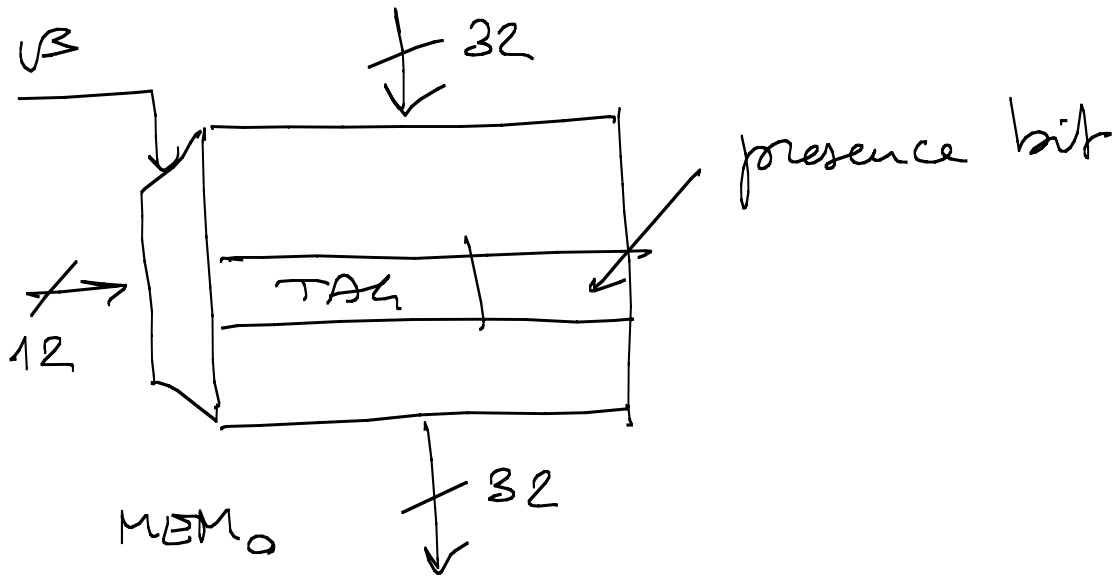


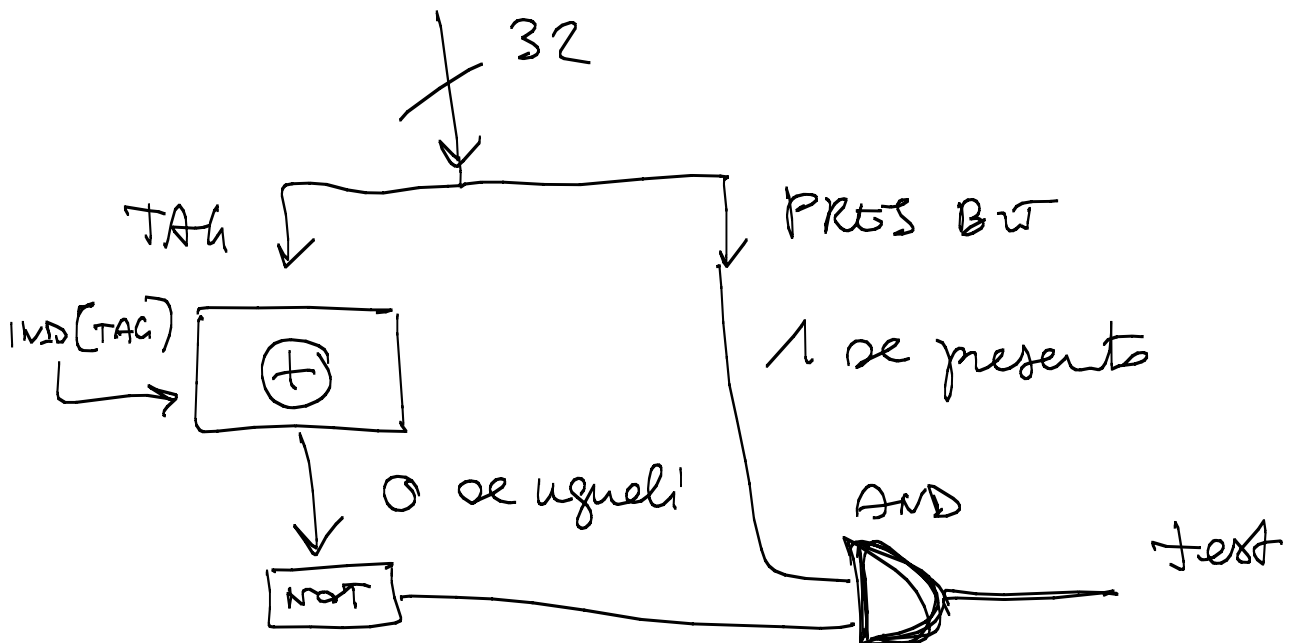
Soluzione con accesso parallelo a M

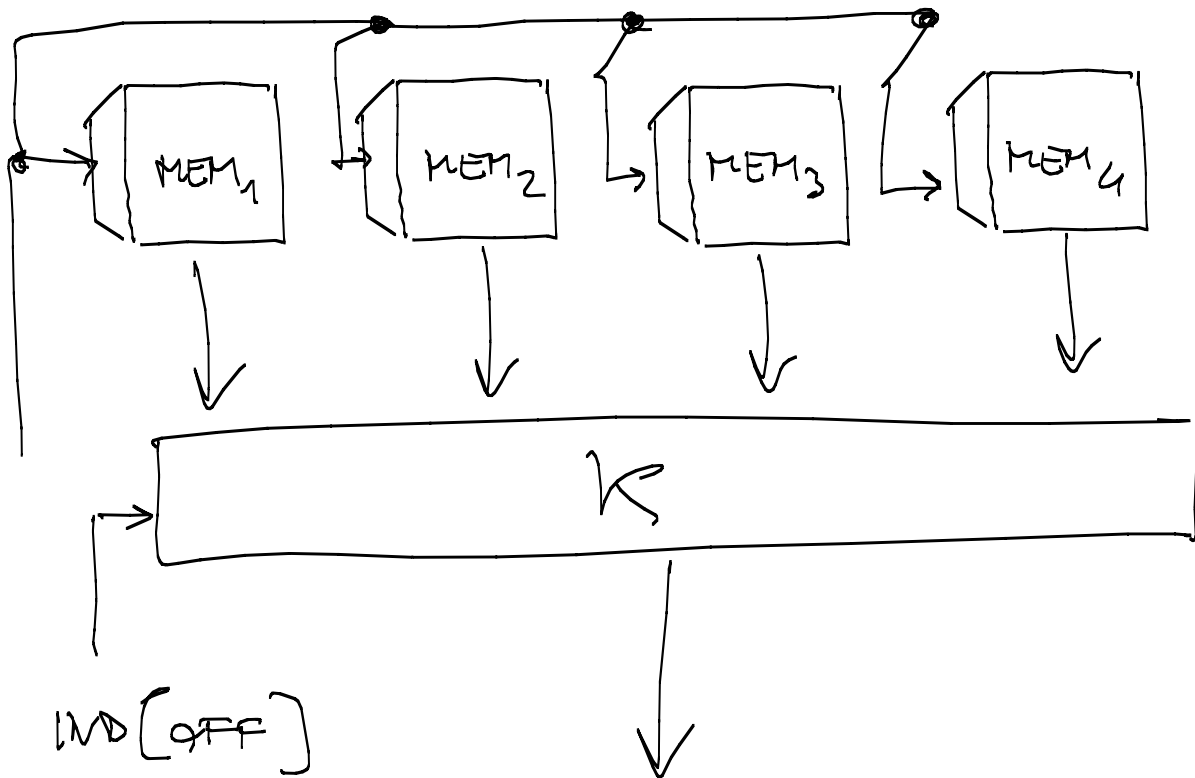
Note Title

05/06/2007



modulo da 16K parole che
contiene TAG e bit di presenza





lettura di 1
parola dello
cache

μ -programma (schema con commenti)

Note Title

05/06/2007

\emptyset . (RDY_p , $\underbrace{TEST(IND, MEM_0)}_{\text{controlla che}}$, $OP = \emptyset \text{ ---}$) nop, \emptyset

controlla che
 $IND[TAG] = MEM_0[IND[BLOCK]][TAG]$

e che

$MEM_0[IND[BLOCK]][PRESBIT] = 1$

e restituisce

1 α vero

0 α falso

(= 1 1 \emptyset) lettura, cache hit

$MEM_{IND[OFF]}[IND[BLOCK]] \rightarrow OUT_p$,

reset RDY_p , set ACK_p ,

$\emptyset \rightarrow ESTOP_p$, \emptyset .

\downarrow modulo di memoria indirizzato
dall'OFFSET: uso un K

(= 111)

scrittura, cache hit
scrittura full in M
& implementare il write through

$MEM_1[IND[Block]] \rightarrow OUT_m[0],$

$MEM_4[IND[Block]] \rightarrow OUT_m[3],$

$IND_p[31..2]00 \rightarrow IND_m,$

$OP_p \rightarrow OP_m, \text{ Set } RDY_m,$

copiare il blocco verso le M

$IN_p \rightarrow MEM_{IND[OFF]}[IND[Block]],$

scrivi nel blocco di cache

1. e salto alla μ -istruzione 1

(= 100)

lettura, cache fault

$IND_p[31..2]00 \rightarrow IND_m,$

$OP_p \rightarrow OP_m, \text{ Set } RDY_m, 2.$

comanda la lettura del blocco in M

(≥ 101)

scritture, cache fault;
questo è il caso complicato
xché dobbiamo instaurare
un nuovo blocco.

Per via del write through non
occorre scrivere il vecchio
blocco.

leggiamo il blocco x le
altre parole ...

$IND_p [31..2]00 \rightarrow IND_m,$

"read" $\rightarrow OP_m, \text{ set } RDY_m, 3.$

Quando la lettura \uparrow

1. ($ACK_m = 0$) map, 1
attendit scrittura, questa può
essere verso una cache per
ragioni di efficienza, anziché
verso la memoria vera e propria

(=1) scrittura write through
effettuata, manda risposta
al processore

$ESTO_m \rightarrow ESTO_p$,

reset ACK_m , reset RDY_p ,

set ACK_p , \emptyset .

e poi ritorna allo stato
iniziale

2. ($ACK_m, ESITA_m = 0 -$) $map, 2$
(= 1 0) blocco nell'interfaccia

$IN_m[0] \rightarrow MEM_1[IND_p[Block]]$

$IN_m[3] \rightarrow MEM_4[IND_p[Block]]$

source blocco in cache \uparrow

$MEM_0[IND_p[Block]][TAG] = IND_p[TAG]$

$MEM_0[IND_p[Block]][PRESENT] = 1$

Aggiorna info di controllo \uparrow
 $ESITA_m \rightarrow ESITA_p$

reset ACK_m , reset RDY_p ,

set ACK_p , ϕ

(= 1 1) errore di lettura

$ESITA_m \rightarrow ESITA_p$, reset ACK_m

reset RDY_m , set ACK_m , ϕ

3. ($ACK_m, INP_p [OFF] = 0 \dots$) $map, 3$
 ($= 100$)

$IN_p \rightarrow OUT_m [00] \rightarrow MEM_1 [IND[Block]]$

$IN_m [1] \rightarrow OUT_m [1] \rightarrow MEM_2 [IND[Block]]$

$IN_m [2] \rightarrow OUT_m [2] \rightarrow MEM_3 [IND[Block]]$

$IN_m [3] \rightarrow OUT_m [3] \rightarrow MEM_4 [IND[Block]]$

"write" $\rightarrow OP_m$, reset ACK_m ,
 set RDY_m , 4.

trattamento esito

($= 101$) ...

($= 110$) ...

($= 111$) ...

stesse case \uparrow

con IN_p nella posizione
 giusta e il resto de
 IN_m (4 parole)

manca

qui ordino il write back e
 scrivo le cache

4. $(ACK_m, ESTO_m = 0 -)$ $ndp, 4$
(= 10)

$MEM_o[IND_p[BLCCO]] [PRESET] = 1$

$MEM_o[IND_p[BLCCO]] [TAG] = IND_p[TAG]$

$0 \rightarrow ESTO_p$, reset ACK_m ,

reset RDY_p , set ACK_p , ϕ

(= 11)

$ESTO_m \rightarrow ESTO_p$, reset ACK_m ,

reset RDY_p , set ACK_p , ϕ