

Escola Politécnica da Universidade de São Paulo
Departamento de Engenharia de Sistemas Eletrônicos - PSI

PSI-2553- Projeto de Sistemas Integrados

Exp 1B

Simulação RTL da arquitetura do Processador Fibonacci

e

Simulação comportamental do algoritmo de Fibonacci

(Parte Prática)

1. OBJETIVOS

2. PARTE EXPERIMENTAL

2.1. Simulação RTL do M_F

2.2. Simulação comportamental do algoritmo de Fibonacci

2.3. Discussão

Objetivos

Esta experiência visa validar o projeto do processador Fibonacci através de 2 simulações:

1. A arquitetura RTL do processador Fibonacci sintetizado pelo aluno e capturado em VHDL na exp1A.
2. O código VHDL comportamental utilizado como ponto de partida deste projeto.

As simulações serão realizadas através do uso de *testbenches*.

Para as simulações comportamentais (através de *testbench*) são fornecidos os seguintes três arquivos:

- fibonacci_tbw_alg (descrição VHDL do módulo topo: estímulos e Fibonacci) (ver figura 1)
- estímulos_alg (descrição VHDL dos estímulos)
- inicio.txt

Para as simulações RTL (através de *testbench*) são fornecidos os seguintes três arquivos:

- fibonacci_tbw_RTL (descrição VHDL do módulo topo: estímulos e Fibonacci) (ver figura 1)
- estímulos_RTL (descrição VHDL dos estímulos)
- inicio.txt

Além destes serão também necessários:

- Código VHDL comportamental original
- Código VHDL RTL do *datapath*, da FSM, do M_F e das células da biblioteca.

Parte Experimental

Criar a sua pasta **X:\psi2553\exp1B** em sua unidade de Rede X: (em Meu Computador).

Copiar para esta pasta o conteúdo da pasta:

Rede=> Newserverlab\psi2553\exp1B.

Observe que serão copiadas duas pastas: **testbench_alg** e **testbench_RTL** e seus respectivos arquivos.

Copie também:

- o arquivo fornecido na exp1 (VHDL comportamental do módulo de Fibonacci) para a pasta **testbench_alg**
- todos os arquivos VHDL usados durante a exp2A (VHDL RTL do *datapath*, da FSM, do M_F e VHDL das células da biblioteca) para a pasta **testbench_RTL**.

2.1. Simulação RTL do M_F

A simulação com o uso do *testbench* será realizada através do programa *Modelsim* disparado de dentro do *Quartus II 9.1*. O uso o simulador ModelSim traz vantagens em relação ao simulador nativo do Quartus, como a facilidade em se acessar os sinais internos do projeto, e uma interface mais adequada a compilação e simulação de arquivos VHDL não sintetizáveis, como arquivos de testbench em geral.

- 1) Vá até a pasta **X:\psi2553\exp1B\testbench_RTL** e observe os arquivos VHDL. Use a figura 1 da apostila para se orientar.
- 2) Dispare o *Quartus II 9.1*.
- 3) A fim de configurar o *Quartus* para utilizar o ModelSim, acesse o menu **Assignments > EDAToolSettings > Simulation**.
 - a) Em **Tool Name**, opte por *Modelsim-Altera*.
 - b) Em **EDA Netlist Writer settings->Format for output netlist**, opte por *VHDL*.
- 4) Em **NativeLink settings**, opte por *Compile testbench*:
 - a) clique em *testbench*
 - b) clique em NEW
 - c) Completar: “**Test bench name**” = **fibonacci_tbw**. “**Top level module in test bench**” = **fibonacci_tbw**. “**Design instance name in test bench**” = **behavior**.
 - d) No campo “**Test bench files**”, adicionar os arquivos *fibonacci_tbw_RTL.vhd* e *estímulos_RTL.vhd*, que estão em **X:\psi2553\exp1B\testbench_RTL**.
 - e) Não habilite a opção “**Use script to set up simulation**”.
 - f) Realize a simulação funcional do arquivo Fibonacci. 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) Selecione os sinais do módulo Fibonacci e rode por alguns microsegundos (utilize o comando **run xx us**) até que o resultado do cálculo apareça na tela.

Observações:

- a) O projeto no Quartus é o do módulo Fibonacci - os arquivos referentes ao *testbench* entraram apenas para a simulação
- b) A **EDA RTL Simulation** é realizada sobre a descrição RTL de entrada.

Vamos agora mostrar como é possível automatizar as simulações.

Feche o ModelSim e retorne ao ambiente Quartus.

- 5) No Quartus, configure a simulação acessando o menu **Assignments > EDAToolSettings > Simulation**.
 - a) Em **NativeLink settings**, opte por **Compile test bench**.
 - b) Na opção **Use script to set up simulation** indique o caminho do arquivo **inicio.txt** (que está em **X:\psi2553\exp1B\testbench_RTL**). Este arquivo é o script para a simulação no ModelSim. Clique em **Ok**.
 - c) Novamente, realize a simulação funcional do arquivo Fibonacci. Para realizar a simulação funcional: **Tools Run EDA Simulation Tool > EDA RTL Simulation**. O ModelSim deverá abrir já com a janela de simulação pronta.
 - d) Observe o arquivo **inicio.txt** e identifique o **valor da entrada** assim como os sinais que serão observados durante a simulação.
 - e) Faça a simulação da arquitetura Fibonacci e verifique se o resultado está correto. Caso contrário verifique quais sinais estão dando um resultado errado para depois verificar e corrigir sua descrição em VHDL. Repita a simulação até que o resultado esteja correto.

A seguir, serão feitas outras duas simulações para diferentes valores de entrada.

- f) Calcule qual é o maior número inteiro (MNI) cujo número de Fibonacci pode ser calculado sem ocorrer *overflow* (estouro). Lembre-se que a saída é de 32 bits, sendo o primeiro bit reservado para o sinal. Modifique o *testbench*. **Repita a simulação para os valores MNI e para (MNI+1)** a fim de observar a ocorrência do estouro.
- g) Modifique o *testbench*, de tal forma que sejam calculados os números de Fibonacci **dos dois números USP da dupla (N_USP1 e N_USP2)** e repita a simulação. Se você estiver trabalhando sem dupla faça a simulação para seu **N_USP e para (N_USP+1)**.

→ No relatório

DADOS EXPERIMENTAIS

- 1. Impressão da carta de tempos para os itens 5e, 5f e 5g indicando os resultados relevantes de cada simulação. Atenção: só incluir no relatório as partes relevantes de cada simulação.**
- 2. Imprima o arquivo inicio.txt e explique o que representa este script.**

ANÁLISE E DISCUSSÃO

- 3. Qual é o valor do MNI e quantos ciclos de relógio foram necessários para calcular o correspondente resultado (simulação 5f)?**
- 4. Quais são os números USP e quantos ciclos de relógio foram necessários para calcular os respectivos resultado (N_USP1 e N_USP2 ou N_USP e N_USP+1) (simulação 5g).**
- 5. Identificar quais foram os estados mais repetidos no processo de cálculo justificando o porquê.**

2.2. Simulação comportamental do algoritmo de Fibonacci

Para simular o código VHDL referente ao comportamento do M_F usaremos diretamente o simulador *ModelSim*, uma vez que o programa Quartus não aceita construções comportamentais de alto nível de abstração, somente código sintetizável.

1. Dispares o ModelSim ALTERA STARTER EDITION 6.5b pelo botão de **Start-> All Programs-> Altera-> ModelSim-Altera 6.5b Starter Edition-> ModelSim-Altera 6.5b Starter Edition**.
2. Na janela do **ModelSim**, clique em **Compile**. Na caixa de diálogo, escolha o Library **work**, procure a pasta **X:\psi2553\exp1B\ testbench_alg** e selecione os três arquivos VHDL. Clique em **Compile** e depois em **Done**.
3. Observe que na aba de Libraries do **ModelSim**, a biblioteca **work** foi criada. Expanda-a e localize as entidades VHDL. Em seqüência, clique com o direito na entidade **fibonacci_tbw** e selecione **simulate without Optimization**. Uma outra aba deverá aparecer no canto inferior esquerdo (**sim**), indicando que o código compilado foi carregado para a simulação. Observe que o projeto topo com as suas instâncias estão presentes.
4. Na janela **ModelSim** selecione **View** e escolha **Wave**, quando aparecerá a janela de simulação. Na aba **sim**, selecione a entidade topo (**fibonacci_tbw**). Selecione o menu **Add => To Wave => All Items in Region**. Isto fará com que todos os sinais e variáveis presentes no código VHDL de **fibonacci_tbw** sejam observáveis na janela **Wave**. Na janela de comando de linha do **Modelsim** o comando **run tempo** para definir o quanto se desejará simular. Tente com **run 1 us**. Repita a simulação ajustando o tempo para que os números da seqüência de Fibonacci possam ser observados.
 - a) Simule o algoritmo para as condições definidas pelo *testbench* fornecido.
 - b) Modifique o *testbench* e repita a simulação para MNI e (MNI+1).
 - c) Modifique o *testbench* e repita a simulação para N_USP1 e N_USP2 ou , se não estive em dupla para N_USP e (N_USP+1).

→ No relatório:

DADOS EXPERIMENTAIS

Impressão da carta de tempos para os itens 4a, 4b e 4c indicando os resultados relevantes de cada simulação. Atenção: só incluir no relatório as partes relevantes de cada simulação.

ANÁLISE E DISCUSSÃO

Observe os resultados das 3 simulações RTL e das respectivas 3 simulações comportamentais. Ao comparar estes resultados, que conclusões você tiraria?

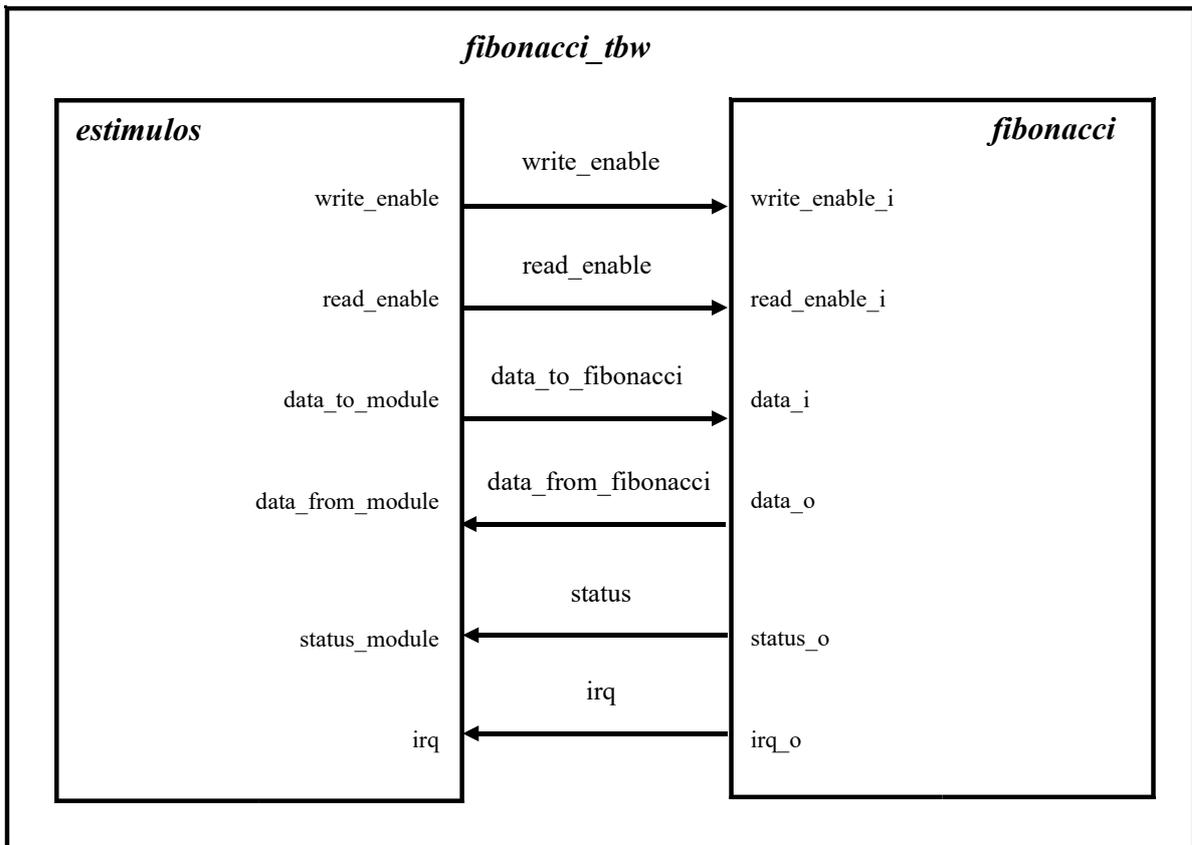


Figura 1: Diagrama de blocos do testbench utilizado nas simulações, ilustrando a esquerda o modulo gerador de estímulos e a direita o modulo Fibonacci sendo testado. Na arquitetura RTL aparecem, além deste, os sinais RST e CLK