Aluna: Carla Beatriz Ferreira (2022097470)
Disciplina: Introdução aos Sistemas Lógicos (TW)
Professor: Marcos Augusto Menezes Vieira
Data: 03/12/2023

UFMG
UNIVERSIDADE FEDERAL DE MINAS GERAIS

Trabalho Prático 2 - Introdução aos Sistemas Lógicos

1. Introdução

A proposta deste trabalho prático é desenvolver um circuito que resolva o problema da necessidade de um cronômetro a ser utilizado para auxiliar a artista Ariana Grande no intervalo entre suas músicas. Na prática, o objetivo é desenvolver um contador de 0 a 9 em verilog, utilizando os conceitos estudados de máquinas de estados finitas.

Esta documentação apresenta as seguintes informações:

1. Introdução
2. Decisões de implementação
3. Máquinas de estado utilizadas como referência para a implementação do código
4. Testes e visualização
   4.1 Visualização das formas de onda do caso de teste
5. Conclusões
6. Bibliografia

2. Decisões de Implementação

A partir da leitura do enunciado e das dúvidas no fórum da página da disciplina no Moodle foi decidido por implementar o circuito utilizando que 1 clock corresponde ao período de 1 segundo (0,5s em 1 e 0,5s em zero) e que não há especificação para o estado inicial do cronômetro antes do primeiro reset para o uso da cantora, implementando então opção sem especificar (com X até o reset) e especificando o início com um reset prévio antes dos testes em si.

Para isso, no código de testes foi adicionado o comando “#1” antes ddos testes para que após o primeiro reset (extra - inicial) a simulação aguarde 1 unidade de tempo antes de começar a computar a contagem, permitindo assim uma melhor visualização desde a primeira saída. Para equilíbrio do tempo da especificação foi reduzida uma unidade de tempo do período inicial também (de #50 para #49). A parte do código que realiza tais funções está exemplificada na Figura 1 abaixo.

```
13      always begin //Inicializando o clock de 1 segundo
14          #5 clk = ~clk; //a cada 0.5 seg
15      end
16
17      initial begin // Inicializações
18          $dumpfile("dump.vcd");
19          $dumpvars;
20          clk = 1;
21          reset = 1; //para começar do zero antes dos testes, e não ficar um valor X indeterminado
22          #1;
23          reset = 0;
24
25          //TESTES
26          #49;// Liga o contador por 5 segundos
27          reset = 1; //Aciona o reset por 1 segundo
28          $display("reset = 1");
29          #10;
30          reset = 0;
31          #120 //deixa o contador por 12 segundos
32          $finish; // Finaliza a simulação
33      end
```

Figura 1 - Código adicionando o reset no início

3. Máquinas de estado utilizadas como referência para a implementação do código

Uma parte fundamental da implementação do circuito foi o entendimento de suas transições de estados e, para isso, foi realizado o desenho do diagrama apresentado na imagem 2 abaixo, que conta de 0 a 9 de forma cíclica quando o reset está desacionado e caso o reset seja acionado, vai para 0000.

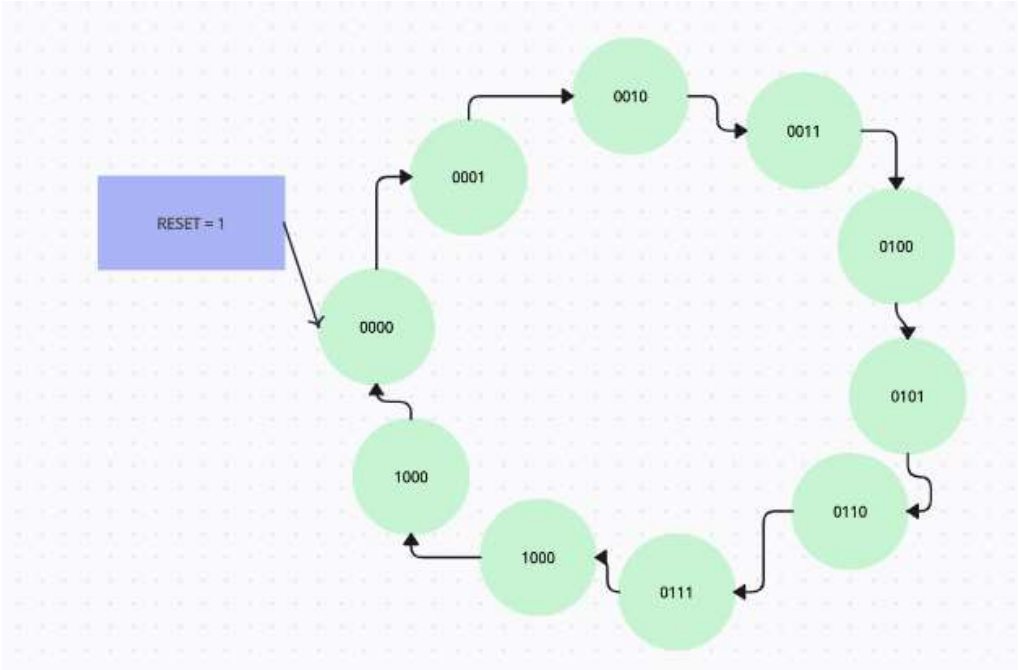

Figura 2 - Diagrama de estados

Também foi usado como referência a implementação de circuitos contadores em hardware, como a implementação em um exemplo retirado da internet e

adicionado abaixo na figura 3. Foi interpretado pelo enunciado que por não precisar do modelo estrutural esta parte não precisava ser propriamente implementada para a entrega do trabalho.

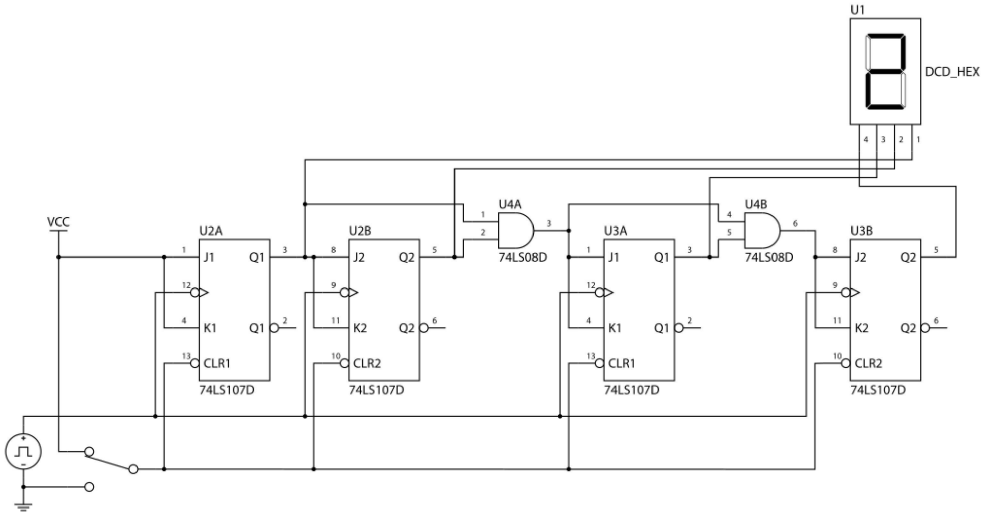

Figura 3 - Exemplo de circuito contador. Título original: "The circuit diagram for a 4-bit TTL counter, a type of state machine". Wikipedia

4. Testes e visualização

O caso de teste da especificação foi:

" 1. O contador é acionado durante 5 segundos,
2. Em sequência, o reset é acionado,
3. Por fim, o contador fica ligado pelos próximos 12 segundos."

e para isso foi utilizado como referência o código do testbench de semáforo apresentado na Figura 3 da especificação.

Além disso, foi acrescentado comandos de display dos estados atuais, para acompanhamento do funcionamento do circuito também pela saída de texto em tela, conforme exemplificado na Figura 4 abaixo.

```
VCD info: dumpfile dump.vcd opened for output.
ZERO
UM
DOIS
TRÊS
reset = 1
QUATRO
ZERO
UM
DOIS
TRÊS
QUATRO
CINCO
SEIS
SETE
OITO
NOVE
ZERO
UM
DOIS
```

Figura 4 - Saída em texto no Log do teste do circuito

4.1 Visualização das formas de onda do caso de teste

Complementando a saída em texto, a visualização das formas de onda do caso de teste estão dispostas abaixo nas Figuras 5 e 6, apresentando tanto a versão com início definido em zero pelo reset como sem a definição do estado inicial prévio

.

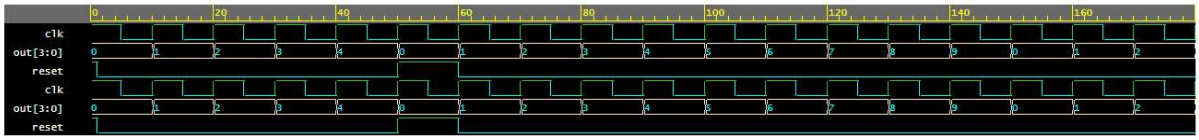

Figura 5 - Forma de onda do caso de teste utilizando o reset para que o contador se inicie em zero

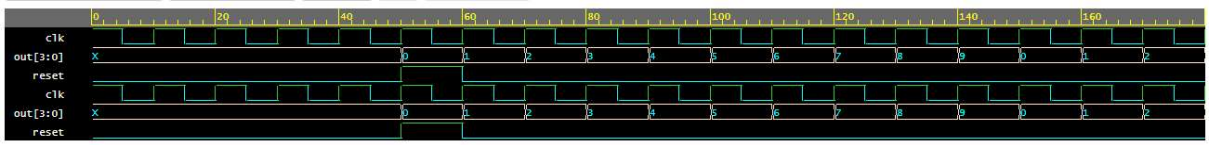

Figura 6 - forma de onda sem utilizar o reset ao início do teste

```
17      initial begin // Inicializações
18          $dumpfile("dump.vcd");
19          $dumpvars;
20          clk = 1;
21          //reset = 1; //para começar do zero antes dos testes, e não ficar um valor X indeterminado
22          //#1;
23          reset = 0;
24
25          //TESTES
26          #50;// Liga o contador por 5 segundos
```

Figura 7 - Exemplificação da parte do código que foi removida, em comparação com a Figura 1

5. Conclusão

A partir desta atividade prática para a resolução do problema que resolva o problema da necessidade de um cronômetro com saída de 4 bits e aprimorar meus conhecimentos em todas as etapas da implementação do circuito, aprimorando e praticando a linguagem Verilog.

Além disso, foi um ótimo exeŕicio de revsĩao juntamente com o aprendizado do novo conteudo apresentado pela disciplina de Introdução aos Sistemas Lógicos e durante sua modelagem e execução pude perceber minhas maiores dificuldades: uso de blocos “always” e de atribuições de valores no arquivo de design. O código implementado na plataforma também está disponível no link https://edaplayground.com/x/NNai.

6. Referências

Slides virtuais da disciplina de Introdução aos Sistemas Lógicos. Disponibilizado via moodle. Departamento de Ciência da Computação. Universidade Federal de Minas Gerais. Belo Horizonte.

Examples. EDA Playground. Disponível em <https://edaplayground.com/x/adiQ>. Acesso em 30 de outubro de 2023.

Criar um Diagrama de Estados. Creately. Disponível em <https://creately.com/pt/lp/software-de-diagrama-de-estados/>. Acesso em 28 de novembro de 2023

Finite-state machine. Wikipedia, Disponível em <https://en.wikipedia.org/wiki/Finite-state_machine>. Acesso em 28 de novembro de 2023