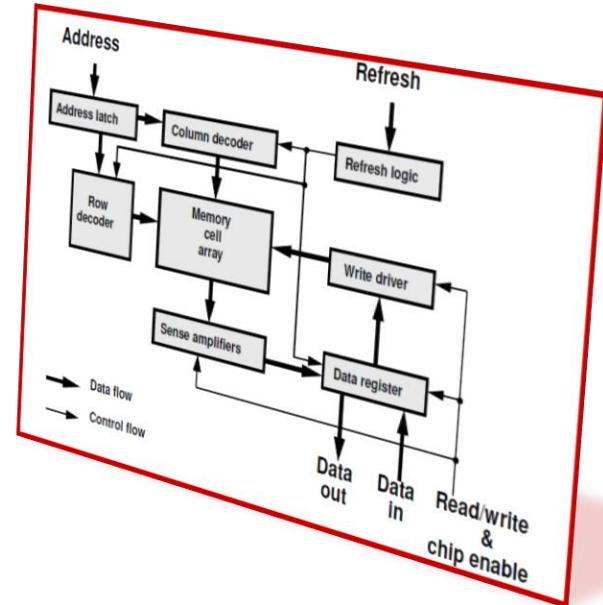


Outline

- Introduction
- Memory Fault Model
- Memory Test Algorithms
- Memory Fault Simulation (*not in exam)
- Memory Test Generation (*not in exam)
- Memory BIST (*not in exam)



Memory Fault Simulation

- What is *memory fault simulation*?
 - ◆ Given test algorithm, memory architecture
 - * find fault coverage of different fault models
- Why fault simulation?
 - ◆ Evaluate fault coverage efficiently,
 - * especially when many fault models
 - ◆ Help test algorithm design and optimization
 - ◆ Fault dictionary can be constructed for easy diagnosis

Sequential Memory Fault Simulation

```
For each fault /*  $N^2$  for 2-cell CF */  
Inject fault;  
For each test element /*  $N$  for March */  
{  
    Apply test element;  
    Report error output  
}
```

- Complexity is N^3 for 2-cell CF
 - ◆ This is very slow

Parallel Fault Simulation [Wu 02]

- Random Access Memory Simulator for Error Screening (RAMSES)
- Consider all faults at a time
- Complexity N^2

S/1

AGR := w0

SPT := @ /* Single-cell fault */

VTM := r0

RCV := w1

CFst <0;s/1>

AGR := v0

SPT := * /* All other cells are suspects */

VTM := r0

RCV := w1

For each test operation

{

If op is AGR then mark victim cells;
If op is RCV then release victim cells;
If op is VTM then report error;

}

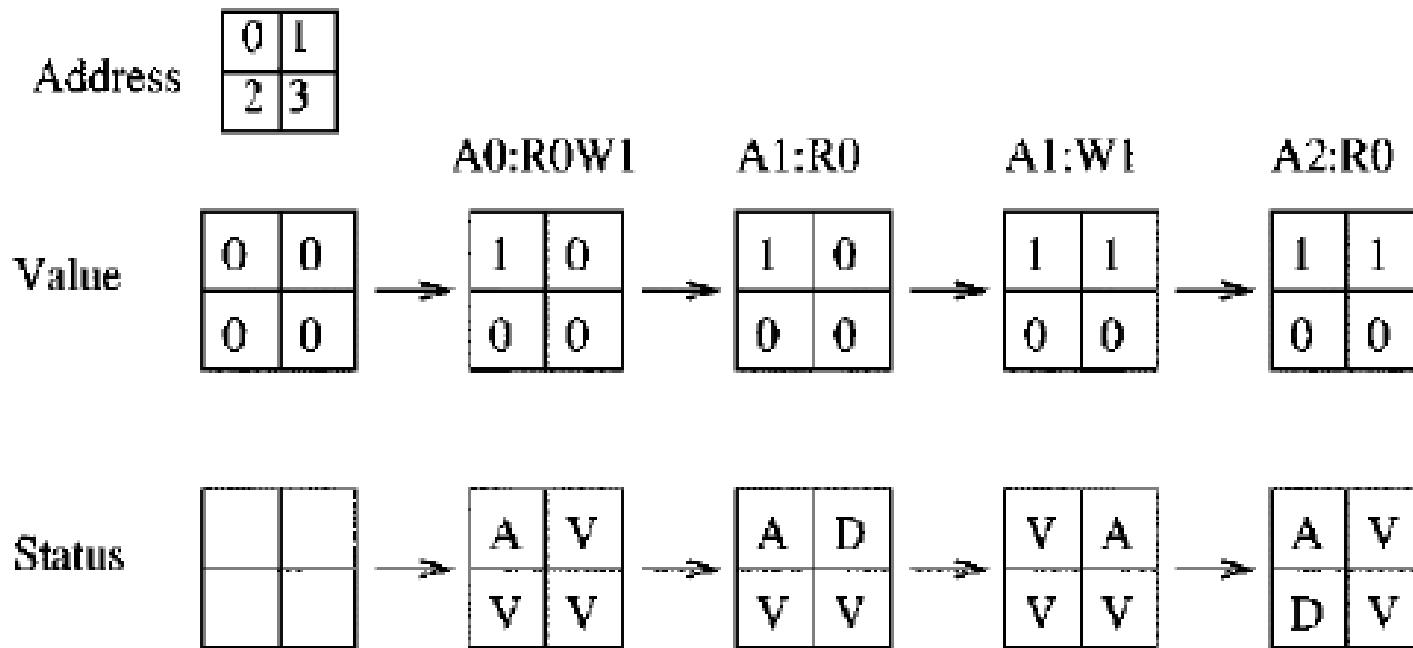
AGR=aggressor
SPT=suspects
VTM=victim
RCV=recover
@=this cell
*=all cells

RAMSES Algorithm

```
For each fault opeartion
    set op_flags;
    if (ARG ⊂ op_flags) {
        for each victim cell {
            set victim flags;
            set aggressor address;
        }
    }
    if (OP eq RCV) {
        clear victim flag;
        clear aggressor entry;
    }
    else if (OP eq VTM) {
        mark detected;
    }
}
```

RAMESE Example

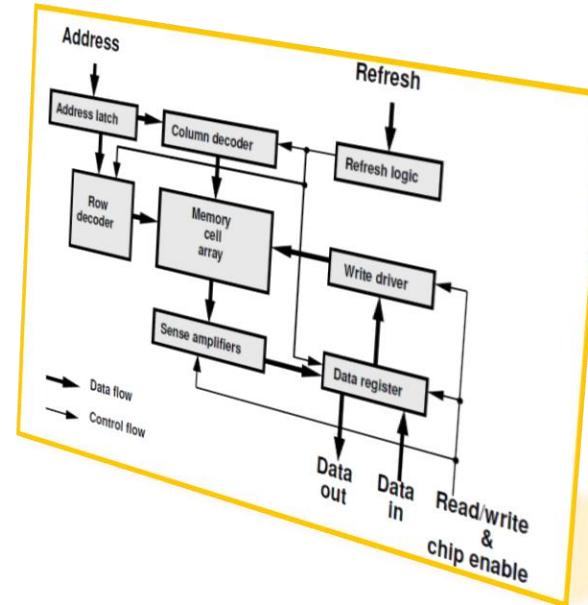
- $CF_{in} <\uparrow / \downarrow >$
- March element $\uparrow(r0, w1)$



D = detect; A = aggressor; V=victim

Outline

- Introduction
- Memory Fault Model
- Memory Test Algorithms
- Memory Fault Simulation
- Memory Test Generation (* not in exam)
- Memory BIST



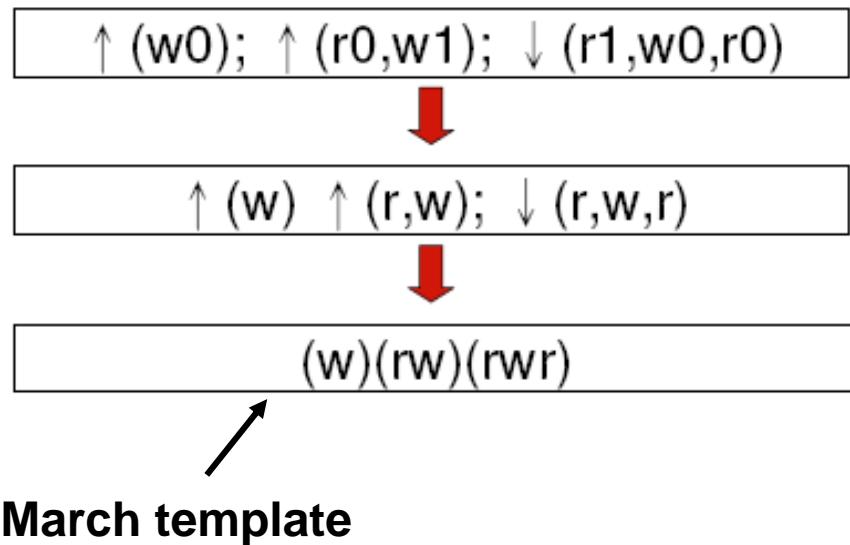
Memory Test Generation

- **What is memory test algorithm generation**
 - ◆ Given a set of target fault models and a test length constraint,
 - * generate a test with the highest fault coverage
- Priority setting for fault models

TAGS [Wu 00]

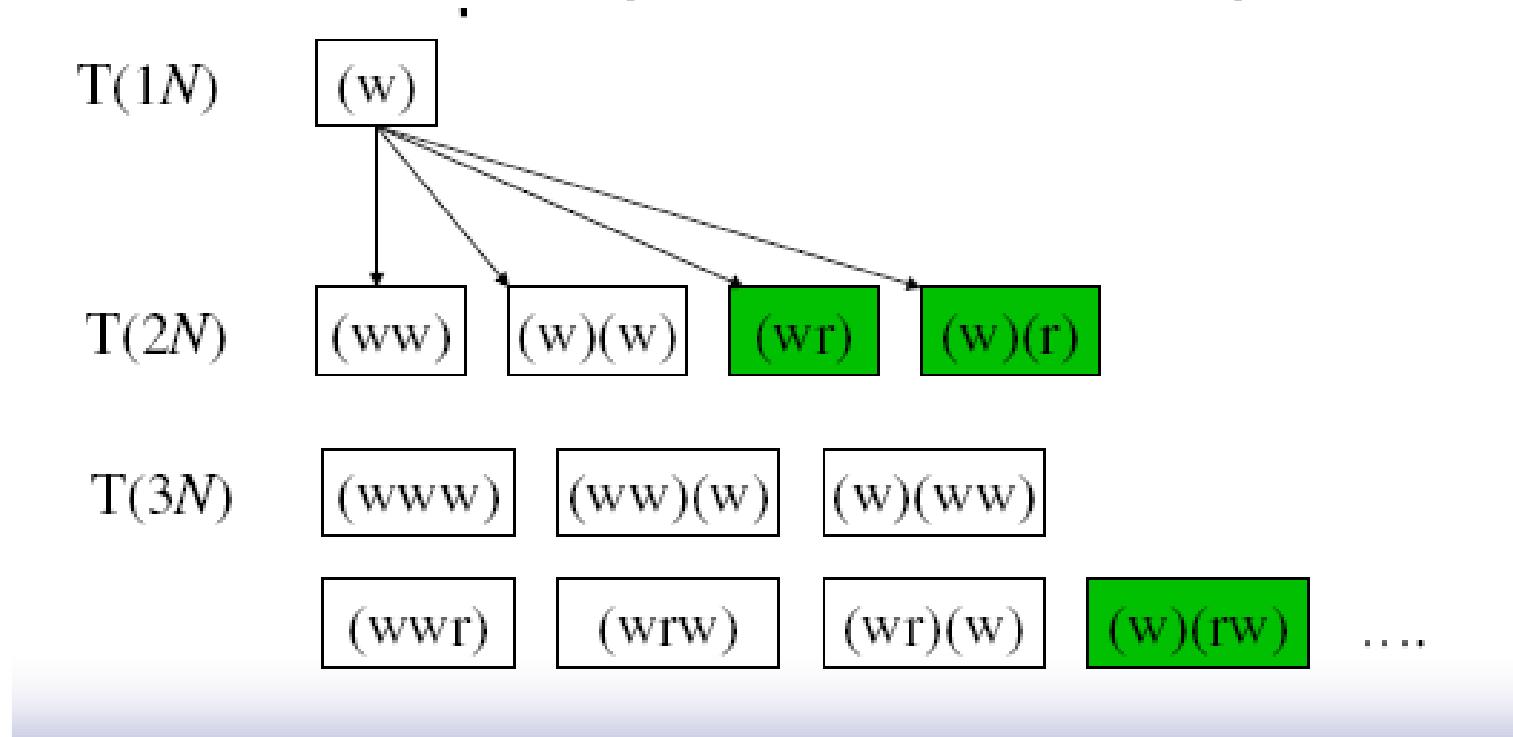
- Test Algorithm Generation by Simulation (TAGS)

- March template abstraction:



Template Set

- **Template set**
 - ◆ $T(cN)$ = a set of c march templates of N read or write
- Exhaustive generation of template set is very expensive
 - ◆ e.g., 6.7 million templates when $N = 9$
- Heuristics should be developed to select useful templates



TAGS Procedure

- 1. Initialize test length as $1N$, $T(1N) = \{(w)\}$;
- 2. Increase test length by $1N$: apply generation options;
- 3. Apply *filtering* heuristics;
- 4. Assign address orders and data backgrounds;
- 5. Fault simulation using RAMSES;
- 6. Drop ineffective tests;
- 7. Repeat 2-6 using the new template set until constraints met;

Template Generation and Filtering

- ***Generation*** heuristics:
 - ◆ (r) insertion
 - ◆ (...r), (r...) expansion
 - ◆ (w) insertion
 - ◆ (...w), (w...) expansion
- ***Filtering*** heuristics:
 - ◆ Consecutive read: (...rr...)
 - ◆ Repeated read: (r)(r)
 - ◆ Tailing single write: ...(w)

3N March Test

- 3N March Test generated by TAGS after step 3

- ◆ WWW Table 8.7

	Test
1	$\uparrow(w_0) \uparrow(w_1, r_1)$
2	$\uparrow(w_0) \downarrow(w_1, r_1)$
3	$\uparrow(w_0) \uparrow(w_0, r_0)$
4	$\uparrow(w_0) \downarrow(w_0, r_0)$
5	$\uparrow(w_0) \uparrow(r_0, w_1)$
6	$\uparrow(w_0) \downarrow(r_0, w_1)$
7	$\uparrow(w_0) \uparrow(r_0) \uparrow(r_0)$
8	$\uparrow(w_0) \uparrow(w_1) \uparrow(r_1)$

- 3N March test selected by RAMSES

- ◆ WWW Table 8.8

	Test
3	$\uparrow(w_0) \uparrow(w_0, r_0)$
5	$\uparrow(w_0) \uparrow(r_0, w_1)$
8	$\uparrow(w_0) \uparrow(w_1) \uparrow(r_1)$

TAGS Examples (1/2)

- WWW Table 8.9

$T(N)$	Name	Match algorithm
$1N$	M_1^1	$\uparrow (w0)$
$2N$	M_1^2	$\uparrow (w0) \uparrow (r0)$
$3N$	M_1^3	$\uparrow (w0) \uparrow (w1) \uparrow (r1)$
$3N$	M_2^3	$\uparrow (w0) \uparrow (r0, w1)$
$3N$	M_1^5	$\uparrow (w0) \downarrow (w1) \uparrow (r1)$
$3N$	M_2^5	$\uparrow (w0) \downarrow (r0, w1)$
$4N$	M_1^6	$\uparrow (w0) \downarrow (r0, w1) \uparrow (r1)$
$4N$	M_2^6	$\uparrow (w0) \downarrow (r0, w1, r1)$
$5N$	M_1^7	$\uparrow (w0) \uparrow (w1) \uparrow (r1, w0) \uparrow (r0)$
$5N$	M_2^7	$\uparrow (w0) \downarrow (r0, w1) \uparrow (r1, w0)$
$5N$	M_3^7	$\uparrow (w0) \uparrow (w1) \uparrow (r1, w0, r0)$
$6N$	M_1^8	$\uparrow (w0) \uparrow (w1) \uparrow (r1, w0) \downarrow (r0, w1)$
$6N$	M_2^8	$\uparrow (w0) \downarrow (r0, w1) \uparrow (r1, w0) \uparrow (r0)$
$6N$	M_3^8	$\uparrow (w0) \uparrow (r0, w1) \uparrow (r1, w0) \uparrow (r0)$
$6N$	M_4^8	$\uparrow (w0) \uparrow (r0, w1) \uparrow (r1, w0, r0)$
$6N$	M_5^8	$\uparrow (w0) \downarrow (r0, w1) \uparrow (r1, w0, r0)$
$7N$	M_1^9	$\uparrow (w0) \uparrow (r0, w1) \uparrow (r1, w0) \downarrow (r0, w1)$

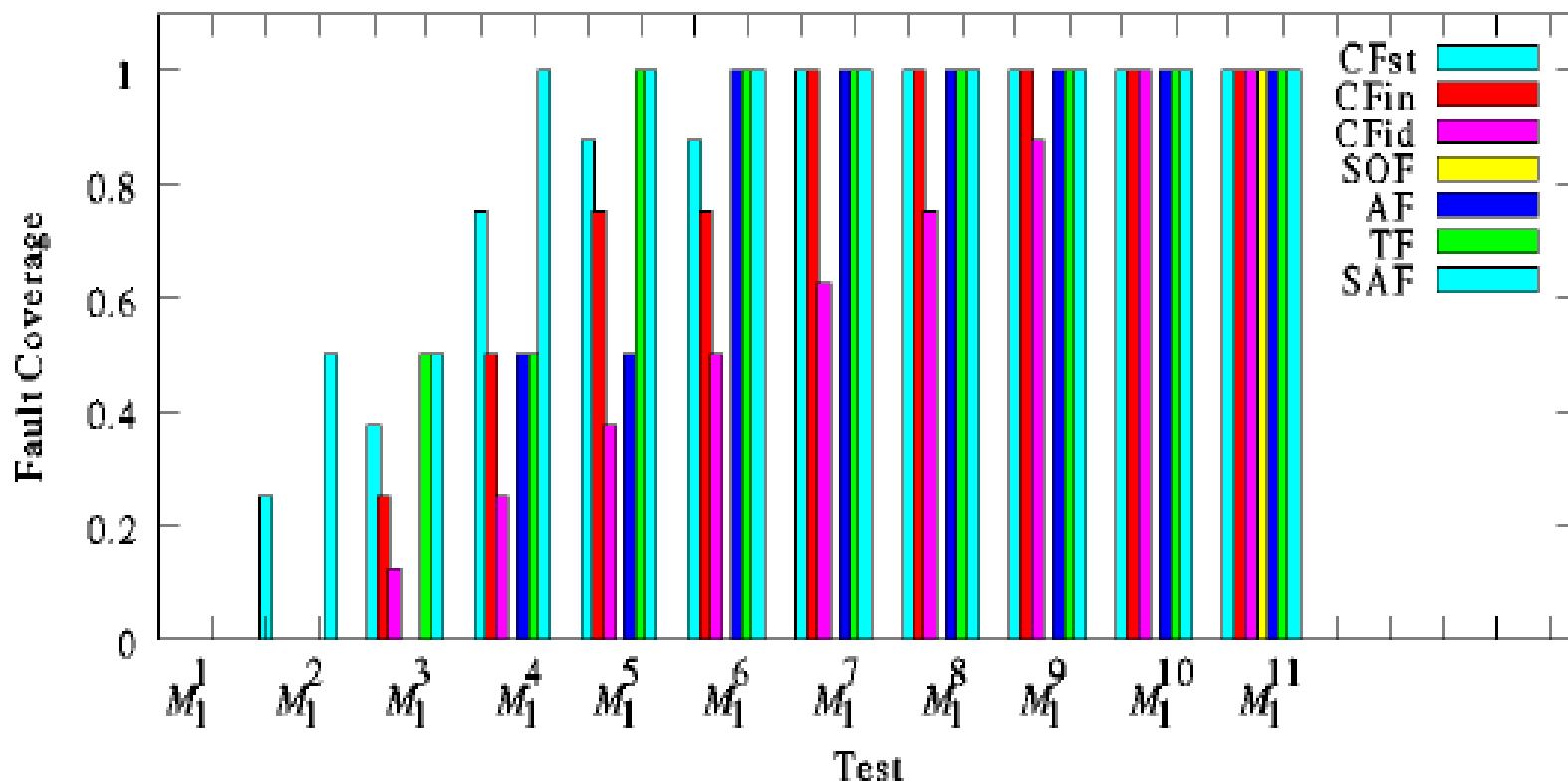
TAGS Examples (2/2)

- WWW Table 8.9 cont'd

7N	M_1^7	$\uparrow(w0) \uparrow(r0, w1) \uparrow(r1, w0) \downarrow(r0, w1)$
7N	M_2^7	$\uparrow(w0) \uparrow(w1) \downarrow(r1, w0) \uparrow(r0, w1, r1)$
7N	M_3^7	$\uparrow(w0) \downarrow(r0, w1) \uparrow(r1, w0, r0) \uparrow(r0)$
7N	M_4^7	$\uparrow(w0) \uparrow(r0, w1) \uparrow(r1, w0, r0) \uparrow(r0)$
8N	M_1^8	$\uparrow(w0) \uparrow(r0, w1) \uparrow(r1, w0) \downarrow(r0, w1)$ $\uparrow(r1)$
8N	M_2^8	$\uparrow(w0) \uparrow(r0, w1) \uparrow(r1, w0)$ $\downarrow(r0, w1, r1)$
9N	M_1^9	$\uparrow(w0) \uparrow(r0, w1) \uparrow(r1, w0) \downarrow(r0, w1)$ $\downarrow(r1, w0)$
9N	M_2^9	$\uparrow(w0) \uparrow(r0, w1) \uparrow(r1, w0)$ $\downarrow(r0, w1, r1) \uparrow(r1)$
10N	M_1^{10}	$\uparrow(w0) \uparrow(r0, w1) \uparrow(r1, w0) \downarrow(r0, w1)$ $\downarrow(r1, w0) \uparrow(r0)$
10N	M_2^{10}	$\uparrow(w0) \uparrow(r0, w1) \uparrow(r1, w0) \downarrow(r0, w1)$ $\downarrow(r1, w0, r0)$
11N	M_1^{11}	$\uparrow(w0) \uparrow(r0, w1) \uparrow(r1, w0) \downarrow(r0, w1)$ $\downarrow(r1, w0, r0) \uparrow(r0)$

RAMSES Simulation Results

- WWW Figure 8.13



RAMSES Simulation Results

- WWW Figure 8.14
 - ◆ Five different $6N$ tests

