

## PSI-3451 Projeto de CI Lógicos Integrados

### Aula 9- Atividade de Aula com Memória e FIFO

A parte prática da aula 9 permitirá ao aluno familiarizar-se com a geração de memórias (SRAM e FIFO) na forma de IP cores do programa Quartus II e descrevendo os testbenches para o seu teste.

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

### PARTE PRÁTICA

Nesta prática, o(a) aluno(a) não receberá nenhum arquivo VHDL. Os seus modelos deverão ser gerados dentro do Quartus II.

#### 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\_9** para armazenar os resultados desta prática. Como serão realizadas 2 jogos de 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\_9\mem\_1** (para a memória SRAM e seu testbench)

**X:\psi3451\aula\_8\fifo\_1** (para a FIFO e seu testbench)

#### 1) Geração do modelo de memória SRAM no Quartus II.

- a) Carregue o programa *Quartus II* da Altera para fazer as simulações através do seguinte caminho: **Iniciar/Programas/Altera/Quartus II 9.1**. (atenção: tenha certeza que é a versão 32 bits do programa)
- b) Crie um projeto com o **New Project Wizard**. Dê como nome do projeto *mem\_1*, apontado-o para a pasta **X:\psi3451\aula\_9\mem\_1**. Na janela referente a arquivos, não inclua nenhum. Assinale como dispositivo, família **Cyclone II** e componente **EP2C35F672C6**. Na caixa de diálogo seguinte, no campo de simulação, adicione em Tool name: **ModelSim-Altera** e no Format: **VHDL**. Clique em **Finish**. Após finalizar esta etapa poderá observar as características do projeto na janela **Project Navigator**.
- c) No opção de menu **Tools**, selecione **MegaWizard Plg-in Manager**. Na caixa de diálogo, opte por criar uma nova variação de Megafunction. Clique em **Next**.
- d) No catálogo de IPs que se seguem, escolha em **Memory Compiler** a opção **RAM: 2-PORT**. Selecione como família, **Cyclone II**, linguagem de saída, **VHDL**, e defina para o arquivo de saída, o nome **mem\_1**. Clique em **Next**. A caixa de diálogo para as definições e configurações da RAM surgirá.

- e) A seguir você deverá definir a sua memória da seguinte forma (quando não especificado, optar por Automático- **Auto**, ou deixar em branco):
- duas portas com read/write
  - 64 endereços
  - saídas e entradas de 8 bits (1 byte)
  - clock único para leitura e escrita
  - criar todos os byte enable.
  - somente entradas registradas.
  - mixed port read-during-write em **Old memory content**.
- f) Ao terminar a operação clique em Yes na caixa de diálogo para incluir o arquivo gerado no seu projeto do Quartus.
- g) Confira no Quartus na aba *Files* da janela *Project Navigator* que o arquivo criado está presente.
- h) Clique duas vezes em *mem\_1.vhd* pra ver o seu conteúdo. Analise toda a descrição, linha a linha. Tenha certeza que tenha encontrado e entendido os itens seguintes:
- compatibilidade dos portos com as suas escolhas
  - a declaração e instanciação do componente altsyncram. Tente associar os generics com as opções realizadas.

## 2) Simulação do modelo de memória SRAM no Modelsim.

- a) Baseado no arquivo de testbench (e seus componentes) utilizado na aula 8 (copie os arquivos do caso do **seq\_1** e altere os seus nomes), planeje os testes para a memória RAM dual-port. (Dica: use a sintaxe de controle **for-loop** de processo para repetir as escritas e leituras para todas as palavras da memória)
- b) No editor de texto (Notepad++, por exemplo), faça as adaptações para as entradas da memória. Gaste o tempo necessário para esta tarefa.
- c) Após aprontar o testbench, retornar ao Quartus.
- d) No programa Quartus II (já aberto no projeto *mem\_1*), acesse o menu de **Assignments->Settings** e na opção “Compile test bench”:
- i) clique em testbench
  - ii) clique em NEW
  - iii) Completar: “Test bench name” = nome da entidade de seu testbench. “Top level module in test bench” = idem. “Design instance name in test bench” = nome da arquitetura de seu testbench.
  - iv) Selecione End simulation para 100ns.
  - v) No campo “Test bench files”, selecionar e adicionar o arquivo de testbench e todos os seus componentes e sub-componentes.
- e) Compile o arquivo *mem\_1.vhd*; observe que este é ainda o arquivo topo (principal) do projeto e, não o *testbench*. Para isto, selecione no menu **Processing > Start > Start Analysis &**

**Synthesis.** O resultado da compilação é apresentado numa janela tipo *pop-up*, pressione **OK**. Uma janela com o relatório da compilação é aberta automaticamente ou pode ser acessada pela opção do menu **Processing > Compilation report**.

- f) Realize a simulação funcional do arquivo plasma. Para realizar a simulação funcional: **Tools > Run EDA Simulation Tool > EDA RTL Simulation**. O ModelSim deverá abrir já com a janela de simulação pronta.
- g) Veja que 100ns já forma simulados (segundo item 2.d.4)

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

- rode mais tempo para ver todos os valores desejados
- teste as escritas pelos portos e verifique que em uma releitura, os valores estão ainda na memória.
- teste os *corner cases*, ou seja, os casos especiais comentados na apostila de conceitos.

Guarde os resultados do Wave para futuras referências e comparações.

**Perguntas:** você conseguiu verificar todas as escritas e leituras para a memória? Verificou o que ocorre em escritas simultâneas ao mesmo endereço? Verificou o que ocorre em caso de mixed port read-during-write?

- h) Ao final de tudo, salve o projeto no Quartus (em uma outra oportunidade, você poderá carregar todos os arquivos utilizado, abrindo este projeto).

### **3) Geração do modelo de FIFO no Quartus II.**

- a) Crie um novo projeto com o **New Project Wizard**. Dê como nome do projeto *fifo\_1*, apontado-o para a pasta **X:\psi3451\aula\_9\fifo\_1**. Na janela referente a arquivos, não inclua nenhum. Assinale como dispositivo, família **Cyclone II** e componente **EP2C35F672C6**. Na caixa de diálogo seguinte, no campo de simulação, adicione em Tool name: **ModelSim-Altera** e no Format: **VHDL**. Clique em **Finish**. Após finalizar esta etapa poderá observar as características do projeto na janela **Project Navigator**.
- b) No opção de menu **Tools**, selecione **MegaWizard Plg-in Manager**. Na caixa de diálogo, opte por criar uma nova variação de Megafunction. Clique em **Next**.
- c) No catálogo de IPs que se seguem, escolha em **Memory Compiler** a opção **FIFO**. Selecione como família, **Cyclone II**, linguagem de saída, **VHDL**, e defina para o arquivo de saída, o nome **fifo\_1**. Clique em **Next**. A caixa de diálogo para as definições e configurações da RAM surgirá.
- d) A seguir você deverá definir a sua memória da seguinte forma (quando não especificado, optar por Automático- **Auto**, ou deixar em branco):
  - 64 endereços
  - saídas e entradas de 8 bits (1 byte)
  - clock único para leitura e escrita
  - apenas o registrador de vazio (empty).

- clear síncrono.
  - fifo síncrona tradicional.
  - somente entradas registradas.
- e) Ao terminar a operação clique em Yes na caixa de diálogo para incluir o arquivo gerado no seu projeto do Quartus.
- f) Confira no Quartus na aba *Files* da janela *Project Navigator* que o arquivo criado está presente.
- g) Clique duas vezes em *fifo\_1.vhd* pra ver o seu conteúdo. Analise toda a descrição, linha a linha. Tenha certeza que tenha encontrado e entendido os itens seguintes:
- compatibilidade dos portos com as suas escolhas
  - a declaração e instanciação do componente scfifo. Tente associar os generics com as opções realizadas.

#### 4) Simulação do modelo de memória SRAM no Modelsim.

- a) Baseado no arquivo de testbench (e seus componentes) utilizado na aula 8 (copie os arquivos do caso do **seq\_2** e altere os seus nomes), planeje os testes para a FIFO. (Dica: use a sintaxe de controle **for-loop** de processo para repetir as escritas e leituras para todas as palavras da memória)
- b) No editor de texto (Notepad++, por exemplo), faça as adaptações para as entradas da memória. Gaste o tempo necessário para esta tarefa.
- c) Após aprontar o testbench, retornar ao Quartus.
- d) No programa Quartus II (já aberto no projeto *fifo\_1*), acesse o menu de **Assignments->Settings** e na opção “Compile test bench”:
- vi) clique em testbench
  - vii) clique em NEW
  - viii) Completar: “Test bench name” = nome da entidade de seu testbench. “Top level module in test bench” = idem. “Design instance name in test bench” = nome da arquitetura de seu testbench.
  - ix) Selecione End simulation para 100ns.
  - x) No campo “Test bench files”, selecionar e adicionar o arquivo de testbench e todos os seus componentes e sub-componentes.
- e) Compile o arquivo *fifo\_1.vhd*; observe que este é ainda o arquivo topo (principal) do projeto e, não o *testbench*. Para isto, selecione no menu **Processing > Start > Start Analysis & Synthesis**. O resultado da compilação é apresentado numa janela tipo *pop-up*, pressione **OK**. Uma janela com o relatório da compilação é aberta automaticamente ou pode ser acessada pela opção do menu **Processing > Compilation report**.
- f) Realize a simulação funcional do arquivo plasma. Para realizar a simulação funcional: **Tools > Run EDA Simulation Tool > EDA RTL Simulation**. O ModelSim deverá abrir já com a janela de simulação pronta.
- g) Veja que 100ns já forma simulados (segundo item 2.d.4)

**Recomendação para a simulação:**

- rode mais tempo para ver todos os valores desejados
- teste escritas e leituras individuais, assim como escritas e leituras em bloco.
- teste ritmos diferentes de escritas e leituras, porém o ritmo de escrita não poderá ser alto demais (observe que, pela sua função dentro do *Snake*, a FIFO não precisa de indicação de full- pense no porquê)
- teste o caso de empty; acerte o protocolo para esta condição.
- teste o sinal de clear.

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

**Perguntas:** você conseguiu verificar o funcionamento do buffer? Verificou a sinalização de vazio? Verificou o funcionamento do *clear*, ou seja, o *flush* de dados?