



UNIVERSIDADE FEDERAL DE VIÇOSA - CAMPUS
FLORESTAL
INSTITUTO DE CIÊNCIAS EXATAS E
TECNOLOGICAS
CIÊNCIA DA COMPUTAÇÃO
INTRODUÇÃO AOS SISTEMAS LÓGICOS DIGITAIS

LÓGICA SEQUENCIAL

GUILHERME SÉRGIO 3854

LARISSA ISABELLE 3871

OTÁVIO SANTOS 3890

PEDRO CARDOSO 3877

FLORESTAL

2019

1. SUMÁRIO

2. INTRODUÇÃO.....	3
3. DIAGRAMA DA MÁQUINA DE ESTADOS	4
4. VERILOG.....	5
5. FPGA.....	5
6. CONSIDERAÇÕES FINAIS	7

2. INTRODUÇÃO

Este relatório consiste no segundo trabalho prático da disciplina de Introdução aos Sistemas Lógicos Digitais. Nesse trabalho foi elaborado o módulo classificador de adjetivos e advérbios da linguagem fictícia *xy/o*, baseando-se no trabalho anteriormente desenvolvido pelo grupo sobre essa mesma linguagem.

Assim, foi feito primeiramente o diagrama de estados, seguindo o conceito de máquina de Moore. Posteriormente foi desenvolvido o código em Verilog dos módulos de transcrição, simulação e síntese da máquina de estados. Por fim, testamos o funcionamento do código na FPGA.

3. DIAGRAMA DA MÁQUINA DE ESTADOS

Para iniciar o desenvolvimento da máquina de estados, fizemos seu diagrama utilizando o software JFLAP com o intuito de facilitar a montagem do diagrama. Para a visualização mais clara das transições e estados do diagrama, fizemos primeiramente o diagrama sem os estados de erro.

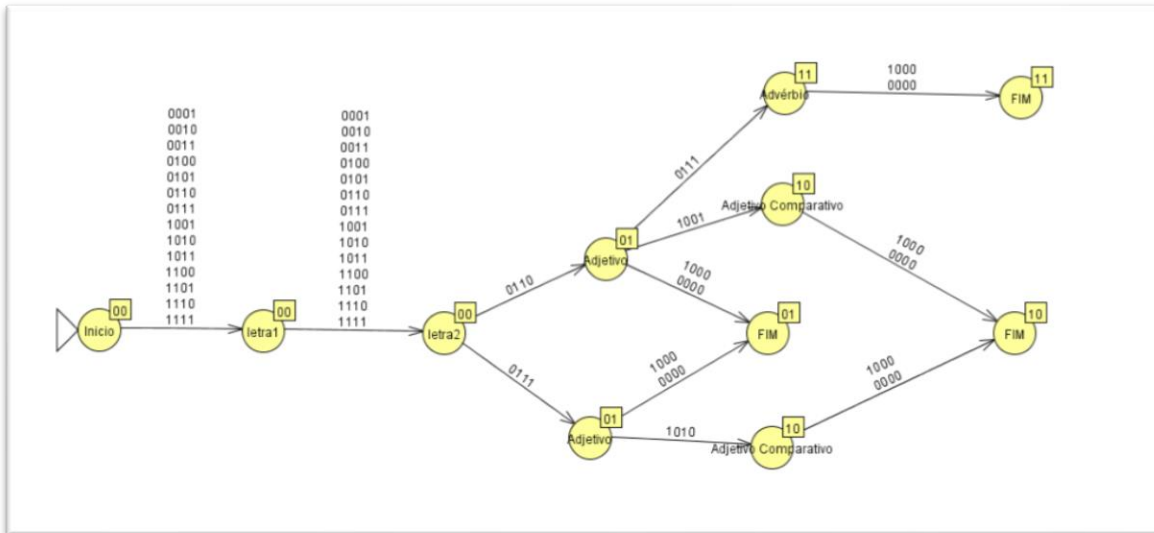


Figura 1 Diagrama da máquina de estados sem o estado de erro.

Posteriormente, no entanto, para melhor compreender o que deveria acontecer na máquina de estados, fizemos a implementação do diagrama com os estados de erro.

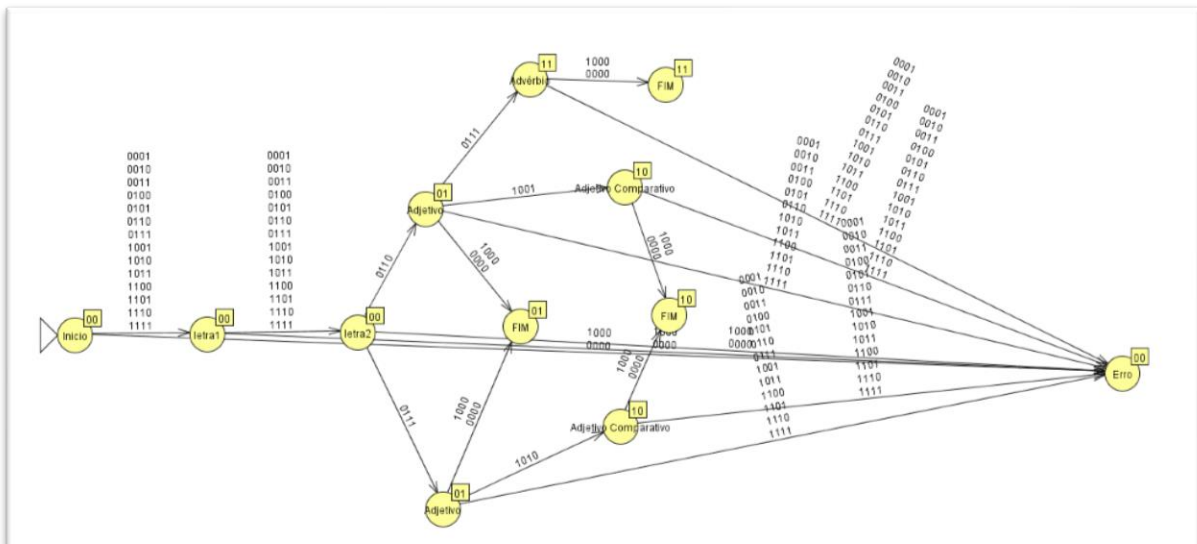


Figura 2 Diagrama da máquina de estados com o estado de erro.

Dessa forma conseguimos compreender o funcionamento da máquina de estados e como deveria ser feita a transcrição para o Verilog.

4. VERILOG

Após o desenvolvimento do diagrama, fizemos o módulo de transcrição da máquina de estados em Verilog, objetivando o teste na FPGA. Seguindo o exemplo mostrado em aula, implementamos o módulo de transcrição com os seguintes parâmetros de entrada:

```
module tp2 (clock, reset, ok, tom, in1, in2, in3, out1, out2, fim);
```

Após fazer o módulo de transcrição, elaboramos o módulo de simulação para testar se a implementação do módulo de transcrição estava correta. Utilizamos o software GTKWave para gerar as formas de onda da simulação.

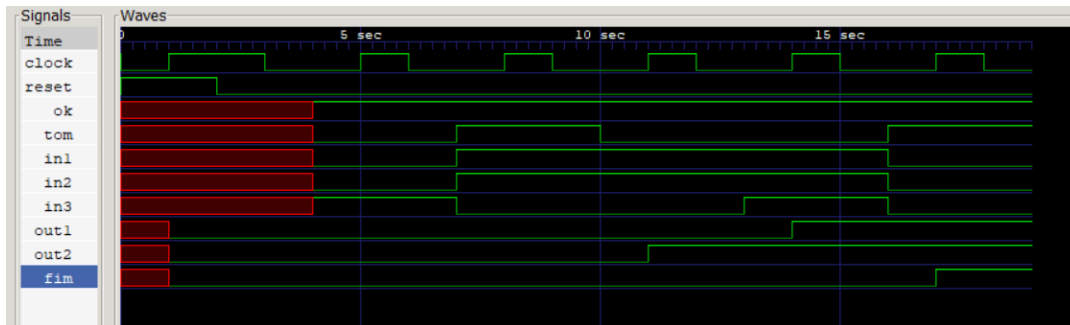


Figura 3 GTKWave simulação.

Dessa forma, conseguimos observar que o código implementado estava gerando as saídas esperadas.

Por fim, utilizando o programa Quartus, geramos o módulo de síntese do código para seu funcionamento da FPGA e definimos como clock um switch.

5. FPGA

Após desenvolvermos o código de transcrição e de síntese em Verilog, testamos o funcionamento do módulo. Veja o a saída do mesmo exemplo mostrado acima na FPGA:

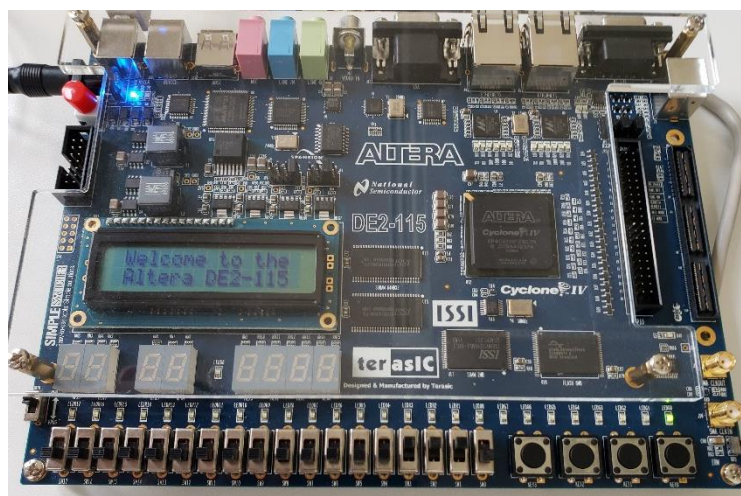


Figura 4 FPGA após a detecção de um adjetivo até o momento.

Observe que um LED verde se acendeu, representando 01, ou seja, adjetivo. O LED vermelho que representa o fim continua apagado.

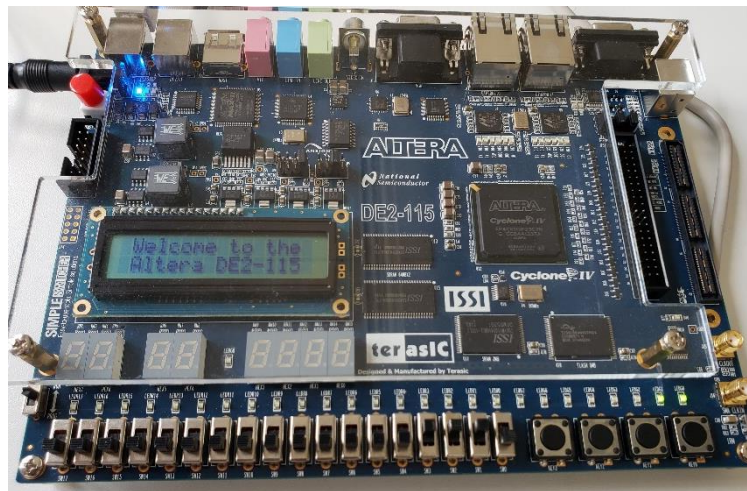


Figura 5 FPGA após a detecção de um advérbio até o momento.

Observe que agora dois LEDs verdes acenderam, representando 11, ou seja, advérbio. O LED vermelho que representa o fim continua apagado.

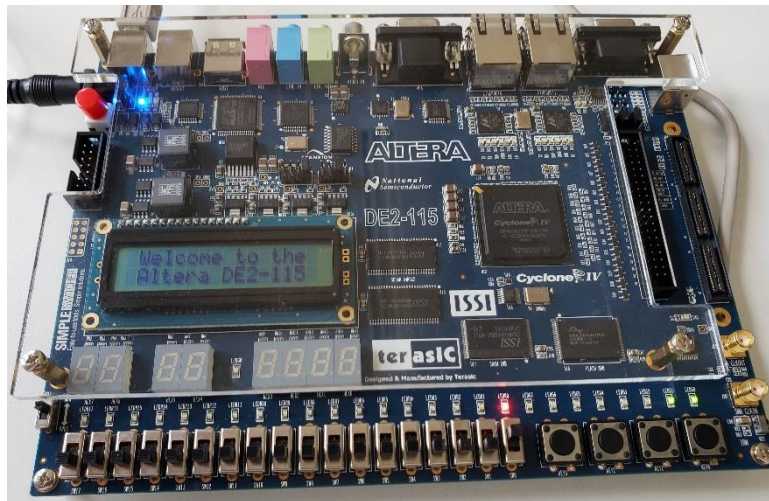


Figura 6 FPGA após chegar ao fim de uma entrada.

Observe que agora os dois LEDs verde continuam acesos, representando 11, advérbio, mas agora também está aceso o LED vermelho representando o fim.

Dessa forma, foi implementado o módulo classificador de adjetivos e advérbios como especificado na documentação.

6. CONSIDERAÇÕES FINAIS

Após a conclusão desse trabalho, foi possível perceber que a implementação de uma máquina de estados requer um bom entendimento prévio de seu diagrama para que o desenvolvimento em Verilog seja feito mais facilmente. Além disso, percebemos a importância de se fazer o módulo de simulação para ver se a implementação está correta antes de se testar na FPGA.

Por fim, concluímos que o correto desenvolvimento de uma máquina de estados requer muitos testes para se obter a correta implementação tanto do módulo de transcrição quanto da síntese para a FPGA.