

Nome, cognome, matricola

Calcolatori Elettronici (12AGA) – esame del 2.7.2021

Domande a risposta chiusa (è necessario rispondere correttamente ad almeno 6 domande).

Tempo: 15 minuti.

1	Si consideri un flip flop di tipo SR. Quante porte logiche sono necessarie per la sua realizzazione?		
2	Quale tra le seguenti soluzioni permette di realizzare un sommatore per numeri interi senza segno con il minimo costo hardware?	Sommatore seriale	A
		Sommatore combinatorio sintetizzato	B
		Ripple carry Adder	C
		Carry Lookahead Adder	D
3	Quale vantaggio offre la microprogrammazione rispetto all'adozione di un'unità di controllo cablata?	Maggiore velocità	A
		Maggiore facilità di progettazione	B
		Minor costo dell'hardware necessario	C
		Maggiore affidabilità	D
4	L'adozione di un codice di Hamming per la correzione degli errori nelle memorie DRAM richiede l'aggiunta di alcuni bit alla memoria. A cosa sono associati i $\log_2 n$ bit necessari?	A ciascun byte	A
		A ciascuna parola	B
		A ciascun blocco	C
		All'intera memoria	D
5	Si consideri una cache direct mapped composta da 128 linee da 32 byte ciascuna. Assumendo che la memoria principale sia composta da 64 Mbyte, qual è la dimensione in bit del campo tag?		
6	Si consideri una memoria EPROM integrata all'interno di un sistema di elaborazione a microprocessore di tipo special purpose. Quale delle seguenti informazioni NON può essere memorizzata al suo interno?	Codice dell'applicazione	A
		Tabella di parametri	B
		Codice per il boot	C
		Variabili dell'applicazione	D
7	Chi pilota i segnali di handshaking presenti in un bus asincrono?	La CPU	A
		La memoria	B
		L'arbitro	C
		Le due unità che devono scambiarsi dati	D

8	Che cosa afferma la legge di Moore?	La frequenza dei processori raddoppia ogni 18 mesi	A	
		Le prestazioni dei processori raddoppiano ogni 18 mesi	B	
		Il numero di transistor integrabili in un singolo circuito integrato raddoppia ogni 18 mesi	C	
		Il consumo medio di un singolo circuito integrato si dimezza ogni 18 mesi	D	
9	Quale delle seguenti istruzioni esegue una divisione per 16 del registro \$s1, scrivendo il risultato in \$s0? Si assuma che \$s1 contenga un numero intero con segno.	srl \$s0, \$s1, 4	A	
		sra \$s0, \$s1, 4	B	
		sla \$s0, \$s1, 4	C	
		Non è possibile eseguire divisioni su numeri con segno usando istruzioni di scalamento	D	
10	Si considerino tre variabili a , b e c in memoria, ciascuna su 32 bit. Si scriva un frammento di codice MIPS che esegua la divisione di a per b , scrivendo il quoziente in c .			

Risposte corrette

1	2	3	4	5	6	7	8	9	10
2	A	B	B	14	D	D	C	B	

lw \$1, a
lw, \$2, b
div \$3, \$1, \$2
sw \$3, c

Nome, cognome, matricola

Parte 2

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

11	<p>Si descriva il meccanismo di gestione di un dispositivo periferico attraverso il meccanismo dell'interrupt vettorizzato. Considerando un'operazione di lettura, si elenchino i vari passaggi dal momento in cui il dispositivo periferico abbia un dato pronto in lettura, evidenziando tutte le operazioni svolte dalla CPU.</p>
12	<p>Progettare un circuito minimo a due livelli che implementi un'unità aritmetica su 1 bit, avente due ingressi di dato (I1 e I2) e un ingresso di operazione (Op), ciascuno su un bit.</p> <p>Se Op=0 gli ingressi vanno sommati, altrimenti il secondo va sottratto al primo. Il circuito deve fornire il risultato (R), il carry (C) e il borrow (B).</p> <p>Compilare la tavola di verità e le mappe di Karnaugh e fornire le funzioni delle uscite implementate dal circuito minimo.</p>

13	<p>Con riferimento ad un'architettura a pipeline ideale</p> <ul style="list-style-type: none"> • se ne descriva l'architettura e il funzionamento • si elenchino le principali cause che ne limitano il comportamento ideale.
14	<p>Si consideri un banco di memoria da 4 Mparole di 32 bit ciascuna, composto da moduli da 512Kparole da 8 bit ciascuno. Si risponda alle seguenti domande:</p> <ul style="list-style-type: none"> - quanti moduli sono necessari per realizzare l'intero bando di memoria? - chi pilota i segnali di indirizzo di ciascun modulo? - chi pilota il segnale di <i>chip enable</i> di ciascun modulo? - chi pilota i segnali di dato di ciascun modulo?

parte 3

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

Si scriva una procedura `maxInTriangolo` in linguaggio Assembly MIPS32 che restituisca il massimo valore fra gli elementi di una matrice quadrata che si trovano lungo il perimetro di un triangolo rettangolo isoscele.

La procedura riceve i seguenti parametri:

- \$a0: indirizzo della matrice quadrata di byte unsigned
- \$a1: indice del vertice del triangolo
- \$a2: dimensione della matrice quadrata

L'ipotenusa del triangolo rettangolo isoscele è parallela (eventualmente coincidente) alla diagonale secondaria della matrice. L'indice del vertice del triangolo passato in \$a1 si trova sempre nella prima riga della matrice (escluso il primo elemento).

Esempio 1: \$a1 = 5 \$a2 = 8. La procedura `maxInTriangolo` restituisce 79.

3	1	41	5	9	26	5	35
89	79	32	3	8	46	26	4
33	8	32	79	50	28	8	4
19	71	69	39	9	37	5	10
58	20	9	74	9	44	59	2
30	7	8	16	40	6	28	6
20	8	9	98	62	80	3	48
25	34	21	1	70	6	7	9

Esempio 2: \$a1 = 7, \$a2 = 8. La procedura `maxInTriangolo` restituisce 74.

3	1	41	5	9	26	5	35
89	79	32	3	8	46	26	4
33	8	32	79	50	28	8	4
19	71	69	39	9	37	5	10
58	20	9	74	9	44	59	2
30	7	8	16	40	6	28	6
20	8	9	98	62	80	3	48
25	34	21	1	70	6	7	9

Di seguito un esempio di programma chiamante:

`DIM = 8`

```
.data
matrice: .byte 3, 1, 41, 5, 9, 26, 5, 35
         .byte 89, 79, 32, 3, 8, 46, 26, 4
         .byte 33, 8, 32, 79, 50, 28, 8, 4
         .byte 19, 71, 69, 39, 9, 37, 5, 10
         .byte 58, 20, 9, 74, 9, 44, 59, 2
         .byte 30, 7, 8, 16, 40, 6, 28, 6
         .byte 20, 8, 9, 98, 62, 80, 3, 48
         .byte 25, 34, 21, 1, 70, 6, 7, 9

.text
.globl main
.ent main
main:
...
```

```

subu $sp, $sp, 4
sw $ra, ($sp)
...
la $a0, matrice
li $a1, 1
li $a2, DIM
jal maxInTriangolo
...           # lettura del risultato calcolato dalla procedura
lw $ra, ($sp)
addiu $sp, $sp, 4
jr $ra
.end main

```

Soluzione proposta

```

.ent maxInTriangolo
maxInTriangolo:
    add $t0, $a0, $a1 # indirizzo del vertice
    lb $v0, ($t0) # massimo corrente

    move $t1, $a1 # contatore iterazioni
    addi $t2, $a2, -1 # aggiornamento indirizzo
cicloIpotenusa:
    add $t0, $t0, $t2
    lb $t3, ($t0)
    bltu $t3, $v0, salto1
    move $v0, $t3
salto1: addi $t1, $t1, -1
    bnez $t1, cicloIpotenusa

    move $t1, $a1 # contatore iterazioni
cicloBase:
    addi $t0, $t0, 1
    lb $t3, ($t0)
    bltu $t3, $v0, salto2
    move $v0, $t3
salto2: addi $t1, $t1, -1
    bnez $t1, cicloBase

    addi $t1, $a1, -1 # contatore iterazioni
cicloAltezza:
    sub $t0, $t0, $a2
    lb $t3, ($t0)
    bltu $t3, $v0, salto3
    move $v0, $t3
salto3: addi $t1, $t1, -1
    bnez $t1, cicloAltezza

    jr $ra
.end maxInTriangolo

```