

Prima prova in itinere di Calcolatori Elettronici – 25.11.2009

Corsi di Laurea in Ing. Gestionale e Ing. delle Telecomunicazioni

A.A. 2009-2010

Cognome		Nome		Matricola										
---------	--	------	--	-----------	--	--	--	--	--	--	--	--	--	--

Istruzioni: *Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova.*

Esercizio 1 – Rappresentazione dell'informazione (8 punti)

Si consideri il numero esadecimale 41F20000. Interpretando tale numero come la rappresentazione esadecimale di un numero codificato secondo lo standard IEEE-754 dei numeri in *floating point* a singola precisione, si determinino:

- 1) i valori binari di segno, esponente polarizzato, e mantissa di tale numero; (3pt)
- 2) il valore decimale di tale numero; (2pt)
- 3) si moltiplichi per -4 il numero ricavato al punto 2), quindi si calcoli la rappresentazione in complemento a due; (2pt)
- 4) si sommi questo numero a 15_{10} tramite addizione algebrica binaria su 8 bit. (1pt)

Nota. I numeri in complemento a 2 siano rappresentati su 8 bit.

Esercizio 2 – Reti combinatorie (8 punti)

Si consideri la funzione non completamente specificata rappresentata dalla mappa in figura. Scrivere l'espressione algebrica di tipo NAND della forma *minima* priva di alee.

$\begin{matrix} cd \\ ab \end{matrix}$	00	01	11	10
00	1	1	0	1
01	0	1	–	1
11	1	–	1	1
10	–	0	1	1

Esercizio 3 – Memorie (7 punti)

Avendo a disposizione dei chip di memoria RAM del tipo 1G x 8 bit, sintetizzare una memoria di tipo 4G x 32 bit, accessibile al byte, alle parole di 16 bit e alle parole di 32 bit. Disegnare il montaggio, dettagliando la logica combinatoria eventualmente utilizzata e commentando la soluzione scelta.

(2pt per l'accesso al byte, 2 pt per l'accesso ai 16 bit, 1pt per l'accesso ai 32 bit, 2pt per l'espansione indirizzi)

Esercizio 4 – Reti sequenziali sincronizzate (10 punti)

Si consideri una rete sequenziale sincronizzata di Moore avente due ingressi u e v ed una uscita z , la cui evoluzione temporale è descritta come segue:

$$Z(t_{i+1}) = | U(t_i) \cdot V(t_i) |_2$$

dove:

$Z(t_{i+1})$ è lo stato di uscita presente all'istante t_{i+1} ;

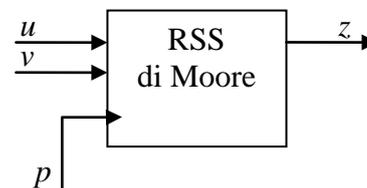
$U(t_i)$ è il numero di '1' ricevuti al piedino u fino all'istante t_i compreso;

$V(t_i)$ è il numero di '1' ricevuti al piedino v fino all'istante t_i compreso;

$t_0, t_1, \dots, t_i, t_{i+1}, \dots$ sono gli istanti di arrivo dei dati.

Si chiede di:

- 1) descrivere e commentare il diagramma degli stati e la tabella degli stati; (5pt)
- 2) disegnare il modello strutturale; (1pt)
- 3) riferendosi al modello strutturale considerato al punto precedente, eseguire la sintesi algebrica e circuitali. (4pt)



Seconda prova in itinere di Calcolatori Elettronici – 29.01.2010

Corsi di Laurea in Ing. Gestionale e Ing. delle Telecomunicazioni

A.A. 2009-2010

Cognome		Nome		Matricola										
---------	--	------	--	-----------	--	--	--	--	--	--	--	--	--	--

Istruzioni: *Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova.* E' possibile consultare l'elenco di istruzioni e degli interrupt del processore 8086 e del sistema operativo MS-DOS.

Esercizio 1 – Traduzione da linguaggio ad alto livello ad Assembler (9 punti)

Dato il seguente listato scritto in linguaggio C, tradurlo in linguaggio **Assembler 8086**

1. *Disegnare* lo stato dello *stack* per ogni chiamata di funzione *dopo l'allocazione delle variabili locali*, e
2. **Tradurre** il listato *commentando* opportunamente il codice scritto.

```
int s = 0, vett[10];

int sum(int a, int b) {
    int r;

    r = a + b;
    return r;
}
```

```
int main() {
    int i, l=5, t;

    while (i < l) {
        t = sum(vett[i], i);
        s = s + t;
        i++;
    }
    return s;
}
```

Esercizio 2 – Programmazione Assembler (15 punti)

Utilizzando il linguaggio Assembler 8086 e l'insieme di *interrupt* disponibili sui processori compatibili x86, si realizzi il programma che faccia quanto segue:

1. chiede all'utente di inserire una cifra esadecimale compresa tra 0 e F; (1pt)
2. se l'utente inserisce uno 0 il programma termina; (1pt)

```
Inserisci una cifra esadecimale compresa tra 0 e F: 5
 1  2  3  4  5
1: 01 02 03 04 05
2: 02 04 06 08 0A
3: 03 06 09 0C 0F
4: 04 08 0C 10 14
5: 05 0A 0F 14 19
```

3. se l'utente inserisce un numero compreso tra 1 ed F il programma stampa a video la tabella pitagorica esadecimale limitata al numero inserito; ad esempio se il numero inserito è 5 il programma stampa:

Si noti che il programma effettua i seguenti passi per creare questa stampa:

- a. Per stampare la riga delle intestazioni stampa 4 spazi quindi stampa il numero di intestazione della colonna seguito da due spazi: questo è un ciclo da 1 a al numero inserito. (3pt)
- b. Il programma stampa la tabella riga per riga. Per ogni riga stampa prima l'intestazione di riga seguita dal simbolo dei due punti (":"), quindi uno spazio, quindi i valori del prodotto intestazione di riga per intestazione di colonna. Il risultato è sempre un numero esadecimale stampato su due cifre seguito da uno spazio. (7pt)
- c. Il programma costruisce la tabella attraverso 2 cicli uno per le colonne ed uno per le righe (3pt)

Esercizio 3 – Questionario Architettura x86 (9 punti)

Rispondere alle seguenti domande in modo conciso ed esauriente.

Si descriva:

1. Memoria cache ad indirizzamento diretto.
2. Gestione delle unità di I/O con meccanismo delle interrupt: descrizione del funzionamento, vantaggi e svantaggi.
3. Interfaccia parallele di ingresso senza *handshake* ed interfaccia parallela di uscita senza *handshake*.
4. Arbitraggio centralizzato del bus.

```
.MODEL small
.stack 100h
.data
    pkey db 13,10,10,"Press key to exit...$" ; Messaggio di fine programma.
.code
start:
    mov ax,@data ; Get the address of the data segment
    mov ds,ax ; Set the DS segment
; Inizio esercizio
; ... esercizio da svolgere ...
; Fine esercizio
    mov dx,offset pkey
    mov ah,9
    int 21h
    mov ax,0C07h
    int 21h
    mov ax, 4C00h
    int 21h
; Legge un carattere (SENZA ECO) dallo standard input (di solito la tastiera)
; Parametri in ingresso: nessuno
; Il carattere letto e' ritornato nel registro AL.
input proc
    mov ax,0C07h ; Function 0Ch = "FLUSH BUFFER AND READ STANDARD INPUT"
    int 21h
    ret
input endp

; output
; Invia un carattere sullo standard output (di solito il video)
; Parametri in ingresso: AL
output proc
    push ax
    push dx
    mov ah, 2
    mov dl, al
    int 21h
    pop dx
    pop ax
    Ret
output endp

; print
; Stampa la stringa presente alla locazione di memoria DS:DX;
; tale stringa deve terminare con il segno "$".
; Parametri in ingresso: DS:DX
; Nessun parametro in uscita.
print proc
    push ax
    mov ah,9 ; Function 09h in AH means "WRITE STRING TO STANDARD OUTPUT"
    int 21h ; Call the DOS interrupt (DOS function call)
    pop ax
    ret
print endp
end start
```

Esame di Calcolatori Elettronici – 29.01.2010

Corsi di Laurea in Ing. Gestionale e Ing. delle Telecomunicazioni

A.A. 2009-2010

Cognome		Nome		Matricola							
---------	--	------	--	-----------	--	--	--	--	--	--	--

Istruzioni: *Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova.* E' possibile consultare l'elenco di istruzioni e degli interrupt del processore 8086 e del sistema operativo MS-DOS.

Esercizio 1 – Rappresentazione dell'informazione

Si considerino i seguenti numeri in base 10: $X_{10} = 13$, $Y_{10} = -5$; mostrando tutti i passaggi necessari, calcolare:

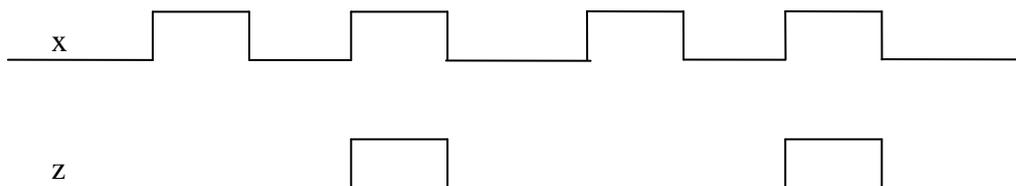
- 1) la rappresentazione in complemento a 2 su 8 bit di X_{10} e di Y_{10} (siano X_{C2} e Y_{C2});
- 2) la somma su 8 bit $S_{C2} = X_{C2} + Y_{C2}$ dei due numeri binari ottenuti al punto precedente;
- 3) la divisione $D_{C2} = X_{C2} / S_{C2}$ ed il resto R_{C2} , ove D_{C2} e S_{C2} sono rappresentati su 8 bit;
- 4) la rappresentazione secondo lo standard IEEE/ANSI 754 dei numeri *floating point* su 4 byte di D_{C2} ;
- 5) la rappresentazione *esadecimale* del numero calcolato al punto precedente.

Esercizio 2 – Reti sequenziali asincrone

Si consideri una Rete Sequenziale Asincrona avente un ingresso x ed una uscita z .

Inizialmente la RSA è in una situazione di stabilità con $x=0$, e $z=0$. La sua evoluzione temporale è tale che:

1. $z=1$ quando sul piedino x si è presentato per due volte *distinte* il livello logico alto.
2. l'uscita z torna a "0", non appena l'ingresso x torna a livello logico basso. Esempio:



Per tale rete:

- 1) si disegni il corrispondente diagramma di flusso,
- 2) il modello strutturale adottato, e
- 3) si sintetizzi il suo circuito logico.

Esercizio 2 – Programmazione Assembler (15 punti)

Utilizzando il linguaggio Assembler 8086 e l'insieme di *interrupt* disponibili sui processori compatibili x86, si realizzi il programma che faccia quanto segue:

1. chiede all'utente di inserire una cifra esadecimale compresa tra 0 e F; (1pt)
2. se l'utente inserisce uno 0 il programma termina; (1pt)

```
Inserisci una cifra esadecimale compresa tra 0 e F: 5
 1  2  3  4  5
1: 01 02 03 04 05
2: 02 04 06 08 0A
3: 03 06 09 0C 0F
4: 04 08 0C 10 14
5: 05 0A 0F 14 19
```

3. se l'utente inserisce un numero compreso tra 1 ed F il programma stampa a video la tabella pitagorica esadecimale limitata al numero inserito; ad esempio se il numero inserito è 5 il programma stampa:

Si noti che il programma effettua i seguenti passi per creare questa stampa:

- a. Per stampare la riga delle intestazioni stampa 4 spazi quindi stampa il numero di intestazione della colonna seguito da due spazi: questo è un ciclo da 1 a al numero inserito. (3pt)
- b. Il programma stampa la tabella riga per riga. Per ogni riga stampa prima l'intestazione di riga seguita dal simbolo dei due punti (":"), quindi uno spazio, quindi i valori del prodotto intestazione di riga per intestazione di colonna. Il risultato è sempre un numero esadecimale stampato su due cifre seguito da uno spazio. (7pt)
- c. Il programma costruisce la tabella attraverso 2 cicli uno per le colonne ed uno per le righe (3pt)

Esercizio 3 – Questionario Architettura x86 (9 punti)

Rispondere alle seguenti domande in modo conciso ed esauriente.

Si descriva:

1. Memoria cache ad indirizzamento diretto.
2. Gestione delle unità di I/O con meccanismo delle interrupt: descrizione del funzionamento, vantaggi e svantaggi.
3. Interfaccia parallele di ingresso senza *handshake* ed interfaccia parallela di uscita senza *handshake*.
4. Arbitraggio centralizzato del bus.

```
.MODEL small
.stack 100h
.data
    pkey db 13,10,10,"Press key to exit...$" ; Messaggio di fine programma.
.code
start:
    mov ax,@data ; Get the address of the data segment
    mov ds,ax ; Set the DS segment
; Inizio esercizio
; ... esercizio da svolgere ...
; Fine esercizio
    mov dx,offset pkey
    mov ah,9
    int 21h
    mov ax,0C07h
    int 21h
    mov ax, 4C00h
    int 21h
; Legge un carattere (SENZA ECO) dallo standard input (di solito la tastiera)
; Parametri in ingresso: nessuno
; Il carattere letto e' ritornato nel registro AL.
input proc
    mov ax,0C07h ; Function 0Ch = "FLUSH BUFFER AND READ STANDARD INPUT"
    int 21h
    ret
input endp

; output
; Invia un carattere sullo standard output (di solito il video)
; Parametri in ingresso: AL
output proc
    push ax
    push dx
    mov ah, 2
    mov dl, al
    int 21h
    pop dx
    pop ax
    Ret
output endp

; print
; Stampa la stringa presente alla locazione di memoria DS:DX;
; tale stringa deve terminare con il segno "$".
; Parametri in ingresso: DS:DX
; Nessun parametro in uscita.
print proc
    push ax
    mov ah,9 ; Function 09h in AH means "WRITE STRING TO STANDARD OUTPUT"
    int 21h ; Call the DOS interrupt (DOS function call)
    pop ax
    ret
print endp
end start
```

Esame di Calcolatori Elettronici – 15.02.2010
Corsi di Laurea in Ing. Gestionale e Ing. delle Telecomunicazioni
A.A. 2009-2010

Cognome		Nome		Matricola					
---------	--	------	--	-----------	--	--	--	--	--

Istruzioni: *Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova.*

Esercizio 1 – Rappresentazione dell'informazione (8 punti)

Si consideri il numero esadecimale 41A60000. Interpretando tale numero come la rappresentazione esadecimale di un numero codificato secondo lo standard IEEE-754 dei numeri in *floating point* a singola precisione, si determinino:

- 1) i valori binari di segno, esponente polarizzato, e mantissa di tale numero; (2pt)
- 2) il valore decimale di tale numero; (2pt)
- 3) si moltiplichi per -4 il numero ricavato al punto 2), quindi si calcoli la rappresentazione in complemento a due; (2pt)
- 4) si sommi questo numero a 17_{10} tramite addizione algebrica binaria su 8 bit. (2pt)

Nota. I numeri in complemento a 2 siano rappresentati su 8 bit.

Esercizio 4 – Reti sequenziali sincronizzate (10 punti)

Descrivere e sintetizzare una rete sincronizzata con **due** ingressi *a, b* ed un'uscita *z*. L'uscita è sempre a zero, tranne quando si presenta una sequenza di tre stati di ingresso consecutivi aventi le seguenti caratteristiche:

- 1) primo stato di ingresso: $a=0, b=0$;
- 2) secondo stato di ingresso: $a=1, b=0$;
- 3) terzo stato di ingresso: $a=0, b=1$.

Per tale rete:

- 1) si disegni il corrispondente diagramma di flusso; (3pt)
- 2) il modello strutturale adottato nel caso in cui si utilizzino flip-flop J-K come sottorete sincrona elementare; (2pt)
- 3) si sintetizzi il suo circuito logico. (5pt)

Esercizio 4 – Programmazione Assembler (15 punti)

Utilizzando il linguaggio Assembler 8086 e l'insieme di *interrupt* disponibili sui processori compatibili x86, si realizzi il programma che faccia quanto segue.

1. Nel segmento dati dichiara:
 - a. una stringa di 9 valori di nome simbolico MAT il cui scopo è contenere gli elementi di una matrice 3x3; (1pt)
 - b. una stringa di 3 valori di nome simbolico VETT il cui scopo è contenere gli elementi un vettore di 3 elementi. (1pt)
2. Chiede all'utente di inserire:
 - a. i valori dei 9 elementi della matrice, come numeri decimali ASCII compresi tra 0 e 5; (1pt)
 - b. i valori dei 3 elementi del vettore, come numeri decimali ASCII compresi tra 0 e 5. (1pt)

3. Stampa a video:
 - a. la matrice così inserita; (2pt)
 - b. il vettore così inserito; (2pt)
 - c. il vettore risultato del prodotto *matrice x vettore*; (5pt)
 - d. i precedenti punti a,b e c sono stampati a video con la formattazione mostrata nell'esempio. (2pt)

A titolo di esempio si mostra qui a lato una tipica esecuzione del programma.

```
Inserisci l'elemento 0,0 della matrice: 1
Inserisci l'elemento 0,1 della matrice: 2
Inserisci l'elemento 0,2 della matrice: 3
Inserisci l'elemento 1,0 della matrice: 4
Inserisci l'elemento 1,1 della matrice: 5
Inserisci l'elemento 1,2 della matrice: 1
Inserisci l'elemento 2,0 della matrice: 5
Inserisci l'elemento 2,1 della matrice: 5
Inserisci l'elemento 2,2 della matrice: 5
Inserisci l'elemento 0 del vettore: 1
Inserisci l'elemento 1 del vettore: 2
Inserisci l'elemento 2 del vettore: 3
Risultato prodotto matrice-vettore:
|1 2 3| |1| |14|
|4 5 1| |2| |17|
|5 5 5| |3| |30|
```

Recupero I^a prova in itinere di Calcolatori Elettronici – 15.02.2010

Corsi di Laurea in Ing. Gestionale e Ing. delle Telecomunicazioni

A.A. 2009-2010

Cognome	Nome	Matricola					
---------	------	-----------	--	--	--	--	--

Istruzioni: *Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova.*

Esercizio 1 – Rappresentazione dell'informazione (8 punti)

Si consideri il numero esadecimale 41A60000. Interpretando tale numero come la rappresentazione esadecimale di un numero codificato secondo lo standard IEEE-754 dei numeri in *floating point* a singola precisione, si determinino:

- 1) i valori binari di segno, esponente polarizzato, e mantissa di tale numero; (2pt)
- 2) il valore decimale di tale numero; (2pt)
- 3) si moltiplichi per -4 il numero ricavato al punto 2), quindi si calcoli la rappresentazione in complemento a due; (2pt)
- 4) si sommi questo numero a 17_{10} tramite addizione algebrica binaria su 8 bit. (2pt)

Nota. I numeri in complemento a 2 siano rappresentati su 8 bit.

Esercizio 2 – Memorie (6 punti)

Avendo a disposizione dei chip di memoria RAM del tipo 2G x 16 bit, sintetizzare una memoria di tipo 4G x 32 bit, accessibile alle parole di 16 bit e alle parole di 32 bit. Disegnare il montaggio, dettagliando la logica combinatoria eventualmente utilizzata e commentando la soluzione scelta.

(2 pt per l'accesso ai 16 bit, 1pt per l'accesso ai 32 bit, 2pt per l'espansione indirizzi, 1pt per la corrett./accurat.)

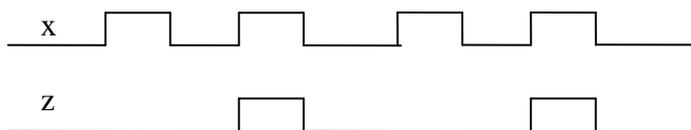
Esercizio 3 – Reti sequenziali asincrone (9 punti)

Si consideri una Rete Sequenziale Asincrona avente un ingresso x ed una uscita z .

Inizialmente la RSA è in una situazione di stabilità con $x=0$, e $z=0$. La sua evoluzione temporale è tale che:

1. $z=1$ quando sul piedino x si è presentato per due volte *distinte* il livello logico alto.
2. l'uscita z torna a "0", non appena l'ingresso x torna a livello logico basso.

Esempio:



Per tale rete:

- 1) si disegni il corrispondente diagramma di flusso, (3pt)
- 2) il modello strutturale adottato, e (2pt)
- 3) si sintetizzi il suo circuito logico. (4pt)

Esercizio 4 – Reti sequenziali sincronizzate (10 punti)

Descrivere e sintetizzare una rete sincronizzata con **due** ingressi a , b ed un'uscita z . L'uscita è sempre a zero, tranne quando si presenta una sequenza di tre stati di ingresso consecutivi aventi le seguenti caratteristiche:

- 1) primo stato di ingresso: $a=0, b=0$;
- 2) secondo stato di ingresso: $a=1, b=0$;
- 3) terzo stato di ingresso: $a=0, b=1$.

Per tale rete:

- 1) si disegni il corrispondente diagramma di flusso; (3pt)
- 2) il modello strutturale adottato nel caso in cui si utilizzino flip-flop J-K come sottorete sincrona elementare; (2pt)
- 3) si sintetizzi il suo circuito logico. (5pt)

Recupero II^a prova in itinere di Calcolatori Elettronici – 15.02.2010

Corsi di Laurea in Ing. Gestionale e Ing. delle Telecomunicazioni

A.A. 2009-2010

Cognome	Nome	Matricola					
---------	------	-----------	--	--	--	--	--

Istruzioni: *Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova.* E' possibile consultare l'elenco di istruzioni e degli interrupt del processore 8086 e del sistema operativo MS-DOS.

Esercizio 1 – Traduzione da linguaggio ad alto livello ad Assembler (9 punti)

Dato il seguente listato scritto in linguaggio C, tradurlo in linguaggio **Assembler 8086**

1. *Disegnare* lo stato dello *stack* per ogni chiamata di funzione *dopo l'allocazione delle variabili locali*, e
2. **Tradurre** il listato *commentando* opportunamente il codice scritto.

```
int s = 0;

int sub(int a, int b) {
    int d;

    d = a - b;

    return r;
}
```

```
int main() {
    int vett[10], i;

    for (i = 0; i < 10; i++) {
        vett[i] = diff(i*2, i);
        s = s + vett[i];
    }

    return s;
}
```

Esercizio 2 – Programmazione Assembler (15 punti)

Utilizzando il linguaggio Assembler 8086 e l'insieme di *interrupt* disponibili sui processori compatibili x86, si realizzi il programma che faccia quanto segue.

1. Nel segmento dati dichiara:
 - a. una stringa di 9 valori di nome simbolico MAT il cui scopo è contenere gli elementi di una matrice 3x3; (1pt)
 - b. una stringa di 3 valori di nome simbolico VETT il cui scopo è contenere gli elementi un un vettore di 3 elementi. (1pt)
2. Chiede all'utente di inserire:
 - a. i valori dei 9 elementi della matrice, come numeri decimali ASCII compresi tra 0 e 5; (1pt)
 - b. i valori dei 3 elementi del vettore, come numeri decimali ASCII compresi tra 0 e 5. (1pt)
3. Stampa a video:
 - a. la matrice così inserita; (2pt)
 - b. il vettore così inserito; (2pt)
 - c. il vettore risultato del prodotto *matrice x vettore*; (5pt)
 - d. i precedenti punti a,b e c sono stampati a video con la formattazione mostrata nell'esempio. (2pt)

A titolo di esempio si mostra qui a lato una tipica esecuzione del programma.

```
Inserisci l'elemento 0,0 della matrice: 1
Inserisci l'elemento 0,1 della matrice: 2
Inserisci l'elemento 0,2 della matrice: 3
Inserisci l'elemento 1,0 della matrice: 4
Inserisci l'elemento 1,1 della matrice: 5
Inserisci l'elemento 1,2 della matrice: 1
Inserisci l'elemento 2,0 della matrice: 5
Inserisci l'elemento 2,1 della matrice: 5
Inserisci l'elemento 2,2 della matrice: 5
Inserisci l'elemento 0 del vettore: 1
Inserisci l'elemento 1 del vettore: 2
Inserisci l'elemento 2 del vettore: 3
Risultato prodotto matrice-vettore:
|1 2 3| |1| |14|
|4 5 1| |2| |17|
|5 5 5| |3| |30|
```

Esercizio 3 – Questionario Architettura x86 (9 p.ti)

Rispondere alle seguenti domande in modo conciso ed esauriente, disegnando schemi e diagrammi ove necessario.

Si descriva:

1. Memoria cache ad indirizzamento associativo a gruppi.
2. Il meccanismo di gestione delle priorità nelle interruzioni: funzionamento e realizzazione circuitale.
3. Interfaccia parallela di ingresso con *handshake*: temporizzazione, schema funzionale e funzionamento.
4. Arbitro del bus a priorità rotante: schema circuitale e funzionamento.

Esame di Calcolatori Elettronici – 04.03.2010
Corsi di Laurea in Ing. Gestionale e Ing. delle Telecomunicazioni
A.A. 2009-2010

Cognome		Nome		Matricola					
---------	--	------	--	-----------	--	--	--	--	--

Istruzioni: *Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova.*

Esercizio 1 – Rappresentazione dell'informazione (8 punti)

Si consideri il numero esadecimale 41640000. Interpretando tale numero come la rappresentazione esadecimale di un numero codificato secondo lo standard IEEE-754 dei numeri in *floating point* a singola precisione, si determinino:

- 1) i valori binari di segno, esponente polarizzato, e mantissa di tale numero; (2pt)
- 2) il valore decimale di tale numero; (2pt)
- 3) si moltiplichi per -4 il numero ricavato al punto 2), quindi si calcoli la rappresentazione in complemento a due; (2pt)
- 4) si sommi questo numero a 3₁₀ tramite addizione algebrica binaria su 8 bit. (2pt)

Nota. I numeri in complemento a 2 siano rappresentati su 8 bit.

Esercizio 2 – Reti sequenziali asincrone (10 punti)

Realizzare un circuito *sequenziale asincrono* avente due ingressi x1 e x2 ed un uscita z, che partendo da una situazione di stabilità con x1=0, x2=0 e z=0 evolva come segue:

- quando x1=0, l'uscita z assume il valore di x2;
- quando x1=1, l'uscita z assume il valore di /x2.

Per tale rete:

- 1) si disegni il corrispondente diagramma di flusso, (3pt)
- 2) il modello strutturale adottato, e (2pt)
- 3) si sintetizzi il suo circuito logico. (3pt)
- 4) Si discuta il caso delle transizioni multiple in ingresso. (2pt)

Esercizio 3 – Programmazione Assembler (15 punti)

Utilizzando il linguaggio Assembler 8086 e l'insieme di *interrupt* disponibili sui processori compatibili x86, si realizzi il programma che faccia quanto segue.

1. Nel segmento dati dichiara due stringa di 4 valori di nome simbolico MAT1 e MAT2 il cui scopo è contenere gli elementi di due matrice 2x2; (1pt)
2. Chiede all'utente di inserire:
 - a. i valori dei 4 elementi della matrice MAT1, come numeri decimali ASCII compresi tra 0 e 7;(1pt)
 - b. i valori dei 4 elementi della matrice MAT2, come numeri decimali ASCII compresi tra 0 e 7.(1pt)
3. Stampa a video:
 - a. le due matrici così inserita; (3pt)
 - b. la matrice risultato della somma di matrici avente le somme espresse in esadecimale; (5pt)
 - c. i precedenti punti a,b e c sono stampati a video con la formattazione mostrata nell'esempio. (2pt)

A titolo di esempio si mostra qui a lato una tipica esecuzione del programma.

```

Inserisci l'elem. 0,0 della matr.1: 1
Inserisci l'elem. 0,1 della matr.1: 2
Inserisci l'elem. 1,0 della matr.1: 4
Inserisci l'elem. 1,1 della matr.1: 5
Inserisci l'elem. 0,0 della matr.2: 3
Inserisci l'elem. 0,1 della matr.2: 4
Inserisci l'elem. 1,0 della matr.2: 2
Inserisci l'elem. 1,1 della matr.2: 5
Matrice 1:
|1 2|
|4 5|
Matrice 2:
|3 4|
|2 5|
Somma matrici:
|4 6|
|6 A|
    
```

Recupero 1^a prova in itinere di Calcolatori Elettronici – 04.03.2010

Corsi di Laurea in Ing. Gestionale e Ing. delle Telecomunicazioni

A.A. 2009-2010

Cognome	Nome	Matricola					
---------	------	-----------	--	--	--	--	--

Istruzioni: *Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova.*

Esercizio 1 – Rappresentazione dell'informazione (8 punti)

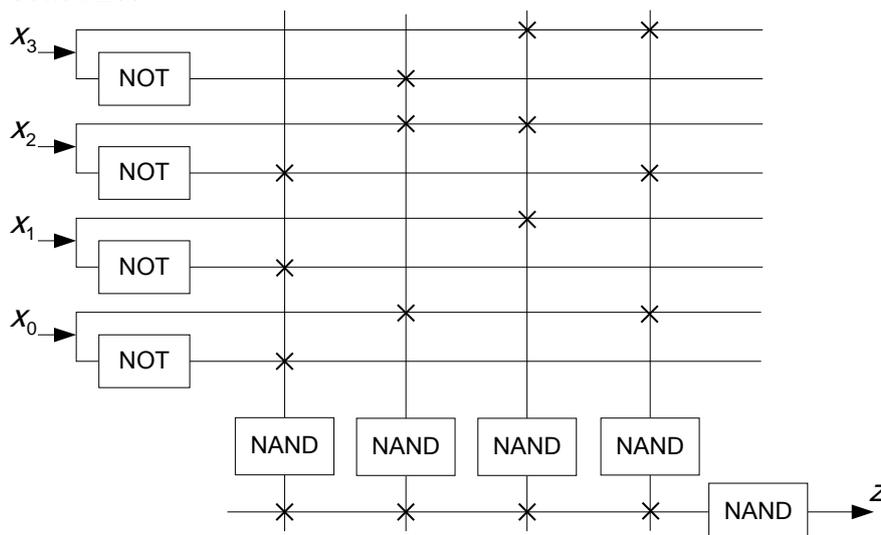
Si consideri il numero esadecimale 41640000. Interpretando tale numero come la rappresentazione esadecimale di un numero codificato secondo lo standard IEEE-754 dei numeri in *floating point* a singola precisione, si determinino:

- 1) i valori binari di segno, esponente polarizzato, e mantissa di tale numero; (2pt)
- 2) il valore decimale di tale numero; (2pt)
- 3) si moltiplichi per -4 il numero ricavato al punto 2), quindi si calcoli la rappresentazione in complemento a due; (2pt)
- 4) si sommi questo numero a 3_{10} tramite addizione algebrica binaria su 8 bit. (2pt)

Nota. I numeri in complemento a 2 siano rappresentati su 8 bit.

Esercizio 2 – Reti combinatorie (6 punti)

Verificare e giustificare il fatto che il circuito in figura è affetto da Alee del I ordine. Modificare poi il circuito in modo da eliminare dette Alee.



Esercizio 3 – Reti sequenziali asincrone (10 punti)

Realizzare un circuito *sequenziale asincrono* avente due ingressi x_1 e x_2 ed un uscita z , che partendo da una situazione di stabilità con $x_1=0$, $x_2=0$ e $z=0$ evolva come segue:

- quando $x_1=0$, l'uscita z assume il valore di x_2 ;
- quando $x_1=1$, l'uscita z assume il valore di $\neg x_2$.

Per tale rete:

- 1) si disegni il corrispondente diagramma di flusso, (3pt)
- 2) il modello strutturale adottato, e (2pt)
- 3) si sintetizzi il suo circuito logico. (3pt)
- 4) Si discuta il caso delle transizioni multiple in ingresso. (2pt)

Recupero II^a prova in itinere di Calcolatori Elettronici – 04.03.2010

Corsi di Laurea in Ing. Gestionale e Ing. delle Telecomunicazioni

A.A. 2009-2010

Cognome	Nome	Matricola					
---------	------	-----------	--	--	--	--	--

Istruzioni: *Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova.* E' possibile consultare l'elenco di istruzioni e degli interrupt del processore 8086 e del sistema operativo MS-DOS.

Esercizio 1 – Traduzione da linguaggio ad alto livello ad Assembler (8 punti)

Dato il seguente listato scritto in linguaggio C, tradurlo in linguaggio **Assembler 8086**

1. *Disegnare* lo stato dello *stack* per ogni chiamata di funzione *dopo l'allocazione delle variabili locali*, e
2. **Tradurre** il listato *commentando* opportunamente il codice scritto.

```
char c = 0;

int calc(int a) {
    int d;

    d = a * 4;

    return r;
}
```

```
int main() {
    int i, m;

    for (i = 16; i < 20; i++) {
        m = calc(i);
        c = (char) (m + 1);
    }

    return c;
}
```

Esercizio 2 – Programmazione Assembler (15 punti)

Utilizzando il linguaggio Assembler 8086 e l'insieme di *interrupt* disponibili sui processori compatibili x86, si realizzi il programma che faccia quanto segue.

1. Nel segmento dati dichiara due stringa di 4 valori di nome simbolico MAT1 e MAT2 il cui scopo è contenere gli elementi di due matrice 2x2; (1pt)
2. Chiede all'utente di inserire:
 - a. i valori dei 4 elementi della matrice MAT1, come numeri decimali ASCII compresi tra 0 e 7;(1pt)
 - b. i valori dei 4 elementi della matrice MAT2, come numeri decimali ASCII compresi tra 0 e 7.(1pt)
3. Stampa a video:
 - a. le due matrici così inserita; (4pt)
 - b. la matrice risultato della somma di matrici avente le somme espresse in esadecimale; (5pt)
 - c. i precedenti punti a,b e c sono stampati a video con la formattazione mostrata nell'esempio. (3pt)

A titolo di esempio si mostra qui a lato una tipica esecuzione del programma.

Esercizio 3 – Questionario Architettura x86 (10 p.ti)

Rispondere alle seguenti domande in modo conciso ed esauriente, disegnando schemi e diagrammi ove necessario.

Si descriva:

1. Descrivere funzionamento ed utilità del sistema di memoria virtuale.
2. *Direct Memory Access* (DMA) e controllore di DMA: funzionamento, vantaggi e svantaggi.
3. Arbitro del bus a priorità fissa: schema circuitale e funzionamento.
4. Il bus asincrono: descrizione del funzionamento e dei segnali impiegati nel caso di ciclo di lettura.

```
Inserisci l'elem. 0,0 della matr.1: 1
Inserisci l'elem. 0,1 della matr.1: 2
Inserisci l'elem. 1,0 della matr.1: 4
Inserisci l'elem. 1,1 della matr.1: 5
Inserisci l'elem. 0,0 della matr.2: 3
Inserisci l'elem. 0,1 della matr.2: 4
Inserisci l'elem. 1,0 della matr.2: 2
Inserisci l'elem. 1,1 della matr.2: 5
Matrice 1:
|1 2|
|4 5|
Matrice 2:
|3 4|
|2 5|
Somma matrici:
|4 6|
|6 A|
```