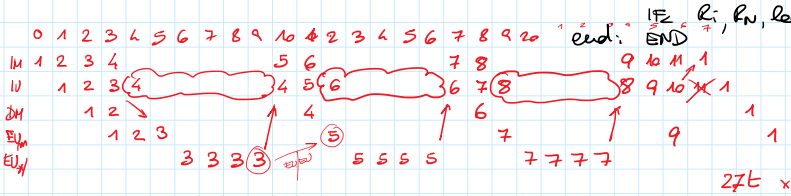
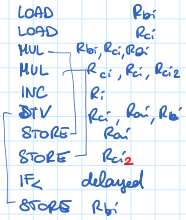
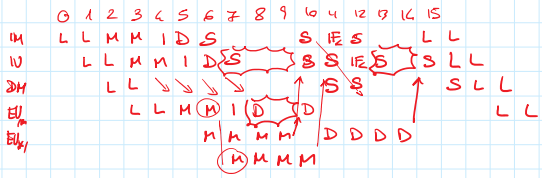
```
for(i=0; i<N; i++) {
    a[i] = b[i] * c[i];
    b[i] = c[i] / a[i];
    c[i] = c[i] * c[i];
}
```

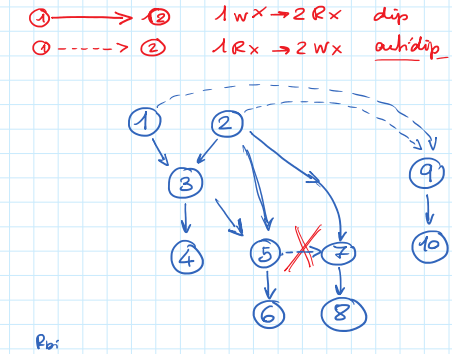
```
loop:  LOAD  RbaseB, Ri, Rbi
       LOAD  RbaseC, Ri, Rci
       MUL   Rbi, Rci, Rai
       STORE RbaseA, Ri, Rai
       DIV   Rci, Rai, Rbi
       STORE RbaseB, Ri, Rbi
       MUL   Rci, Rci, Rci
       STORE RbaseC, Ri, Rci
       INC   Ri
       IF    Ri, RN, loop
end:   END
```



27 t x iterazione

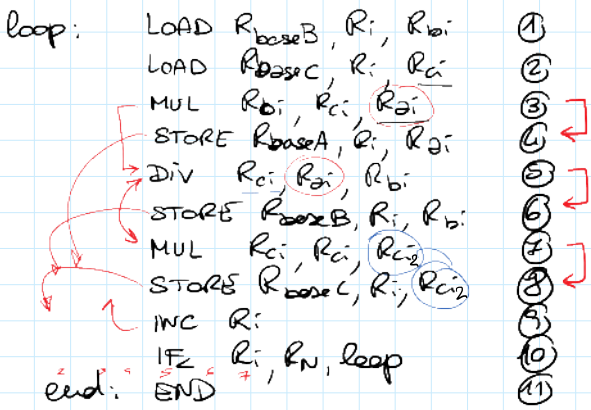


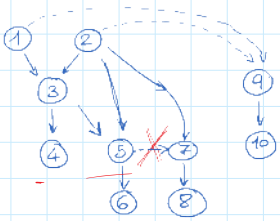


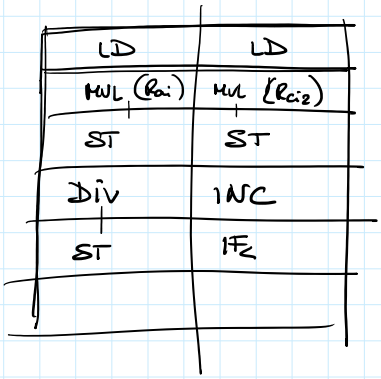


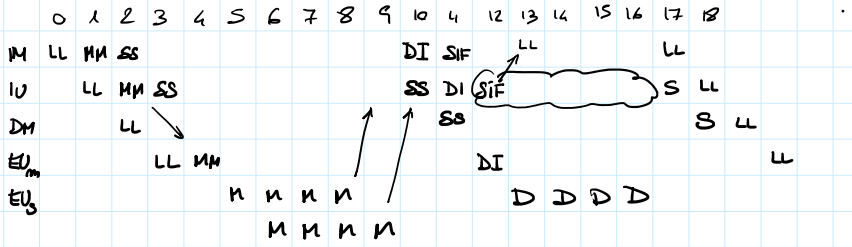
⑦ MUL Rci, Rci, Rci2
STORE RbaseC, Ri, Rci2

Proc superscalare a 2 vie

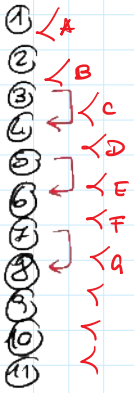




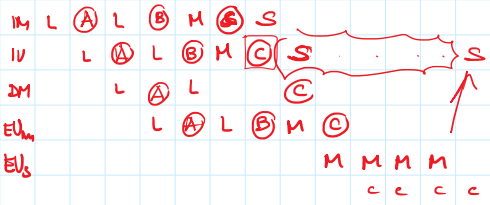




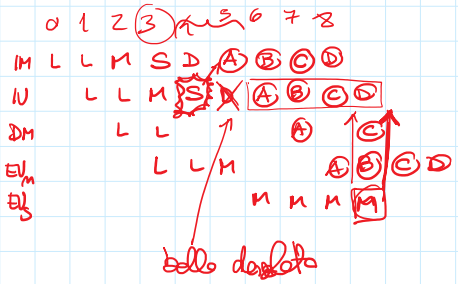
```
loop: LOAD  RbaseB, Ri, Rbi
      LOAD  RbaseC, Ri, Rci
      MUL   Rbi, Rci, Rai
      STORE RbaseA, Ri, Rai
      DIV   Rci, Rai, Rbi
      STORE RbaseB, Ri, Rbi
      MUL   Rci, Rci, Rci-2
      STORE RbaseC, Ri, Rci-2
      INC   Ri
      IFZ   Ri, RN, loop
end:  END
```

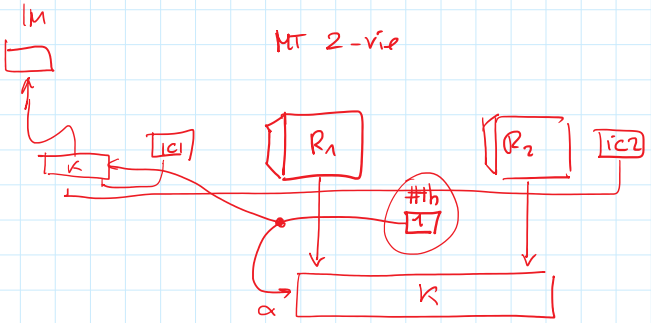


Multithreading interleaved



Blocking





MT simmetrico a 2 vie

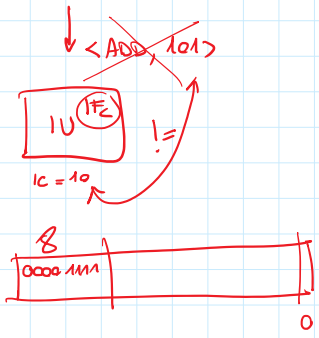
out of order — issue (IU)
— exec (EU)

salto ritardato

salto non ritardato

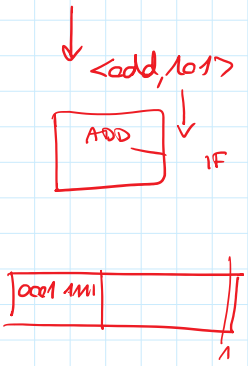
Loop 10
IC = 100
101

IF< Ri Rn Loop
ADD



salto ritardato

IF< Ri Rn loop, delayed



```
float a[N], b[N]
for( i = 1; i < N-1; i++ )
    b[i] = ( a[i-1] + a[i] + a[i+1] ) / 3;
```

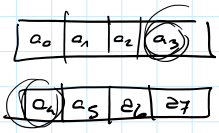
+/- FP 2 stadi
*/÷ FP 4 stadi

b fault in sola scrittura ⇒ allocare lo spazio in C ma non necessario caricare da M la linea allocata

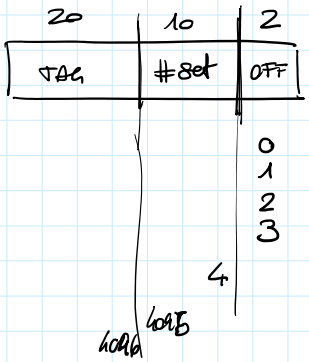
$\sigma = 16$) N/σ

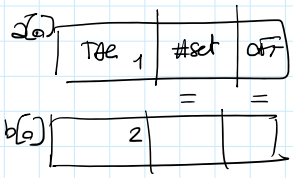
2

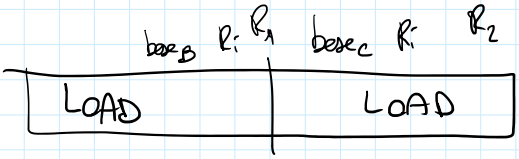
$$N_{fault} = \underset{\text{codice}}{1} + \frac{a}{\sigma}$$

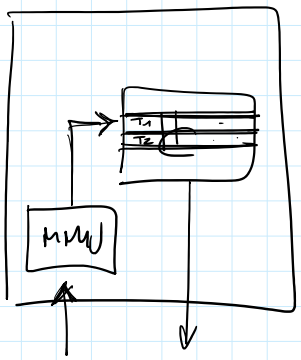


$\sigma = 4$

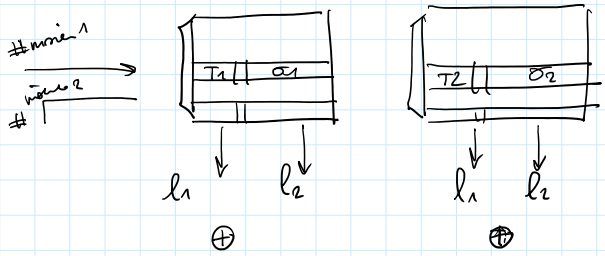


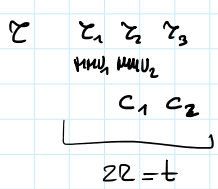


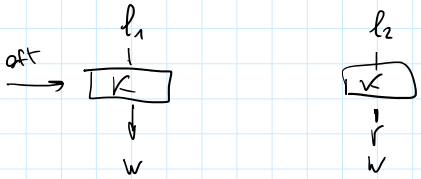




1° linea 2° linea





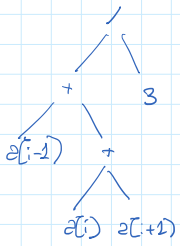


II DM

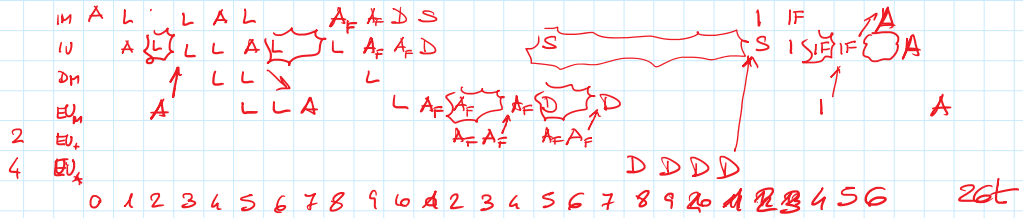
6 float a[N], b[N]

martedì 5 dicembre 2017 08:56

```
for( i = 1; i < N-1; i++ )
    b[i] = ( a[i-1] + a[i] + a[i+1] ) / 3;
```




```
loop: 1  ADD   Ri, #-1, Ri-1        ┐ IU ← EU
      2  LOAD  Rbase A, Ri-1, R1     ┘
      3  LOAD  Rbase A, Ri, R2
      4  ADD   Ri, #1, Ri+1          ┐ IU EU
      5  LOAD  Rbase A, Ri+1, R3     ┘
      6  ADD   R2, R3, R4            ┐ EU - EU
      7  ADD   R1, R4, R5            ┘
      8  DIV   R5, #3, R6            ┐ EU - EU
      9  STORE Rbase B, Ri, R6       ┘ IU EU
     10  INC   Ri
     11  IFZ   Ri Rn, loop
```



$$T_s = \frac{26t}{11} \approx 2{,}4t$$

8

martedì 5 dicembre 2017 08:56

11

martedì 5 dicembre 2017 08:57