

RELATÓRIO PRÁTICA 4

Título: Laboratory exercise 7 – Finite State Machines

NOME: Luis Henrique Ponciano dos Santos NUSP: 155777660

NOME: Gabriel de Araujo NUSP: 14571376

NOME: Gabriel Demba NUSP: 15618344

Observação importante: nesse exercício foi solicitado somente as partes 1 e 4, ou seja, a máquina de estados de 9 estados e a implementação do conversor de código morse com máquina de estados

Parte 1: Máquina de estados com 9 estados

Visão geral

- O exercício propõe o desenvolvimento de uma Máquina de Estados Finitos (FSM) para reconhecer sequências de quatro entradas consecutivas iguais, sejam "1111" ou "0000". A FSM possui uma entrada chamada w e uma saída z, que será ativada (valor 1) sempre que uma dessas sequências for detectada, incluindo casos de sobreposição

Descrição do código

- O código implementa uma máquina de estados finitos (FSM) que possui 9 estados. A máquina transita entre os estados com base na entrada de controle w, e o estado atual é armazenado em flip-flops D. O processo de transição de estado é controlado por w, e o próximo estado é determinado com base no estado atual. Se o valor de w for 0, a FSM transita para um estado diferente; se for 1, ela pode transitar para outro conjunto de estados. O estado de saída z é ativado (valor 1) nos estados E e I, caso contrário, z é 0. O estado atual é mantido e atualizado a cada borda de subida do clock ou quando o reset é acionado, retornando ao estado inicial. O código define a lógica de transição de cada estado e controla a saída z com base nos estados específicos.

Parte 4: Código morse com máquina de estados

Visão geral

- Função: utilizou-se máquina de estados para implementar o conversor de código morse visto na última aula.

Descrição do código

- O código implementa uma máquina de estados finitos (FSM) que exibe sequências de código Morse correspondentes a letras (A até H) selecionadas por switches. Cada letra

é representada por uma sequência de pontos e traços, que são armazenados em um vetor chamado `morse_code`. A FSM possui estados para identificar o início da sequência, exibir um ponto ou traço, e pausar entre os sinais. A entrada de clock controla as transições de estado, e o reset reinicia a FSM. Os pontos e traços são exibidos como pulsos em um LED, enquanto a letra correspondente é exibida em um display de 7 segmentos. O sistema também inclui temporizadores para definir a duração dos pontos, traços e pausas, garantindo a exibição precisa do código Morse selecionado. A máquina de estados começa no estado IDLE, que é o estado de espera. Nesse estado, o sistema não faz nada até que o botão de início seja pressionado. Quando o botão é pressionado, a máquina transita para o estado FIND_FIRST_ONE, onde o sistema verifica os bits do código Morse da letra selecionada, procurando o primeiro bit 1 (traço ou ponto). Se um 1 for encontrado, o sistema então entra nos estados DOT ou DASH dependendo se o bit encontrado é um ponto ou um traço. No estado DOT, o LED acende por um curto período de tempo, representando um ponto, e depois apaga. No estado DASH, o LED acende por um período maior, representando um traço, e também apaga depois. Após exibir o ponto ou traço, o sistema entra no estado PAUSE, onde há uma pausa entre os sinais. A máquina então verifica o próximo bit do código Morse e repete o processo até que toda a sequência de código Morse seja exibida. Quando toda a sequência for exibida, o sistema retorna ao estado IDLE, pronto para reiniciar o processo.

Simulações

- O comportamento foi estudado a partir de uma simulação via placa FPGA.
- Resultados: o comportamento nos mostrou que foi possível converter corretamente os valores em código morse conforme o esperado através de uma máquina de estados.