Chapter 3
RAM Testing

Jin-Fu Li
Advanced Reliable Systems (ARES) Lab.
Dept. of Electrical Engineering
National Central University
Jhongli, Taiwan

Outline

- March Tests
- Typical RAM Faults Testing
- AFs Testing
- NPSFs Testing
- Converting Bit-Oriented RAM Tests into Word-Oriented RAM Tests

March Tests

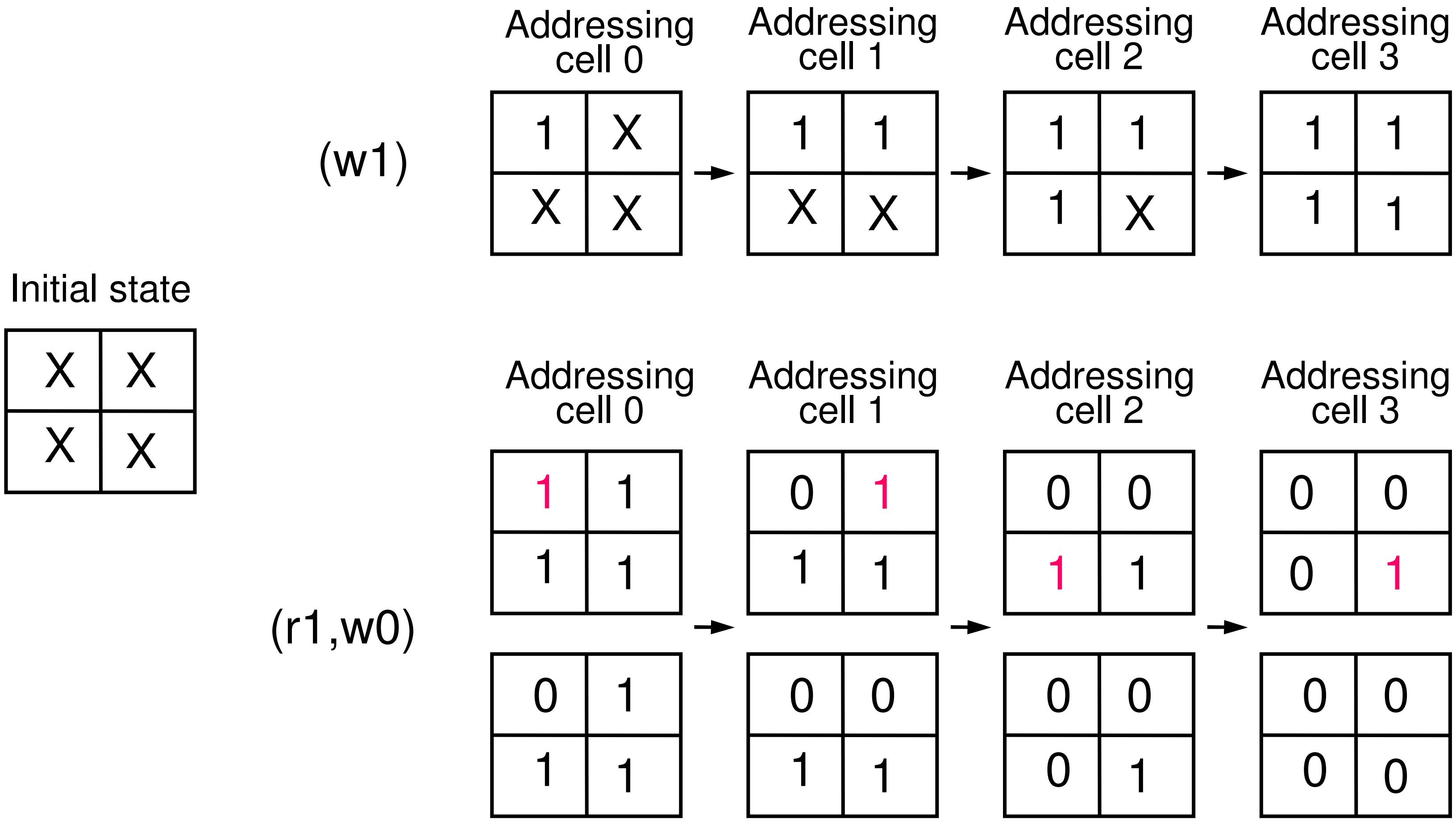
- A *march test* consists of a finite sequence of *march elements*
- A march element
  - A finite sequence of Read and/or Write operations applied to every cell in memory in either increasing address order (cell 0 to cell n-1) or decreasing address order (cell n-1 to cell 0)
- All operations of a march element are done before proceeding to the next address
- The march tests are a preferred method for RAM testing
  - Linear complexity, regularity, and symmetry

March Test Notation

- r*x*: a read *x* operation
- w*x*: a write *x* operation
- ⇑ : increasing addressing sequence (from 0 to n-1)
- ⇓ : decreasing addressing sequence (from n-1 to 0)
- ⇕ : either increasing or decreasing addressing sequence

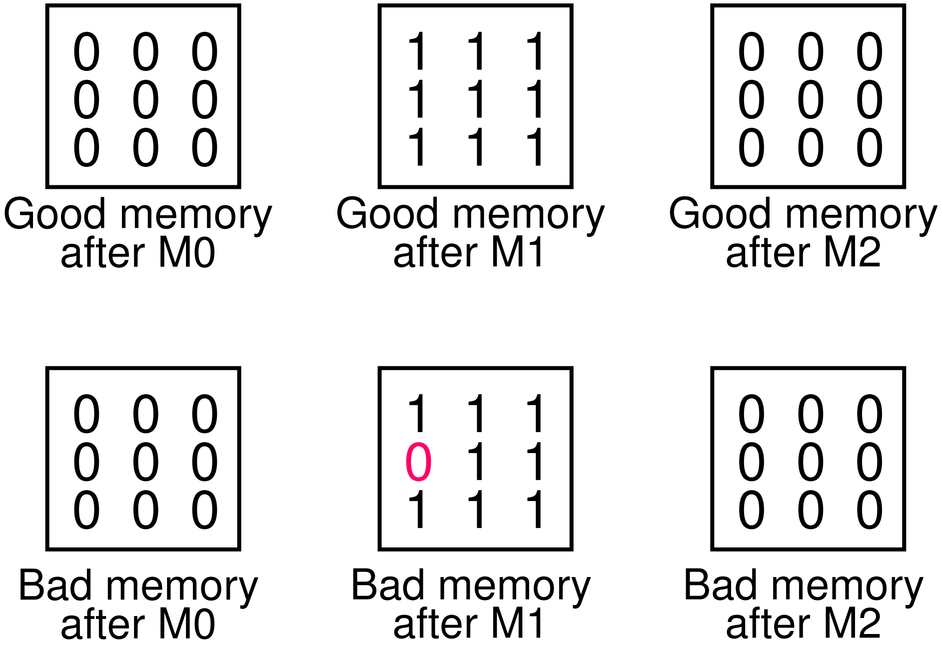
An Example of March Test

- An example of march test $\{\Uparrow(w1);\Uparrow(r1,w0)\}$



March Tests for SAFs & TFs

- MATS+: $\{\Updownarrow(w0);\Uparrow(r0,w1);\Downarrow(r1,w0)\}$
- MATS+ detection of SA0 fault



0 0 0 0 0 0 0 0 0	1 1 1 1 1 1 1 1 1	0 0 0 0 0 0 0 0 0
Good memory after M0	Good memory after M1	Good memory after M2
0 0 0 0 0 0 0 0 0	1 1 1 0 1 1 1 1 1	0 0 0 0 0 0 0 0 0
Bad memory after M0	Bad memory after M1	Bad memory after M2

March Tests for SAFs & TFs

- MATS+ detection of SA1 fault

0	0	0
0	0	0
0	0	0

Good memory
after M0

1	1	1
1	1	1
1	1	1

Good memory
after M1

0	0	0
0	0	0
0	0	0

Good memory
after M2

0	0	0
1	0	0
0	0	0

Bad memory
after M0

1	1	1
1	1	1
1	1	1

Bad memory
after M1

0	0	0
1	0	0
0	0	0

Bad memory
after M2

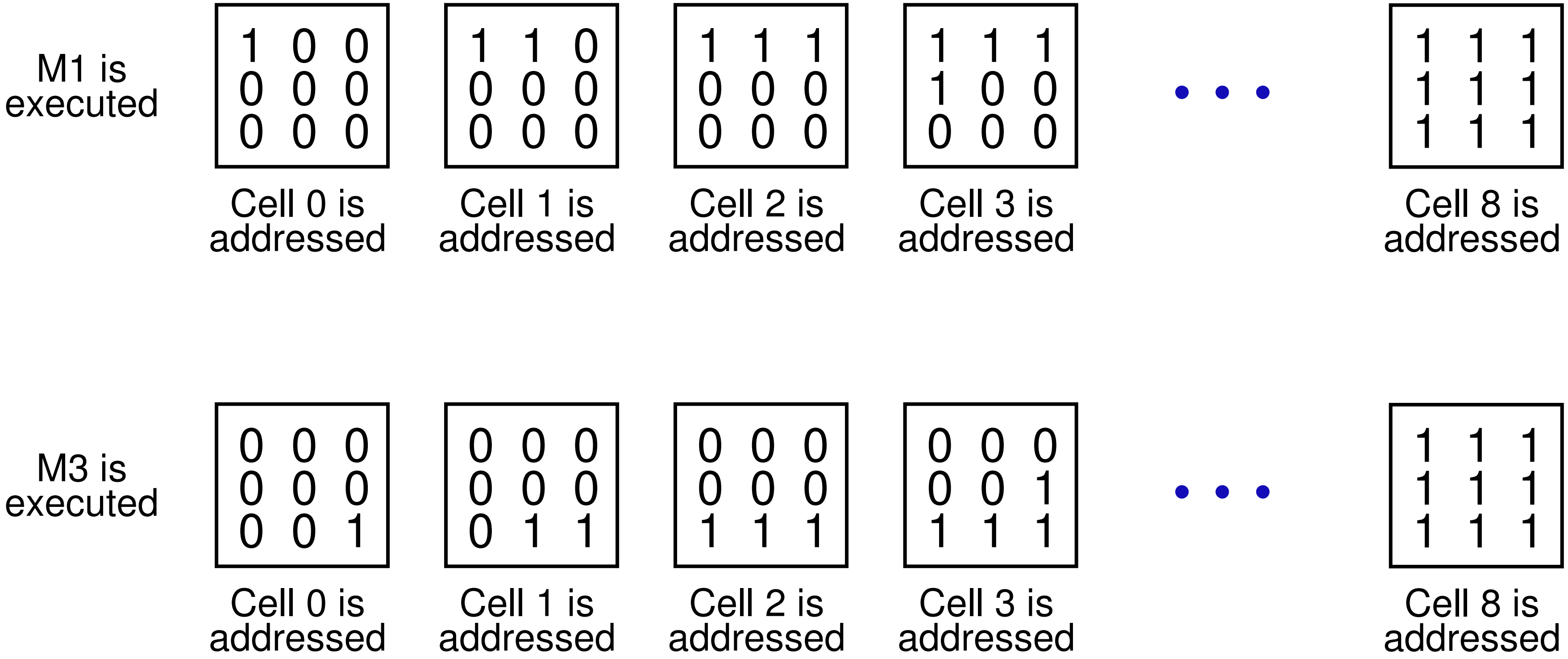
- MATS+ detection of TFu & TFd can be proved in the same way

Tests for Detecting SAFs & TFs

- Conditions for detecting SAFs & TFs
  - SAFs & TFs can be detected by a march test which contains the following two march elements (or single march element containing both elements)
- $(\ldots, w0, r0, \ldots)$ to detect SA1 faults and TFd
- $(\ldots, w1, r1, \ldots)$ to detect SA0 faults and TFu

March Tests for CFs

- March C - : $\{\Updownarrow(w0);\Uparrow(r0,w1);\Uparrow(r1,w0);\Downarrow(r0,w1);\Downarrow(r1,w0);\Updownarrow(r0)\}$
- Detection of CFs



M1 is executed

$\begin{matrix}1&0&0\\0&0&0\\0&0&0\end{matrix}$ Cell 0 is addressed

$\begin{matrix}1&1&0\\0&0&0\\0&0&0\end{matrix}$ Cell 1 is addressed

$\begin{matrix}1&1&1\\0&0&0\\0&0&0\end{matrix}$ Cell 2 is addressed

$\begin{matrix}1&1&1\\1&0&0\\0&0&0\end{matrix}$ Cell 3 is addressed

• • •

$\begin{matrix}1&1&1\\1&1&1\\1&1&1\end{matrix}$ Cell 8 is addressed

M3 is executed

$\begin{matrix}0&0&0\\0&0&0\\0&0&1\end{matrix}$ Cell 0 is addressed

$\begin{matrix}0&0&0\\0&0&0\\0&1&1\end{matrix}$ Cell 1 is addressed

$\begin{matrix}0&0&0\\0&0&0\\1&1&1\end{matrix}$ Cell 2 is addressed

$\begin{matrix}0&0&0\\0&0&1\\1&1&1\end{matrix}$ Cell 3 is addressed

• • •

$\begin{matrix}1&1&1\\1&1&1\\1&1&1\end{matrix}$ Cell 8 is addressed

March Tests for CFs

- Conditions for detecting CFs
  - A march test which contains one of the two pairs of march elements of Case A & Case B can detect simple CFs (CFin, CFst, CFid)
  - Case A
    - 1. $\Uparrow(rx,\cdots,w\bar{x})\ \Uparrow(r\bar{x},\cdots,wx)$
    - 2. $\Downarrow(rx,\cdots,w\bar{x})\ \Downarrow(r\bar{x},\cdots,wx)$
  - Case B
    - 1. $\Uparrow(r\bar{x},\cdots,wx)\ \Uparrow(rx,\cdots,w\bar{x})$
    - 2. $\Downarrow(r\bar{x},\cdots,wx)\ \Downarrow(rx,\cdots,w\bar{x})$
- A.1 (A.2) will sensitize the CFs, and it will detect the fault, when the value of the fault effect is x’ (x), by the rx (rx’) operation of the first (second) march element when the coupled cell has a higher (lower) address than the coupling cell

March Tests for DRFs

- Data retention faults (DRFs)
  - DRF has two subtypes
    - A stored '1' will become a '0' after a time T
    - A stored '0' will become a '1' after a time T
- Conditions for detecting DRFs
  - Any march test can be extended to detect DRFs
  - The detection of each of the two DRF subtypes requires that a memory cell be written into the corresponding logic states
- If we are interested in detecting simple DRFs only
  - The delay elements can be placed between any two pairs of march elements, e.g., $\Uparrow(rx,\cdots,w\bar{x})$; Del; $\Downarrow(r\bar{x},\cdots,wx)$

March Tests for AFs

- Conditions for detecting AFs

Condition	Element
1	$\Uparrow(rx, \cdots, w\bar{x})$
2	$\Downarrow(r\bar{x}, \cdots, wx)$

- Condition 1
  - Read the value x from cell 0, then write x’ to cell 0, …, read the value x from cell n-1, then write x’ to cell n-1
- Condition 2
  - Read the value x’ from cell n-1, then write x to cell n-1, …, read the value x’ from cell 0, then write x to cell 0

March Tests for AFs

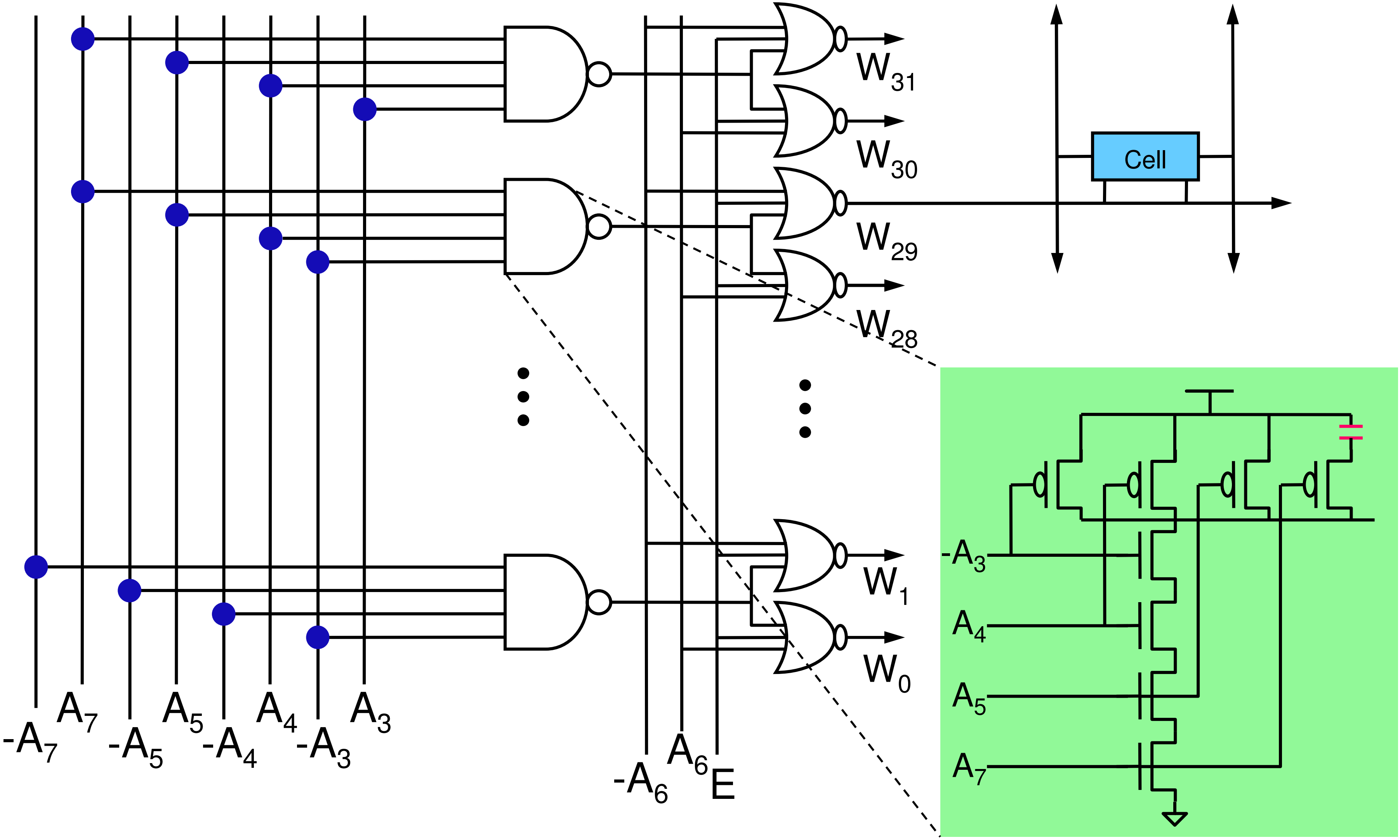
- Sufficiency of the conditions for detecting AFs
  - Fault A & B:
    - Detected by every test that detects SAFs. When address Ax is written and read, C_x will appear either SA0 or SA1.
  - Fault C:
    - Detected by first initializing the entire memory to an expected value x or x'. Any subsequent march element operation that reads the expected value x and ends by writing x' detects fault C
  - Fault D:
    - The memory my return a random result. The fault must be generated when A_x is written, and detected when either A_w and A_v is read
    - Condition 1 detects fault D1 and D2
    - Condition 2 detects fault D1 and D3

March Tests for AFs

- Necessity of the conditions for detecting AFs
  - Remove rx from Condition 1
    - A test can not detect fault A or B for the case they always return x’
  - Remove rx’ from Condition 2
    - A test can not detect fault A or B for the case they always return x
  - Remove rx or wx’ from Condition 1
    - A test can not detect fault D2
  - Remove rx’ or wx from Condition 2
    - A test can not detect fault D3
  - Remove both write operations
    - A test can not detect fault C and fault D1

Tests for Specific AFs

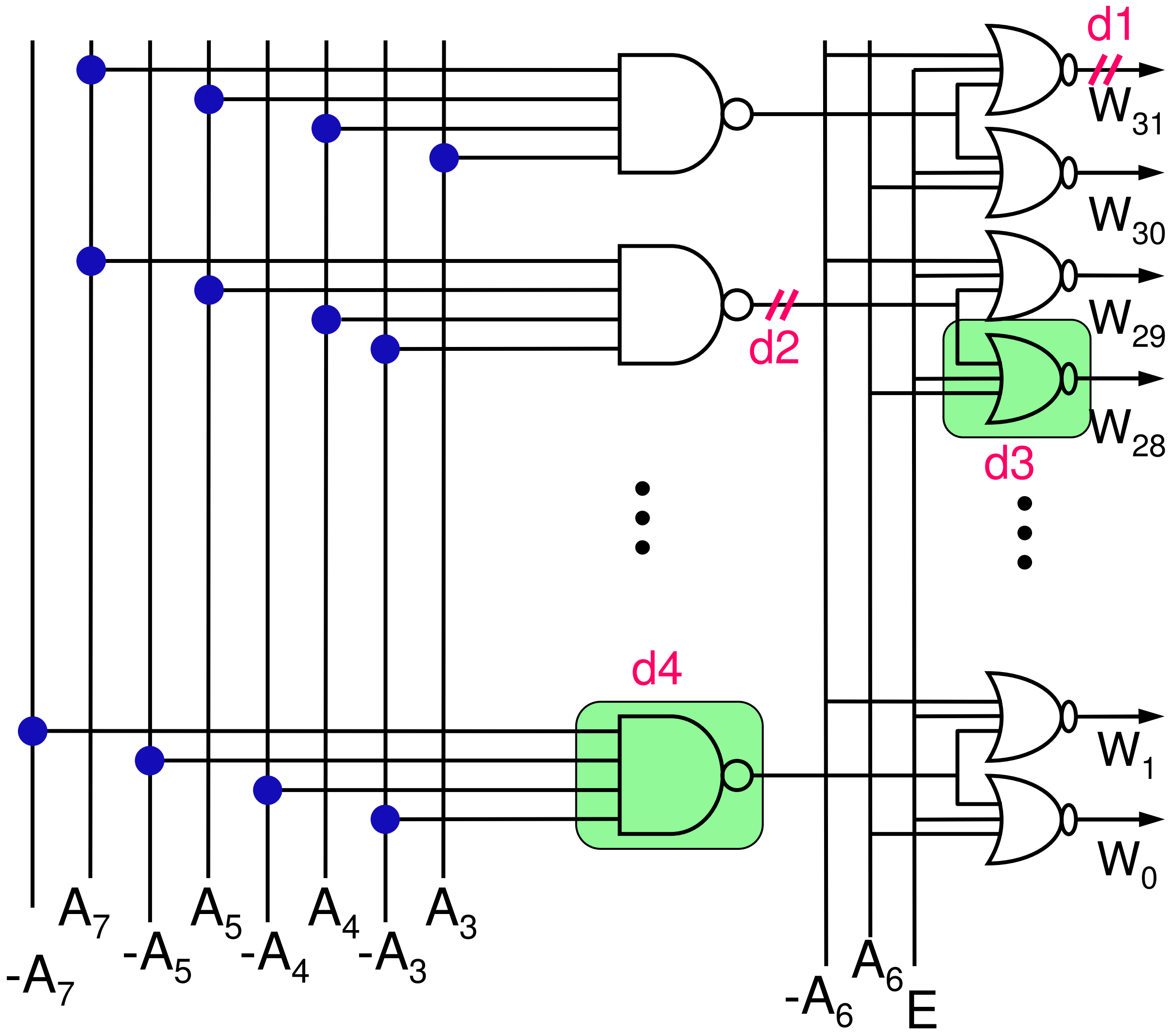
- A open defect in the address decoder



Why Non-Detection by March Tests?

- A 6N march test algorithm $\Updownarrow(w0);\Uparrow(r0,w1);\Downarrow(r1,w0,r0)$
- M1 reads the initialized value and writes logic 1 in each RAM cell in ascending address order
  - Address 10110 -> 10111, which modifies the A_3 bit
  - Hence, word line (10110) is disable like a fault-free case
- Similarly, M2 is executed in descending address order
  - Address 10110->10101, which modifies A4 and A3 bits
  - Word line (10110) is disable again, i.e., the fault is not detected
- Sequential-fault detection
  - Test sequence dependent

Open Defects in an Address Decoder



- Open defects in an address decoder
  - Inter-gate defects
  - Intra-gate defects
- Inter-gate defects
  - E.g., d1 and d2
  - Stuck-at faults
- Intra-gate defects
  - E.g., d3 and d4
  - Sequential faults

Tests for Specific AFs

- For the 4-input NAND gate in the previous slide with defect 4 (shaded) following test sequence can be applied
  - Keep column decoder address constant
  - Keep A_6=0
  - Let $A_7A_5A_4A_3$=0000, Write(1);
  - Let $A_7A_5A_4A_3$=0001, Write(0);
  - Let $A_7A_5A_4A_3$=0000, Read(1);
  - Let $A_7A_5A_4A_3$=0010, Write(0);
  - Let $A_7A_5A_4A_3$=0000, Read(1);
  - Let $A_7A_5A_4A_3$=0100, Write(0);
  - Let $A_7A_5A_4A_3$=0000, Read(1);
  - Let $A_7A_5A_4A_3$=1000, Write(0);
  - Let $A_7A_5A_4A_3$=0000, Read(1);

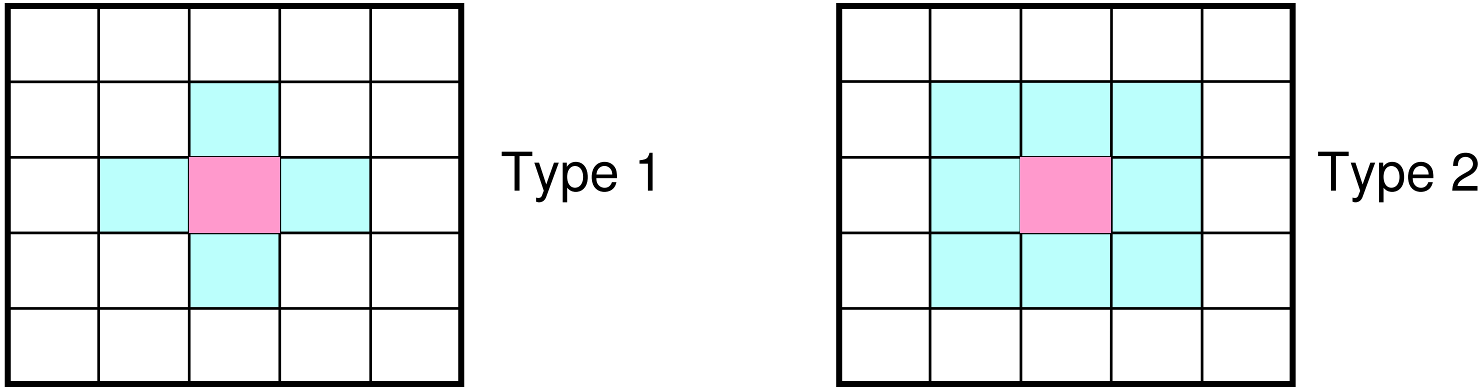
Tests for Specific AFs

- Test algorithm for a M-bit address decoder

```
Column _address=0
For i=0 to $2^{M-1}$ Do
    Base_address=2*i
    Write "0" to Base_address
    For j=0 to M Do
        Write_address=Based_address $XOR_{binary} 2^j$
        Write "1" to Write_address
        Read "0" from Base_address
    End For
End For
```

Tests for NPSFs

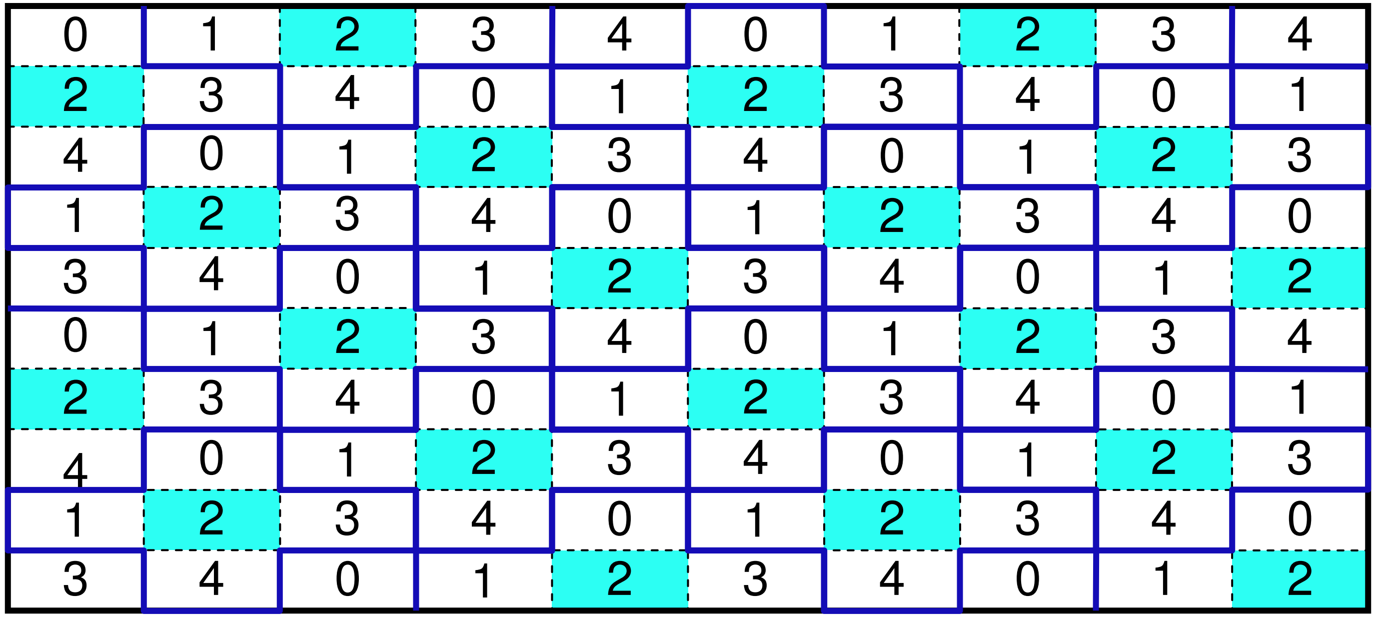
- Neighborhood Pattern Sensitive Faults (NPSFs)
  - Type 1 and type 2 neighborhoods



- The physical layout of the RAM core and the technology determine which cells could affect each other
  - Usually type 1 neighborhood is used because the deleted neighborhood is most likely affects the based cell

Tests for NPSFs

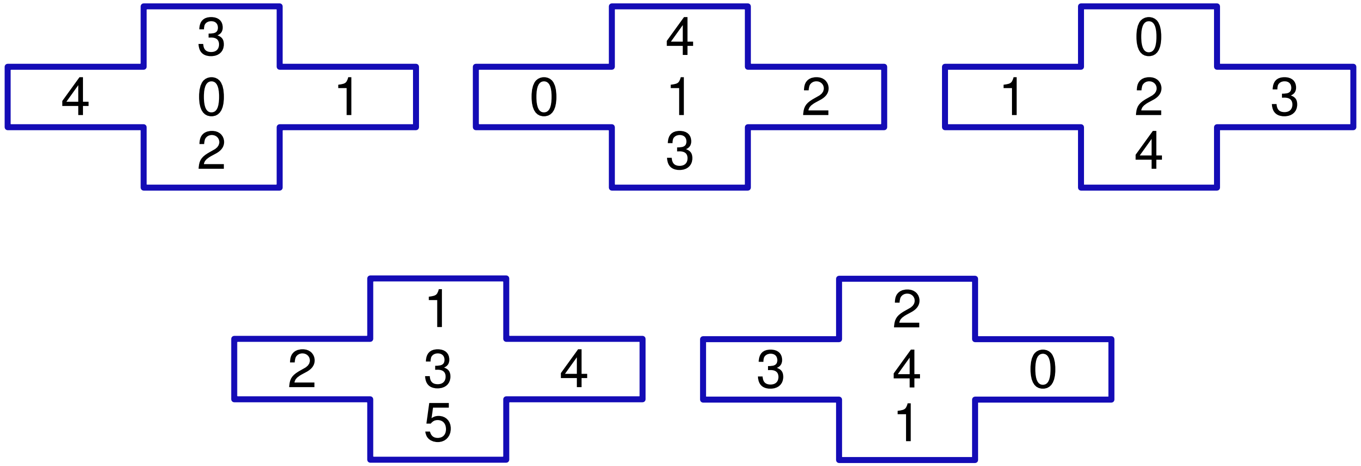
- Type 1 tiling neighborhood
  - The figure shows that a cell-2 as base cell
  - The deleted neighborhood of all base cells-2 is formed by a cell-0, a cell-1, a cell-3, and a cell-4



0	1	2	3	4	0	1	2	3	4
2	3	4	0	1	2	3	4	0	1
4	0	1	2	3	4	0	1	2	3
1	2	3	4	0	1	2	3	4	0
3	4	0	1	2	3	4	0	1	2
0	1	2	3	4	0	1	2	3	4
2	3	4	0	1	2	3	4	0	1
4	0	1	2	3	4	0	1	2	3
1	2	3	4	0	1	2	3	4	0
3	4	0	1	2	3	4	0	1	2

Tests for NPSFs

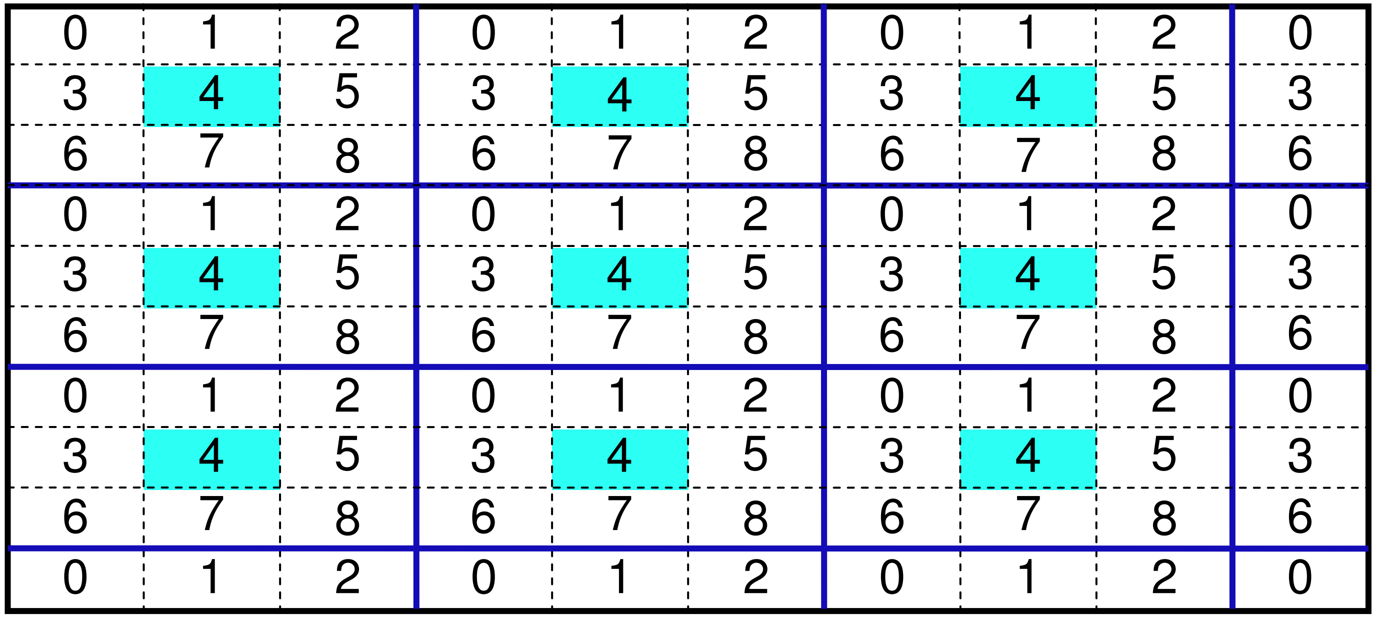
- Five base cells using type 1 tiling neighborhood



- SNPSF test
  - When all static neighborhood patterns are applied simultaneously to the neighborhoods of all based cells-2, they are automatically applied to the neighborhoods of all base cells in the memory
  - With $n/5*2^5$ write operations

Tests for NPSFs

- Type 2 tiling neighborhood
  - Similar to type 1 NPSFs tiling method



0	1	2	0	1	2	0	1	2	0
3	4	5	3	4	5	3	4	5	3
6	7	8	6	7	8	6	7	8	6
0	1	2	0	1	2	0	1	2	0
3	4	5	3	4	5	3	4	5	3
6	7	8	6	7	8	6	7	8	6
0	1	2	0	1	2	0	1	2	0
3	4	5	3	4	5	3	4	5	3
6	7	8	6	7	8	6	7	8	6
0	1	2	0	1	2	0	1	2	0

Tests for NPSFs

- Two-group method for type 1 neighborhood
  - Based on the duality of cells: a cell is a base cell in one group while it is a deleted neighborhood cell in the other group

A	2	B	2	A	2	B	2
2	C	2	D	2	C	2	D
B	2	A	2	B	2	A	2
2	D	2	C	2	D	2	C
A	2	B	2	A	2	B	2
2	C	2	D	2	C	2	D
B	2	A	2	B	2	A	2
2	D	2	C	2	D	2	C

1	A	1	B	1	A	1	B
C	1	D	1	C	1	D	1
1	B	1	A	1	B	1	A
D	1	C	1	D	1	C	1
1	A	1	B	1	A	1	B
C	1	D	1	C	1	D	1
1	B	1	B	1	A	1	A
D	1	C	1	D	1	C	1

  - This method can not extend to test type 2 NPSFs because it depends on the duality concept

Tests for NPSFs

- With type 2 neighborhoods
  - The cells 0, 2, 6, and 8 are corner cells, whereas cells 1, 3, 5, and 7 are middle cells. Thus the duality concept does not hold
- With the two-group method
  - Each group consists of N/2 based cells and N/2 deleted neighborhood cells formed by 4 subgroups
  - Each subgroup consists of N/8 cells formed by all cells-A, all cells-B, all cells-C or all cells-D
  - A new test pattern can be applied to all N/2 neighborhoods of a group by writing into all N/8 cells of a subgroup, thus reducing the number of write operations by a factor of 4

Detection & Location of NPSFs

- Normally, all required patterns must be applied to the neighborhood, and after each pattern the base cell must be read. In this way all NPSFs can be located
- Basic NPSF location algorithm

```
Step 1: write base cells with 0;
Step 2: loop
        apply a patern; {it could change the base cell from 0 to 1}
        read base cell;
        endloop;
Step 3: write base cells with 1;
Step4: loop
        apply a pattern;{it could change the base cell from 0 to 1}
        read base cell;
        endloop;
```

Detection & Location of NPSFs

- When the read operations are performed only after certain patterns, it is only possible to detect the NPSF
- Basic NPSF detection algorithm

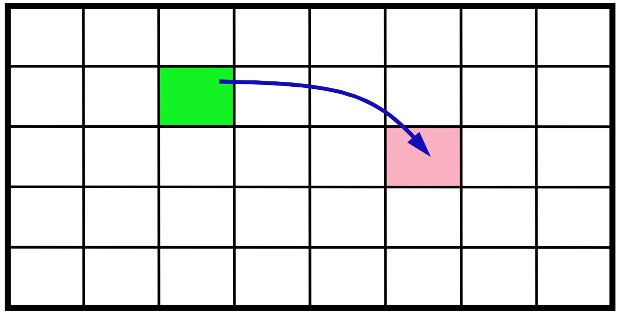
```
Step 1: write base cells with 0;
Step 2: loop
          apply a patern; {it could change the base cell from 0 to 1}
          read base cell;
          endloop;
Step 3: write base cells with 1;
Step4: loop
          apply a pattern;{it could change the base cell from 0 to 1}
          endloop;
          read base cell;
```

Tests for Word-Oriented Memories

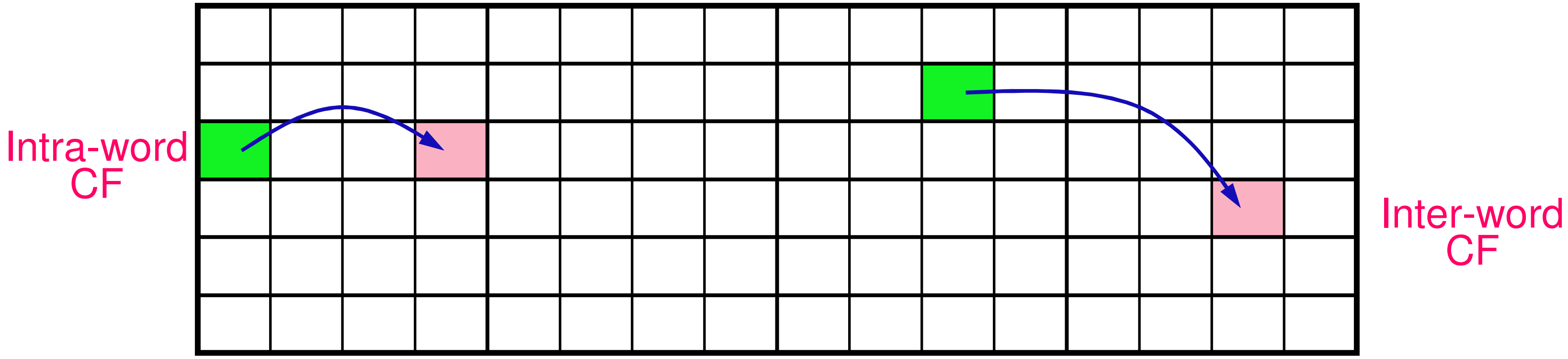
- Fault models for word-oriented memories (WOMs)
  - Only the class of memory cell array faults for bit-oriented memories (BOMs) has to be extended in order to cover WOMs
- The fault models for WOMs can be classified into two classes
  - Single-cell faults
    - SAFs, TFs, data retention faults (DRFs), etc.
  - Faults between memory cells
    - CFs
- Two classes of faults between memory cells for WOMs needed to be considered

Tests for Word-Oriented Memories

- CFs in BOMs



- CFs in WOMs
  - *Inter-word* CFs & *intra-word* CFs



Converting March Tests

- Any given BOM march test can be converted to a WOM march test
  - With additional tests to cover intra-word faults
- A WOM march test is a concatenation of two march tests
  - {Inter-word march test, intra-word march test}
- The inter-word march test can directly be obtained from the BOM march test
  - Replace the bit-operation “r0”, “w0”, “r1”, and “w1” with the word-operation “rD”, “wD”, “rD’”, and “wD’”, where D is called *data background*

Converting March Tests

- The intra-word faults can be detected by *a single march element* with *different operations* and *data backgrounds*
  - E.g., intra CFst can be covered by $(wd_1, rd_1, \ldots, wd_n, rd_n)$ with various data backgrounds (DBs)
  - Note that the DBs can be applied in any order
- The above intra-word test can be modified as follows, without any impact on the fault coverage
  - Extra Read operations can be added
  - The single march element can be divided into any number of march elements, and for each march element the addressing order can be chosen freely

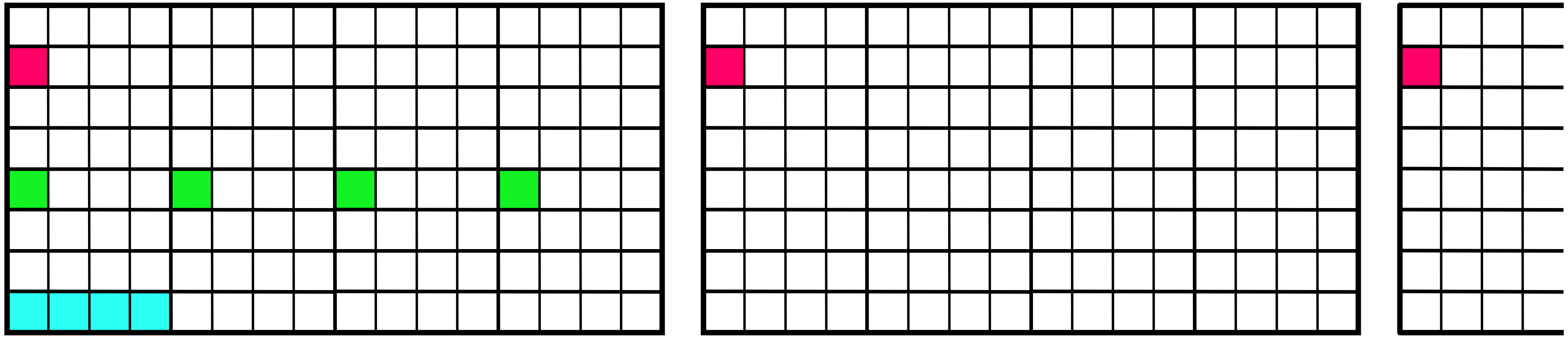
Compact WOM Tests

- If you have a bit-oriented march test, then you can obtain a compact WOM test with
  - Replace the bit-operation “r0”, “w0”, “r1”, and “w1” with all-0 and all-1 data backgrouds
  - Concatenate a march element $(wd, w\overline{d}, r\overline{d}, wd, rd)$ for d={0101..01,0011..11, …}
- For example, the March C- can be extended as follows to test a memory with 4-bit words

$$
\begin{aligned}
&\{\Updownarrow(w0000);\Uparrow(r0000,w11111);\Uparrow(r11111,w0000);\\
&\Downarrow(r0000,w1111);\Downarrow(r1111,w0000);\Updownarrow(r0000);\\
&\Updownarrow(w0101,w1010,r1010,w0101,r0101);\\
&\Updownarrow(w0011,w1100,r1100,w0011,r0011)\}
\end{aligned}
$$

Memory Organization & WOM Tests

- WOMs can be organized internally in many different ways
  - Adjacent; interleaved; sub-arrays



🟦 adjacent
🟩 interleaved
🟥 sub-array

  - For sub-array organized WOMs, the BOM tests for CFs will detect the CFs within a B-bit word such that no intra-word tests are required

Summary

- March tests for typical RAM cell faults have been presented
- Tests for address decoder faults and specific address decoder faults have been introduced
- March tests for NPSFs have been introduced
- Converting BOM march tests into WOM march tests has been discussed

References

- [1] M. Sachdev,"Defect Oriented Testing for CMOS Analog and Digital Circuits", Kluwer Academic, 1999.
- [2] M.-L. Bushnell and V.-D. Agrawal,"Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits", Kluwer Academic, 2000.
- [3] A. J. van de Goor and C. A. Verruijt,"An Overview of Deterministic Functional RAM Chip Testing", ACM Computing Surveys, vol. 22, no. 1, March 1990.
- [4] A. J. van de Goor, I. B. S. Tlili, and S. Hamdioui,"Converting March Tests for Bit-Oriented Memories into Tests for Bit-Oriented Memories", MTDT, pp. 46-52, 1998.
- [5] A. J. van de Goor and G. N. Gaydadjiev,"March U: a test for unlinked memory faults", IEE Proc. Circuit Devices Syse., vol.144, no. 3, pp.155-160, 1997.
- [6] D. S. Suk and S. M. Reddy," A march test for functional faults in semiconductor random-access memories" IEEE Trans. Comput., C-30, 12, pp.982-985, 1981.
- [7] K. L. Cheng, M. F. Tsai, and C. W. Wu,"Neighborhood pattern-sensitive fault testing and diagnostics for random-access memories", IEEE Trans. CAD, vol.21, no.11, pp. 1328-1336, nov., 2002.
- [8]C. F. Wu, C. T. Huang, and C. W. Wu"RAMSES: a fast memory fault simulator," DFT99, pp.199-202, 1996.