



**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:** 07

---

## ATIVIDADE TEÓRICA

**1) Em um modelo de descrição comportamental, o que deve ser colocado na lista de sensibilidade de um *process* de um circuito sequencial?**

O *signal clock*, *signals* de *reset* e *set* assíncronos, para casos em que deseje estabelecer.

**2) Para que serve a função *rising\_edge*( ) em VHDL?**

É usado no *if* para indicar se há uma subida de *clock*, proporcionando realizar alterações da saída do circuito condicionadas à subida de *clock*.

**3) Como é descrita a função *rising\_edge*( ) em VHDL?**

Utilizando "*event*" é possível estabelecer a seguinte comparação para detectar a subida de clock:

`CLK'event and CLK='1'`

**4) Qual a função dual da *rising\_edge*( ) em VHDL?**

É a função que provoca uma mudança na borda de descida do *clock* de um circuito: *falling\_edge*( ).

**5) Qual a causa de uma ferramenta de síntese, após o processo de compilação, gerar um *warning* dizendo que a descrição do seu circuito gerou um *latch*?**

A causa é que não foi fornecido explicitamente um estado de saída para todas as condições de entrada possíveis. Por conta disso, o circuito precisará lembrar o estado das saídas anteriores para poder fornecer uma saída no caso em que não foi explicitamente listado a condição de entrada atual.

**6) O que difere em uma descrição de um circuito sequencial, uma entrada síncrona de uma entrada assíncrona em VHDL?**

Na entrada síncrona as entradas dependem da borda de subida de *clock*. Ao passo que, na entrada assíncrona não há essa dependência.

**7) Com que finalidade deve ser usado sinais intermediários dentro de um *process* em VHDL?**

Quando precisamos usá-los dos dois lados de um operador de atribuição dentro do *process*.

**8) Por que não devemos usar *buffers* no lugar de sinais intermediários em descrições de circuitos sequenciais VHDL?**

Porque os *buffers* são dependentes do tipo da ferramenta de síntese.

**9) Por que devemos descrever circuitos sequenciais no modelo *behavioral* ao invés do modelo *data-flow*?**

Apesar de ser possível descrever os circuitos seqüenciais em ambos os modelos, usamos o *behavioral* porque ele facilita a descrição do comportamento do circuito sequencial e o modelo *data\_flow* torna este tipo de descrição muito mais trabalhosa.

## **ATIVIDADE PRÁTICA**

**1) Faça o Exercício 1 ou o Exercício 2 da página 86 do livro.**

**2) Escreva um *testbench* para o sistema em VHDL que você implementou no item anterior, tomando como base de exemplo o *testbench* que o professor disponibilizou a você nessa atividade.**

**3) Compile, simule e verifique o comportamento do seu circuito.**

**4) Faça o Exercício 6 da página 87 do livro.**

**5) Escreva um *testbench* para o sistema em VHDL que você implementou no item anterior, tomando como base de exemplo o *testbench* que o professor disponibilizou a você nessa atividade.**

**6) Compile, simule e verifique o comportamento do seu circuito.**

**7) Abra o arquivo VHDL: *mean\_4\_clocks.vhd*. Nos comentários desse arquivo, bem como no arquivo *README*, está a descrição do comportamento esperado do modelo. O exemplo 4.11, da página 192 da referência [2] do plano de aulas, também descreve o comportamento esperado.**

**8) Crie um *testbench* para simular o modelo.**

**9) Identifique *bugs* no modelo. Existem dois erros que impedem que o sistema funcione de forma correta. Descreva detalhadamente os erros identificados e como eles afetam o comportamento do modelo.**

**Dica 1: O primeiro erro tem relação a ordem de atribuições sequenciais. Altere a ordem ou transforme em atribuições concorrentes;**

**Dica 2: O segundo erro tem relação com a precisão no cálculo da divisão por dois.**

**10) Corrija os erros e verifique o funcionamento correto usando o *testbench*.**