) =1;
R(i) (=1) }

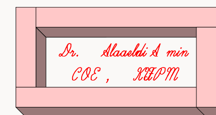
Step 3. For *i=n-1 to 0*
{ R(i) (=1);
W(i) =0;
R(i) (=0) }

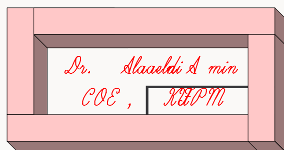
Step 4. Repeat steps 1, 2, 3 *Interchanging 0's and 1's.*

{⇑ (W0) ⇑ (R0, W1, R1) ⇓ (R1, W0, R0)}

Evaluation
- Simple
- Moderate Fault Coverage
- Detects *All STUCK* Faults and *Decoder Faults*
- Does not detect all single coupling faults
- O(n) Complexity

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 31





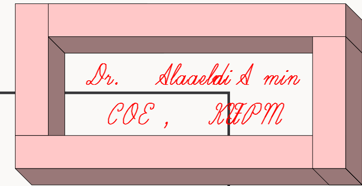
Theorem: Detection of Address Decoder Faults

- *A March test satisfying conditions 1 & 2 detects all address decoder faults.*

Condition	March element
1	⇑ (rx, ..., w $\bar{x}$)
2	⇓ (r $\bar{x}$, ..., wx)

- ... Means any # of read or write operations
- *wx* element Must Precede condition 1
  - $x \in \{0, 1\}$

COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 32



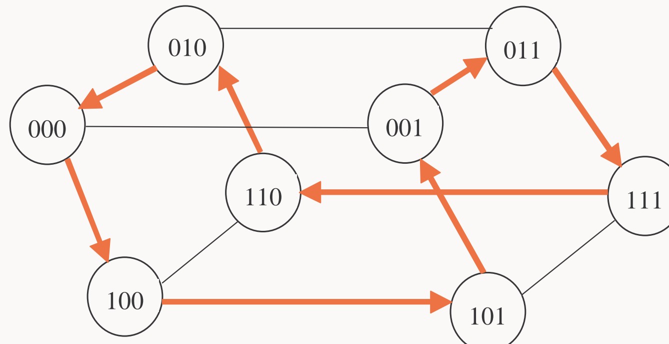
Testing For SNPSFs

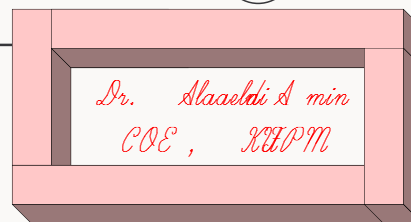
Fault Detection Condition

- Each Base Cell Must be Read in State 0 and State 1 for All Possible Patterns of its Deleted Neighborhood
- For a Neighborhood of Size k, There Are 2^k Static Neighborhood Patterns.
- To Minimize the Total Number of WRITE Cycless, a ***Hamiltonian Sequence*** is used (2^k Write Operations) {*Remember that only 1Bit Can be Written at a Time*}
- ***Hamiltonian Sequence*** Is a K-Bit Sequence Where Each Pattern Differs by 1-Bit From the Preceding One (Hamming Distance = 1)

Hamiltonian Tour for k=3

Each Vertex Visited once





Testing For Active/Dynamic (ANPSFs)

Fault Detection Condition for ANPSFs

- Each Base Cell Must be Read in State 0 and State 1 for All Possible ***Changes*** in its Deleted Neighborhood.
- For a Neighborhood of Size k, $k2^k$ Patterns are Required {*Every Vertex has k* ***adjacent*** *Vertices*}.

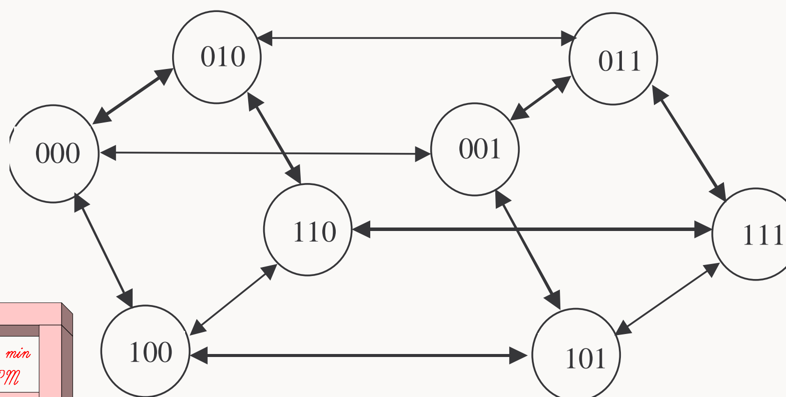
Fault Detection Condition for PNPSFs

- Each Base Cell Must be Written and Read in State 0 and State 1 for All Possible Patterns of its Deleted Neighborhood.
- For a Neighborhood of Size k, There are 2^k Passive Neighborhood Patterns (PNPs).
- To Minimize The Number of Operations, ANPSFs and PNPSFs, Can be Tested Together Using an ***Eulerian Sequence*** (**$k2^k$ Patterns**).

Testing For Active/Dynamic (ANPSFs)

- A k-bit *Eulerian* Graph is Constructed with a Vertex for each k-Bit Pattern.
- Two Vertices are Connected iff their corresponding patterns differ by exactly one bit.
- When Two Vertices X, Y, are Connected, They are Connected by Exactly Two Directed Arcs
- An *Eulerian* Sequence is a Tour Through an *Eulerian* Graph Such that Each Directed ***Arc*** is Traversed Exactly Once.
- A *Hamilitonian* Sequence is a Tour Through an Eulerian Graph Such That Every ***Vertex*** is Visited Exactly Once.

Testing For Active/Dynamic (ANPSFs)



Dr. Alaaeldi A min
COE , KFUPM

Dr. Alaaeldi A min
COE , KFUPM

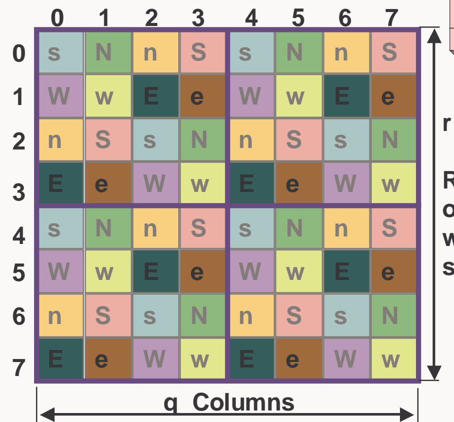
3-Bit Eulerian Graph (k =3)

Every ARC in the Graph is Visited ONCE

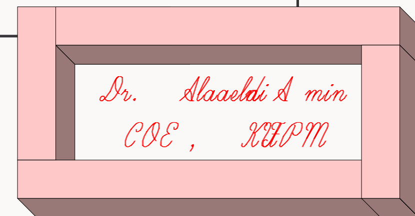
SNPSF Test Algorithm

Memory Array Tiling

1. Initialize The Whole Array to 0's (*n writes*)
2. Consider *n,e,w,s* as Base Cells & *N,E,W,S* as their deleted Neighborhood
3. Write into the Deleted Neighborhood Cells the 16 Possible Pattern Combinations, In the order given by a Hamiltonian Sequence (*Final Pattern Will Be 0000*). ($n/2 * 2^{k-1}$ *Writes*)
4. After Each New Pattern is Written into the Deleted Neighborhood, The State of the Base Cells is Verified by a A READ(0) Operation ($n/2 * 2^k$ *Reads*)
5. Write 1's into the Base Cells *n,e,w,s* (*n/2 writes*)



COE – KFUPM Dr. Alaaeldin Amin (COE 545) Slide Number 37

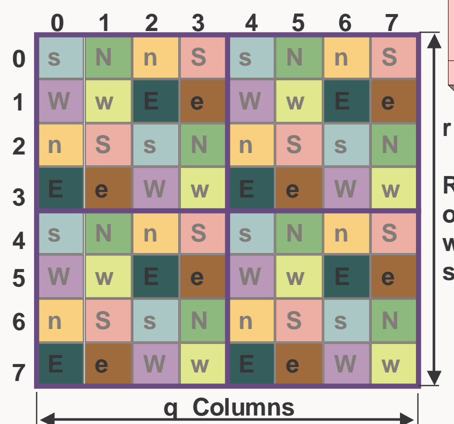


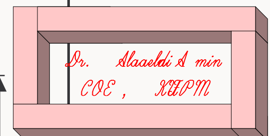
SNPSF Test Algorithm

Memory Array Tiling

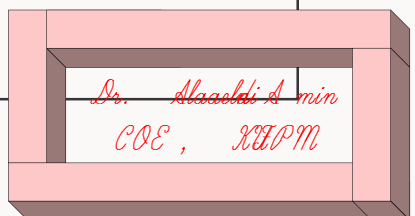
6. Write into the Deleted Neighborhood Cells the 16 Possible Pattern Combinations, In the order given by a Hamiltonian Sequence (*Final Pattern Will Be 0000*). ($n/2 * 2^{k-1}$ *Writes*)
7. After Each New Pattern is Written into the Deleted Neighborhood, The State of the Base Cells is Verified by a A READ(1) Operation ($n/2 * 2^k$ *Reads*)
8. Repeat Steps 2-7 Considering *N,E,W,S* as Base Cells & *n,e,w,s* as their deleted Neighborhood

- The Algorithms is O($2^k n$)
- To test For ANPSF's and PNPSF's, An Eulerian Sequence is Followed instead of the Hamiltonian Sequence. This Makes Algorithms O($k2^k n$)





COE – KFUPM Dr. Alaaeldin Amin (COE 545)



RAMs with Built In Self Test (BIST)

BIST Logic Structure

1- Control Logic

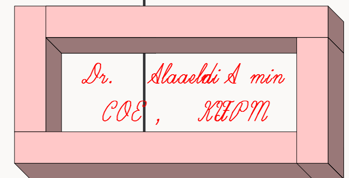
- ❖ Initiates, Stops, and Supervises The Test Sequence (Test Algorithm)

2- Address Generation Logic

- ❖ Generates Required Address Sequence

3- Data-Generation & Response-Verification Logic

4- Test Trigger Logic



BIST RAM Test Architectures

1- Single-Array Single-Bit (SASB)

- ❖ *A Single Bit in A Single Array of the RAM Chip is Tested at a Time*

2- Single-Array Multiple-Bit (SAMB)

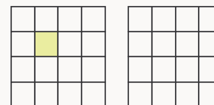
- ❖ *Multiple Bits (e.g. a Full Row) in A Single Array of the RAM Chip are Tested at a Time*

3- Multiple-Array Single-Bit (MASB)

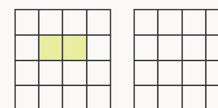
- ❖ *Single Bits in Multiple Arrays of the RAM Chip are Tested at a Time*

4- Multiple-Array Multiple-Bit (MAMB)

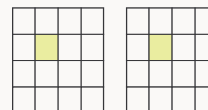
- ❖ *Multiple Bits (e.g. a Full Row) in Multiple Arrays of the RAM Chip are Tested at a Time*



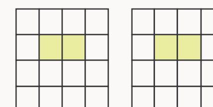
Single Array Single Bit (SASB)



Single Array Multiple Bit (SAMB)



Multiple Array Single Bit (MASB)



Multiple Array Multiple Bit (MAMB)