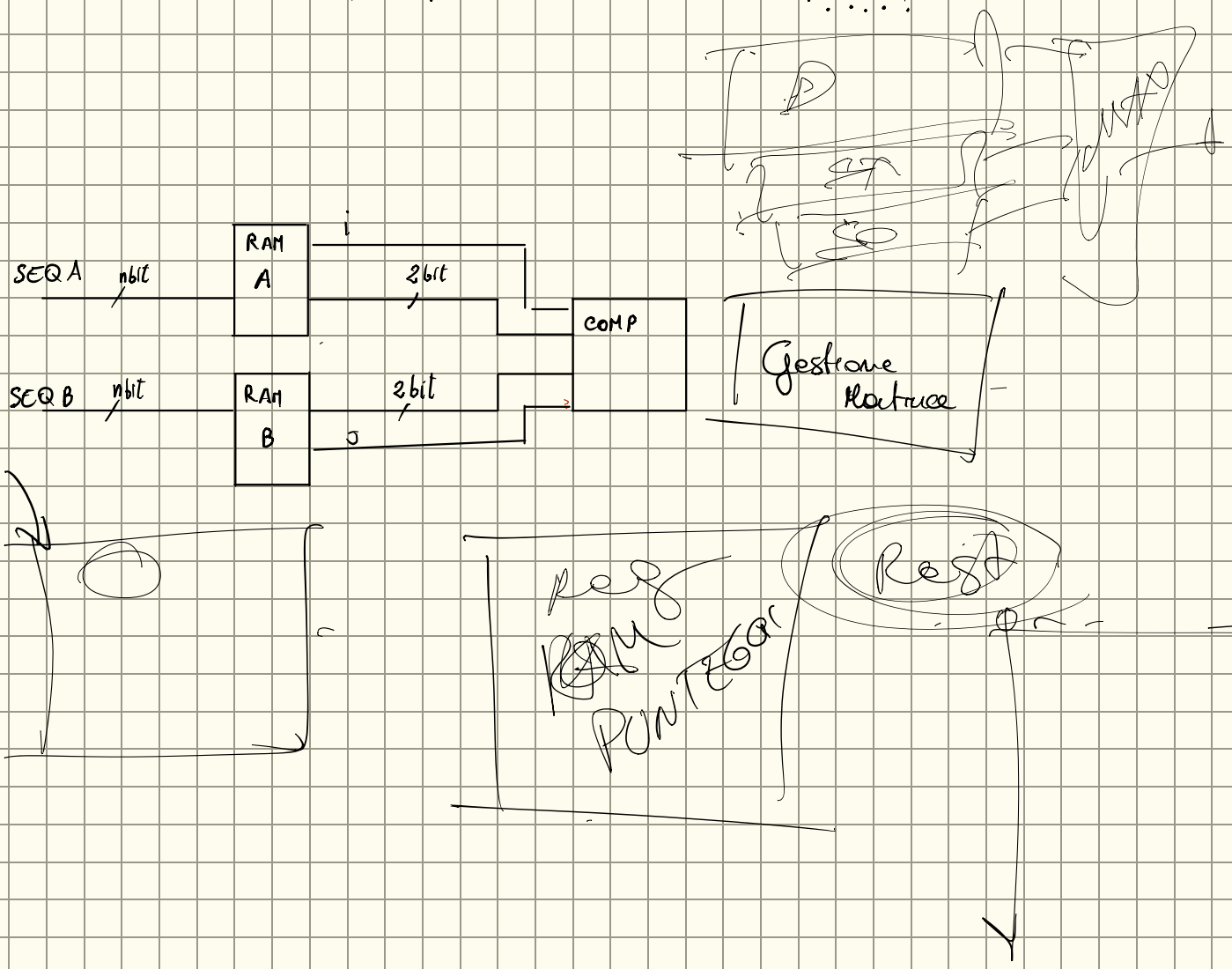
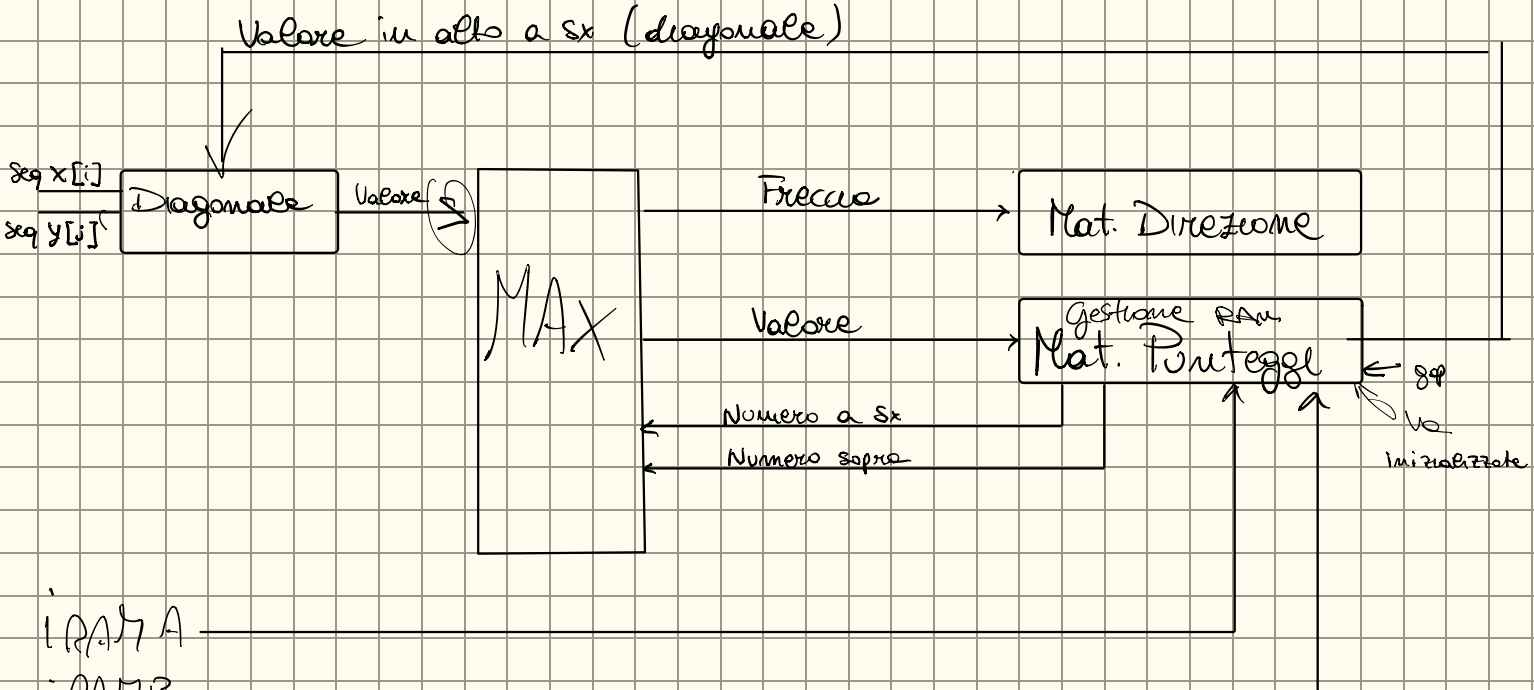
PROGETTO ESD

LA SEQUENZA VA A BLOCCO?

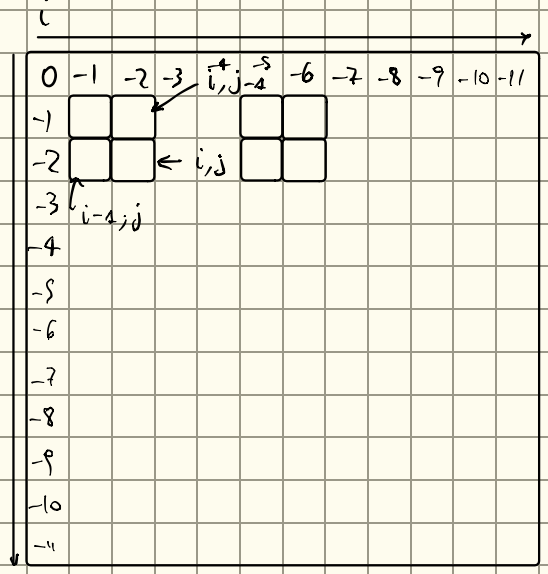
SI PUÓ SPEZZETTARE CON 2 RAM!!!!!

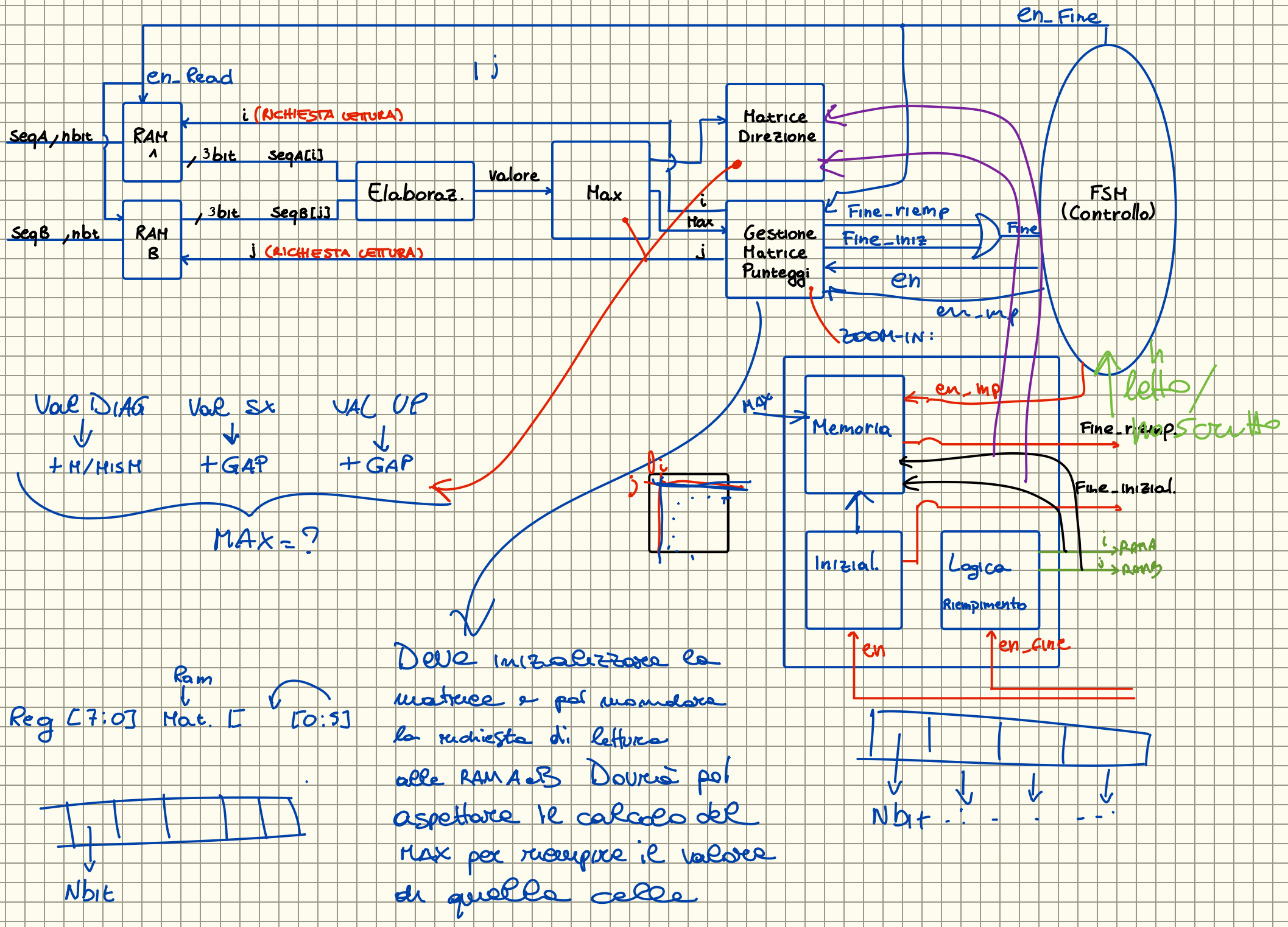


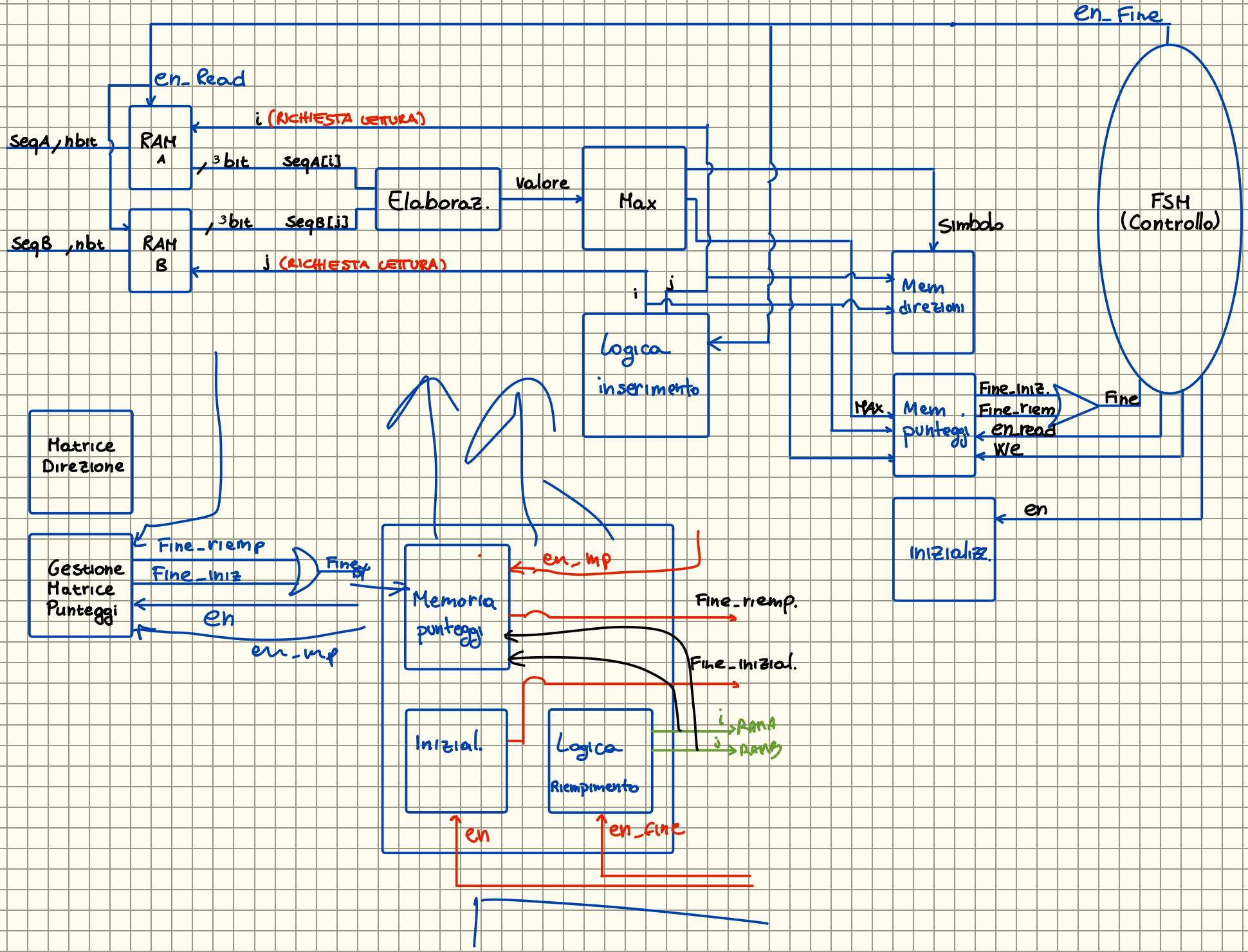


128

```
for (i = 0; i < N; i++) {
  for (j = i; j < N; j++)
  {
    mat[i][j] = max[i][j]
    mat[j][i] = ~~max~~
  }
}
```




en_Fine
en_Read
i j
SeqA, nbit
RAM A
i (RICHIESTA LETTURA)
3 bit
SeqA[i]
Elaboraz.
Valore
Max
Matrice Direzione
SeqB, nbt
RAM B
3 bit
SeqB[j]
j (RICHIESTA LETTURA)
i
Max
j
Gestione Matrice Punteggi
Fine_riemp
Fine_Iniz
Fine
en
en_mp
ZOOM-IN:
FSM (Controllo)
Val DIAG
+M/MISM
Val Sx
+GAP
VAL UP
+GAP
MAX=?
MAX
Memoria
en_mp
Fine_riemp
Fine_Inizial
i RAMA
j RAMB
Inizial.
Logica Riempimento
en
en_fine
letto/scritto
Ram
Reg [7:0]
Mat. [ ]
[0:5]
Nbit
DEVE inizializzare la matrice e poi mandare la richiesta di lettura alle RAM A e B. Dovrà poi aspettare il calcolo del MAX per recuperare il valore di quella cella
Nbit


en_Fine
en_Read
RAM A
SeqA, nbit
i (RICHIESTA LETTURA)
3 bit
SeqA[i]
RAM B
SeqB, nbt
3 bit
SeqB[j]
j (RICHIESTA LETTURA)
Elaboraz.
Valore
Max
Simbolo
FSM (Controllo)
i
j
Logica inserimento
Mem direzioni
Mem punteggi
MAX
Fine_iniz.
Fine_riem
en_read
We
Fine
Matrice Direzione
Gestione Matrice Punteggi
Fine_riemp
Fine_iniz
Fine
en
en_mp
Memoria punteggi
Fine_riemp.
Fine_inizial.
Inizial.
Logica Riempimento
i RAM
j RAM
en
en_fine
Inizializz.
en

RAM PUNTEGGI

- leggere
- scrivere
- ADDRA
- ADDRB
- dataA
- dataB

0 → 0
1 e 1.5 → -1
2 2.5 → -2
3 3.5 → -3
4 4.5 → -4

NXN

IF(rst) {
.
.
.
.
}

RAM 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
RAM 0 -1 -2 -3 -4 -1 -2 -3 -4

5X5

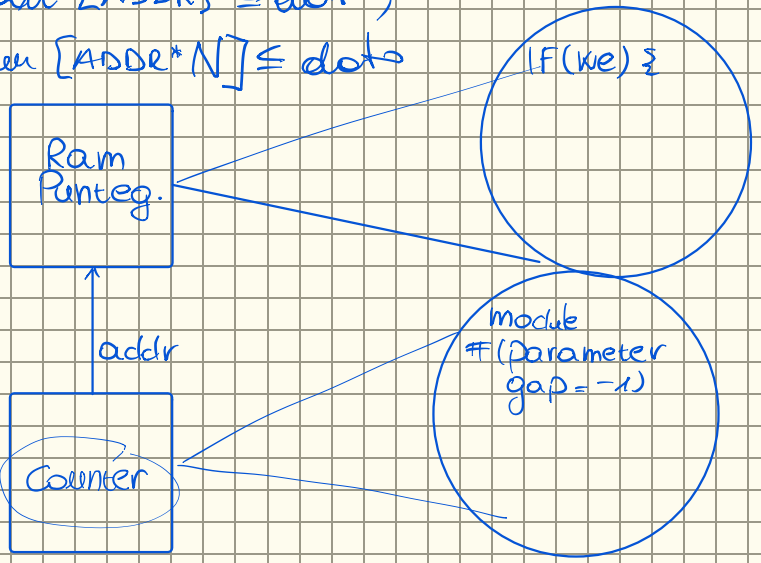
128 celle

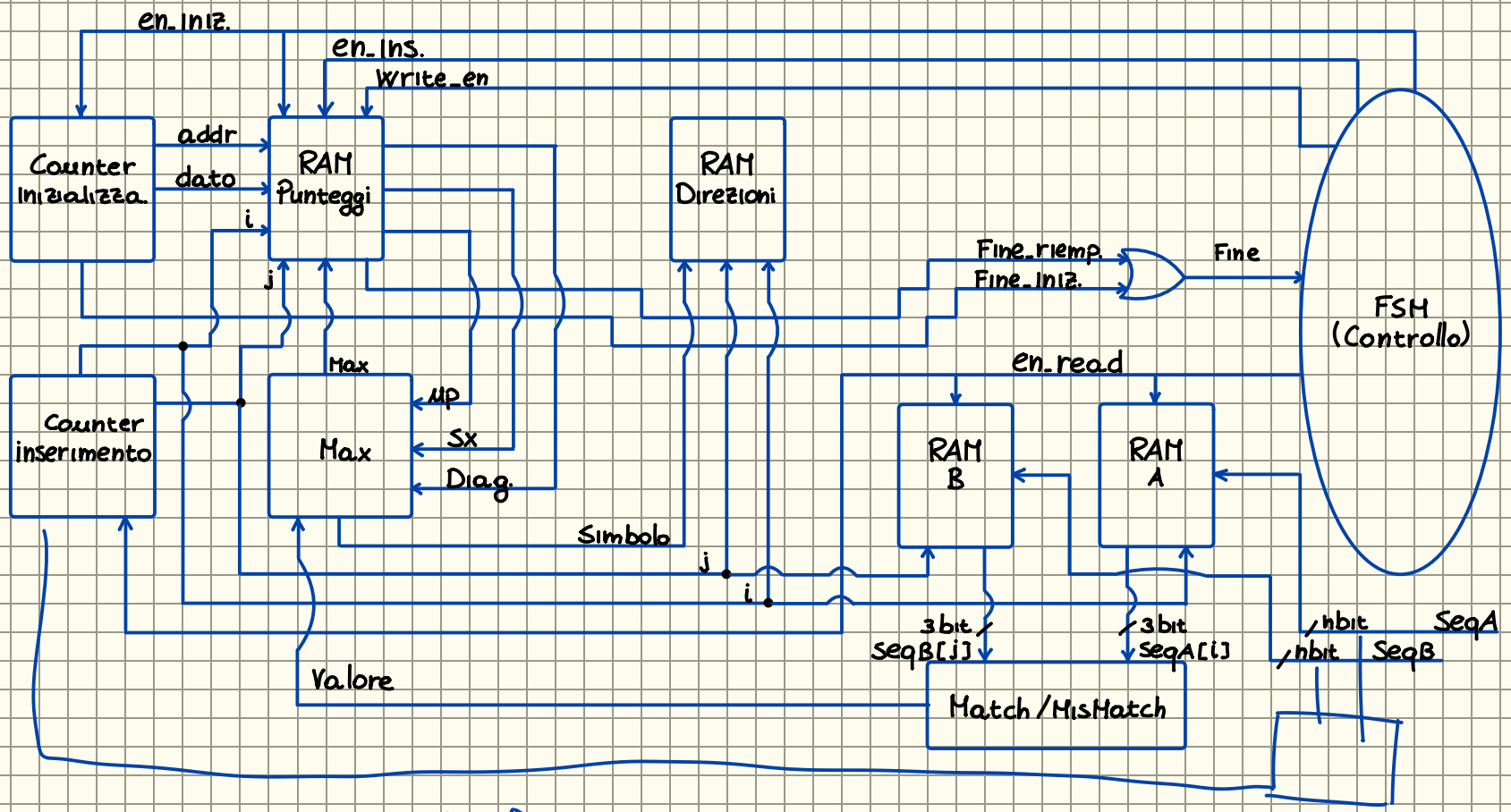
A G C T A C . . . . . X X X X X X X X X X X X

RAM 0 -1 -2 -3 -4 . . . . -128 -1 . . . -2 . . . . . -128 . . .

5

ram[ADDR] = data;
ram[ADDR*N] = data





5

ADDR ← a cui dobbiamo scrivere

log (h) =

1001 → 4 bit

4 → 8 → 15

if(en_ins)

- if(en_leMura)

  d1_Sx = ram[(i-1)+N*j]

  d1_Up = ram[i+N*(j-1)]

  d2_d = ram[(i-1)+N*(j-1)]

  } — VANNO MANDATI A MAX

10010101

11000000

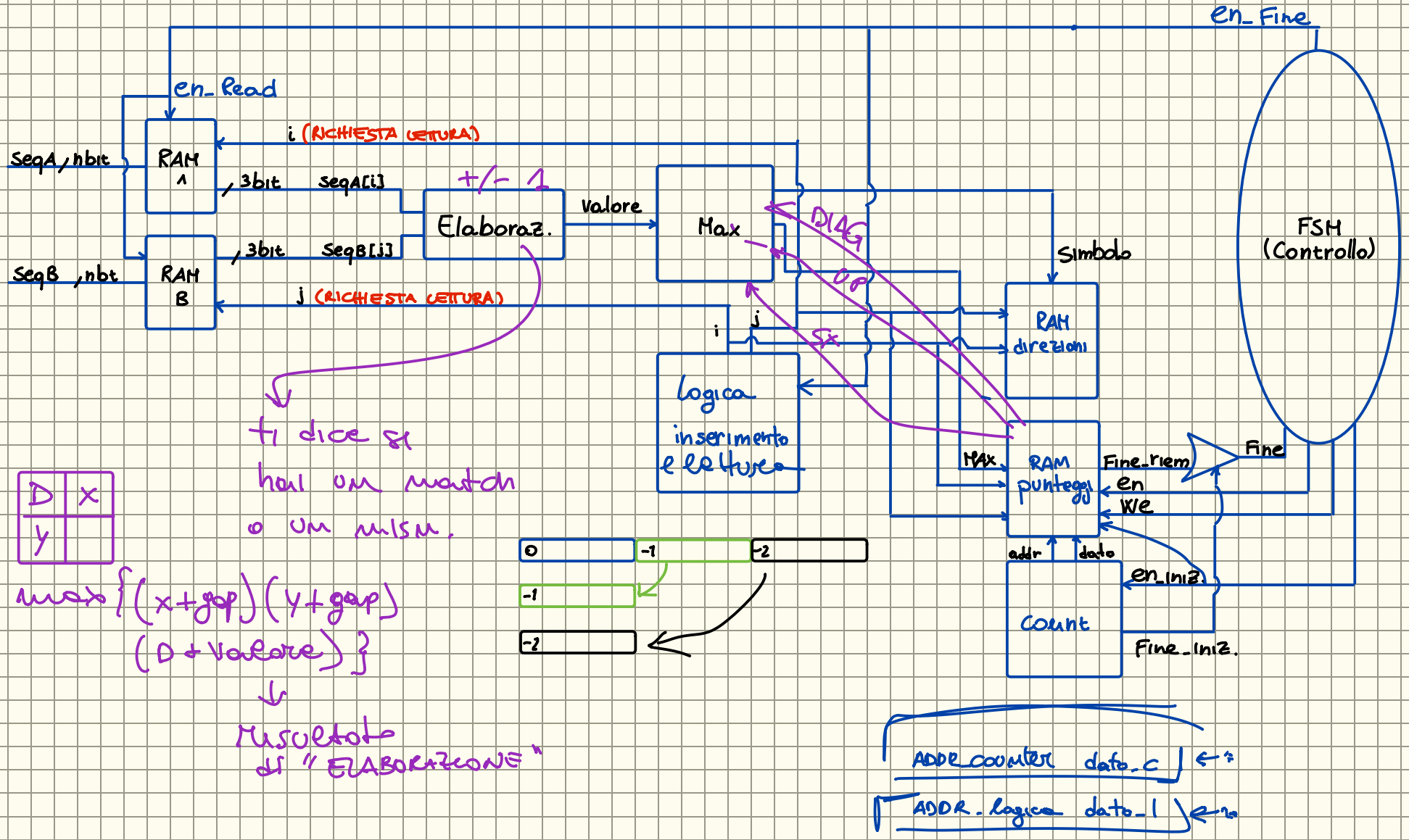
if(en_scrittura)

ram[i+N*j] = data

0 → 128

-128 → 0

256



ti dice se
hai un match
o un mism.

D	x
y	

$\max\{(x+gap)(y+gap)(D+Valore)\}$

↓

Multiplexer
di "ELABORAZIONE"

$$i*(j+1)+(N*j)-(i*j) = \cancel{i*j} + i*1 + N*j - \cancel{i*j} = i + N*j$$

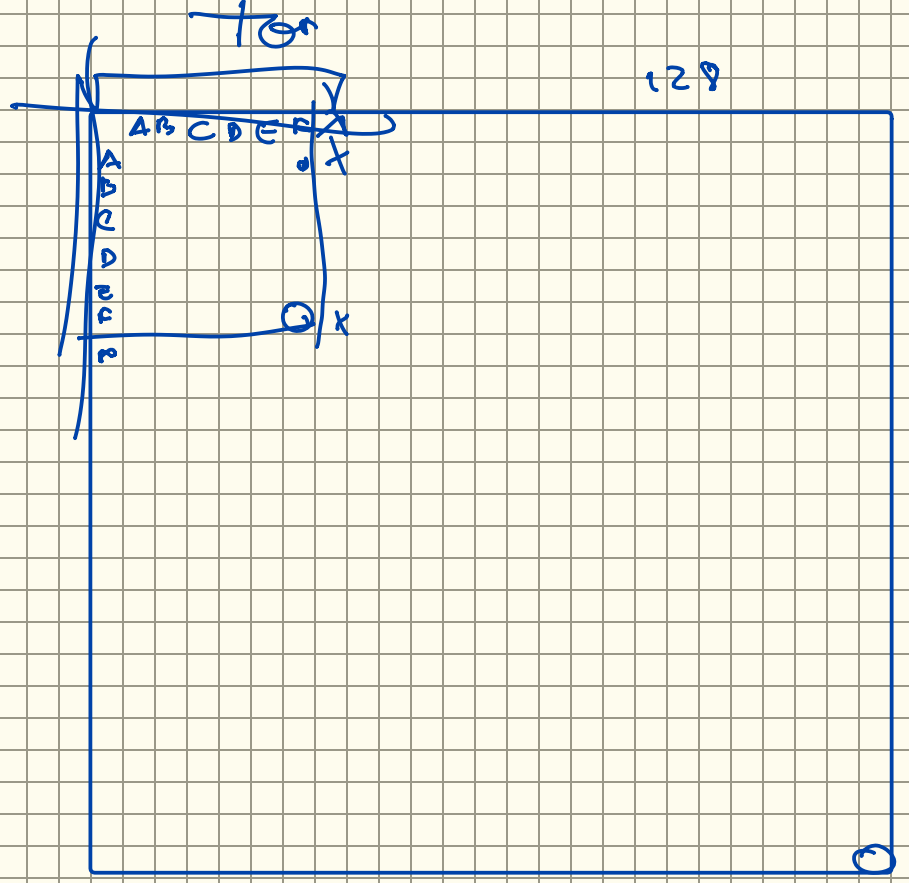
RAM PUNTEGGI

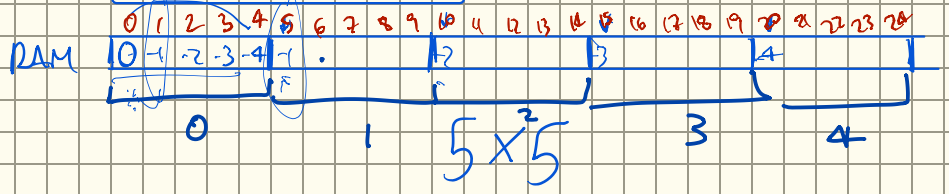
- leggere
- scrivere
- ADDRA
- ADDRB
- dataA
- dataB

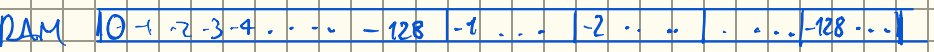
0 → 0
1 e 1.5 → -1
2 2.5 → -2
3 3.5 → -3
4 4.5 → -4

NXN

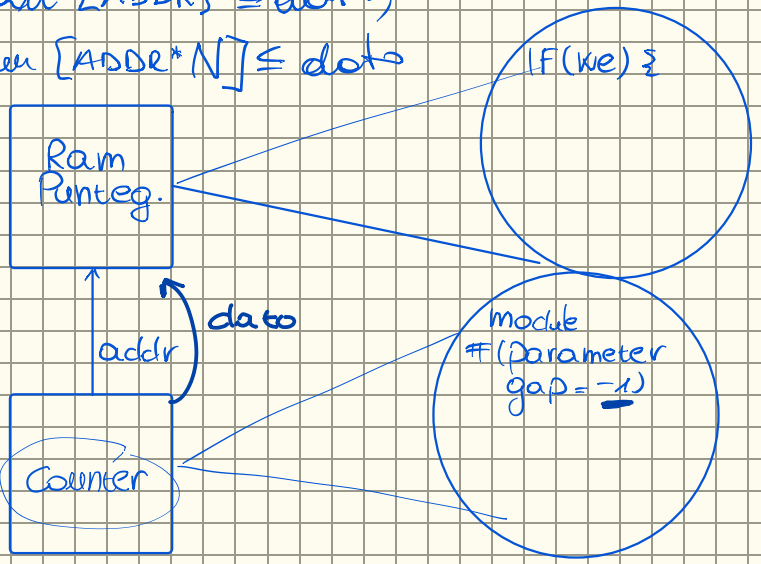
IF(rst) {
. . . . .
}



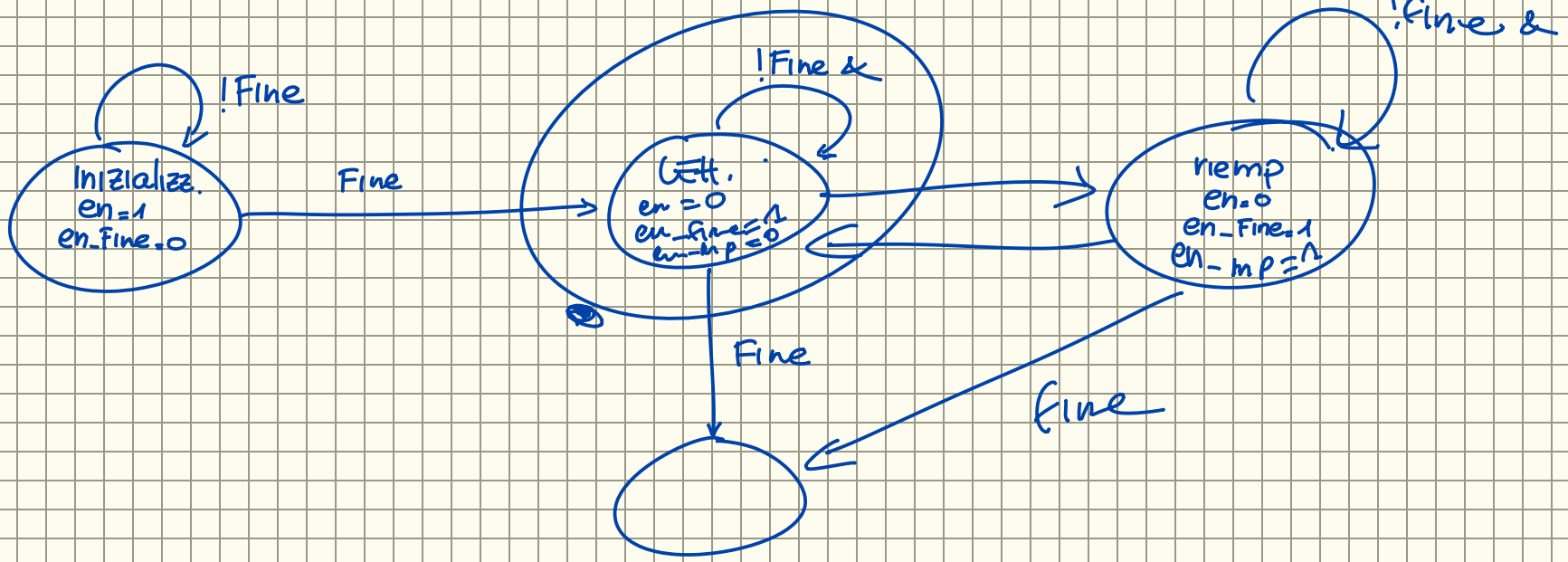




ram[ADDR] ≤ dato;
ram[ADDR*N] ≤ dato



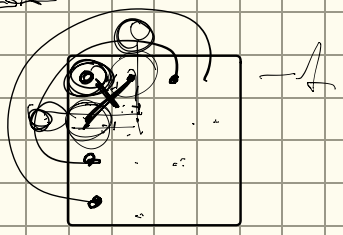

```
IF(en_iniz) {
  ram[addr] ≤ dato
  ram[addr * N] ≤ dato
}

  dato = 0 → nel registro
, always @ (. . . . )
  dato_nxt = dato + gap;
```

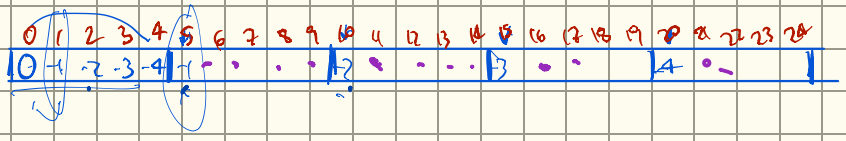


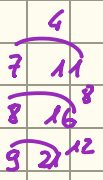
- Gestire la Matrice con una RAM da 16.384 locazioni?
- Le RAM delle FPGA hanno possibilità di scelta rispetto a quante porte utilizzare.

128

```
for (i = 0; i < N; i++) {
    for (j = i; j < N; j++)
    {
        mat[i][j] = max[i]
        mat[j][i] = max
    }
}
```







0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31	32	33	34	35

$(N-1) \cdot X$

$N-1$

0	-1	-2	-3	-4
5 -1	6	7	8	9
10 -2	11	12	13	14
15 -3	16	17	18	19
20 -4	21	22	23	24

5x5