

Sistemas Digitais

Aula 06 - Descrevendo e Simulando Circuito em VHDL

Apresentação

Nesta aula, você usará os conceitos vistos nas aulas de VHDL. Será necessário o conhecimento do software Quartus II para realizar a compilação e a simulação do circuito. Aqui você irá aplicar os conceitos de “process” e “if-then-else” na prática, verificando a sequencialidade da execução.



Vídeo 01 - Apresentação

Objetivos

Ao final das atividades previstas para esta aula, você será capaz de:

- Diferenciar trechos de códigos concorrentes e sequenciais;
- Utilizar o comando “process” e “if-then-else”;
- Compilar e simular no software Quartus II.

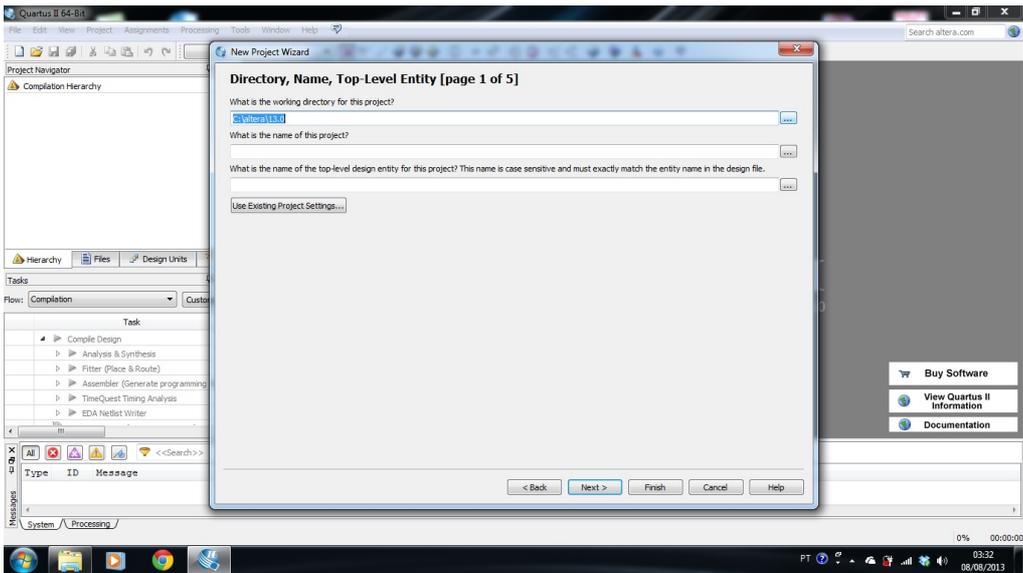
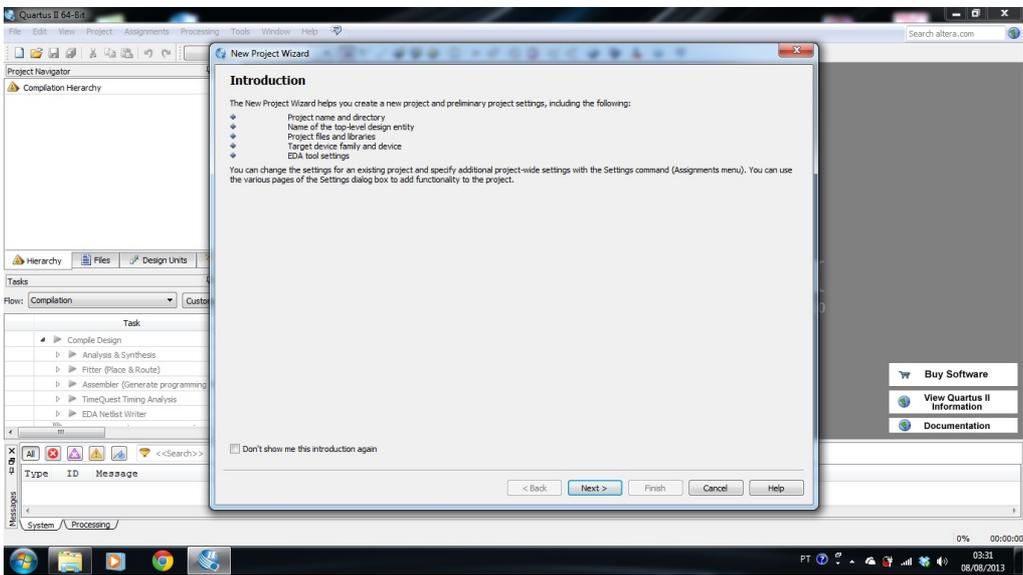
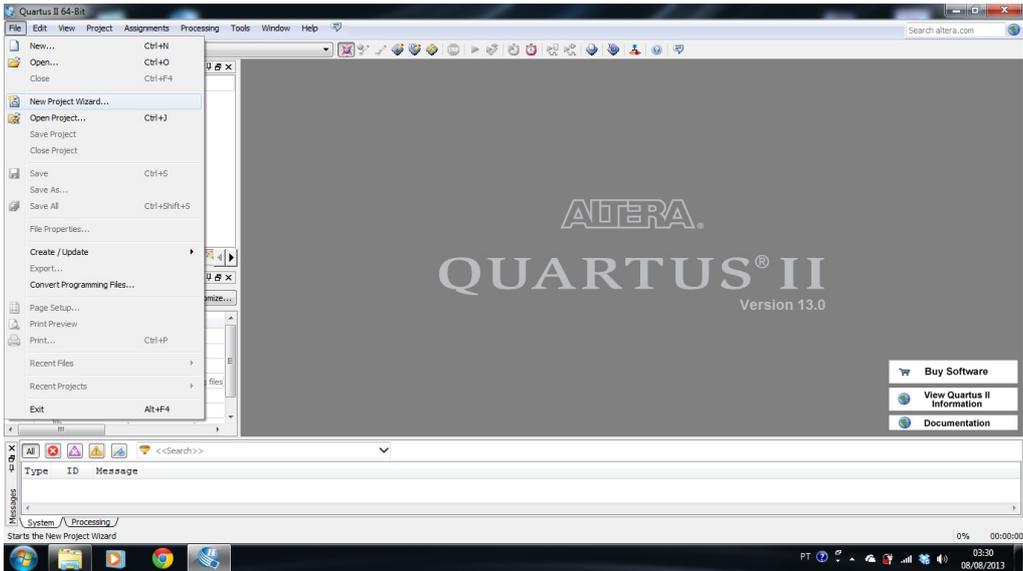
Relembrando: Criando um Projeto no Quartus II

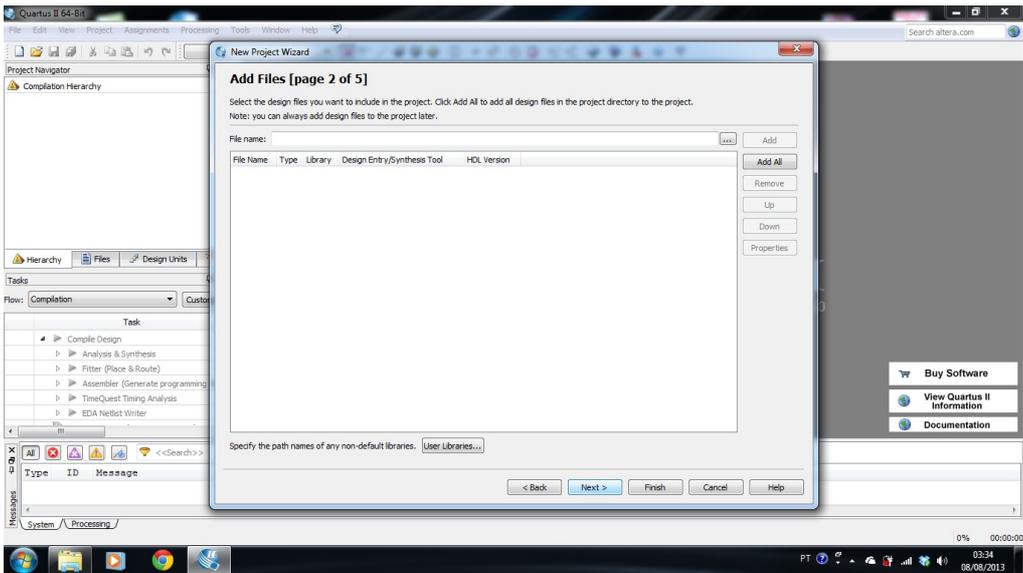
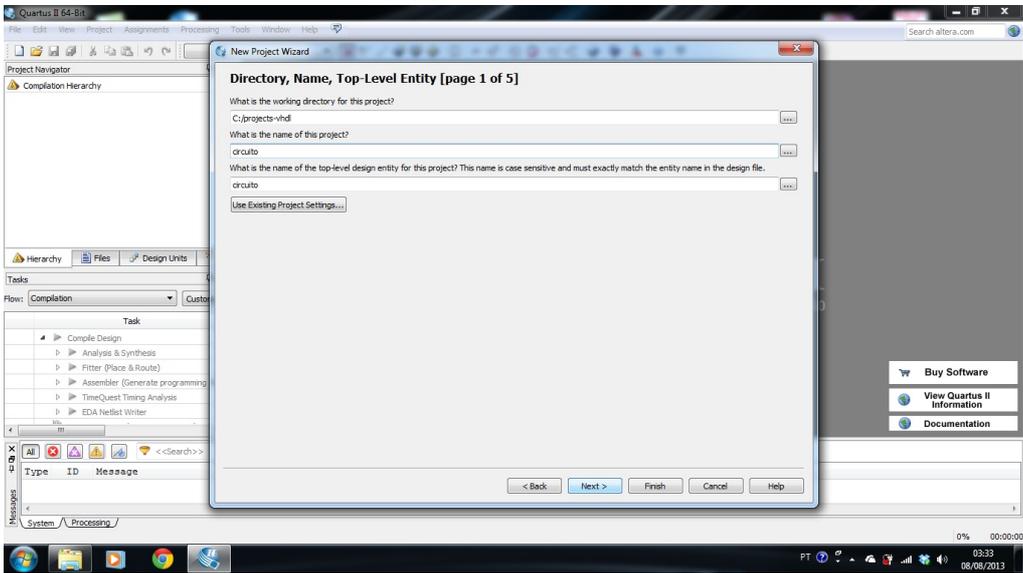
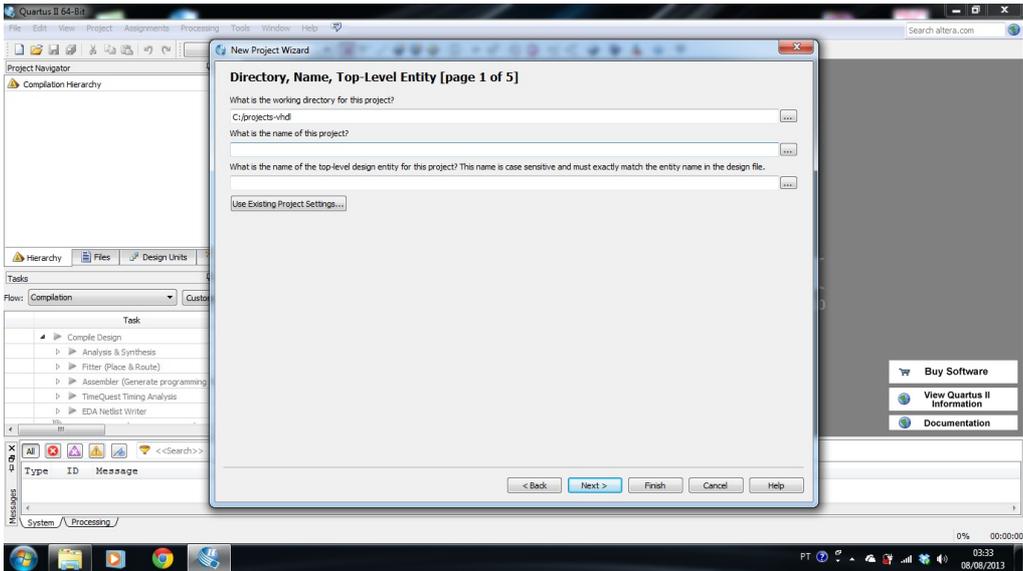
Inicialmente, vamos lembrar como criar um projeto no software Quartus II. Veja no slideshow abaixo.

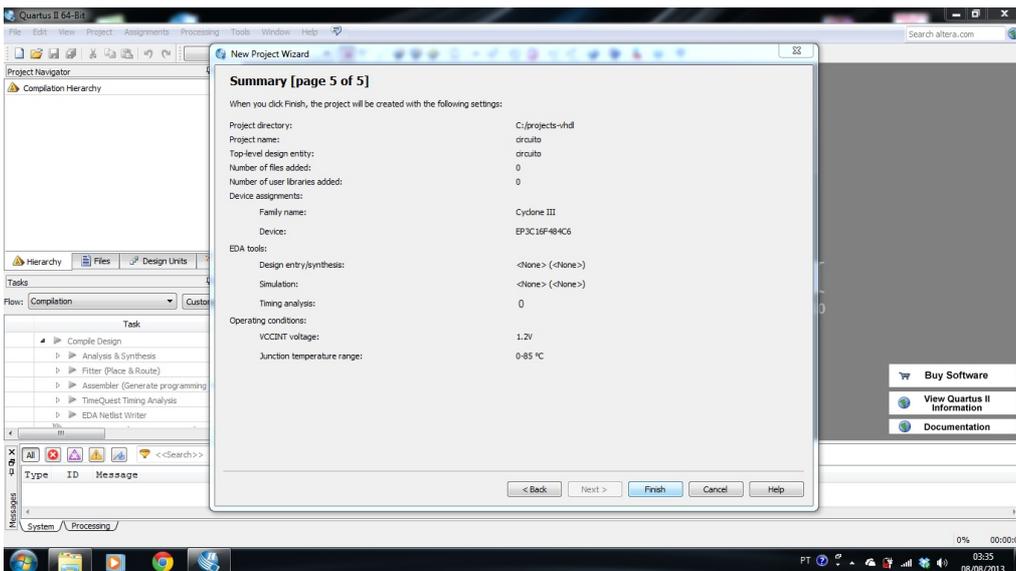
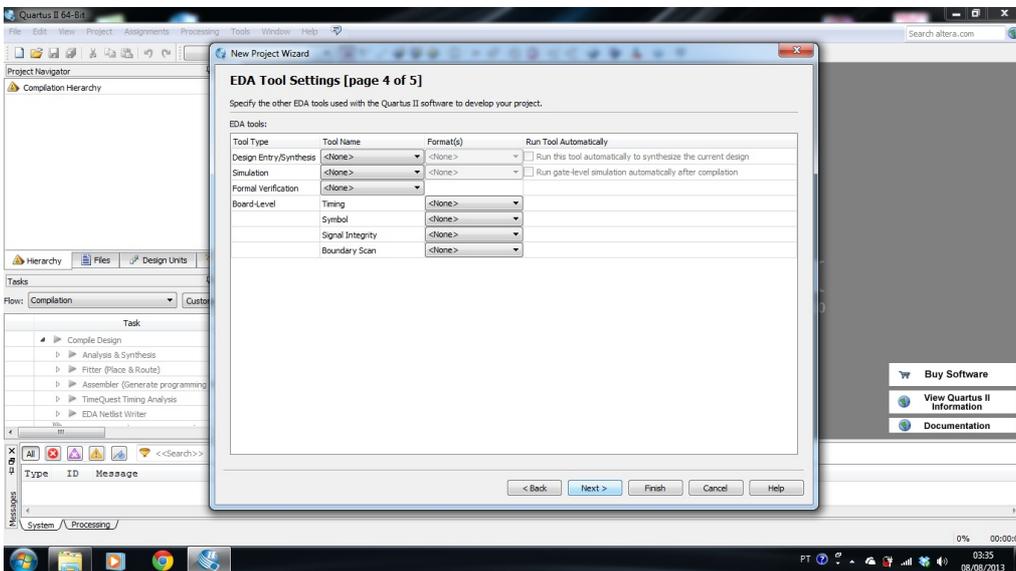
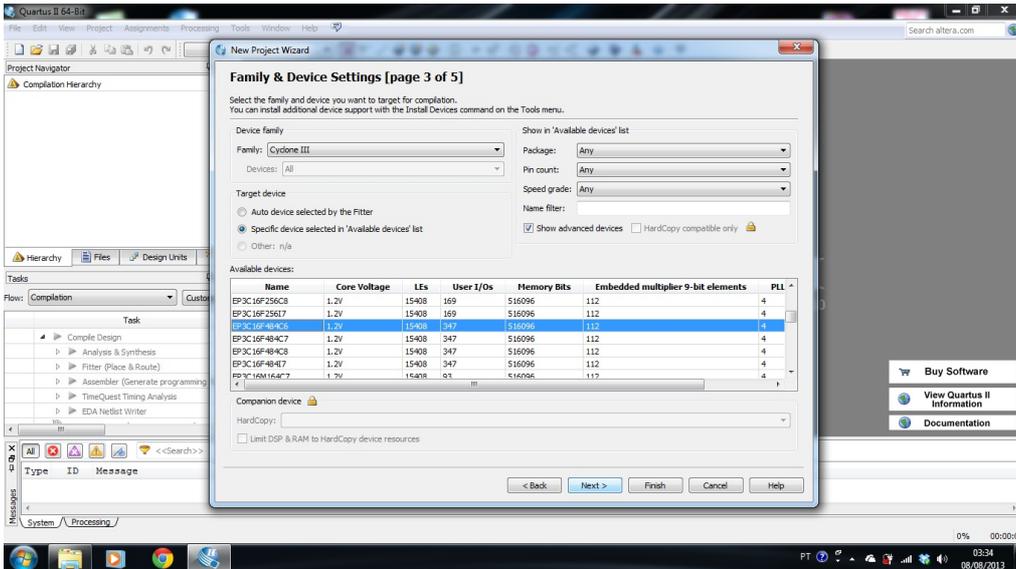
Lembretes:

- Escolha um diretório apropriado para armazenar o projeto [Passo 1 do “New Project Wizard”];
- O nome do projeto deve ser o mesmo nome da entidade principal [Passo 1 do “New Project Wizard”];
- O dispositivo (Device) o qual vamos trabalhar futuramente é o EP3C16F484C6, que é o chip FPGA da nossa placa DE0. Então, adote-o desde já [Passo 3 do “New Project Wizard”];











Vídeo 02 - Atribuições de Sinais e Concorrentes e Sequenciais



Vídeo 03



Vídeo 04

Analizando a Tabela Verdade do Circuito

Nesta aula, você utilizará o circuito mostrado na Figura 1 e verá como construí-lo utilizando instruções de seleção.

Figura 01 - Circuito Exemplo

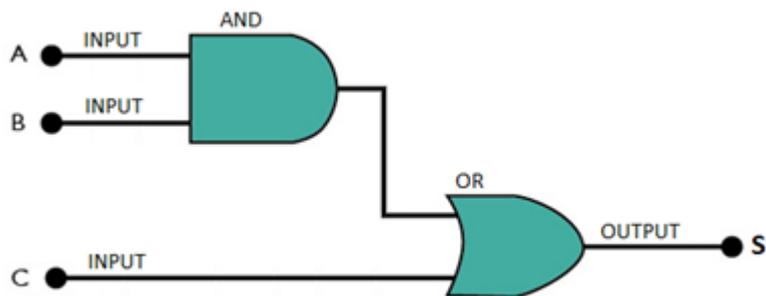


Figura 01: Circuito Exemplo

Precisamos retirar a tabela verdade do circuito para, então, descrever o mesmo. Como temos um circuito com três entradas, a tabela verdade (mostrada abaixo) terá oito linhas.

A	B	C	S
0	0	0	-
0	0	1	-
0	1	0	-
0	1	1	-
1	0	0	-
1	0	1	-
1	1	0	-
1	1	1	-

Tabela 1 - Tabela Verdade do Circuito Exemplo

Atividade 01

1. Complete a Tabela 1 a partir do circuito da Figura 1.

Soma de Produtos e Produto das Somas

Conseguiu completar a tabela verdade facilmente? Se não, faça novamente e retire as dúvidas com o seu tutor. A construção de uma tabela verdade pode ser considerada "o primeiro passo" para você conseguir "andar" no mundo da eletrônica digital.

Vamos agora focar nas condições em que a saída do circuito é “verdadeira” ou “nível lógico 1”. Esse tipo de análise por ser realizado a partir das equações SOP (sum-of-products, ou soma de produtos em português) e POS (product of sums, ou produto de somas em português), como exemplificado nas equações abaixo.

$$Y = \overline{M}N + \overline{M}\overline{N} + MN$$

(Soma de Produtos)

$$Y = (M + \overline{N}).(\overline{M} + N)$$

(Produto de Somas)

Nesta aula prática, vamos utilizar a equação soma de produtos, considerando a “soma” de todas as condições de verdadeiro, e cada condição sendo descrita em forma de “produto”. Entendeu?

Se não, então vamos exemplificar! Considere três eventos, A, B e C, relacionados pela tabela 2.

A	B	C	Resultado
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0

A	B	C	Resultado
1	1	1	0

Tabela 2 - Tabela Verdade Exemplo

Verifique que para o resultado ser “verdadeiro” são possíveis duas condições, ($A = 0; B = 0; C = 1$) ou ($A = 1; B = 0; C = 0$), sendo operação “OU” representada pela soma. Para a primeira condição é necessário que $A = 0$ e $B = 0$ e $C = 1$, assim como na segunda é necessário que $A = 1$ e $B = 0$ e $C = 0$, as operações “E” representando o produto.

Neste método, quando uma variável é igual à '0', utilizamos esta variável negada (barrada). Se ela for igual a '1', ela não será barrada. Assim, $A = 0$ e $B = 0$ e $C = 1$ pode ser escrito como $\overline{A} \overline{B} C$, enquanto que $A=1$ e $B=0$ e $C=0$ pode ser escrito como $A \overline{B} \overline{C}$. Portanto, a expressão que descreve a tabela verdade acima é:

$$Resultado = \overline{A} \overline{B} C + A \overline{B} \overline{C}$$

Voltando para a tabela verdade do circuito trabalhado na aula, temos quatro condições em que a saída do circuito é “verdadeira”.

Atividade 02

1. Descreva a expressão booleana do nosso circuito, mostrada na Figura 1, através da tabela verdade da Atividade 01 e utilizando soma de produtos.

Descrevendo o Circuito Utilizando a Estrutura “Process”

Se a sua expressão booleana, resolvida na tarefa 2, for equivalente à equação abaixo, você está correto. Lembrando que aqui estamos admitindo apenas equações de soma de produtos.

$$S = \overline{A} \overline{B} C + \overline{A} B C + A \overline{B} C + A B \overline{C} + A B C$$

Como vocês viram, existem várias instruções de atribuição de sinais. Nesta aula, vamos utilizar instruções de atribuição de sinais de seleção, ou melhor, o famoso e conhecido IF-THEN-ELSE, estrutura que vocês trabalharam em outras disciplinas.

Sabemos que o VHDL executa seus comandos de forma concorrente. Para que um conjunto de instruções seja executado de forma sequencial é necessário utilizar a estrutura especial chamada de processos (process). Logo, para as instruções IF-THEN-ELSE funcionarem de forma correta é necessário colocá-las dentro de um "process".

Agora vamos descrever parte da expressão booleana utilizando o IF-THEN-ELSE. Veja que cada produto é uma condição de seleção. O trecho de código abaixo mostra a implementação da condição $\overline{A} \overline{B} C$. Tente complementar o restante do código com as demais parcelas da expressão lógica. Lembre-se, ao final, que $S = 0$ para os demais casos que não os explícitos nesta equação.

```

1 if ((A='0') and (B='0') and (C='1')) then
2   s <= '1';
3   elsif (...)
4   ...
5 end if;
```

Atividade 03

Conseguimos montar a expressão lógica da tabela verdade, descrita como:

$$S = \overline{A} \overline{B} C + \overline{A} B C + A \overline{B} C + A B \overline{C} + A B C$$

Vimos também o início do código para $\overline{A} \overline{B} C$. Vamos terminar o código?

1. Crie um projeto novo. A entidade terá 3 entradas (A, B e C) e 1 saída (S), todos do tipo std_logic.
2. Crie a arquitetura. Dentro dela, crie um *process*. É dentro do *process* que parte do código que vimos anteriormente deverá ser escrito.

3. Dentro do *process*, escreva uma sequência de *if* e *elsif* que expresse a equação descrita acima. Lembre-se que a saída *S* será 1 para os casos descritos na equação ($\overline{A} \overline{B} C, \overline{A} B C, \dots, A B C$) e será 0 para todos os demais casos.
4. Compile o projeto.

Compilando e Simulando

Ocorreu algum erro na compilação? Se não, parabéns! Você realmente estudou o assunto. Se sim, citaremos alguns dos principais erros na programação em VHDL (na lista abaixo) para que verifique se seu erro pertence a algum desses citados.

- O projeto, o arquivo com extensão “vhd” e a entidade têm o mesmo nome?
- Foram colocados ponto-e-vírgula nos lugares necessários? Como exemplo, no final de instruções, final da entidade, final da arquitetura etc.
- Foram colocadas as variáveis necessárias nos argumentos do “process”?
- Foram colocados o “begin” e o “end” do “process”?

Caso o seu erro não tenha sido reportado aqui, leia novamente a aula e caso persista o erro, procure a tutoria ou a monitoria.

Atividade 04

1. Simule o circuito no *Simulation Waveform Editor* utilizando valores aleatórios. Verifique se as saídas estão corretas de acordo com a entrada.

A simulação foi de acordo com o que você esperava? Qualquer erro, você deve refazer este roteiro.

Com os conhecimentos adquiridos até agora, nós já conhecemos as principais portas lógicas e sabemos montar sua tabela verdade. Também sabemos escrever as expressões lógicas a partir de qualquer tabela verdade. Em VHDL conseguimos descrever estes circuitos em código utilizando os elementos básicos de programação (estruturas de decisão, loops). Agora é com você! Muitas das questões online, desafio e exercícios presenciais utilizam programação em VHDL. Pratique bastante no VHDL utilizando todos o que você aprendeu.

Resumo

Nesta aula prática, você reforçou os seus conhecimentos sobre o software Quartus II, aplicou conceitos de extração de tabela verdade e criação de expressões booleanas. Também foram utilizados os conceitos de “process” e “IF-THEN-ELSE”, vistos na aula anterior a esta.

Referências

COSTA, Cesar da. **Projetos de Circuitos Digitais com FPGA**. Editora Érica, 2009.

PEDRONI, Volnei. **Eletrônica Digital moderna e VHDL**. Rio de Janeiro: Elsevier, 2010.