



Universidade Federal de Santa Catarina
Centro Tecnológico – CTC
Departamento de Engenharia Elétrica



“Introdução a circuitos Combinatórios em VHDL ”

Prof. Héctor Pettenghi Roldán*

Hector@eel.ufsc.br

Florianópolis, Agosto de 2019.

Plano de Aula

“Projeto de Sistemas Digitais com VHDL”

- **Objetivos:**
 - Apresentar uma visão geral de VHDL
 - Exemplo de descrição VHDL
 - Introdução ao Quartus II – ferramentas de desenvolvimento
 - Estudo de caso / exercício

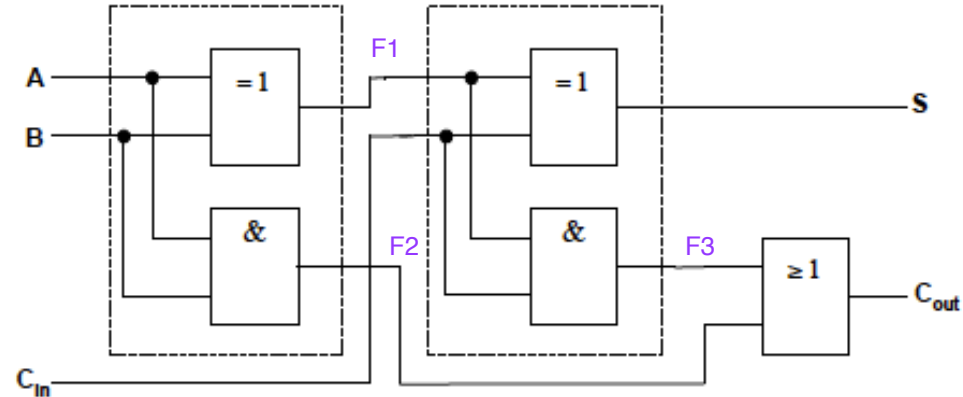
VHDL - Visão Geral

- **VHDL = VHSIC Hardware Description Language**
- **VHSIC = Very High Speed Integrated Circuits. Programa do governo dos USA do início dos anos 80. No final da década de 80, VHDL se tornou um padrão IEEE (Institute of Electrical and Electronic Engineers).**
- **Existem diversas ferramentas para simular e sintetizar (gerar hardware) circuitos descritos em VHDL.**
- **Outras linguagens de descrição de hardware: Verilog, SystemC, AHDL, Handel-C, System Verilog, Abel, Ruby, ...**
- **O projeto de um circuito digital pode ser descrito em VHDL em diversos níveis de abstração.**
- **Descrições em VHDL podem ser simuladas (executadas em um simulador).**
- **Descrições em VHDL podem ser utilizadas para gerar um hardware (arquivo para configuração de um FPGA, por exemplo).**
- **A geração de estímulos para simulação VHDL é realizada por intermédio de testbenches.**

Descrição de circuito digital em VHDL

ENTITY

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

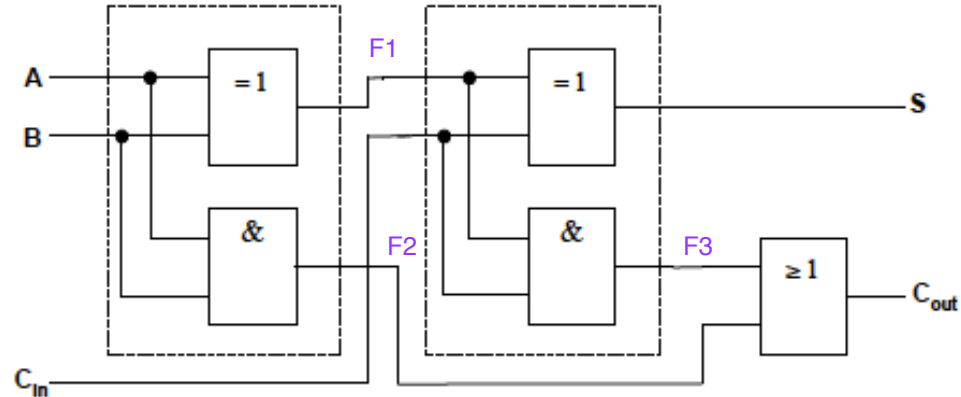


```
entity fulladder is
port (A: in std_logic;
      B: in std_logic;
      Cin: in std_logic;
      S: out std_logic;
      Cout: out std_logic
      );
end fulladder;
```

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

Descrição de circuito digital 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).

Descrição completa de *full adder* em VHDL

LIBRARIES

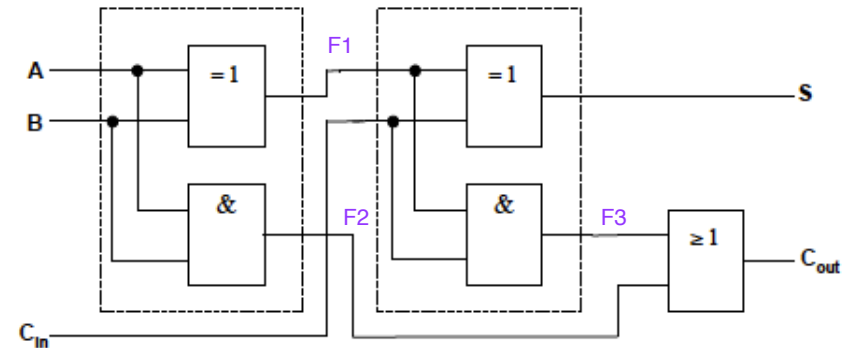
```
library IEEE;  
use IEEE.Std_Logic_1164.all;
```

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;
```



Descrição completa do somador de 8 bits em VHDL

LIBRARIES

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

ENTITY

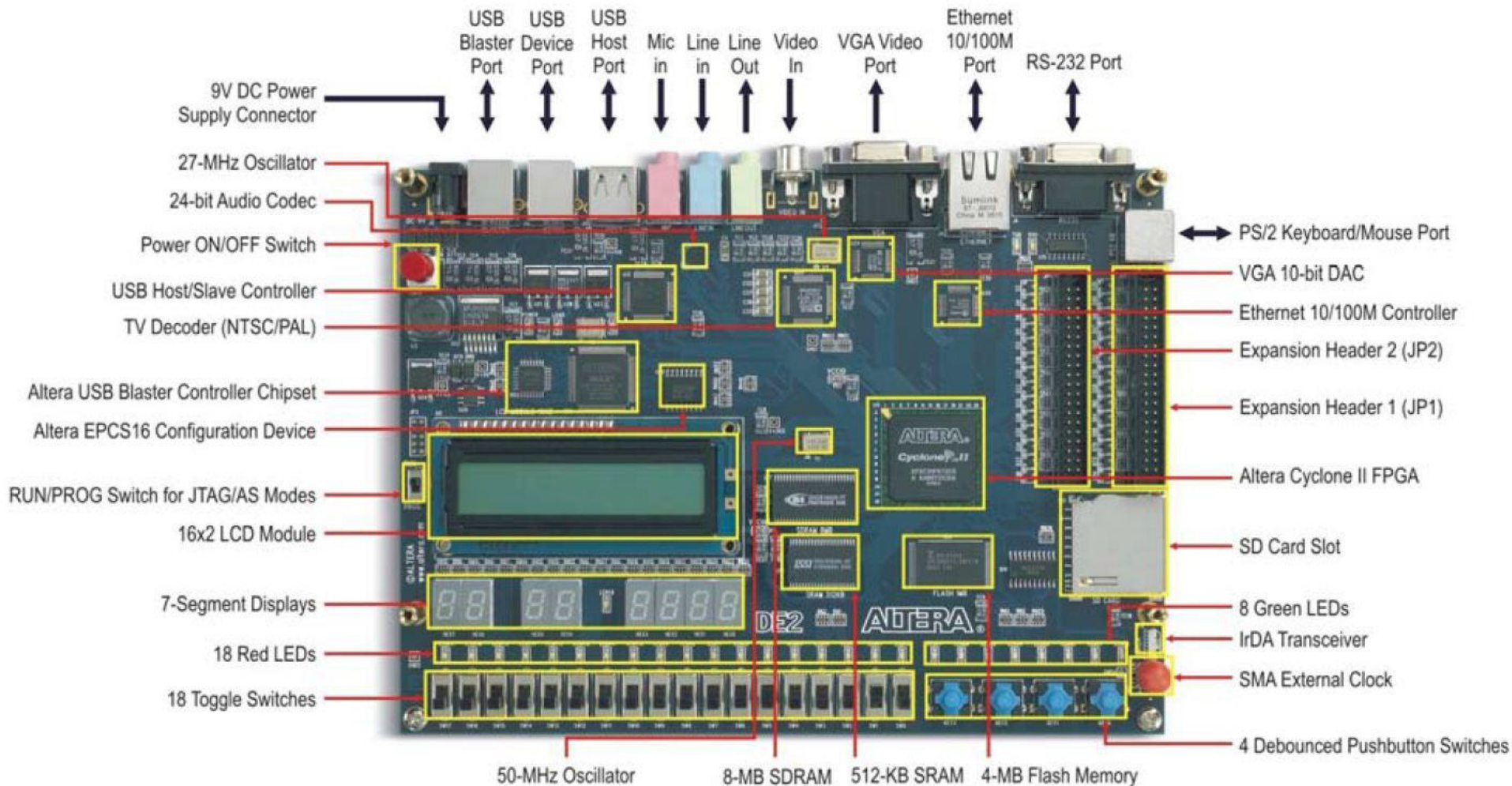
```
entity adder_8bits is  
  port (A: in std_logic_vector(7 downto 0);  
        B: in std_logic_vector(7 downto 0);  
        Cin: in std_logic;  
        S: out std_logic_vector(8 downto 0));  
end adder_8bits;
```

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;
```

Plataforma de prototipação FPGA Altera – DE2-SoC

Kit DE2-SoC da Altera



Interface com o usuário (entrada e saída)

- Placa DE2-SoC possui 18 LEDs vermelhos denominados LEDR₁₇₋₀, 8 LEDs verdes denominados LEDG₇₋₀ e 17 chaves denominadas SW₁₇₋₀
- As conexões entre esses componentes e os pinos do FPGA da placa estão definidas no arquivo *DE2_pin_assignments.qsf*, disponível no site da disciplina
- São utilizados “vetores” para facilitar o acesso aos LEDs e chaves da placa
- Exemplo: SW[0] é o elemento 0 do vetor SW.
- No código em VHDL, usar sempre os nomes definidos no arquivo *DE2_pin_assignments.qsf*.

Interface com o usuário (entrada e saída)

Código VHDL para somador de 8 bits com “leitura” das chaves e “escrita” nos LEDs

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

entity adder_8bits is
port (SW: in std_logic_vector(17 downto 0);
      LEDR: out std_logic_vector(17 downto 0);
      LEDG: out std_logic_vector(7 downto 0));
end adder_8bits;

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) <= SW(16);
cpa_1 : for j in 0 to 7 generate
    cpa_j:    fulladder port map( A => SW(j), B => SW(j+8), Cin => Cout(j),
                                S => LEDR(j) , Cout => Cout(j+1));
end generate cpa_1;
LEDR(8) <= Cout(8);
end circuito_logico;
```

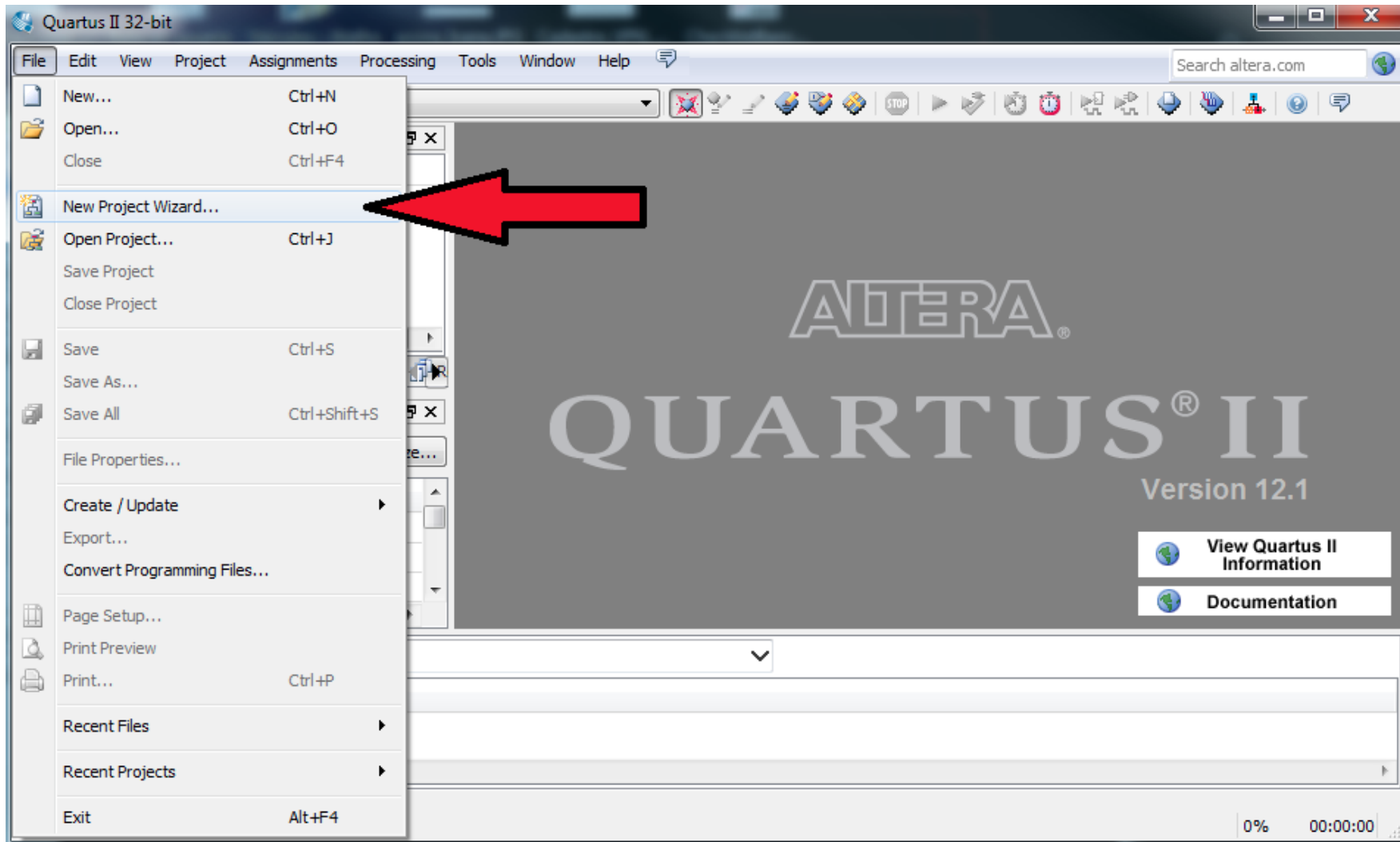
Tarefa a ser realizada na aula prática

Tarefa a ser realizada na aula prática

1. Utilizando a ferramenta Quartus II da Altera, criar um projeto VHDL que implemente o circuito apresentado no **slide 11** incluindo a componente *full adder* do **slide 6**.
2. Realizar a simulação do circuito (VHDL) por intermédio de diagramas de formas de onda.
3. Visando fixar o conhecimento do fluxo de ferramentas de projeto.
4. Testar o circuito no kit DE2-SoC, usando as chaves SW(16 downto 0) para entrar com os operandos, e observar os resultados nos LEDs LEDR(8 downto 0).

Resumo do tutorial: ***Etapas 1 - Design Entry***

1. [Quartus II] File -> New Project Wizard



Resumo do tutorial: ***Etapas 1 - Design Entry***

2. Escolher a pasta e nome do projeto

Directory, Name, Top-Level Entity

Pasta existente na área de trabalho

What is the working directory for this project?

...\Desktop\510228\

What is the name of this project?

adder_8bits

What is the name of the top-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

adder_8bits

Use Existing Project Settings...

Help < Back Next > Finish Cancel

Resumo do tutorial: *Etapas 1 - Design Entry*

3. Selecionar o dispositivo alvo FPGA

New Project Wizard

Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.

Device family

Family: Cyclone II

Devices: All

Target device

☐ Auto device selected by the Fitter

☒ Specific device selected in 'Available devices' list

☐ Other: n/a

Show in 'Available devices' list

Package: Any

Pin count: Any

Speed grade: Any

Name filter:

☒ Show advanced devices ☐ HardCopy compatible only

Available devices:

Name	Core Voltage	LEs	User I/Os	Memory Bits	Embedded multiplier 9-bit elements	PLL	al Cl
EP2C35F484C7	1.2V	33216	322	483840	70	4	16
EP2C35F484C8	1.2V	33216	322	483840	70	4	16
EP2C35F484I8	1.2V	33216	322	483840	70	4	16
EP2C35F672C6	1.2V	33216	475	483840	70	4	16
EP2C35F672C7	1.2V	33216	475	483840	70	4	16
EP2C35F672C8	1.2V	33216	475	483840	70	4	16
EP2C35F672I8	1.2V	33216	475	483840	70	4	16
EP2C35F1284C6	1.2V	33216	322	483840	70	4	16

Companion device

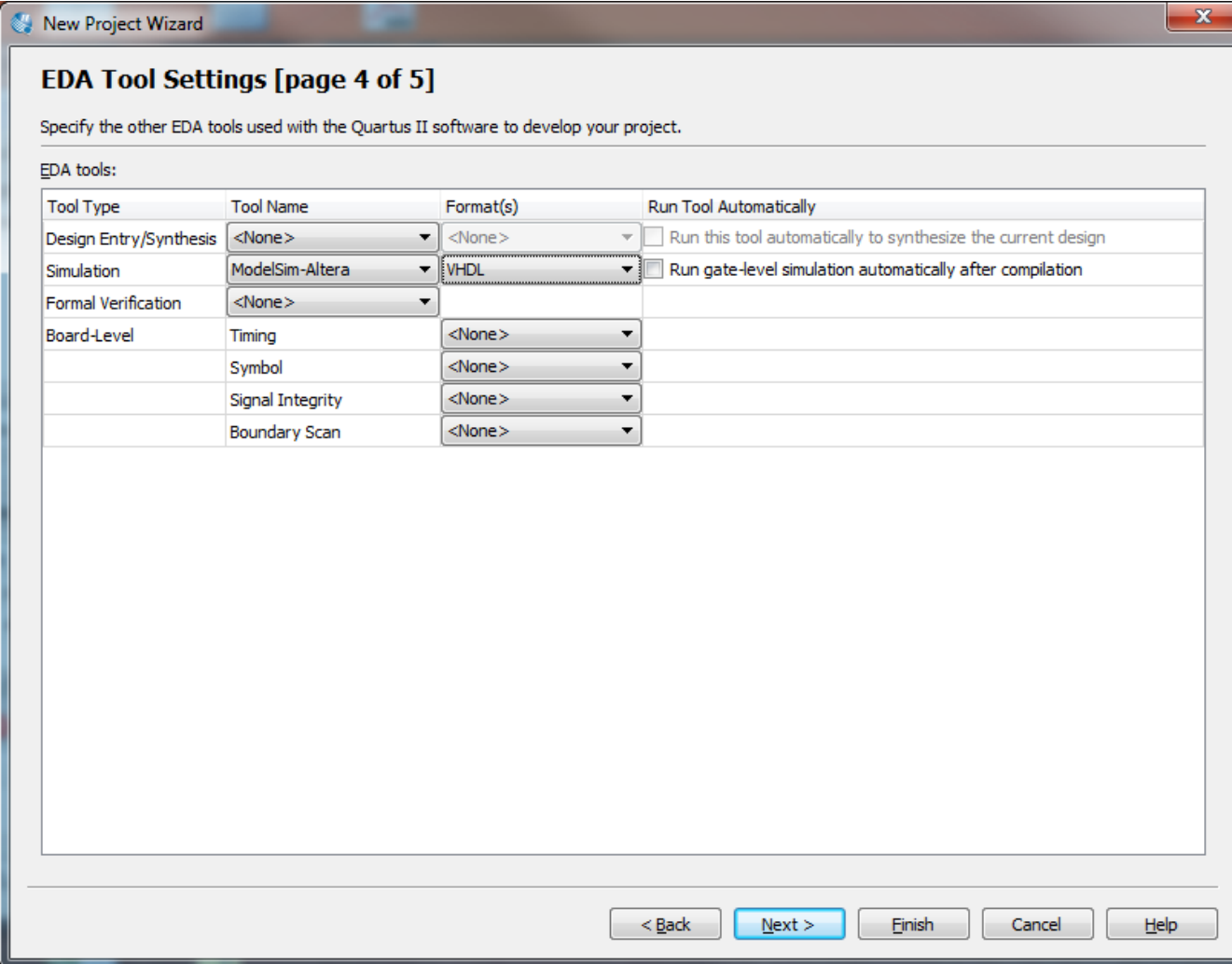
HardCopy:

☐ Limit DSP & RAM to HardCopy device resources

< Back Next > Finish Cancel Help

Resumo do tutorial: *Etapa 1 - Design Entry*

4. Configurar a ferramenta de simulação



New Project Wizard

EDA Tool Settings [page 4 of 5]

Specify the other EDA tools used with the Quartus II software to develop your project.

EDA tools:

Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synthesis	<None>	<None>	<input type="checkbox"/> Run this tool automatically to synthesize the current design
Simulation	ModelSim-Altera	VHDL	<input type="checkbox"/> Run gate-level simulation automatically after compilation
Formal Verification	<None>		
Board-Level	Timing	<None>	
	Symbol	<None>	
	Signal Integrity	<None>	
	Boundary Scan	<None>	

< Back Next > Finish Cancel Help

Resumo do tutorial: *Etapa 1 - Design Entry*

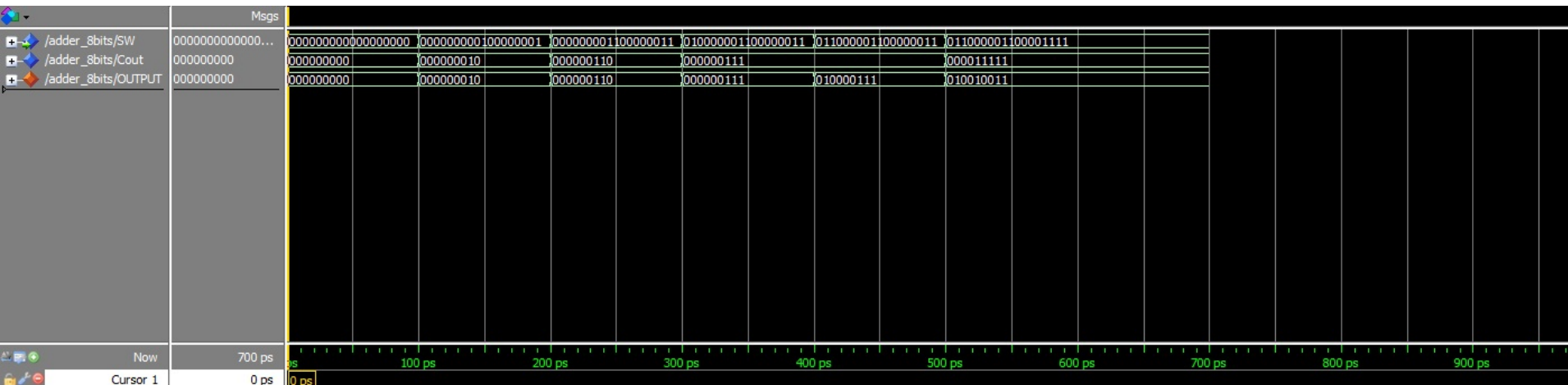
5. File -> New -> VHDL File.

6. Copiar as fontes VHDL dos slide 6 e 11 para novos arquivos, e salvar.

Resumo do tutorial: *Etapa 2 - Simulação*

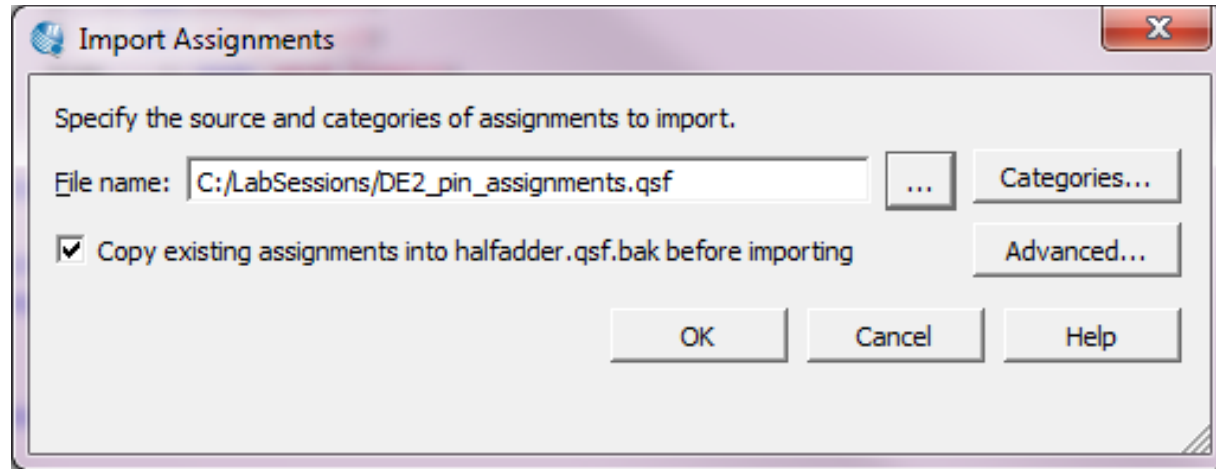
7. [**ModelSim**] Simulação Funcional – Teste do circuito
-> não considera informação de temporização.

8. Resultado esperado da simulação:



Resumo do tutorial: *Etapa 3 – prototipação FPGA*

9. Assignments -> Import Assignments (procurar no site e usar o arquivo DE2_pin_assignments.qsf)



10. Com isso, os pinos do FPGA foram associados aos sinais da entity do VHDL

11. Compilar o VHDL (síntese)

12. ATENÇÃO!!! Verificar se o nome da entity é o mesmo nome do projeto, para evitar erros na síntese.

13. A compilação resulta em dezenas de warnings devido aos pinos não conectados do arquivo .qsf

Resumo do tutorial: *Etapa 3 – prototipação FPGA*

14. Programação – FPGA é carregado com circuito, configurando fisicamente elementos de processamento e roteamento.

Tools – Programmer. Hardware Setup – USB-Blaster. Start!

