

Nome, cognome, matricola

Calcolatori Elettronici (12AGA) – esame del 18.1.2023

Domande a risposta chiusa (è necessario rispondere correttamente ad almeno 6 domande).
Non è possibile consultare alcun tipo di materiale. Tempo: 15 minuti.

1	Si consideri un circuito sequenziale sincrono con 50 ingressi, 70 uscite e 150 stati. Qual è il numero minimo di flip flop necessari per la sua implementazione?		
2	Dove è memorizzata la Interrupt Vector Table in un sistema general purpose?	Nei registri dell'Interrupt Controller	A
		All'interno della memoria ROM	B
		All'interno della memoria RAM	C
		In memoria secondaria	D
3	Si consideri una cache con le seguenti caratteristiche • 256 linee da 32 byte • Meccanismo set associative a 4 vie con sostituzione LRU. Assumendo che gli indirizzi emessi dal processore siano su 32 bit, qual è la dimensione del campo tag associato a ogni linea?	8 bit	A
		21 bit	B
		24 bit	C
		26 bit	D
4	Quale dei seguenti dispositivi può diventare un master di un bus?	Memoria	A
		Interfaccia di periferico	B
		DMA controller	C
		Interrupt Controller	D
5	Si consideri un sistema che utilizza il meccanismo della memoria virtuale: dove si trova la Memory Address Table (MAT)?	Nella MMU	A
		Nella TLB	B
		In Memoria Secondaria	C
		In Memoria Principale	D
6	Quale vantaggio introduce l'utilizzo di una Unità di Controllo microprogrammata rispetto ad una Cablata?	Maggiore velocità	A
		Minor costo dell'hardware	B
		Minor consumo	C
		Maggiore facilità di progettazione	D
7	A cosa serve il rinfresco delle memorie DRAM?	A ridurre gli effetti dei guasti indotti da radiazioni	A
		A ridurre il tempo di ciclo della memoria	B
		A risolvere il problema creato dal Destructive Read-Out	C
		A permettere alla memoria di mantenere nel tempo le informazioni	D
8	Quale tra i fenomeni elencati a lato può causare uno stallo in un processore con pipeline?	L'esecuzione di una istruzione in virgola mobile	A
		Un errore in un'operazione aritmetica	B
		Un'operazione di I/O	C
		L'esecuzione di un'istruzione NOP	D

9	Considerando il linguaggio MIPS a quale tipo di formato corrisponde l'istruzione sw ?	S-type	A	
		R-type	B	
		I-type	C	
		J-type	D	

10	Dato il seguente frammento di codice in linguaggio MIPS si indichi il valore finale del registro \$t3 main: addi \$t1, \$0, 5 addi \$t2, \$0, 6 slt \$t4, \$t2, \$t1 beq \$t4, \$0, L1 addi \$t3, \$0, 5 j L2 L1: addi \$t3, \$0, 8 L2:	\$t3 = 8	A	
		\$t3 = 5	B	
		\$t3 = 0	C	
		\$t3 = 6	D	

Risposte corrette

1	2	3	4	5	6	7	8	9	10
8	C	B	C	D	D	D	A	C	A

Domande a risposta aperta (sino a 5 punti per ogni domanda) – Non è possibile consultare alcun materiale -
Tempo: 40 minuti.

11

Si consideri un sistema a processore che include

- una memoria di 512 byte
- una cache direct-mapped composta da 4 linee da 4 byte ciascuna.

Si assuma che inizialmente la cache contenga i blocchi 0, 1, 2, 3. Si determini il numero del blocco presente in ciascuna linea della cache al termine della sequenza di accessi in memoria corrispondente agli indirizzi riportati in tabella e si riporti il risultato nella figura.

Sequenza degli accessi alla memoria

Accesso 1	000010101
Accesso 2	100010111
Accesso 3	000011001
Accesso 4	000000000
Accesso 5	001110100
Accesso 6	001010111
Accesso 7	100011001
Accesso 8	010111100

Contenuto iniziale della cache

Linea	Blocco
0	0
1	1
2	2
3	3

Contenuto finale della cache

Linea	Blocco
0	
1	
2	
3	

12	<p>Si consideri il meccanismo della memoria virtuale: si descriva la sequenza di operazioni eseguite per trasformare ciascun indirizzo logico generato dal processore nel conseguente indirizzo fisico. Per ciascuna operazione, si specifichi quali sono i moduli hardware o i componenti software coinvolti.</p>
----	--

13	<p data-bbox="132 73 1544 163">Si elenchino le operazioni che si susseguono in un sistema a processore che utilizza il meccanismo dell'interrupt vettorizzato dal momento in cui una periferica attiva una richiesta di interrupt al momento in cui la CPU esegue la prima istruzione della corrispondente procedura di servizio dell'interrupt (ISR).</p>
----	--

14	<p>Progettare un circuito di controllo per un impianto di illuminazione stradale costituito dai seguenti ingressi:</p> <ol style="list-style-type: none"> 1. Interruttore di accensione/spengimento impianto (I) 2. Segnalatore di luminosità (L) 3. Interruttore di accensione permanente impianto (P) <p>Il sistema è costituito da una uscita U che comanda l'accensione dei corpi illuminanti.</p> <p>Quando l'interruttore I è spento l'impianto viene disabilitato ($U=0$). Altrimenti l'uscita è alta quando l'interruttore P è attivo oppure se l'indicatore L di luminosità indica scarsa luminosità (ciò avviene quando L è basso).</p> <p>Si richiede di:</p> <ul style="list-style-type: none"> - riempire la tavola di verità delle uscite - riempire la mappa di Karnaugh - specificare la funzione minima derivante dalla copertura della mappa.
----	--

Nome, Cognome, Matricola:.....

Esercizio di programmazione

sino a 12 punti – è possibile consultare solamente il foglio consegnato con l’instruction set MIPS - tempo: 60 minuti

Si scriva, in linguaggio Assembly MIPS, una procedura denominata *EvenParity* che esegua il calcolo della parità per i numeri contenuti nel *nibble* meno significativo dei byte di un vettore. La parità è *Pari* se il numero di “1” contenuti nel nibble è pari; è *Dispari*, altrimenti.

Il bit più significativo del byte verrà utilizzato per memorizzare il risultato del calcolo: 0 se la parità è Pari, 1 se è Dispari.

In uscita la procedura fornisce il numero totale di elementi del vettore che hanno una parità Pari.

I parametri passati alla procedura attraverso i registri sono (nell’ordine indicato):

- indirizzo del vettore
- numero di elementi del vettore.

La procedura ritorna il seguente parametro in uscita:

- numero di elementi del vettore che hanno una parità Pari.

Esempio

vettore in ingresso	vettore in uscita
0B	8B
05	05
01	81
04	84
06	06
02	82
0C	0C

Valore in uscita: **3**

Si lavori nell’ipotesi di non avere *overflow* durante i calcoli.

Di seguito un esempio di programma chiamante:

```
DIM=7
.data
vettore: .byte 11 5 1 4 6 2 12

.text
.globl main
.ent main

main: subu $sp, $sp, 4
      sw $ra, ($sp)
      ...
      la $a0, vettore
      li $a1, DIM
      jal EvenParity
      ...
      lw $ra, ($sp)
      addiu $sp, $sp, 4
      jr $ra
.end main
```