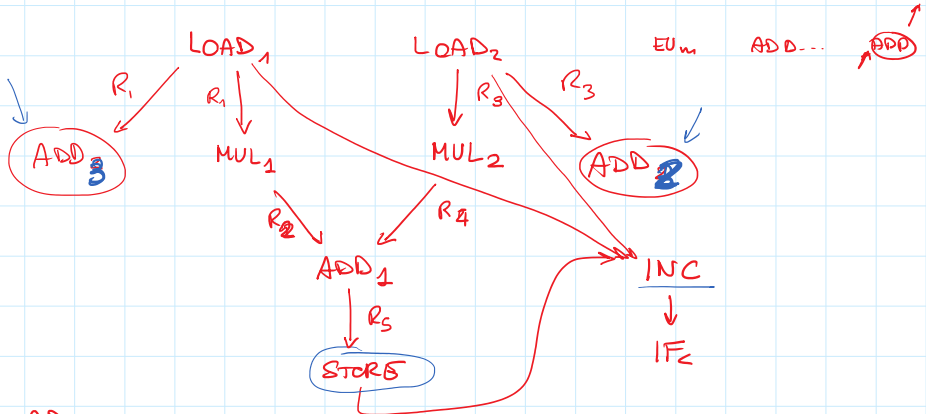
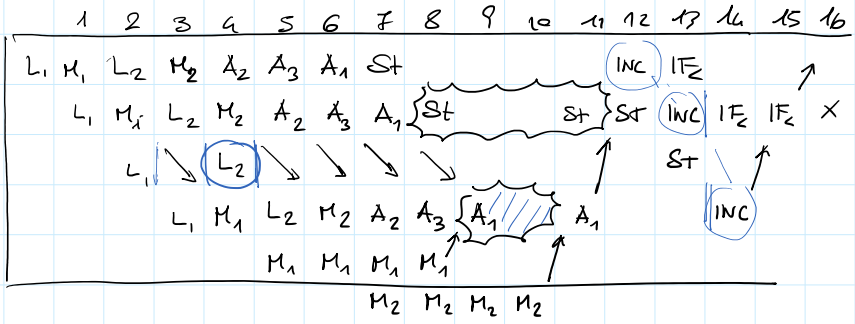
```
for (int i=0; i<N; i++)
    a[i] = c1*b[i] + c2*c[i];
    sumc += c[i];
    sumb += b[i]; }
```

```
ADD R0, R0, Rsumc
ADD R0, R0, Rsumb
loop : LOAD1 Rb, Ri, R1
       MUL1  Rc1, R1, R2
       LOAD2 Rc, Ri, R3
       MUL2  Rc2, R3, R4
       ADD1  R2, R4, R5
       STORE Ra, Ri, R5
       ADD2  R3, Rsumc, Rsumc
       ADD3  R1, Rsumb, Rsumb
       INC   Ri
       IF<   Ri, RN, loop
```

- LOAD1
- MUL1
- LOAD2
- MUL2
- ADD2
- ADD3
- ADD1
- STORE
- INC
- IF<

N pari






```
for(i=0; i<N; i++) {
    ≡
}
```
⇒
```
for(i=0; i<N; i+=2)
    ≡ [i]
    ≡ [i+1]
}
```
⇒
```
for(i=0; i<N; i+=2)
    ≡
    if((i+1)>N) break
    ≡ [i+1]
}
```

non importa che N sia Pari

ADD2 ADD1
ADD3 STORE
INC
IF< Ri, RN, Loop, delayed (1)
STORE Ra', Ri, R5

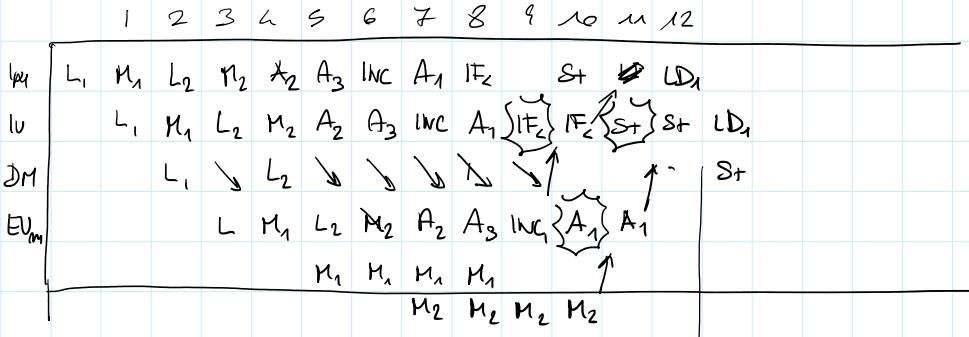
Hp Ra' = @base di A - 1

LOAD1
MUL1
LOAD2
MUL2
ADD2
ADD3
ADD1
INC
IF<
STORE

delayed

L1
M1
L2
M2
A2
A3
INC
A1
IF<
ST

delayed

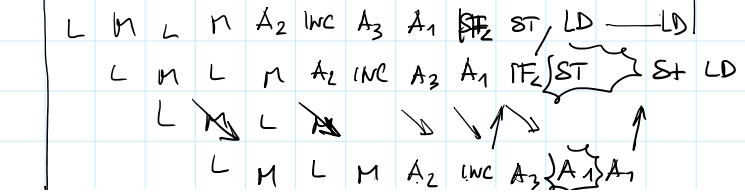


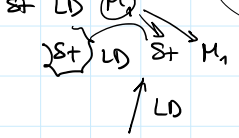
18 t
↓
12 F

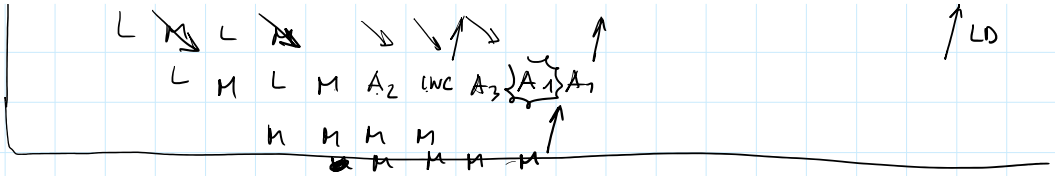
$T = \frac{12t}{10} = \frac{6}{5}t$

$T_C = 10N \cdot T$

+ #fall · Tfault



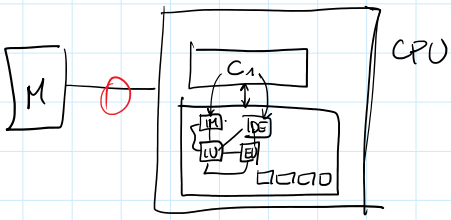



L
L
L
M
L
M
A2
LWC
A3
A1
A4
LD
M
M
M
M

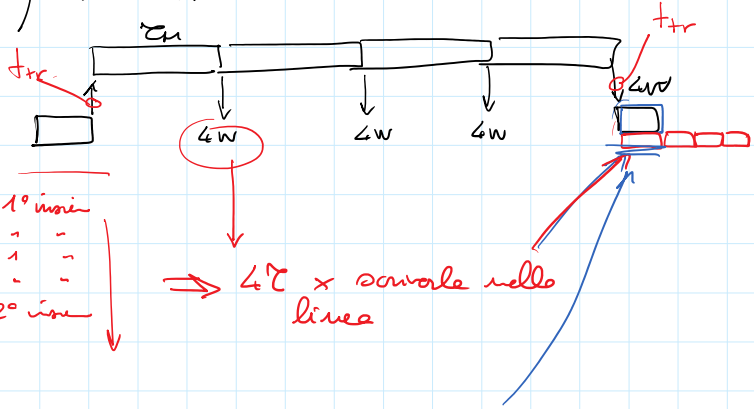
T_{transf}

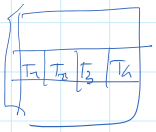
venerdì 16 dicembre 2016 11:52

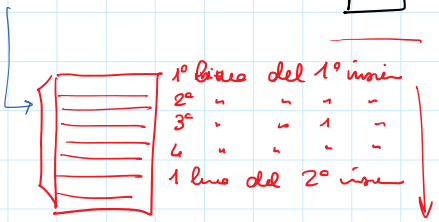
- Cache di 1° livello assoc. su insiemi a 4 vie $\sigma = 16$ (on chip)
- M interleacciate $m = 4$ (off chip)



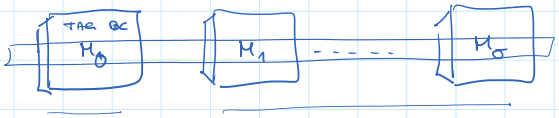
$\left(\frac{\sigma}{m}\tau_M\right) \rightarrow 4\tau_M$







$\Rightarrow 4\tau$ x scrivere nella linea



$\Rightarrow$ 4w → cache (1τ) =:

$$T_{transf} = 2(\tau + t_{tr}) + \frac{\sigma}{m}\tau_M + \underbrace{(m-1)\tau}_{\emptyset}$$

```
for (int i = 0; i < N; i++)
    a[i] = c1 * b[i] + c2 * c[i];
    sumc += c[i];
    sumb += b[i];
```

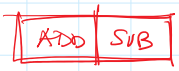
venerdì 16 dicembre 2016 12:19

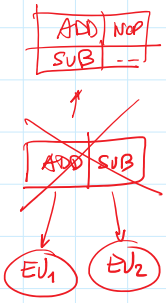
```
                ADD R0, R0, Rsumc
                ADD R0, R0, Rsumb
loop : LOAD  Rb, Ri, R1
       MUL   Rc1, R1, R2
       LOAD  Rc, Ri, R3
       MUL   Rc2, R3, R4
       ADD   R2, R4, R5
       STORE Ra, Ri, R5
       ADD   R3, Rsumc, Rsumc
       ADD   R4, Rsumb, Rsumb
       INC   Ri
       IF<   Ri, RN, loop
```

LOAD	MUL
LOAD	MUL
ADD$_1$	NOP
STORE	ADD$_2$
ADD$_3$	INC
IF<	NOP

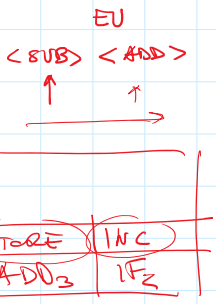
EU Superscalare

ADD R1 R2 R3
SUB R4 R2 R5





LD$_1$	MUL$_1$
LD$_2$	MUL$_2$
ADD$_1$	ADD$_2$
STORE	ADD$_3$
INC	NOP
IF<	



LD$_1$	MUL$_1$
LD$_2$	MUL$_2$
ADD$_1$	ADD$_2$
STORE	INC
ADD$_3$	IF<

8/6/2011
venerdì 16 dicembre 2016 12:29

int A[N]
∀ i = 0 .. N-1 ;
if (A[i%C] = ∅) then A[i] = 0
else A[i] = A[i] * A[i] + 1

N = 512 K
C = 2K

EU 1t ∀ operazione

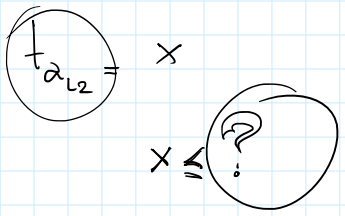
CDM associativo 2K parole
σ = 8
write through

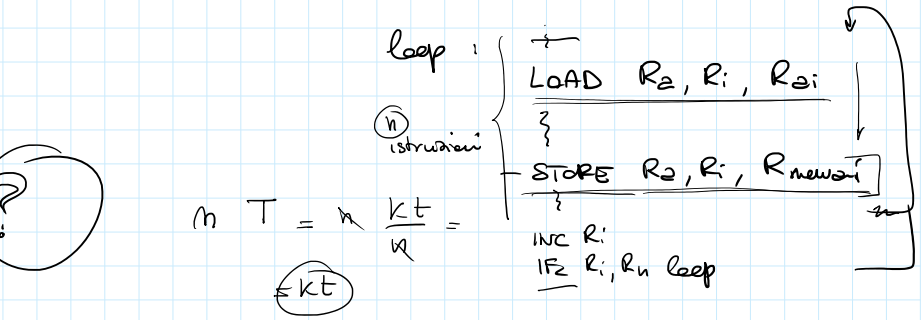
M interallacciata 4 moduli
r_M = 100 t

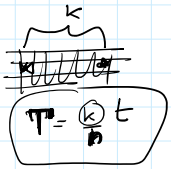
$P_{then} \approx \emptyset$
$P_{else} \approx 1$

A in L2 (tutto!)

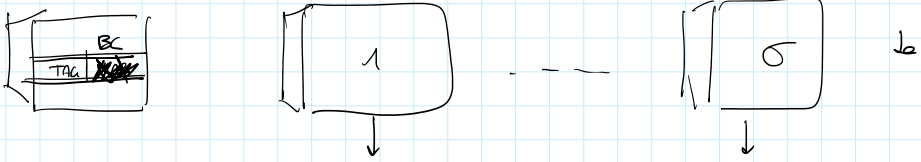
Cache secondaria (L2)
on-chip 1M parole

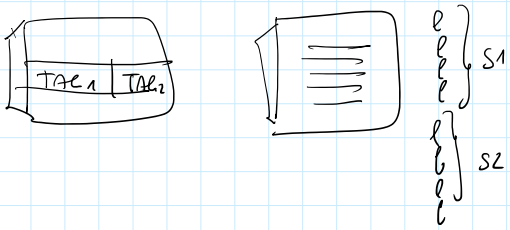


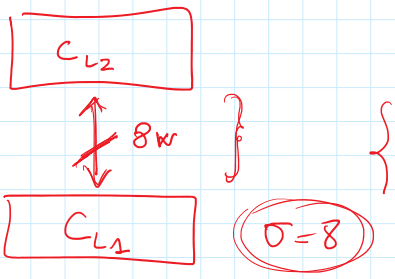




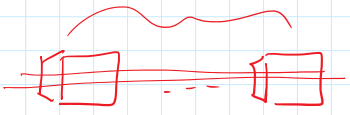
n T = n kt / K = kt





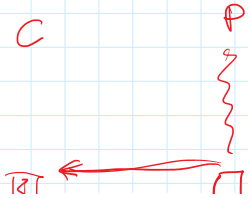


σ_2 / w
$\frac{\sigma_1}{\sigma_2}$

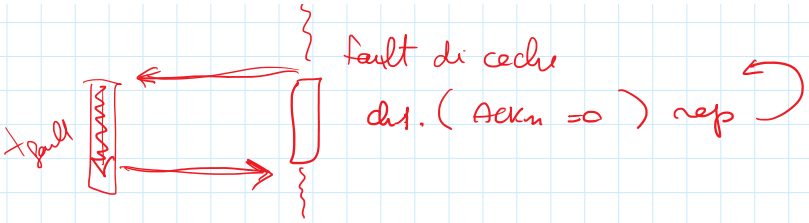


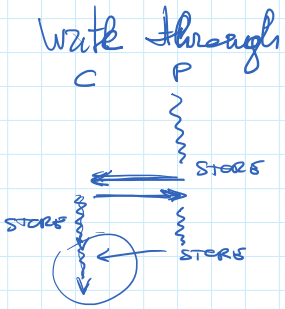
σ_1

politica : Write back : devo controllare banda di memoria ?



Write through
C P


fault di cache
dir. (AckM = 0) rep


write through
C
P
STORE
STORE
STORE