

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

PSI-2553- Projeto de Sistemas Integrados

Captura da Arquitetura RTL do Processador Fibonacci em VHDL

O objetivo desta experiência é criar um modelo estrutural VHDL da arquitetura RTL do processador Fibonacci sintetizado anteriormente. A validação deste modelo por simulações será feita posteriormente.

Cada grupo deverá partir dos arquivos VHDL fornecidos e preencher as linhas de código incompletas de acordo com o resultado da síntese RTL, realizada durante a aula de teoria (17/03).

Para a captura, são fornecidos três arquivos *template*:

- fibonacci_incompleto_RTL.vhd (descrição do processador, datapath+FSM)
- template_datapath_RTL.vhd (descrição do *datapath*)
- template_FSM_Moore_RTL.vhd (descrição da FSM)

A biblioteca é composta dos modelos VHDL (comportamental) de 5 blocos funcionais:

- subtrator
- somador
- igual/diferente
- multiplexador 2->1
- registrador (com sinal *load*)

A figura 1 abaixo ilustra o esquema hierárquico do M_F.

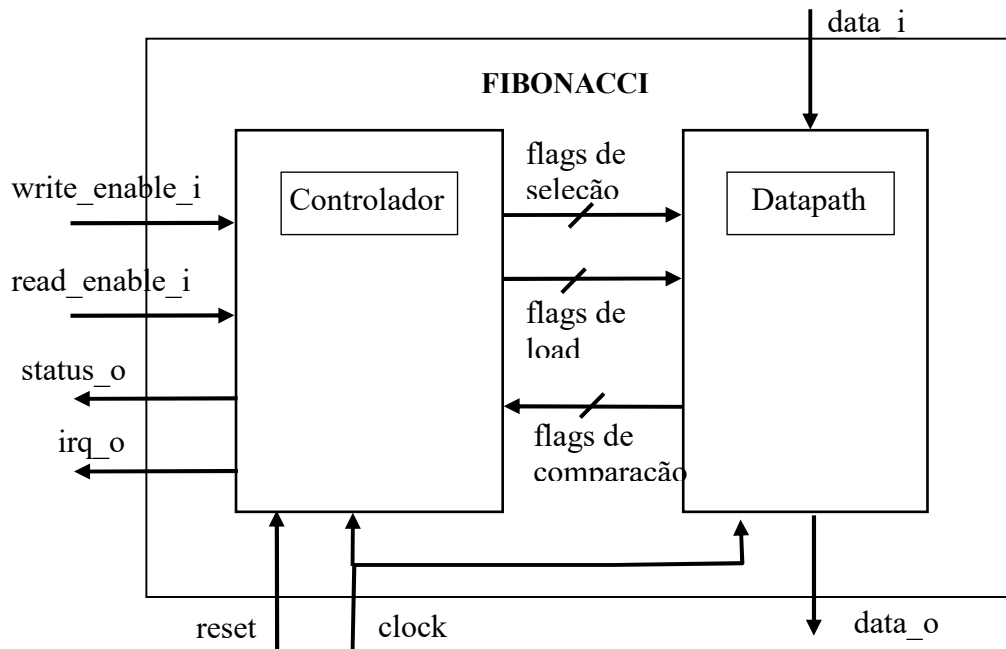


Figura 1 – Esquema hierárquico do M_F

Inicie a experiência criando a pasta **PSI2553** e dentro dela a pasta **exp1A**.

Copie para esta pasta todos os arquivos VHDL (são 8) presentes em **Rede\Newserverlab\psi2553\exp1A\modelo_rtl**

Antes de iniciar a edição dos arquivos VHDL padrão, observe atentamente o conteúdo de cada um deles. Certifique-se que você entende a estrutura global de cada descrição. A fim de personalizar cada padrão aos resultados obtidos pela síntese manual (controle e *datapath*) é necessário ter em mãos:

- O desenho do esquema resultante da síntese da arquitetura RTL do *datapath*.
- A tabela de transição de estados de sua FSM (controlador) devidamente codificada.
- O desenho do esquema do M_F, composto dos seus 2 módulos (*datapath* e FSM cf. figura 1), identificando os nomes de todos os *flags*, sinais de entrada e de saída.

A partir deste momento você deverá editar os arquivos (**nesta sequência**):

- 1- template_datapath_RTL.vhd
- 2- template_FSM_Moore_RTL.vhd
- 3- fibonacci_incompleto_RTL.vhd

Em cada arquivo será necessário personalizar todas as linhas incompletas com os dados obtidos no projeto esquemático.

Personalização do arquivo VHDL do *datapath*

Observe que o código padrão fornecido é um *netlist* de componentes de biblioteca.

A fim de criar o arquivo VHDL de seu *datapath*, modifique o arquivo padrão efetuando as seguintes tarefas:

- a) Atribuir à entidade (*entity*) o mesmo nome definido no arquivo topo *fibonacci_incompleto.vhd* (que será utilizado mais adiante).
- b) Personalizar todos os portos de entrada e saída. Use os mesmos nomes que você adotou no seu projeto esquemático.
- c) Verificar se todos os módulos funcionais que compõem o seu *datapath* estão presentes na biblioteca. Se faltar algum, acrescente (não esquecendo de criar também a entidade correspondente).
- d) Declarar todos os sinais internos (conexões entre os componentes de seu *datapath*) de acordo com o seu esquema RTL. Use os mesmos nomes que você adotou no seu projeto esquemático.
- e) Instanciar cada componente do esquema RTL. Por exemplo, se houver 3 registradores será necessário declarar 3 instâncias: *reg_1*, *reg_2*, *reg_3*. Os portos de cada instância devem ter os mesmos nomes adotados no seu projeto esquemático.
- f) Declarar os nomes de todos os sinais dos PORT MAP de cada módulo presente na sua arquitetura. Use os mesmos nomes que você adotou no seu projeto esquemático.
- g) Salvar o seu trabalho dando ao arquivo VHDL de seu *datapath* o mesmo nome que o de sua entidade.

Verificação do arquivo VHDL do *datapath*

A fim de verificar se o arquivo VHDL personalizado está livre de erros de sintaxe, será usado o compilador VHDL do programa *Quartus II 9.1 da Altera*.

- a. a) Dispare o programa através do caminho: *Iniciar/Programas/Altera/Quartus II 9.1*. Aparecerá a janela de introdução do *wizard*. Clique *Next*.
- b) Indique **X:\psi2553\exp1A\RTL_model** como o caminho de sua pasta de trabalho. Indique os nomes do projeto esquemático e módulo do *datapath*. Clique *Next*.
- c) Adicione o arquivo VHDL do *datapath*. Estabeleça o caminho do arquivo, após o qual pressione *Add*. Clique *Next*.
- d) Selecione a família *CycloneII* e o dispositivo **EP2C35F672C6** (em *Filters*, defina o *Speed Grade* como *Any*). Clique *Next* e *Finish*.

Após finalizar esta etapa será possível observar as características do projeto na janela *Project Navigator* (observe que há três fichas).

- e) Clique na opção de menu **Processing** e selecione **Compiler Tool**. Nesta janela, usando o botão direito do mouse, ative a opção **Analysis & Synthesis**.
- f) Clique **Start Analysis and Synthesis**. Caso ocorram erros, corrija-os e repita a sequência de compilação até que o circuito compile sem erros.
- g) Salve o resultado.

Personalização do arquivo VHDL da FSM

Observe que o código padrão fornecido é composto por 3 processos: SEQ (para os sinais de *reset* e *clock*), COMB (para as transições de estados) e SAI (para as condições de saída).

A fim de criar o arquivo VHDL de seu controlador (FSM), modifique o arquivo padrão efetuando as seguintes tarefas:

- a) Verificar se todos os portos de entrada e saída apresentam os mesmos nomes que você adotou no seu projeto esquemático.
- b) Personalizar as condições de transição de estados no processo COMB;
- c) Personalizar as saídas no processo SAI. Atenção: por segurança, todas as saídas devem ter valores atribuídos para cada estado.
- d) Atribuir o nome ao módulo (entity) de acordo com o definido no arquivo topo *fibonacci_incompleto.vhd*.
- e) Salvar o seu trabalho dando ao seu arquivo o mesmo nome que o de sua entidade.

Verificação do arquivo VHDL da FSM

A fim de verificar se o arquivo VHDL personalizado está livre de erros de sintaxe, repita o procedimento utilizado para o *datapath*.

Personalização do arquivo VHDL do M_F

Observe que o código padrão fornecido é um *netlist* composto pelos 2 componentes cujas descrições comportamentais foram criadas.

A fim de criar o arquivo VHDL de seu processador M_F, modifique o arquivo padrão (*fibonacci_incompleto.vhd*), efetuando as seguintes tarefas:

- a) Verificar se todos os portos de entrada e saída apresentam os mesmos nomes que você adotou no seu projeto esquemático. Se houver discrepâncias, corrija.
- b) Personalizar os nomes dos portos na descrição dos componentes.
- c) Declarar todos os sinais internos (conexões entre os componentes de seu *datapath*) de acordo com o seu esquema RTL. Use os mesmos nomes que você adotou no seu projeto esquemático.
- d) Declarar as conexões entre os componentes e os portos da entidade.

- e) Salvar o seu trabalho dando ao seu arquivo o mesmo nome que o de sua entidade.

→ Para o relatório final:

DADOS EXPERIMENTAIS: Salvar os 3 códigos VHDL (após terem sido corretamente compilados) *datapath*, controlador e M_F.