Lecture 16
Pattern Sensitive and Electrical Memory Test

- Notation
- Neighborhood pattern sensitive fault algorithms
- Cache DRAM and ROM tests
- Memory Electrical Parametric Tests
- Summary

Notation

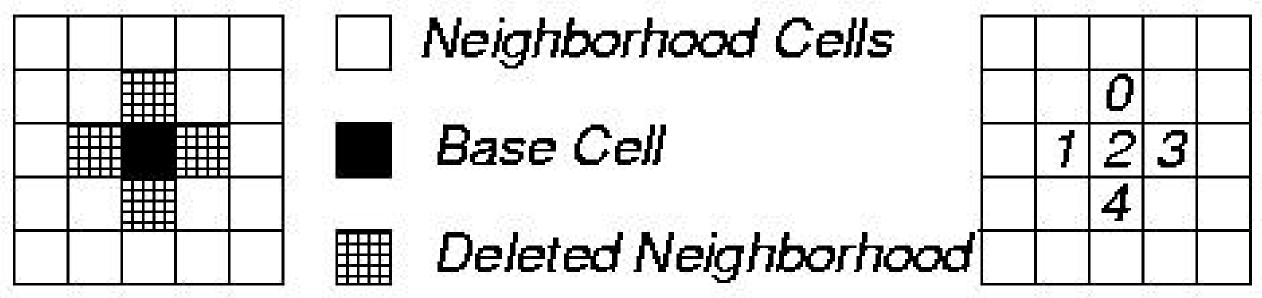
- ANPSF -- Active Neighborhood Pattern Sensitive Fault
- APNPSF – Active and Passive Neighborhood PSF
- Neighborhood -- Immediate cluster of cells whose pattern makes base cell fail
- NPSF -- Neighborhood Pattern Sensitive Fault
- PNPSF -- Passive Neighborhood PSF
- SNPSF -- Static Neighborhood Pattern Sensitive Fault

Neighborhood Pattern Sensitive Coupling Faults

- Cell i's ability to change influenced by all other memory cell contents, which may be a 0/1 pattern or a transition pattern.
- Most general k-Coupling Fault
- Base cell -- cell under test
- Deleted neighborhood -- neighborhood without the base cell
- Neighborhood is single position around base cell
- Testing assumes read operations are fault free

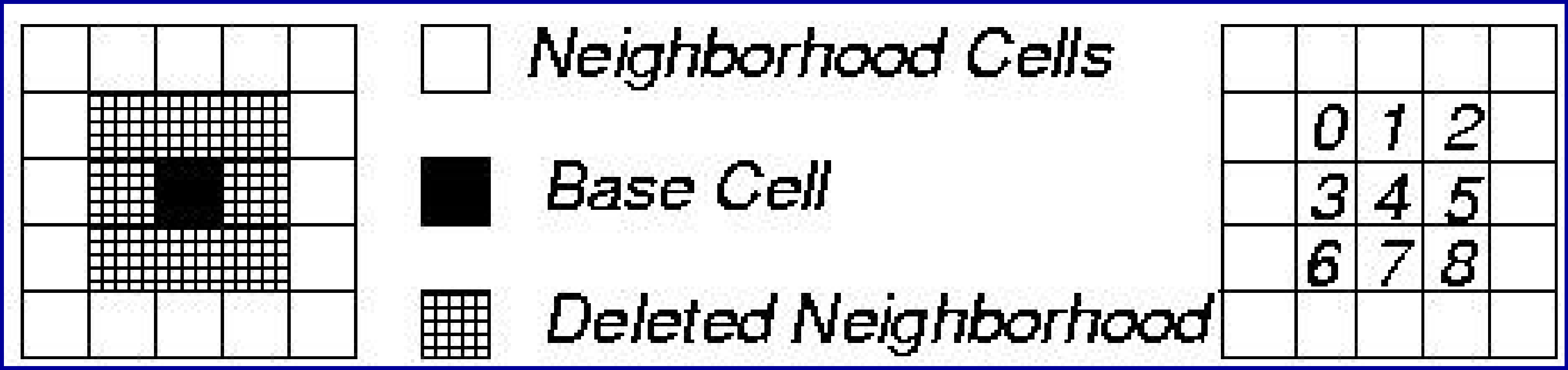
Type 1 Active NPSF

- Active: Base cell changes when one deleted neighborhood cell transitions
- Condition for detection & location: Each base cell must be read in state 0 and state 1, for all possible deleted neighborhood pattern changes.
- $C_{i,j}\ \langle d_0, d_1, d_3, d_4\ ; b\rangle$
- $C_{i,j}\ \langle 0, \downarrow, 1, 1; 0\rangle$ and $C_{i,j}\ \langle 0, \downarrow, 1, 1; \updownarrow\rangle$



Type 2 Active NPSF

- Used when diagonal couplings are significant, and do not necessarily cause horizontal/vertical coupling



Passive NPSF

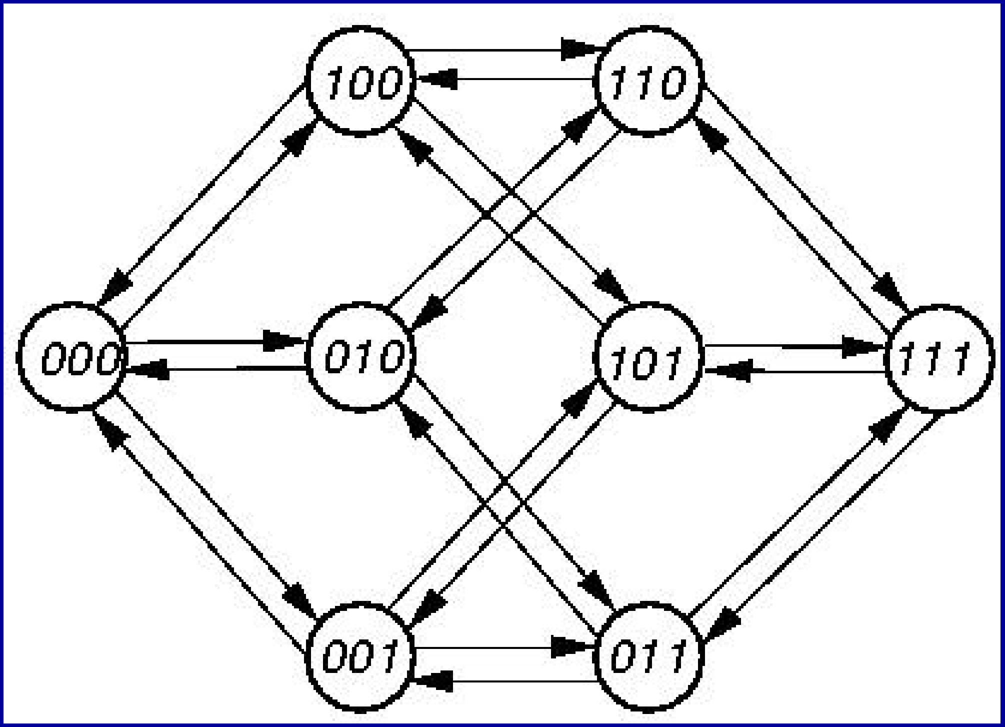
- **Passive: A certain neighborhood pattern prevents the base cell from changing**
- **Condition for detection and location: Each base cell must be written and read in state 0 and in state 1, for all deleted neighborhood pattern changes.**
- **↑/ 0 (↓/1) -- Base cell fault effect indicating that base cannot change**

Static NPSF

- Static: Base cell forced into a particular state when deleted neighborhood contains particular pattern.
- Differs from active -- need not have a transition to sensitive SNPSF
- Condition for detection and location: Apply all 0 and 1 combinations to k-cell neighborhood, and verify that each base cell was written.
- $C_{i,j}$ < 0, 1, 0, 1; - / 0> and
  $C_{i,j}$ < 0, 1, 0, 1; - / 1>

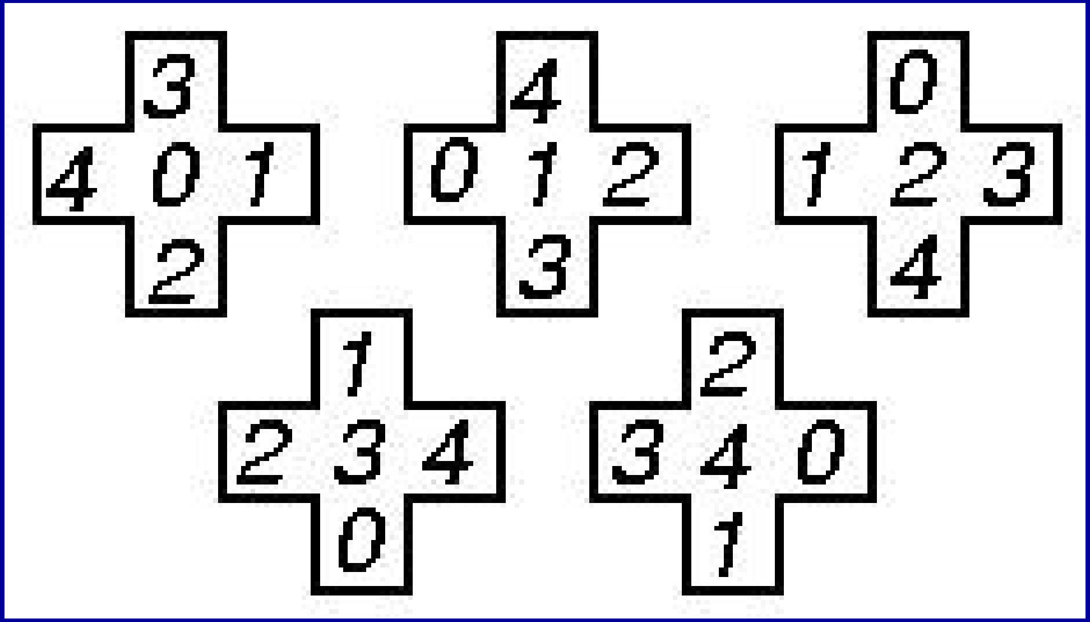
Eulerian / Hamiltonian Graph Tour Sequences

- Both used for writing shorter patterns
- Hamiltonian – traverses each graph node once
- Eulerian – traverses each graph arc exactly once



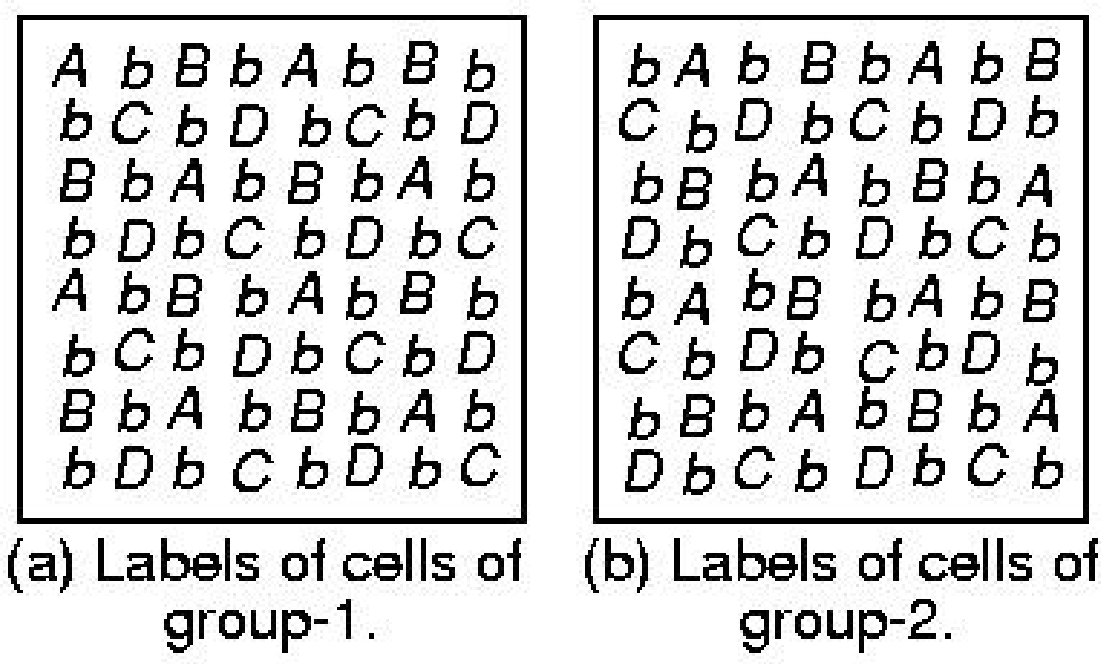
Type 1 Tiling Neighborhoods

- Write changes **k** different neighborhoods
- Tiling Method: Cover all memory with non-overlapping neighborhoods



Two Group Method

- Only for Type-1 neighborhoods
- Use checkerboard pattern, cell is simultaneously a base cell in group 1, and a deleted neighborhood cell in 2



A	b	B	b	A	b	B	b
b	C	b	D	b	C	b	D
B	b	A	b	B	b	A	b
b	D	b	C	b	D	b	C
A	b	B	b	A	b	B	b
b	C	b	D	b	C	b	D
B	b	A	b	B	b	A	b
b	D	b	C	b	D	b	C

(a) Labels of cells of group-1.

b	A	b	B	b	A	b	B
C	b	D	b	C	b	D	b
b	B	b	A	b	B	b	A
D	b	C	b	D	b	C	b
b	A	b	B	b	A	b	B
C	b	D	b	C	b	D	b
b	B	b	A	b	B	b	A
D	b	C	b	D	b	C	b

(b) Labels of cells of group-2.

NPSF Fault Detection and Location Algorithm

```
write base-cells with 0;
loop
    apply a pattern; { it could change the
        base-cell from 0 to 1. }
    read base-cell;
endloop;
write base-cells with 1;
loop
    apply a pattern; { it could change the
        base-cell from 1 to 0. }
    read base-cell;
endloop;
```

NPSF Testing Algorithm Summary

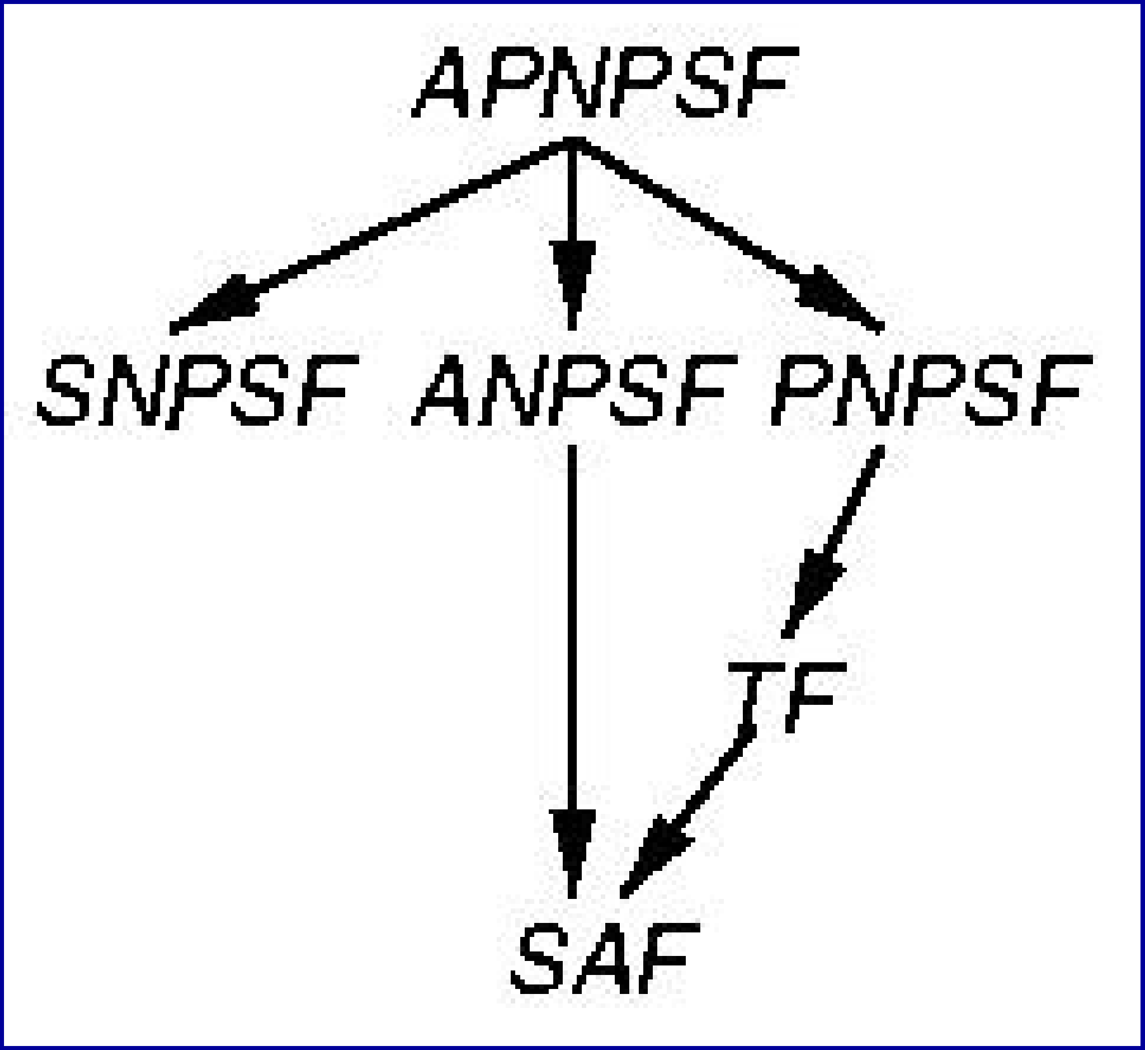
- A: active, P: passive, S: static
- D: Detects Faults, L: Locates Faults

Algorithm	Fault Location?	Fault Coverage					Operation Count
				NPSF			
		SAF	TF	A	P	S	
TDANPSF1G	No	L		D			163.5 n
TLAPNPSF1G	Yes	L	L	L	L	L	195.5 n
TLAPNPSF2T	Yes	L	L	L	L		5122 n
TLAPNPSF1T	Yes	L	L	L	L		194 n
TLSNPSF1G	Yes	L				L	43.5 n
TLSNPSF1T	Yes	L				L	39.2 n
TLSNPSF2T	Yes	L				L	569.78 n
TDSNPSF1G	No	L				D	36.125 n

NPSF Testing Algorithms

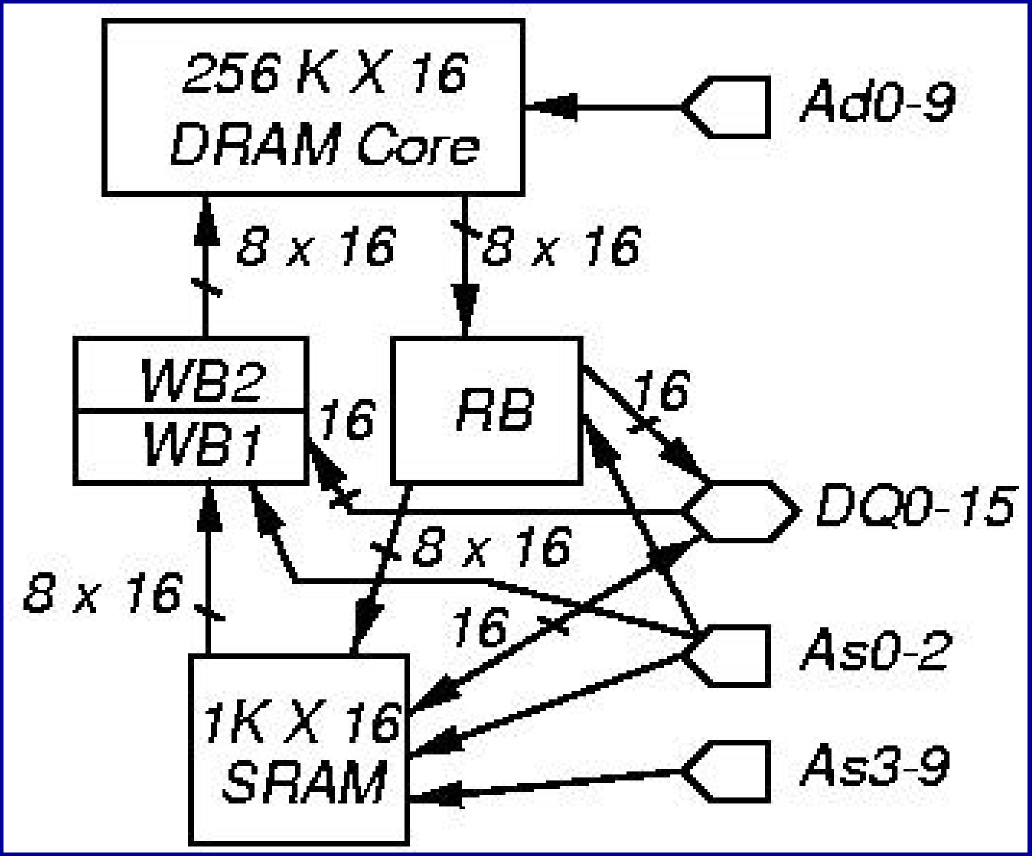
Algorithm	Neighborhood	Method	k
TDANPSF1G	Type-1	2 Group	5
TLAPNPSF1G	Type-1	2 Group	5
TLAPNPSF2T	Type-2	Tiling	9
TLAPNPSF1T	Type-1	Tiling	5
TLSNPSF1G	Type-1	2 Group	5
TLSNPSF1T	Type-1	Tiling	5
TLSNPSF2T	Type-2	Tiling	9
TDSNPSF1G	Type-1	2 Group	5

Fault Hierarchy



Cache DRAM Testing

- Combines DRAM with SRAM cache



Required Cache DRAM Tests

- DRAM Functional Test
- SRAM Functional Test
- Data Transfer Test between SRAM and DRAM
- High-Speed Operation Test (100 MHz)
- Concurrent Operation Test
- Cache Miss Test

Testing Extremely Fast DRAMS -- RAMBUS

- Use cheap and paid for ATE for die-sort test, burn-in test, & failure analysis
- Use expensive, high-speed Hewlett-Packard 500 MHz HP-8300, F660 ATE with design-for-testability (DFT) hardware for high-speed interface logic test
  - Allows direct memory core access at pins
  - Bypasses high-speed bus
  - Use cheap, slow ATE for memory test with PLL
  - Need low-inductance socket, short cables, PLL jitter testing, & time-domain reflectometry
  - Need critical path-delay fault timing tests

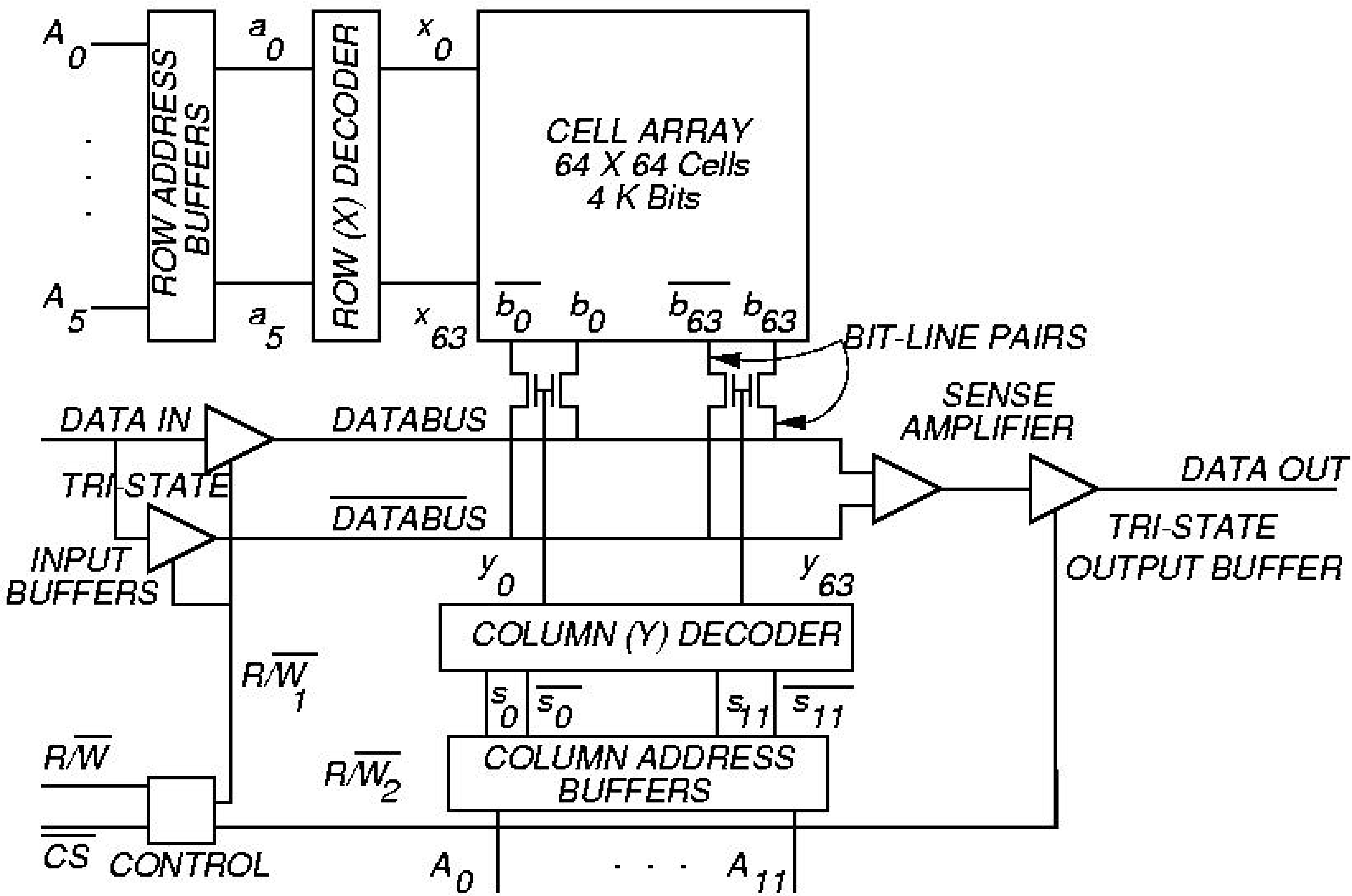
Functional ROM Testing

- Unidirectional SAF model -- only sa0 faults or only sa1 faults
- Store cyclic redundancy code (CRC) on ROM, ATE reads ROM & recomputes CRC, compares with ROM CRC
  - Tests single-bit errors, double-bit errors, odd-bit errors, multiple adjacent errors

Electrical Testing

- Test for:
  - Major voltage / current / delay deviation from part data book value
  - Unacceptable operation limits
  - Divided bit-line voltage imbalance in RAM
  - RAM sleeping sickness -- broken capacitor, leaks -- shortens refresh interval

RAM Organization



DC Parametric Tests

- Production test -- done during burn-in
  - Applied to all chips
  - Chips experience high temperature + over-voltage power supply
  - Catches initial, early lifetime component failures -- avoid selling chips that fail soon

Test Output Leakage Current

1. Apply high to chip select, deselect chip
2. Set chip pins to be in tri-state mode
3. Force high on each data-out line – measure I_{OZ}
4. Force low on each data-out line – measure I_{OZ}
5. Select chip (low on chip select)
6. Set read, force high on each address/data line, measure I_I
7. Set read, force low on each address/data line, measure I_I

Possible Test Outcomes:

1. I_{OZ} < 10 mA and I_I < 10 mA (passes)
2. $I_{OZ} \geq$ 10 mA (fails)
3. $I_I \geq$ 10 mA (fails)

Voltage Bump Test

- Tests if power supply variations make RAM read out bad data -- DRAM C shorted to supply

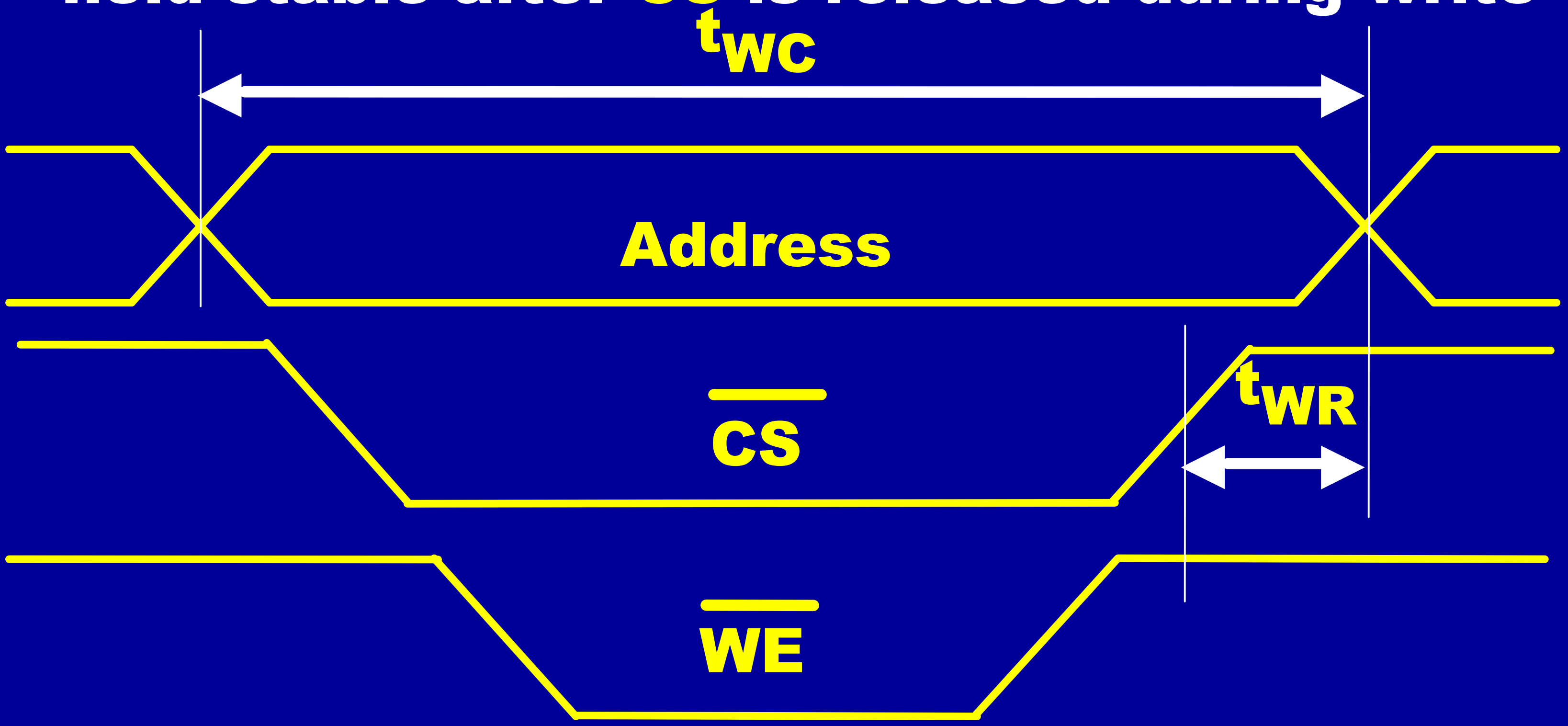
1. 2. 3. 4.	Zero out memory. Increase power above V_{CC} in 0.01 V. steps. For each voltage, read memory. Stop as soon as 1 is read anywhere, record supply V. as V_{high} Fill memory with 1's. Decrease power below V_{CC} in 0.01 V. steps. For each voltage, read memory. Stop as soon as 0 is read anywhere, record supply V. as V_{low}.
	Possible Test Outcomes:
1.	V_{high} and V_{low} inconsistent with data book (fails)

AC Parametric Tests

- Set a DC bias voltage level on pins
- Apply AC voltages at some frequencies & measure terminal impedance or dynamic resistance
- Determines chip delays caused by input & output C's
- No information on functional data capabilities or DC parameters

Write Release Time Tests

- t_{WC} – Write Cycle Time – minimum time required for 1 write cycle
- t_{WR} -- Address set-up time sensitivity – Write Release Time that address must be held stable after $\overline{CS}$ is released during write



Access Time Tests

1. Split memory into 2 halves.
2. Write 0 in 1st half and 1 in other half.
3. Read entire memory and check correctness.
4. Alternate between addresses in two halves

1. Speed up read access time until reading fails, and take that time as access time delay.

- **Characterization:**

  Use MATS++ with increasingly shorter access time until failure.

  Use March C instead of MATS++.

- **Production test:** run MATS++ at specified access time, and see if memory fails.

Running Time Tests

Method:

Perform read operations of 0s and 1s from alternating addresses at specified rapid speed.

Alternate characterization method:

Alternate read operations at increasingly rapid speeds until an operation fails.

Sense Amplifier Recovery Fault Tests

- Write operation followed by read/write at different address

Method:

1. Write repeating pattern ddddddd$\bar{d}$ to memory locations (d is 0 or 1);
2. Read long string of 0s (1s) starting at 1st location up to location with $\bar{d}$.
3. Read single 1 (0) from location with $\bar{d}$.
4. Repeat Steps 2 and 3, but writing rather than reading in Step 2.

Dual-Port SRAM Tests

Standby Current Test

Method:

Check all 4 possibilities for voltage combinations at 2 ports. 4 more Combinations occur if both ports have either TTL or CMOS level inputs.

Possible Test Outcomes:

Test fails if one port does not meet the current specification.

Tests of Dual-Ported RAMs

Test both RAM access ports simultaneously

- Write data into interrupt location of 1 port
- Monitor **INT** output of other port to see if interruption sensed at other port

Arbitration Test

- Test arbitration hardware between 2 ports in RAM
- If semaphore does not set or release, or if RAM locks up, then chip is faulty

Semaphore Testing Method:
For each port, request, verify, and release each semaphore latch.

Memory Testing Summary

- Multiple fault models are essential
- Combination of tests is essential:
  - March -- SRAM and DRAM
  - NPSF -- DRAM
  - DC Parametric -- Both
  - AC Parametric -- Both
- Inductive Fault Analysis is now required