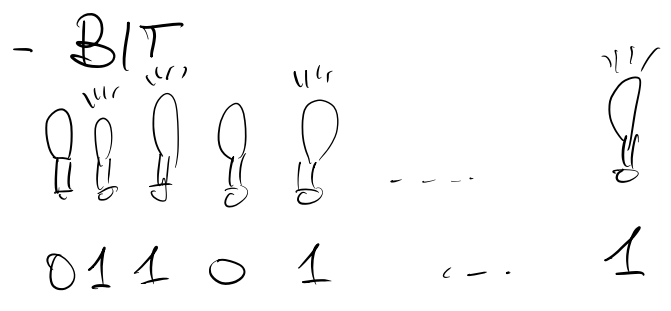


MEMORIA DIGITALE



1 BYTE =
8 BIT

INDIRIZZO

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |
| | | | | | | | | |

b
B

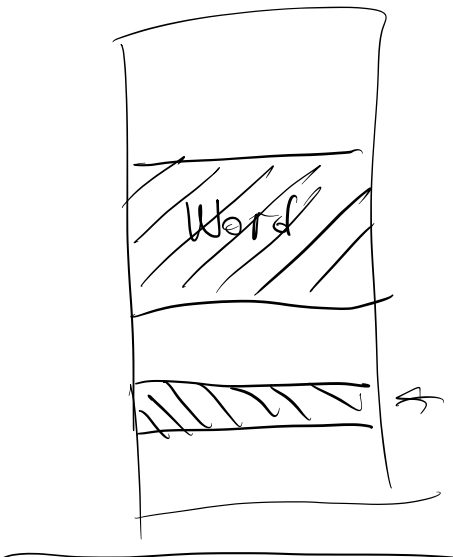
1 9 9 9 9 9 9 9 9 9 9

1 BIT \rightarrow 2 values possible
1 BYTE $\rightarrow 2^8 = 256$ values

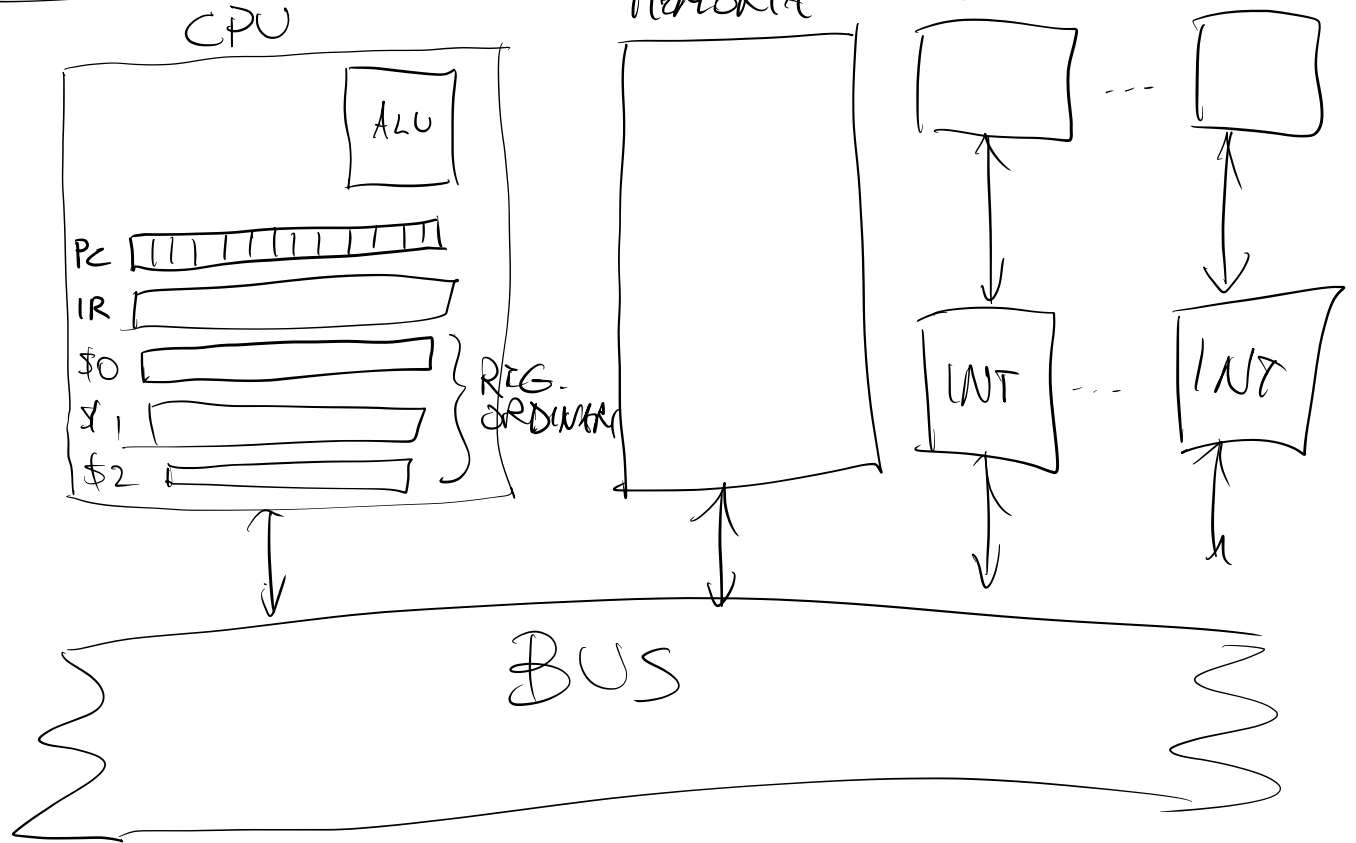
0000 0000 = 0
0000 0001 = 1
0000 0010 = 2
0000 0011 = 3
⋮

1111 1111 = 255

KiB = 2^{10} B (1024)
MiB = 2^{20} B (1,048,576)
GiB = 2^{30} B (1,073,741,824)
TiB = 2^{40} B (1,099,511,627,776)

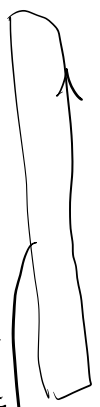
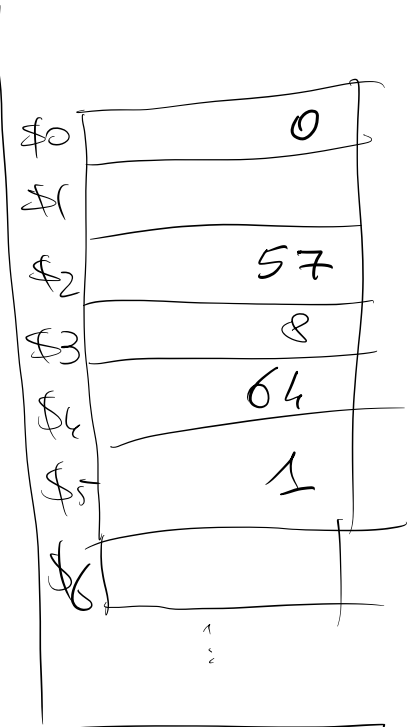


MACCHINA DI VON NEUMANN

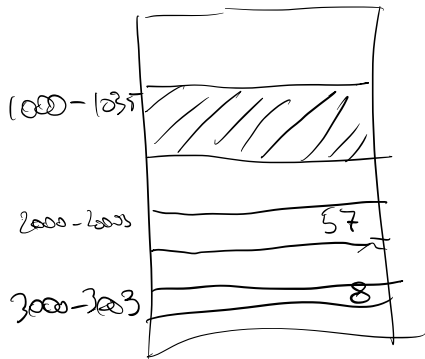


- FASI —
- 1) FETCH
 - 2) DECODE
 - 3) EXECUTE

1000: lw \$2, 2000(\$0)
 1004: add \$3, \$0, \$0
 1008: add \$4, \$3, \$0
 1012: mult \$4, \$4, \$4
 1016: slt \$5, \$4, \$2
 1020: beq \$5, \$0, +8
 1024: addi \$3, \$3, +1
 1028: j 1008
 1032: sw \$3, 3000(\$0)



MEM



PC



IR



slt

set if less than

slt \$5, \$4, \$2

$sl: \$5 \leftarrow 1$
 $no: \$5 \leftarrow 0$

beq

branch if equal

beq \$5, \$0, +8

\$5 = \$0?
 si: +8 al PC
 no: niente

j

jump

j 1008

"metti 1008 nel PC"

if

\$4 ≥ \$2
 a \$032
 Vali



LINGUAGGI AD
ALTO LIVELLO

- COBOL
- FORTRAN
- C
- GO



LINGUAGGI
IMPERATIVI

- Lisp
- ML
- Haskell



LINGUAGGI
FUNZIONALI