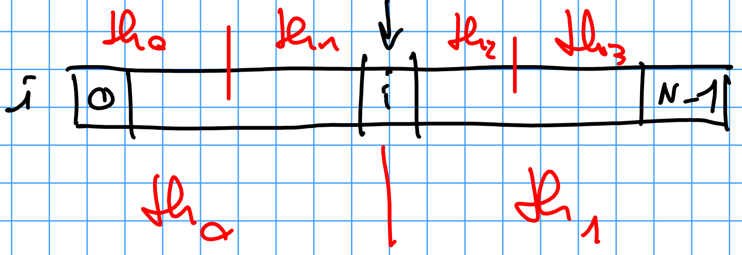
```
for(int i=0; i<N; i++)
    x[i] = f(x[i]);
```



13

```
#pragma omp parallel for
for(                )
    x[i] = f(x[i]);
```

icc

g++ -fopenmp

13/6/2016

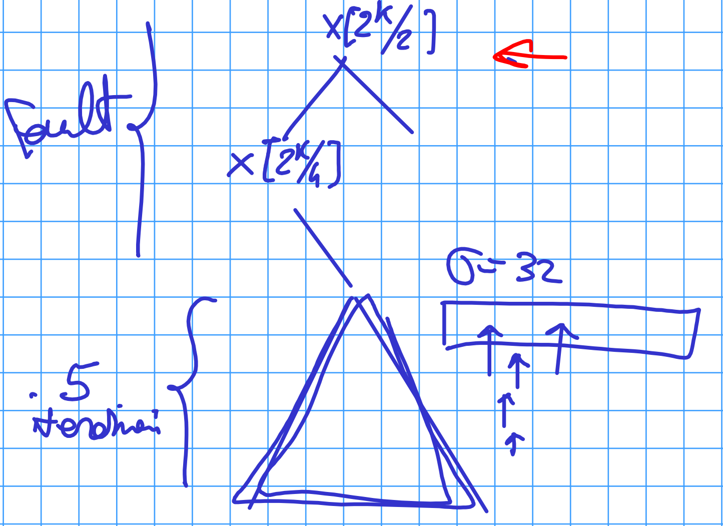
```
/* ricerca binaria: cerca x cominciando da metà e poi continuando nella metà alta o bassa */
int start = 0;
int stop  = n;
int found = -1;
while(stop - start > 1) {
        int i = start + (stop-start)/2;
         if(A[i] == x) {found = i; break;}
        if(x < A[i]) { stop = i; } else { start = i; }
}
if(A[start] == x) found = start;
if(A[stop] == x)  found = stop;

/* ricerca esaustiva: cerca x controllando le celle una per una fino a che l'array è finito o l'elemen
int found = -1;
for(int i=0; i<n; i++)
        if(A[i] == x) { found = i; break; }
```

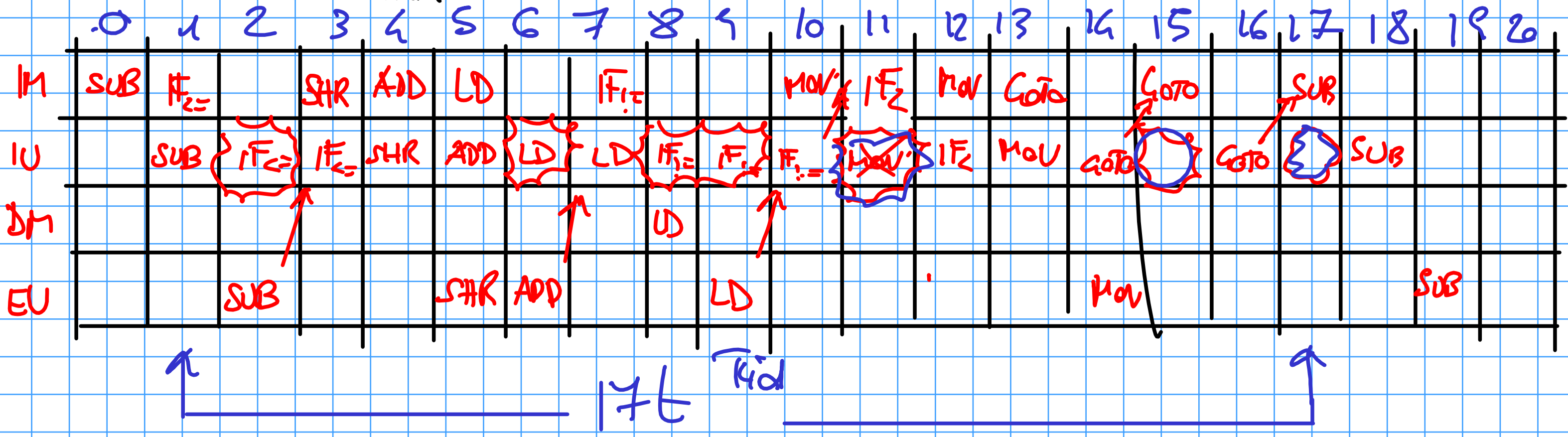
```
while :   SUB   Rstop, Rstart, Rtemp
          IF<=  Rtemp, #1, end
          SHR   Rtemp, #1, Rtemp
          ADD   Rstart, Rtemp, Ri
          LOAD  RbaseA, Ri, Rai
          IF!=  Rai, Rx, cont
then :    MOV   Ri, Rfound
          GOTO  end
          IF<   Rx, Rai, then1
else1 :   MOV   Ri, Rstart
          GOTO  cont1
then1 :   MOV   Ri, Rstop
cont1 :   GOTO  while
```

$T_{c\,id}$ (senza fault!)

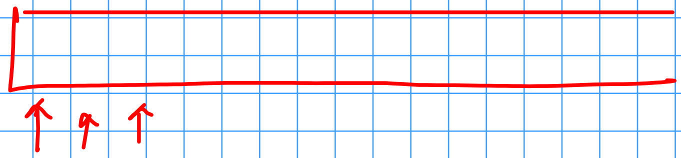
T_c (con
#fault * T_{trasf}

```
while :   SUB   Rstop, Rstart, Rtemp
          IF<=  Rtemp, #1, end
          SHR   Rtemp, #1, Rtemp
          ADD   Rstart, Rtemp, Ri
          LOAD  RbaseA, Ri, Rai
          IF!=  Rai, Rx, cont
then :    MOV   Ri, Rfound
          GOTO  end
cont:     IF<   Rx, Rai, then1
else1 :   MOV   Ri, Rstart
          GOTO  cont1      GOTO while
then1 :   MOV   Ri, Rstop
cont1 :   GOTO  while
```





$$T_{cid} + \approx k \cdot T_{trasf}$$



cache con prefetch

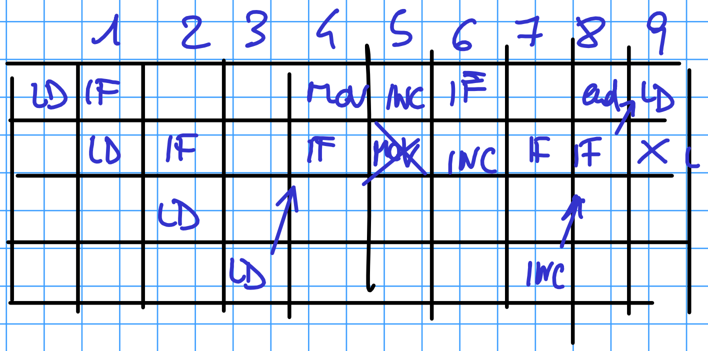
```
/* ricerca esaustiva: cerca x controllando
int found = -1;
for(int i=0; i<n; i++)
        if(A[i] == x) { found = i; break; }
```

A[0] . . . . A[31]

A[32] A[63]

A[64] . . . .

```
①  loop :  LOAD   R_baseA , R_i , R_ai
②          IF_=   R_ai , R_x , cont
           MOV    R_i , R_found
           GOTO   end
③  cont :  INC    R_i
④          IF_<   R_i , R_N , loop
   end :   ....
```



$$\underbrace{9t \times \sigma}_{\geq\ T_{loop}}$$

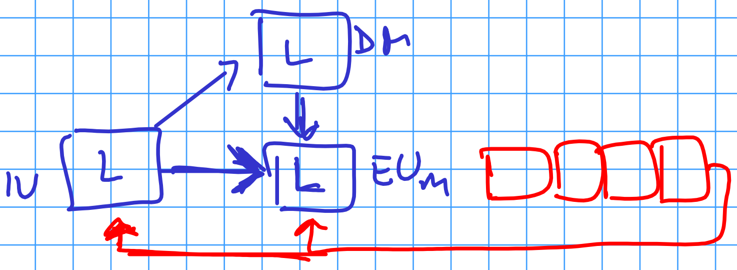
$$\begin{tikzcd}X & Y \arrow[r, bend right, leftrightarrow] & Z\end{tikzcd}$$

```
for(i=0; i<N; i++) {
    for(j=0; j<N; j++) {
        x[j] = x[j] + x[j]*x[j];
        sum[i] = sum[i] + x[j];
    }
    somma = somma + sum[i];
}
```

```
loop i :  CLEAR   Rj
loop j :  LOAD    Rbasex, Rj, Rxj
          MUL     Rxj, Rxj, R2xj
          ADD     Rxj, R2xj, Rxj          dip EU EU
          STORE   Rbasex, Rj, Rxj         dip IU-EU
          LOAD    Rbasesum, Ri, Rsi
          ADD     Rxj, Rsi, Rsi
          STORE   Rbasesum, Ri, Rsi       IU-EU
          INC     Rj
          IF<     Rj, RN, loop j
          ADD     Rsomma, Rsi, Rsomma
          INC     Ri
          IF<     Ri, RN, loop i:
```

```
loopi :  CLEAR  Rj
loopj :  LOAD   Rbasex, Rj, Rxj
         MUL    Rxj, Rxj, R2xj     ┐ dip EU EU
         ADD    Rxj + R2xj, Rxj   <┘
         STORE  Rbasex, Rj, Rxj   <┘ dip IU-EU
         LOAD   Rbasesum, Ri, Rsi
         ADD    Rxj, Rsi, Rsi      ┐
         STORE  Rbasesum, Ri, Rsi <┘ IU-EU
         INC    Rj
         IF<    Rj, RN, loopj
         ADD    Rsomma, Rsi, Rsomma
         INC    Ri
         IF<    Ri, RN, loopi
```

①



<*, xj.xj, xj> <LOAD, "xj">

Tid 19t

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20		
IM	L	M	A	ST							LD	A	ST			INC	IF<		ADD	LD			
IU		L	M	A	ST						ST	LD	A	ST		ST	INC	IF<	IF<	~~AD~~	LD		
DM			L									ST	LD				ST					LD	
EUm				L	M	A				A				LD	A			INC					LD
EU*						M	M	M	M														

```
loopi :  CLEAR  Rj
loopj :  LOAD   Rbasex, Rj, Rxj
         MUL    Rxj, Rxj, R2xj     ┐ dip EU EU
         ADD    Rxj + R2xj, Rxj   <┘
         STORE  Rbasex, Rj, Rxj   <┘ dip IU-EU
         LOAD   Rbasesum, Ri, Rsi
      A1 ADD    Rxj, Rsi, Rsi      ┐
         STORE  Rbasesum, Ri, Rsi <┘ IU-EU
         INC    Rj
         IF<    Rj, RN, loopj, delayed < STORE Rbasex', Rj, Rxj
         ADD    Rsomma, Rsi, Rsomma
         INC    Ri
         IF<    Ri, RN, loopi:
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
IM	L	M	A	L	A1	INC	ST	IF					IF	ST	L			
IU		L	M	A	L	A1	INC	ST					ST	IF	ST	L		
DM			L	↘	↘	L								ST		ST	L	
EUm				L	M	A				A	L	A,	INC					L
EUx						M	M	M	M									

14t

```
    Loopi : CLEAR  Rj
(1) Loopj : LOAD   Rbasex, Rj, Rxj
(2)         MUL    Rxj, Rxj, R2xj
(3)         ADD    Rxj, R2xj, Rxj        dip EU EU
(4)         STORE  Rbasex, Rj, Rxj       dip IU-EU
(5)         LOAD   Rbasesum, Ri, Rsi
(6)         ADD    Rxj, Rsi, Rsi
(7)         STORE  Rbasesum, Ri, Rsi     IU-EU
(8)         INC    Rj
(9)         IF<    Rj, RN, Loopj
            ADD    Rsomma, Rsi, Rsomma
            INC    Ri
            IF<    Ri, RN, Loopi:
```

```
    1   2   3   4   5   6   7   8   9   10  11  12  13  14
L   M   L   I   A   A   S                   IF  S   L
    L   M   L   1   A   A   S               S   IF  S   L
        L       L                               S       S   L
            L   M   L   I   A       A   A                       L
                    M   M   M   M
```