

*Cognome e nome dello studente:*

*Matricola:*

1. [4] Convertire in decimale, il numero binario in virgola mobile codificato mediante la codifica IEEE 754: 1100 0110 0000 0000 0000 0000 0000. Quale sarà la rappresentazione decimale del numero binario in virgola mobile codificato successivamente. Qual è la risoluzione della codifica in virgola mobile IEEE 754? Scrivere in complemento a 2 su 8 bit la sottrazione espressa da numeri in base 10: 8-10 e calcolarne il risultato in binario.
2. [7] Calcolare mediante un algoritmo firmware la moltiplicazione tra i due numeri interi binari 1100 e 11 su 4 bit. Scrivere l'algoritmo e progettare il circuito firmware associato all'algoritmo scelto. **Dimensionare tutti i cammini e definire la loro funzione.** Estendere il circuito disegnato per implementare anche la divisione tra numeri interi su 4 bit e spiegarne il funzionamento. E' un circuito sincrono o asincrono? Perché? Progettare l'unità di controllo del circuito firmware che avete disegnato per eseguire la moltiplicazione. Quale sarà il suo cammino critico? E quale sarà la sua complessità?
3. [7] Progettare una macchina a stati finiti che implementa la funzione “Search” in un testo scritto. La macchina scandisce il testo dall'inizio alla fine, leggendo un carattere alla volta, e segnala in uscita (con un 1) tutte le volte che viene trovata la stringa “ 11 ”. La macchina scorre il testo sequenzialmente, leggendo un carattere alfanumerico alla volta. Il carattere alfanumerico può essere una qualsiasi lettera dell'alfabeto, un numero, uno spazio o un qualsiasi carattere di interpunzione: “!”, “?”,... Si supponga che lo stato iniziale coincida con il carattere nullo: S0 = “”. **Viene richiesta la presenza di uno spazio sia prima che dopo “11”.** Rappresentare la funzione stato prossimo della macchina a stati finiti come SOP, come PLA e come ROM. Semplificare il più possibile tutte le SOP. Quali specifiche occorre dare per le PLA e le ROM? Qual è l'implementazione più vantaggiosa? Cosa sono i mintermini di una funzione? Rappresentare la Macchina a Stati Finiti come Macchina di Huffman. Si possono calcolare il cammino critico e la complessità? In caso affermativo calcolare cammino critico e complessità. In caso negativo, spiegare perché non si possono calcolare.
4. [2] Cosa si intende per sommatore ad anticipazione di riporto e per sommatore a propagazione di riporto. In base a quali principi cardine sono realizzati? Quali sono i vantaggi e svantaggi di questi due sommatori?
5. [3] Progettare con sole porte NAND un multiplexer a due ingressi. Calcolare il cammino critico e complessità del multiplexer così realizzato e confrontarlo con un multiplexer realizzato utilizzando combinazioni di porte AND, OR e NOT.
6. [3] Progettare **con le porte logiche** la porta di scrittura di un register file con 4 registri da 2 bit e un **segnale di reset** che consente di portare a zero il numero del registro specificato per la porta di scrittura. E' più opportuno utilizzare flop-flop o latch per i registri di questo register file? Perché? Perché i latch sincroni vengono chiamati “trasparenti”?
7. [3] Progettare con le porte logiche un moltiplicatore hardware per numeri interi su 2 bit. Calcolare complessità e cammino critico.
8. [2] Supponiamo che l'istruzione corrente si trovi all'indirizzo 0x0000 0300, scrivere l'istruzione assembler di branch che effettua il salto condizionato all'indirizzo 0x000002C0, e tradurla in linguaggio macchina. Si scelgano due registri a piacere.
9. [2] Disegnare il ciclo di esecuzione di un'istruzione su un'architettura MIPS a singolo ciclo. Quante fasi si distinguono? Quando l'architettura capisce di che istruzione si tratta? I componenti appartenenti a quali fasi vengono configurati dall'unità di controllo? Perché? Cos'è un'ISA? Possono due CPU avere la stessa ISA? Due CPU diverse devono avere una ISA necessariamente diversa? Descrivere come viene suddivisa in modo logico per convenzione una memoria principale dai processori MIPS e quali sono le ragioni per la scelta dei confini dei diversi segmenti.
10. [2] Semplificare l'espressione logica: xyz + !yz.