

Nome, cognome, matricola

Calcolatori Elettronici (12AGA) – esame del 27.6.2017 - A

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 decoder con 3 ingressi e senza segnali di enable. Quante sono le sue linee di uscita?			
2	Si consideri un circuito sequenziale che adotta il modello di Huffman. Assumendo che la parte combinatoria abbia un ritardo pari a 100ns, quale tra i valori a lato può corrispondere alla frequenza massima di funzionamento del circuito?	80 KHz	A	
		8 MHz	B	
		80 MHz	C	
		800 MHz	D	
3	Si consideri un'unità di controllo microprogrammata la cui memoria di microcodice contiene 1580 parole da 71 bit. Quanti bit compongono il μ PC?	7	A	
		10	B	
		11	C	
		18	D	
4	A che cosa serve il rinfresco nelle memorie DRAM?	A ridurre gli effetti dei guasti indotti dalle radiazioni	A	
		A risolvere il problema creato dal fenomeno del Destructive Read-Out	B	
		A permettere alla memoria di mantenere nel tempo le informazioni	C	
		A ridurre il tempo di ciclo della memoria	D	
5	Si consideri il meccanismo della memoria virtuale. Quale delle seguenti affermazioni è vera?	Il meccanismo della memoria virtuale permette di scrivere programmi che fanno riferimento a indirizzi fisici	A	
		Il meccanismo della memoria virtuale è supportato interamente dal software	B	
		Il meccanismo della memoria virtuale è supportato interamente dall'hardware	C	
		Il meccanismo della memoria virtuale permette di scrivere programmi indipendenti dalla dimensione della memoria fisica esistente nel sistema	D	
6	Nella fase di programmazione del sistema, i 5 bit inviati all'8259 tramite la ICW2 sono 00110. Qual è il codice messo sul bus dall'8259 quando deve essere servita la periferica connessa a IR7?			
7	Si confrontino le tecniche CAV e CLV. Quale delle affermazioni a destra è vera?	La tecnica CAV permette di massimizzare la quantità di dati memorizzata sul disco.	A	
		La tecnica CAV permette di minimizzare la complessità del controllore del disco.	B	
		La tecnica CAV permette di massimizzare il numero di tracce presenti sul disco.	C	
		La tecnica CAV permette di massimizzare la quantità di dati memorizzata in ciascuna traccia sul disco.	D	
8	A quale delle seguenti istruzioni corrisponde un codice macchina su 6 byte?	ADD AX, 3580	A	
		MUL 3580	B	
		SUB VAR, 3580	C	
		MOV AX, 3580	D	
9	Si scriva un frammento di codice in Assembler x86 che determini quale tra i valori con segno presenti in AX e BX è più grande. Il frammento deve scrivere nella variabile X di tipo byte il valore 0 se AX contiene un valore maggiore o uguale a quello di BX, il valore 1 diversamente.			

Risposte corrette

1	2	3	4	5	6	7	8	9
8	B	C	C	D	55 (37h)	B	C	

Domanda 9 (esempio di soluzione)

```
CMP  AX, BX
JGE  L1
MOV  X, 1
JMP  FINE
```

L1: MOV X, 0

FINE:

Nome, cognome, matricola

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

10	Si disegni il circuito ottimizzato che implementa la funzione $ab' + abc + a'b'c' + a'bcd$.
11	Si consideri un sistema composto da 6 moduli che possono diventare master del bus, e si disegnino le connessioni tra i 6 moduli e l'arbitro nel caso si utilizzi il meccanismo del polling, illustrandone il funzionamento.

12	<p>Si consideri un sistema a microprocessore dotato di una memoria di 4Kbyte e di una cache set associative a 2 vie LRU composta di 8 linee da 32 byte ciascuna.</p> <p>Assumiamo che il processore acceda in memoria in sequenza ai seguenti blocchi (tra parentesi il corrispondente esadecimale): 25 (19), 50 (32), 60 (3C), 61 (3D), 100 (64), 122 (7A), 31 (1F), 35 (23). Al termine di tale sequenza di accessi la posizione dei blocchi nella cache è riportato nella figura di sinistra. Il processore continua poi ad accedere in memoria, accedendo in sequenza ai blocchi 41 (29), 50 (32), 52 (34), 60 (3C), 61 (3D), 62 (3E).</p> <p>Si riporti nella figura di destra il contenuto finale della cache.</p> <div><div><div>Insieme 0</div><table><tr><td>60</td></tr><tr><td>100</td></tr><tr><td>25</td></tr><tr><td>61</td></tr><tr><td>50</td></tr><tr><td>122</td></tr><tr><td>31</td></tr><tr><td>35</td></tr></table></div><div><div>Insieme 0</div><table><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr></table></div><div><div>Insieme 1</div><table><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr></table></div><div><div>Insieme 2</div><table><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr></table></div><div><div>Insieme 3</div><table><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr></table></div></div>	60	100	25	61	50	122	31	35																																
60																																									
100																																									
25																																									
61																																									
50																																									
122																																									
31																																									
35																																									
13	<p>Si elenchino le caratteristiche di un processore RISC.</p>																																								

Esercizio di programmazione

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

Un problema comune per compilatori ed editor di testo consiste nel determinare se le parentesi all'interno di una stringa sono bilanciate e correttamente annidate. Ad esempio:

- | | |
|-----------------------|---|
| 1. $((1+2)*3)-(4-8))$ | OK: parentesi bilanciate e correttamente annidate |
| 2. $(1+2)+4-1($ | NO: parentesi non correttamente annidate |
| 3. $(1+2*(5-3)$ | NO: parentesi non bilanciate |

Si scriva in linguaggio Assembly 8086 una procedura **verificaParentesi** che controlli la correttezza delle parentesi all'interno di una stringa. La lunghezza della stringa DIM è definita tramite costante. La procedura riceve l'offset della stringa tramite stack. Non è ammesso l'uso di variabili.

La procedura **verificaParentesi** deve restituire attraverso il registro BX:

- il valore DIM se le parentesi sono bilanciate e correttamente annidate
- altrimenti, la posizione della prima parentesi che non soddisfa i requisiti di bilanciamento e annidamento.

Negli esempi proposti, il valore restituito è:

- | | |
|-----------------------|----------|
| 1. $((1+2)*3)-(4-8))$ | BX = DIM |
| 2. $(1+2)+4-1($ | BX = 6 |
| 3. $(1+2*(5-3)$ | BX = 0 |

Una possibile realizzazione dell'algoritmo in pseudocodice è la seguente:

```
char s[DIM];
int par = 0;
for (i=0; i<DIM; i++)
{
    if (s[i] == '(')
    {
        par++;
        push(i);
    }
    else if (s[i] == ')')
    {
        par--;
        if (par < 0) return i;
        else pop();
    }
}
if (par!=0)
    return pop();
return DIM;
```

Di seguito un esempio di programma chiamante:

```
DIM EQU 17
.MODEL small
.STACK
.DATA
string DB "(((1+2)*3)-(4-8))"

.CODE
.STARTUP
...
PUSH OFFSET string
CALL verificaParentesi
ADD SP, 2
...
.EXIT
```