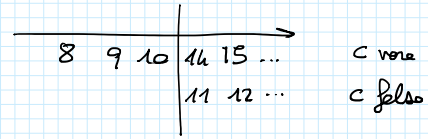
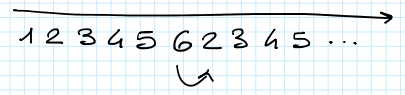
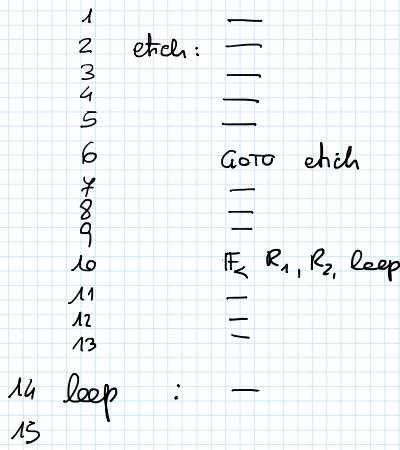


salto
 ↳ condizionali
 ↳ incondizionali

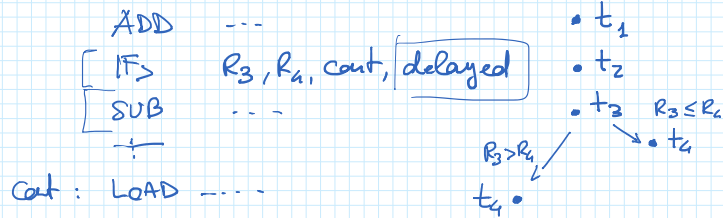
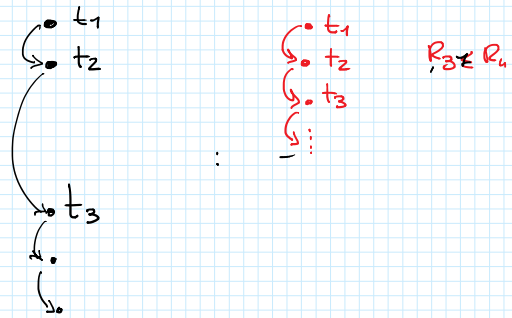


SALTO RITARDATO

salto , delayed
 (, delayed (n)

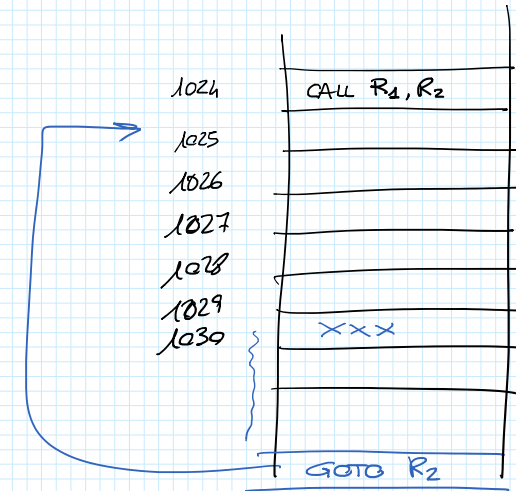
```

ADD R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub>
IF<sub>Z</sub> R<sub>3</sub>, R<sub>4</sub>, cont
SUB R<sub>5</sub>, R<sub>6</sub>, R<sub>7</sub>
⋮
cont: LOAD R<sub>7</sub>, R<sub>8</sub>, R<sub>9</sub>
⋮
    
```



```
CALL Rf, Rret
GOTO R;
```

$IC \leftarrow Rf, IC+1 \rightarrow Rret$



R1	1030
R2	0

IC	1024
----	------

R1	1030
R2	1025

IC	1030
----	------

param in write
param integer

$$y = f(x)$$

$$y = x^2 + 3x - 1$$

```
int f(int *x) { int xx = (*x); ...
int f(int x) { int y = x*x + 3*x - 1; return (y); }
```

```
int main ( ) {
...
for (int i = 0; i < N; i++) {
    a[i] = f(i);
}
```

MODALITÀ di PASSAGGIO dei PARAMETRI

PER VALORE

PER RIFERIMENTO

COME

PARAMETRI nei REGISTRI

PARAMETRI in MEMORIA

$f(x)$ x : valore in un registro

```
ADD ..., ..., R17
CALL R1, R2
→ STORE Rnext, R1, R18
```

y : valore in registro (R18)

```
f: MUL R17, R17, R18
MUL R17, #3, R19
ADD R18, R19, R18
SUB R18, #-1, R18
...
GOTO R2
```

$$R18 = x^2$$

$$R19 = 3x$$

$$R18 = 3x + x^2$$

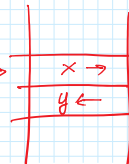
$$R18 = x^2 + 3x - 1$$

x : passato in memoria x valore

```
STORE RbaseParam, #0, ...
CALL R1, R2
```

y : passato in memoria x valore

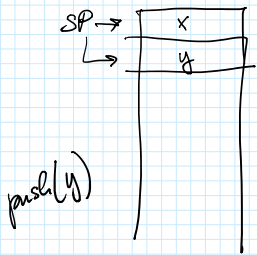
```
f: LOAD RbaseParam, #0, R17
MUL
MUL
ADD
SUB
STORE RbaseParam, #1, R18
GOTO R2
```



y: posto in memoria x valore



Passaggio parametri mediante STACK



quale reg uso come stack pointer : R20

stack che cresce verso indirizzi
crescenti di
Memoria

```

ADD ..., ..., R17
ADD R20, #1, R20      SP++
STORE R20, #0, R17   push(x)
CALL R1, R2
LOAD R20, #0, ...    y
SUB R20, #1, R20
...

```

```

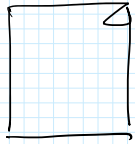
f : LOAD R20, #0, R5 } pop(x)
SUB R20, #1, R20
...
ADD R20, #1, R20
STORE R20, #0, R6
GOTO R2

```

base on R5
produciendo R6

USO REGISTRI

venerdì 28 ottobre 2016 11:53



.DRISC

R_i R_{baseA} R_{ret}
 ...

i	1	5
baseA	2	17
ret	3	21
f	4	
i		⋮

CALL R_f, R_{ret}

loop:

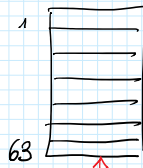


ADD $R_i, \#1, R_i$ $i++$
 HZ $R_i, R_N, loop$

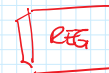


.DRISC

i-out
 .exe

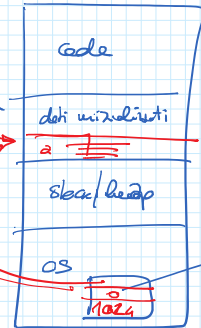


processo



```
int a[1024];
int i = 0;
int N = 1024;
for (i = 0; i < N; i++)
    a[i] = i;
```

$R_a = R_1$
 $R_i = R_2$
 $R_N = R_3$



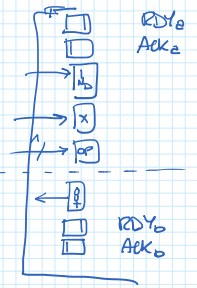
MV

area salvataggio
 registri

Esercizi

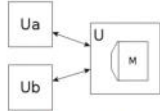
venerdì 28 ottobre 2016 12:17

U_a U_b "domanda risposta"



Si richiede di progettare una unità firmware U che interagisce con altre due unità firmware U_a e U_b . U mantiene al suo interno una memoria M di 2K posizioni (parole da 32 bit) e implementa tre distinte operazioni esterne:

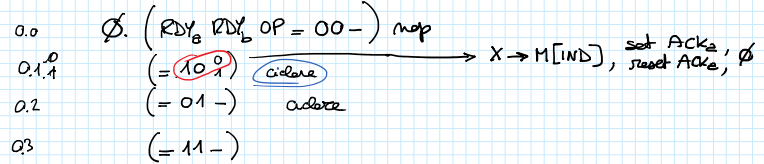
- scrive un valore X nella memoria all'indirizzo IND $OP=0$
- cerca un valore X nella memoria e ne sostituisce tutte le eventuali occorrenze con X/8192 (si assume che tutte le posizioni di memoria siano significative ai fini della ricerca)
- calcola il minimo fra tutti gli elementi contenuti nella memoria e lo restituisce come risultato dell'operazione esterna



Le prime due operazioni vengono richieste solo dall'unità U_a , la terza solo dall'unità U_b . U_a , U_b ed U interagiscono con U secondo un protocollo a domanda risposta.

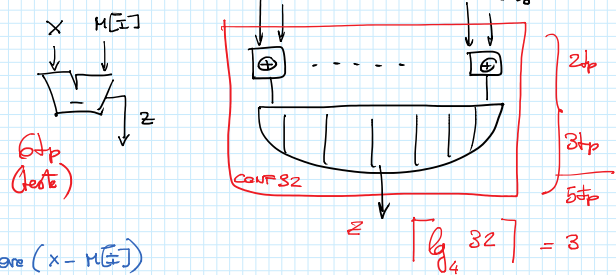
Si richiede di progettare l'unità U e di fornire il tempo medio di elaborazione, sotto le seguenti assunzioni:

- siano disponibili porte logiche AND e OR da al più 4 ingressi
- il ritardo introdotto da ALU che implementano operazioni standard sia $t_{alu}=6t_p$
- siano disponibili unicamente ALU che eseguono sottrazione e addizioni fra parole da 32 bit, quindi con un α di controllo da 1 bit
- il tempo di accesso alla memoria M sia $t_m=10t_p$
- il numero di microistruzioni da eseguire per ciascuna delle operazioni esterne sia minimo.



oppure
precedere alle U_a U_b fino
senza un flag TURNO

comparatore \times $M[I]$
32 bit



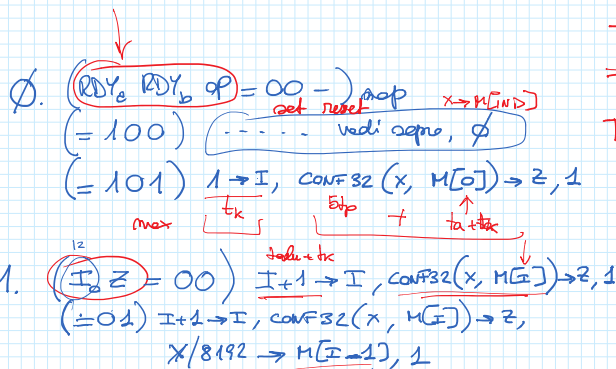
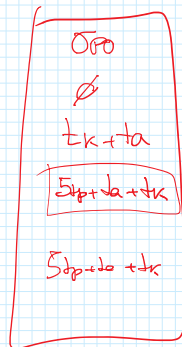
zero ($X - M[I]$)

CONF32 (X, M[I])

i. $(\neg \text{Zero}(X - M[I]) = 00)$ $I+1 \rightarrow I, 1$
(=01) $X/8192 \rightarrow M[I]$

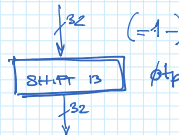
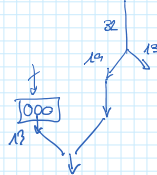
i-1. $\text{Zero}(X - M[I]) \rightarrow Z$

i. ($I_0, Z = \dots$)



TOP
 $T_{avg} = 0$
 $T_{avg} = 0$
 $T_{avg} = 0$
6 frasi (4 uguali)
2tp OR

1 bit x stato
5 superiori
di cui
max 3
left 2
Contappuntate
4 superiori
XAND
1tp
 $\Rightarrow T_{avg} = T_{avg} = 3tp$



$$\pi = \sum p_i k_i$$

$$\left. \begin{matrix} OP=0 \\ OP=1 \end{matrix} \right\} \text{ da } U_a \left\{ \begin{matrix} k_0 = 1 \\ k_1 = 1 + 2k \end{matrix} \right. \quad \left. \begin{matrix} p_0 = 0.5 \\ p_1 = 0.5 \end{matrix} \right.$$

$$\pi = \sum \left(\frac{1}{2} \cdot 1 + \frac{1}{2} (1 + 2k) \right)$$

