

**Compito del 7 febbraio 2003**

**SCELTA DI ESERCIZI**

**0/** Indicare il valore decimale del numero binario 10010110 nell'ipotesi che esso sia espresso in rappresentazione (a) naturale, (b) modulo e segno, (c) complemento a uno, (d) complemento a due.

**1/** Definire l'operatore XOR, e spiegarne l'utilità nella realizzazione di sommatore e di circuiti di controllo di parità.

**1/** Definire le funzioni logiche di due variabili “>”, “=” e “<”, e mostrarne l'utilizzo nella realizzazione di comparatori.

**2/** Disegnare il diagramma di stato (segnalando eventuali condizioni di indifferenza sugli ingressi) per una macchina sequenziale di Moore capace di generare le seguenti sequenze di uscita: *abababababa...*, *abcabcabcab...*, *abcdabcdabc...*. Quanti stati ha la macchina? Qual è il minimo numero di flip-flop necessario per realizzarla?

**3/** Illustrare il ruolo dell'unità di controllo all'interno di un microprocessore, indicandone ingressi e uscite. Discutere in particolare i passi di controllo relativi all'esecuzione di un'istruzione di salto condizionato.

**4/** Un banco di una memoria RAM per un computer di tipo byte-addressable ha dimensione  $D = 32\text{M} \times 1$  byte. Il banco è realizzato con moduli da  $B = 512\text{K} \times 4$  byte.

1) Calcolare il numero  $n$  di moduli necessari alla costruzione del banco, il numero  $d$  di bit necessari a indirizzare il banco, e il numero  $b$  di bit di indirizzo di ciascun modulo.

2) Illustrare l'organizzazione interna del banco, ed in particolare il modo in cui i bit della parola di indirizzo pilotano i vari moduli attraverso l'opportuna logica combinatoria.

**5/** Tracciare lo schema a blocchi di un'interfaccia di ingresso/uscita, con porte mappate a partire dall'indirizzo 07FH. Il bus dati è a 8 bit. L'interfaccia deve consentire la lettura dello stato (6 bit) e la scrittura dei controlli (4 bit), e fare uso del minimo numero di porte possibile.

6/ Data l'istruzione assembly 8086 `MOV VAR[BP][SI],BH`, dove `VAR` è un vettore di byte, dire:

- 1) qual è il modo di indirizzamento utilizzato per l'operando in memoria, e quale il registro di segmento per esso utilizzato;
- 2) quanti byte di memoria occupa l'istruzione, e quali informazioni sono specificate nella sua codifica in linguaggio macchina;
- 3) quanti accessi al bus sono richiesti complessivamente per il fetch e l'esecuzione dell'istruzione, supponendo che tutti gli accessi in memoria siano effettuati ad indirizzi pari.

7/ Dato il seguente programma assembly 8086:

```
DATA SEGMENT
    ...
    STRING DB 'giacche'
    STRING_END LABEL BYTE
    STRING_L EQU STRING_END-STRING
    BUFFER DB STRING_L DUP(?)
    ...
DATA ENDS

CODE SEGMENT
    ...
    XOR BX,BX
    MOV CX,STRING_L
CICLO:
    MOV AL,STRING[BX]
    SUB AL,'a'
    ADD AL,'0'
    MOV BUFFER[BX],AL
    INC BX
    LOOP CICLO
PIPP0:
    ...
CODE ENDS
```

- 1) determinare il contenuto del buffer di memoria `BUFFER` quando l'esecuzione raggiunge l'etichetta `PIPP0`;
- 2) stabilire il valore esadecimale dello spiazzamento di `BUFFER`, sapendo che `OFFSET STRING` vale 23;
- 3) spiegare il significato dell'istruzione `LOOP CICLO`.

7/ Dato il seguente programma assembly 8086:

```
DATA SEGMENT
    ...
    STRING DB 'calcolatori'
    STRING_END LABEL BYTE
    STRING_L EQU STRING_END-STRING
    BUFFER DB STRING_L DUP(?)
    ...
DATA ENDS

CODE SEGMENT
    ...
    XOR BX,BX
    MOV CX,STRING_L
CICLO:
    MOV AL,STRING[BX]
    SUB AL,'a'
    ADD AL,'B'
    MOV BUFFER[BX],AL
    INC BX
    LOOP CICLO
PIPP0:
    ...
CODE ENDS
```

- 1) determinare il contenuto del buffer di memoria **BUFFER** quando l'esecuzione raggiunge l'etichetta **PIPP0**;
- 2) stabilire il valore decimale dello spiazzamento di **BUFFER**, sapendo che **OFFSET STRING** vale 12H;
- 3) spiegare il significato dell'istruzione **LOOP CICLO**.