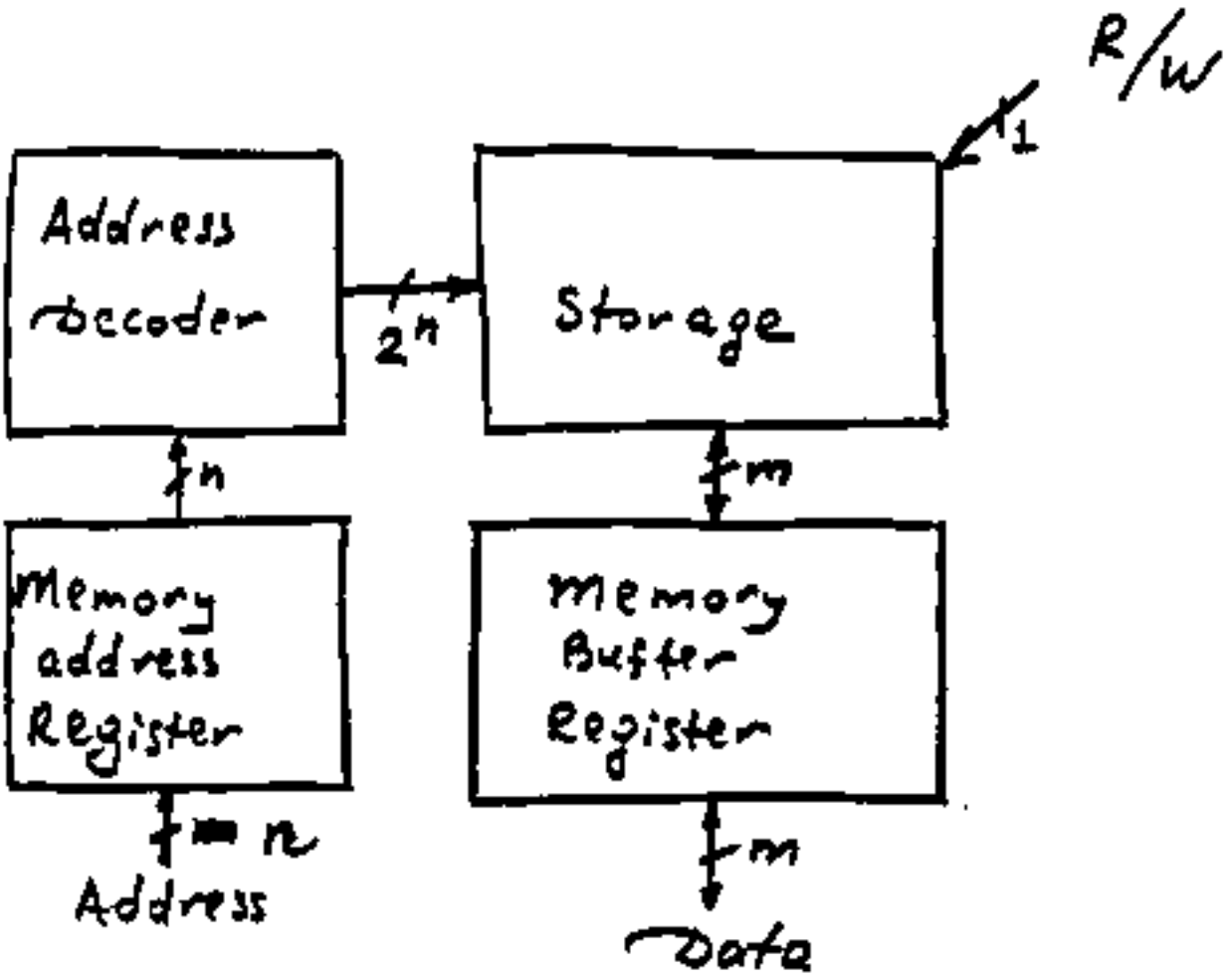
Memory Testing



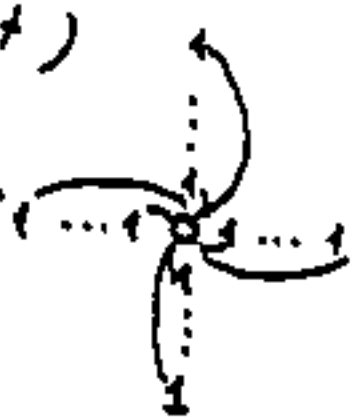
Memory Organization

Faults in a Storage (Semiconductor memories)

1. Stuck-at faults in binary cells
2. Intermittent faults in cells
3. Row or columns bridgings (AND or OR typ
4. Pattern sensitivity (crosstalks)
   (most difficult to detect)

4.1. 

4.2. 

4.3. 

5. Hold-time errors ⇒ different cells have different hold-time ⇒
   graceful degradation of performance

DATA RETENTION FAULTS

Decoding Errors

1. Wrong addressing
2. No addressing
3. Multiple addressing

Test Complexity: $L(N)$ - number of READ and WRITE operation in a test

N - number of binary cells

For exhaustive test $L(N) \simeq 2^N$ (unrealis

For most testing procedures

$$C_1 N \leq L(N) \leq C_2 N^2$$

Goals of Memory Testing

1. Check the capability of each cell to store 0 and 1 (to avoid faulty cells) in computing mode). LOCATE FAULTY CELLS.
2. Check each cell for addressability & and readability
3. Check for pattern sensitivity (crosstalks)
4. Check hold-times of each cell

DETERMINISTIC TESTS

Examples of Memory Testing Procedures

8.5

1. MSCAN

WRITE: $C_i \leftarrow 0$

READ: $C_i = 0$ (?)

$i = 1, \ldots, N$

WRITE: $C_i \leftarrow 1$

READ: $C_i = 1$ (?)

$L(N) = 4N$

Detects stuck-at faults in cells

Van der Goor "Memory Testing"
Prentice Hall

8.6

2. Row/Column Bar

Column Bar:

WRITE

```
1 0 1 0 ... 1 0
1 0 1 0 ... 1 0
. . . . ... . .
1 0 1 0 ... 1 0
```

READ ↗

WRITE

```
0 1 0 1 ... 0 1
0 1 0 1 ... 0 1
. . . . ... . .
0 1 0 1 ... 0 1
```

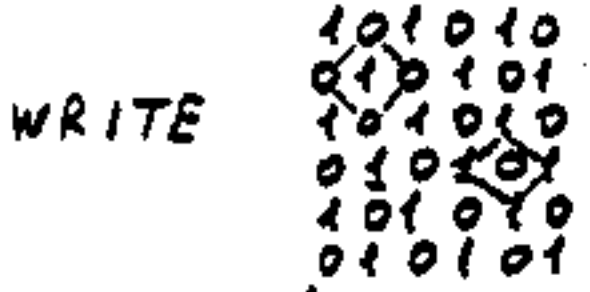
(complimentary pattern)

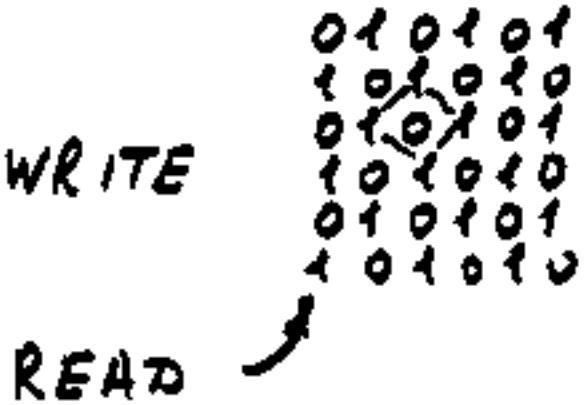
READ ↗

$$L(N) = 4N$$

Detects stuck-at faults in cells, column/row bridgings, 50% of wrong addressings

3. Chessboard Test





$L(N) = 4N$

Detects stuck-at faults in cells, row and column bridgings and some pattern sensitive errors (every 0 (1) surrounded by 1's (0's))

3' Marching 1's and 0's (GALPAT)

1) WRITE $C_i \leftarrow 0$ $i = 0, 1, \ldots, N-1$

2) For $i = 0, 1, \ldots, N-1$

READ $C_i = 0$?

WRITE $C_i \leftarrow 1$

READ $C_i = 1$?

3) For $i = N-1, N-2, \ldots, 0$

WRITE $C_i \leftarrow 0$

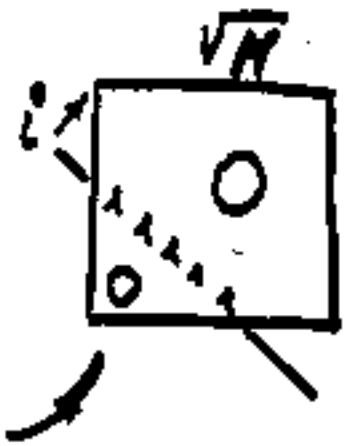
READ $C_i = 0$?

4) Repeate 1)-3) ~~with~~ interchanging 0's and 1's

$L(N) = 12N$

Detects decoding errors, minimal check on cell interactions

4. Shifted Diagonal



WRITE $i = 1, 2, \ldots, 2\sqrt{N}-1$

READ

WRITE complimentary pattern

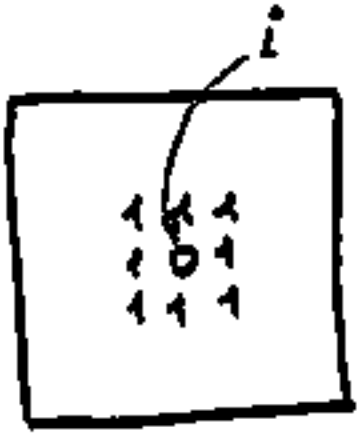
READ complimentary pattern

$$L(N) = 8N^{3/2}$$

detects stuck-at faults, row and column bridgings, wrong addressing, pattern sensitive faults (every two cells in the same row or column are neighbours)

5. Ping-Pong Procedures

5.1 Local Ping-Pong



WRITE $i = 1, \ldots, N$

READ

WRITE complimentary pattern

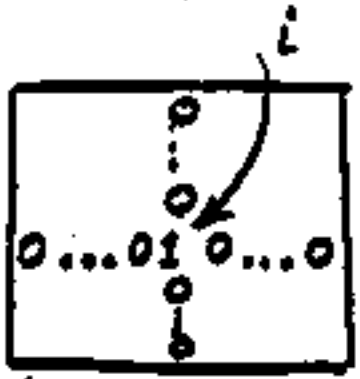
READ complimentary pattern

9 WRITES + 1 READ for every i

$L(N) = 9N$

FOR THE FIRST TWO STEPS.

5.2. Column / Row Ping-Pong



WRITE $i = 1, \ldots, N$

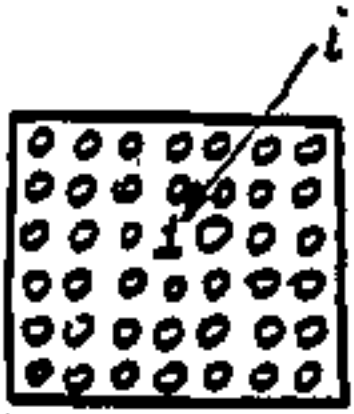
READ

WRITE complimentrary pattern

READ complimentary pattern

$$L(N) = 4N^{3/2} = 4N^{1.5}$$

5.3. Full Ping-Pong



WRITE $i = 1, \ldots, N$

READ

WRITE complimentary pattern

READ complimentary pattern

$$L(N) = 2N^2 + 2N$$

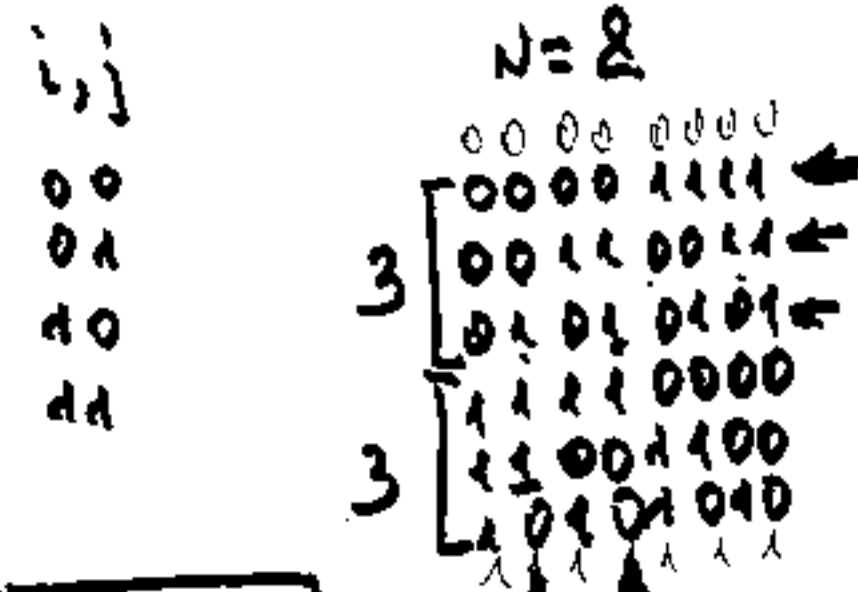
For ping-pong procedures

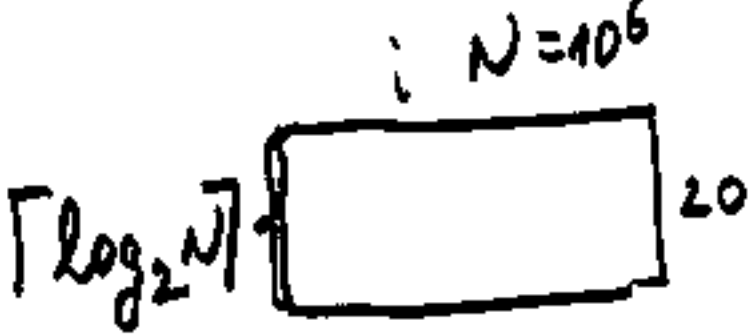
$$L(N) \geq gN^2 + 2N$$

g is the size of neighbourhoods

K=2 - couplings

STATIC PSFs
DYNAMIC





↓0, ↑0
↓1, ↑1

DYNAMIC $L(N) = 2N \cdot 2\lceil \log_2 N \rceil =$ $2\lceil \log_2 N \rceil$ NUMBER OF BACKGROUNDS

$4N \log_2 N$

k-couplings crosstalks between

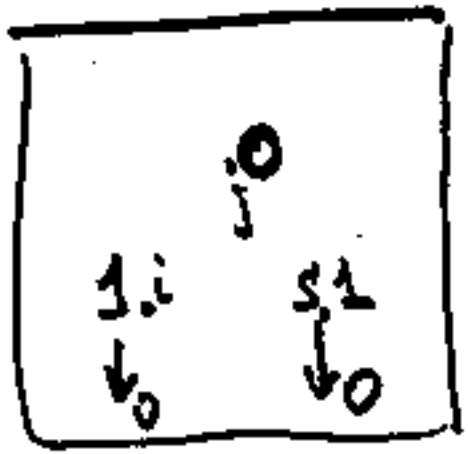
any k cells in the memory

fix a pattern in k=2 ($k \geq 2$)

change data in one cell ⇒

data will be changed in another cell
of the k-tuple.

3-couplings (i, j, s)



SCRAMBLING TABLE ?

Summary on Test Complexities

1. Devices with constant complexity $T(m) = Const$

   m-input XOR gates, adders, subtractors
   MAJORITY GATES, BIT-SLICE DEVICES

2. Devices with square root complexity

   $T(m) = c\sqrt{m}$

   Uniform NAND (NOR) trees

3. Devices with linear complexity $T(m) = cm$

   m-input AND, OR, NAND, NOR gates
   Reed-Muller networs, fanout-free networks
   multiplexers, k out of m

4. Devices with exponential complexity

   $T(m) = c\,2^m$

   2-level AND-OR (OR-AND) networks,
   decoders