

VERIFICA DI INFORMATICA (SOLUZIONE)

Liceo Scientifico "A. Volta"
classe 5° B, 25/01/2008
prof. Magni Claudio

1)

Prima trovo +29:

29	1
14	0
7	1
3	1
1	1
0	

$$29_{10} = 11101_2 \rightarrow +29_{10} = 011101_2$$

Ora complemento i bit e sommo 1:

$$100010 + 1 = 100011$$

$$-29_{10} = 100011_2$$

2)

Isolo i bit a gruppi di 4 e converto in una cifra esadecimale: $\underbrace{1011}_{11} \underbrace{0100}_4 \rightarrow = B4_{16}$

3)

Numero naturale: tutti i bit sono utilizzati per il modulo, il max num è: $2^n - 1$

Complemento a due: un bit è utilizzato per il segno, il max num è: $2^{(n-1)} - 1$

Per entrambe le codifiche i numeri esprimibili sono 2^n . Infatti quello è il numero di combinazioni possibili con n bit; ogni codifica utilizza tutte le sequenze disponibili (nessuna "sparisce"). Va notato che nel modulo e segno ci sono due zeri (+ e -), perciò i numeri distinti rappresentabili sono $2^n - 1$.

4)

Byte = 8 bit

Byte si abbrevia in 'B', mentre bit in 'b'.

Kilo e Mega sono prefissi che moltiplicano l'unità di misura a cui si riferiscono per, rispettivamente, 2^{10} e 2^{20} .

$$1 \text{ KB} = 1 \text{ Kilobyte} = 2^{10} \text{ B} = 1024 \text{ B} = 8 * 1024 \text{ bit}$$

5)

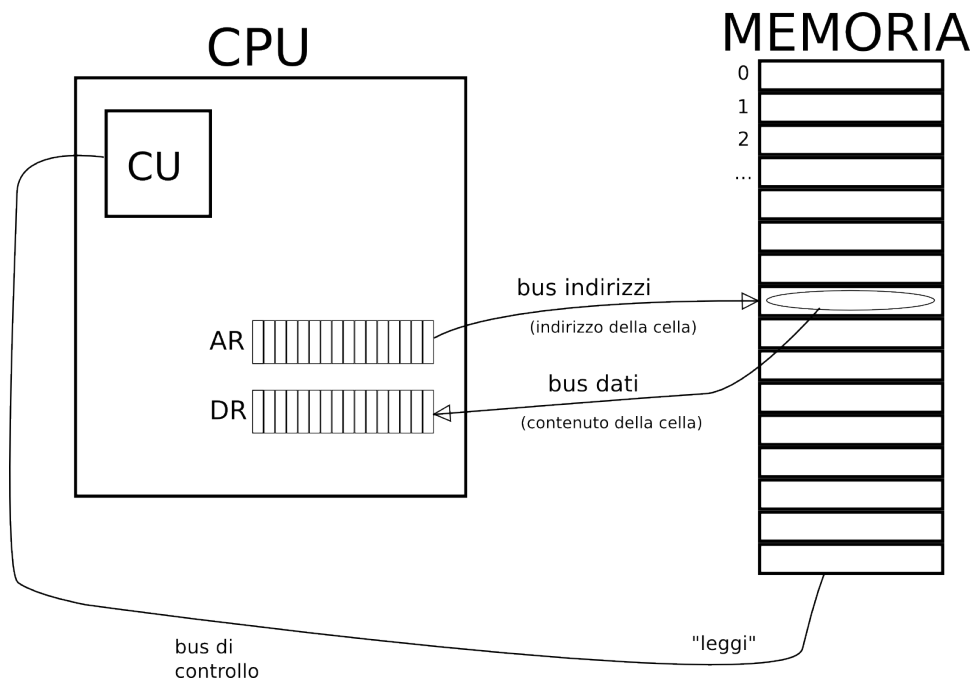
Il disegno mostra un modello molto semplificato di CPU e memoria.

La CPU contiene un dispositivo che manda e riceve segnali di controllo sul bus di controllo (CU = Control Unit). Inoltre contiene due importanti registri: AR (Address Register), DR (Data Register). Il primo contiene una sequenza di bit che corrisponde all'indirizzo di una cella di memoria, mentre il secondo contiene la parola di memoria indirizzata dall'AR.

La lunghezza in bit del registro AR determina il numero max di parole (= celle) che la memoria può avere. Infatti, se n sono i bit del registro, l'indirizzo più grande che riusciamo ad esprimere è $2^n - 1$: celle con un indirizzo maggiore non possono quindi essere utilizzate.

La lunghezza in bit del registro DR deve essere ovviamente la stessa di una parola di memoria, altrimenti non riuscirebbe a contenerla.

NOTA: il bus è condiviso tra tutti i dispositivi. Nell'immagine ho mostrato un collegamento diretto tra CPU e mem, in quanto esiste un collegamento logico.



Operazione di lettura:

1. *cpu*: carica l'indirizzo della parola nel registro indirizzi
2. *cpu*: lo trasmette alla memoria tramite bus indirizzi
3. *cpu*: richiede operazione di lettura tramite bus controllo
4. *mem*: trasmette la parola indirizzata sul bus dati
5. *mem*: segnala lettura conclusa tramite bus controllo
6. *cpu*: copia la parola nel registro dati tramite bus dati

Operazione di scrittura:

1. *cpu*: carica l'indirizzo della parola nel registro indirizzi
2. *cpu*: lo trasmette alla memoria tramite bus indirizzi
3. *cpu*: carica nel registro dati il dato da scrivere e lo trasmette tramite bus dati
4. *cpu*: richiede operazione di scrittura tramite bus controllo
5. *mem*: scrive, copia i dati ricevuti dal bus dati nella cella
6. *mem*: segnala scrittura conclusa tramite bus controllo

6)

Risposta riassuntiva.

- **Elettroniche:** veloci, accesso casuale (il tempo con cui prendo un dato è costante, non dipende dalla posizione), poco capienti, dato salvato sotto forma di tensione elettrica, in genere volatili (spegnendo il pc si perdono tutti i dati).
- **Magnetiche:** accesso diretto (il tempo con cui prendo un dato dipende dalla posizione), capienti, dato salvato sotto forma di campo magnetico, non volatili, soggette a guasti.
- **Ottiche:** accesso diretto, meno capienti e veloci rispetto a quelle magnetiche, ma affidabili (i dati non rischiano di essere persi), dato letto/scritto tramite incisioni fatte con laser.
- **Nastri:** molto lenti, molto capienti, accesso sequenziale (per prendere un dato devo scorrere tutti i dati intermedi), affidabili, utili per salvare dati sequenziali che non vengono più utilizzati.

7)

La gerarchia delle memorie è una organizzazione logica delle memorie del computer, inventata per poter utilizzare al meglio le caratteristiche di ogni memoria. Può essere rappresentata come una piramide in cui in basso vengono messe le memorie più lente e capienti, in alto le più veloci e poco capienti.

Viene sfruttato il principio di località, secondo cui quando si ha bisogno di un byte, in genere si ha bisogno anche dei byte "vicini" a quello. Pensate per esempio ad un file di Word, in cui volete correggere una frase: potreste leggere da disco solo la frase che volete modificare, ma il pc ugualmente carica tutto il file in memoria RAM, ritenendo che probabilmente sarà necessario leggere/modificare altre parti del file.

Quindi, nonostante la CPU lavori solo con parole, dal disco viene portato il blocco che contiene la parola nella RAM. Da lì viene portata nella Cache la pagina (insieme di parole < blocco) che contiene la parola. Infine la CPU lavora sulla parola ora presente nella Cache secondo il metodo visto nella domanda 5.