



# FORMULE UTILI PER ARCHITETTURA



# BUS TERMINOLOGIA

$T$  = periodo = 1 ciclo di clock

$f$  = frequenza = cicli di clock al secondo

Banda passante = bit passanti al secondo

Linee = linee del bus

Livelli segnali = livelli che indicano il valore binario del segnale mandato

(es. Livello 4 indica che viene mandata e ricevuta la sequenza 11)



## BUS FORMULE

$$f = \frac{1}{T} \quad T = \frac{1}{f}$$

$$\textit{Banda passante} = f * \textit{linee} * \log_2(\textit{livelli segnale})$$

$$\textit{BANDA PASSANTE 1 LINEA} = \frac{\textit{BANDA PASSANTE}}{\textit{LINEE}}$$



# BUS SKEW

skew= tempo massimo necessario per la consegna del bit ad ogni ciclo di clock

Bisogna fare molta attenzione al testo dell'esercizio per capire cosa vuole il prof, dato che ogni esercizio con lo skew è diverso e non esiste un metodo generale per risolverlo. Le conoscenze seguenti sono solo per determinati esercizi, quindi è utile capire bene lo skew per applicarlo all'esercizio

Per gli esercizi bisogna dividere in due tipologie

- sequenziale
- parallelo



## BUS SKEW SEQUENZIALE

Per questa tipologia di esercizi bisogna fare attenzione, dato che per ogni linea sequenziale bisogna aggiungere 1 skew(es. per 4 linee bisogna fare 4\*skew)

Quindi il ciclo di clock minimo è numero di linee sequenziali \* skew

$$SKEW DI 1 LINEA = \frac{T}{LINEE}$$

$$SKEW GLOBALE = T$$



## BUS SKEW PARALLELO

In questo caso basta contare un solo skew ad ogni ciclo di clock

Quindi se chiede quale è il minimo ciclo di clock possibile il ciclo di clock deve essere maggiore uguale allo skew



# CACHE TERMINOLOGIA

TAG=regione della memoria mappata sulla linea

LINE=linea della cache

WORD=numero di word nella linea

BYTE=grandezza in byte del word nella linea

DIM. CACHE=grandezza totale della cache

BIT In OGNI LINEA=quanti bit abbiamo in ogni linea

DIM. MEMORIA=grandezza memoria principale(es. ram)

per TAG, LINE, WORD, BYTE sono intesi come il numero di bit necessari per rappresentare il loro valore

(es. per 2048 linee servono 11 bit)

(per trasformare in bit basta la formula:  $\log_2(x)$ )

OFFSET=WORD+BYTE



## CACHE FORMULE

$$DIM. CACHE = 2^{LINE} * \frac{(1+TAG+8*2^{(WORD+BYTE)})}{8}$$

(se si vuole in byte si divide con 8 se si vuole in bit non serve)

$$DIM. MEMORIA = 2^{TAG} * DIM. CACHE$$

$$OFFSET = BITINDIRIZZO - TAG - LINE$$

(bit indirizzo è inteso come lunghezza totale dell'indirizzo quindi TAG+LINE+OFFSET solitamente 32)





# CACHE TIPOLOGIA DI ESERCIZIO

Dato un indirizzo  $N$  in memoria principale trovare la linea e il byte in cui si trova in cache

Metodo A)

$$LINEAeOFFSET = INDIRIZZO\ N \bmod DIM\ CACHE$$

$$LINEA = LINEAeOFFSET \div \#BYT\ E\ IN\ OGNI\ LINEA$$

$$BYTE = LINEAeOFFSET \bmod \#BYT\ E\ IN\ OGNI\ LINEA$$

Metodo B)

$$LINEAeOFFSET = INDIRIZZO\ N \bmod 2^{LINE+OFFSET}$$

$$LINEA = LINEAeOFFSET \div 2^{OFFSET}$$

$$BYTE = LINEAeOFFSET \bmod 2^{OFFSET}$$



# PAGE TABLE TERMINOLOGIA

MEM. PRINCIPALE= memoria principale (es. Ram)

MEM. VIRTUALE= memoria fisica in memoria di massa

DIM. PAGINA=OFFSET= quante locazioni vengono contenute in una pagina

DIM. INDIRIZZO MEM. PRINCIPALE= numero di bit necessari per mappare la mem. principale

#INDIRIZZI VIRTUALI=numero di indirizzi virtuali esistenti

#RIGHE PAGE TABLE=quante righe servono per mappare la mem virtuale



## PAGE TABLE FORMULE

$$DIM. INDIRIZZO MEM. PRINCIPALE = \log_2 (DIM. MEM. PRINCIPALE)$$

$$\#RIGHE PAGETABLE = \frac{DIM. MEM. VIRTUALE}{DIM. PAGINA}$$

$$RIGA PAGETABLE = 1 + DIM. INDIRIZZO MEM. PRINCIPALE$$

$$DIM. PAGETABLE = \#RIGHE PAGETABLE * RIGA PAGETABLE$$



# PAGE TABLE TIPOLOGIA DI ESERCIZIO

Se richiede il riempimento migliore:

$$RIEMPIMENTO\ MIGLIORE = \left\lfloor \frac{DIM.\ MEM.\ PRINCIPALE}{DIM.\ PAGINE} \right\rfloor$$

Se richiede il riempimento peggiore:

$$RIEMPIMENTO\ PEGGIORE = \left\lfloor \frac{DIM.\ MEM.\ PRINCIPALE}{(2 * DIM.\ PAGINE) - 1} \right\rfloor$$



# ARDUINO

Con l'esercizio riguardante il prescaling dato un determinato counter a N bit e la frequenza:

$$OVERFLOW = 2^{(BIT\ TIMER)}$$

$$OVERFLOW\ IN = \frac{PRESCALING * OVERFLOW}{f}$$

Durante l'esercizio prestare attenzione alle unità di misura, che da quello dipende il risultato( ms, ns ecc.)