



FEDERAL UNIVERSITY
OF SANTA CATARINA

Laboratório Introdução a VHDL

Parte 2: Tarefa de Projeto Hierárquico

EEL7123/EEL510457 – Tópicos Avançados de Sistemas Digitais/Circuitos
Aritméticos

Objetivos

- Entender o conceito de **Projeto Hierárquico**.
- Implementar um **Projeto Hierárquico** em **VHDL** usando **component** e **port map**.
- Aprender a usar **std_logic_vector** para representar vetores de variáveis booleanas.
- Realizar um **estudo de caso** visando fixar os conceitos e as estruturas estudadas.

Introdução

Tarefa

Emulação na placa DE2

Envio do material

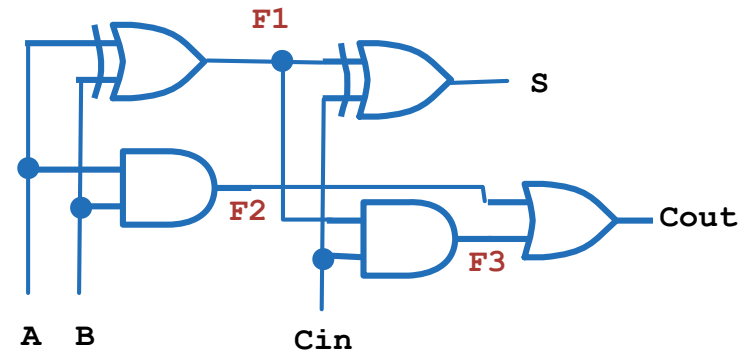
Introdução

- **Projeto Hierárquico**
 - Abordagem de projeto usada não somente em **VHDL**.
 - **Ideia**: compartimentalizar o projeto em múltiplos **componentes** que podem ser **criados separadamente** e depois **integrados** e **reutilizados**.
 - Facilita a **leitura**, **entendimento** e **manutenção** do código.

Introdução: Criação de *full-adder* em VHDL

ENTITY

```
entity fulladder is
port (A: in std_logic;
      B: in std_logic;
      Cin: in std_logic;
      S: out std_logic;
      Cout: out std_logic
    );
end fulladder;
```



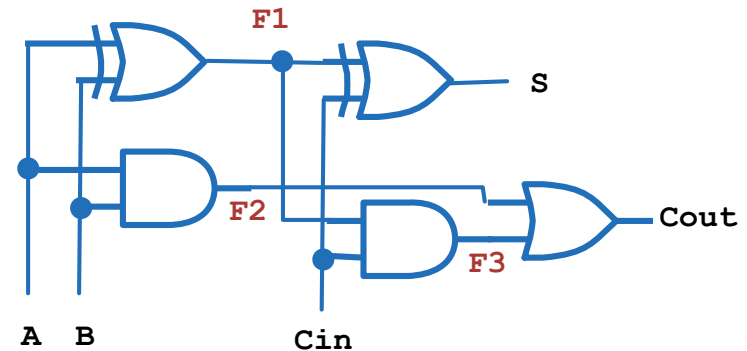
A	B	C _{in}	C _{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

ENTITY – define os “pinos” do circuito digital (sinais), ou seja, a **interface** entre a lógica implementada e o mundo externo.

Introdução: Criação de *full-adder* em VHDL

ARCHITECTURE

```
architecture circuito_logico of
fulladder is
signal F1, F2, F3: std_logic;
begin
    F1 <= A xor B;
    F2 <= A and B;
    S <= F1 xor Cin;
    F3 <= F1 and Cin;
    Cout <= F3 or Cin;
end circuito_logico;
```



ARCHITECTURE – define a funcionalidade do circuito digital, utilizando os “pinos” de entrada e saída listados na ENTITY em questão. Uma ENTITY pode possuir diversas implementações diferentes (diversas ARCHITECTURES).

Introdução: Criação de *full-adder* em VHDL

LIBRARIES

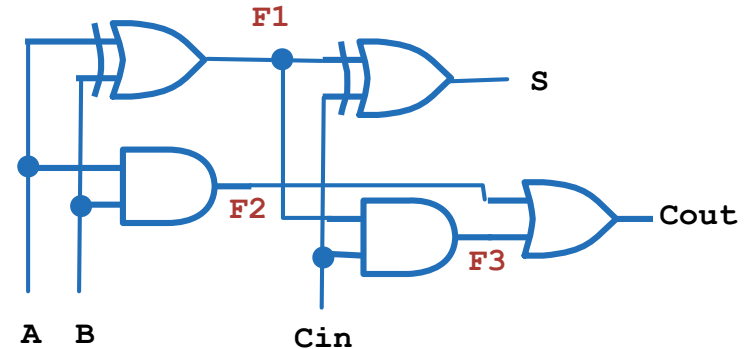
```
library IEEE;  
use IEEE.Std_Logic_1164.all;
```

ENTITY

```
entity fulladder is  
  port (A: in std_logic;  
        B: in std_logic;  
        Cin: in std_logic;  
        S: out std_logic;  
        Cout: out std_logic  
        );  
end fulladder;
```

ARCHITECTURE

```
architecture circuito_logico of fulladder is  
  signal F1, F2, F3: std_logic;  
begin  
  F1 <= A xor B;  
  F2 <= A and B;  
  S <= F1 xor Cin;  
  F3 <= F1 and Cin;  
  Cout <= F3 or F2;  
end circuito_logico;
```



Introdução: Somador de 8 bits como projeto hierárquico

LIBRARIES

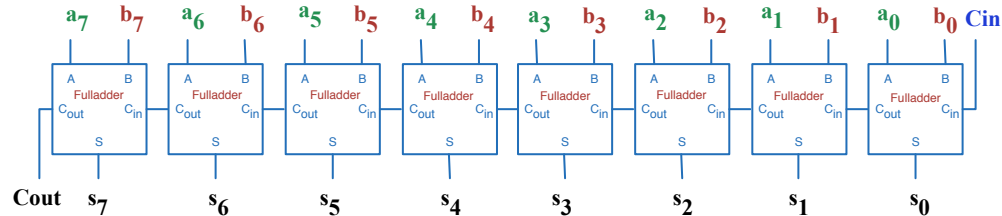
```
library IEEE;
use IEEE.Std_Logic_1164.all;
```

ENTITY

```
entity adder_8bits is
port (A: in std_logic_vector(7 downto 0);
      B: in std_logic_vector(7 downto 0);
      Cin: in std_logic;
      S: out std_logic_vector(8 downto 0));
end adder_8bits;
```

ARCHITECTURE

```
architecture circuito_logico of adder_8bits is
signal Cout: std_logic_vector(8 downto 0);
component fulladder is
port (A: in std_logic;
      B: in std_logic;
      Cin: in std_logic;
      S: out std_logic;
      Cout: out std_logic);
```



```
end component;
begin
Cout(0) <= Cin;
cpa_1 : for j in 0 to 7 generate
    cpa_j:    fulladder port map( A => A(j), B => B(j), Cin => Cout(j),
                                S => S(j) , Cout => Cout(j+1));
end generate cpa_1;
S(8) <= Cout(8);
end circuito_logico;
```


Introdução

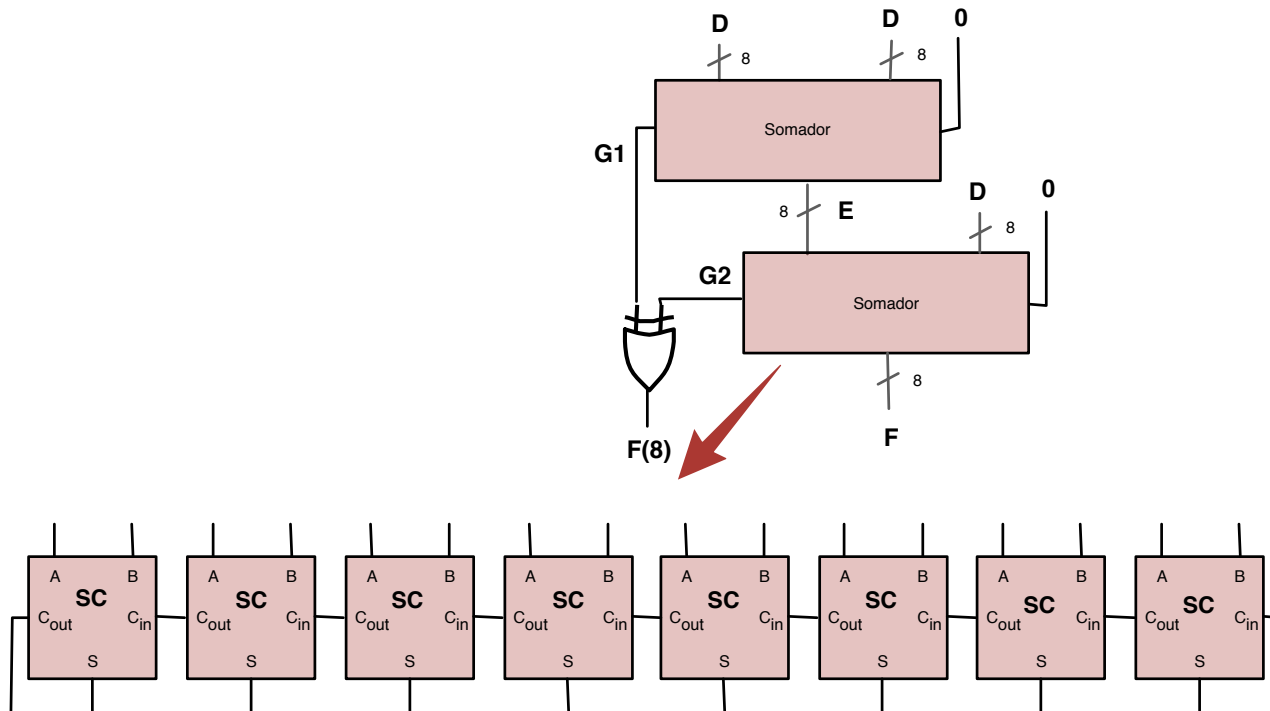
Tarefa

Emulação na placa DE2

Envio do material

Tarefa

- **Projeto Hierárquico**
 - Multiplicador por 3, ($F=3 \times D$), construído usando **somadores**, que por sua vez foram construídos com **full adders**



Tarefa

- **Projeto Hierárquico**
 - Descrição em **VHDL**:

```
library IEEE;
use IEEE.Std_Logic_1164.all;
entity mult3 is
port (D: in std_logic_vector(7 downto 0);
      F: out std_logic_vector(8 downto 0));
end mult3;
architecture mult3arch of mult3 is
  signal E: std_logic_vector(7 downto 0);
  signal G1, G2: std_logic;
  component adder_8bits is
port (A: in std_logic_vector(7 downto 0);
      B: in std_logic_vector(7 downto 0);
      Cin: in std_logic;
      S: out std_logic_vector(8 downto 0));
end component;
begin
  SUM1: adder_8bits port map (); -- a fazer pelo aluno
  SUM2: adder_8bits port map (); -- a fazer pelo aluno
  F(8) <= G1 xor G2;
end mult3arch;
```

Tarefa

- **Passo 1:** Criar projeto no **Quartus II**
 - Acessar **File -> New Project Wizard** e criar um projeto como feito na aula anterior.
 - Sugestão de nome do projeto: **IntroVHDL**
 - Dispositivo: **Pode ser qualquer um**
- **Passo 2:** Criar arquivos do tipo **VHDL** com a descrição dos componentes internos (**fulladder.vhd** e **adder_8bits.vhd**).
 - **File -> New -> Design Files -> VHDL File**
 - **File -> Save As** com nomes desejados
 - **Atenção:** nomes dos **arquivos** devem ser iguais aos das **entities**

Tarefa

- **Passo 3:** Criar outro arquivo **VHDL** com a descrição componente topo (**mult3.vhd**).
 - **File -> New -> Design Files -> VHDL File**
 - **File -> Save As** como **mult3.vhd**.
 - Com **somasub.vhd** aberto: **Project -> Set As Top Level Entity**
 - Complete o código dado no slide 11.
- **Passo 4:** Faça a **síntese** do circuito a partir de **Processing -> Start Compilation**



Tarefa

- **Passo 5:** Uma vez compilado e verificado que não tem erros de sintaxe **simular** o circuito no **ModelSim** da seguinte forma.

- a) A=60.
- b) A=170.
- c) A=200.

Faça **print** da simulação mostrando claramente as entradas e as saídas. **Explique os resultados obtidos quando coloque as fotos no Lab_Intro_VHDL.doc. Explique o que é o bit de saída F(8).**

- **Passo 6:** Visualize o diagrama de blocos gerado **Tools-> Netlist Viewers-> RTL Viewer**. Faça **print** do diagrama de blocos.

Introdução

Tarefa

Emulação na placa DE2

Envio do material

Emulação na placa DE2

Passo 7: Fazer a emulação na placa DE2 do **Multiplicador**

Com a emulação, o fluxo de **projeto→simulação→emulação** será completado. Os passos a seguir para a emulação se mostram a seguir:

7.1: Renomear o arquivo mult3.vhd como “usertop.vhd”.

7.2: A *entity* deve ser a seguinte:

```
entity usertop is
port(
  CLOCK_50:in std_logic;
  CLK_500Hz:in std_logic;
  RKEY:in std_logic_vector(3 downto 0);
  KEY:in std_logic_vector(3 downto 0);
  RSW:in std_logic_vector(17 downto 0);
  SW:in std_logic_vector(17 downto 0);
  LEDR:out std_logic_vector(17 downto 0);
  HEX0,HEX1,HEX2,HEX3,HEX4,HEX5,HEX6,HEX7: out std_logic_vector(6
downto 0));
end usertop;

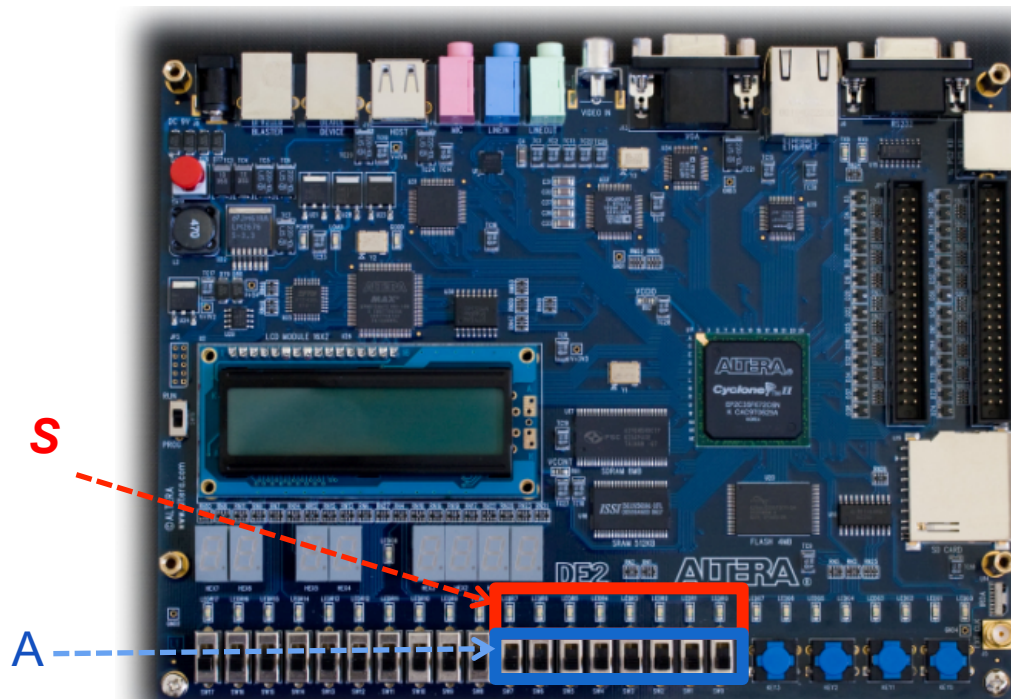
architecture circuito of usertop is
begin
...
```

←----- Não se esqueçam de trocar!

Emulação na placa DE2

Passo 7: Fazer a emulação na **DE2** do multiplicador.

7.3: Para esta tarefa teremos de associar **D(7 downto 0)** com **SW(7 downto 0)** e **D(8 downto 0)** com **LEDR(8 downto 0)**.



Emulação na placa DE2

Passo 7: Fazer a emulação na placa DE2 do **multiplicador de 3** usando **VPN** e **x2goclient**

7.4.a (Opção usando emulador via x2goclient/VPN): Com as devidas adaptações feitas nos arquivos abra o terminal de comandos (ver tutorial de acesso remoto ao Quartus no Moodle) e digite:

```
cd /path
```

onde “path” é o caminho até os arquivos VHDL que deseja utilizar na emulação. Em seguida, compile com o comando (os .vhd nessa ordem):

```
fpgacompile fulladder.vhd adder_8bits.vhd usertop.vhd
```

Para emular o circuito, na mesma pasta, digite

```
./fpgatest
```

Assim que fizer isso, uma janela com a emulação do seu circuito abrirá. Quando desejar terminar a emulação, feche a janela e, em seguida, use Ctrl+C para liberar o terminal.

Ao final da utilização, não se esqueça de efetuar o logout no X2Go Client.

Emulação na placa DE2

Passo 7: Fazer a emulação na placa DE2 do **multiplicador de 3** usando o **emulador on-line** (sem necessidade de ter o VPN da UFSC ligado).

7.4.b (Opção usando emulador on-line): Com as devidas adaptações feitas nos arquivos abra o emulador <http://150.162.54.54:5000/> e faça **UPLOAD** dos arquivos VHDL que deseja utilizar na emulação. Defina *usertop.vhd* como topo da hierarquia usando o botão **SET TOP LEVEL** e compile. Em seguida, vai para o emulador, inicie a emulação e verifique o funcionamento.

7.5: Faça **print** ou **fotografia** de A=60, A=170 e A=200. **Explique os resultados obtidos no Lab_Intro_VHDL.doc.**

Introdução

Tarefa

Emulação na placa DE2

Envio do material

Envio do material

Passo 8: Os alunos deverão entregar na tarefa disponível no Moodle com:

- Os arquivos **VHDLs** da tarefa.
- O arquivo **Lab_Intro_VHDL.doc** incluindo o *print* de tela da **simulação** do passo 5, do **diagrama de blocos** do passo 6, os *prints* de tela da **emulação** e as dúvidas que tiveram (caso existam). **O aluno deve explicar os resultados obtidos na simulação e emulação.** O aluno pode incluir na tarefa de forma opcional um vídeo mostrando o funcionamento no emulador para combinações pedidas.