



Disciplina: Laboratório de Sistemas Digitais

Professor: Ricardo de Oliveira Duarte

Estudantes: Igor Braga de Lima, Matheus Vinícius Freitas Oliveira dos Santos e Stéphanie Pereira Barbosa

Turma: PN5

Guia de aula: 08

ATIVIDADE TEÓRICA

1) Com é que definimos que um modelo de implementação de FSMs é um modelo de Moore?

Pode-se definir uma implementação de FSM como um modelo de Moore quando a saída só depende dos estados atuais da FSM.

2) Com é que definimos que um modelo de implementação de FSMs é um modelo de Mealy?

Em uma implementação de FSM que usa modelo Mealy, a saída não depende apenas dos estados atuais da FSM, mas também das entradas atuais.

3) Qual o procedimento mais direto e simples de implementação de FSMs em VHDL?

Modularizar a FSM em duas partes, uma para lidar com a parte síncrona e outra para lidar com a parte combinacional.

4) De que forma o tipo de dados enumerado pode ser útil na descrição de FSMs em VHDL?

O tipo enumerado é útil, pois pode proporcionar uma melhor descrição dos estados da FSM, possibilitando atribuir nomes aos estados.

5) Na página 93 do livro tem um erro sutil. Qual é esse erro?

No código PS e NS são definidos como *signal*, mas no texto é afirmado que PS e NS são *variables*.

```
-- architecture
architecture fsm1 of my_fsm1 is
    type state_type is (ST0,ST1);
    signal PS,NS : state_type;
begin
```

6) Por que devemos sempre usar a cláusula *when others* sempre que usar uma declaração *case* em uma descrição de um circuito em VHDL?

Devemos usar a título de completude das possíveis saídas que o circuito pode

gerar, pois apesar de representar uma boa prática de codificação, o código nunca deve chegar nesse ponto.

7) Como é que eu devo implementar uma saída do modelo Moore dentro de uma declaração *case* do *process* combinacional?

A saída deve ser uma função única e exclusivamente do estado atual. Por exemplo, a saída vai estar dentro da cláusula *when*, mas fora da condicional *if*, a qual depende do valor da(s) entrada(s).

8) Por que eu devo atribuir um valor inicial a(s) saída(s) da FSM em um *process* combinacional logo após a cláusula *begin* do *process*?

Para prevenir a geração de um *latch* inesperado.

9) Atribuir valores às saídas da máquina no início do *process* interferem ou não no valor lógico produzido dentro do *process* que será refletido na saída da máquina? Por quê?

Não, pois apenas a atribuição final da(s) saída(s) é considerada quando o *process* termina.

10) O que devo fazer para ter como saída do meu circuito o acesso ao estado atual da FSM?

É preciso declarar um sinal de saída intermediário, o qual irá armazenar o estado da máquina.

11) Como é que eu devo implementar uma saída do modelo Mealy dentro de uma declaração *case* do *process* combinacional?

A saída do modelo deve ser implementada a partir de uma declaração de *if* em que a condição analisada é a entrada.

12) Podemos descrever uma FSM híbrida, isto é uma máquina que tem saídas Moore e saídas Mealy?

Sim.

13) Em que situação eu devo descrever uma FSM em VHDL com o método *one-hot encoding* em contrapartida ao método de codificação binária de estados?

Quando a FSM possuir poucos estados, de modo que cada estado corresponda a um bit do registrador de estados.

14) Quais as vantagens e desvantagens do método de descrição *one-hot encoding*?

A vantagem é que ele é um método mais direto que o método de codificação binária de estados, pois associamos cada estado a um bit do registrador de estados. Por sua vez, a desvantagem é que a FSM gerada tem um registrador

de estados muito maior que a FSM gerada pelo método de codificação binária de estados.

ATIVIDADE PRÁTICA

- 1) Faça o Exercício 2 da página 108 do livro.
- 2) Escreva um *testbench* para o sistema em VHDL que você implementou no item anterior.
- 3) Compile, simule e verifique o comportamento do seu circuito.
- 4) Faça o Exercício 8 da página 114 do livro.
- 5) Escreva um *testbench* para o sistema em VHDL que você implementou no item anterior.
- 6) Compile, simule e verifique o comportamento do seu circuito.
- 7) Refaça o Exercício 2 da página 108 do livro, agora com o método de descrição *one-hot encoding*.
- 8) Escreva um *testbench* para o sistema em VHDL que você implementou no item anterior.
- 9) Compile, simule e verifique o comportamento do seu circuito.