

Máquinas de Estados em VHDL

Versão 2022

INTRODUÇÃO

Esta experiência apresenta **máquinas de estados descritas em VHDL** e sua aplicação em um circuito digital simples. Ao final da experiência, os alunos terão conhecimento sobre o desenvolvimento de sistemas digitais mais complexos, compostos por fluxo de dados e unidade de controle.

OBJETIVO

Após a conclusão desta experiência, os seguintes tópicos deverão ser conhecidos pelos alunos:

- *Descrição de máquinas de estados em VHDL;*
- *Aplicação de máquinas de estados como unidade de controle de um circuito digital;*
- *Estudo de um circuito digital simples.*

1. PARTE EXPERIMENTAL

1.1. Projeto de uma Máquina de Estados em VHDL

Esta atividade tem como objetivo a familiarização com a descrição de uma máquina de estados em VHDL. Seu código fonte deve ser estudado, assim como o comportamento da máquina de estados descrita.

- Considere um diagrama de estados dado na Figura 1.
- Considere agora uma **descrição comportamental** em VHDL (`maquina_estados.vhd`) apresentada na Figura 2, que implementa a máquina de estados representada no diagrama da Figura 1. Estude o código fonte fornecido.

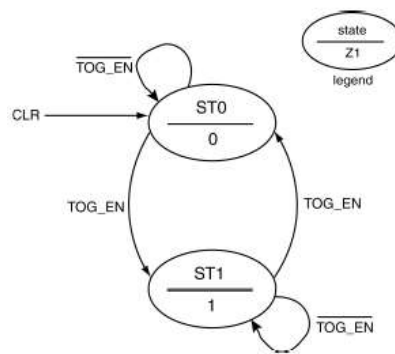
DICA: Sobre descrição de máquinas de estados em VHDL, consulte as páginas 630 a 632 do livro-texto (**Wakerly**) ou o capítulo 7 da referência **Free Range VHDL** ou o link https://balbertini.github.io/vhdl_fsm-pt_BR.html.

- Apresente um **pseudocódigo** (código genérico que imita uma linguagem de programação) que descreve o funcionamento da máquina de estados elaborado em b).
- Elabore um projeto na ferramenta Quartus e compile o código VHDL (Figura 2). Gere um **Diagrama de Transição de Estados** pela ferramenta (menu **Tools** -> **NetList Viewers** -> **State Machine Viewer**) e compare com o diagrama apresentado na Figura 1. Quais são as diferenças?

OBS: Para criar e compilar corretamente um código VHDL veja o Anexo **Tutorial para criação de circuitos digitais em VHDL no Quartus Prime 16.1**.

- Execute uma ou mais simulações para verificar o funcionamento apresentado. Anexe a carta de tempos no Planejamento.
- Submeta o arquivo QAR (`maquina_estados_txxby.qar`) do circuito junto com o Planejamento, onde xx é o número da turma e yy é o número da bancada do grupo.

Figura 1 – Um diagrama de estados.



Fonte: Mealy & Tappero (2016).

Figura 2 – Código fonte VHDL correspondente.

```

-- library declaration
library IEEE;
use IEEE.std_logic_1164.all;
-- entity
entity my_fsm1 is
    port ( TOG_EN : in  std_logic;
          CLK,CLR : in  std_logic;
          Z1 : out std_logic);
end my_fsm1;
-- architecture
architecture fsm1 of my_fsm1 is
    type state_type is (ST0,ST1);
    signal PS,NS : state_type;
begin
    sync_proc: process(CLK,NS,CLR)
    begin
        -- take care of the asynchronous input
        if (CLR = '1') then
            PS <= ST0;
        elsif (rising_edge(CLK)) then
            PS <= NS;
        end if;
    end process sync_proc;

    comb_proc: process(PS,TOG_EN)
    begin
        Z1 <= '0'; -- pre-assign output
        case PS is
            when ST0 => -- items regarding state ST0
                Z1 <= '0'; -- Moore output
                if (TOG_EN = '1') then NS <= ST1;
                else NS <= ST0;
                end if;
            when ST1 => -- items regarding state ST1
                Z1 <= '1'; -- Moore output
                if (TOG_EN = '1') then NS <= ST0;
                else NS <= ST1;
                end if;
            when others => -- the catch-all condition
                Z1 <= '0'; -- arbitrary; it should never
                NS <= ST0; -- make it to these two statements
        end case;
    end process comb_proc;
end fsm1;

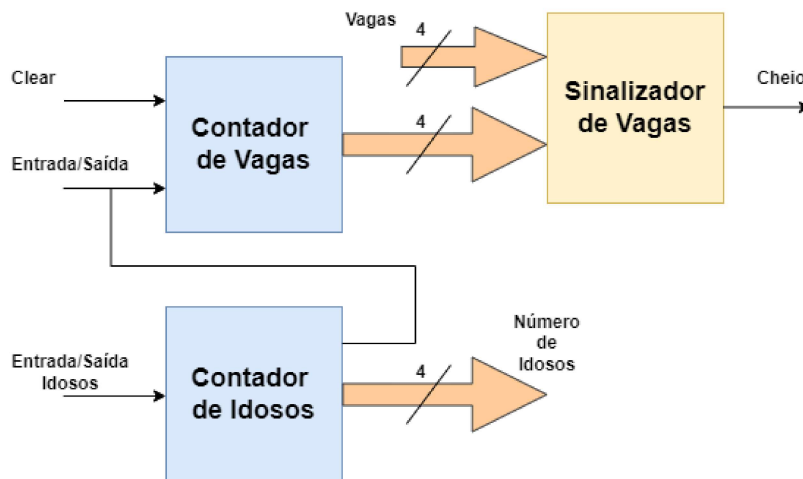
```

Fonte: Mealy & Tappero (2016).

1.2. Projeto de um Fluxo de Dados em VHDL

Modifique o projeto do circuito descrito na Exp. 5 - **Circuitos Digitais em VHDL** adicionando um contador que cuida da contagem do número de idosos que entraram no estacionamento, conforme apresentado na Figura 3.

Figura 3 – Diagrama de blocos do circuito.



Fonte: Autor.

- Implemente o circuito em VHDL. Insira o código no Planejamento.
- Execute a ferramenta RTL Viewer do Quartus. Verifique se o circuito está corretamente interconectado. Insira o Diagrama Lógico obtido no Planejamento.
- Simule o circuito completo na ferramenta Quartus. Insira a Carta de Tempos do circuito no Planejamento.
- Elabore uma Tabela de Testes do circuito com entradas, saídas e sinais intermediários. Insira a Tabela de Testes no Planejamento.
- Entregue o arquivo QAR (nome_projeto_txxbyy.qar) do circuito junto com o Planejamento, onde xx é o número da turma e yy é o número da bancada do grupo.

1.3. Projeto de uma Unidade de Controle em VHDL

Elabore uma Unidade de Controle para o Fluxo de Dados desenvolvido na Seção 1.2 que tenha as seguintes funções:

- Se existirem vagas, permitir a entrada de um veículo. Dar prioridade para a entrada de idosos.
- A cada entrada de veículo, incrementar o contador de vagas, se não atingiu o máximo de vagas.
- A cada saída de veículo, decrementar o contador de vagas (o sinal **Cheio** deve estar desligado).
- Se sair um veículo de idoso, o contador de idosos também deve ser decrementado.

- Elabore uma Máquina de Estados em VHDL para a Unidade de controle descrita na ferramenta Quartus. Insira o código no Planejamento.
- Realize uma simulação do módulo na ferramenta Quartus. Insira a Carta de Tempos no Planejamento.
DICA: Acrescente sinais de depuração, se algum resultado não for satisfatório. Corrija o módulo e simule novamente.
- Elabore uma Tabela de Testes do circuito com entradas, saídas e sinais intermediários. Insira a Tabela de Testes no Planejamento.
- Entregue o arquivo QAR (nome_projeto_txxbyy.qar) do circuito junto com o Planejamento, onde xx é o número da turma e yy é o número da bancada do grupo.

1.4. Implementação do Sistema Digital

Nesta atividade, o projeto do sistema digital em VHDL deverá ser programado na placa de desenvolvimento FPGA DE0-CV. Para isto, precisam-se dos arquivos `nome_projeto_txbyy.qar` das partes 1 (Fluxo de Dados) e 2 (Unidade de Controle) do circuito.

- a) Abra o arquivo do Fluxo de Dados.
- b) Compile o circuito e programe a FPGA DE0-CV. Faça uma designação de sinais adequada (usando **chaves** e **LEDs**) para observar o funcionamento do circuito (**veja o Anexo FPGA DE0-CV – Pintable**).

OBS: Não usem o botão da placa FPGA para o *Clock* do circuito. Coloquem a entrada em um pino da interface GPIO. Usaremos o dispositivo Analog Discovery para simular o botão da placa.
- c) Execute os testes planejados na Tabela de Testes. Caso algum teste não seja bem sucedido, insira sinais de depuração e faça novamente os testes até que o problema seja sanado. Anote os resultados no Relatório.
- d) Abra o arquivo da Unidade de Controle.
- e) Execute a ferramenta **State Machine Viewer** (menu **Tools > Netlist Viewers > State Machine viewer**) e analise a saída obtida (o que falta, está completa?). Insira a figura no Relatório.
- f) Compile o circuito e programe a FPGA DE0-CV. Faça uma designação de sinais adequada (usando **chaves** e **LEDs**) para observar o funcionamento do circuito (**veja o Anexo FPGA DE0-CV – Pintable**).

OBS: Não usem o botão da placa FPGA para o *Clock* do circuito. Coloquem a entrada em um pino da interface GPIO. Usaremos o dispositivo Analog Discovery para simular o botão da placa.
- g) Execute os testes planejados nas Tabelas de Testes. Caso algum teste não seja bem sucedido, insira sinais de depuração e faça novamente os testes até que o problema seja sanado. Anote os resultados no Relatório.
- h) Integre o Fluxo de Dados à Unidade de Controle.
- i) Realize os testes no circuito completo. Anote os resultados no Relatório.

1.5. Desafio (Opcional)

O professor irá propor um desafio sobre esta experiência.

2. BIBLIOGRAFIA

- ALMEIDA, F.V. de; SATO, L.M.; MIDORIKAWA, E.T. **Tutorial para criação de circuitos digitais em VHDL no Quartus Prime 16.1**. Apostila de Laboratório Digital. Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica da USP. Edição de 2017.
- HIRAMA, K.; KINOSHITA, J. **Circuitos Digitais em VHDL**. Apostila do Laboratório Digital A. Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica da USP. Edição de 2021.
- HIRAMA, K. **Manual de Acesso Remoto e Uso da Bancada do Laboratório Digital**. Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica da USP. Versão 1.1, 2020.
- ALTERA. **DE0-CV User Manual**. 2015.
- ALTERA. **Quartus Prime Introduction Using VHDL Designs**. 2016.
- ALTERA. **Quartus Prime Introduction to Simulation of VHDL Designs**. 2016.
- D'AMORE, R. **VHDL - descrição e síntese de circuitos digitais**. 2ª edição, LTC, 2012.
- MEALY, B.; TAPPERO, F. **Free Range VHDL - The no-frills guide to writing powerful code for your digital implementations**. Free Range Factory, Janeiro de 2018 (v.1.21).
- TOCCI, R. J.; WIDMER, N.S.; MOSS, G.L. **Sistemas Digitais: Princípios e Aplicações**. Prentice-Hall, 11ª ed., 2011.
- WAKERLY, John F. **Digital Design Principles & Practices**. 4th edition, Prentice Hall, 2006.

3. RECURSOS NECESSÁRIOS

- 1 Computador pessoal.
- 1 Placa de desenvolvimento FPGA DE0-CV com o dispositivo Cyclone V 5CEBA4F23C7N
- 1 Dispositivo Analog Discovery.
- 1 Ferramenta Intel Quartus Prime 16.1.
- 1 Ferramenta Waveforms do Analog Discovery.

ANEXOS

- Tutorial para Criação de Circuitos Digitais em VHDL
- FPGA DE0-CV - Pintable

Histórico de Revisões

Prof. Edson T. Midorikawa – versão 2019

Profs. Kechi Hirama, Jorge Rady de Almeida Júnior, Sérgio Roberto de Mello Canovas – versão 2020

Profs. Kechi Hirama, Anarosa A. F. Brandão, Glauber de Bona – versão 2021

Profs. Kechi Hirama, Jorge Rady de Almeida Júnior, Renan Cerqueira Afonso Alves – versão 2022