SERVIÇO PÚBLICO FEDERAL · MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE VIÇOSA · UFV
CAMPUS FLORESTAL

Trabalho prático 2 - ISL

Circuitos sequenciais

Bruno Vicentini Ribeiro - 5907
Vitor Mathias Andrade - 5901

Florestal - MG
2025

Sumário

1. Introdução

Este trabalho tem como objetivo aplicar os conhecimentos adquiridos ao longo das aulas de Introdução aos Sistemas Lógicos (ISL) por meio do desenvolvimento de uma máquina lógica, a qual será responsável por controlar os movimentos dos robôs em um campeonato de corrida em pista numérica, personalizada com uma sequência total de algarismos de 0 a 9.

Sua principal função é monitorar o trajeto percorrido, identificar falhas e validar se o robô completa o percurso conforme as condições estabelecidas.

2. Organização

A pista é composta por seis algarismos, dispostos da seguinte forma: (5, 9, 0, 0, 7, 1) (um algarismo 5, um algarismo 9, dois algarismos 0, um algarismo 7 e um algarismo 1). Para verificar a trajetória dos robôs, uma Máquina de Estados Finitos classifica as condições em três categorias:

- Sucesso total (S): sem erros;
- Sucesso parcial (P): até um erro;
- Falha (F): dois ou mais erros.

Em situações de falha, um display de 7 segmentos exibirá a letra F; para sucesso total, exibirá S; e para sucesso parcial, P.

3. Desenvolvimento

Para o desenvolvimento da máquina lógica e para garantir seu correto funcionamento, foram utilizadas três ferramentas de programação: JFlap, Verilog e FPGA.

- Diagrama de transição de estados no JFlap:

Para monitorar as movimentações do robô, foi criado um diagrama de transição de estados, representando a sequência de estados, os eventos que provocam a transição entre eles e as ações resultantes de cada mudança.

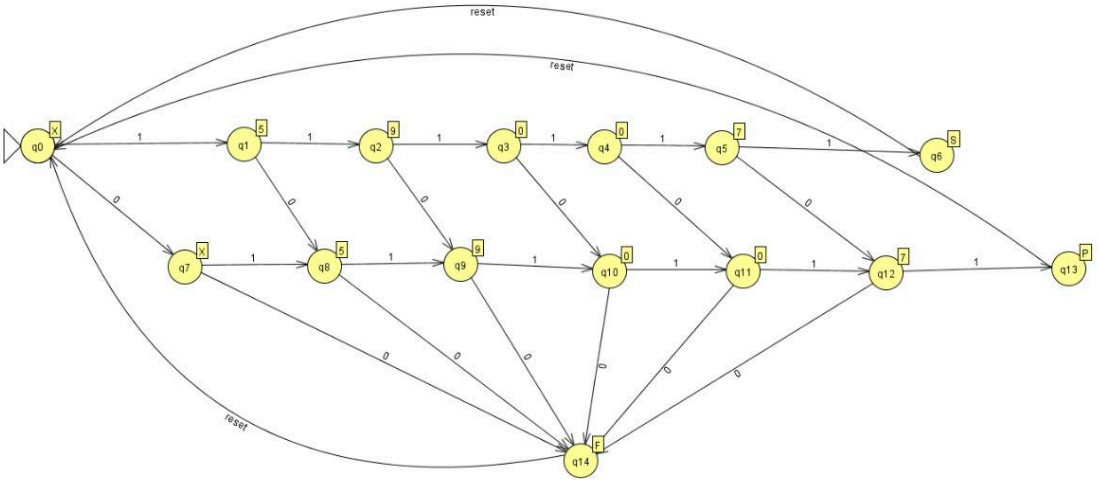

Figura 1: Diagrama de transição de estados no JFlap.

Estrutura do diagrama de transição de estados:

- Estado inicial (q0): o ponto de partida é onde o robô inicia o processamento da sequência. Ao ler corretamente o primeiro número (5), ele transita para q1. Se houver erro na leitura, o sistema transitará para o q7;

- Estados intermediários (q1, q2, q3, q4, q5, q7, q8, q9, q10, q11 e q12):
    - Leitura correta: transita para o próximo estado;
    - Leitura incorreta: conforme o número de erros acumulados, o sistema pode desviar para um caminho que levará ao estado de falha (F) ou sucesso parcial (P).

- Estados finais:
    - Sucesso total (S, estado q6): o robô completa a trajetória sem erros;
    - Sucesso parcial (P, estado q13): o robô comete até um erro durante a sequência, mas consegue finalizar o percurso;
    - Falha (F, estado q14): o robô comete dois erros, o que leva o sistema a classificar a trajetória como uma falha.

Nesse contexto, o valor 1 indica que o número informado está correto, pertencendo à sequência definida e seguindo a ordem estabelecida. Já o valor 0 sinaliza que o número informado está incorreto, mesmo que eventualmente possa fazer parte da sequência, pois não segue a ordem correta prevista pela transição de estados. Além disso, os rótulos "RESET" indicam a opção de reiniciar o sistema.

- Implementação de máquina de estados finitos em Verilog:

Utilizando o Verilog HDL, foi criada uma máquina de estados finitos que lê as entradas e, com base nelas, desloca-se para o próximo estado. Se os valores inseridos forem corretos, o estado final será de sucesso total; se houver um erro durante a sequência, o estado final será de sucesso parcial; e, se houver dois erros, o estado final será de falha. As entradas válidas são apenas entre o intervalo de 0 a 9.

Robo.v

```
module PistaRobos (
    input clk,
    input insere,
    input reset,
    input [3:0] numero,
    output reg led_erro,
    output reg [3:0] EstadoDisplay,
    output reg C1, C2, C3, C4, C5, C6, C7
);
//declaracao das entradas OK

    parameter S0 = 4'b0000,
              S1 = 4'b0001,
              S2 = 4'b0010,
              S3 = 4'b0011,
              S4 = 4'b0100,
              S5 = 4'b0101,
              S6 = 4'b1111,
              SP = 4'b0110, // Sucesso Parcial
              F = 4'b0111; // Falha

// definicao dos estados OK



    reg [3:0] TP;
    reg [3:0] EstadoAtual, proxEstado;
    reg [3:0] pista [0:5]; // Pista com os números
    integer posic;
    integer parcial;
//variaveis internas OK

//Inicialização
    initial begin
        pista[0] = 4'b0101;
        pista[1] = 4'b1001;
        pista[2] = 4'b0000;
        pista[3] = 4'b0000;
        pista[4] = 4'b0111;
        pista[5] = 4'b0001;
        EstadoAtual = S0;
     posic = 0;
        led_erro = 0;
          TP = S0;
          parcial = 0;
        EstadoDisplay = S0;
    end
```

Figura 2: Implementação da máquina de estados finitos em Verilog.

- Implementação do módulo de simulação:
  Utilizando o Verilog HDL, foi criado um módulo de simulação para gerar ondas.

PistaRobos_tb.v

```v
`include "Robo.v"
`timescale 1ns/1ps

module PistaRobos_tb;


    reg clk;
    reg reset;
    reg insere;
    reg [3:0] numero;

    wire led_erro;
    wire [3:0] display_estado;
    wire C1, C2, C3, C4, C5, C6, C7;


    PistaRobos uut (
        .clk(clk),
        .reset(reset),
        .insere(insere),
        .numero(numero),
        .led_erro(led_erro),
        .display_estado(display_estado),
        .C1(C1), .C2(C2), .C3(C3), .C4(C4), .C5(C5), .C6(C6), .C7(C7)
    );


    always #5 clk = ~clk;

    initial begin

        clk = 0;
        reset = 0;
        insere = 0;
        numero = 4'b0000;
```

Figura 3: Implementação do módulo de simulação.

- Simulação de formas de ondas:

O GtkWave foi utilizado para visualizar a simulação de ondas por meio do arquivo .vcd. Essa simulação foi de grande importância, pois permitiu a detecção prévia de erros no código em estudo antes de sua implementação no FPGA, garantindo maior eficiência e confiabilidade no desenvolvimento do projeto.

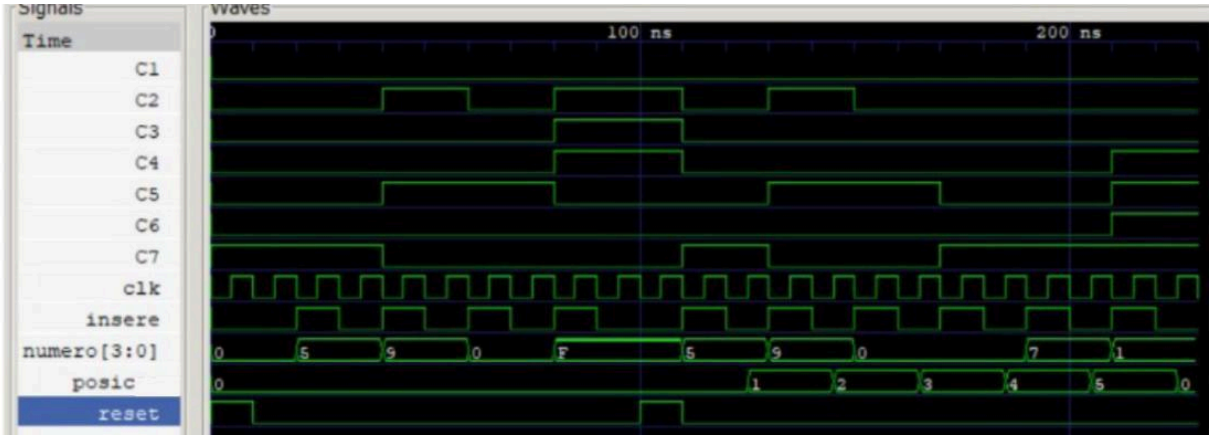

Figura 4: Simulação de formas de ondas.

- Implementação no FPGA DE2-115:

Após a implementação do código em Verilog e a realização dos testes, foi realizada a implementação no FPGA DE2-115. O vídeo com a demonstração do funcionamento se encontra na aba de resultados.

4. Resultados

Os resultados deste trabalho podem ser visualizados no link a seguir: https://youtu.be/e9RW1VqFJ8Y?si=JB0fV30pBZE3ZDN1.

Durante os testes no FPGA, observou-se que o sistema funcionou conforme o esperado. A seguir, são detalhados os principais resultados:

- Será exibido no display de sete segmentos a entrada inserida pelo usuário;
- Caso a sequência correta seja digitada, ao final da inserção do último número será exibido no display a letra “S”, indicando sucesso total; caso tenha um erro na sequência, após a inserção do último número da sequência será exibido no display a letra “P”, indicando sucesso parcial. Caso tenham ocorrido dois erros ao inserir a sequência, será exibido a letra “F”, indicando falha.

5. Conclusão

Em conclusão, a prática deste trabalho permitiu consolidar os conhecimentos adquiridos ao longo das aulas, utilizando JFlap, Verilog e GTKWave para realizar a modelagem e os testes do código antes da implementação no FPGA. Esse conjunto de ferramentas se mostrou eficiente ao transformar ideias em implementações reais, reforçando a importância de sua aplicação no desenvolvimento de sistemas digitais.

Além disso, a implementação da máquina lógica para monitorar o trajeto percorrido pelos robôs em uma pista numérica demonstrou uma aplicação direta de conceitos como a máquina de estados finitos. Durante o processo, foi possível identificar previamente os erros e corrigi-los por meio das simulações feitas no GTKWave.

Desse modo, o trabalho possibilitou aos seus desenvolvedores aprimorar habilidades de resolução de problemas, promovendo uma abordagem mais prática diante dos desafios.

A partir dos resultados, confirma-se o êxito do trabalho desenvolvido, uma vez que todas as entradas foram processadas corretamente e as saídas foram exibidas conforme esperado.

6. Referências

[1] Máquina de Moore (Linguagem, Autômatos e Programação). 2023. Disponível em: https://youtu.be/DwVn4MUeqss?si=Fq1duLVcbV-ZU3Lx. Último acesso em: 14 de janeiro de 2025.

[2] JFLAP. 2018. Disponível em: https://youtube.com/playlist?list=PLeaAjeNjt7tTAH3LvvMVeR_rOVOgLLx6D&si=GWpPLFiyBnNaoH9h. Último acesso em: 14 de janeiro de 2025.

[3] [CIRCUITOS DIGITAIS] Aula 58 - Diagrama de Transição de Estados. 2020. Disponível em: https://youtu.be/Q0QNvjRt_pk?si=PoMKzLz7a5sfly1c. Último acesso em: 14 de janeiro de 2025.