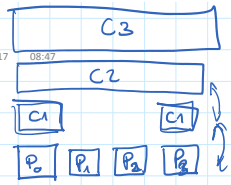
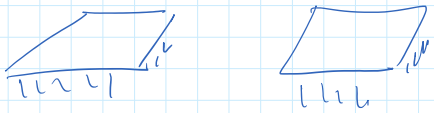
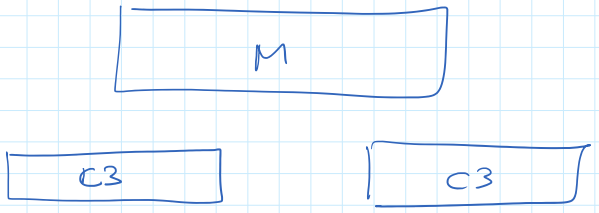
lunedì 20 novembre 2017 08:47

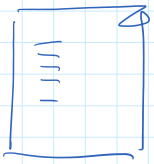




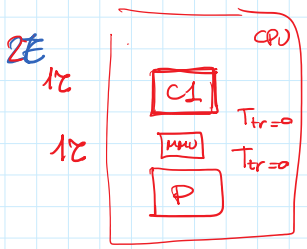


Modelli di costo

$T_{id} + (\text{costo fault di cache})$



$T_c = a\tau + b t_2$



$T_c = 100\tau + 90 t_2$

cache 1° livello comune x dati e progr
di tipo esec. in insieme

t_2 ? in caso di hit

$T_c = 100\tau + 90(2\tau)$ (3τ)

tempo di completamento ideale

costo del trattamento dei fault

(# fault) · (costo del singolo fault)

fault → dipende dal codice (e in parte dell'hw)

costo del singolo fault → dipende dall'hw

WS

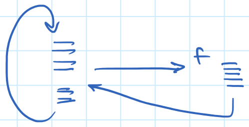
Capacità

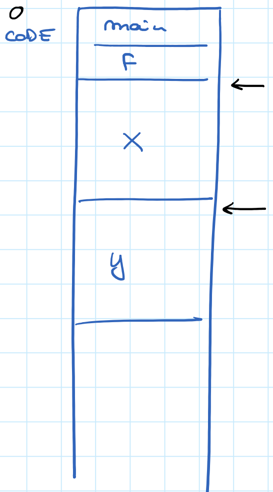
fault

lunedì 20 novembre 2017 09:37

FAULT → CODICE / DATI

for( ) {
  x[i] = f(y[i])
}





+ proprietà di località
+ " " riuso

→ costo del caricamento "iniziale" — { fault fisiologici: $\frac{\#codice}{\sigma}$
→ si riesce a mantenerlo in cache

(se non riusciamo) aggiungere #fault

Pag 4K = 12 bit

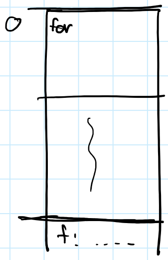
#PL	OFF_{PAG}
#PF	OFF_{PAG}

IND LOG

TAG	#linea	3

10 bit; off $\sigma = 8$ 3bit

```
      0  for :  LOAD  Rbasey, Ri, Ryi      <- input parametri di f
      1         CALL  Rf, Rret
area k 2        STORE Rbasex, Ri, Rres     <- output param di f
      3         INC   Ri
      4         IF<   Ri, RN, for
      5         END
      6  f  :   ...
      7         ...
                GOTO  Rret
area k+1 :
```



$\sigma = 8$

cache ad indirizzamento diretto

$$\#fault = N\,(1\,fault_{for} + 1\,fault_{f})$$

N = n° iter

DATI

R	W	R	W	R	W
y_0	x_0	y_1	x_1	y_2	x_2

vettori da N posizioni

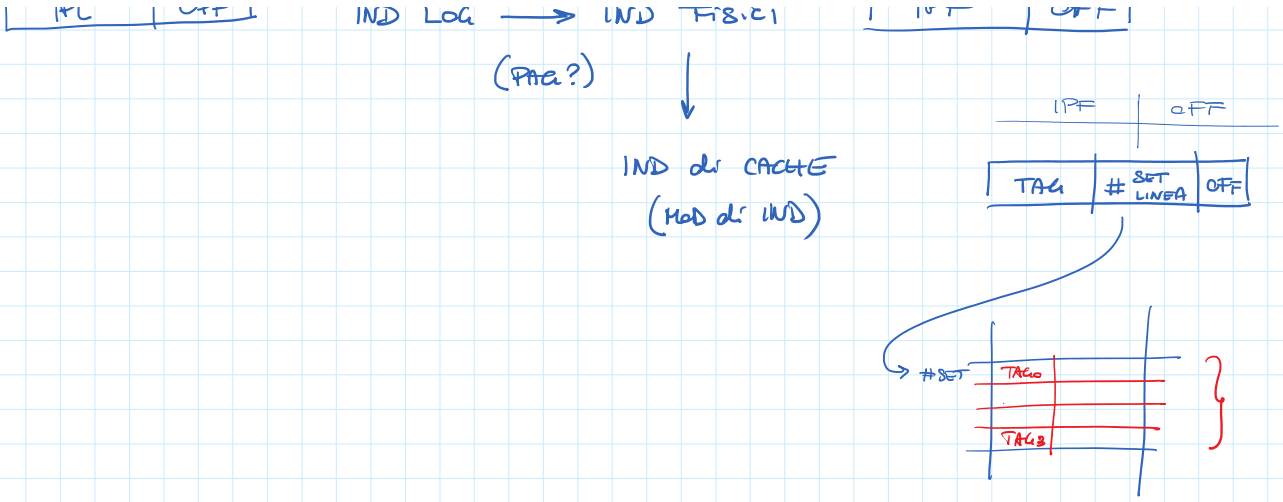
#fault fisiologici + costo di fault non fisiologici

x, y → $2\left(\frac{N}{\sigma}\right)$

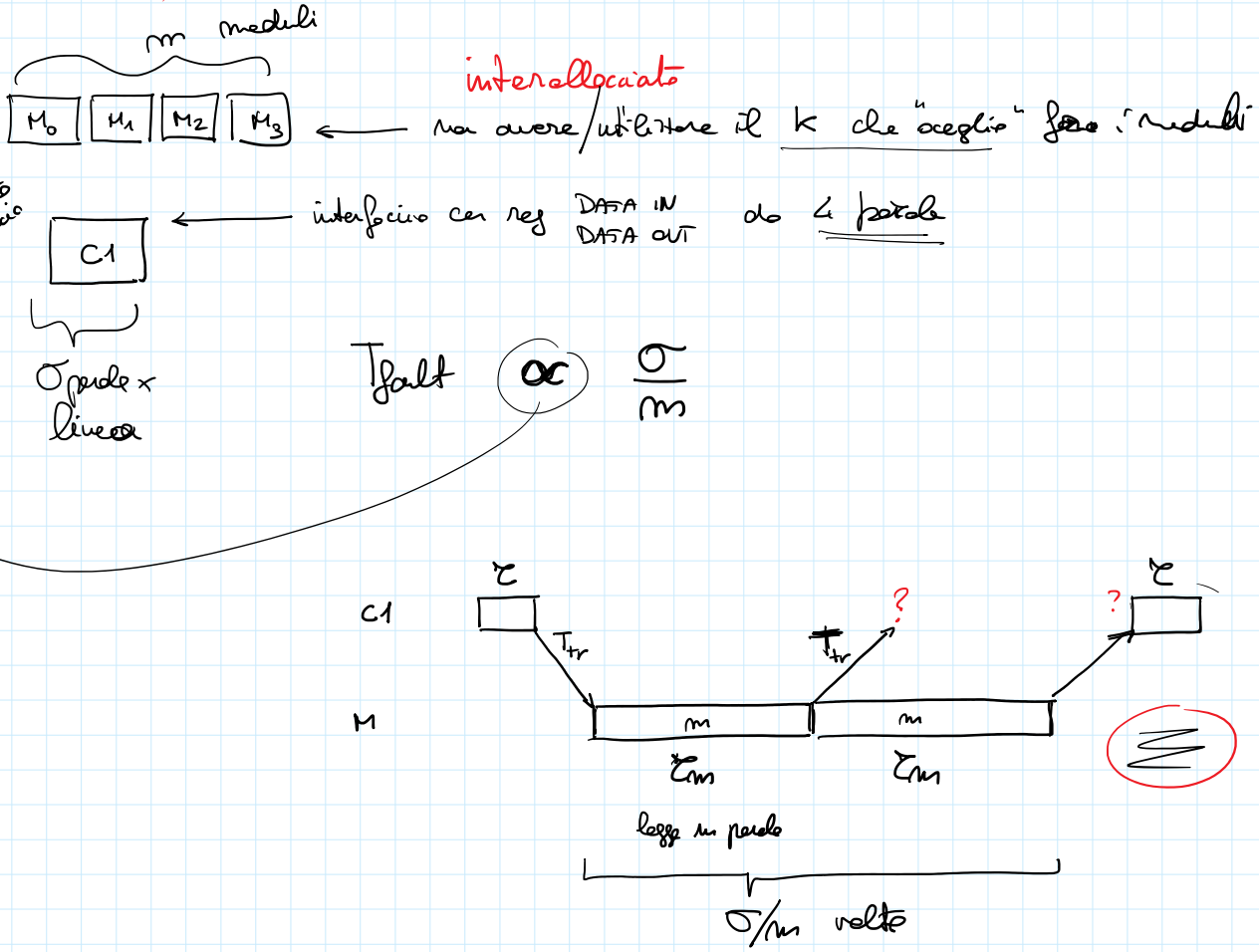
IPL	OFF

IND LOG → IND FISICI (PAG?)

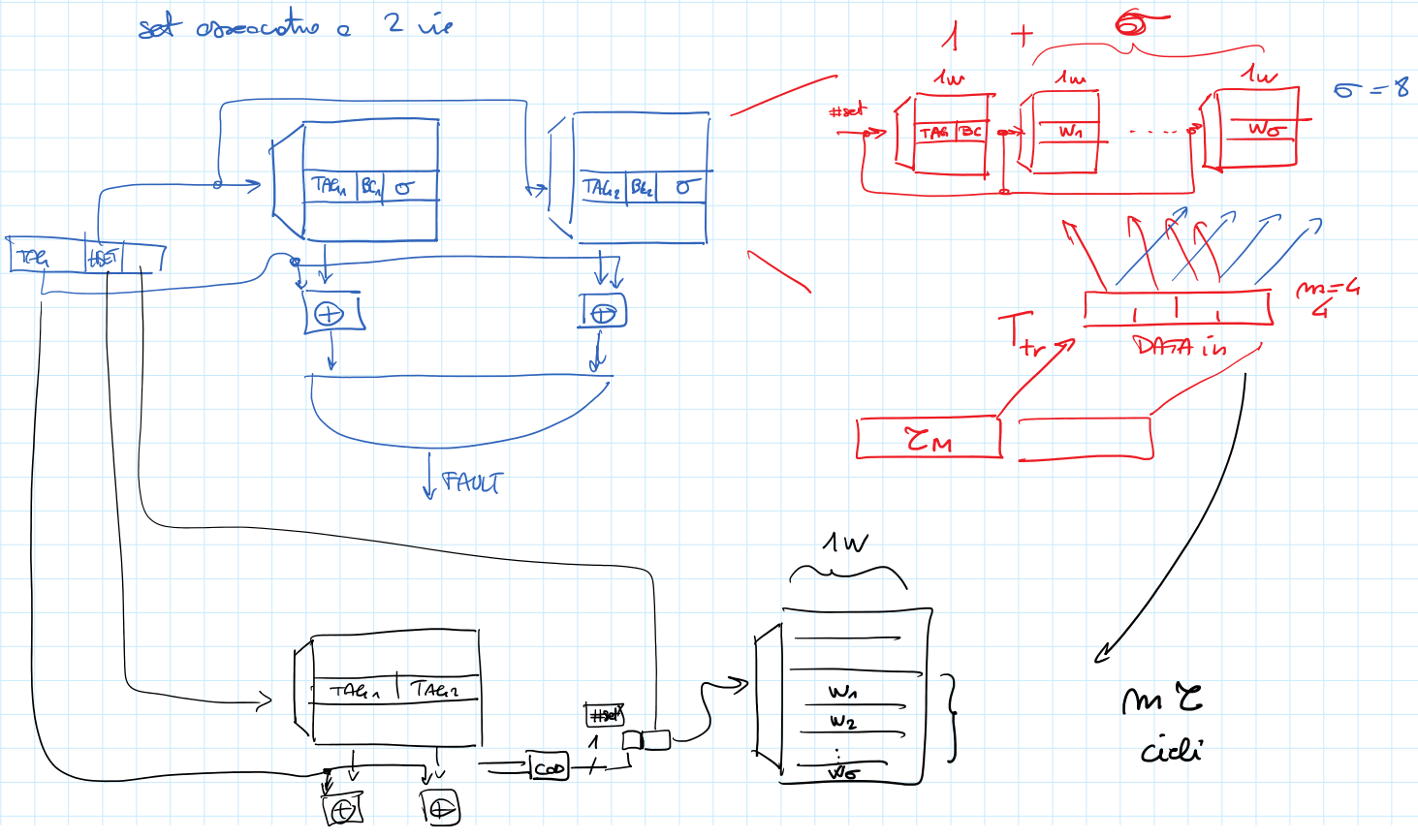
IPF	OFF

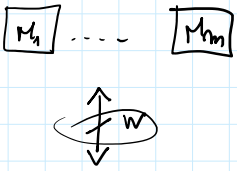

IND LOG
IND FISICI
(PAG?)
IND di CACHE
(MOD di IND)
IPF
oFF
TAG
SET LINEA
OFF
SET
TAG0
TAG1

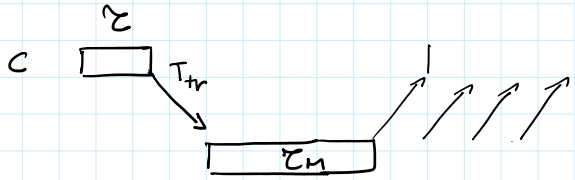
T_{fault} (T_{trasf})

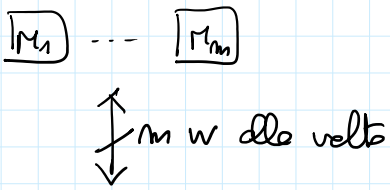


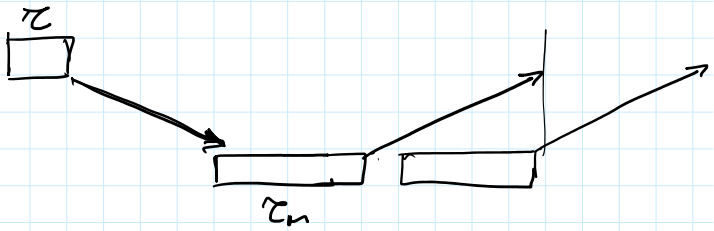
$$\left(2T_{tr}+2\tau\right)+\frac{\sigma}{m}\tau_m$$



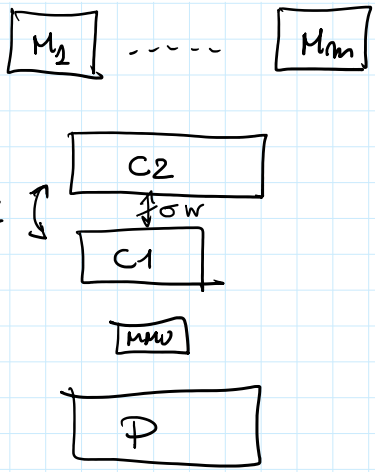








$T_{fault} \propto \frac{\sigma}{m}$ $\frac{\sigma}{m} T_{tr}$



$\sigma_2 \neq \sigma_1$ $\sigma_2 > \sigma_1$ $\sigma_2 = 16$

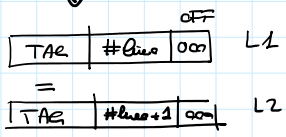
$\sigma_1 = 8$

località (C1 - C2) ?

se accedo L1 in C1
è probabile che accederò a L2 subito dopo

L2 è "consecutivo" a L1

Ind. diretto →



lunedì 20 novembre 2017 10:35

$T_c = T_{id} + \#fault \cdot T_{fault}$

Politiche di gestione delle cache

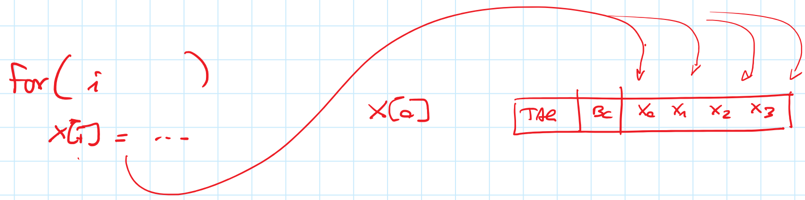
↳ rimpiazzi (LRU)

↳ scritture

- LETTURA (LOAD, ciclo)
  - IND → M → W
  - fault ⇒ la parola va cercata nel livello superiore della gerarchia
- SCRITTURA (STORE)
  - IND, VAL → M → OK
  - fault ⇒ ?

↓

scritture: WB, WT



fault?

- se scrivo tutta la linea ⇒ il fault consiste solo nell'allocare la linea
  1. trovo "ultimo"
  2. nella TAG e P=1
- se scrivo 1 solo valore ⇒ fault "normale"

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

#fault ?

$\frac{\#istr}{\sigma} + 2\frac{N}{\sigma}$

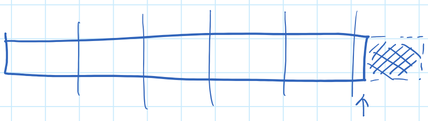
$T_c = T_{id} + \#fault \cdot T_{fault}$

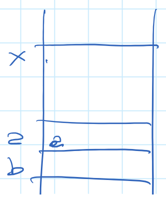
$\frac{\#istr}{\sigma} + 1\frac{N}{\sigma}$ xché non considero le scritture

Hp1 vettore X "allineato" alle linee di cache

base x = [... | 0 ... 0] ($\log_2 \sigma$ bit)

Hp2 padding x l'ultima linea di cache

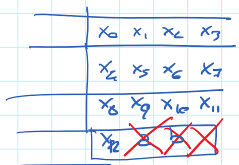


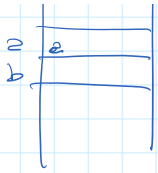



```
{ int x[13];
  float a,b;
```

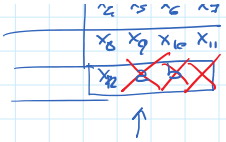
⇓

$\sigma = 4$





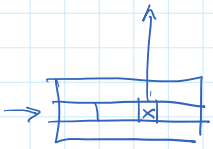
float a, b;



WRITE BACK

∀ scrittura deve essere riportata al livello superiore

↖ scrittura asincrona



1) ricevo la richiesta di STORE
2) scrivo in C e ordino la scrittura in M
3) do risposta (ESITO) al P

P STORE ind, x

WRITE THROUGH

∀ scrittura avviene solo nel livello di cache considerato
+
quando la linea diventa "vittima" x rimpiazzo
⇒ riporto la modifica nel livello superiore

```
for(i        )
   x[i] = i;
```

```
for: STORE R_base+x, Ri, Ri      write back
     INC Ri
     IF< Ri, RN, for
     END
```

T_C di 1 iterazione

$$3(c_{l0}+c_{l1}) + (T_{store} + T_{inc} + T_{IF<})$$
$$3(2\tau + t_2) + (2\tau + t_2 + 1\tau + 2\tau)$$
$$6\tau + 3t_2 + 5\tau + t_2 \Rightarrow 11\tau + 4t_2$$
$$4t_2 \rightarrow 8\tau$$
$$23\tau$$