INF 250 - Lista de Exercícios 1.3

Máquinas de Estados Finitos (FSM)

Exercício 1 - Nível Básico

Comando:

Crie uma tabela verdade para um sistema com 2 entradas (COIN, TIMER) e 2 saídas (OPEN, ALARM), com as seguintes regras:

1. **OPEN** fica ativa no primeiro pulso de **COIN = 1**.
2. **ALARM** fica ativa no segundo pulso de **COIN = 1**.
3. No terceiro pulso de **COIN = 1**, ambas saídas desligam e o sistema volta ao estado inicial.
4. **TIMER = 1** desliga tudo e volta ao estado inicial a qualquer momento.
5. Considere que **COIN** gera pulsos (0→1→0).

Sua tarefa:

- Elabore a tabela de estados
- Desenhe o diagrama de estados
- Implemente o código Verilog

Exercício 2 - Nível Intermediário

Comando:

Desenvolva uma FSM para controle de uma cancela automática com 3 entradas (SENSOR_CARRO, BOTAO_ABRIR, EMERGENCIA) e 2 saídas (MOTOR_SUBIR, MOTOR_DESCER), seguindo estas regras:

1. **MOTOR_SUBIR** ativa quando **BOTAO_ABRIR = 1** e não há emergência.
2. **MOTOR_DESCER** ativa automaticamente 5 segundos após **MOTOR_SUBIR** parar.
3. Se **SENSOR_CARRO = 1** durante descida, **MOTOR_DESCER** para e **MOTOR_SUBIR** ativa.
4. **EMERGENCIA = 1** para qualquer movimento e mantém cancela parada.
5. Sistema retorna ao estado inicial quando **EMERGENCIA = 0**.

Sua tarefa:

- Elabore a tabela de estados
- Desenhe o diagrama de estados

- Implemente o código Verilog

Exercício 3 - Nível Básico+

Comando:

Projete uma máquina de estados para um sistema de segurança residencial com 2 entradas (SENSOR, ARM) e 3 saídas (LED_VERDE, LED_VERMELHO, SIRENE), com as seguintes regras:

1. **LED_VERDE** fica ligado quando sistema está desarmado (**ARM = 0**).
2. **LED_VERMELHO** pisca quando sistema está armado (**ARM = 1**) e não há detecção.
3. **SIRENE** ativa no primeiro pulso de **SENSOR = 1** com sistema armado.
4. **LED_VERMELHO** fica fixo (sem piscar) quando sirene está ativa.
5. Sistema desarma automaticamente após 10 segundos de sirene ativa.

Sua tarefa:

- Elabore a tabela de estados
- Desenhe o diagrama de estados
- Implemente o código Verilog

Exercício 4 - Nível Intermediário+

Desenvolva uma máquina de estados para uma máquina de café que aceita moedas de diferentes valores. O sistema possui entradas para detecção de moedas de 25, 50 e 100 centavos (M25, M50, M100) e uma entrada de seleção de bebida (SEL). As saídas são: CAFE (libera café), TROCO_25, TROCO_50 (liberam troco) e RESET_SYSTEM (reinicia para próximo cliente). O café custa 75 centavos. Quando o valor inserido for igual ou superior a 75 centavos e SEL=1, o café é liberado. Se o valor for superior, o troco correto deve ser dado usando o menor número de moedas possível. O sistema deve contar o valor acumulado: 25→50→75→100→125→150. Se o cliente inserir mais de 150 centavos, a máquina rejeita moedas adicionais. Após liberar café e troco, RESET_SYSTEM fica ativo por um ciclo e retorna ao estado inicial.

Exercício 5 - Nível Avançado

Projete uma FSM para controle de um elevador de 3 andares (0, 1, 2) com botões internos B0, B1, B2 e sensores de andar S0, S1, S2. As saídas são MOTOR_SOBE, MOTOR_DESCE, PORTA_ABRE e LED_ANDAR (2 bits indicando andar atual). O elevador inicia no andar 0 com porta aberta. Quando um botão é pressionado, a porta fecha, o elevador se move até o andar solicitado (MOTOR_SOBE ou MOTOR_DESCE ativos conforme necessário) e para quando o sensor correspondente detecta chegada. A porta então

abre e permanece aberta até nova solicitação. Se múltiplos botões forem pressionados, o elevador atende o mais próximo primeiro. O sistema deve ter estados para: parado_porta_aberta em cada andar, movendo_para_cima, movendo_para_baixo e fechando_porta. Considere que os sensores S0, S1, S2 são mutuamente exclusivos e apenas um fica ativo por vez indicando a posição atual.

Exercício 6 - Nível Intermediário

Crie uma máquina de estados finitos para um sistema de irrigação automática que monitora umidade do solo através de dois sensores: UMIDADE_BAIXA e UMIDADE_CRITICA. O sistema possui também uma entrada CHUVA que detecta precipitação. As saídas são BOMBA_AGUA (ativa irrigação), ALERTA_SECA (LED de aviso) e MODO_ECONOMIA (reduz consumo). O funcionamento é: se UMIDADE_BAIXA=1 e CHUVA=0, ativa BOMBA_AGUA. Se UMIDADE_CRITICA=1, ativa BOMBA_AGUA independente da chuva e também ALERTA_SECA. Durante irrigação, se CHUVA=1, para a bomba mas mantém monitoramento. MODO_ECONOMIA ativa quando não há irrigação por mais de 3 ciclos consecutivos. O sistema tem 6 estados principais: monitorando, irrigando_normal, irrigando_critico, pausado_por_chuva, modo_economia e erro_sensor (quando ambos sensores ficam ativos simultaneamente).

Orientações Gerais:

1. **Tabela de Estados**: Deve incluir estado atual, entradas, próximo estado e saídas
2. **Diagrama de Estados**: Use círculos para estados e setas rotuladas para transições
3. **Código Verilog**: Implemente usando always blocks para lógica sequencial e combinacional
4. **Reset**: Considere um sinal de reset assíncrono para todos os exercícios
5. **Codificação**: Use codificação binária ou Gray para os estados

Dicas:

- Comece sempre identificando todos os estados possíveis
- Defina claramente as condições de transição
- Teste mentalmente alguns casos antes de finalizar
- No Verilog, separe a lógica de próximo estado da lógica de saída