

Nome, cognome, matricola

Calcolatori Elettronici (12AGA) – esame del 20.7.2021

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

Tempo: 15 minuti.

1	Quante righe ha la tavola di verità di un codificatore prioritario 4:2 con enable?		
2	Si consideri un'unità di controllo microprogrammata basata su una memoria di microcodice composta da 510 parole da 18 bit: qual è il parallelismo del μ PC?	6 9 18 Nessuno dei precedenti	A B C D
3	Quale vantaggio offre una memoria secondaria di tipo SSD basata su MLC Flash rispetto a una memoria secondaria di tipo HDD?	Minore consumo di potenza Minor costo Capacità di memoria maggiore Velocità di scrittura maggiore	A B C D
4	Si consideri un bus sincrono. Che cosa si intende per ciclo di wait durante un'operazione di lettura?	Un ritardo periodico sul segnale di Ready Un ritardo aggiuntivo sul ciclo di lettura dati L'aggiunta di cicli aggiuntivi di attesa per ottenere più tempo Un ritardo periodico sul segnale di Clock	A B C D
5	Si consideri una cache set associative a 4 vie composta da 128 insiemi in cui ciascuna linea corrisponde a 32 byte che usa la tecnica nota come write-back. Calcolare quanti dirty bit sono presenti nell'intera cache.		
6	Quale dei seguenti dispositivi può diventare un master di un bus?	DMA controller Interfaccia di periferico Memoria Interrupt Controller	A B C D
7	A cosa serve il rinfresco delle memorie DRAM?	A ridurre gli effetti dei guasti indotti da radiazioni A ridurre il tempo di ciclo della memoria A risolvere il problema creato dal Destructive Read-Out A permettere alla memoria di mantenere nel tempo le informazioni	A B C D

8	Quale tra i fenomeni elencati a lato può causare uno stallo in un processore con pipeline?	Un miss nella cache dati	A	
		Un errore in un'operazione aritmetica	B	
		Un'operazione di I/O	C	
		L'esecuzione di un'istruzione NOP	D	

9	Assumendo che \$s0=a, \$s1=b, \$s2=h, \$s3=j Quale dei seguenti segmenti di codice realizza lo pseudo-codice C: if (a == b) h = j + 1 h = j + 2	beq \$s0, \$s1, L1 addi \$s2, \$s3, 1 L1: addi \$s2, \$s3, 2	A	
		bne \$s0, \$s1, L1 add \$s2, \$s3, 2 L1: add \$s2, \$s3, 1	B	
		bne \$s0, \$s1, L1 addi \$s2, \$s3, 1 L1: addi \$s2, \$s3, 2	C	
		beq \$s0, \$s1, L1 add \$s2, \$s3, 1 L1: add \$s2, \$s3, 2	D	

10	<p>Quale valore (in esadecimale) sarà presente in \$t1, dopo l'esecuzione dell'istruzione:</p> <p>andi \$t1, \$t2, 0xFFFFB</p> <p>assumendo che \$t2 contenga il valore 4?</p>	
----	--	--

Risposte corrette

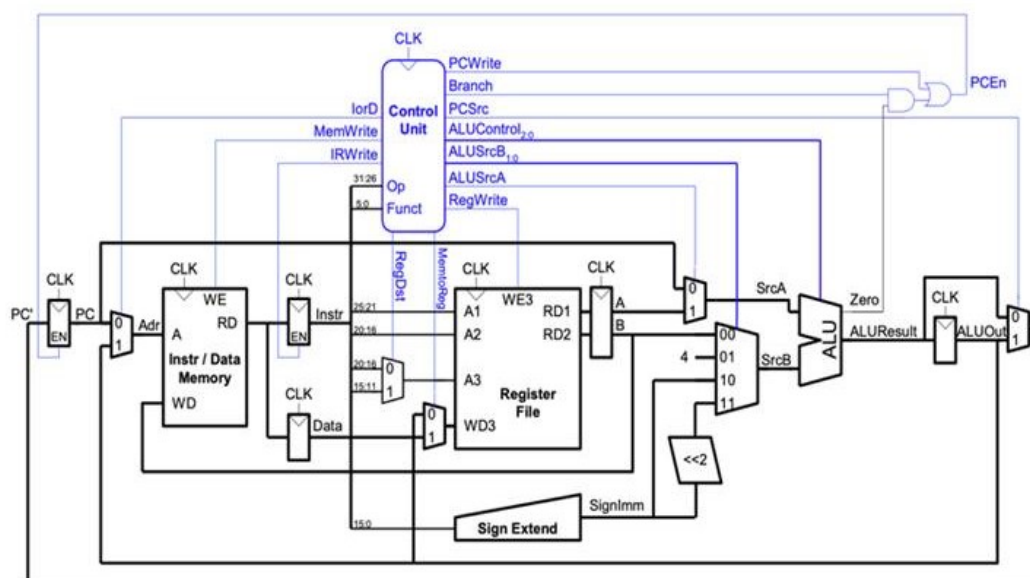
1	2	3	4	5	6	7	8	9	10
32	B	A	C	512	A	D	A	C	0

Parte 2

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 128kB e dotato di una cache direct mapped da 8 linee, ciascuna da 32 byte. Assumendo che inizialmente le 8 linee contengano i primi 8 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
	0 0000 0000 0011 0011			
	1 0000 0000 0001 1000			
	0 0000 0010 1000 1110			
	0 0000 0000 1011 1110			
	0 0000 0000 1001 1111			
	1 0000 0000 0011 0011			
	0 0000 0101 0001 0011			
	0 0000 1010 0101 0100			
	0 0000 0011 0011 0100			
	0 0100 0000 0011 0110			
	0 0000 1000 1001 1000			
	1 0000 0000 0011 0111			

12	Utilizzando la tabella riportata, si elenchino le micro-operazioni eseguite da un processore MIPS durante la fase di esecuzione (ignorando il fetch) dell'istruzione addi \$s1, \$s2, 40.
----	---



Funzione svolta	ALU Control _{2,0}
100000 (add)	010 (Add)
100010 (sub)	110 (Sub)
100100 (and)	000 (And)
100101 (or)	001 (Or)
101010 (slt)	111 (SLT)

[illegible]

13	<p>Con riferimento al meccanismo del DMA</p> <ul style="list-style-type: none">• Si descrivano le funzionalità offerte dal DMA Controller• Si evidenzino i vantaggi offerti dal DMA rispetto ad altri meccanismi di trasferimento dati da/verso periferiche, quali l'I/O programmato o l'interrupt• Si descrivano le connessioni tra il DMA controller e il resto del sistema• Si illustrino i passaggi attraverso i quali avviene un trasferimento in DMA, partendo dalla fase di programmazione da parte della CPU.
----	--

14	<p>Un sommatore può essere realizzato utilizzando varie architetture, tra le quali</p> <ul style="list-style-type: none">• Sommatore seriale• Sommatore di tipo Ripple Carry Adder• Sommatore di tipo Carry Lookahead Adder. <p>Per ciascuna di esse:</p> <ol style="list-style-type: none">1. Si elenchino i componenti hardware richiesti, specificando quanti componenti sono necessari per realizzare un sommatore su n bit e descrivendo i collegamenti tra i vari componenti2. Se ne descriva il funzionamento3. Si illustrino vantaggi e svantaggi.
----	---

Nome, cognome, matricola

20 luglio 2021 - parte 3: esercizio di programmazione

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

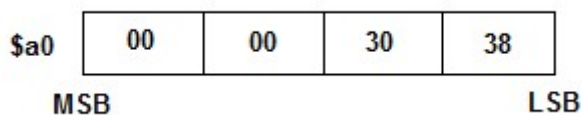
Si scriva una procedura `CalcoloCF` in linguaggio Assembly MIPS32 che restituisca una stringa contenente uno pseudo CodiceFiscale di 12 caratteri, a fronte dei seguenti parametri ricevuti:

- `$a0`: indirizzo della stringa contenente il COGNOME
- `$a1`: indirizzo della stringa contenente il NOME
- `$a2`: indirizzo della stringa contenente la data di nascita nel formato DDMMYY
- `$a3`: indirizzo della stringa contenente il CF-12 calcolato.

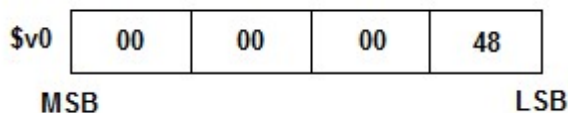
Questa la struttura del CF-12:

- Da 1 a 3, le prime 3 consonanti del COGNOME
- Da 4 a 6, le prime 3 consonanti del NOME
- Da 7 a 11, la data di nascita con lettera al posto del mese, ovvero Gennaio (01) diventa "A", Febbraio (02) diventa "B" e così via. Per calcolare la lettera corrispondente al mese dovete usare una procedura `MonthToChar` a cui passare in `$a0` (a partire dal LSB) i due caratteri estratti e corrispondenti al mese e che restituisce in `$v0` il carattere corrispondente.

Esempio: il mese di Agosto nella data è indicato con "08", corrispondente ai valori **0x30** e **0x38** nella tabella ASCII.



In uscita `MonthToChar` restituisce in `$v0` il valore "H", corrispondente al valore **0x48** nella tabella ASCII.



- Da 12 a 12, CRC calcolato come XOR di tutti i caratteri da 1 a 11.

Precisazioni:

- Tutti i Cognomi/Nomi hanno almeno 3 consonanti.
- Tutti caratteri alfabetici sono Maiuscoli.
- Non vengono richiesti controlli di congruenza dei dati in ingresso.
- Il 12° carattere della stringa CF-12 può non essere un carattere ASCII.
- La procedura `MonthToChar` NON è oggetto di questa prova d'esame e quindi NON dovrà essere sviluppata dallo studente.
- L'esempio di programma chiamante, qui sotto allegato, produce il codice **RSSGNN12H92g**

Di seguito un esempio di programma chiamante:

```
.data
cognome: .asciiz "ROSSI"
nome:    .asciiz "GENNARO"
data:    .asciiz "120892"
CF:      .space 12
```



```

        .text
        .globl main
        .ent main
main:    subu $sp, $sp, 4
        sw $ra, ($sp)

        la $a0, cognome
        la $a1, nome
        la $a2, data
        la $a3, CF
        jal CalcoloCF

        lw $ra, ($sp)
        addiu $sp, $sp, 4
        jr $ra
        .end main

        .ent MonthToChar
MonthToChar:
        # Parametro in ingresso $a0 = 2 CARATTERI ESTRATTI DALLA DATA
        # (esempio: "01" per Gennaio)
        # Parametro in uscita $v0 = "A" per l'esempio di Gennaio
        ... # codice della procedura da non sviluppare
        jr $ra
        .end MonthToChar

```

Soluzione proposta

```
.data
cognome: .asciiz "ROSSI"
nome: .asciiz "GENNARO"
data: .asciiz "120892"
CF: .space 12
```

```
.text
.globl main
.ent main
main: subu $sp, $sp, 4
sw $ra, ($sp)
la $a0, cognome
la $a1, nome
la $a2, data
la $a3, CF
jal CF12
lw $ra, ($sp)
addiu $sp, $sp, 4
jr $ra
.end main
```

```
.ent CalcoloCF
CalcoloCF:
move $t0, $a0
move $t1, $a3          # output
li $t3, 0 # flag
li $t4, 0 # contatore
li $t5, 0 # crc
ciclo: beq $t4, 3, controllo
lb $t2, ($t0)
beqz $t2, controllo
bltu $t2, 'B', next
bgtu $t2, 'Z', next
beq $t2, 'E', next
beq $t2, 'I', next
beq $t2, 'O', next
beq $t2, 'U', next
xor $t5, $t5, $t2
sb $t2, ($t1)          # Memorizzo il carattere
addiu $t1, $t1, 1
addiu $t4, $t4, 1
next: addiu $t0, $t0, 1
j ciclo

controllo:
addiu $t3, $t3, 1
move $t0, $a1
li $t4, 0
bne $t3, 2, ciclo

controllo_data:
move $t0, $a2
```

```

lb $t3, ($t0)
sb $t3, ($t1)
lb $t3, 1($t0)
sb $t3, 1($t1)
addiu $t0, $t0, 2          #posizione inizio MESE
addiu $t1, $t1, 2          #posizione stringa out
#
lb $a0, ($t0)
sll $a0, $a0, 8
lb $t3, 1($t0)
or $a0, $a0, $t3
jal MonthToChar
sb $v0, ($t1)
addiu $t0, $t0, 2          #posizione inizio ANNO
addiu $t1, $t1, 1          #posizione stringa out
lb $t3, ($t0)
sb $t3, ($t1)
lb $t3, 1($t0)
sb $t3, 1($t1)

move $t2, $a3
li $t3, 0                  #accumulatore
ciclo_xor:
bgt $t2, $t1, fine
lb $t9, ($t2)
xor $t3, $t3, $t9
addiu $t2, $t2, 1
j ciclo_xor

fine:
addiu $t1, $t1, 2          #posizione stringa per XOR
sb $t3, ($t1)
jr $ra

.end CalcoloCF

.ent MonthToChar
MonthToChar:
subu $sp, $sp, 4
sw $ra, ($sp)
subu $sp, $sp, 12
sw $t0, ($sp)
sw $t1, 4($sp)
sw $t2, 8($sp)

move $t0, $a0
srl $t0, $t0, 8
subu $t0, $t0, '0'
mul $t0, $t0, 10
move $t1, $a0
li $t2, 0xFF
and $t1, $t1, $t2
subu $t1, $t1, '0'
addu $t1, $t1, $t0

```

```
#in $t1 il numero  
subu $t1, $t1, 1    # per partire con A  
addu $v0, $t1, 'A'
```

```
lw $t2, 8($sp)  
lw $t1, 4($sp)  
lw $t0, ($sp)  
addiu $sp, $sp, 12  
lw $ra, ($sp)  
addiu $sp, $sp, 4  
jr $ra  
.end MonthToChar
```