

Domande (in blu) e risposte

MSB LSB

Peso di quella cifra 8 4 2 1 1/2

Il numero binario 1 0 1 0 , 1 vale 10,5 in decimale perché... = $1 \times 8 + 1 \times 2 + 1 \times 1/2$

Peso di quella cifra $2^3 2^2 2^1 2^0 2^{-1}$ (ricordo che per ogni X si ha $X^0 = 1$ e $X^{-n} = 1 / X^n$)

Per convertire in binario la parte intera (10) va ... divisa per 2 prendendo i resti della divisione intera

$10 : 2 = \dots$
 $5 : 2 = \dots$ Resto 0
 $2 : 2 = \dots$ Resto 1
 $1 : 2 = \dots$ Resto 0
 $0 : 2 = \dots$ Resto 1

0,5 x 2 = 1 ← e la parte frazionaria (0,5) va ... moltiplicata per 2 prendendo la parte intera del prodotto

0,0 x 2 = si ferma, continua

↑ LSB
↓ MSB

Un numero **decimale** con N cifre permette di scrivere 10^N numeri diversi da 0...0 a 9...9
 Un numero **binario** con N cifre permette di scrivere 2^N numeri diversi da 0...0 a 1...1

1 Byte (abbreviazione **B**) = 8 bit (abbreviazione **b**)

1 Kb = 2^{10} bit = 1024 bit (1 KB = 2^{10} Byte = 1024 Byte)

Con quasi 1 Byte (con 7 bit) la codifica ASCII codifica un carattere della tastiera (ad esempio **A** è codificato con 0100 0001 (abbreviato con 4 1 esadecimale) e **B** con 0100 0010 (abbreviato con 4 2 esadecimale). Cosa significa allora "codificare" A? Significa associare un numero binario alla lettera A così se la tastiera vuole inviare al computer una **A** gli invia quel numero e se il computer riceve in ingresso quel numero sa che ha ricevuto in ingresso una A

Per quanti caratteri diversi bastano 7 bit? Per $2^7 = 128$ caratteri

La codifica **UNICODE** inizialmente è stata pensata per usare **2 Byte** per ogni carattere, ma poi si è visto che il numero era limitante e oggi la **UTF-8** (Unicode Transformation Format, 8 bit) ha **lunghezza variabile** e può essere lunga **da 1 Byte fino a 4 Byte**

Cosa fa la CPU? **Esegue i programmi**

Cosa è un programma? **Una serie di istruzioni**

Dettaglia meglio cosa fa la CPU per eseguire un programma composto da una serie di istruzioni sapendo che la prossima istruzione da eseguire è quella il cui indirizzo RAM è scritto nel registro Program Counter (un registro è una memoria in grado di contenere un solo numero, più o meno lungo ma un solo numero):

Fetch 1: il primo passo fatto dalla CPU dopo aver finito di eseguire una istruzione è di andare a leggere in RAM la prossima istruzione (quella il cui indirizzo è nel Program Counter), di portare l'istruzione dentro alla CPU, di memorizzarla nel Registro Istruzioni e di aggiornare il Program Counter aumentando il suo numero di 1 o di più a secondo di quanto occupa una istruzione;

Esecuzione 1: esegue l'istruzione scritta nel Registro Istruzioni

Fetch 2: va a leggere in RAM la prossima istruzione, quella il cui indirizzo è nel PC

Dalla CPU escono 3 gruppi di fili detti bus: il bus indirizzi, il bus dati e il bus di controllo

Quando si dice che il computer è a 64 bit, si dice che ha 64 fili quale bus? **Bus dati**

Cosa vuol dire che su un bus la comunicazione è **parallela**? Che ad esempio i 64 bit di un numero viaggiano tutti insieme e in un solo passaggio si trasferisce tutto il numero

Nella comunicazione **seriale** i fili sono 2 e permettono il passaggio di una cifra per volta, per cui per trasportare un numero da 64 bit servono 64 invii (passaggi)

La comunicazione seriale è necessariamente lenta, impiegando molti invii per trasferire un solo numero? No, perché se un **singolo passaggio è molto veloce** può essere ad alta velocità anche la comunicazione seriale

Come vengono usati i 3 bus per andare a leggere la variabile X memorizzata in RAM all'indirizzo 570 e portarla nel registro accumulatore A?

- 1) Si mette **570 sul bus indirizzi** (in un registro detto Address Register);
- 2) Se dice sul bus di controllo due cose: a) che l'indirizzo attualmente presente sul bus indirizzi è un **indirizzo di RAM** e non un indirizzo di periferica; b) che a quell'indirizzo la CPU **vuole leggere** e non scrivere;
- 3) Ora la RAM sa tutto quello che deve sapere e si mette a leggere all'indirizzo 570. **Dopo un certo tempo ha finito e il dato richiesto è pronto sull'uscita;**
- 4) Il dato ora presente sul **bus dati** viene memorizzato nell'**accumulatore A**