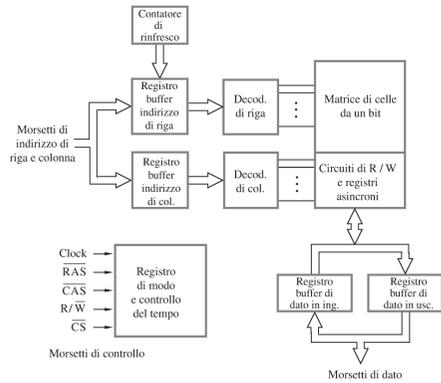


DRAM sincrone

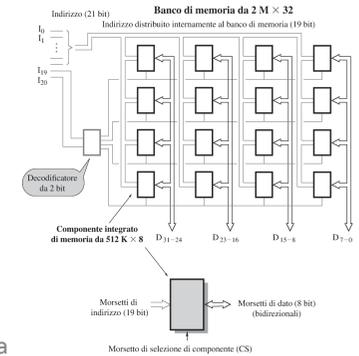
- Le memorie RAM dinamiche **sincrone** (SDRAM) usano un segnale di clock e aggiungono al loro interno un circuito di controllo che serve al rinfresco. Riescono a ricevere comandi in pipeline, quindi possono ricevere un nuovo comando prima che il precedente sia stato completato



10

Moduli di memoria

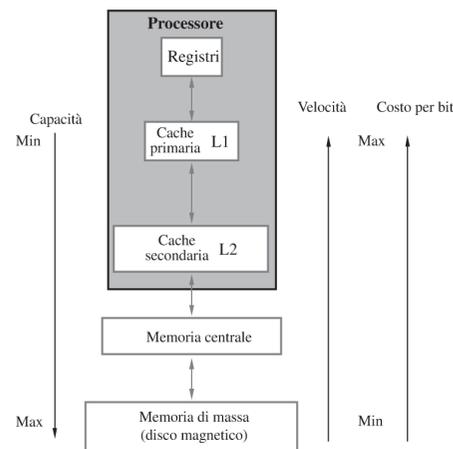
- Organizzazione di un modulo da 2M x 32 bit con chip da 512K x 8 bit
- Matrice 4 x 4 chip
- Per indirizzare 2 M parole (1 parola = 32 bit) occorrono 21 bit
- I 2 bit più significativi selezionano la riga attraverso il decodificatore da 2 bit che genera il segnale CS
- I 19 bit restanti si usano per indirizzare la parola all'interno di ciascuno dei 4 chip della riga (per indirizzare 512K occorrono 19 bit)



Prof. Tramontana

11

Gerarchia di memoria (S. 8.5)



Prof. Tramontana

12

Memoria cache e località (S. 8.6)

- La memoria cache ha lo scopo di mostrare al processore un sistema di memoria più veloce di quanto non sia la sola memoria centrale. L'efficacia dipende dalla proprietà di **località** che il programma esibisce
- L'analisi statistica dimostra che per la maggior parte del tempo si eseguono poche istruzioni ripetutamente, es. cicli con corpo di lunghezza breve, mentre il resto del programma va raramente in esecuzione
- Località temporale**: se si preleva un'istruzione nel ciclo di clock i , allora con elevata probabilità si preleva la stessa istruzione nel ciclo di clock $i+p$, con $p > 0$ piccolo
 - E' una conseguenza dalla presenza di cicli
- Località spaziale**: se si usa il dato all'indirizzo i , allora con probabilità elevata si usa pure il dato all'indirizzo $i+q$, con $q \neq 0$ piccolo (non si sa quando si usa il dato in $i+q$)
 - E' conseguenza della presenza di dati aggregati in strutture (es. vettori)

Prof. Tramontana

13

Località

- ▶ I **due principi** di località temporale e spaziale, sebbene **indipendenti**, sono spesso verificati simultaneamente
 - ▶ Es. durante la scansione di un vettore
 - ▶ Le stesse istruzioni (località temporale) eseguono su locazioni adiacenti in memoria (località spaziale)
 - ▶ Quindi, **entrambe le proprietà di località sono verificate**
- ▶ Riformulando i due principi insieme
 - ▶ **Con probabilità elevata entro breve tempo e per più volte la stessa parola di memoria o parole a breve distanza da essa saranno usate**
 - ▶ Ovvero, piccoli gruppi di parole di memoria sono usate più volte entro breve tempo

Prof. Tramontana

14

Cache hit

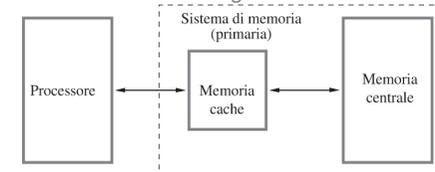
- ▶ Il processore manda al sistema di memoria comandi di lettura o scrittura generando indirizzi validi per lo spazio di indirizzamento e il controllore di cache esamina se la parola indirizzata si trova in cache
- ▶ Se l'operazione di lettura o scrittura si può eseguire su cache, si dice che si è verificato un **hit** (successo) di lettura o scrittura in cache, quindi *cache read hit* o *cache write hit*
- ▶ Se l'operazione è di **lettura**, e la cache contiene il dato, la memoria centrale non è coinvolta
- ▶ Se l'operazione è di **scrittura**, e la cache contiene il dato, si può procedere con una *scrittura immediata (write through)* in memoria centrale; o con una *scrittura differita (write back o copy back)* che aggiorna subito solo un *bit di modifica*, mentre la memoria centrale sarà aggiornata quando si dovrà liberare la posizione (blocco) della cache
- ▶ La scrittura immediata è più semplice, ma obbliga a scrivere la memoria anche quando non necessario, ovvero quando la cache viene aggiornata più volte. I processori usano spesso il modo **write back**

Prof. Tramontana

16

Uso della cache

- ▶ L'unità di controllo della cache lavora in base al principio di località spaziale e temporale: quando il processore usa una parola di memoria il **blocco** che la contiene è scritto in copia nella cache, dove il processore lo troverà se lo dovesse usare nuovamente
- ▶ Nella cache il **blocco** occupa una posizione (o regione) detta **posizione di cache**, o linea di cache, o blocco di cache. Quando il processore usa una qualsiasi delle parole del blocco la trova già in cache
- ▶ Lo spazio in cache è limitato. Quando le posizioni di cache sono tutte occupate e il processore richiede una parola non in cache bisogna trovare una posizione da liberare. La procedura di scelta è detta **algoritmo di sostituzione** (LRU, FIFO)



15

Cache miss

- ▶ Se l'operazione è di lettura e la parola indirizzata non si trova in cache si verifica un evento di **miss di lettura di cache**
- ▶ Il processore deve attendere che il blocco che contiene la parola venga caricato dalla memoria centrale e inserito in una posizione della cache. Dopo che il blocco è stato caricato in cache la parola viene letta dal processore, questo è detto modo di *lettura differita (read back)*
- ▶ In alternativa, la parola potrebbe essere mandata al processore non appena caricata in cache, senza attendere il completamento della lettura del blocco, questo modo è detto *lettura immediata (load through)*
- ▶ Se l'operazione è di scrittura e la parola non si trova in cache si ha una **miss di scrittura di cache**, ovvero cache write miss (o semplicemente miss)
- ▶ Se si adotta la *scrittura immediata (write through)*, la parola viene scritta subito in memoria centrale, oppure con la *scrittura differita (write back)* il blocco viene caricato in cache e quindi subito dopo viene aggiornato

Prof. Tramontana

17