venerdì 9 dicembre 2016 11:17

COPROCESSORE SLAVE

for { i = 0 , i < N , i++ } { V[i] = Z[i] * C[i] }

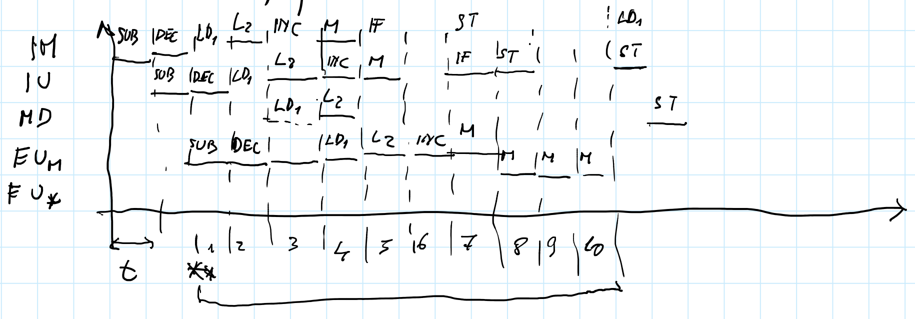
V , Z , C $\longrightarrow$ FP

3 cicli slave per *

RV , RZ , RC $\longrightarrow$ indirizzo base vettore

```
        SUB   Ri, Ri, Ri
        DEC   RV
LOOP :  LD1   RC, Ri, Rf1
        LD2   RZ, Ri, Rf2
        INC   Ri
        MUL   Rf1, Rf2, Rf1
        IF<   Ri, RN, LOOP    <--- delayed branch
        ST    RV, Ri, Rf1
```

$n * t * 10 \longrightarrow$ reale
ideale $\longrightarrow 6 * n * t$
} eff. 60%



for { i = 0 , i < N , i++ } { A[i] = B[i] + C[i] ;
D[i] = B[i] * C[i] }

A – D ≡ interi ; EU singola / t

n° fault B/C $\doteq n/\sigma$ = dim pagine

n° fault $\doteq 2 * n/\sigma$

```
        SUB   Ri, Ri, Ri

LOOP :  IF≥   Ri, RN, FINE
        LD1   RB, Ri, Rb
        LD2   RC, Ri, Rc     EU-EU
IU-EU ( ADD   Rb, Rc, Ra
        ST1   RA, Ri, Ra
IU-EU ( MUL   Rb, Rc, Ra
        ST2   RD, Ri, Ra
        INC   Ri
        GOTO  LOOP
FINE :
```

$t_{id} = t_{calc} = 9 * n * t$

tempo reale $13 * n * t$

Eff. monoproc. pr. $= \frac{9}{13}$

n° pagine 1 per B , 1 per C , 2 codice

Agg. di A/D pag. $\longrightarrow 2 * n/\sigma$

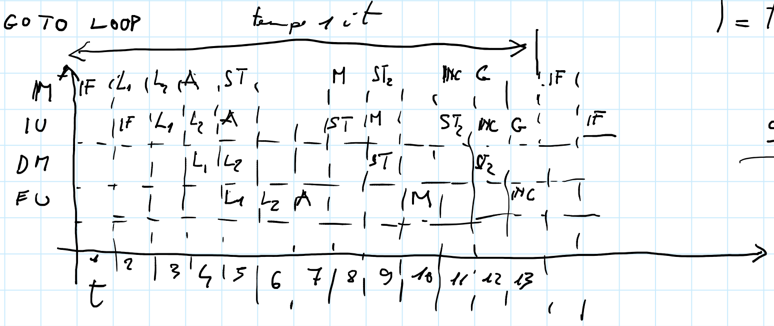
Numero fault comp. $\doteq \overset{\text{dati}}{\overline{4 * n/\sigma}} + \overset{\text{istr.}}{\overline{2}}$

Agg. write-through $\doteq 2 * n/\sigma + 2$

$$T = t_{calc} + (4 * n/\sigma + 2) * T_{trasf.} + T_{agg.}$$

$\downarrow$

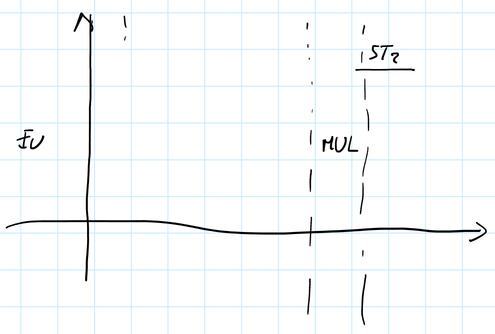
$4 * n * t$

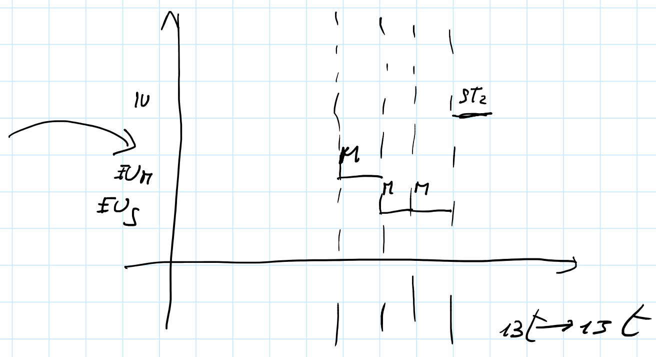


$\underline{9 * n * t}$

$$13 * n * t + \left[2 * \left(\frac{n}{\sigma} \right) + 2 \right] T_{trasf.}$$

EU Pipeline   Master +/- IF, slave * IF ~ 2t



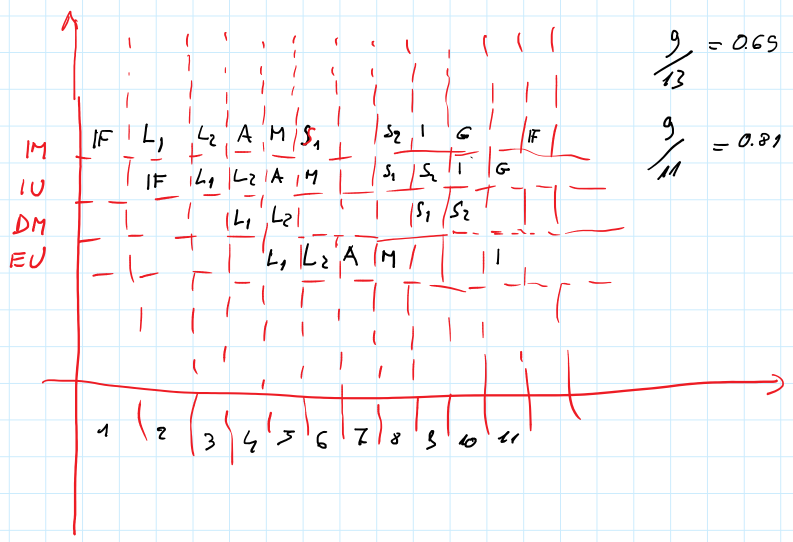


9/13 → 9/15

1° ott. sul codice → aumenta distanza delle dipendenze IU-EU

13 → 11

```
LOOP : IF>
       LD1
       LD2
       ADD
       MUL
       ST1
       ST2
       INC
       GOTO
```


IM
IU
DM
EU
IF L_1 L_2 A M S_1 S_2 I G IF
IF L_1 L_2 A M S_1 S_2 I G
L_1 L_2 S_1 S_2
L_1 L_2 A M I
1 2 3 4 5 6 7 8 9 10 11

$\frac{9}{13} = 0.69$

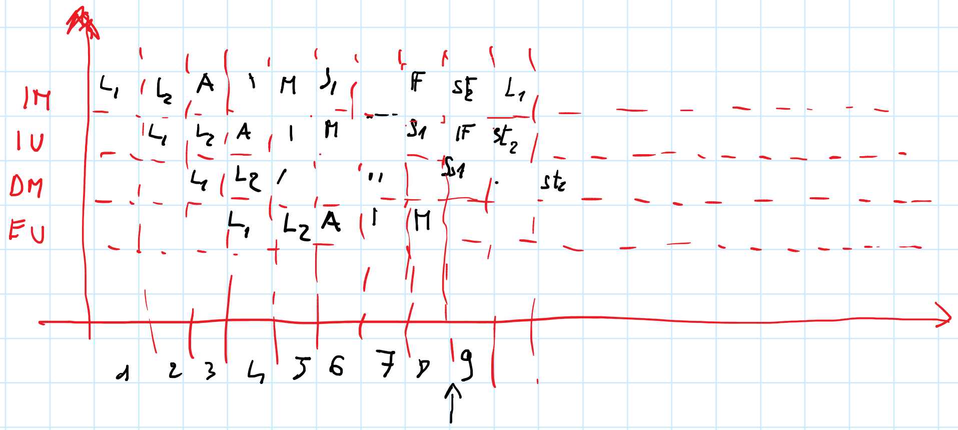
$\frac{9}{11} = 0.81$

2ª ott. annulla dst + delayed branch

```
      SUB  Ri, Ri, Ri
      DEC  RD            ] delayed branch
      DEC  RA
LOOP: L1
      L2
      ADD
      INC
      MUL
      ST1
      IF<  LOOP   ⟵ non si salta solo 1 volta alla fine
      ST2
```

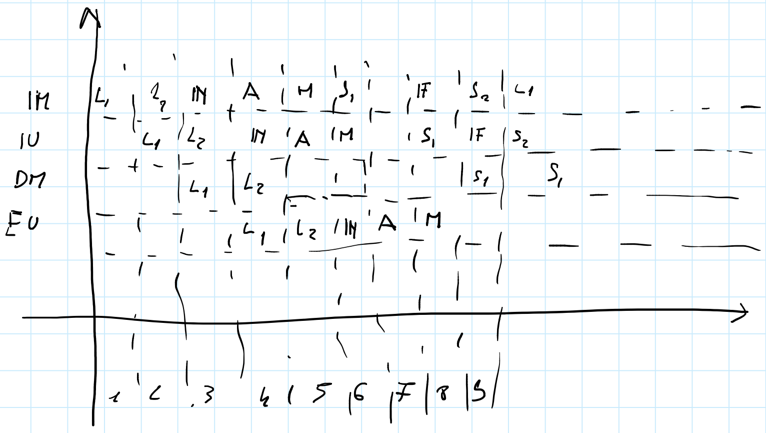
almeno 1 volta eseg.

eff. $8/9 \rightarrow 0.9$



venerdì 9 dicembre 2016 11:18

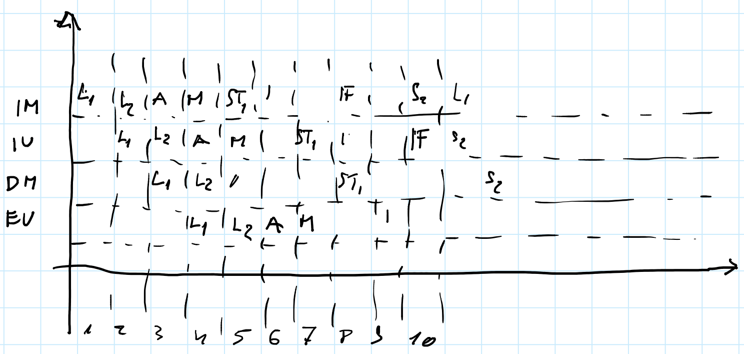
```
LOOP: L1
      L2
      INC
      ADD
      MUL
      ST1
      IF        LOOP
      ST2
```



Permutazione distanze aumenta da INC

diminuisce per ADD

```
        SUB  Ri, Ri, Ri
        DEC  RB
LOOP:   LD1
        LD2
        ADD
        MUL
        ST1
        INC
        IF <
        ST2
```



```
for {i = 0, i < N, i++}
    { A[i] = B[i] * c1 + D[i] * c2
      D[i] = B[i] * A[i]
      B[i] = D[i] * c3 + Q[i] }
```

$c_1, c_2, c_3 \Rightarrow$ caricate nei reg. prima di entrare nel loop

$\overset{ott.}{\neq}$ $B[i] * D[i] + c_1 * c_2$

22 t

```
        LOOP: IF≥  Ri, RN, FINE
exp1          LD1  RB, Ri, Rb
              LD2  RD, Ri, Rd
              MUL1 Rb, R1, Rt1
              MUL2 Rd, R2, Rt2
              ADD1 Rt1, Rt2, Rt1
              ST1  RA, Ri, Rt1
exp2          MUL3 Rt1, Rb, Rt1
              ST2  RD, Ri, Rt1
exp3          LD3  RQ, Ri, Rq
              MUL4 Rt1, R3, Rt1
              ADD2 Rt1, Rq, Rt1
              ST3  RB, Ri, Rt1
              INC  Ri
              GOTO LOOP
```

$R_i = c_i$

```
LOOP: LD1
      LD2
      LD3
      MUL1          calcolo
      MUL2
      ADD1
MUL4 → MUL3
      ADD2
      INC
      ST1
      ST2           store e salto
      IF
      ST3
```

15 t