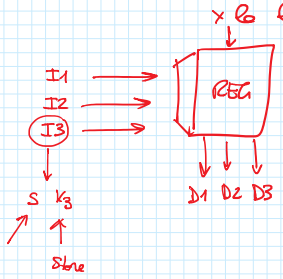
store$_0$. "write" → OP, REG[IR.RA] + REG[IR.RB] → IND, REG[IR.RC] → DATAOUT, set RDY$_m$, store$_1$

store$_1$. (ACK$_m$, OR(ESITO), INT = 0 - -) nop, store$_1$
(= 1 0 0) IC + 1 → IC, ch$_0$.
(= 1 0 1) IC+1 → IC, tratt interruzioni
(= 1 1 -) .... tratt eccezioni

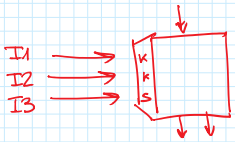
STORE R_{base}, R_{ind}, R_{dato}
REG[dato] → M[REG[base] + R[indice]]

3 letture ⇒ MREG deve avere 3 k × 6 letture



add$_0$. (INT = 0) REG[IR.RA] + REG[IR.RB] ⇒ REG[IR.RC], ... ch$_0$

MREG deve avere 2K × 6 letture
1S × 6 scritture



store$_0$. REG[IR.RC] → DATAOUT, store1
store$_1$. REG[IR.RA] + REG[IR.RB] → IND, "write" → OP, set RDY$_m$, store$_2$ } store$_0$ 2T + te
store$_2$. (ACK$_m$, OR(ESITO), INT = 0 - -) nop, store$_2$ ⟵ store$_1$
(= 1 0 0) K+1 → IC, ch$_0$
reset ACK$_m$
....

CPU M
T T_M

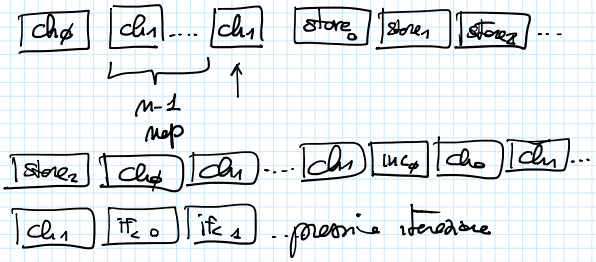
```
for (i=0; i<N; i++)
    X[i] = 0;
```

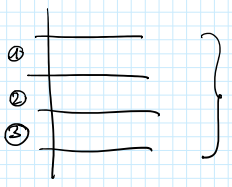
N = 1024

IC →

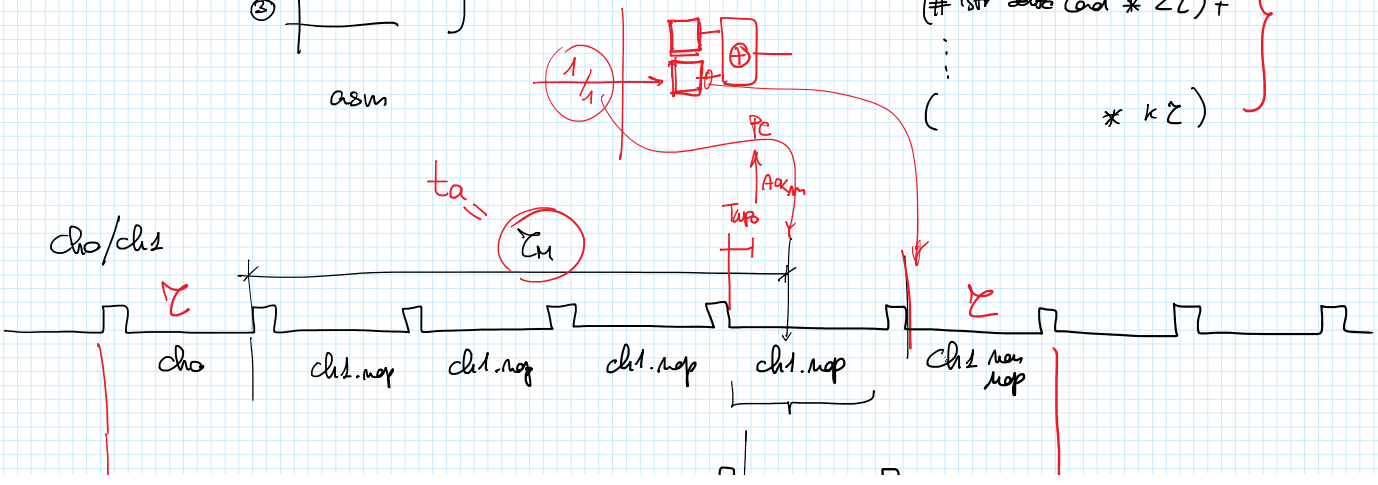
```
loop:  STORE  Rbase x , Ri ,  R0     // X[i]=0;
       INC    Ri
       IF<    Ri , RN , loop
cont:  ....                  ↑ -2
```

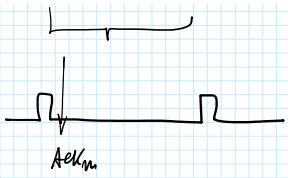
Interpreta fw





#istruzioni ASM ($T_{ch0} + T_{ch1}$) + (# istr Alog * T) + (# istr salto cond * 2T) + ... (* kT)





$T_{ch0} + T_{ch1} = 2\tau + t_a$

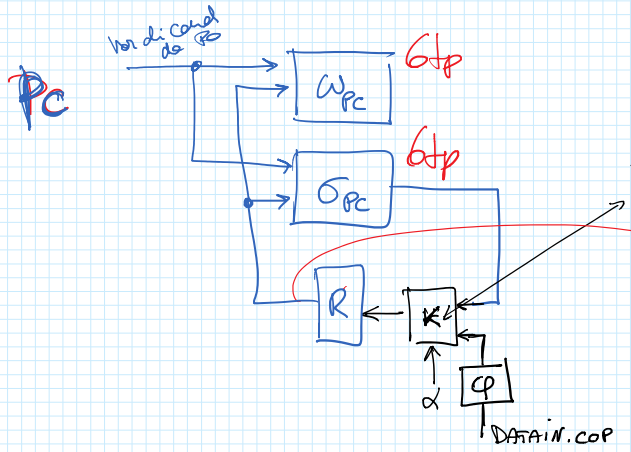
$\rightarrow T_{exec\ op\ AL} = \tau$

$\rightarrow T_{exec\ salto\ cond} = 2\tau$

$T_{exec\ load} = 2\tau + t_a$

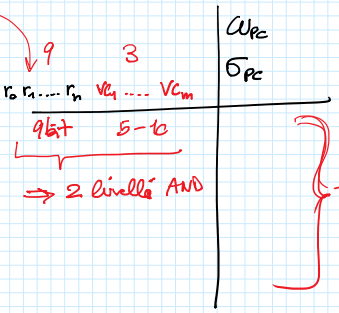
$T_{exec\ store} = 3\tau + t_a$

$T_{exec\ salti} = \tau$



ch_1 ( ) ... DATAIN.COD → RC

$K \begin{cases} R \neq ch_1 & \sigma_{PC} \rightarrow R \\ R == ch_1 & DATAIN.COP \rightarrow R \end{cases}$



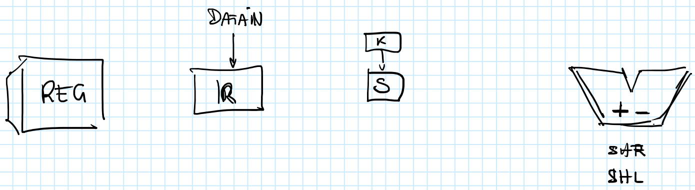
$\lceil \log_8 2^{11} \rceil$

$\frac{\log_2 2^{11}}{\log_2 2^3} = \frac{11}{3}$

2^{11} max "1"

PO

mercoledì 9 novembre 2016 15:23



} ALU "semplici" {
× arith log
× (IC+1) (IC+offset)
× indirizzi memoria
R[base]+R[indice]

moltiplicazione

mul∅. (INT=0) REG[IR.RA] * REG[IR.RB] → REG[IR.RC], IC+1 → IC, clk∅

mul∅. → 5∅T

```
      B        A
    1011  ×  101
    ----
    1011
   0000 -
  1010 -
  ------
```

	R	B	A
R=0	1011	10110	10
	1011+	101100	1
	101100		
	110111		

R=0
for (i=0 ; i< #bit ; i++) {
  if (bit meno significativo di A==0) {
    SHR (A,1)
    SHL (B,1)
  } else {
    R = R + B;
    SHR (A,1)
    SHL (B,1);
  }
}

VALUTAZIONE delle PRESTAZIONI

mercoledì 9 novembre 2016 15:39

$$T = \tau \cdot \sum_i (p_i \cdot k_i)$$

p_{ADD}
p_{SUB}
p_{INC}
⋮
} 1τ

$p_{ADD} \cdot 1 + p_{SUB} \cdot 1 - \dots$

$1(p_{ADD} + p_{SUB} + \dots)$

IF — salto cond. — then — cont:

if then else — salto — then — salto — cont — else — cont

ARITH LOG	40%	CORTE
	10%	LUNGHE
	30%	LOAD/STORE
	20%	SALTO

Porta log corta

$p_{STORE} \cdot 3$

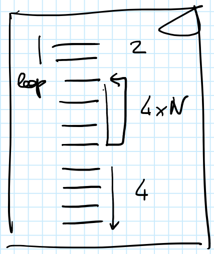
$p_{LOAD} \cdot 2$

$p_{salti\ cond} \cdot 2$

$p_{salto} \cdot 1$

```
T f( T ) {
  for(   ) {
    if( ) {  } else {  } }
  }
}
```

⟹



.asm

10 istruzioni

$$T = \frac{8}{5}\tau$$

$$10 \times \frac{8}{5}\tau = 16\tau$$

?

MIX

stessa macchina ASM

A_1 A_2

D-RISC

stessa classe di prog
x tutte e due

BENCHMARK

SUITE BENCHMARK

A_1 A_2

D-RISC$_1$ D-RISC$_2$

{PROG}

CPI

clock per instruction