Problemi di approssimazione

Corso di Laboratorio di Calcolo

La somma di N numeri

```
#define N 10000000
float x = 7., Sf = 0.;
int i, k = 7, Si = 0 ;
for (i = 0; i < N; i++) {
   Sf += x;
   Si += k;
}
printf("%d %.0f\n",Si,Sf);
```

$$\sum_{i=1}^{10^7} 7 = 7 \times 10^7 = 70M$$

La somma di N numeri

```
#define N 10000000
float x = 7., Sf = 0.;
int i, k = 7, Si = 0 ;
for (i = 0; i < N; i++) {
   Sf += x;
   Si += k;
}
printf("%d %.0f\n",Si,Sf);
```

$$\sum_{i=1}^{10^7} 7 = 7 \times 10^7 = 70M$$

```
> gcc -o somma.exe somma.c
> ./somma.exe
> 70000000  77603248
```

Cosa avviene in memoria

$x=7=2^2(1+2^{-1}+2^{-2})$

$i=2396746$ $Sf=16777222=2^{24}(1+2^{-22}+2^{-23})$

x = 0 10000001 (1) 11000000000000000000000

2^2

Sf = 0 10010111 (1) 00000000000000000000011

2^{24}

Cosa avviene in memoria

$x=7=2^2(1+2^{-1}+2^{-2})=2^2 2^{22}(2^{-22}+2^{-23}+2^{-24})$

$i=2396746 \quad Sf=16777222=2^{24}(1+2^{-22}+2^{-23})$

sposta di 21 bit

x = 0 10010111 (0) 00000000000000000000111

2^{24}

S = 0 10010111 (1) 00000000000000000000011

Cosa avviene in memoria

$x=7=2^2(1+2^{-1}+2^{-2})$

$i=2396746$ $S=16777222=2^{24}(1+2^{-22}+2^{-23})$

x = 0 10010111 (0) 00000000000000000000011

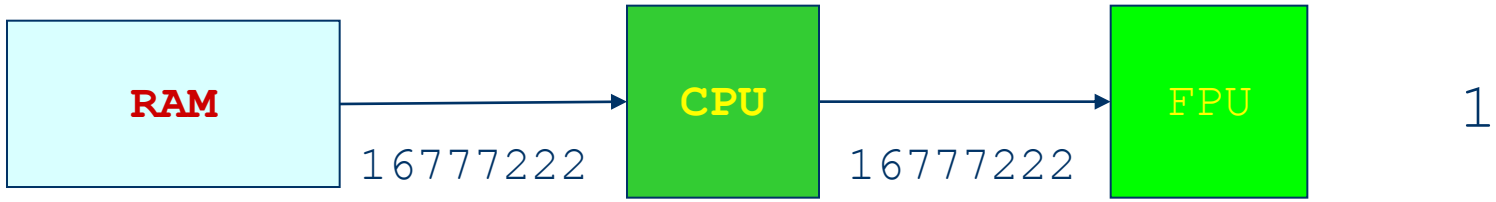
Sf= 0 10010111 (1) 00000000000000000000011

Sf= 0 10010111 (1) 00000000000000000000110

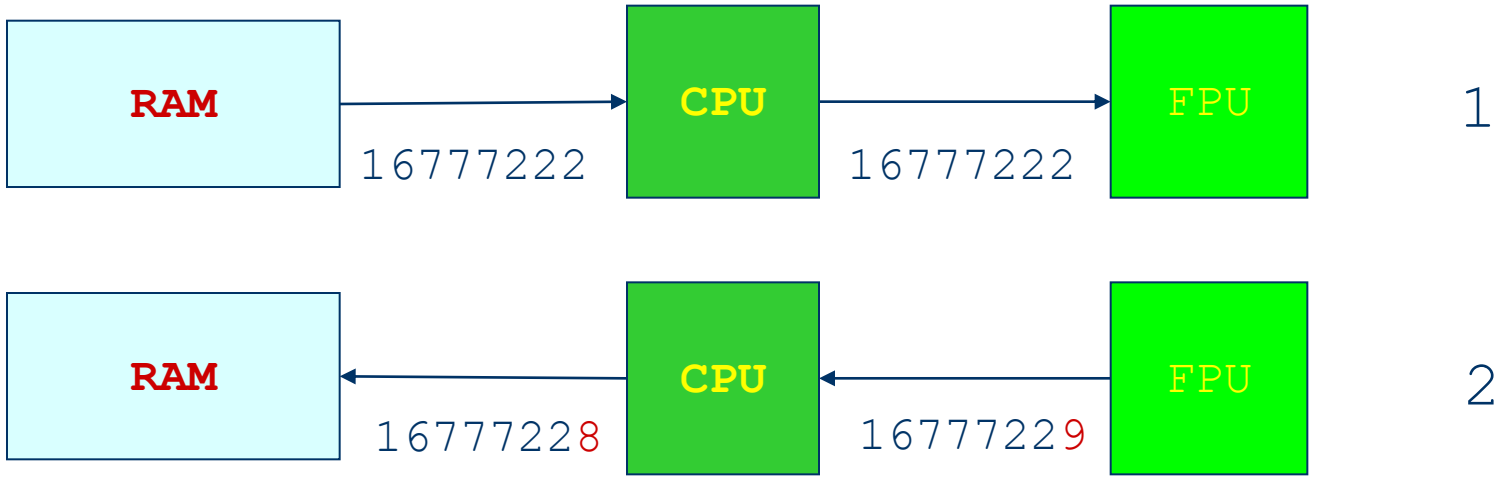
$Sf= 2^{24}(1+2^{-21}+2^{-22})=16777228$ invece di 16777229

Perché S è maggiore di 70 milioni?

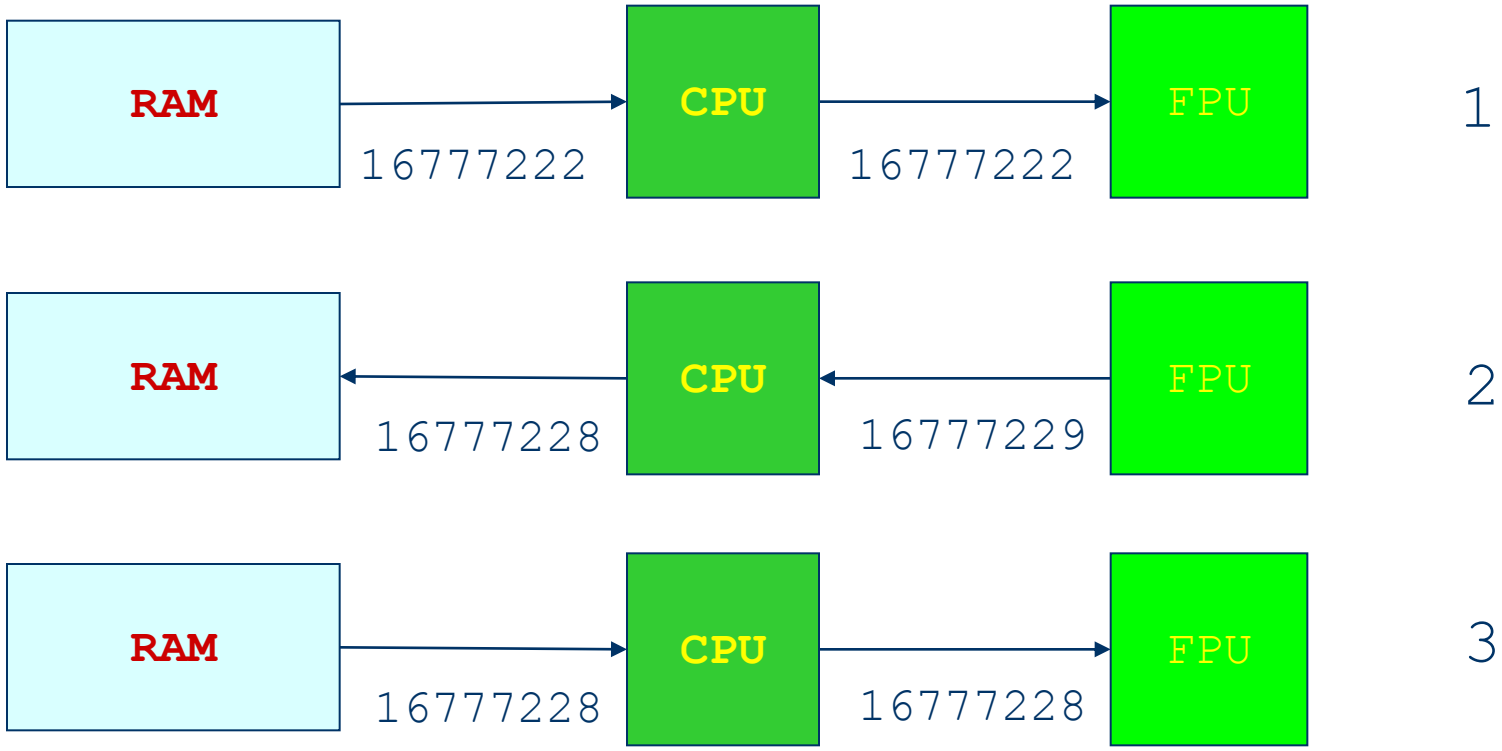
La FPU usa 80 bit



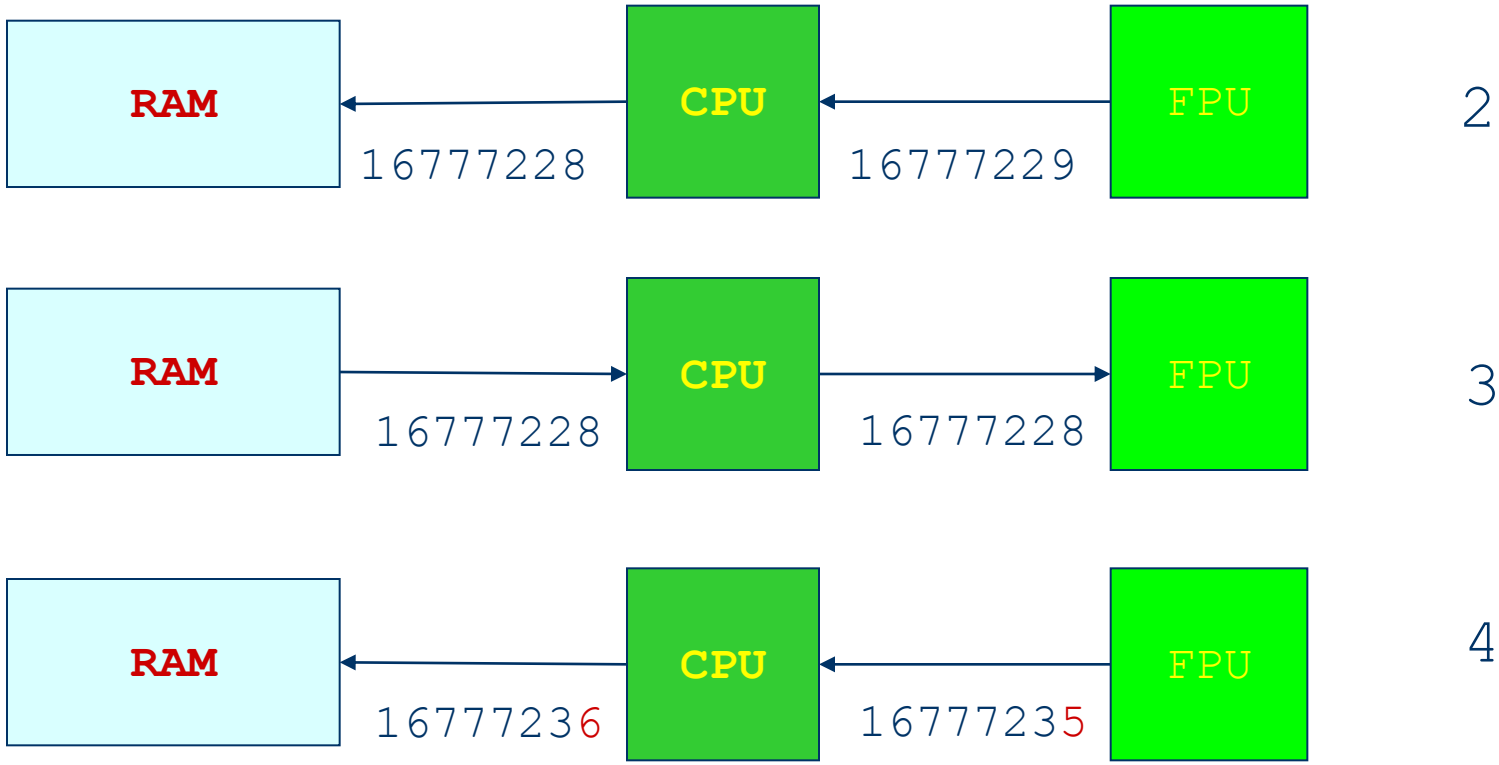
Perché S è maggiore di 70 milioni?



Perché S è maggiore di 70 milioni?



Perché S è maggiore di 70 milioni?



Come si controlla?(1)

```
float sum=0., corr=0., x=7.;
int i;
for (i=0; i<N; i++) {
  float tmp, y;
  y = corr + x;   x=7, y=7
  tmp = sum + y; tmp = 7 tmp = 14
   corr = (sum - tmp) + y; corr = 0
  sum = tmp; sum = 7 sum = 14
}
sum += corr;
```

Come si controlla?(2)

```
float sum=0., corr=0., x=7.;
int i;
for (i=0; i<N; i++) {
  float tmp, y;
  y = corr + x;    /* corr = 0 → y = x */
  tmp = sum + y;   /* tmp = sum + x */
  corr = (sum - tmp) + y; /* sum - sum - x + x */
  sum = tmp;       /* sum = tmp */
}
sum += corr;
```

Come si controlla?(3)

```
float sum=0., corr=0., x=7.;
int i;
for (i=0; i<N; i++) {
   float tmp, y;
   y = corr + x;
   tmp = sum + y;
   corr = (sum - tmp) + y;
   sum = tmp;
}
sum += corr;
```

Se $\texttt{tmp} = \texttt{sum}+x+\delta$

$\texttt{corr}=\texttt{sum}-(\texttt{sum}+x+\delta)+x=-\delta$

Al ciclo successivo $y=-\delta+x$

Come si controlla?(4)

```
float sum=0., corr=0., x=7.;
int i;
for (i=0; i<N; i++) {
  float tmp, y;
  y = corr + x;  /* corr=0, y=7 */
  tmp = sum + y; /* sum=16777222, tmp=16777228 */
  corr = (sum - tmp) + y; /*corr=(16777222-
                            16777228)+7=1 */
  sum = tmp;     /* sum=16777228 */
}
sum += corr;
```

Come si controlla?(5)

```
float sum=0., corr=0., x=7.;
int i;
for (i=0; i<N; i++) {
  float tmp, y;
  y = corr + x;   /* corr=1, y=8 */
  tmp = sum + y; /* sum=16777228, tmp=16777236 */
  corr = (sum - tmp) + y; /*corr=(16777228-
                            16777236)+7=-1 */
  sum = tmp;      /* sum=16777236 */
}
sum += corr;
```

Come si controlla?(6)

```
float sum=0., corr=0., x=7.;
int i;
for (i=0; i<N; i++) {
  float tmp, y;
  y = corr + x;  /* corr=-1, y=6 */
  tmp = sum + y; /* sum=16777236, tmp=16777242 */
  corr = (sum - tmp) + y; /*corr=(16777236-
                            16777242)+7=1 */
  sum = tmp;     /* sum=16777242 */
}
sum += corr;
```