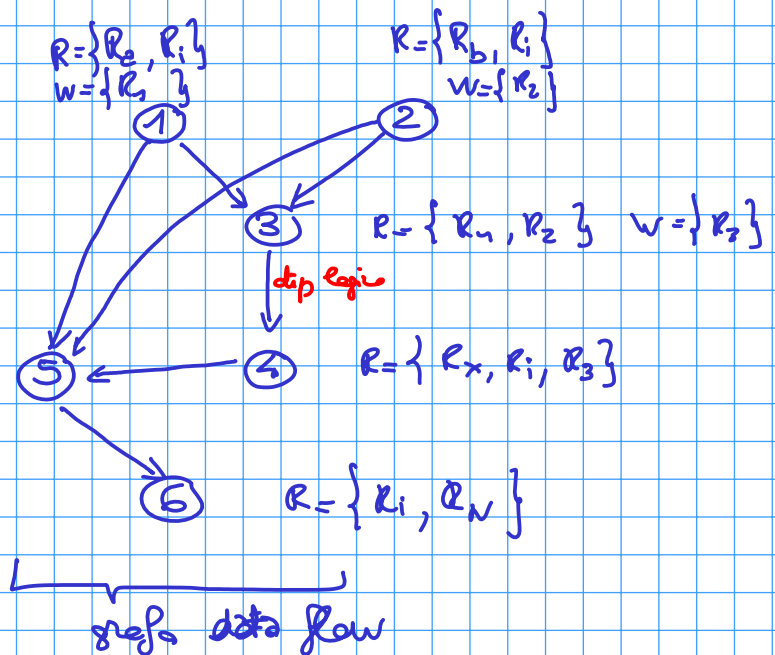
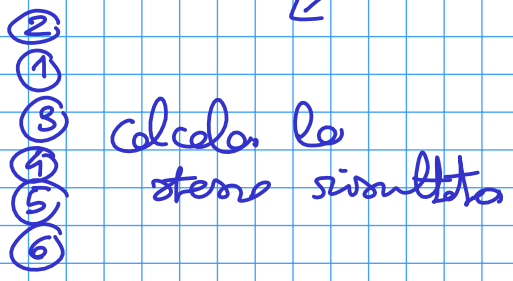
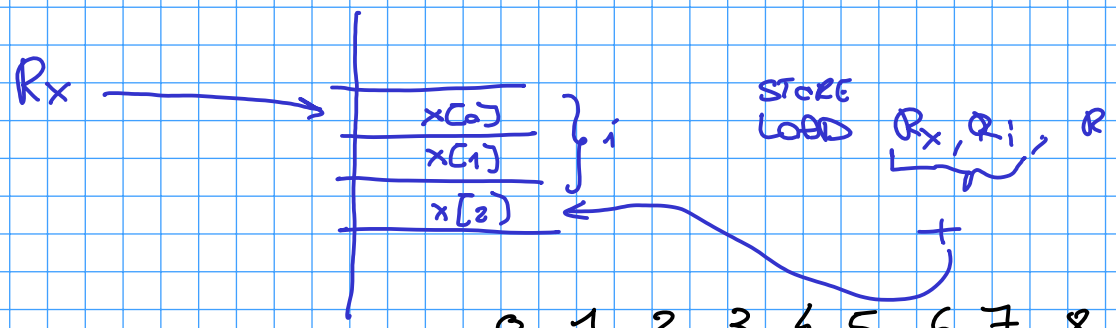
$x[i] = a[i] + b[i]$

```
loop: ① LOAD  R2 , Ri , R1      a[i]
      ② LOAD  Rb , Ri , R2      b[i]
      ③ ADD   R1 , R2 , R3
      ④ STORE Rx , Ri , R3
      ⑤ INC   Ri
      ⑥ IFc   Ri , RN , loop
```

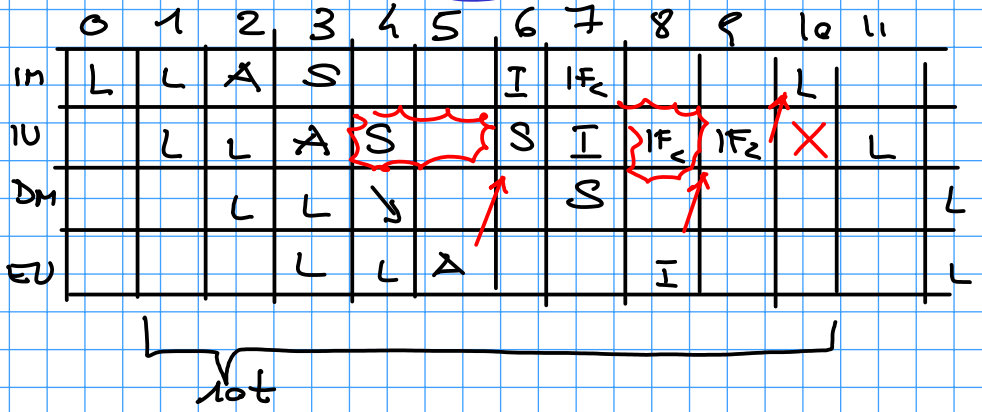




Utilizzare un Registro base modificato (decrementato di 1)

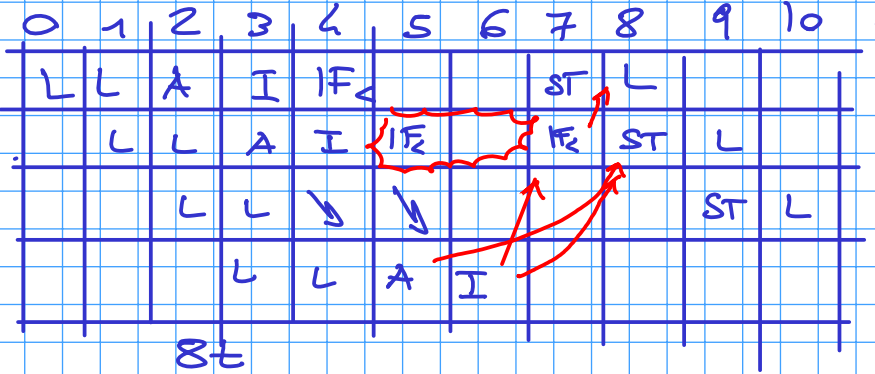



```
loop: ① LOAD  R2 , Ri , R1
      ② LOAD  Rb , Ri , R2
      ③ ADD   R1 , R2 , R3
      ④ STORE Rx , Ri , R3
      ⑤ INC   Ri
      ⑥ IFc   Ri , RN , loop
```



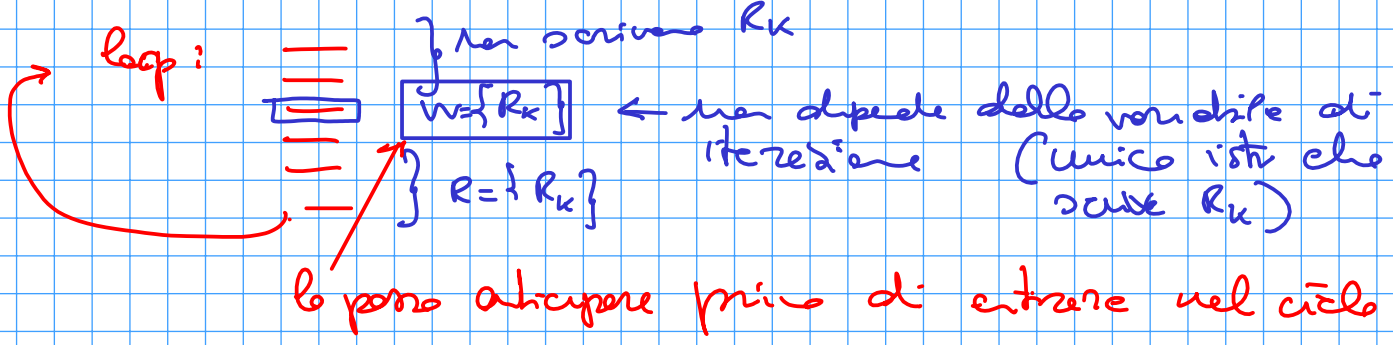
$\varepsilon = \frac{6}{10}$

```
loop: LOAD  Ri
      LOAD  Rb
      ADD   R1 , R2 , R3
      INC   Ri
      IFc   Ri , RN , loop , delayed
      STORE Rx , Ri , R3
```

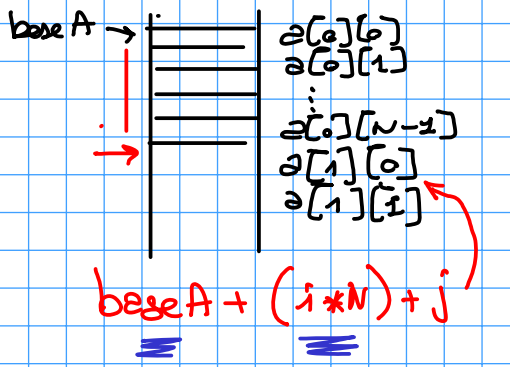


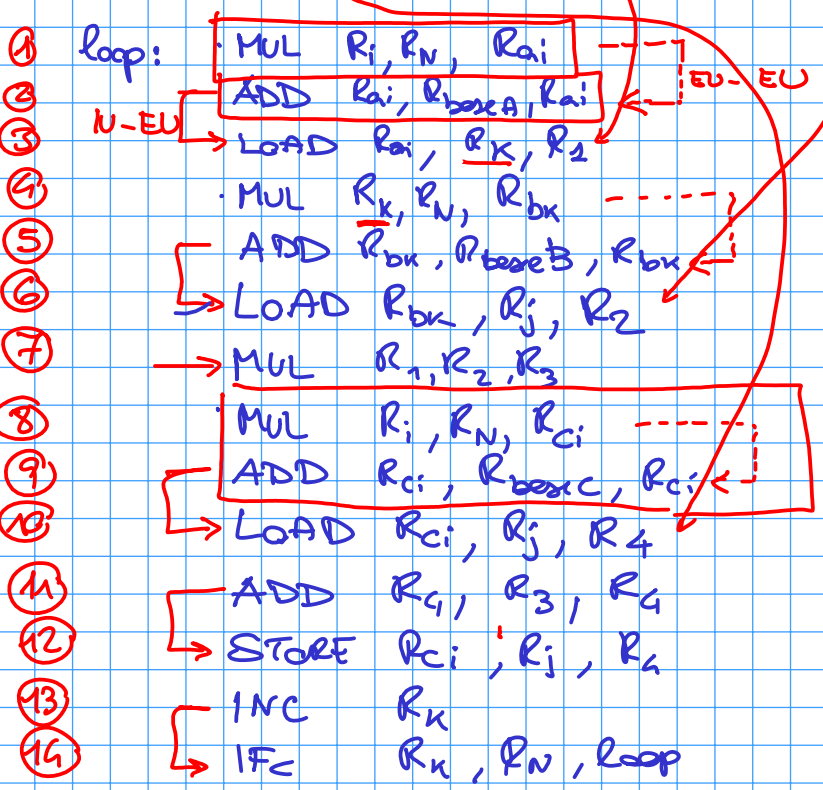
$\varepsilon = \frac{6}{8} = \frac{3}{4}$

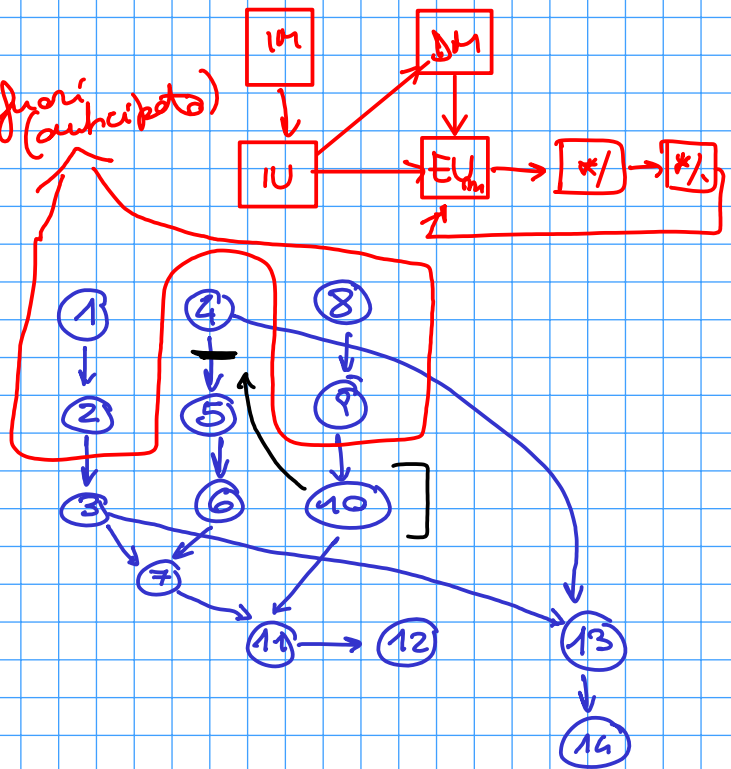
RIMOZIONE degli INVARIANTI

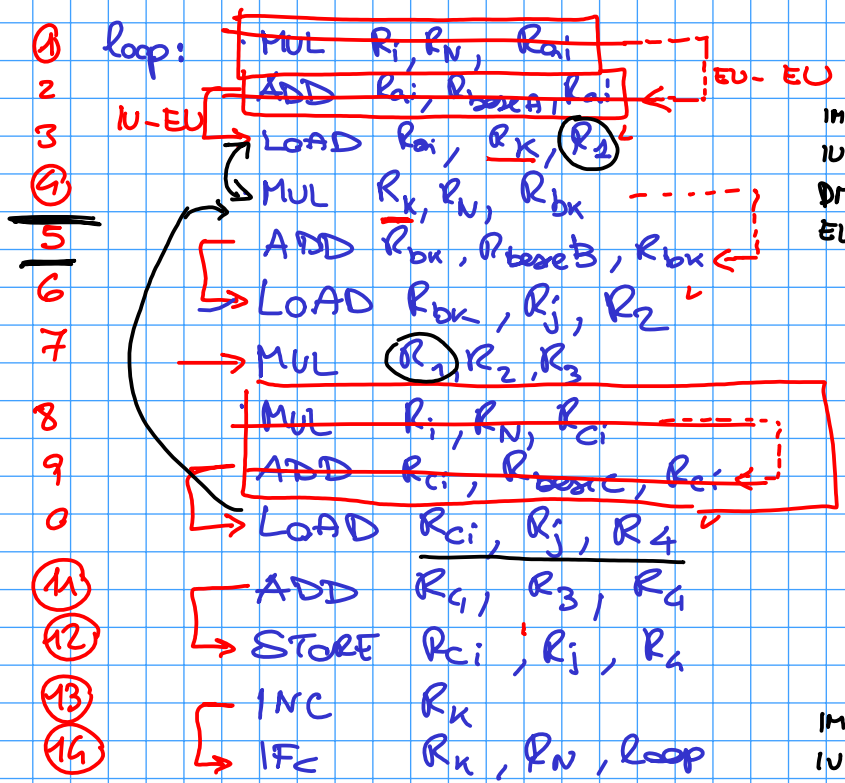


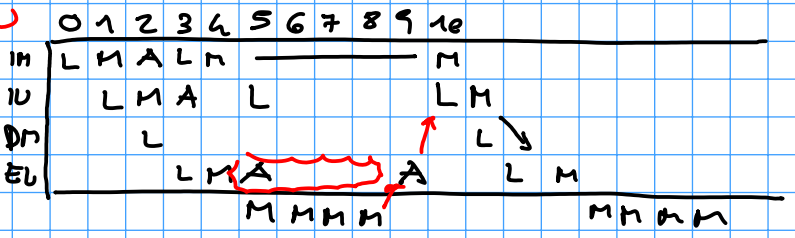

```
for(i  ....  )
  for(j ....  ){
    c[i][j] = 0
[[[ for (k          ){
      c[i][j] += a[i][k] * b[k][j];
```





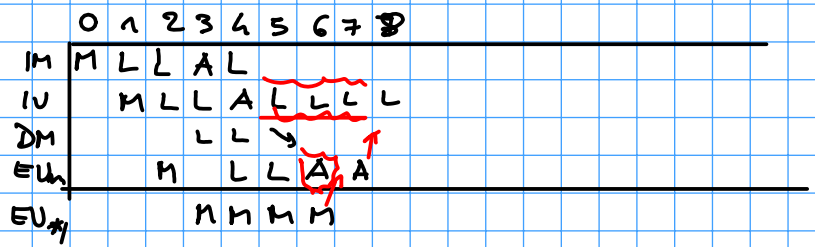






MUL RK RN RbK
LOAD Rai RK R1
LOAD Rci Rj R4
ADD RbK, RbaseB, RbK
LOAD RbK, Rj, R2

7 11 12 13 16

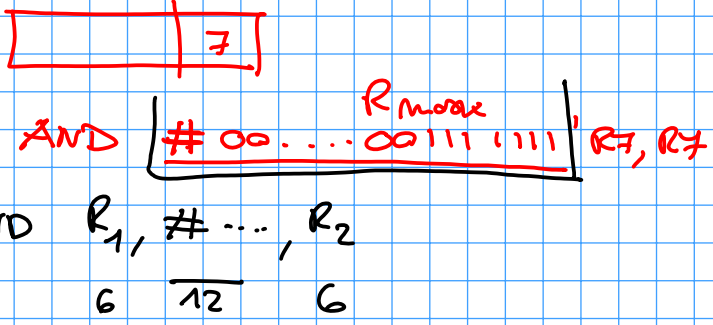


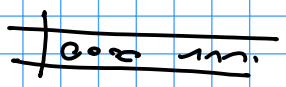
Si consideri il seguente frammento di codice, che lavora sui vettori di interi A, B, C e D, ciascuno da N=8K posizioni:

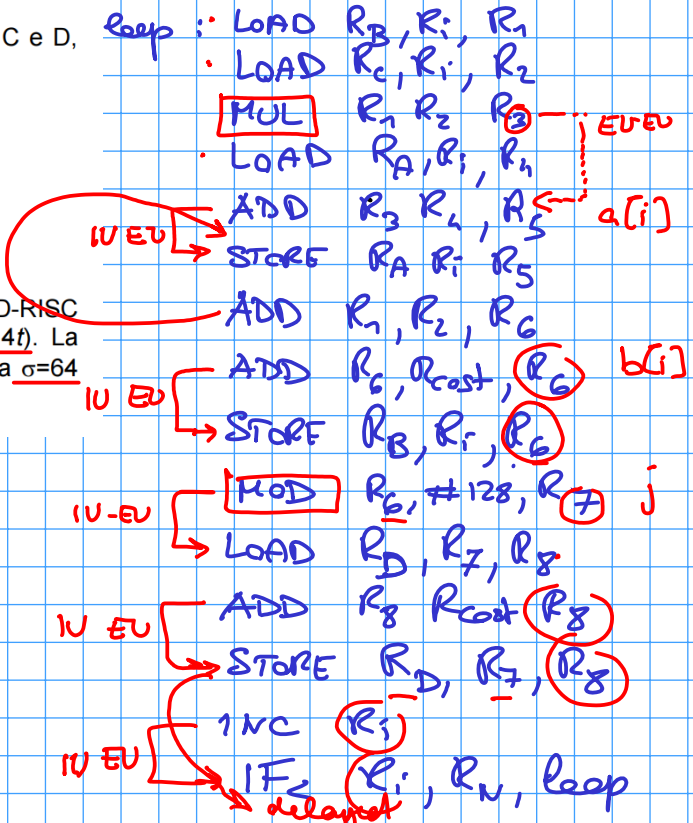
```
for(i=0; i<N; i++)  {
   a[i]=a[i]+b[i]*c[i];
   b[i]=b[i]+c[i]+cost;
   j=b[i] mod 128;
   d[j]=d[j]+cost;
}
```

Si supponga che il risultato della compilazione venga eseguito su un processore D-RISC pipeline con EU parallela (EU slave che esegue moltiplicazione o divisione in $4t$). La gerarchia di memoria comprende una cache di primo livello con linee (blocchi) da σ=64 parole e la memoria centrale interallacciata (m moduli).

Si chiede di







$T_{id} = kt$

WORKING SET

```
for(i=0; i<N; i++)  {
   a[i]=a[i]+b[i]*c[i];
   b[i]=b[i]+c[i]+cost;
   j=b[i] mod 128;
   d[j]=d[j]+cost;
}
```

PREFETCH

NON DEALLOCARE

codice	LOC	RIUSO
A	V	—
B	V	—
C	V	—
D	—	V

⇒ tutto: 1 linea di cache × A, B, C

tutto