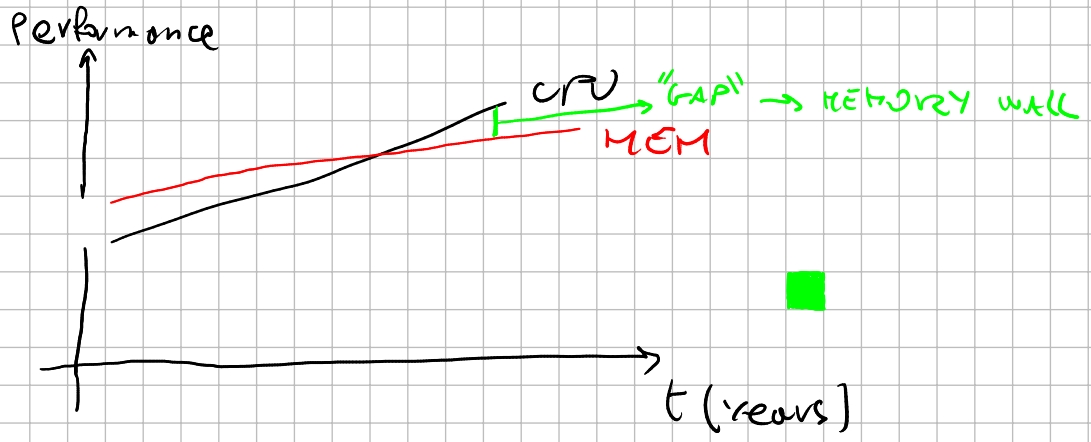
MEMORY WALL



DYNAMIC MEMORY

DYNAMIC	STATIC
CAPACITOR	D.F.F. → TRANSISTORS
- AREA (COST) +	- AREA (COST) -
- DELAY -	- DELAY + (fast)
- SCALABILITY +	- SCALABILITY +
- RESILIENCE -	- RESILIENCE +
- ENERGY (cost) +	- ENERGY (cost) -

In the bottle there ~~is~~ WAS some liquid

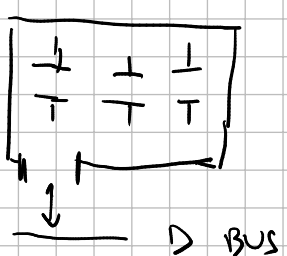
Reading destroys the previously stored information

Restore previously stored info (based on what has been read)

Reading in dynamic is slower than in static: the read in dynamic requires a rewrite

LEAKAGE

Restoring info is necessary → internal refreshing (maintenance)

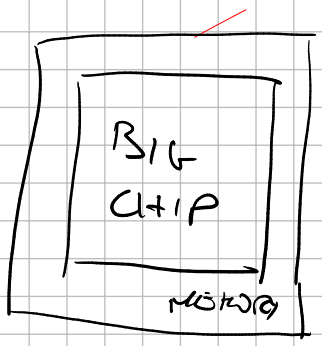


REFRESH

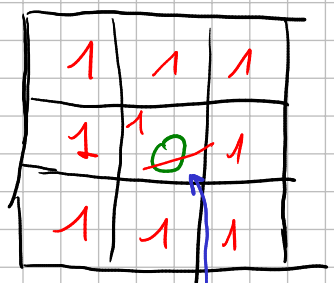
(INTERNAL) READING (& WRITING)

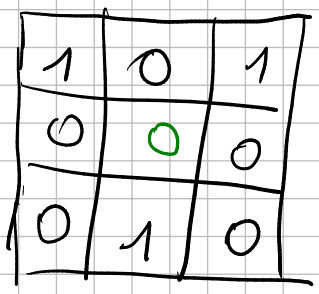
BETTER

error management



← WORSE



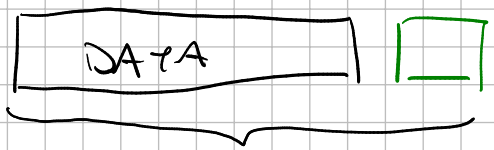


"SOFT" ERROR

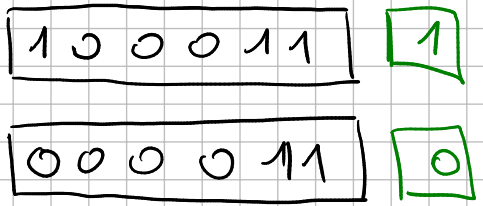
PARITY BIT CODE [ODD / EVEN]

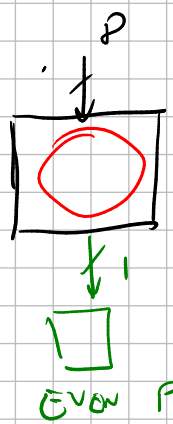
2 standards none is better than the other

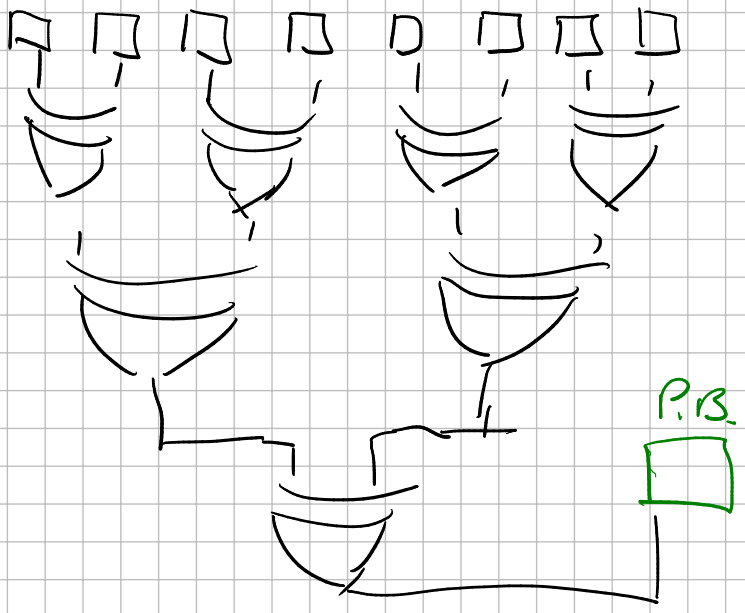
Even



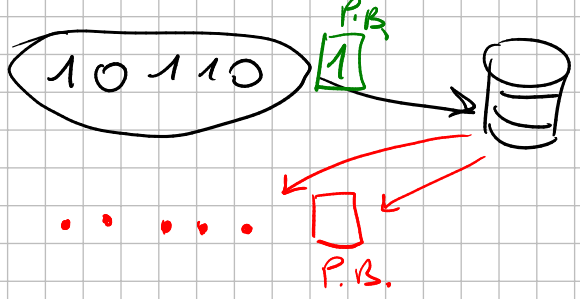
of "1" should be EVEN





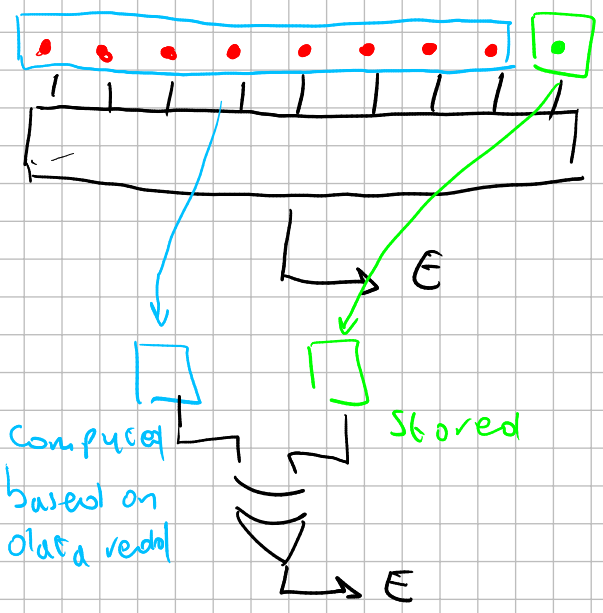


PARITY BIT = SINGLE ERROR DETECTION CODE



11011 1 α
10110 0 β
11010 1 γ

(α) → Parity bit not compliant! → ERROR!
(β) → P. B. not compliant! → ERROR!
(γ) → P. B. Compliant



E = "FALSE" → no error
E = "TRUE" → error

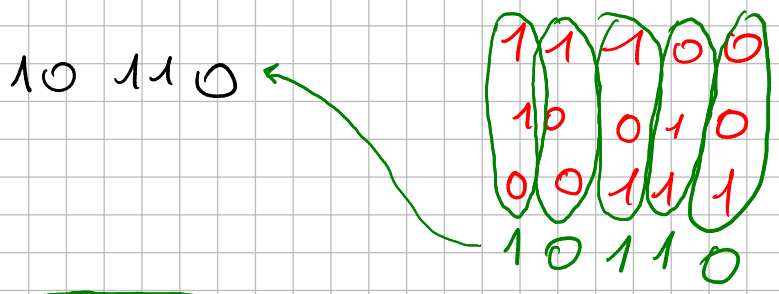
- ADDITIONAL STORAGE:

  10 BITS DATA 1 P.BIT → $+\frac{1}{10}$ of additional storage

  4 BITS DATA 1 P.BIT → $+\frac{1}{4}$ of additional storage

- MORE HARDWARE TO COMPUTE & CHECK P.B.
- MORE TIME TO COMPUTE/CHECK P.B.

"MAJORITY VOTING"



USERDATA	MAJORITY VOTING

MULTIPLE ERROR CORRECTION CODE

Written

1 1 0 0 0 1	1
0 1 1 0 0 0	0
0 0 0 1 1 0	0
1 1 1 0 0 0	1
0 1 0 1 1 1	

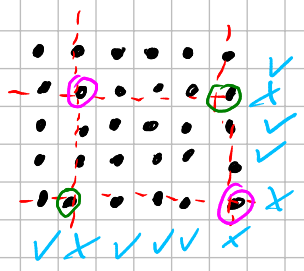
Vertical & horizontal

P.B. → Protects a block of data

Read

1 1 0 0 0 1	1	✓
0 1 1 0 1 0	0	✗
0 0 0 1 1 0	0	✓
1 1 1 0 0 0	1	✓
0 1 0 1 1 1		
✓ ✓ ✓ ✗ ✓		

Single error correction code



Multiple error detection
but not correction

↓

ERROR CORRECTING CODE MEMORIES (ECC)

A	a_j
B	b_j
C	c_j
~~D~~ D'	d'_j
E	e_j
F	f_j
V	$V_j = a_j \oplus b_j \oplus c_j \oplus d_j \oplus e_j \oplus f_j$

$V_j \oplus d_j = a_j \oplus b_j \oplus c_j \oplus e_j \oplus f_j$

$V'_j = a_j \oplus b_j \oplus c_j \oplus d'_j \oplus e_j \oplus f_j$

$V'_j = V_j \oplus d_j \oplus d'_j$

Cyclic
Redundancy
Code