

1) Progettare la rete sequenziale Level Level Clocked (LLC) riconoscitrice della sequenza "mano" ipotizzando che l'alfabeto di ingresso sia costituito solo dai caratteri: "a, m, n, o" e che l'uscita sia costituita solo da "stringa riconosciuta, stringa non riconosciuta". È sufficiente descrivere le equazioni di eccitazioni di un solo flip-flop, per semplicità utilizzare flip-flop di tipo D.

2) Dato il processore z64, descrivere la parte della macchina sequenziale del SCO atta all'esecuzione dell'istruzione `movl $0xbbbb, %r12` considerando anche la parte di fetch. Disegnare prima l'architettura interna del processore.

3) Descrivere come il processore z64 gestisce le interruzioni per attivare i driver ed inoltre disegnare il diagramma di flusso del ciclo istruzione considerando la gestione delle richieste di interruzione e del rilascio dei bus per l'arrivo di una Bus Request.

4) Descrivere le possibili organizzazioni dei bus di connessione tra un processore, la memoria di lavoro e i dispositivi di ingresso/uscita. Dopodiché descrivere le differenze tra un bus sincrono ed uno asincrono.

5) Scrivere un segmento di programma in linguaggio assembly del processore didattico a 32 bit di 6 istruzioni in cui ci sia almeno un conflitto define use ed uno load use. Modificarlo per evitare i conflitti oppure, nel caso in cui non sia possibile evitare tutti i conflitti, inserire il minor numero di NOP. Inoltre dare le linee guida per risolvere via hardware tali tipi di conflitti tramite la propagazione in avanti. Per comodità degli esaminandi si riporta la sintassi delle istruzioni:

- `nop - operL/A regsorg1, regsorg2, regdest`
- `load regdest, offset(regbase)`
- `store regsorgM, offset(regbase)`
- `jumpX indirizzo`