

# Architettura dei Calcolatori

## Prova d'esame – 06 Giugno 2019 – 1h30min

### PARTE 1 – RISPOSTA SINGOLA

Alle seguenti domande corrisponde una sola risposta corretta fra quelle elencate.

- Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda
- Una risposta errata fa perdere il punteggio negativo riportato a fianco della domanda
- Una risposta lasciata in bianco viene valutata 0

1) [2, -0.5] Assumendo interi a 8 bit, la rappresentazione in complemento a due del numero decimale 128 è:

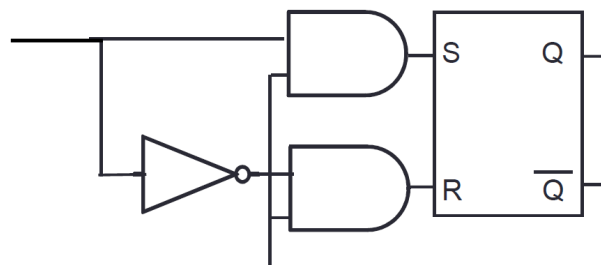
- A) 10000000
- B) 11111111
- C) 01111111
- D) Nessuna delle precedenti risposte

2) [2, -0.5] Nel formato IEEE 754, a che numero decimale corrisponde la seguente rappresentazione binaria?

1 01111111 100000000000000000000000

- A) Meno infinito
- B) -1077936128
- C) -1.5
- D) Nessuna delle precedenti risposte

3) [2, -0.5] Il circuito in figura



- A) Rappresenta un half adder dove non viene riportato in uscita il valore del carry-out
- B) Rappresenta un bistabile Set-Reset
- C) Rappresenta un Latch D
- D) Non è un componente combinatorio notevole

4) [2, -0.5] Le estensioni SIMD di un'architettura:

- A) Servono per implementare costrutti di sincronizzazione (es. utilizzando *load reserved* e *store conditional* in RISC V)
- B) Implementano protocolli di coerenza tra le cache dati di uno (o più) processori simmetrici
- C) Permettono di realizzare una stessa operazione su più operandi diversi
- D) Nessuna delle precedenti risposte

## PARTE 2 – (POSSIBILI) RISPOSTE MULTIPLE

Ognuna delle seguenti domande può avere una o più risposte CORRETTE. Ogni risposta esatta viene calcolata +1, ogni risposta errata -0.5, e ogni risposta lasciata in bianco 0.

5) Per creare una ALU a  $n$  bit a partire da  $n$  ALU a 1 bit, occorre:

- A) Collegare il risultato della ALU  $i$ -esima all'ingresso della ALU  $(i+1)$ -esima
- B) Collegare il carry-in della ALU  $i$ -esima al carry-out della ALU  $(i+1)$ -esima
- C) Collegare il carry-out della ALU  $i$ -esima al carry-in della ALU  $(i+1)$ -esima
- D) Ottenere il bit di Overflow per operazioni in complemento a 2 attraverso un XOR tra i carry-out delle due ALU operanti sui bit più significativi

6) Per realizzare un sommatore di numeri a virgola mobile

- A) E' opportuno togliere tante cifre della mantissa quanta la differenza negli esponenti dei due addendi
- B) E' opportuno shiftare a destra la mantissa del numero con l'esponente minore affinché entrambi gli addendi abbiano lo stesso esponente
- C) E' possibile sommare le mantisse di due numeri aventi lo stesso esponente e normalizzare il risultato. In tal caso non occorre controllare la presenza di overflow/underflow.
- D) Potrebbe essere necessario arrotondare il risultato affinché rispetti il formato IEEE 754, ed eventualmente rinormalizzarlo.

7) Un sistema di branch prediction:

- A) Consente di risolvere il problema degli structure hazards, soprattutto in architetture provviste di cache
- B) Permette di ridurre le penalità dovute a control hazards, soprattutto in pipeline lunghe
- C) Può essere implementato con una semplice ALU a  $n$  bit
- D) Se di tipo dinamico, tiene traccia del risultato degli ultimi salti condizionali, e assume che il comportamento sia identico in futuro.

8) Il blocco di istruzioni seguente è idempotente, cioè lo stato della macchina al termine dell'esecuzione è identico a quello d'inizio, eccettuato il *program counter*.

- A) `andi x11, x11, 0`
- B) `add x11, x11, x0`
- C) `xor x28, x28, x29`  
`xor x28, x28, x29`
- D) `beq x0, x0, x0`

### PARTE 3 – DOMANDE APERTE

Una risposta esatta fa acquisire il punteggio positivo riportato a fianco della domanda. Una risposta errata può eventualmente causare una penalità che dipende dalla gravità dell'errore. Una risposta lasciata in bianco viene calcolata: 0.

9) [3 pt.] Si scriva in assembly RISC-V la seguente funzione C.

```
int count(int *save, int k) {
    int i = 0;
    while (save[i] == k) {
        i += 1;
    }
    return i;
}
```

10) [5 pt.] Si spieghino, con proprietà di linguaggio, i concetti di cache, set associativity e replacement policy

# Architettura dei Calcolatori

## Prova d'esame – 06 Giugno 2019 – 1h30min

Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_

Matricola: \_\_\_\_\_ Corso di Laurea: \_\_\_\_\_

**Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 9 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Questa è l'unica pagina che dovete consegnare. Per comodità avete anche un copia di questa pagina per calcolare il voto da sole/soli durante la correzione.**

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					2	-0,5
2					2	-0,5
3					2	-0,5
4					2	-0,5
5						
6						
7						
8						
9						
10						

**Risposta alla domanda 9 (9 pt):**

**Risposta alla domanda 10 (7 pt):**



# Architettura dei Calcolatori

## Prova d'esame – 06 Giugno 2019 – 1h30min

Nome: \_\_\_\_\_ Cognome: \_\_\_\_\_

Matricola: \_\_\_\_\_ Corso di Laurea: \_\_\_\_\_

**Indicare le risposte corrette apponendo una croce nella casella corrispondente. Per superare la prova bisogna aver raggiunto almeno 9 punti nelle domande a risposta singola/multipla, ed almeno 15 complessivamente. Questa è l'unica pagina che dovete consegnare. Per comodità avete anche un copia di questa pagina per calcolare il voto da sole/soli durante la correzione.**

	Risposte				Punti/ Penalità	
	A	B	C	D		
1					2	-0,5
2					2	-0,5
3					2	-0,5
4					2	-0,5
5						
6						
7						
8						
9						
10						

**Risposta alla domanda 9 (9 pt):**

**Risposta alla domanda 10 (7 pt):**