

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Calcolatori Elettronici I — a.a. 2007–2008

**Compito # 1 del 31 marzo 2008**

*Cognome e Nome dello studente:* \_\_\_\_\_

**A/** Gli antichi Romani non utilizzavano un sistema di numerazione posizionale, e non eseguivano le moltiplicazioni a mano con il metodo usato oggi. Il loro ingegnoso algoritmo per la moltiplicazione di due numeri positivi  $a$  e  $b$  era il seguente:

0. Scrivi  $a$  e  $b$  uno accanto all'altro.
1. Dimezza  $a$  tralasciando l'eventuale resto, e poni il risultato in colonna sotto  $a$ . Raddoppia  $b$ , e poni il risultato in colonna sotto  $b$ .
2. Ripeti il passo precedente con gli ultimi valori scritti in colonna, fintanto che il risultato della divisione è maggiore di zero.
3. Il risultato cercato è la somma dei valori nella colonna ' $b$ ' corrispondenti a valori *dispari* nella colonna ' $a$ '.

La tabella qui sotto mostra come esempio il calcolo del prodotto  $71 \times 14 = 994$ :

colonna ' $a$ '	colonna ' $b$ '	somma parziale
<b>71</b>	<b>14</b>	14
35	28	42
17	56	98
8	112	98
4	224	98
2	448	98
1	896	994
0	—	<b>994</b>

- a. Scrivere la *subroutine* assembly 8086 di tipo “far” ROMUL per il calcolo del prodotto di due numeri interi positivi con l'algoritmo sopra descritto. Lo scambio di dati col chiamante deve avvenire attraverso lo stack.
- b. Scrivere la sezione di codice del chiamante necessaria ad eseguire il prodotto di 71 e 14. Fornire una codifica di macchina plausibile per l'istruzione di chiamata a ROMUL, indicando il numero di cicli di bus richiesti per il suo fetch ed esecuzione.
- c. Descrivere il contenuto dello stack prima, durante e dopo la chiamata.

**B/** Un'interfaccia di ingresso gestisce due dispositivi, uno a controllo di programma, e l'altro sotto controllo di interruzione. L'interfaccia è mappata a partire dall'indirizzo  $5Ch$ .

- a. Disegnare l'hardware completo di interfaccia, inclusa la logica di decodifica degli indirizzi e di selezione delle porte.
- b. Fornire i diagrammi temporali delle transazioni con la CPU.

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Calcolatori Elettronici I — a.a. 2007–2008

**Compito # 2 del 31 marzo 2008**

*Cognome e Nome dello studente:*

---

**A/** Gli antichi Romani non utilizzavano un sistema di numerazione posizionale, e non eseguivano le moltiplicazioni a mano con il metodo usato oggi. Il loro ingegnoso algoritmo per la moltiplicazione di due numeri positivi  $a$  e  $b$  era il seguente:

0. Scrivi  $a$  e  $b$  uno accanto all'altro.
1. Dimezza  $a$  tralasciando l'eventuale resto, e poni il risultato in colonna sotto  $a$ . Raddoppia  $b$ , e poni il risultato in colonna sotto  $b$ .
2. Ripeti il passo precedente con gli ultimi valori scritti in colonna, fintanto che il risultato della divisione è maggiore di zero.
3. Il risultato cercato è la somma dei valori nella colonna ' $b$ ' corrispondenti a valori *dispari* nella colonna ' $a$ '.

La tabella qui sotto mostra come esempio il calcolo del prodotto  $71 \times 14 = 994$ :

colonna ' $a$ '	colonna ' $b$ '	somma parziale
<b>71</b>	<b>14</b>	14
35	28	42
17	56	98
8	112	98
4	224	98
2	448	98
1	896	994
0	—	<b>994</b>

- a. Disegnare la sezione di parte operativa e scrivere il microcodice di controllo necessari a realizzare l'istruzione di macchina ROMUL VAR, R2 — con VAR variabile di memoria, e R2 registro sia sorgente che destinazione — in una CPU a singolo bus interno.
- b. Fornire una codifica di macchina plausibile per l'istruzione di cui sopra, indicando il numero di cicli di macchina e di bus richiesti per il suo fetch ed esecuzione.

**B/** Il sistema di numerazione in uso nell'antica Roma comprendeva le cifre I, V, X, L, C, D, M, ad indicare rispettivamente 1, 5, 10, 50, 100, 500, 1000. La particolarità del sistema è che quando un numero minore è anteposto ad uno maggiore, il primo viene sottratto dal secondo (ad es., "MCMLXIV" =  $1000 - 100 + 1000 + 50 + 10 - 1 + 5 = 1964$ ).

Scrivere la *macro* assembly 8086 ROM2AX che consenta di convertire una stringa (ASCII) di memoria in cifre romane nel suo equivalente binario in AX.