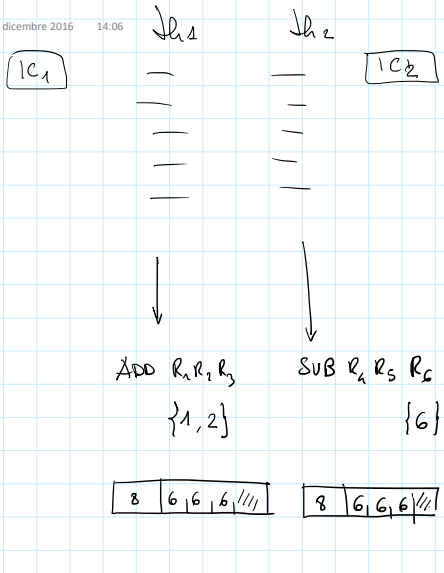
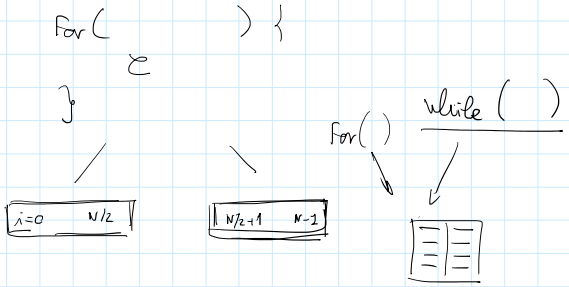
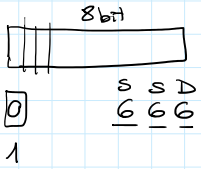

Th1
Th2
IC1
IC2
ADD R1 R2 R3
{1,2}
SUB R4 R5 R6
{6}
8 6 6 6
8 6 6 6
IF<
8 6 6
STORE
8 6 6 6


For ( ) {
}
i=0 N/2
N/2+1 N-1
For( )
while ( )


8bit
0
1
S S D
6 6 6

SUPPORTO FW ✳ MULTITHREADING

mercoledì 14 dicembre 2016 14:30

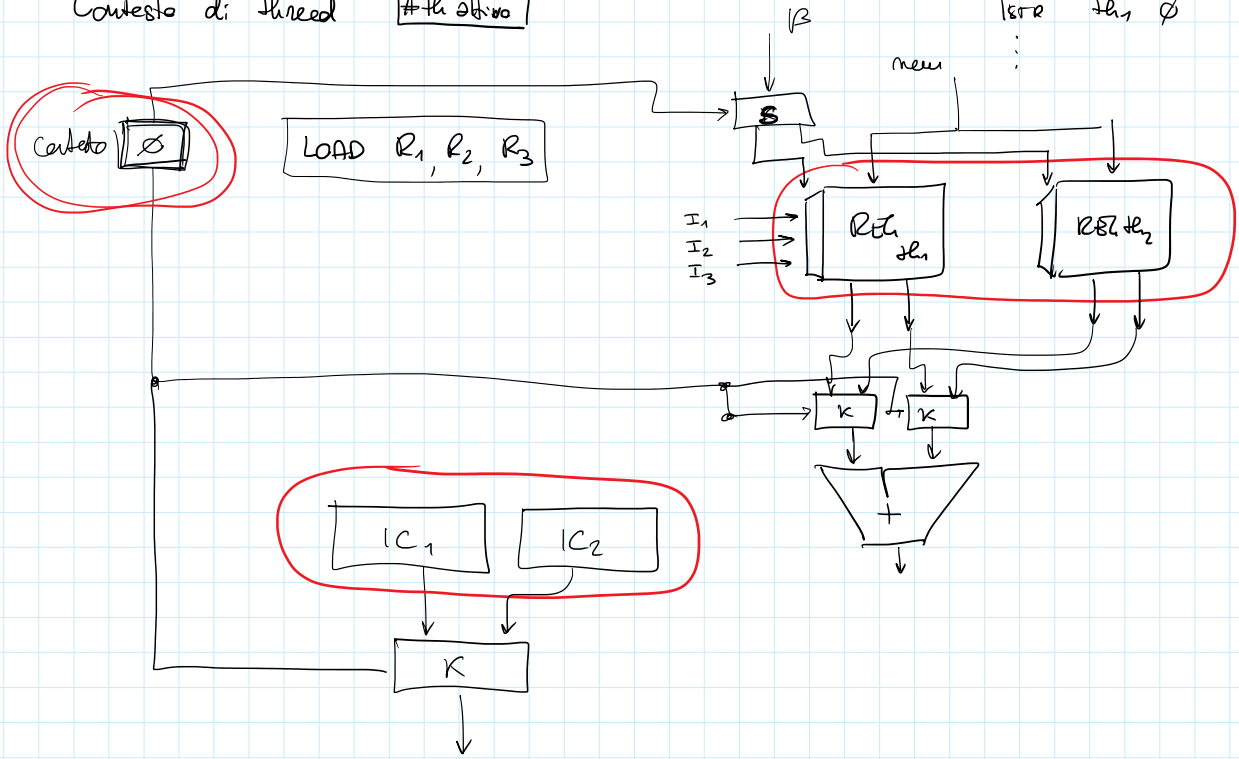
∀ thread IC REG

Interleaving 2 vie

ISTR th1 ∅
ISTR th2 1
ISTR th1 ∅
⋮

1 bit contesto

Contesto di thread #th attivo



multithreading

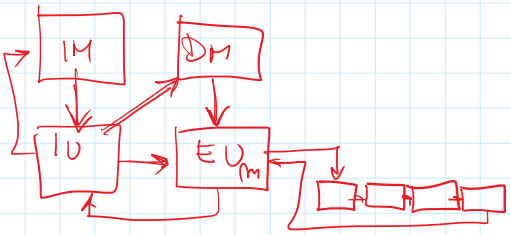
INTERLEAVED

∀t (contesto + 1) % nthread → contesto

BLOCKING

∀ volta che verifico la presenza di una dip (bolle ≥ 2)

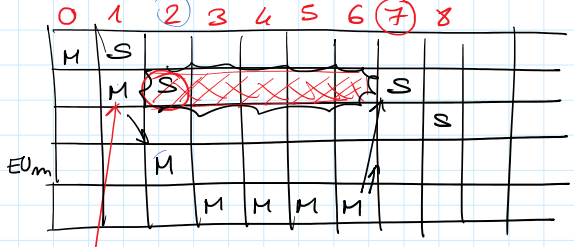
(contesto +1) % nthread → contesto



MUL R1 R2 (R3)

STORE R0 R1 (R3)

MUL intero 2 stadi



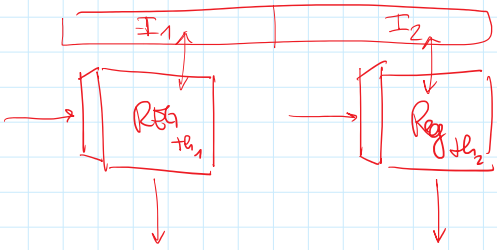
inc cont di R3 che andrò di nuovo a ∅ al ciclo 7

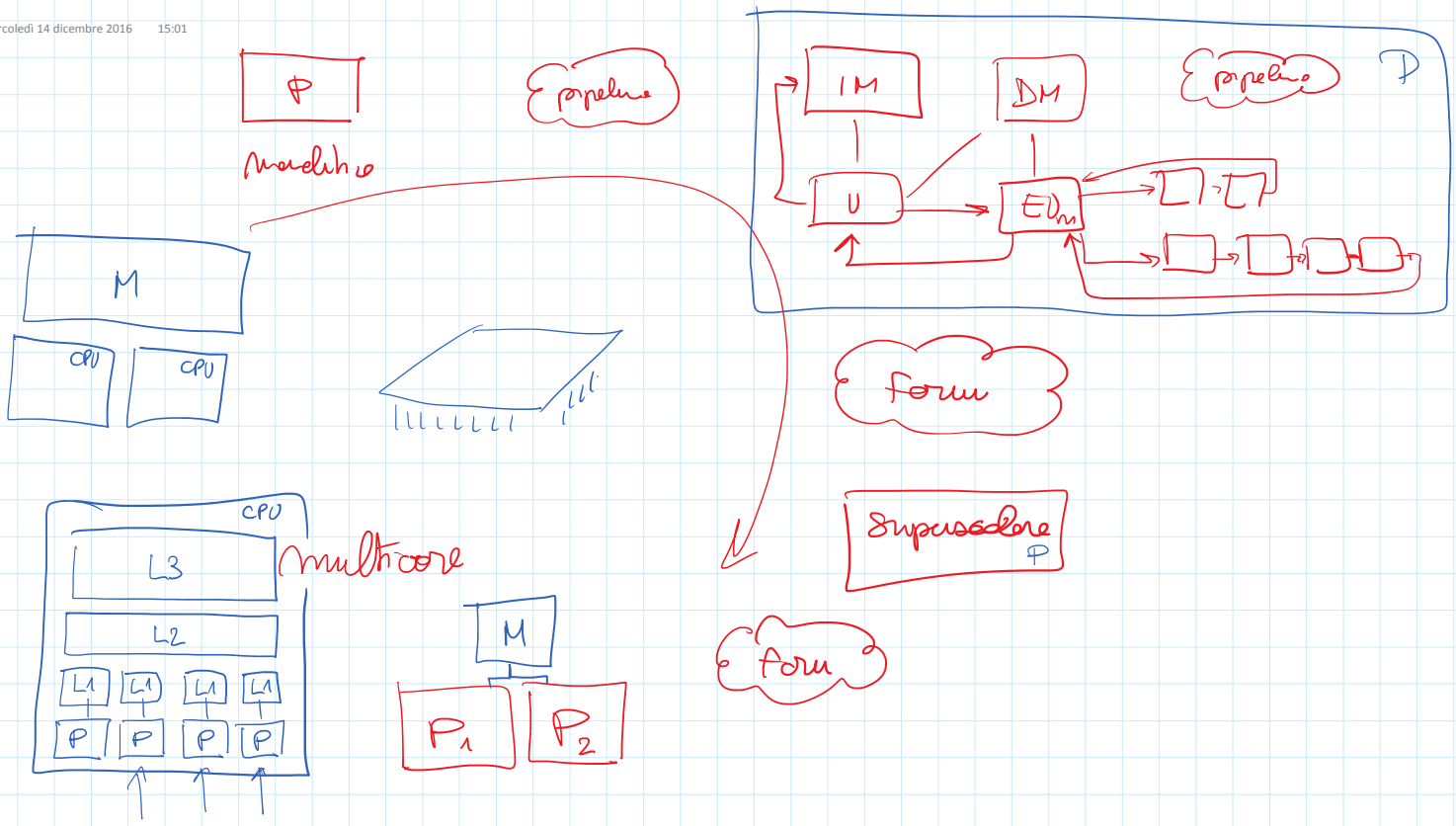
$T_C = \#istr \cdot T + \#fault * T_{transf}$

$T = \left(\frac{6}{2}\right) t$

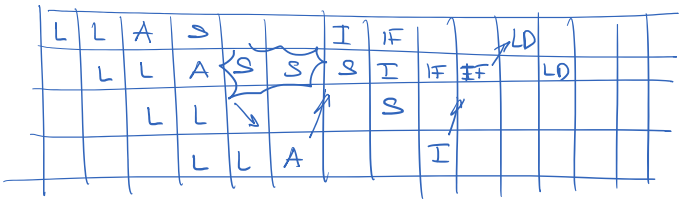
#fault * T_{transf}

MULTITHREADING SYMMETRICO (SIMULTANEOUS)

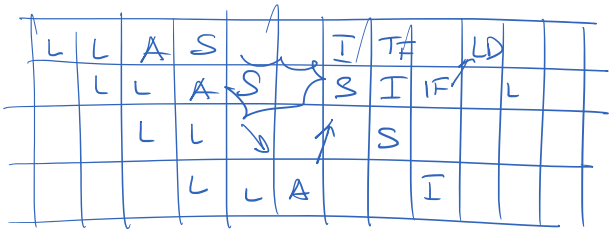



P
pipeline
modello
IM
DM
pipeline
P
U
M
CPU
CPU
forum
CPU
L3
L2
L1
L1
L1
L1
multicore
Supercalcolatore
P
M
P1
P2
forum

LOAD R_1
LOAD R_2
ADD $R_1 R_2$ R_3
STORE R_3
INC R_i
IF$_2$ R_i,



Core 1



Core 2

th_1 th_2

for(i=0 & N/2) for(i N/2+1 N-1)

ESERCIZI

mercoledì 14 dicembre 2016 15:25

D-RISC Pipeline

EU parallelo EUm

$EU_{*/INT}$ in 4 t (e 4 stadi)

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

T_c del codice

→ versione standard

→ versione ottimizzata

```
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 R1, Rsumb, Rsumb
      INC Ri
      IF< Ri, RN, loop
```

a[i] = c1×b[i] + c2×c[i];

```
for(  )
    a[i] += b[i],

for(  ) {
    a[0] += b[i]
```

```
LOAD Rb, Ri, R1
LOAD Rc, Ri, R2
ADD R1, R2, R3
STORE Ra, Ri, R3

LOAD Re, R0, R00
loop: LOAD Rb, Ri, R2
      ADD R2, R00, R00
      INC Ri
      IF< Ri, RN, loop
cat: STORE R0, R0, R00
```

10 istruzioni

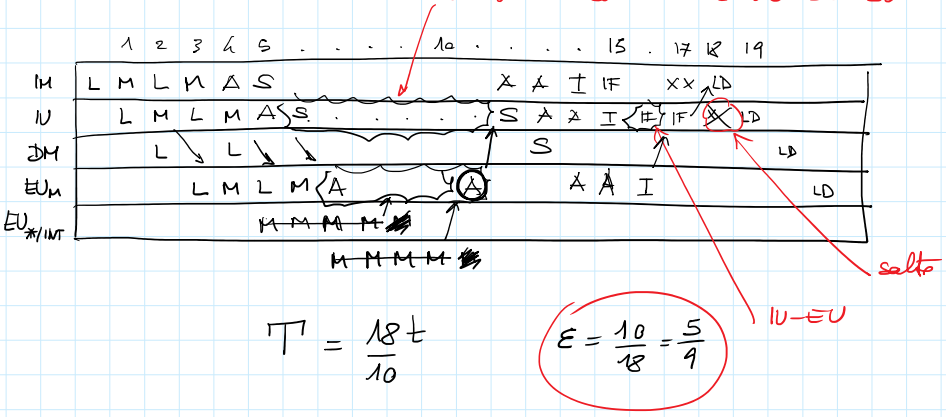
25t

$T = \frac{25t}{10}$

$T_c = \#istr \cdot T + \#fault \cdot T_{bassa}$

10? N? i? 25631? $N*10$

```
~~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  R1, Rsumb, Rsumb
       INC  Ri
       IF<  Ri, RN, loop
       XXX
```



$$T = \frac{18}{10}t \qquad \varepsilon = \frac{10}{18} = \frac{5}{9}$$

$$T_{cid} \simeq \#istr \cdot T = \frac{18}{10}t\,\cancel{10N} = 18Nt$$

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

WORKING SET

- Codice del corpo del ciclo (RIUSO + LOCALITÀ)
- i, sumc, sumb
- # linee di a, b e c (LOCALITÀ, NO RIUSO)

$$\#fault \simeq \frac{3N}{\sigma}$$

se abbiamo salto e

$$\#fault \simeq \frac{2N}{\sigma}$$