

Prima Prova in Itinere di Calcolatori Elettronici – 02.11.2006

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

A.A. 2006-2007

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

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

Esercizio 1

Si considerino due numeri in base 10, $X_{10} = 11$ e $Y_{10} = 19$; calcolare:

- 1) le rappresentazioni in base 2 di X_{10} e Y_{10} espresse su 8 bit (X_2 e Y_2);
- 2) la somma A_2 dei due numeri binari ottenuti al punto 1);
- 3) il prodotto P_2 dei due numeri binari ottenuti al punto 1);
- 4) la rappresentazione in complemento a 2 del numero Y_2 ;
- 5) la sottrazione S_2 in binario tra P_2 ed il numero Y_2 ;
- 6) la rappresentazione in base 10 di P_2 e S_2 (P_{10} e S_{10}).

Si consideri quindi il numero razionale decimale formato dalle cifre " P_{10}, S_{10} ", definito come F_{10} , si calcoli:

- 7) la rappresentazione secondo lo standard IEEE/ANSI 754 dei numeri *floating point* su 4 byte;
- 8) la rappresentazione esadecimale del numero calcolato al punto 7).

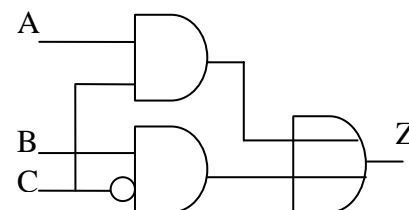
Esercizio 2

Si consideri il *circuito combinatorio* mostrato in figura:

- 1) Si controlli se tale circuito ha alee statiche del primo ordine sul livello 1; in caso positivo, descrivere come possono essere rimosse, indicando anche la forma SP priva di alee.

Si consideri quindi un secondo circuito descritto dalla seguente tabella di verità:

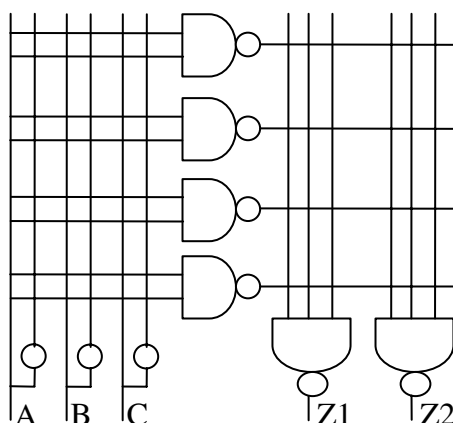
AB\C	0	1
00	1	-
01	1	-
11	1	-
10	0	0



- 2) Si calcoli la forma canonica SP, indicando se ci sono alee, ed in tal caso si provveda a rimuoverle.

Siano Z_1 e Z_2 le forme SP ricavate dal punto 1) e dal punto 2) rispettivamente:

- 3) Si programmi la seguente PLA:



Esercizio 3

Si consideri una *rete sequenziale asincrona* dotata di due ingressi x e r ed una uscita z , il cui comportamento è specificato come segue:

- se $r=0$, $z=0$ indipendentemente dal valore presente sul piedino i .
- se $r=1$, il valore di z vale 1 dopo che sul piedino i è stata riconosciuta la sequenza 1,0,1 (non interallacciata).

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

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

Esercizio 4

Si vuole descrivere il comportamento di un semaforo con quattro lampade tramite una *rete sequenziale sincronizzata di Moore*. Il semaforo commuta periodicamente il suo stato come indicato in figura:

R		
		G
	V	

La rete che descrive il semaforo ha in ingresso, oltre al segnale di clock, un piedino x che:

- quando è posto a 1 abilita il funzionamento del semaforo nel modo indicato;
- quando è posto a 0 spegne il semaforo.

Questa rete ha tre piedini di uscita, z_R , z_G e z_V , ognuno del quale quando vale 1 accende una la corrispondente lampada del semaforo.

Si scelga la codifica degli stati in modo che questi siano adiacenti.

Si chiede di:

- 1) disegnare il diagramma di flusso;
- 2) disegnare il modello strutturale in cui la sottorete SN è composta da un flip-flop D-positive-edge-triggered (per la variabile di stato più significativa, y_1) e da un flip-flop SR sincronizzato (per la variabile di stato meno significativa, y_0);
- 3) eseguire la sintesi algebrica e circuitale.

Esercizio 5

Si consideri una *rete sequenziale sincronizzata di Mealy ritardato* avente due ingressi x_1 e x_2 ed una uscita z . L'uscita z vale 1 quando la rete riconosce sui piedini di ingresso la sequenza 00,01,10,11.

Si chiede di:

- 1) disegnare il diagramma di flusso;
- 2) disegnare il modello strutturale a cui fare riferimento per la sintesi;
- 3) eseguire la sintesi algebrica e circuitale.

Seconda Prova in Itinere di Calcolatori Elettronici – 01.12.2006

Corso di Laurea di Ing. Gestionale e di Ing. delle Telecomunicazioni

A.A. 2005-2006

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 1

Dato il seguente listato scritto in linguaggio C 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.

Nota: i numeri interi sono rappresentati su 2 byte, i caratteri su 1 byte

```

int g;

int
beta(int x)
{
    return x - 1;
}

int
alpha(int a, char b)
{
    char c;
    int d;

    c = b - 'a';
    if ((c % 2) == 0)
        d = a + 1;
    else
        d = beta(a);
    return d;
}

int
main()
{
    int i, j;
    char k;

    k = 't';
    for (i = 10; i > 0; i--) {
        j = alpha(i, k);
        g = g + j;
    }
    return g;
}

```

Esercizio 2

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

1. stampa sullo schermo 10 volte il carattere '*' nelle posizioni di coordinate definite nel vettore VETT;
2. attende che l'utente prema uno dei seguenti tasti: 'a', 'd', 's', 'w', e 'z': se il tasto premuto corrisponde al carattere 'z' il programma termina;
3. a seconda del tasto premuto sposta il cursore di una posizione rispetto a quella attuale in direzione:
 - tasto 'a': sinistra,
 - tasto 'd': destra,
 - tasto 's': basso,
 - tasto 'w': alto,
4. controlla la nuova posizione del cursore sia posta all'interno dello schermo (dimensioni 80x25);
5. controlla che nella nuova posizione sia già presente il carattere '*':
 - a. se è presente passa al punto 6,
 - b. altrimenti scrive il carattere '*', quindi torna al punto 2;
6. stampi la stringa "Game Over!", quindi stampi su linee successive dello schermo il numero di spostamenti del cursore:
 - a. verso sinistra,
 - b. verso destra,
 - c. verso il basso,
 - d. verso l'altro;
7. invii sull'interfaccia di uscita, disponibile agli indirizzi RSR e RBR, i caratteri ASCII necessari a stampare la somma totale di tali spostamenti.

Esercizio 3

Si descriva l'**interfaccia parallela di ingresso con handshake** indicando:

- a) la *temporizzazione* durante un ciclo di trasferimento;
- b) lo schema di principio per implementarla con la piedinatura;
- c) la *tabella di verità della rete combinatoria* in esso contenuta;
- d) il *diagramma di stato della rete sequenziale asincrona* che implementa il protocollo di handshake e pilota le linee di comando in uscita.

Seconda Prova in Itinere di Calcolatori Elettronici – 01.12.2006

Corso di Laurea di Ing. Gestionale e di Ing. delle Telecomunicazioni

A.A. 2005-2006

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

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.

Istruzioni per coloro che recuperano una prova in itinere:

1. chi recupera la prima prova deve svolgere gli esercizi 1, 2 e 3;
2. chi recupera la seconda prova deve svolgere gli esercizi 4, 5, e 6.

Istruzioni per coloro che sostengono l'appello di esame ordinario:

1. gli esercizi da svolgere sono 2, 3, 4 e 5;
2. è consentito l'utilizzo di materiale didattico, ma non di appunti.

Esercizio 1

Avendo a disposizione dei chip di memoria RAM del tipo 256 byte x 4, sintetizzare una memoria di tipo 1Kbyte x 16. Disegnare il montaggio commentando la soluzione scelta.

Esercizio 2

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 $\neg x_2$.

Si discuta il caso delle transizioni multiple in ingresso.

Esercizio 3

Realizzare un circuito sequenziale sincronizzato di Moore con Flip-Flop S-R che riconosca la sequenza **0,1,1,0** interallacciata.

Esercizio 4

Dato il seguente listato scritto in linguaggio C 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.

Nota: i numeri interi sono rappresentati su 2 byte.

```
int g;

int uno(int x)
{
    int r;

    r = x + 1;
    return r;
}

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

    d = a / b;
    if (d > 2)
        d--;

    else
        d++;

    return d;
}

int main()
{
    int i, j;

    j = 2;
    i = uno(j);
    g = due(i, j);
    return g;
}
```

Esercizio 5

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

1. stampa carattere '*' nella posizione di coordinate definite nei byte aventi nome simbolico STARTCOL e STARTROW;
2. attende che l'utente prema il tasto 'a' (senza farne l'eco);
3. cancella il carattere "*" dalla posizione precedente e lo stampa nella una posizione immediatamente più a destra nello schermo;
4. controlla se il carattere "*" appena spostato ha raggiunto il bordo destro dello schermo (colonna 79), in tal caso cancella il carattere e lo stampa nuovamente nella colonna 0 della stessa riga e continua dal punto 3;
5. ad ogni spostamento del carattere "*" stampa il numero di colonna della posizione corrente nell'angolo in alto a sinistra dello schermo (si tenga presente che questo valore varia tra 0 e 79 e pertanto sono necessari due caratteri ASCII).

Esercizio 6

Si descriva il record di attivazione nella traduzione da linguaggio C a linguaggio Assembler.

Esame scritto di Calcolatori Elettronici – 12.01.20067

Corso di Laurea di Ing. Gestionale e di Ing. delle Telecomunicazioni

A.A. 2006-2007

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

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.

Istruzioni per coloro che recuperano una prova in itinere:

1. chi recupera la prima prova deve svolgere gli esercizi 1, 2 e 3;
2. chi recupera la seconda prova deve svolgere gli esercizi 4, 5, e 6

Istruzioni per coloro che sostengono l'appello di esame ordinario:

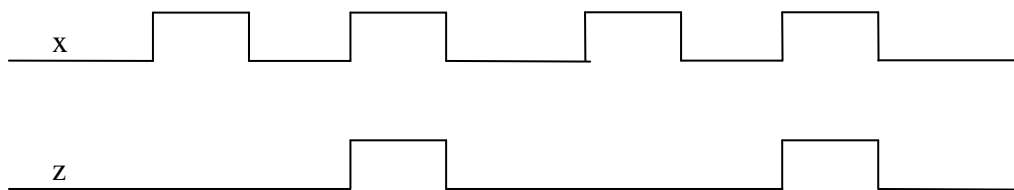
1. gli esercizi da svolgere sono 1, 2, 4 e 5;
2. è consentito l'utilizzo di materiale didattico, ma non di appunti

Esercizio 1

Si consideri una Rete Sequenziale Asincrona avente un ingresso x ed una uscita z , che inizialmente 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:

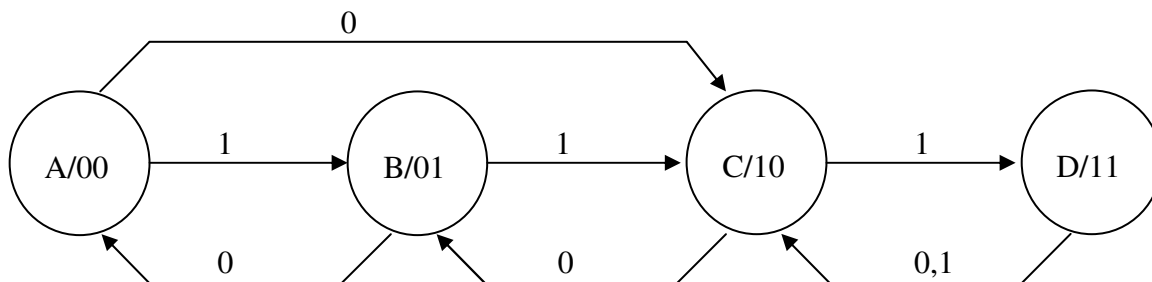


Per tale rete:

- a) si disegni il corrispondente diagramma di flusso, e
- b) si sintetizzi il suo circuito logico.

Esercizio 2

Realizzare tramite una rete sequenziale sincronizzata di Moore un circuito che ha un piedino di ingresso x e due piedini di uscita $z1$ e $z2$, e che ha implementi il seguente diagramma di stato:



Si scelga la codifica degli stati in modo che questi siano adiacenti.

Si chiede di:

- 1) disegnare il modello strutturale in cui la sottorete SN è composta da flip-flop J-K;
- 2) eseguire la sintesi algebrica e circuitale.

Esercizio 3

Si descriva il Flip-Flop D-Latch evidenziandone:

- a) tabella di flusso,
- b) diagramma di temporizzazione e
- c) la sintesi per mezzo di un Flip-Flop S-R.

Esercizio 4

Dato il seguente listato scritto in linguaggio C 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.

Nota: i numeri interi sono rappresentati su 2 byte, i caratteri su 1 byte

```
int z=0;

int
mygetc(int x)
{
    int r;

    r = 65 + x;
    return r;
}

int
calc(char a, char b, char c)
{
    int q;

    q = a + b + c;
    return q;
}
int
```

```
main()
{
    int i, j, k;
    char c1, c2, c3;

    for (i = 0; i < 26; i++) {
        j = 26 - i;
        c1 = mygetc(i);
        c2 = mygetc(j);
        if (c1 > c2)
            c3 = c1 - c2;
        else
            c3 = c2 - c1;
        k = calc(c3, c2, c1);
        z = z + k;
    }
    return z;
}
```

Esercizio 2

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

1. attenda che l'utente digiti un numero compreso tra 3 e 9.
2. memorizzi il numero digitato nella cella ampia un byte di nome simbolico N;
3. calcoli il valore della successione di Fibonacci per il numero N, memorizzando tale valore nella cella di nome R;
4. stampi su una nuova riga il valore decimale (codificato da cifre ASCII) del numero R;
5. invii le cifre ASCII del risultato appena stampato anche all'interfaccia parallela di uscita collegata agli indirizzi simbolici TSR e TBR.

Nota. La successione di Fibonacci è una sequenza di numeri interi naturali definibile assegnando i valori dei due primi termini, $F_0 := 0$ ed $F_1 := 1$, e chiedendo che per ogni successivo sia $F_n := F_{n-1} + F_{n-2}$. Essa produce i seguenti valori: 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Esercizio 3

Si descriva l'*arbitro a priorità rotante* a cui possono essere connessi fino a 4 dispositivi master, mostrando come questo arbitro conceda il *grant* ai dispositivi collegati quando sono attive tutte le linee di *request*.

Esame scritto di Calcolatori Elettronici – 29.03.2007

Corso di Laurea di Ing. Gestionale e di Ing. delle Telecomunicazioni

A.A. 2006-2007

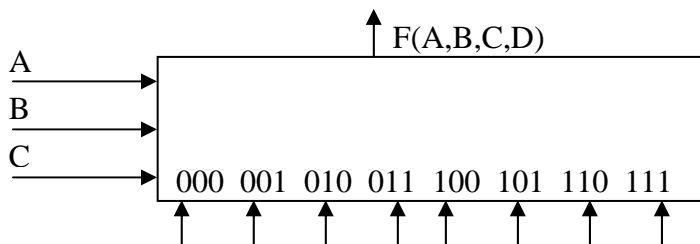
Cognome		Nome		Matricola										
---------	--	------	--	-----------	--	--	--	--	--	--	--	--	--	--

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 1

Sintetizzare la funzione di 4 variabili non completamente specificate, rappresentata nella tabella, mediante un commutatore 8 ad 1, le cui variabili di selezione siano pilotate dalle variabili A, B, e C rispettivamente (vedi figura).

	AB	00	01	11	10
CD					
00		0	1	0	-
01		0	0	-	0
11		1	1	-	0
10		0	0	1	-



Esercizio 2

Realizzare tramite una rete sequenziale sincronizzata di Moore con flip-flop J-K che riconosca la sequenza 101. Si scelga la codifica degli stati in modo che questi siano adiacenti. Si chiede di:

- 1) disegnare il diagramma di stato;
- 2) disegnare il modello strutturale;
- 3) eseguire la sintesi algebrica e circuitale.

Esercizio 3

Dato il seguente listato scritto in linguaggio C 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.

Nota: i numeri interi sono rappresentati su 2 byte, i caratteri su 1 byte

```
int g=0;

int
test(char c)
{
    int m;

    if (c < 'a')
        return 0;
    else
        m = c%4;
    return m;
}
```

```
int
main()
{
    int i,r;
    char c;

    for (i = 0; i < 26; i++) {
        c = i + 'a';
        r = test(c);
        if (r == 0)
            g = g + i;
    }
    return g;
}
```

Esercizio 4

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

1. conti le triple 010 presente nella locazione ampia 16 bit di nome simbolico ALFA;
2. memorizzi tale numero di triple nella locazione ampia un byte di nome simbolico T;
3. memorizzi il quadrato di T nella locazione ampia un byte di nome simbolico Q;
4. stampi a video il valore decimale (codificato da cifre ASCII) del numero Q.

Esame scritto di Calcolatori Elettronici – 19.04.2007

Corso di Laurea di Ing. Gestionale e di Ing. delle Telecomunicazioni

A.A. 2006-2007

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

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 1

Illustrare graficamente e testualmente il montaggio di una memoria 1Kx16bit avendo a disposizione due banchi di memoria da 512x16bit, accessibile al byte e alla word. Si riportino le tabelle di verità per le eventuali rete combinatorie utilizzate per la realizzazione di tale memoria.

Esercizio 2

Realizzare tramite una *rete sequenziale asincrona* che riconosca le sequenze 010 interallacciate. Si scelga la codifica degli stati in modo che questi siano adiacenti. Si chiede di:

- 1) disegnare il diagramma di stato;
- 2) disegnare il modello strutturale;
- 3) eseguire la sintesi algebrica e circuitale.

Esercizio 3

Dato il seguente listato scritto in linguaggio C 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.

Nota: i numeri interi sono rappresentati su 2 byte, i caratteri su 1 byte

```
int g=0;

int
test(char c)
{
    int m;

    if (c < 'a')
        return 0;
    else
        m = c%4;
    return m;
}
```

```
int
main()
{
    int i,r;
    char c;

    for (i = 0; i < 26; i++) {
        c = i + 'a';
        r = test(c);
        if (r == 0)
            g = g + i;
    }
    return g;
}
```

Esercizio 4

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

1. conta il numero di bit posti uguale a '1' della locazione ampia 8 bit di nome simbolico ALFA;
2. memorizzi tale numero nella locazione ampia un byte di nome simbolico T;
3. se tale numero è pari salta al punto 4, altrimenti sostituisce ogni bit a '0' contenuto in *alfa* con un bit a '1';
4. stampi a video il valore decimale (codificato da cifre ASCII) del numero T.

Esame scritto di Calcolatori Elettronici – 06.09.2007

Corso di Laurea di Ing. Gestionale e di Ing. delle Telecomunicazioni

A.A. 2006-2007

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

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 1

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 2

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

Esercizio 3

Dato il seguente listato scritto in linguaggio C 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.

Nota: i numeri interi sono rappresentati su 2 byte, i caratteri su 1 byte

```
int g=9;
char c = '2';

int
hacker(char alfa)
{
    int beta, gamma;

    gamma = 20;
    beta = alfa + gamma / 2 + g - 2;

    return beta;
}
```

```
int
main()
{
    int a,b;
    char d;

    a=10;
    b=11;
    if hacker(a + b) > g
        d = c;
    else
        c = 'k';
    return 0;
}
```

Esercizio 4

Si considerino tre vettori di nome simbolico VETT1, VETT2, e VETT3 ognuno composto da 32 elementi da 8 bit. Scrivere un programma in linguaggio **Assembler 8086** che esegua quanto segue:

1. per ogni elemento di VETT1 e VETT2 esegue la somma e memorizza il risultato nel vettore VETT3;
2. dopo ogni somma stampa a schermo la rappresentazione ASCII del numeri estratti da VETT1, VETT2 e VETT3.

Esame scritto di Calcolatori Elettronici – 26.09.2007

Corso di Laurea di Ing. Gestionale e di Ing. delle Telecomunicazioni

A.A. 2006-2007

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

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 1

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 2

Realizzare un circuito sequenziale sincronizzato con Flip-Flop J-K che riconosca le sequenze le sequenze **0,1,0** interallacciate.

Esercizio 3

Dato il seguente listato scritto in linguaggio C 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.

Nota: i numeri interi sono rappresentati su 2 byte, i caratteri su 1 byte

```
int g=9;
char c = '2';

int
hacker(char alfa)
{
    int beta, gamma;

    gamma = 20;
    beta = alfa + gamma / 2 + g - 2;

    return beta;
}
```

```
int
main()
{
    int a,b;
    char d;

    a=5;
    b=7;
    if hacker(a + b) < g
        d = c;
    else
        c = 'k';
    return 0;
}
```

Esercizio 4

Si considerino tre vettori di nome simbolico VETT1, VETT2, e VETT3 ognuno composto da 32 elementi da 8 bit. Scrivere un programma in linguaggio **Assembler 8086** che esegua quanto segue:

1. per ogni elemento di VETT1 e VETT2 esegue la somma e memorizza il risultato nel vettore VETT3;
2. dopo ogni somma stampa a schermo la rappresentazione ASCII del numeri estratti da VETT1, VETT2 e VETT3.