

## PSI-3451 Projeto de CI Lógicos Integrados

### Aula 5- Atividade de Aula com *Testbenches*

A parte prática da aula 5 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 2 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 com protocolo (*testbench\_seq\_1*): ambiente de simulação do módulo fsm\_main\_simplificado.

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\_5** 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\_5\comb** (para o testbench do circuito combinacional)

**X:\psi3451\aula\_5\seq\_1** (para o testbench do circuito sequencial fornecido- FSM Main simplificado)

**X:\psi3451\aula\_5\seq\_2** (o testbench do circuito sequencial modificado do anterior- FSM Main completo)

- Ainda com o **Windows Explorer**, selecione **Rede=> NEWSERVERLAB => psi3451 => aula\_5** 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\_5\comb** o arquivo *num\_gen\_1.vhd* utilizado na aula 4.

- b) Relembre o funcionamento do módulo `num_gen_direct` e reconheça os pinos de entrada para o controle de dados para as operações dos muxes (rever a apostila da aula, se necessário).
- Do gerador de estímulos:
- c) Abra o arquivo `stimuli_comb.vhd` da pasta **X:\psi3451\aula\_5\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 `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_comb.vhd` da pasta **X:\psi3451\aula\_5\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 `num_gen_direct`
  - a conexão entre os dois módulos
- g) No Modelsim, crie a biblioteca `work` na pasta **X:\psi3451\aula\_5\comb**.
- h) Compile **todos os arquivos** seguindo a hierarquia bottom-up.
- i) Faça a simulação da entity `tb_num_gen_direct_combinatorial`.

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

- adicione os sinais desejado para a janela Wave e rode a simulação por um tempo compatível com os tempos de `wait` do `stimuli_module`.
- observe que você não precisará gerar estímulos para as entrada através do comando **force**
- 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 `num_gen_direct`? 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\_5\seq\_1** o arquivo *fsm\_main\_1.vhd*.
- b) Revise o funcionamento do módulo *fsm\_main\_simplificado* na especificação do Snake e na apostila de conceitos desta aula; reconheça os pinos de entrada para o controle dos estados.

➤ Do gerador de clock:

- c) Abra o arquivo *clock\_generator.vhd* da pasta **X:\psi3451\aula\_5\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\_5\seq\_1**.
- f) Analise toda a descrição, linha a linha. Tenha certeza que tenha encontrado e entendido os itens seguintes:
  - a compatibilidade dos portos do gerador de estímulos e do *fsm\_main\_simplificado*.
  - 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
  - os protocolos nos estados de inicialização, de alimento e de passo e o sequenciamento esperado para o funcionamento do módulo *fsm\_main\_simplificado*.

➤ Do testbench (topo):

- j) Abra o arquivo *testbench\_seq\_1.vhd* da pasta **X:\psi3451\aula\_5\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\_simplificado*
  - a conexão entre os dois módulos
- l) No Modelsim, crie a biblioteca *work* na pasta **X:\psi3451\aula\_5\seq\_1**.
- m) Compile **todos os arquivos** seguindo a hierarquia bottom-up.
- n) Faça a simulação da entity *tb\_fsm\_main\_seq\_1* (lembre-se que é o topo).

### **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. Atenção especial para a inserção e observação dos sinais de estado do módulo `fsm_main_simplificado` para se certificar da correta transição de estados.
- após a primeira rodada, modifique livremente o seu código para testar outras transições de estado possíveis e demais condições de entrada para verificar os efeitos.

Guarde os resultados do Wave para futuras referências e comparações.

**Perguntas:** Conseguiu verificar a passagem por todos os estados? Você conseguiu verificar todas as opções de controle do módulo `fsm_main_simplificado`? Seguindo as recomendações, a simulação mostrou o comportamento esperado do *testbench* como descrito na apostila de conceitos?

### **3) Captura e simulação do da máquina `fsm_main` no modelo comportamental com a inserção do estado IDLE**

- a) Faça uma cópia do arquivo `fsm_main_1.vhd` de **X:\psi3451\aula\_5\seq\_1** para a pasta **X:\psi3451\aula\_5\seq\_2**. Modifique o nome para `fsm_main_2.vhd`
- b) Em seguida modifique o novo arquivo `fsm_main_2.vhd` (com o programa **Notepad++**), denominando a entity simplesmente como `fsm_main`.
- c) Incorpore o estado IDLE, como especificado na figura 1, onde as alterações estão anotadas em vermelho. Atenção: esta nova versão será aquela utilizada no Snake.
- d) Observe que, agora, os sinais `cnt_rdy` e `fsm_s_done` passam a ter função dentro desta nova máquina (lembre-se que estes sinais não tinham atuação no circuito do item 2).

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

- a) Copie para a pasta **X:\psi3451\aula\_5\seq\_2** todos os arquivos de testbench de **X:\psi3451\aula\_5\seq\_1**.
- b) Modifique os nomes de todos os arquivos para compatibilizar com a enumeração da pasta "seq\_2".
- c) Relembre o funcionamento do novo módulo `fsm_main` e reconheça os pinos de entrada para o controle dos estados .
- d) Estude o comportamento alterado do novo `fsm_main` e realize as eventuais modificações no gerador de estímulos, necessárias para a simulação da máquina de estados (incluído o estado IDLE).
- e) Compile os arquivos e simule ( o testbench é o topo).
- f) **Perguntas:** Conseguiu verificar a passagem pelo novo estado IDLE e as suas transições? Seguindo as recomendações, a simulação mostrou o comportamento esperado do *testbench* como descrito na apostila de conceitos?

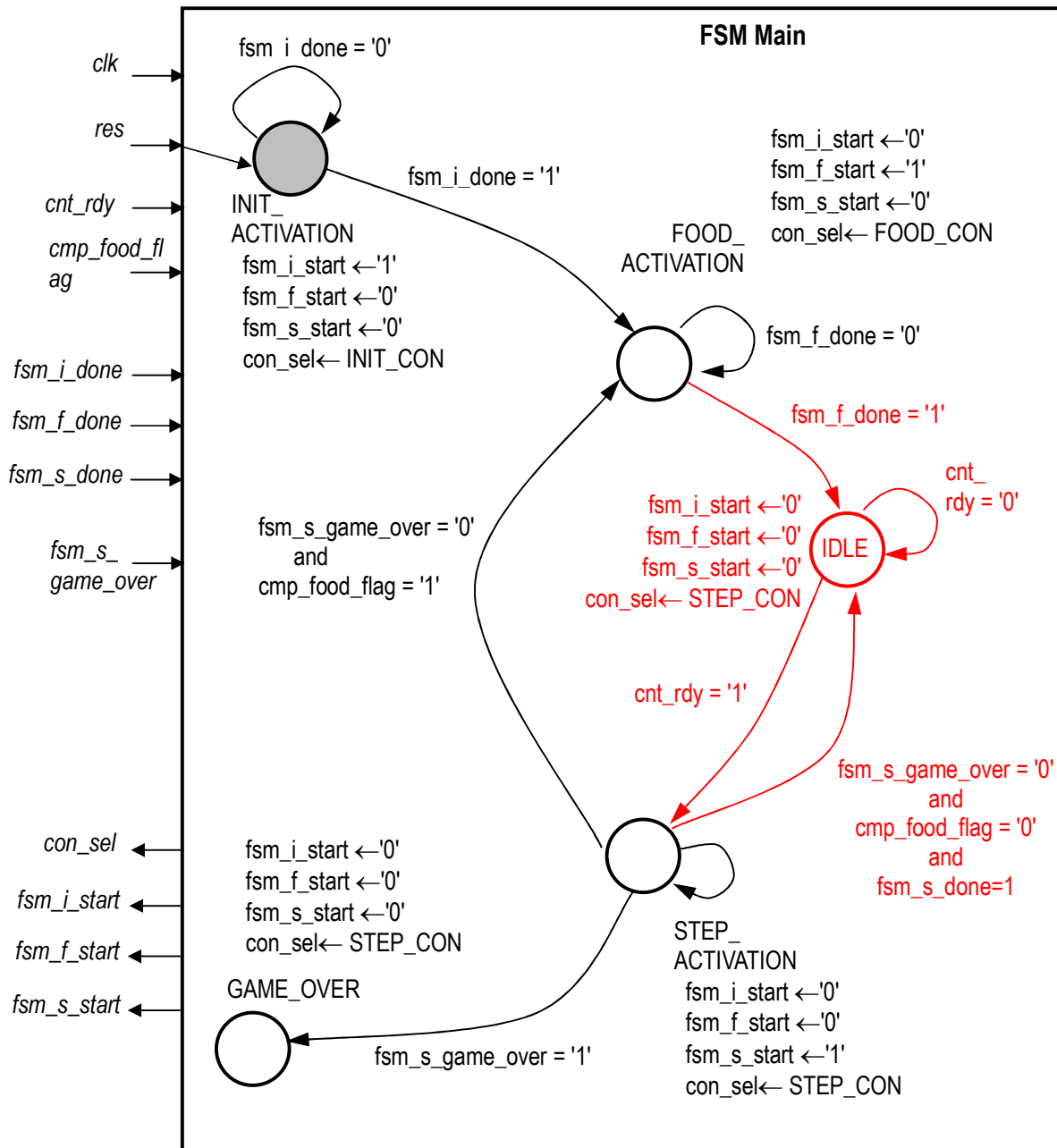


Figura 1. Diagrama de estados de FSM Main completo (com IDLE incluído)