

PSI-3451 Projeto de CI Lógicos Integrados (2018)

Aula 6

A parte prática da aula 6 pretende reforçar com o(a) aluno(a) o contato com a construção *generate*, em particular, com o modelo estrutural.

Tenha certeza que sabe responder as perguntas listadas abaixo após a observação dos arquivos ou dos resultados de simulações.

PARTE PRÁTICA

São fornecidos 2 arquivos VHDL de entidades, um somador *ripple-carry* e uma unidade lógica aritmética; o primeiro terá duas versões de simulação enquanto que o segundo terá apenas uma. No total serão analisados e simulados 3 arquivos VHDL.

Em cada experimento, o respectivo arquivo VHDL será submetido à sequência: **captura-compilação-simulação**, a mesma utilizada nas aulas anteriores.

Preparação das Pastas e Arquivos para Simulação

- Ligar o computador e entrar no ambiente **Windows**.
- Utilizando o **Windows Explorer** acesse a sua área de trabalho na unidade de rede **X**.
- Na sua área de trabalho crie uma pasta **X:\psi3451\aula_6** para armazenar os resultados desta prática. Como serão realizadas 3 simulações, recomendamos que sejam criadas pastas separadas para cada uma das simulações com o objetivo de salvar adequadamente os resultados. Por exemplo, crie as pastas:
X:\psi3451\aula_6\rca_2 (para o arquivo *rc_adder_2* - entidade *rc_adder* e o testbench)
X:\psi3451\aula_6\rca_3 (para o arquivo *rc_adder_3* - entidade *rc_adder*)
X:\psi3451\aula_6\alu_1 (para o arquivo *alu_1* - entidade *alu*)
- Ainda com o **Windows Explorer**, selecione **Rede=> NEWSERVERLAB => psi3451 => aula_6** e copie os arquivos VHDL das sub-pastas para as correspondentes, **X:\psi3451\aula_6\rca_2**, **X:\psi3451\aula_6\rca_3** e **X:\psi3451\aula_6\alu_1**.

1) Captura, compilação e simulação do somador *ripple_carry* no modelo estrutural

- Do projeto a ser verificado (DUV):
 - a) Abra o arquivo *rc_adder_2.vhd* da pasta **X:\psi3451\aula_6\rca_2**. Ele está codificado no modelo VHDL estrutural (*structural*).
 - b) Analise toda a descrição, linha a linha. Tenha certeza que tenha encontrado e entendido os itens seguintes:
 - a declaração do componente *full_adder_1*
 - os parâmetros *generic*

- a construção *generate*
 - as instanciações e os nomes das instâncias
 - *port map*
 - a correspondência do *rc_adder* com os sinais de E/S apresentados na apostila de conceitos.
- Do gerador de estímulos:
- c) Abra o arquivo *stimuli_rc_adder.vhd* da pasta **X:\psi3451\aula_6\rca_2**.
- d) Analise toda a descrição, linha a linha. Tenha certeza que tenha encontrado e entendido os itens seguintes:
- sendo o gerador de estímulos para o módulo *num_gen_direct*, a compatibilidade entre os portos dos dois módulos
 - o procedimento de geração de vetores de teste e a sua chamada na arquitetura
 - o processo de simulação (*process*) sem a lista de sensibilidade (portanto, com *wait*)
 - a construção *wait*
- Do *testbench* (topo):
- e) Abra o arquivo *testbench_rc_adder.vhd* da pasta **X:\psi3451\aula_6\rca_2**. Ele está codificado no modelo VHDL estrutural.
- f) Analise toda a descrição, linha a linha. Tenha certeza que tenha encontrado e entendido os itens seguintes:
- a instanciação do módulo *stimuli_rc_adder*
 - a instanciação do módulo *rc_adder*
 - a conexão entre os dois módulos
- g) Modifique o valor default do parâmetro *generic (width)* para 8
- Simulando:
- h) No Modelsim, crie a biblioteca *work* na pasta *rca_2*.
- i) Compile inicialmente o arquivo *full_adder_1.vhd* (da aula 2) contendo a entidade *full_adder_1(dataflow)*.
- j) Compile em seguida o arquivo *rc_adder_2.vhd* (entidade *rc_adder*), depois do testbench em padrão *bottom-up*.

Perguntas: todas as entidades aparecem na biblioteca *work*?

- k) Faça a simulação da entidade *testbench_rc_adder* (cf. descrito em exemplos trabalhados na práticas anteriores).

Recomendação: adicione estímulos ao testbench e sinais internos no Wave a fim de para verificar:

- o funcionamento do *ripple carry adder*
- teste casos de propagação de *carry-out* e de *overflow*

Guarde os resultados do Wave para futuras referências e comparações.

Perguntas: seguindo as recomendações, a simulação mostrou o comportamento esperado do circuito como descrito na apostila de conceitos? É o mesmo resultado da simulação do item 5) da Aula 2?

2) Captura, compilação e simulação da unidade lógica aritmética

É objetivo desta seção incorporar a entidade *rc_adder_2* dentro da unidade lógica aritmética como um componente e simular a entidade *ALU*. O circuito deve estar funcional, de acordo com a descrição da apostila de conceitos, pois será usada no projeto final do *Snake*.

- a) Abra o arquivo *alu_1.vhd* para a pasta **X:\psi3451\aula_6\alu_1**.
- b) Analise toda a descrição, linha a linha. Tenha certeza que tenha encontrado e entendido os itens seguintes:
 - instanciação do *rc_adder_2*
 - a inferência dos dois muxes como ilustrado na apostila de conceitos
- c) Copie os arquivos de testbench de **X:\psi3451\aula_6\rca_2** para **X:\psi3451\aula_6\alu_1**. Modifique-os de tal forma a ficarem compatíveis com as entradas da ULA (dados e controle).
- d) No Modelsim, crie a biblioteca *work* na pasta *alu_1*
- e) Compile, em sequência, primeiro o arquivo *full_adder_1.vhd (dataflow)* e depois o *rc_adder_2.vhd* além dos arquivos de *testbench*.
- f) Compile agora o arquivo de *testbench* e faça a simulação.

Recomendação: crie os estímulos para verificar:

- o funcionamento do circuito de acordo com o descrito na apostila de conceitos (todas as condições)
- verifique as várias condições de funcionamento da *ALU*

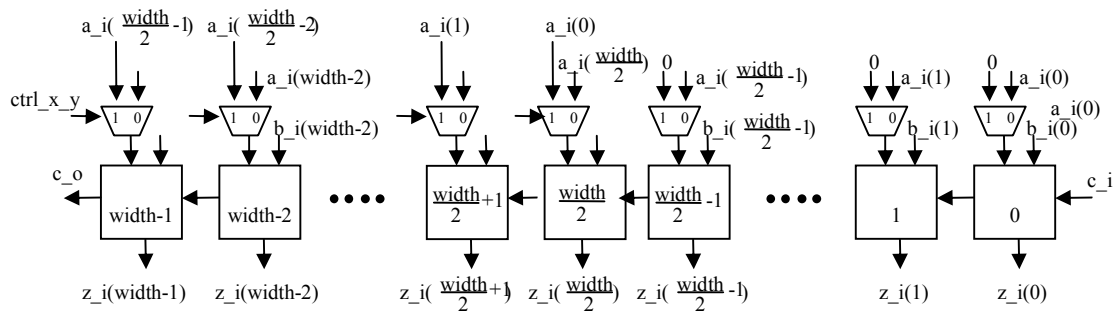
Guarde os resultados do Wave para uso no futuro.

Perguntas: seguindo as recomendações, a simulação mostrou o comportamento esperado do circuito como descrito na apostila de conceitos?

3) Captura, compilação e simulação do somador *ripple_carry* no modelo estrutural (atividade com *generate*)

Nesta seção, o(a) aluno(a) irá exercitar o uso da construção *generate*. O arquivo VHDL *rc_adder_3.vhd* disponível no *NEWSERVERLAB* contém uma versão modificada da entidade *rc_adder_2* (primeira parte desta prática). Atenção: o arquivo está incompleto, ou seja, as modificações introduzidas ainda precisam ser completadas pelo(a) aluno(a) antes de simular o circuito.

O circuito a ser modelado é o apresentado na figura abaixo (idêntico ao apresentado na apostilas de conceitos). Veja que há uma semelhança ao deslocador do módulo *ALU* (segunda parte desta prática).



- Abra o arquivo `rc_adder_3.vhd` da pasta `X:\psi3451\aula_6\rca_3`.
- Analise toda a descrição, linha a linha. Tenha certeza que tenha encontrado e entendido os itens seguintes:
 - as novas construções do *generate-if*
 - a parte não modificada do `rc_adder` (idêntica ao da seção 1) e a parte já modificada com a inferência dos *muxes* para a implementação do *shifter*, como ilustrado na apostila de conceitos
- Complete o projeto de tal forma que os *muxes* estejam implementados para todos os bits; use *generate-if* nos pontos que faltam.
- Copie os arquivos de *testbench* de `X:\psi3451\aula_6\rca_2` para `X:\psi3451\aula_6\rca_3`. Modifique-os de tal forma a ficarem compatíveis com o novo bloco ripple_carry (atente para as diferenças quanto às entradas).
- No Modelsim, crie a biblioteca *work* na pasta `rca_3`.
- Compile inicialmente o arquivo `full_adder_1.vhd` (da aula 2) contendo a entidade `full_adder_1` (*dataflow*).
- Compile em seguida o arquivo `rc_adder_3.vhd` (entidade `rc_adder`) e os arquivos de *testbench*.
- Compile, em sequência, primeiro o arquivo `full_adder_1` (*dataflow*) e depois o `rc_adder_3`, além dos *testbenches*.
- Faça o procedimento de simulação.

Recomendação: realize a simulação para verificar:

- o funcionamento do circuito de acordo com o descrito na apostila de conceitos
- simule as várias condições existentes para *ctrl_x_y*

Guarde os resultados do Wave para uso no futuro.

Perguntas: seguindo as recomendações, a simulação mostrou o comportamento esperado do circuito como descrito na [apostila de conceitos](#)? Verificou comportamento semelhante ao do conjunto `rc_adder` e *shifter* da seção 1?