

# Corso di Architettura degli Elaboratori

Scritto d'Esame - 9/2/2026

- Scrivete in testa ai fogli che intendete consegnare, in lettere maiuscole:

Cognome, Nome, Numero di Matricola.

- Non è possibile consultare libri, note, o appunti personali. Non è possibile usare strumenti elettronici. Non è possibile parlare con gli altri studenti.
- Rispondere alle domande in modo puntuale e conciso.

1. **(punti 2)** Qual è il ruolo dell'indirizzo di ritorno all'interno di un record di attivazione?
2. **(punti 2)** Descrivere il ruolo del multiplexer nelle architetture dei calcolatori.
3. **(punti 4)** Disegnare un circuito che controlla il livello di liquido in un serbatoio, in particolare che assicuri che il liquido non debordi. Il circuito ha una sola uscita a 1 bit che, se 1 apre la valvola di sfogo del serbatoio, facendo defluire il liquido. Il circuito ha un ingresso  $L$  a 8 bit, che indica il livello corrente del serbatoio. Il circuito memorizza due parametri,  $max$  e  $ok$ , che contengono numeri binari a 8 bit. Il valore dell'uscita è normalmente 0, ma se  $L \geq max$ , allora va a 1, e torna successivamente a 0 solo quando  $L \leq ok$  (si suppone che  $max > ok$ ). Il circuito ha anche 2 ingressi di controllo, interi a 8 bit, per settare  $max$  e  $ok$ , ognuno con un relativo bit di load. Il funzionamento del circuito è regolato da un clock. Tutti i numeri a 8 bit sono interi in complemento a 2. È possibile usare tutti i circuiti visti a lezione.
4. **(punti 3)** Si consideri il numero decimale  $-37$ . Lo si converta in binario usando sia la codifica modulo e segno che la codifica in complemento a 2 e si descriva il metodo usato.
5. **(punti 4)** Si consideri la funzione booleana  $f(a, b, c, d) = (\bar{a}c + b + c + b)d\bar{b} + \bar{b}d + \bar{b}\bar{a}c$ . Se ne scriva la mappa di Karnaugh e la corrispondente rappresentazione in forma minimale.
6. **(punti 4)** Scrivere un programma Assembly Hack che implementi l'algoritmo descritto dal seguente pseudo-codice, assumendo che  $x$  ed  $y$  siano interi positivi memorizzati rispettivamente in MEM[0] e MEM[1]. Nello svolgimento, l'uso di variabili Assembly Hack è opzionale.

```

int k = 5;
int z = 2x + y;
while(z >= y)
{
    z = z - 5;
    k = k + 1;
    MEM[k] = z;
}

```

7. **(punti 3)** Illustrare graficamente l'evoluzione della pila della VM Hack corrispondente all'esecuzione della seguente funzione, assumendo che i segmenti siano correttamente inizializzati ed i valori di `argument[0]` e `argument[1]` siano, rispettivamente, 1 e 2.

```

function myfunction 1
    push argument 1
    pop local 0
    push argument 0
    push local 0
    lt
    if-goto ELSE
    push argument 0
    return
label ELSE
    push constant 3
    push argument 0
    push local 0
    add
    gt
    return

```

8. **(punti 2)** Calcolare il bit di parità per le seguenti sequenze di bit: 100100, 110101, 001000. Giustificare la risposta.
9. **(punti 3)** Si descriva (con diagramma e/o testo) cosa succede se ad un processore arrivano i seguenti interrupt, assumendo un sistema di gestione degli interrupt con priorità (numeri maggiori corrispondono a priorità maggiore): interrupt (disp. A) con priorità 3 a  $t = 0 \mu s$ ; interrupt (disp. B) con priorità 5 a  $t = 5 \mu s$ ; interrupt (disp. C) con priorità 4 a  $t = 8 \mu s$ ; interrupt (disp. D) con priorità 6 a  $t = 15 \mu s$ . Si assuma che ogni routine di gestione dell'interrupt richieda  $10 \mu s$ .