

## Cache associativa

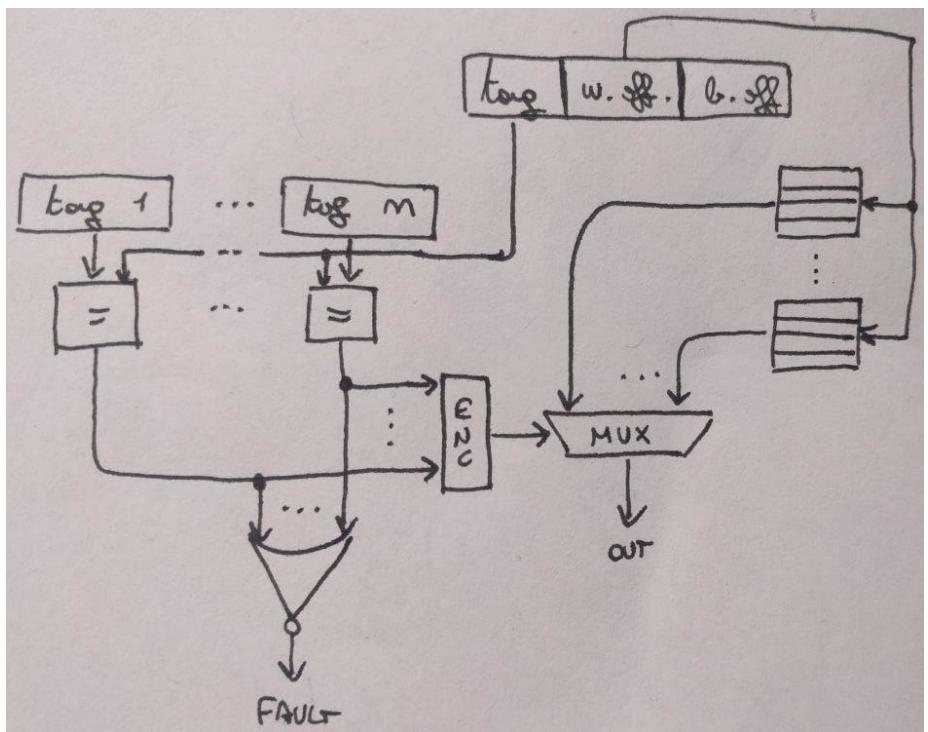
Supponiamo di avere una cache da 4 blocchi di 4 parole da 32 bit. Dato l'indirizzo  $i$ , distinguiamo:

- $i[1 : 0]$ : byte offset (sempre 00 se gli accessi sono allineati alla parola);
- $i[3 : 2]$ : word offset;
- $i[31 : 4]$ : tag.

Ciascuna riga della cache contiene le 4 parole con tag  $t$ , e  $t$  è memorizzato in un registro associato alla riga. Quando il processore accede a  $i$ :

- si determina se il tag è contenuto nella cache tramite dei comparatori (tanti quante sono le righe della cache);
- se è presente, si legge/scrive il valore in quella posizione;
- altrimenti si cerca nel livello successivo, e si inserisce nella cache: in una posizione libera se ce n'è una disponibile, sostituendo un altro valore altrimenti.

Ha il vantaggio di consentire la scelta di quale posizione liberare se la cache è piena e di occupare tutte le posizioni prima di iniziare a sostituire, ma è molto costosa per il numero di comparatori e perché i tag devono essere letti tutti contemporaneamente, quindi sono memorizzati in registri (non una memoria a porta singola).



cache associativa, solo lettura, senza bit di validità