

## PSI-3451 Projeto de CI Lógicos Integrados

### Aula 8- Atividade de Aula com *Testbenches*

A parte prática da aula 8 permitirá ao aluno familiarizar-se com testbenches, inicialmente testando alguns modelos e, posteriormente, desenvolvendo outros.

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

### PARTE PRÁTICA

São fornecidos 3 arquivos VHDL de testbenches, associados aos modelos descritos na apostila de conceitos:

- *Testbench* para simulação de circuitos combinacionais (*testbench\_comb*): ambiente de simulação do módulo alu (da aula 5).
- *Testbench* para simulação de circuitos sequenciais simples (*testbench\_seq\_1*): ambiente de simulação do módulo fsm\_main (da aula 3).
- *Testbench* para simulação de circuitos sequenciais com protocolo (*testbench\_seq\_2*): ambiente de simulação do módulo fsm\_food\_with\_data (da aula 6).

Em cada experimento, os arquivos VHDL serão submetidos 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\_8 para armazenar os resultados desta prática. Como serão realizadas 6 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\_8\comb (para o testbench do circuito combinacional)  
X:\psi3451\aula\_8\seq\_1 a X:\psi3451\aula\_8\seq\_5 (para os circuitos sequenciais )
- Ainda com o **Windows Explorer**, selecione **Rede=> NEWSERVERLAB => psi3451 => aula\_8** e copie os arquivos correspondentes a cada pasta para criada.

#### 1) Captura, compilação e simulação do testbench para o circuito combinacional.

- Do projeto a ser verificado (DUV):

- a) Copie para a pasta **X:\psi3451\aula\_8\comb** o arquivo *alu.vhd* utilizado na aula 5. Como este último contém submódulos, copie todos eles, como por exemplo o *rca\_adder\_2.vhd*.
  - b) Relembre o funcionamento do módulo *alu* e reconheça os pinos de entrada para o controle de dados para as operações (rever a apostila da aula, se necessário).
- Do gerador de estímulos:
- c) Abra o arquivo *stimuli\_comb.vhd* da pasta **X:\psi3451\aula\_8\comb**.
  - 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 *alu*, a compatibilidade de seus portos
    - 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\_comb.vhd* da pasta **X:\psi3451\aula\_8\comb**. 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\_module*
    - a instanciação do módulo *alu*
    - a conexão entre os dois módulos
  - g) No Modelsim, crie a biblioteca *work* na pasta **X:\psi3451\aula\_8\comb**.
  - h) Compile os arquivos seguindo a hierarquia bottom-up.
  - i) Faça a simulação da entity *tb\_alu\_combinatorial*.

#### **Recomendação para a simulação:**

- observe que você não precisará gerar estímulos para as entrada através do comando **force**
- adicione os sinais desejado para a janela Wave e rode a simulação.
- após a primeira rodada, adicione mais vetores de teste (chamadas do procedure) para testar todas as condições de entrada.

. Guarde os resultados do Wave para futuras referências e comparações.

**Perguntas:** você conseguiu verificar todas as opções de controle do módulo *alu*? Seguindo as recomendações, a simulação mostrou o comportamento esperado do *testbench* como descrito na [apostila de conceitos](#)? Como você compara o uso do *testbench* com o uso do arquivo \*.tcl feito até o momento?

## 2) Captura, compilação e simulação do testbench para o primeiro circuito sequencial.

➤ Do projeto a ser verificado (DUV):

- a) Copie para a pasta **X:\psi3451\aula\_8\seq\_1** o arquivo *fsm\_1.vhd* utilizado na aula 3.
- b) Relembre o funcionamento do módulo fsm\_main e reconheça os pinos de entrada para o controle dos estados (rever a apostila da aula, se necessário).

➤ Do gerador de clock:

- c) Abra o arquivo *clock\_generator.vhd* da pasta **X:\psi3451\aula\_8\seq\_1**.
- d) Analise toda a descrição, linha a linha. Tenha certeza que tenha encontrado e entendido os itens seguintes:
  - o porto clk
  - o período estabelecido para o sinal de relógio

➤ Do gerador de estímulos:

- e) Abra o arquivo *stimuli\_seq\_1.vhd* da pasta **X:\psi3451\aula\_8\seq\_1**.
- f) 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 fsm\_main, a compatibilidade de seus portos
  - a instanciação do bloco gerador de clock
  - o procedimento de reset e a sua chamada na arquitetura
  - 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 dentro dos procedimentos

➤ Do testbench (topo):

- j) Abra o arquivo *testbench\_seq\_1.vhd* da pasta **X:\psi3451\aula\_8\seq\_1**. Ele está codificado no modelo VHDL estrutural.
- k) Analise toda a descrição, linha a linha. Tenha certeza que tenha encontrado e entendido os itens seguintes:
  - a instanciação do módulo stimuli\_module
  - a instanciação do módulo fsm\_main
  - a conexão entre os dois módulos
- l) No Modelsim, crie a biblioteca *work* na pasta **X:\psi3451\aula\_8\seq\_1**.
- m) Compile os arquivos seguindo a hierarquia bottom-up.
- n) Faça a simulação da entity *tb\_fsm\_main\_seq*.

### **Recomendação para a simulação:**

- observe que você não precisará gerar estímulos para as entrada através do comando **force**
- adicione os sinais desejado para a janela Wave e rode a simulação.
- após a primeira rodada, adicione mais vetores de teste (chamadas do procedure) para testar todas as condições de entrada.

. Guarde os resultados do Wave para futuras referências e comparações.

**Perguntas:** você conseguiu verificar todas as opções de controle do módulo fsm\_main? Seguindo as recomendações, a simulação mostrou o comportamento esperado do *testbench* como descrito na apostila de conceitos? Como você compara o uso do testbench com o uso do arquivo \*.tcl feito até o momento?

### 3) Captura, compilação e simulação do testbench para o segundo circuito sequencial.

- Do projeto a ser verificado (DUV):
  - a) Observe na pasta **X:\psi3451\aula\_8\seq\_2** o arquivo *fsmd\_3.vhd*; este é baseado no *fsmd\_1.vhd* utilizado na aula 6.
  - b) Relembre o funcionamento do módulo fsm\_food\_with\_data e reconheça os pinos de entrada para o controle dos estados (rever a apostila da aula, se necessário).
  - c) Analise toda a descrição, linha a linha. Tenha certeza que tenha encontrado e entendido os itens modificados desta implementação:
    - a mudança do gerador do número aleatório para o recebimento do dado de registrador de contagem
    - o retorno ao valor 0 do contador após atingir o valor máximo
    - os incrementos do contador
- Do gerador de estímulos:
  - d) Abra o arquivo *stimuli\_seq\_2.vhd* da pasta **X:\psi3451\aula\_8\seq\_2**.
  - e) 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 fsm\_food\_with\_data, a compatibilidade de seus portos
    - a instanciação do bloco gerador de clock
    - o procedimento de reset e a sua chamada na arquitetura
    - a correlação entre os sinais de start e done.
    - o processo de simulação (process) sem a lista de sensibilidade (portanto, com wait)
    - a construção wait dentro do procedimento de reset
- Do testbench (topo):
  - f) Abra o arquivo *testbench\_seq\_2.vhd* da pasta **X:\psi3451\aula\_8\seq\_2**. Ele está codificado no modelo VHDL estrutural.

- g) Analise toda a descrição, linha a linha. Tenha certeza que tenha encontrado e entendido os itens seguintes:
- a instanciação do módulo `stimuli_module`
  - a instanciação do módulo `fsm_food_with_data`
  - a conexão entre os dois módulos
- h) No Modelsim, crie a biblioteca *work* na pasta **X:\psi3451\aula\_8\seq\_2**.
- i) Compile os arquivos seguindo a hierarquia bottom-up.
- j) Faça a simulação da entity *tb\_fsm\_main\_seq*.

**Recomendação para a simulação:**

- observe que você não precisará gerar estímulos para as entrada através do comando **force**
- adicione os sinais desejado para a janela Wave e rode a simulação.
- alongue bastante o tempo de simulação para ver o efeito do contador na geração de números "aleatórios".

Guarde os resultados do Wave para futuras referências e comparações.

**Perguntas:** você conseguiu verificar todas as opções de controle do módulo `fsm_food_with_data`? Seguindo as recomendações, a simulação mostrou o comportamento esperado do *testbench* como descrito na apostila de conceitos?

**4) Modificação de Testbench para `fsm_main` modificado**

- a) Copie para a pasta **X:\psi3451\aula\_8\seq\_3** todos os arquivos *\*.vhd* de **X:\psi3451\aula\_8\seq\_1**.
- b) Apague o arquivo *fsm\_main\_1.vhd* e copie para a pasta o arquivo *fsm\_main\_2.vhd* da aula 3.
- c) Modifique os nomes dos demais arquivos para compatibilizar com a enumeração da pasta "seq\_3".
- d) Relembre o funcionamento do novo módulo `fsm_main` e reconheça os pinos de entrada para o controle dos estados (rever a apostila prática da aula 3, se necessário).
- e) Realize as eventuais modificações no gerador de estímulos, necessárias para a simulação da máquina de estados.
- f) Compile os arquivos e simule

**Perguntas:** Conseguiu verificar a passagem por todos os estados? A simulação mostrou o comportamento esperado do circuito como descrito na apostila de conceitos?

**5) Elaboração de Testbench para `fsmd` (com datapath explícito)**

Como parte do projeto do Snake, o *fsmd\_2.vhd* da aula 6, `fsmd` com datapath explícito, o(a) aluno(a) deve verificar que o circuito foi projetado corretamente. Elabore os testbenches para os dois submódulos `fsm_2` e `datapath_2`. Para isto:

- a) Copie para a pasta **X:\psi3451\aula\_8\seq\_4** os arquivos *fsm\_2.vhd* de **X:\psi3451\aula\_6\fsmd\_2** e para **X:\psi3451\aula\_8\seq\_5** os arquivos *datapath.vhd* (com todos os submódulos).
- b) Desenvolva os testbenches para cada um dos circuitos com as estruturas vistas nas seções anteriores (gerador de clock e os vários procedimentos).
- c) Compile os arquivos e simule.