Prima prova in itinere di Calcolatori Elettronici – 18.11.2008

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

A.A. 2008-2009

Cognome		Nome		Matricola								

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

Esercizio 1 – Rappresentazione dell'informazione (6 punti)

Si considerino i seguenti numeri $X_{10} = 32$ e $Y_{10} = 56{,}25$. Determinare:

1) la loro rappresentazione in complemento a 2: $\mathbf{X_{C2}}$ e $\mathbf{Y_{C2}}$; (1pt)
2) la sottrazione $\mathbf{D_{C2} = X_{C2} - Y_{C2}}$ (2pt)
3) la rappresentazione secondo lo standard **IEEE/ANSI 754 su 4 byte di D_{C2}** (2pt); (2pt)
4) la rappresentazione **ottale** ed **esadecimale** della stringa di 32 bit calcolato al punto precedente. (1pt)

Nota. I numeri in complemento a 2 siano rappresentati considerando 8 bit per la parte intera e 8 bit per la parte frazionaria.

Esercizio 2 – Reti combinatorie (*8 punti*)

Avendo a disposizione un addizionatore per 2 numeri naturali a 8 bit A e B, *realizzare* (disegnare) e *commentare* una ALU in grado di eseguire le operazioni di

1) somma A + B, (2pt)
2) incremento di B, (2pt)
3) differenza A – B. (2pt)

Nota. Si utilizzino le porte logiche ritenute necessarie per realizzare la logica combinatoria richiesta.

Esercizio 3 – Memorie (6 *punti*)

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

(2pt per l'acceso ai 4 bit, 2 pt per l'accesso al byte, 1pt per l'accesso ai 16 bit, 1pt per l'espansione indirizzi)

Esercizio 4 – Reti sequenziali asincrone (*8 punti*)

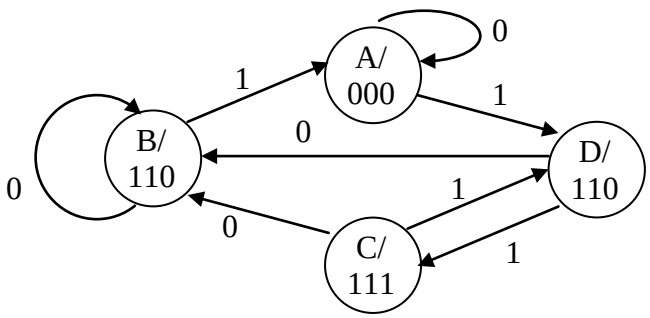
Descrivere e sintetizzare una rete sequenziale asincrona due ingressi x_1 e x_0 e un uscita z, che, partendo da uno stato di riposo in cui ingressi e uscita sono a livello 0, pone l'uscita a 1 quando riconosce **o** la sequenza di stati di ingresso 00,01 **o** la sequenza di stati di ingresso 10,11.

Per tale rete, pilotata in modo *fondamentale* e *senza transizione multiple in ingresso*, si richiede di:

1) disegnare il diagramma di flusso; (2pt)
2) indicare il modello strutturale a cui fare riferimento per la sintesi; (1pt)
3) eseguire la sintesi algebrica e circuitale; (3pt)
4) discutere il comportamento della rete in caso di transizioni multiple in ingresso. (2pt)

Esercizio 5 – Reti sequenziali sincronizzate (*5 punti*)

Si consideri una rete sequenziale sincronizzata di Moore rappresentata dal seguente diagramma di stato:



Si chiede di:

1) disegnare il modello strutturale in cui la sottorete sequenziale sia composta da flip-flop S-R sincronizzato; (1pt)
2) riferendosi al modello strutturale considerato al punto precedente, eseguire la sintesi algebrica e circuitale. (4pt)

Prima prova in itinere di Calcolatori Elettronici – 05.12.2008

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

A.A. 2008-2009

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 (10 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 g = 0;

int getOffset() {
    return 2^5;
}
int lowercodes(char c) {
    int t;

    t = (int) (c + getOffset());
    return t;
}
```

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

    i = 0;
    while (i < 10) {
        vett[i] = lowercodes('A' + i);
        g = g + vett[i] - 97;
        i++;
    }
    return g;
}
```

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. dichiara nel segmento dati un vettore avente nome simbolico VETT avente 10 elementi di ampiezza un byte; (1pt)
2. per 10 volte:
    a. chiede all'utente di digitare da tastiera un numero da tastiera compreso tra 40 e 99 e lo converte in binario; (2pt)
    b. memorizza tale numero in un elemento di VETT; (1pt)
    c. stampa a video la rappresentazione esadecimale, ottale e binaria di tale numero; (3pt)
    d. quindi si posiziona sulla linea successiva; (1pt)
3. terminata l'immissione degli elementi del vettore, *successivamente*:
    a. calcola la media dei numeri inseriti e la memorizza la parte intera nella locazione avente nome simbolico MI e la parte frazionaria nella locazione avente nome simbolico MD; (4pt)
    b. stampa a video tale media come un numero decimale razionale avente due cifre decimali (esempio 12,34). (3pt)

Esercizio 3 – Questionario Architettura x86 (8 punti)

Rispondere alle seguenti domande in modo coinciso ed esauriente.

1. Descrivere il modello architetturale memoria-memoria illustrandone vantaggi e svantaggi e fornendo un piccolo esempio in linguaggio Assembler. (1pt)
2. Cosa contiene il registro IP di un processore ? (1pt)
3. Descrivere il meccanismo di calcolo dell'indirizzo effettivo del processore i8086. (1pt)
4. Descrivere e fornire un esempio di indirizzamento indiretto attraverso un registro e indiciato per il processore i8086. (1pt)
5. Si descriva l'interfaccia parallela di ingresso con *handshake* indicando:
    a. la *temporizzazione* durante un ciclo di trasferimento; (1pt)
    b. lo schema di principio per implementarla con la piedinatura; (1pt)
    c. la *tabella di verità della rete combinatoria* in esso contenuta; (1pt)
6. Descrivere il ciclo di lettura del bus asincrono (diagramma temporale e commento). (1pt)

Esame di Calcolatori Elettronici – 12.12.2008 – Prova scritta

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

A.A. 2008-2009

Cognome		Nome		Matricola								

Istruzioni: Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova, ad eccezione dell'elenco di istruzioni e degli interrupt del processore 8086.

Esercizio 2 – Reti combinatorie e progettazione circuitale (5 *punti*)

Avendo a disposizione alcuni Flip-Flop di tipo D-positive-edge-triggered, alcuni multiplexer e le necessarie porte combinatorie realizzare un progetto in stile funzionale di un registro parallelo a 4 bit (2pt) con scorrimento a sinistra (3pt).

Esercizio 4 – Reti sequenziali asincrone (*8 punti*)

Si consideri una *rete sequenziale asincrona* dotata di due ingressi x e r e una uscita z, funzionante nel modo seguente:

- se r = 1, il valore di z vale 1 dopo che l'ingresso ha assunto i valori 1,0,1;
- se r = 0, il valore di z vale 0 indipendentemente dal valore presente sull'ingresso x.

Per tale rete, pilotata in modo *fondamentale* e *senza transizione multiple in ingresso*, si richiede di:

1) disegnare il diagramma di flusso; (2pt)
2) indicare il modello strutturale a cui fare riferimento per la sintesi; (1pt)
3) eseguire la sintesi algebrica e circuitale; (3pt)
4) discutere il comportamento della rete in caso di transizioni multiple in ingresso. (2pt)

Esercizio 5 – Reti sequenziali sincronizzate (5 *punti*)

Si consideri una rete sequenziale sincronizzata di Mealy ritardato che riconosca le sequenze di ingresso 1,0,0,1 *interallacciate.* Si chiede di:

1) disegnare il diagramma di flusso; (2pt)
2) disegnare il modello strutturale; (1pt)
3) riferendosi al modello strutturale considerato al punto precedente, eseguire la sintesi algebrica e circuitale. (2pt)

Esercizio 7 – 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. dichiara nel segmento dati un vettore avente nome simbolico NUMS avente 10 elementi di ampiezza un byte; (1pt)
2. per 5 volte: (1pt)
    a. chiede all'utente di digitare da tastiera un numero esadecimale codificato ASCII compreso tra 00 e FF e lo converte in binario; (3pt)
    b. memorizza tale numero in un elemento di NUMS; (1pt)
    c. stampa a video la rappresentazione decimale codificata ASCII di tale numero; (2pt)
    d. quindi si posiziona sulla linea successiva; (1pt)
3. terminata l'immissione degli elementi del vettore, *successivamente*:
    a. calcola la somma dei numeri inseriti e ne stampa la rappresentazione esadecimale codificata ASCII; (3pt)
    b. chiede all'utente se desidera ripetere l'intero procedimento: se l'utente risponde affermativamente riprende dal punto 1 altrimenti il programma termina. (3pt)

Recupero Iª prova in itinere di Calcolatori Elettronici – 12.12.2008

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

A.A. 2008-2009

Cognome		Nome		Matricola									

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

Esercizio 1 – Rappresentazione dell'informazione (6 punti)

Si considerino i seguenti numeri $X_{10} = -11$ e $Y_{10} = 3.75$, *mostrando tutti i passaggi necessari*, calcolare:

1) la rappresentazione in **complemento a 2 di X_{10}**, sia X_{C2}, e in **base due di Y_{10}**, sia Y_2; (1pt)
2) la somma binaria $\mathbf{S_2 = X_2 + Y_2}$ dei due numeri binari ottenuti al punto 1); (1pt)
3) la rappresentazione in **base 10 di S_2**; (1pt)
4) la rappresentazione secondo lo standard **IEEE/ANSI 754** dei numeri *floating point* su 4 byte di $\mathbf{S_2}$; (2pt)
5) la rappresentazione **esadecimale** del numero calcolato al punto precedente. (1pt)

Esercizio 2 – Reti combinatorie e progettazione circuitale (*8 punti*)

Avendo a disposizione alcuni Flip-Flop di tipo D-positive-edge-triggered, alcuni multiplexer e le necessarie porte combinatorie realizzare un progetto in stile funzionale di un registro parallelo a 4 bit (2pt) con scorrimento a sinistra (3pt) e funzione di caricamento parallelo (3pt).

Esercizio 3 – Memorie (6 *punti*)

Avendo a disposizione dei chip di memoria RAM del tipo 256 MByte x 8 bit, sintetizzare una memoria di tipo 1024 MByte 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'acceso al byte, 2 pt per l'accesso ai 16 bit, 1pt per l'accesso ai 32 bit, 1pt per l'espansione indirizzi)

Esercizio 4 – Reti sequenziali asincrone (*8 punti*)

Si consideri una *rete sequenziale asincrona* dotata di due ingressi x e r e una uscita z, funzionante nel modo seguente:

- se r = 1, il valore di z vale 1 dopo che l'ingresso ha assunto i valori 1,0,1;
- se r = 0, il valore di z vale 0 indipendentemente dal valore presente sull'ingresso x.

Per tale rete, pilotata in modo *fondamentale* e *senza transizione multiple in ingresso*, si richiede di:

1) disegnare il diagramma di flusso; (2pt)
2) indicare il modello strutturale a cui fare riferimento per la sintesi; (1pt)
3) eseguire la sintesi algebrica e circuitale; (3pt)
4) discutere il comportamento della rete in caso di transizioni multiple in ingresso. (2pt)

Esercizio 5 – Reti sequenziali sincronizzate (5 *punti*)

Si consideri una rete sequenziale sincronizzata di Mealy ritardato che riconosca le sequenze di ingresso 1,0,0,1 *interallacciate.*

Si chiede di:

1) disegnare il diagramma di flusso; (2pt)
2) disegnare il modello strutturale; (1pt)
3) riferendosi al modello strutturale considerato al punto precedente, eseguire la sintesi algebrica e circuitale. (2pt)

Recupero IIª prova in itinere di Calcolatori Elettronici – 12.12.2008

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

A.A. 2008-2009

Cognome		Nome		Matricola									

Istruzioni: Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova, ad eccezione dell'elenco di istruzioni e degli interrupt del processore 8086.

Esercizio 6 – Traduzione da linguaggio ad alto livello ad Assembler (10 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 g = 0;
char c;

char alfa(int a) {
    char m;

    m = 'a' + (char) a;
    return m;
}

int beta(int b) {
    int n;

    n = b * 2;
    return n;
}
```

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

    i = 0;
    j = 2;
    while ((i+j) < 5) {
        c = alfa(beta(i));
        if (c >= 'a' || c <= 'z') {
            g++;
        }
    i++;
    }
    return g;
}
```

Esercizio 7 – 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. dichiara nel segmento dati un vettore avente nome simbolico NUMS avente 10 elementi di ampiezza un byte; (1pt)
2. per 5 volte: (1pt)
    a. chiede all'utente di digitare da tastiera un numero esadecimale codificato ASCII compreso tra 00 e FF e lo converte in binario; (3pt)
    b. memorizza tale numero in un elemento di NUMS; (1pt)
    c. stampa a video la rappresentazione decimale codificata ASCII di tale numero; (2pt)
    d. quindi si posiziona sulla linea successiva; (1pt)
3. terminata l'immissione degli elementi del vettore, *successivamente*:
    a. calcola la somma dei numeri inseriti e ne stampa la rappresentazione esadecimale codificata ASCII; (3pt)
    b. chiede all'utente se desidera ripetere l'intero procedimento: se l'utente risponde affermativamente riprende dal punto 1 altrimenti il programma termina. (3pt)

Esercizio 8 – Questionario Architettura x86 (8 punti)

Rispondere alle seguenti domande in modo coinciso ed esauriente.

1. Cosa è l'operazione di fetch ? (1pt)
2. Descrivere l'arbitro elementare asincrono, la tabella di verità, la sua sintesi, e il montaggio quando sono presenti due unità master. (4pt)
3. Descrivere il processo di compilazione di un programma ad alto livello composto da più moduli. (1pt)
4. Quali sono le principali tecniche per aumentare la banda di un bus, e quali sono i loro limiti. (2pt)

```
.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 – 14.01.2008 – Prova scritta

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

A.A. 2008-2009

Cognome		Nome		Matricola								

Istruzioni: Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova, ad eccezione dell'elenco di istruzioni e degli interrupt del processore 8086.

Esercizio 2 – Reti combinatorie e progettazione circuitale (5 *punti*)

Avendo a disposizione un multiplexer 4 → 1 realizzare la funzione z = f(A,B,C) indicata nella seguente tabella.

C\A,B	**00**	**01**	**11**	**10**
0	-	0	1	1
1	1	1	-	0

Esercizio 4 – Reti sequenziali asincrone (*8 punti*)

Realizzare un circuito *sequenziale asincrono* che partendo da una situazione di stabilità con x=0 e z=0 faccia quanto segue: sul piedino **z = MSB(|∑ valori ricevuti sul piedino x|$_4$)** dove l'operatore MSB restituisce il bit più significativo dell'espressione. Nota: si usi un *cortocircuito* per la sottorete sequenziale asincrona.

Esercizio 5 – Reti sequenziali sincronizzate (5 *punti*)

Realizzare un circuito sequenziale sincronizzato con Flip-Flop D+edge-triggered che riconosca le sequenze le sequenze **0**,**1**,**1**,**0** interallacciate.

Esercizio 7 – 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. dichiara nel segmento dati un vettore avente nome simbolico VETT avente 5 elementi di ampiezza di 16 bit ed un vettore avente nome simbolico COPPIE avente 5 elementi ampi un byte; (1pt)
2. per 5 volte: (1pt)
    a. chiede all'utente di digitare da tastiera un numero decimale codificato ASCII compreso tra 00 e 99 e lo converte in binario; (2pt)
    b. memorizza tale numero in un elemento di VETT; (1pt)
    c. conta quante coppie di bit '01' sono presenti in tale numero e lascia il risultato nell'elemento di indice corrispondente del vettore COPPIE; (2pt)
    d. stampa a video la rappresentazione binaria dell'elemento memorizzato in VETT1 ed il numero di coppie contate; (2pt)
    e. quindi si posiziona sulla linea successiva; (1pt)
3. terminata l'immissione degli elementi del vettore, *successivamente*:
    a. calcola la somma dei numeri inseriti e ne stampa la rappresentazione esadecimale codificata ASCII; (3pt)
    b. chiede all'utente se desidera ripetere l'intero procedimento: se l'utente risponde affermativamente riprende dal punto 2 altrimenti il programma termina. (2pt)

```
.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
```

Recupero Iª prova in itinere di Calcolatori Elettronici – 14.01.2009

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

A.A. 2008-2009

Cognome		Nome		Matricola									

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

Esercizio 1 – Rappresentazione dell'informazione (6 punti)

Si considerino i seguenti numeri $X_{10} = -5$ e $Y_{10} = 2.25$, *mostrando tutti i passaggi necessari*, calcolare:

1) la rappresentazione in **complemento a 2 di X_{10}**, sia X_{C2}, e in **base due di Y_{10}**, sia Y_2; (1pt)
2) la somma binaria S_2**= X_2+ Y_2** dei due numeri binari ottenuti al punto 1); (1pt)
3) la rappresentazione in **base 10 di S_2**; (1pt)
4) la rappresentazione secondo lo standard **IEEE/ANSI 754** dei numeri *floating point* su 4 byte di **S_2**; (2pt)
5) la rappresentazione **esadecimale** del numero calcolato al punto precedente. (1pt)

Nota. X_{C2} è rappresentato su 8 bit, Y_2 e S_2 sono rappresentati su 8 bit per la parte intera e 8 bit per la parte decimale.

Esercizio 2 – Reti combinatorie e progettazione circuitale (*8 punti*)

Avendo a disposizione un multiplexer 4 → 1 realizzare la funzione z = f(A,B,C) indicata nella seguente tabella.

C\A,B	**00**	**01**	**11**	**10**
0	-	0	1	1
1	1	1	-	0

Esercizio 3 – Memorie (6 *punti*)

Avendo a disposizione dei chip di memoria RAM del tipo 128 MByte x 8 bit, sintetizzare una memoria di tipo 512 MByte 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'acceso al byte, 2 pt per l'accesso ai 16 bit, 1pt per l'accesso ai 32 bit, 1pt per l'espansione indirizzi)

Esercizio 4 – Reti sequenziali asincrone (*8 punti*)

Realizzare un circuito *sequenziale asincrono* che partendo da una situazione di stabilità con x=0 e z=0 faccia quanto segue: sul piedino **z = MSB(|∑ valori ricevuti sul piedino x|$_4$)** dove l'operatore MSB restituisce il bit più significativo dell'espressione. Nota: si usi un *cortocircuito* per la sottorete sequenziale asincrona.

Esercizio 5 – Reti sequenziali sincronizzate (5 *punti*)

Realizzare un circuito sequenziale sincronizzato con Flip-Flop D+edge-triggered che riconosca le sequenze le sequenze **0**,**1**,**1**,**0** interallacciate.

Recupero IIª prova in itinere di Calcolatori Elettronici – 14.01.2009

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

A.A. 2008-2009

Cognome		Nome		Matricola									

Istruzioni: Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova, ad eccezione dell'elenco di istruzioni e degli interrupt del processore 8086.

Esercizio 6 – Traduzione da linguaggio ad alto livello ad Assembler (10 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 g = 5;
char c1 = 'a';

char alfa(char k) {

    return k – 'a' + '0';
}

int beta(int n) {
    int m;

    m = n + alfa(n);
    return m;
}
```

```
int main()
{
    int i;

    for (i = 0; i < 5; i++) {
        c1 = c1 + i;
        g = alfa(c1);
        if (g < 10)
            g = beta(g);
    }

    return g;
}
```

Esercizio 7 – 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. dichiara nel segmento dati un vettore avente nome simbolico VETT avente 5 elementi di ampiezza di 16 bit ed un vettore avente nome simbolico COPPIE avente 5 elementi ampi un byte; (1pt)
2. per 5 volte: (1pt)
    a. chiede all'utente di digitare da tastiera un numero decimale codificato ASCII compreso tra 00 e 99 e lo converte in binario; (2pt)
    b. memorizza tale numero in un elemento di VETT; (1pt)
    c. conta quante coppie di bit '01' sono presenti in tale numero e lascia il risultato nell'elemento di indice corrispondente del vettore COPPIE; (2pt)
    d. stampa a video la rappresentazione binaria dell'elemento memorizzato in VETT1 ed il numero di coppie contate; (2pt)
    e. quindi si posiziona sulla linea successiva; (1pt)
3. terminata l'immissione degli elementi del vettore, *successivamente*:
    a. calcola la somma dei numeri inseriti e ne stampa la rappresentazione esadecimale codificata ASCII; (3pt)
    b. chiede all'utente se desidera ripetere l'intero procedimento: se l'utente risponde affermativamente riprende dal punto 2 altrimenti il programma termina. (2pt)

Esercizio 8 – Questionario Architettura x86 (8 punti)

Rispondere alle seguenti domande in modo coinciso ed esauriente.

1. Quale relazione intercorre tra le istruzioni in linguaggio macchina ed il linguaggio Assembler ? (1pt)
2. Descrivere il *flag* di *carry* (CF) del registro di stato e fornire un esempio di utilizzo. (1pt)
3. Descrivere l'arbitro a priorità rotante. (1pt)
4. Descrivere l'interfaccia parallela di uscita con *handshake*, con diagramma funzionale, RC e RSA (3pt)
5. Descrivere i concetti di "prologo" ed "epilogo" nella programmazione a moduli. (2pt)

```
.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 – 04.06.2009 – Prova scritta

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

A.A. 2008-2009

Cognome		Nome		Matricola									

Istruzioni: Non è ammesso l'utilizzo di materiale didattico o appunti durante questa prova, ad eccezione dell'elenco di istruzioni e degli interrupt del processore 8086. *Scrivere in modo chiaro su fogli A4.*

Esercizio 1 – Rappresentazione dell'informazione

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

1) la rappresentazione in complemento a 2 su 8 bit di X_{10} e di Y_{10} (siano $\mathbf{X_{C2}}$ e $\mathbf{Y_{C2}}$) ;
2) la somma su 8 bit $\mathbf{S_{C2} = X_{C2} + Y_{C2}}$ dei due numeri binari ottenuti al punto precedente;
3) la divisione $\mathbf{D_{C2} = X_{C2} / S_{C2}}$ ed il resto $\mathbf{R_{C2}}$, 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.

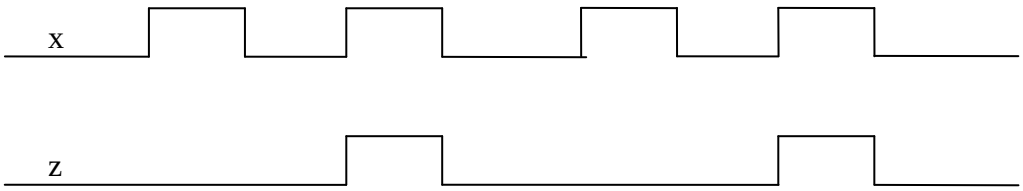
Esercizio 2 – Reti sequenziali asincrone

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 3 – Programmazione Assembler

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 digitare da tastiera 3 cifre (ad. es. 321), ne fa l'eco a video e ne memorizza il corrispondente valore binario nella locazione ampia 16 bit avente nome simbolico `CONTATORE`;
2) si posiziona su una nuova linea e chiede all'utente di digitare 1 cifra (ad es. 5), ne fa l'eco a video e ne memorizza il corrispondente valore binario nella locazione ampia 8 bit avente nome simbolico `PASSO`;
3) inizia un ciclo che ad ogni passo esegue le seguenti operazioni:
    a) si posiziona su una nuova linea e stampa a schermo il valore di `CONTATORE` in esadecimale;
    b) decrementa `CONTATORE` di una quantità uguale al valore contenuto nella variabile `PASSO`;
    c) se `CONTATORE` non è inferiore o uguale a zero torna al punto a) altrimenti il programma termina.

Domande orali *(scrivere la risposta in modo chiaro, preciso, coinciso ed esauriente)*

1) Si consideri la programmazione a moduli nel linguaggio Assembler 8086: descrivere quali **operazioni** sono effettuate dal **chiamante** di una procedura.
2) Si descriva **l'arbitro asincrono elementare**, indicando la tabella di flusso e la sintesi per realizzarlo.
3) Si descrivano le caratteristiche e la *temporizzazione* del **ciclo di scrittura** del **bus asincrono** e si evidenzino le differenze rispetto a quello sincrono.

```
.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
```