

Nome, cognome, matricola

Calcolatori Elettronici (12AGA) – esame del 23.6.2014

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 contatore sincrono in grado di contare in avanti e all'indietro tra 0 e 127: quanti flip flop compaiono nella sua implementazione?		
2	Si consideri un'unità di controllo microprogrammata basata su una memoria di microcodice composta da 650 parole da 58 bit: qual è il parallelismo del μ PC?	6	A
		10	B
		58	C
		Nessuno dei precedenti	D
3	Si consideri una memoria RAM che utilizza il codice di parità: quale delle seguenti affermazioni è <u>falsa</u> ?	Ogni parola ha un bit aggiuntivo per memorizzare il codice di parità	A
		Ogni volta che si legge una parola, se ne ricalcola il bit di parità e lo si confronta con il bit di parità memorizzato	B
		Ogni volta che si scrive una parola, si calcola il bit di parità del valore che si sta scrivendo, e lo si memorizza nell'apposito bit associato alla parola	C
		Se in fase di lettura si rileva una discrepanza tra il codice di parità memorizzato e quello ricalcolato, si procede alla correzione del valore letto	D
4	Si considerino i processori RISC: quale delle seguenti affermazioni è vera?	Tutte le istruzioni possono avere un operando memorizzato in una cella di memoria	A
		In assenza di stalli, tutte le istruzioni richiedono un solo colpo di clock per essere eseguite	B
		Esiste un'unica cache per dati e codice	C
		Solo le istruzioni di load e store possono accedere alla memoria	D
5	Si consideri un 8255 programmato in modo 1: a quanti indirizzi sarà associato?	1	A
		2	B
		4	C
		Dipende se il sistema adotta il metodo isolated IO o memory mapped	D
6	Quanti sono i segnali di ingresso / uscita dell'arbitro del bus in un sistema composto da 6 unità master che usa la tecnica del daisy chaining?	3	A
		6	B
		8	C
		12	D
7	Quale delle seguenti funzioni è svolta dalla Memory Management Unit (MMU)?	Gestione degli errori di parità verificatisi nella memoria principale	A
		Gestione dei dischi	B
		Trasferimento di blocchi di dati da memoria a periferico (e viceversa)	C
		Conversione degli indirizzi logici in fisici	D
8	Quale delle seguenti istruzioni assembler è corretta dal punto di vista sintattico (ossia è correttamente processata dall'assemblatore)?	MOV AX, TAB[BX][SI]	A
		ADD BX, [BX*2]	B
		SUB [SI], [DI]	C
		MUL AX, DX	D
9	Si supponga di aver memorizzato due numeri interi con segno in due variabili UNO e DUE (ciascuna su 32 bit). Si scriva un frammento di codice di 4 istruzioni in grado di sommare i due numeri e scrivere il risultato nella variabile DUE.		

Risposte corrette

1	2	3	4	5	6	7	8	9
7	b	d	d	c	a	d	a	

Domanda 9

MOV AX, word ptr UNO
ADD word ptr DUE, AX
MOV AX, word ptr UNO + 2
ADC word ptr DUE + 2, AX

Nome, cognome, matricola

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

10	<p>Si progetti una memoria composta da 1Mparole da 16 bit ciascuna utilizzando moduli composti ciascuno da 128Kparole da un byte ciascuna.</p>
11	<p>Si consideri un sistema avente un Address bus da 16 bit e una cache direct mapped composta da 8 linee da 64 byte ciascuna. Si assuma che inizialmente la cache contenga i blocchi 512, 513, 514, 515, 516, 517, 518, 519. Successivamente, il processore esegue una sequenza di 6 accessi in memoria, emettendo i seguenti indirizzi</p> <ol style="list-style-type: none">1. 0000 1000 0000 01002. 1000 0000 0100 01003. 0100 0001 1100 01004. 0100 0001 1100 01005. 0100 0001 1000 01006. 0000 0000 1000 0100 <p>Si determini per ogni accesso se esso causa un hit o miss, nonché il contenuto finale della cache (ossia quale blocco è contenuto in ciascuna linea al termine della sequenza).</p>

12	<p>Si descriva l'architettura e il funzionamento di un DMA Controller (precisandone le connessioni con il resto del sistema) e si illustrino i vantaggi che il suo uso offre.</p>
13	<p>Si descrivano le operazioni eseguite da un sistema a processore dal momento in cui una periferica (connessa all'Interrupt Controller) scatena una richiesta di interrupt, sino al momento in cui il processore esegue la prima istruzione della relativa procedura di servizio dell'interrupt. Si assuma che il sistema utilizzi il meccanismo dell'interrupt vettorizzato.</p>

Esercizio di programmazione

sino a 12 punti – è possibile consultare qualunque materiale cartaceo - tempo: 60 minuti

Si scriva una **procedura sostituisce** in grado di espandere una stringa precedentemente inizializzata sostituendo tutte le occorrenze del carattere '%' con un'altra stringa data. Siano date quindi le seguenti tre stringhe in memoria:

- `str_orig`, corrispondente al testo compresso da espandere
- `str_sost`, contenente la il testo da sostituire in `str_orig` al posto di '%'
- `str_new`, che conterrà la stringa espansa (si supponga che abbia dimensione sufficiente a contenerla).

Di seguito un esempio di funzionamento:

```
str_orig db "% nella citta' dolente, % nell'eterno dolore, % tra la perduta gente"
```

```
str_sost db "per me si va"
```

```
str_new sarà quindi
```

```
"per me si va nella citta' dolente, per me si va nell'eterno dolore, per me si va tra la  
perduta gente"
```

La procedura lavora sulle stringhe come *variabili globali*, riceve mediante *stack* la lunghezza di `str_orig` e di `str_sost`, e restituisce, sempre mediante *stack*, la lunghezza della nuova stringa. Di seguito un esempio di programma chiamante:

```
[...]
PUSH 68          ; lunghezza str_orig
PUSH 12          ; lunghezza str_sost
PUSH AX          ; lunghezza stringa finale
call sostituisce
POP lung_new
ADD SP, 4
[...]
```