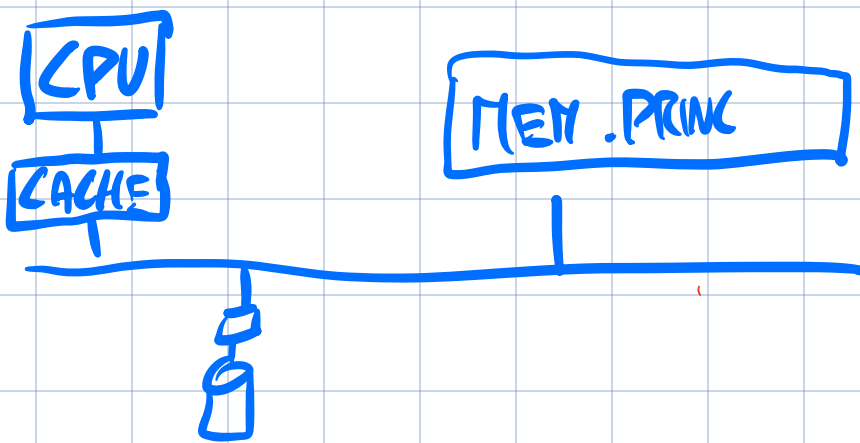
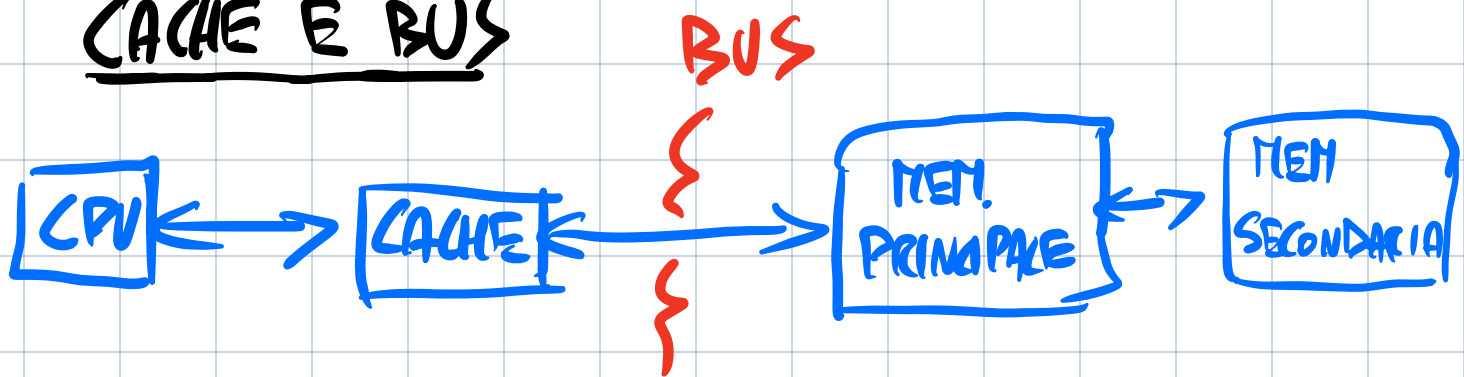


ASPETTI IMPLEMENTATIVI DELLE C.

CACHE E BUS



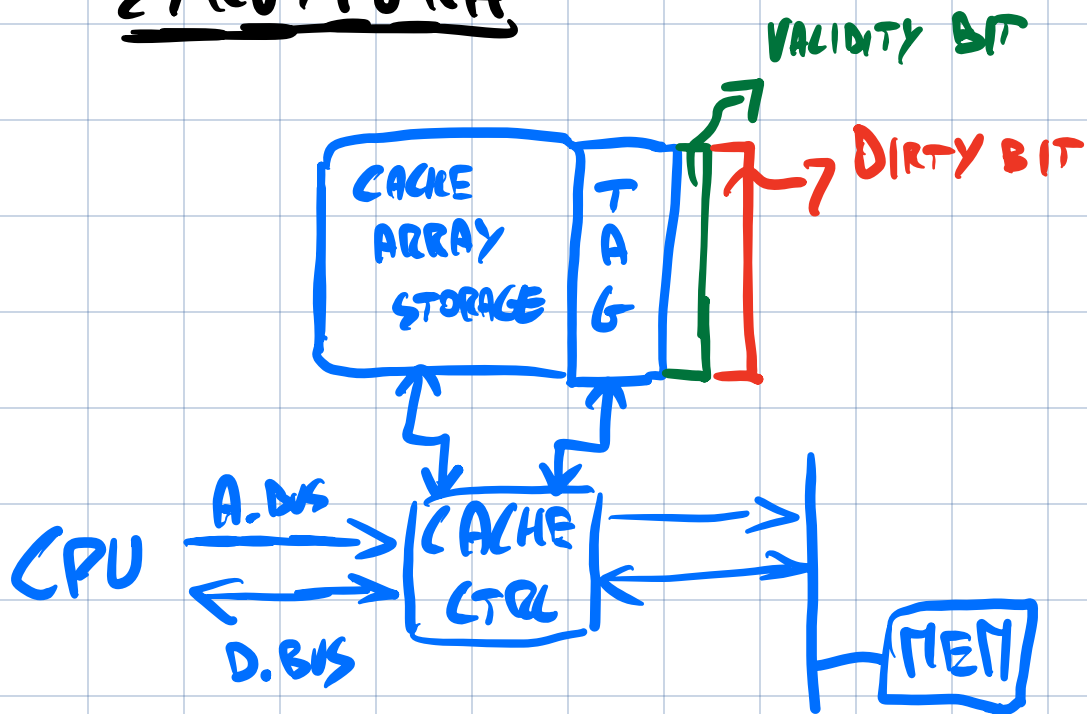
C. VICINO A CPU \Rightarrow DECONGESTIONA IL BUS

C. SFRUTTA LOCALITÀ DEI RIFERIMENTI

- L. TEMPORALE
- L. SPAZIALE

BLOCCHI DI C.: AIUTA PER L. SPAZIALE

STRUTTURA



ACCESSO

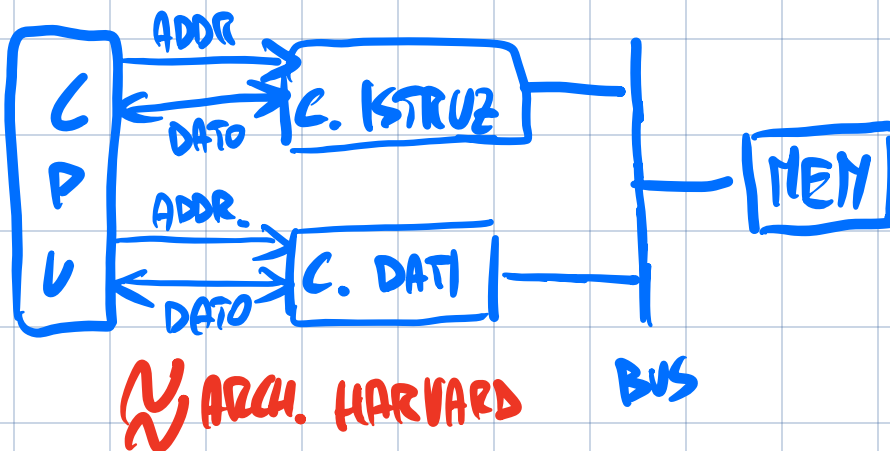
- IN CASO DI HIT: CPU LAVORA SU C.
- IN CASO DI MISS: DUE STRATEGIE
 - A. C CTRL CARICA IL BLOCCO DALLA M.
Dopo, LA CPU COMPIE ACCESSO
 - B. C CTRL CARICA LA CELLA DALLA MEM.
Dopo, LA CPU COMPIE ACCESSO
Dopo, C. CTRL FINISCE DI CARICARE BLOCCO
⇒ + VELOCE ⇒ + COSTOSA
DETTO "LOAD THROUGH"

QUANTE CACHE

1. UNA SOLA C. AGNOSTICA \Rightarrow FLESSIBILE

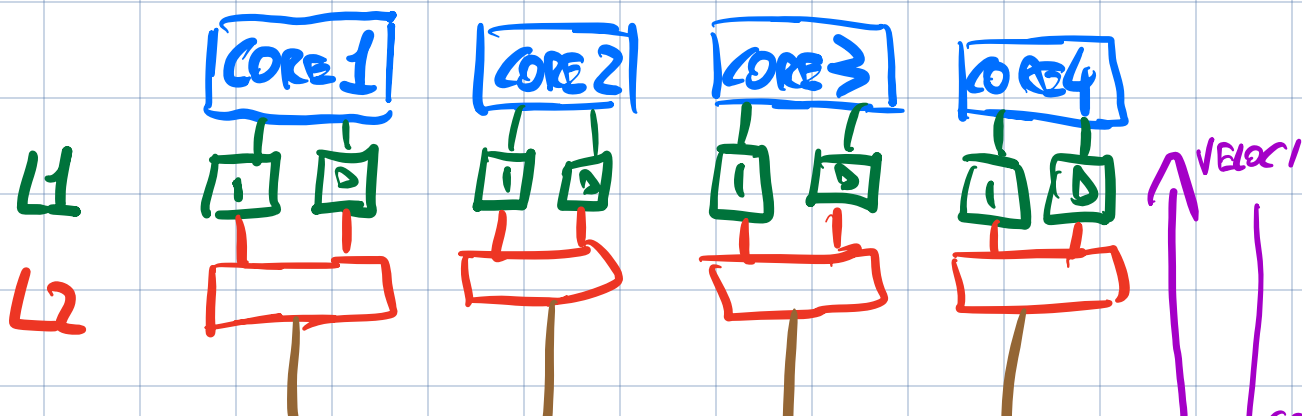


2. C. PER ISTRUZ. + C. PER DATI

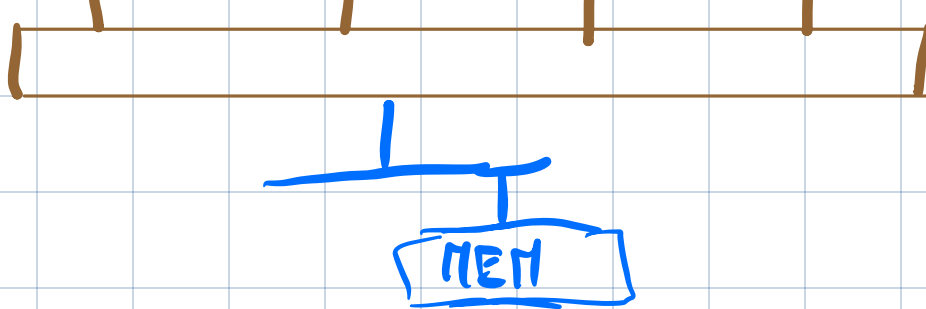


• CACHE MULTILIVELLO

- USATE IN CASCATA
- FINO A 3 LIVELLI DI C.
- PRESTAZ. NOTO MAGGIORI



L3



GRANDI
↓

PARAMETRI DI C.

- **DIMENSIONE:** INFLUENZA PREST. E COSTO

DA QUALCHE KB A QUALCHE MB

NON TROPPO GRANDE \swarrow COSTO DI S. RAM
 \searrow COSTO DI C. CTRL

- **MAPPING:** COME SI MAPPA I BLOCCHI DI M. E LINEE DI C.

- **DIRECT MAPPING:** $\# \text{LINEA} = \# \text{BLOCCO} \% \# \text{LINEE}$

SEMPLICE, SENSIBILE AD ACCESSI

INTERMITTENTI A BL. MAPPATI SU STESSA LINEA

- **ASSOCIATIVE MAPPING:**

MASSIMA FLESSIBILITÀ

MOLO COSTOSO \Rightarrow NON SI USA

- **SET ASSOCIATIVE:**

- LINEE RAGGRUPPATE IN **INSIEMI** DI **K**

ELEMENTI $K = 2, 4, 8$

$\# \text{INSIEME} = \# \text{BLOCCO} \% \# \text{INSIEMI}$

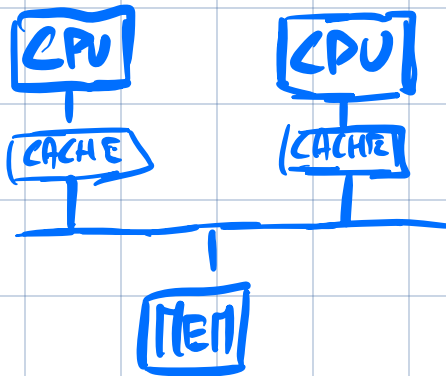
C. CTRL SCEGLIE

$$\hookrightarrow \# \text{INSIEMI} = \frac{\# \text{LINEE}}{K}$$

UNA DELLE K LINEE DI UN INSIEME

- AGGIORNAMENTO MEM:
 - WRITE BACK
 - WRITE THROUGH
- DIMENSIONE DELLE LINEE: 4, 16, 32 B

COERENZA DELLA C.



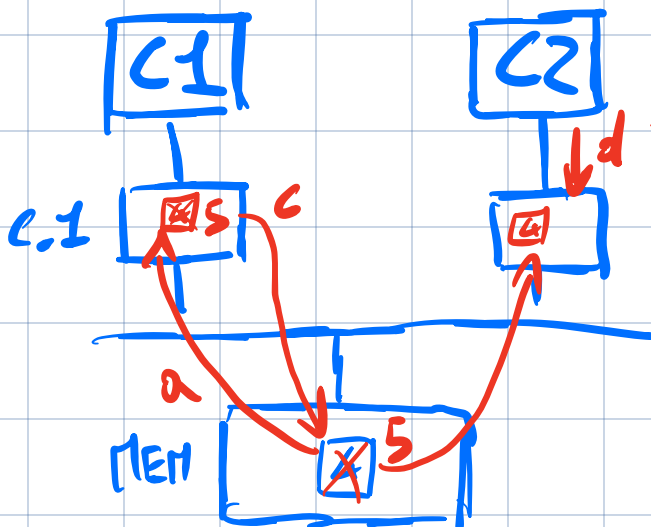
- LE CPU SPESSO SI SPARTISCONO MEMORIA

- A VOLTE MEM CONDIVISA

- SE UN BLOCCO SI TROVA IN PIÙ DI UN C.?

=> INCOERENZA

- WRITE THROUGH NON RISOLVE



=> C2 LEGGE 4! NON 5!
=> INCOERENZA

PERCHÉ CPU1 AGGIORNA C.1 E MEM., MA NON C.2

⇒ SI USANO "PROTOCOLLI DI COERENZA DI C."

1. ZONE DI "NON - CACHEABLE MEMORY"

2. BUS WATCHING O BUS SNOOPING

BROADCAST
↑
~
↓

TUTTE LE CPU OSSERVANO SEMPRE IL BUS

QUANDO UNA CPU SCRIVE IN C. MANDA UN MSG SU BUS

Q. WRITE INVALIDATE: MSG INVALIDE EVENTUALI

COPIE DI BLOCCO IN ALTRE C.

Q. WRITE UPDATE: MSG CHE AGGIORNA EVENTUALI

COPIE DEL BLOCCO IN ALTRE C.

⇒ PIÙ COMPLESSO

3. PROTOCOLLI COMPLESSI: MESI, MOESI

• SCAMBIO DI MSG TRA CPU