

18/02/2021

lunedì 5 giugno 2023 17:09

Nome, cognome, matricola

Calcolatori Elettronici (12AGA) –esame del 1.9.2020**Domande a risposta chiusa** (è necessario rispondere correttamente ad almeno 6 domande).

Tempo: 15 minuti.

1	Si consideri una memoria composta da 32Kparole da 20 bit ciascuna. Quanti bit di indirizzo compaiono tra i suoi segnali di ingresso?	$2^5 \cdot 2^{10} = 2^{15} \Rightarrow 15$	
2	Quanti transistor sono necessari per implementare una cella di SRAM?	1	A
		4	B
		6	<input checked="" type="checkbox"/> C
		Dipende dalla dimensione della memoria	D
3	Quale delle seguenti affermazioni è vera?	Quando un processore riceve una richiesta di interrupt, la corrispondente procedura di servizio dell'interrupt viene attivata non appena termina l'istruzione in corso	<input checked="" type="checkbox"/> A
		Quando un processore riceve una richiesta di interrupt, la corrispondente procedura di servizio dell'interrupt viene attivata non appena termina il programma in corso	B
		Quando un processore riceve una richiesta di interrupt, la corrispondente procedura di servizio dell'interrupt viene attivata al termine del periodo di clock in corso	C
		Quando un processore riceve una richiesta di interrupt, la corrispondente procedura di servizio dell'interrupt viene attivata non appena il programma in corso accetta di venire sospeso	D
4	Quale dei moduli elencati NON è presente all'interno di un microcontrollore?	CPU	A
		FPGA	<input checked="" type="checkbox"/> B
		Memoria RAM	C
		Interfaccia di periferico	D
5	Si consideri un sistema per l'arbitraggio del bus tra 20 unità master che utilizza il meccanismo del polling. Quanti segnali sono necessari per l'arbitraggio?	$2 + \log_2 20 = 7$	
6	Si consideri la memoria di microcodice esistente in un'unità di controllo microprogrammata, e si assuma che la memoria sia composta da 200 parole da 80 bit ciascuna. Quanti bit sono necessari per il μ PC?	7	A
		8	<input checked="" type="checkbox"/> B
		80	C
		200	D
6	Si consideri il meccanismo della memoria virtuale: dove è memorizzato il TLB?	Nella memoria principale	A
		Nella memoria secondaria	B
		Nella cache	C
		Nella MMU	<input checked="" type="checkbox"/> D

7	Quanto vale il parametro CPI in un processore CISC?	Sempre meno di 1	A	
		Circa 1	B	
		Sempre più di 1	C	
		Dipende dal processore	D	

9	Su quanti bit è rappresentato l'immediato all'interno del codice macchina di un'istruzione MIPS di tipo I?	8	A	
		16	B	
		32	C	
		Dipende dall'istruzione	D	

10	Quale valore (in decimale) sarà presente in \$s0 dopo l'esecuzione dell'istruzione andi \$s0, \$s1, 7 assumendo che \$s1 contenga il valore 255?	<pre> 111 11 11 1 0000111 = 7 </pre>
----	--	--------------------------------------

Risposte corrette

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

Nome, cognome, matricola

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

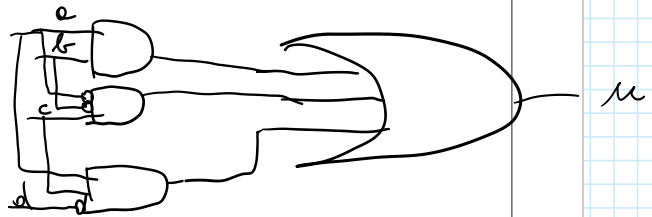
- 11 Si consideri un processore connesso ad una memoria da 64KB e dotato di una cache direct mapped da 16 linee, ciascuna da 32 byte. Assumendo che inizialmente le 16 linee contengano i primi 16 blocchi di memoria (quindi la linea 0 contiene il blocco 0, la linea 1 il blocco 1, e così via), si determini quali dei seguenti 12 accessi in memoria da parte del processore provocano un hit, e quali un miss, scrivendo H o M nella colonna di destra della corrispondente riga nella tabella.
- | Indirizzo | Blocco | Linea acceduta | H/M |
|---------------------|------------|----------------|-----|
| 0100 0000 0011 0011 | $2^8 + 4$ | 4 | M |
| 0100 0001 0001 1000 | $2^8 + 8$ | 8 | M |
| 0000 0100 1000 1110 | 36 | 4 | M |
| 0010 0000 1011 1110 | $2^8 + 5$ | 5 | M |
| 0100 0000 1001 1111 | $2^8 + 4$ | 4 | M |
| 0100 0000 0011 0011 | $2^8 + 4$ | 4 | H |
| 0000 1010 0001 0011 | 20 | 0 | M |
| 0000 1010 0101 0100 | $2^6 + 12$ | 2 | M |
| 0000 0011 0011 0100 | 25 | 3 | M |
| 0000 0011 0011 0110 | 25 | 3 | H |
| 0000 1000 1001 1000 | $2^6 + 4$ | 4 | M |
| 0000 0000 0001 1001 | 0 | 0 | M |
- 12 Si consideri un sistema a microprocessore che utilizza il meccanismo dell'interrupt vettorizzato.
- Si illustri il processo che va dal momento in cui un periferico attiva la richiesta di interrupt a quello in cui parte l'esecuzione della prima istruzione della corrispondente procedura di servizio dell'interrupt.
- ① DISP. invia richiesta di int. all' IC
 - ② L' IC invia l'int. alla CPU
 - ③ CPU completa l'ISTRUZ. corrente
 - ④ CPU invia l'int. ACK
 - ⑤ L' IC scrive l'identificativo del DISP. sul DBUS
 - ⑥ CPU salva il contenuto nello STACK → PC e REG. DI STATO
 - ⑦ CPU accede all' IVT leggendo l'indirizzo dell' ISR
 - ⑧ CPU esegue ISR
 - ⑨ Al termine, ripristina PC e REG. DI STATO

- 13 Si consideri la funzione Booleana di 4 variabili $f = ab + acd' + a'b'c$. Si richiede di
- Scrivere la tabella di verità per f , utilizzando la prima tabella riportata sotto
 - Disegnare la mappa di Karnaugh, utilizzando la seconda tabella riportata sotto
 - Identificare l'espressione booleana minima che implementa la funzione f
 - Descrivere il circuito minimo che implementa f , specificando in particolare da quante e quali porte è composto.

abcd	f
0000	0
0001	0
0010	1
0011	1
0100	0
0101	0
0110	1
0111	1
1000	0
1001	1
1010	0
1011	1
1100	0
1101	1
1110	1
1111	1

ab \ cd	00	01	11	10
00	0	0	1	1
01	0	0	0	0
11	1	1	1	1
10	0	0	0	1

$$f = ab + a'b'c + acd'$$



- 14 Si scrivano le microistruzioni (inclusive della fase di fetch) eseguite da un processore avente l'architettura in figura durante l'esecuzione dell'istruzione ADD R1, R2, 37.
Tale istruzione scrive in R1 il risultato della somma tra R2 e 37.

$MAR \leftarrow PC$

$Y = 0$

$CARRY = 4$

$Z \leftarrow Y + PC + CARRY$

RD

$PC \leftarrow Z$

NFC

$MDR \leftarrow BUS\ EST.$

$IR \leftarrow MDR$

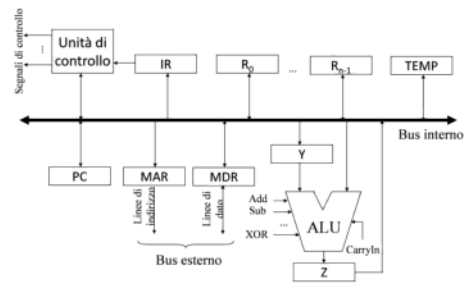
FETCH

$Y \leftarrow R2$

$Z \leftarrow Y + 37$

$R1 \leftarrow Z$

ADD



Nome, Cognome, Matricola:.....

Esercizio di programmazione

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

Si scriva, in linguaggio Assembly MIPS32, una procedura `SpaceRemove`, che esegua una elaborazione degli elementi contenuti in un vettore di byte `vetRX[]` e restituisca un vettore di byte `vetTX[]` di dimensione minore o eguale a quella di `vetRX[]`.

Il vettore `vetRX[]` presenta la seguente struttura :

<i>posizione</i>	<i>vetRX[]</i>
0	Indirizzo Destinazione
1	Indirizzo Origine
2	1^ Byte Messaggio
3	2^ Byte Messaggio
4	3^ Byte Messaggio
...
N	M-esimo Byte Messaggio
N+1	<EOM>

La procedura dovrà leggere i contenuti di `vetRX[]` e costruire `vetTX[]` con le seguenti regole:

- scambiare gli indirizzi Origine e Destinazione
- memorizzare in `vetTX[]`, a partire dalla posizione `vetTX[2]`, i valori letti tra `vetRX[2]` e `vetRX[N]` (elemento che precede il carattere <EOM> (0x03)), avendo cura di scartare i caratteri blank (0x20) .
- concludere `vetTX[]` con il carattere <EOM> (0x03)

Esempio :

```
vetRX[] 0x84, 0xFA, 0x09, 0x54, 0x20, 0x42, 0x19, 0x20, 0x41, 0xB1, 0x03
vetTX[] 0xFA, 0x84, 0x09, 0x54, 0x42, 0x19, 0x41, 0xB1, 0x03
```

I parametri sono passati alla procedura attraverso i registri:

- `$a0` contiene l'indirizzo di `vetRX[]`
- `$a1` contiene l'indirizzo di `vetTX[]`

Di seguito un esempio di programma chiamante:

```
DIM = 64
.data
vetRX: .byte 0x84, 0xFA, 0x09, 0x54, 0x20, 0x42, 0x19, 0x20, 0x41,
0xB1, 0x03
vetTX: .space DIM
.text
.globl main
.ent main
main: [...]
la $a0, vetRX
la $a1, vetTX
jal SpaceRemove
[...]
.end main
```