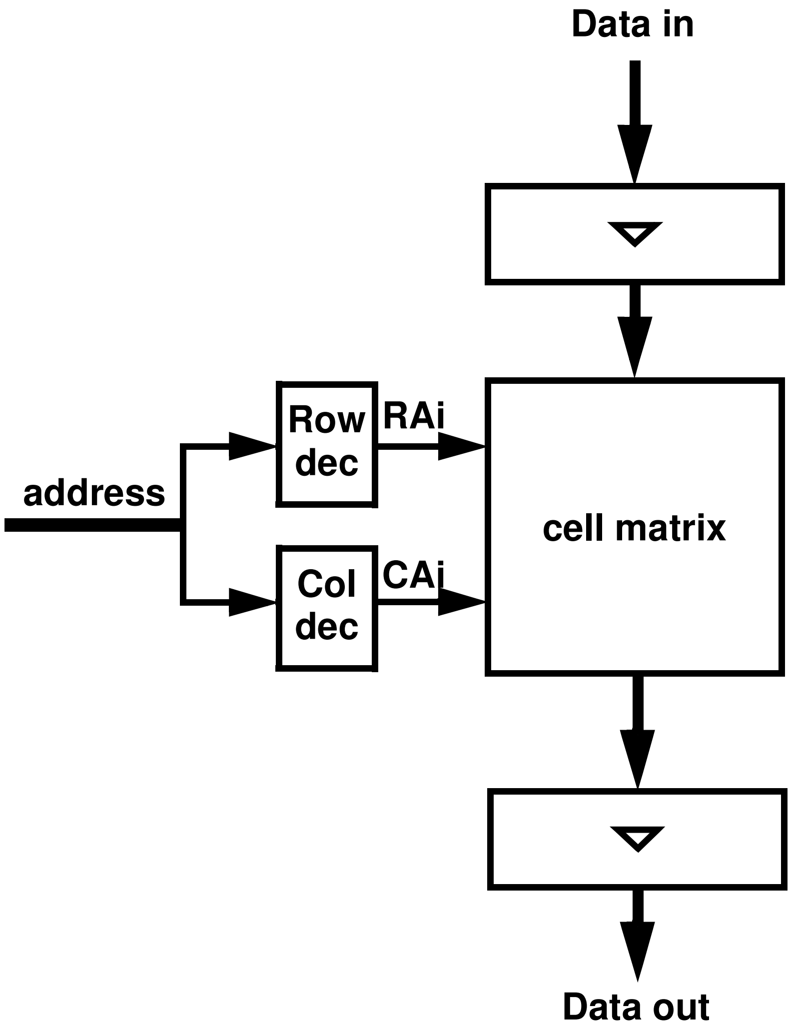
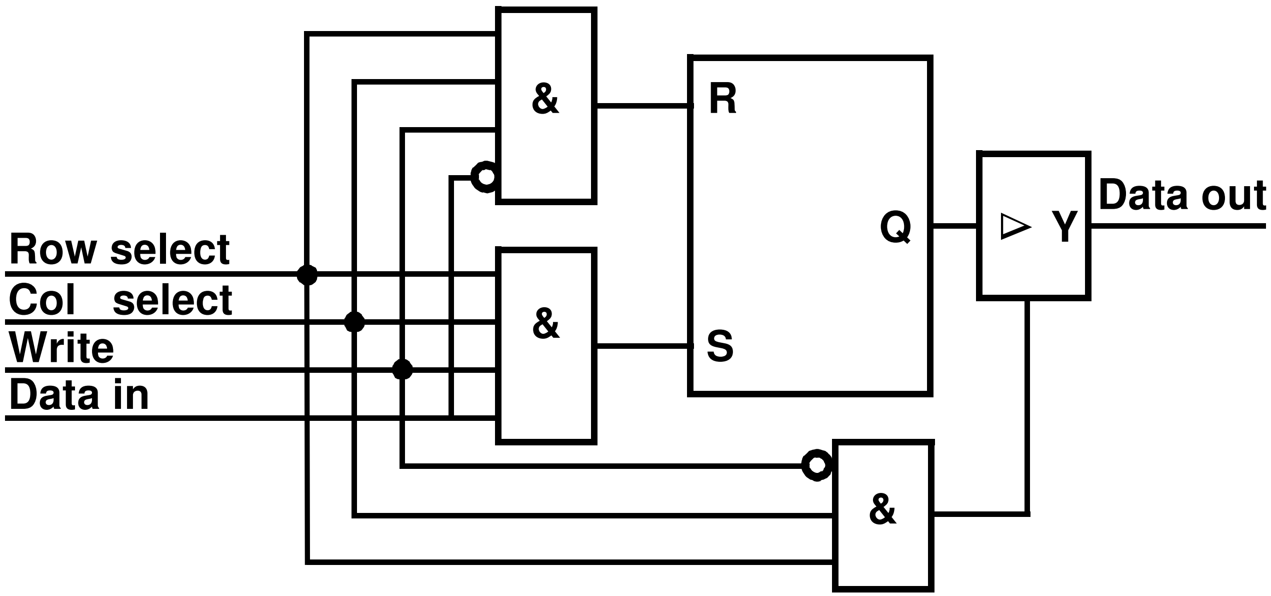
Memory testing

- **Functional model of a memory**

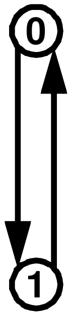
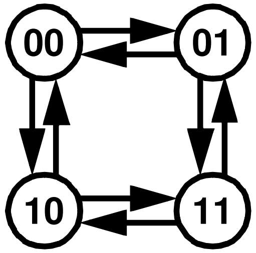




&
R
Data out
Q
Y
Row select
Col select
Write
Data in
&
S
&

Permanent faults

- **Fault model:**

Functional unit	Fault location	Fault	Effect		Functional test
			Write	Read	
Cell matrix	Single cell	Q: s-a-v	no effect	s-a-v	
		Y: s-a-Z	no effect	s-a-c	
		D s-a-v	missing $v \to \overline{v}$ transition	after the first write only v	
	Adjacent cells	Cross-correlation (cross-talk, short)	i. written j. too	AND/ OR/ complement	

Functional unit	Fault location	Fault	Effect		Functional test
			Write	Read	
Write amplifier	Data line	s-a-v	All cells on this line D: s-a-v		
		cross-talk, short	Parasitic correlation between all cell pairs of the same address belonging to this two lines		
	Enable	s-a-inactive	~ All cells D: s-a-c		
		s-a-active			

<table>
<tr><th rowspan="2">Functional unit</th><th rowspan="2">Fault location</th><th rowspan="2">Fault</th><th colspan="2">Effect</th><th rowspan="2">Functional test</th></tr>
<tr><th>Write</th><th>Read</th></tr>
<tr><td rowspan="6">Address decoder</td><td rowspan="3">Input</td><td>A_i: s-a-1</td><td colspan="2">$A_i = 0 \Rightarrow SEL(A) = O_{A \vee 2^i}$</td><td rowspan="2">Test $\forall A_i$</td></tr>
<tr><td>A_i: s-a-0</td><td colspan="2">$A_i = 1 \Rightarrow SEL(A) = O_{A \wedge \overline{2^i}}$</td></tr>
<tr><td>$A_i - A_j$ short</td><td colspan="2">$A_i \neq A_j \Rightarrow$
$SEL(A) = O_{A \vee (2^i \vee 2^j)}$ or
$SEL(A) = O_{A \wedge \overline{(2^i \vee 2^j)}}$</td><td>Test $\forall A_i, A_j$</td></tr>
<tr><td rowspan="3">Output</td><td>O_s s-a-inactive</td><td colspan="2">All cells of O_s inactive</td><td>Test $\forall O_s$</td></tr>
<tr><td>O_s s-a-active</td><td colspan="2">$SEL(A) = O_s \Rightarrow$ fault free
$SEL(A) \neq O_s \Rightarrow$ W.w.
$SEL(A), O_s$</td><td rowspan="2">Test $\forall O_s, O_t$</td></tr>
<tr><td>$O_s - O_t$ short</td><td colspan="2">$SEL(A) \notin \{O_s, O_t\} \Rightarrow$ fault free
$SEL(A) \in \{O_s, O_t\} \Rightarrow$ W.w.
O_s, O_t</td></tr>
</table>

Test algorithms grouped by run-time complexity

Name	Complexity	Characteristics
Linear	$O(N)$	for all N cells do a selected cell written (1), all adjacent cells read (k)
Quadratic	$O(N^2)$	for all N cells do a selected cell written (1), read all cells (N) f
3/2	$O(N^{3/2})$	“marching” diagonal
Logarithmic	$O(N \times \log(N))$	Partitioning according to address bits
Linear, structure independent	$O(N)$	Permanent faults

Logarithmic test

Init		Address	1		2		3		4		5		6		7		8	
0	0	1	0	0	1	1	1	1	0	0	1	1	1	1	0	0	0	0
0	0		1	1	1	1	0	0	0	0	0	0	1	1	1	1	0	0
		$\log(N)$	0	1	1	1	1	0	0	0	1	0	1	1	0	1	0	0
			0	1	1	1	1	0	0	0	1	0	1	1	0	1	0	0

- **Linear, structure independent memory test**

W0	R0W1	R1W0	R0	R0W1	R1W0	R0
↘	↘	↘	↘	↗	↗	↗

Application example: SN74xx family

- **Modified Hamming Code**

Dimension	3-State	Open-Collector
8	’636	’637
16	’616 ’630	’617 ’631
32	’632 ’634	’633 ’635

- **Example: SN 74 LS 636 8-Bit parallel EDC and ECC circuit**
  - **Modified (13,8) Hamming-code (8 informations- + 5 check bits)**
  - $\bar{c} = (DB_0, \ldots, DB_7, CB_0, \ldots, CB_4)$
  -

$$\Rightarrow \quad G = \begin{bmatrix} 1\,0\,0\,0\,0\,0\,0\,0 & 1\,1\,0\,1\,0 \\ 0\,1\,0\,0\,0\,0\,0\,0 & 1\,0\,1\,1\,0 \\ 0\,0\,1\,0\,0\,0\,0\,0 & 0\,1\,1\,1\,0 \\ 0\,0\,0\,1\,0\,0\,0\,0 & 1\,1\,0\,0\,1 \\ 0\,0\,0\,0\,1\,0\,0\,0 & 1\,0\,1\,0\,1 \\ 0\,0\,0\,0\,0\,1\,0\,0 & 0\,1\,1\,0\,1 \\ 0\,0\,0\,0\,0\,0\,1\,0 & 0\,1\,0\,1\,1 \\ 0\,0\,0\,0\,0\,0\,0\,1 & 0\,0\,1\,1\,1 \end{bmatrix}$$

-

$$\Rightarrow \quad H = \begin{bmatrix} 1&1&0&1&1&0&0&0&1&0&0&0&0 \\ 1&0&1&1&0&1&1&0&0&1&0&0&0 \\ 0&1&1&0&1&1&0&1&0&0&1&0&0 \\ 1&1&1&0&0&0&1&1&0&0&0&1&0 \\ 0&0&0&1&1&1&1&1&0&0&0&0&1 \end{bmatrix}$$

- **Why is this a modified Hamming-code?**

$$\bar{S} = \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} = H \times \bar{c}^T = \begin{bmatrix} 1&1&0&1&1&0&0&0&1&0&0&0&0 \\ 1&0&1&1&0&1&1&0&0&1&0&0&0 \\ 0&1&1&0&1&1&0&1&0&0&1&0&0 \\ 1&1&1&0&0&0&1&1&0&0&0&1&0 \\ 0&0&0&1&1&1&1&1&0&0&0&0&1 \end{bmatrix} \times \bar{c}^T$$

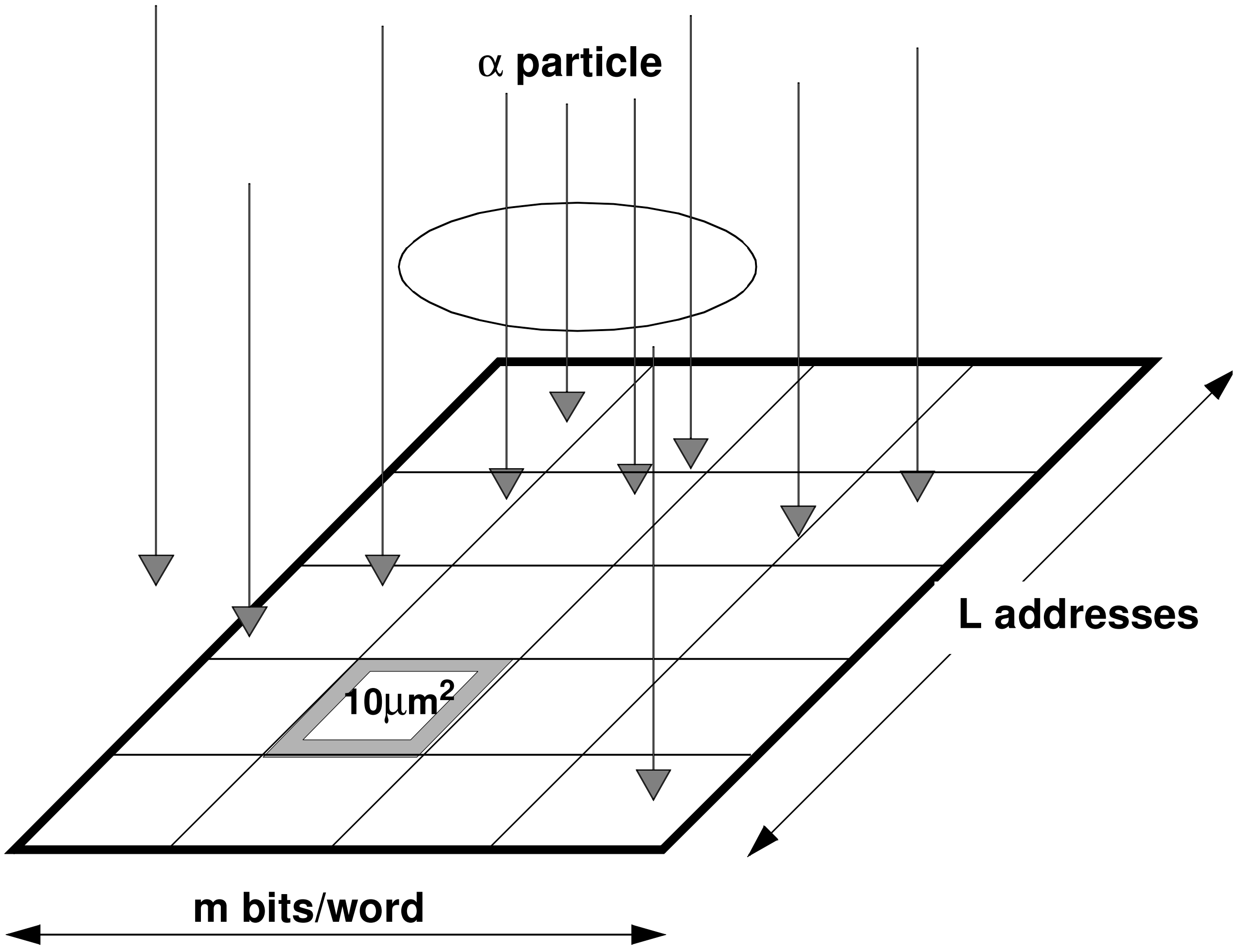
$$\bar{S}^* = \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \\ S_0 + S_1 + S_2 + S_3 + S_4 \end{bmatrix} = \begin{bmatrix} 1&1&0&1&1&0&0&0&1&0&0&0&0 \\ 1&0&1&1&0&1&1&0&0&1&0&0&0 \\ 0&1&1&0&1&1&0&1&0&0&1&0&0 \\ 1&1&1&0&0&0&1&1&0&0&0&1&0 \\ 1&1&1&1&1&1&1&1&1&1&1&1&1 \end{bmatrix} \times \bar{c}^T$$

Architecture

- **Testability: information and check bits can be written separately**

Importance of error correction in semiconductor memories

- **Main origin of transient faults: hits by α particles $\Rightarrow$ bit flips**



Modelling:

- **Particle hits are uncorrelated both in space and time**
  - **Characterized by**

$$\text{Intensity} = \frac{Hits}{Area \times Time}$$

  - **Each hit causes a fault**
  - **Hits form a Poisson process**
- **Probabilities**
  - **The surface S_0 will be hit i times during a time period t:**

$$P^{i}(\ ) = \frac{(\quad)^{i}}{i!}e^{-\lambda S_0 t}$$ **, where**

λ**: Intensity of the particles**
t: Length of the time period
S_0**:"Effective" area**

  - **E.g. no-hits($i = 0$):** $P^{0}(\ ) = e^{-\lambda S_0 t}$

Impact of the EDC

- **Memory without ECC:**
  - **Reliable until all bits in all words remain non-hit**
- **Memory with word ECC:**
  - **Increased word length from m to M > m (check bits)**
  - **Reliable until no word has incorrigible faults**

  - **E.g. single bit ECC** $P(\quad) = [\quad]^{M} + \left(\quad \right) P^{1}(\quad)[\quad]^{M-1}$

$$= e^{-\lambda M S_0 t} + M \cdot \quad \cdot \quad \cdot (\quad)^{M-1} \qquad = e^{-M\lambda S_0 t}(\quad)$$

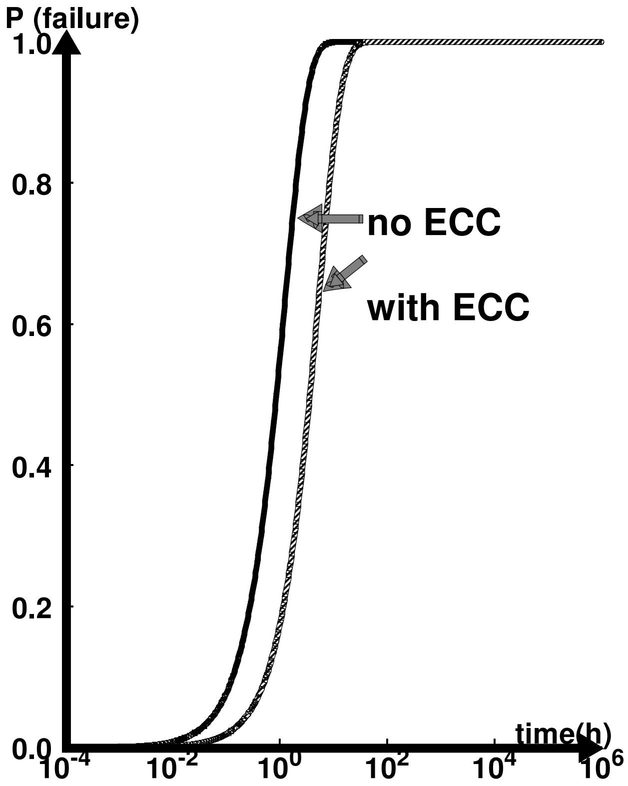
  - **Probability of all words being correct:** $P_{\mathrm{RAM},\,ECC} = \prod_{a=1} P(\quad)$

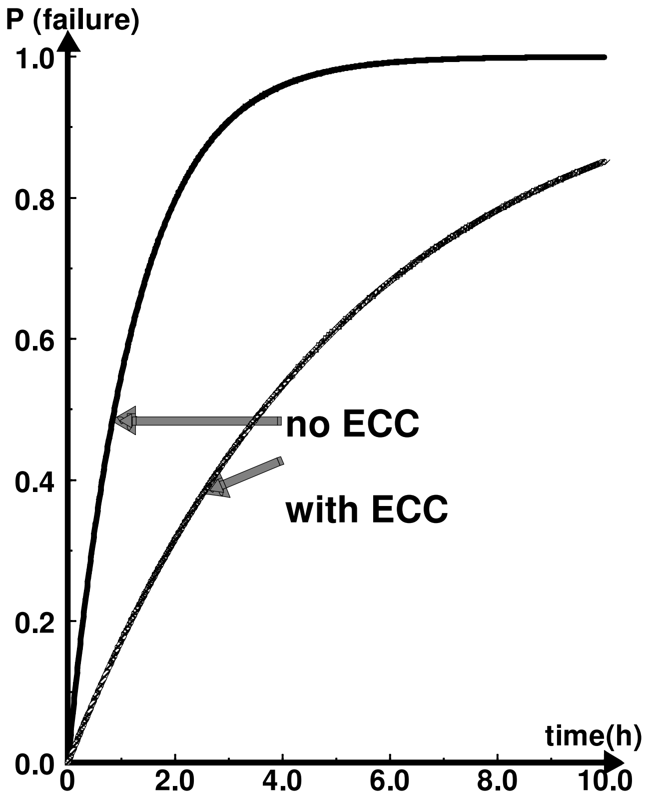
$$= \left[e^{-M\lambda S_0 t}(\quad) \right]^{L} = e^{-ML\lambda S_0 t}(\quad)^{L} \quad .$$

  - **If** $m \approx M$**, then** $P_{\mathrm{RAM},\,ECC} = P_{RAM} \times (1 + M\lambda S_0 t)^{L}$

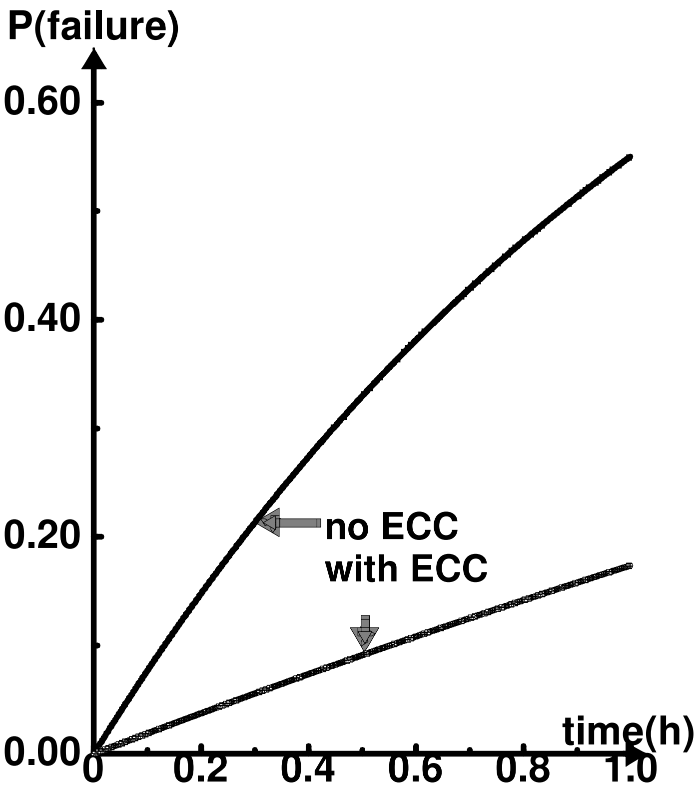
Comparison of the failure probabilities

- Parameters: (1979)
  λ: $1 cm^{-2} h^{-1}$ Intensity der particles
  m: 8Word width of the memory without ECC
  M: 11 Word width of the memory with ECC
  L: 1MegabitLength of the memory
  s_0: $10\ \mu m^2$ Effective area





Initial phase



Consequences:

- The same model can be used, if failure means
  "uncorrectable or undetected"
  "undetected" (only EDC)
- As greater the error correction capability of the ECC used, as longer the MTBF
- EDC increases the reliability, but
  - After a certain period, failures are unavoidable due to accumulated hits⇒
  - read-check-(correct-rewrite) operations are needed for every word in a regular period (memory scrubbing)
  - memory scrubbing can be combined with refresh cycles by dynamic memories
- Permanent faults in the memory cells are detected and corrected
- Effect of data transfer (address, data) faults?
  - All words are correct, but at a wrong location⇒
    - undetected by word EDC
    - Both words are faulty
  - Redundance needed to include the address into the EDC?
    E.g. 1 MByte memory (20 Address bits), modified Hamming code
    - Separate check bits for addresses: 7 addressed- + 5 date check bits= 12 check bits
    - Integrated into a single date: 28 bits to check ⇒ 7 check bits