DATA FLOW

X[i] Y[i]

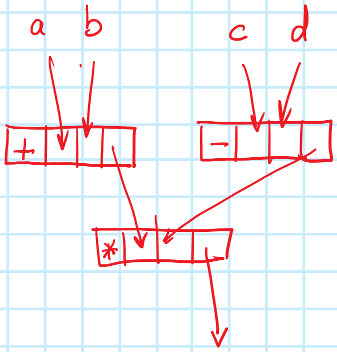
$(a+b) * (c-d)$

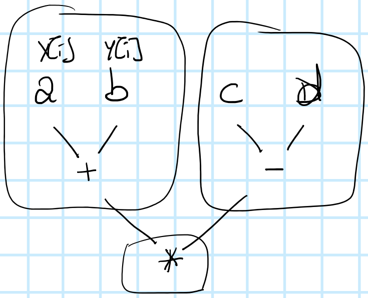
$t_1 * t_2$

② | LOAD x[i]
LOAD Y[i]
ADD R_1

① | SUB c-D R_2

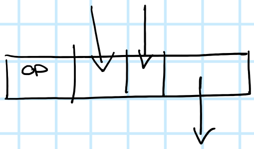
③ MUL





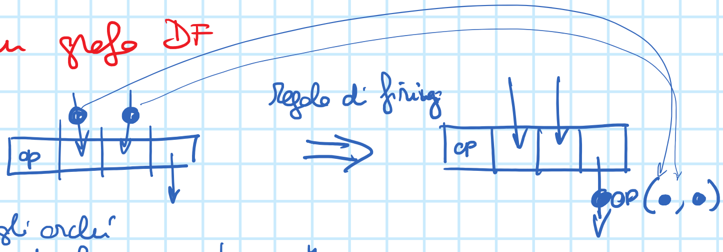
GRAFO DATA FLOW

istruzioni DF

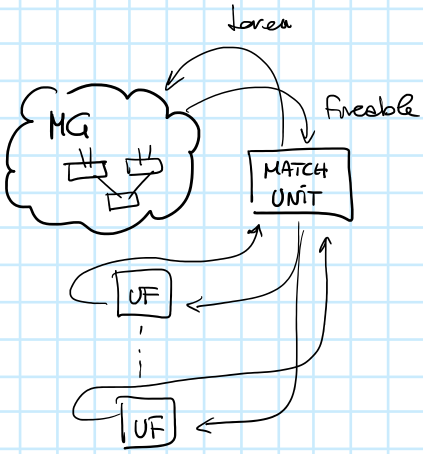


"Calcolo" di un grafo DF

fireable



"tutti gli archi entranti hanno un token"

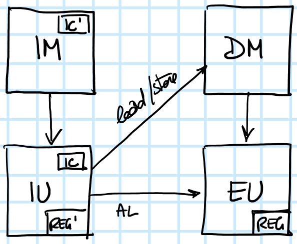


PROCESSORE D-RISC "PIPELINE" (HARVARD)

mercoledì 29 novembre 2017 09:35

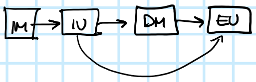
instruction memory
fetch

instruction unit
decode + exec salti



data memory
execute LD/ST

execution unit
execute AL/LD



ADD

IM → IU

IR

OPCODE Ra ....

ADD	a	b	c

IU decodifica
→ EU
<+, a, b, c>

EU
$R_a + R_b \rightarrow R_c$

LOAD STORE

IM → IU

LOAD	a	b	c
STORE	a	b	c

IU decodifica
Ra+Rb'
<"LD", ind> Rc' → DM
<"ST", ind, val>
→ EU <"ld", c>

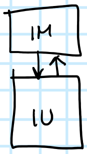
DM esegue op richiesta
se LD invia a EU
<val>

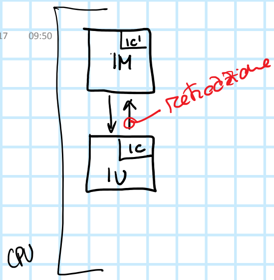
EU
IU → <"ld", c>
EU → <val>
} val → Rc

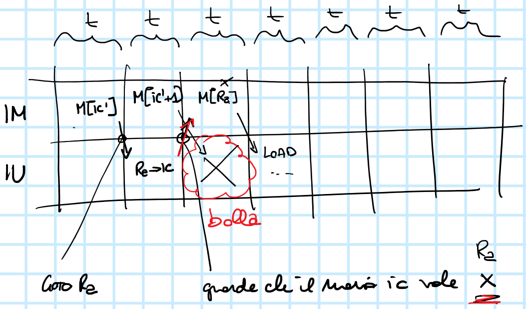
SALTI IM → IU

GOTO	a	

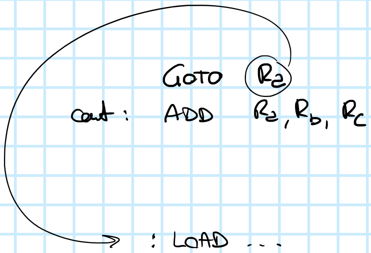
IU
$R_a \rightarrow IC$
manda ad IM il nuovo valore di IC

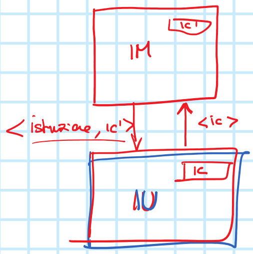






← tempo di servizio (o latenza, tanto IM ed IU sono sequenziali) di IM ed IU





while true {
se IU mi manda new IC
then manda alla IU M[new IC]
new IC -> IC'
else manda alla IU M[IC']
IC'++
}

while (true) {
...
if (<istruzione, IC'>. IC' == IC)
then (decodifica <istruzione, IC'>. istruzione)
else nop

delayed branch

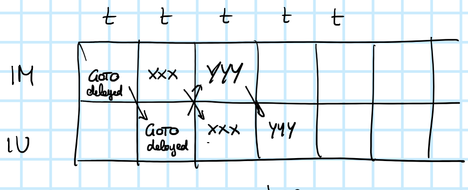
GOTO Ra, delayed = 1
XXX

Ra → YYY

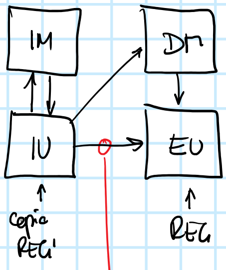
come se: XXX / GOTO Ra → YYY

sse "Bernstein"

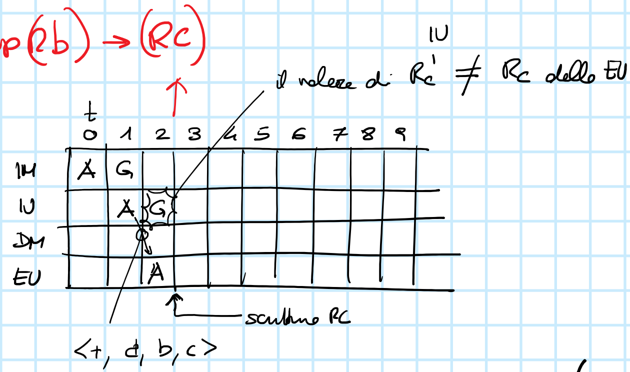
XXX non altera registri letti da istruzione di salto



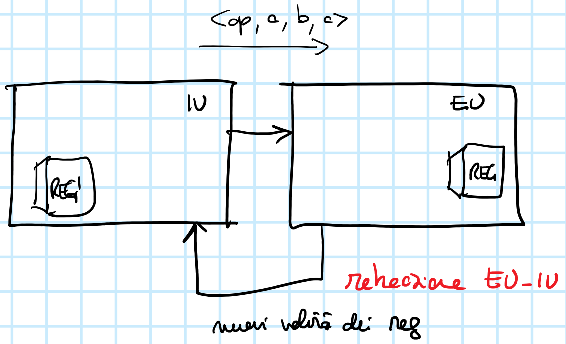
nessuna bolla

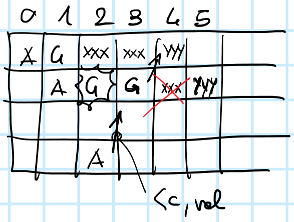


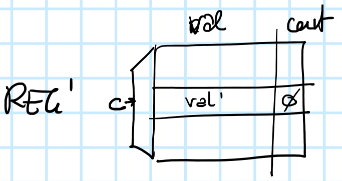
<op, a, b, c> ⇒ EU: fai (Ra) op (Rb) → (Rc)



evidenzia il problema







sse IU manda <op, - - c> ad EU

cont(c)++

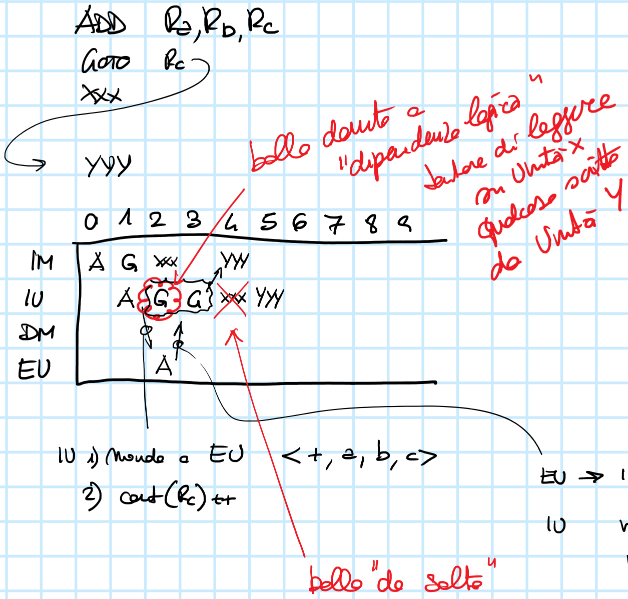
quando IU legge Rc

se cont(Rc) ≠ 0 ⇒ si blocca

ogni volta che arriva < val, x > dal canale retroazione EU→IU

val → Reg'[x]. valore

Reg'[x]. cont --



IU 1) manda a EU <+, a, b, c>

2) cont(Rc)++

EU → IU <val, c>

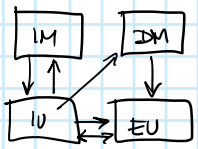
IU val → REG'[c]. valore

REG'[c]. cont --

esecuzione del programma

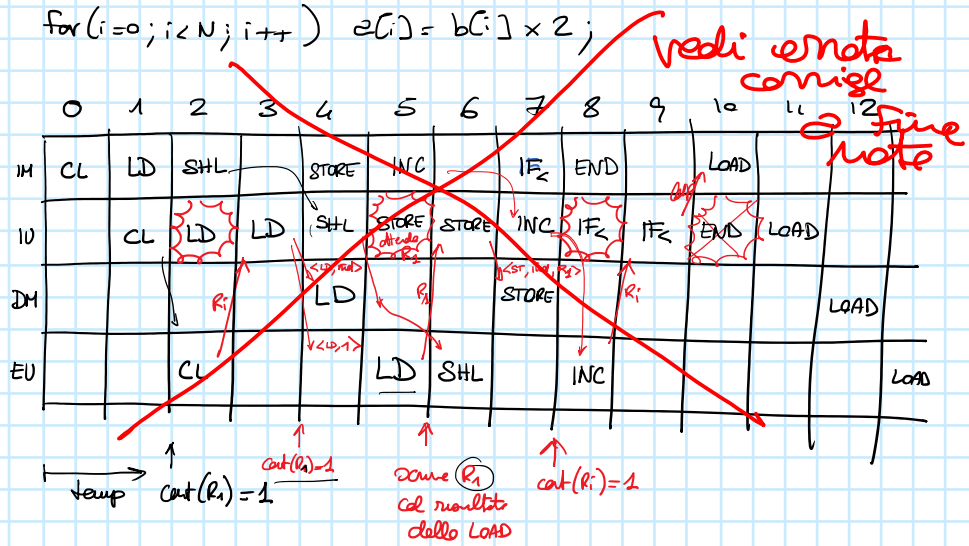
```
t  CLEAR  Ri
loop: t  LOAD   R_baseB, Ri, R1
      t  SHL    R1, #1, R1
      t  STORE  R_baseA, Ri, R1
      t  INC    Ri
      t  IF<    Ri, RN, loop
      END
```

Tid = 6t




```
for(i=0; i<N; i++) c[i] = b[i] × 2;
```

vedi nota correzione fine nota



tempo di completamento del codice: init + 1ª iterazione } 10t in questo caso

$$\varepsilon = \frac{6t}{10t} = 0.6 \qquad T_{id} \,/\, T(n)$$

mercoledì 29 novembre 2017 10:58

2 problemi:

bolle "da salto" | bolle "da dipendenze logiche"

SOLUZIONI

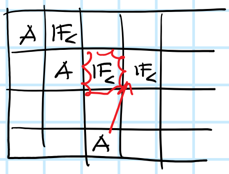
usare delayed branch

⇓

riorganizzare codice assembler D-RISC

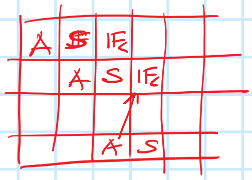
```
loop:  SUB  Ri, #1, Ri
       ...
       ADD  Ra, Rb, Rc
       IF<  Rc, RN, loop
       Ri --
```

k−1, loop+1



3t × 2 ist

$\varepsilon = 2/3$



3t × 3 ist

$\varepsilon = 1$

ERRATA CORRIGE (esecuzione programma for(i=0; i<N; i++) a[i] = b[i] * 2;)

```
       t CLEAR  Ri          <——> ADD R0 R0 Ri
loop:  t LOAD   RbaseB, Ri, R1
       t SHL    R1, #1, R1
       t STORE  RbaseA, Ri, R1
       t INC    Ri
       t IF<    Ri, RN, loop
         END
```

Tid = 6t

	0	1	2	3	4	5	6	7	8	9	10	11	12
IM	ADD	LD		SHL	ST			INC	IF<		END	LD	
IU		ADD	LD	LD	SHL	ST	ST	ST	INC	IF<	IF<	END	LD
DM			Ri		LD		R1		ST	Ri			
EU			ADD			LD	SHL			INC			

su IU: cont(Ri)=1, cont(Ri)=0, cont(R1)=1, cont(R1=0), cont(Ri)=1, cont(Ri=0)

dip EU-IU (ADD → LD)

dip EU-IU (SHL → ST)