

LUNEDÌ ORE 12 LEZ. DI Recupero

Uniproc. Parallelo

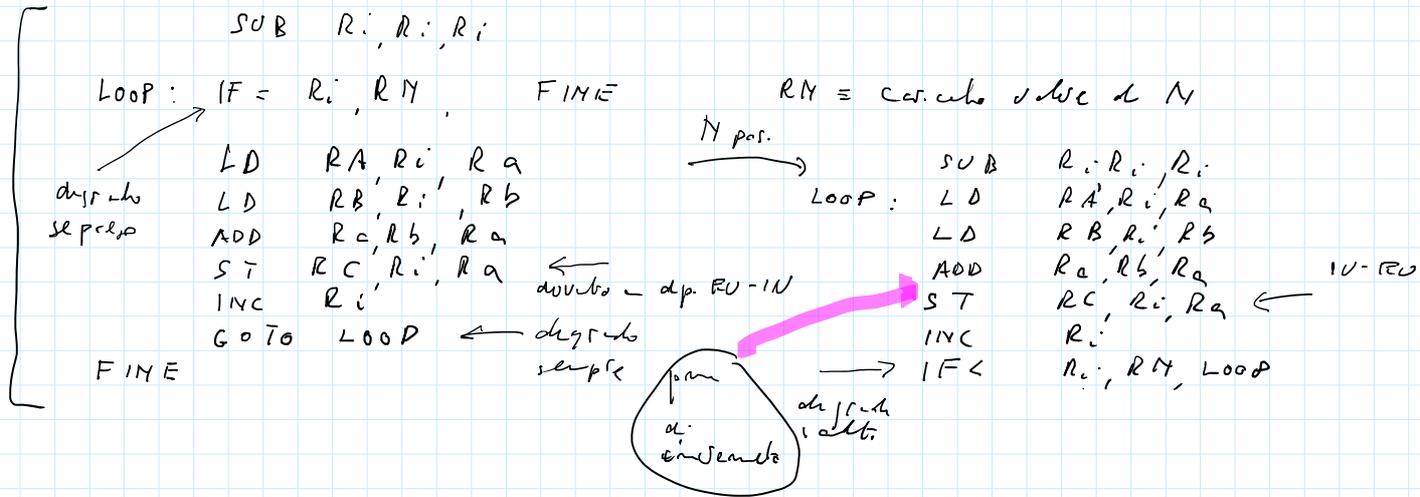
Delayed branch dovuto ai salt.

Delayed Branch \rightarrow Integre routine e compilazione

```
for { i = 0, i < N, i++ } { C[i] = A[i] + B[i] }
```

a) caso base compilazione

RA \equiv registro la cui base è di A



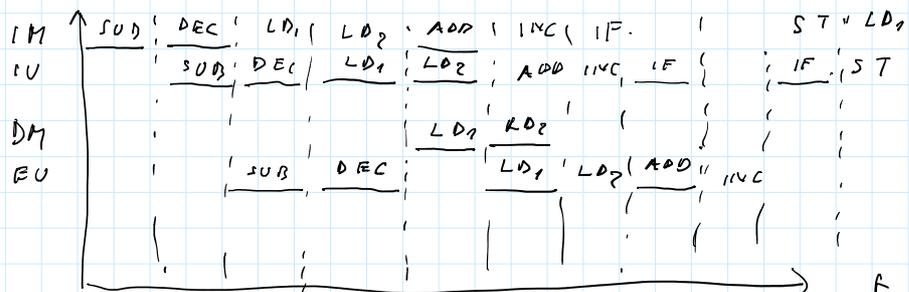
Delayed Branch

```

SUB  Ri, Ri, Ri
DEC  RC
LOOP: LD1  RA, Ri, Rc
      LD2  RB, Ri, Rb
      ADD  Ra, Rb, Rc
      INC  Ri
      IF< Ri, RM, LOOP
      ST   RC, Ri, Ra
    
```

espr. un'ops

← i.e. delay
 ← conten. ind. dc
 meno 1



← tempo del loop (i.t.)

Degrado dovuto alle dipendenze logiche IU-EU

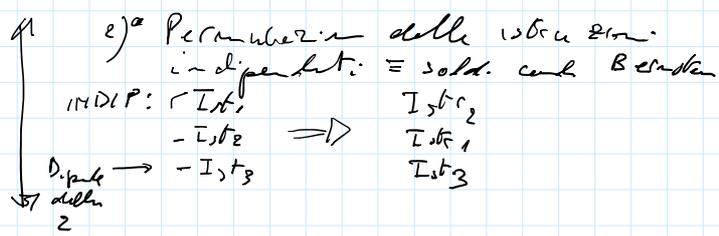
Ottimizzazione Statiche

1) Allocazione Register

selezione
 del numero
 risorse di reg. usate

```

ADD  R1
...
R2
ST   R3
    
```



Permutazione Ist.

Funzione Obiettivo: Distanza media tra due ist. in dipendenza

S. considero tutte le ist. in dipendenza
tutte le copp. istruzione,

$$\frac{\sum_{i,j} d(i,j)}{\text{n° coppie}}$$

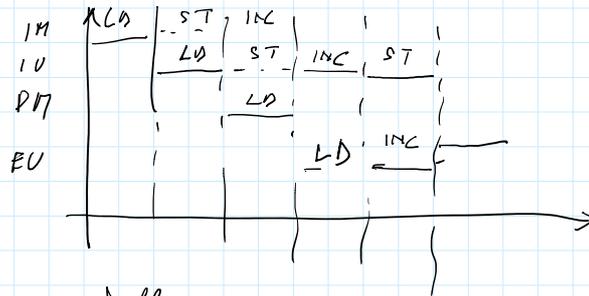
Ottimizzazione D. - anche → lavorare su interpretazione (anche di dati)

1) Out of Order

istruzione prima
e seconda
può passare
avanti ed
una bloccata
nella IU



Non è bloccata
e ubr. succ.
è eseguibile
IU diventa più complessa
ma più performante ist.



ballo
senza
0-0-0

← r. di zero ballo
con 0-0-0

Riduzione del grado di parallelismo

Predizione del salto

⇒ dipendenza logica sul r.a. tra cond.

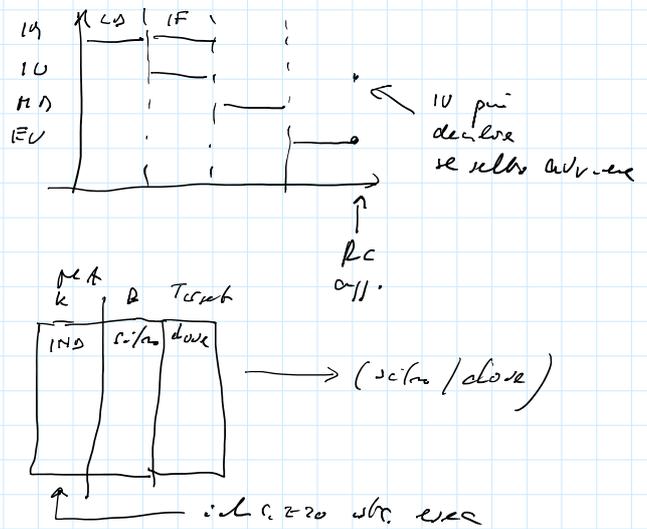
LD R_a, R_i, R_c
IF R_c, ...

Predizione del salto

decide se salto avviene e se con quale di cond.

back track se previsione errata

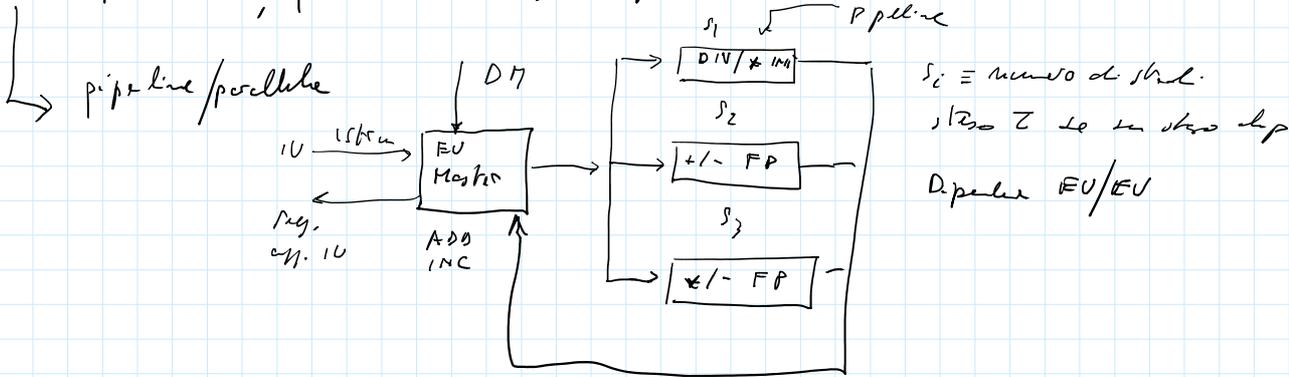
Prediction ≡ storia con dell'ultima volta



Altra tecnica di riduzione dei gradi di parallelismo

Accelerare EU → grado è proporzionale a ρ → lunghezza delle code tra IU e UE

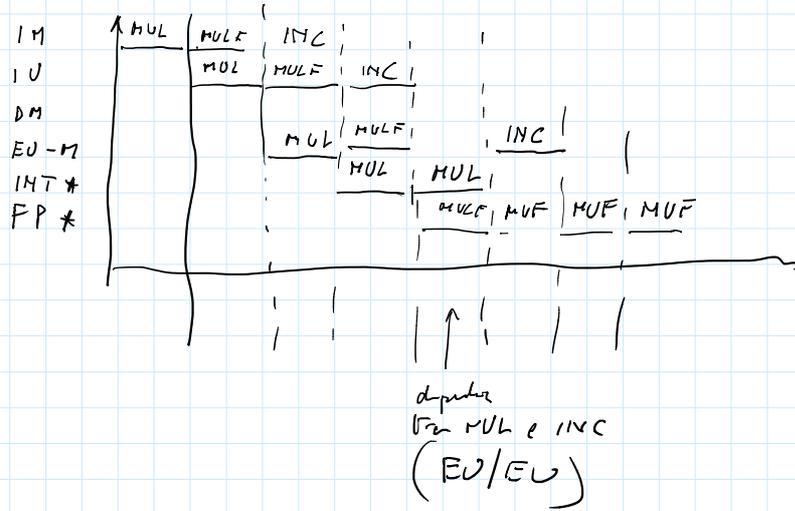
Se rateo EU più veloce, ρ diminuisce, lunghezza code diminuisce



EV P.pipeline Parallela

MUL R_a, R_b, R_c
 MULF R_x, R_y, R_z
 INC R_c

INC = 1t nel modulo
 MUL = 3t (1 nel modulo
 2 nel pipeline)
 MULF = 5t (1 nel modulo
 4 nel pipeline)



Problema dipendenze

soliti

- delay branch

esiti..

- alt. statiche ← compilatore ↔ costo nullo sull'ord.

- alt. dinamiche ← interprete ↔ aumento di complessità
 unità di elab.

OUT of ORDER
 PREDIZIONE JBTG
 r.dinamiche

⇒ aumento costo

Conversione ??

Architetture su scala ri

Unproc. \rightarrow un'hi di lusso istruzione

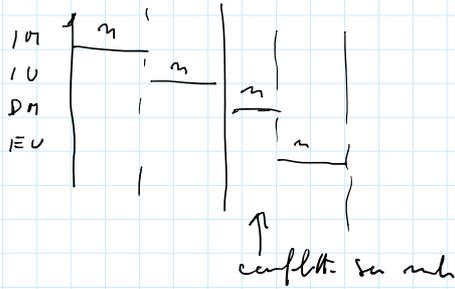
Superscalare \rightarrow un'hi di lusso m istruzioni m fluss e dep. de clock

Tempo di esc. $\frac{t_c}{m}$ Perf = $\frac{m}{t_a}$] Ideali

Memoria Istr. / Dati \rightarrow Interallaccata (Modulare, 2 uti car. in moduli diversi.)

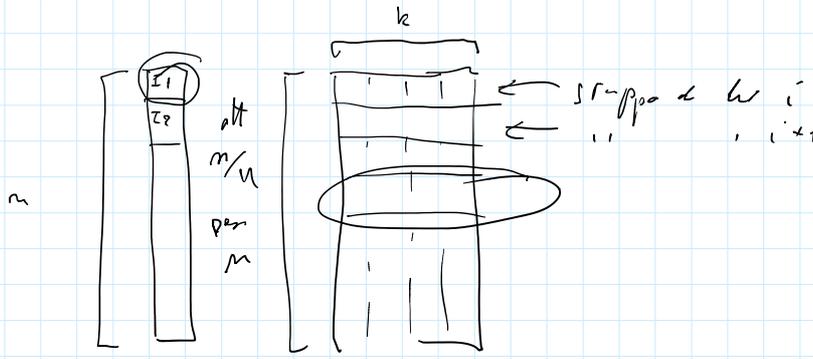
$$m = 2 - 8$$

16-32 Sp5.



Gruppo di m istruzioni

Strategie stat. \rightarrow un p. decupare il programma in un'hi di el più m istruzioni



Comp

LAL → LM → gruppi

L1W o VL1W
very long instruction word

Regole per costruire gruppi
in un gruppo

- a) istruzioni non possono avere dip. IU/EU
- b) una sola istruzione di salto
- c) solo la prima istruzione del gruppo può avere target di un salto

