



Circuitos Digitais II - 6882

André Barbosa Verona
Nardênio Almeida Martins

Universidade Estadual de Maringá
Departamento de Informática

Bacharelado em Ciência da Computação

Aula de Hoje

a) Projeto e Simulação das seguintes funções lógicas:

1. AND
2. OR
3. NAND
4. XNOR

Obs 1: Cada função deve ser implementada em um código separado

Obs 2: As funções devem usar modelagem comportamental

Aula de Hoje

b) Projeto e Simulação das seguintes funções lógicas num único código:

- 1. AND**
- 2. OR**
- 3. NAND**
- 4. XNOR**

Obs 1: Todas as funções devem ser implementadas em um único código

Obs 2: As funções devem usar modelagem comportamental

Fundamentos de Lógica

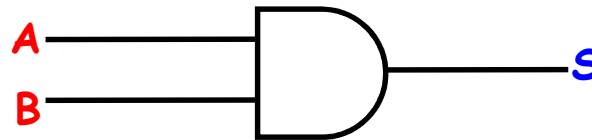
Tabela Verdade

TV da Porta AND

Entradas		Saída
A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

Função AND Representação: $S = A.B$

Símbolo da Porta AND



Modelagem Comportamental em VHDL:

Se a variável de entrada A for igual a 1 e a variável de entrada B for igual a 1, então a saída S será igual a 1, senão a saída S será igual a 0.

Fundamentos de Lógica

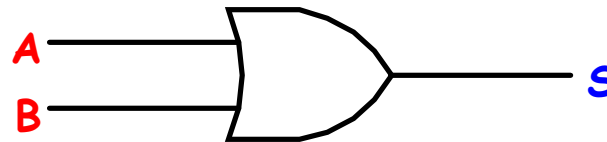
Tabela Verdade

TV da Porta OR

Entradas		Saída
A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

Função OR Representação: $S = A+B$

Símbolo da Porta OR



Modelagem Comportamental em VHDL:

Se a variável de entrada A for igual a 0 e a variável de entrada B for igual a 0, então a saída S será igual a 0, senão a saída S será igual a 1.

Fundamentos de Lógica

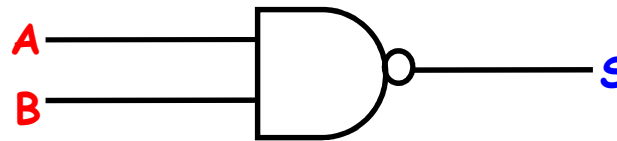
Tabela Verdade

TV da Porta NAND

Entradas		Saída
A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

Função NAND Representação: $S = \overline{A.B}$

Símbolo da Porta NAND



Modelagem Comportamental em VHDL:

Se a variável de entrada A for igual a 1 e a variável de entrada B for igual a 1, então a saída S será igual a 0, senão a saída S será igual a 1.

Fundamentos de Lógica

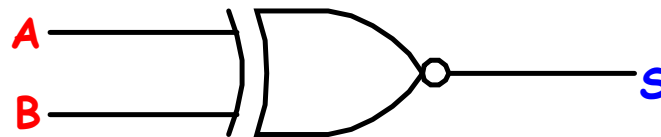
Tabela Verdade

TV da Porta XNOR

Entradas		Saída
A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

Função XNOR Representação: $S = \overline{A \oplus B} = A \odot B$

Símbolo da Porta XNOR



Modelagem Comportamental em VHDL:

Se as variáveis de entrada A e B forem iguais, então a saída S será igual a 1, senão a saída S será igual a 0.

HDL - Linguagem de Descrição de Hardware

Estrutura Básica de um Código em VHDL

LIBRARY IEEE; USE IEEE.STD_LOGIC_1164.all; USE IEEE.STD_LOGIC_UNSIGNED.all;	LIBRARY (PACOTES)
ENTITY exemplo IS PORT (<descrição dos pinos de I/O>); END exemplo;	ENTITY (PINOS DE I/O)
ARCHITECTURE teste OF exemplo IS BEGIN ... END teste;	ARCHITECTURE (ARQUITETURA)

HDL - Linguagem de Descrição de Hardware

Estrutura do Código

-- Modelagem Comportamental em VHDL:

-- Função AND

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;  
  
ENTITY porta_and IS  
    PORT (a, b : IN BIT;  
          f : OUT BIT);  
END porta_and;
```

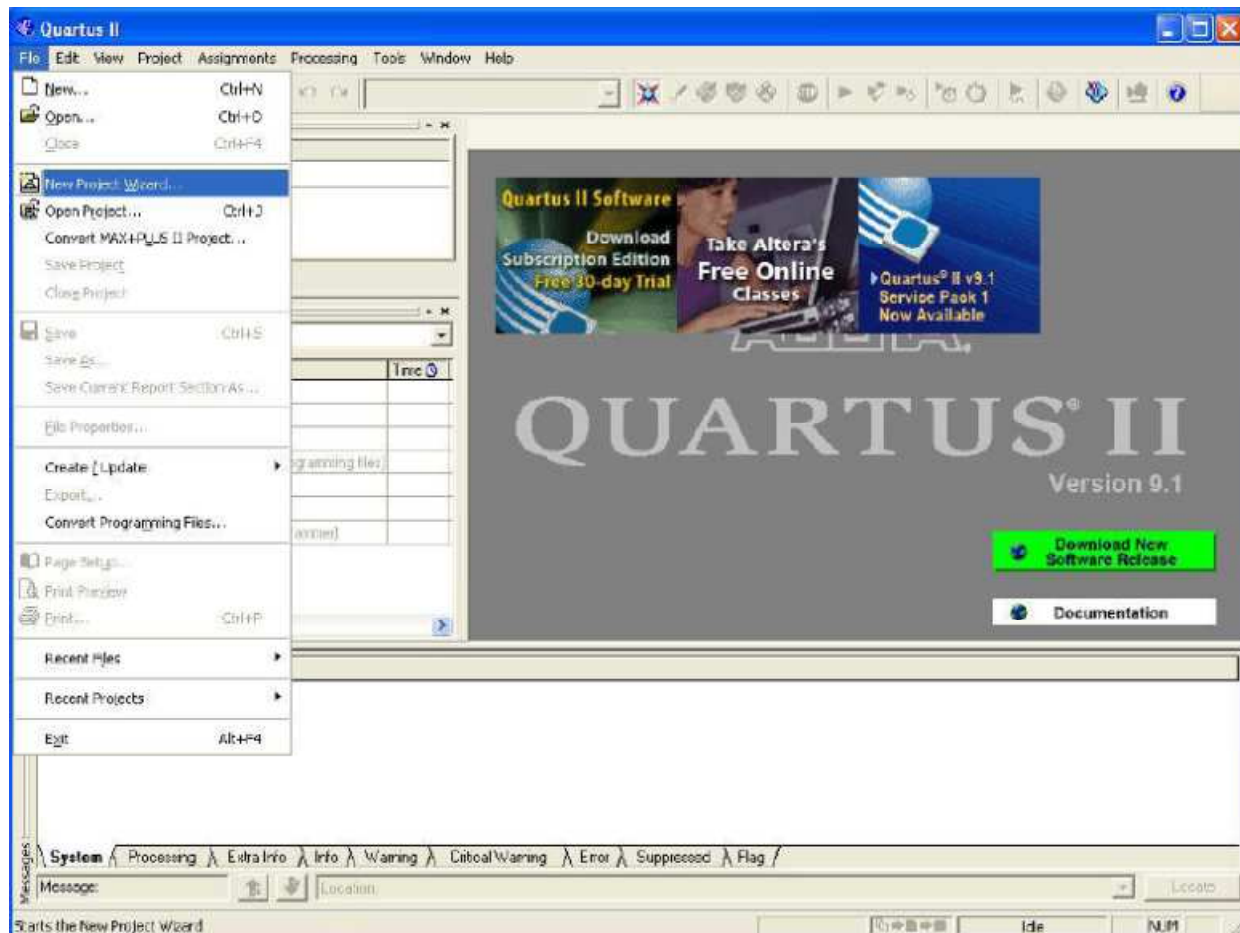
```
ARCHITECTURE logica OF porta_and IS  
BEGIN  
    PROCESS (a, b)  
    BEGIN  
        IF (a='1') AND (b='1') THEN  
            f <= '1';  
        ELSE  
            f <= '0';  
        END IF;  
    END PROCESS;  
END logica;
```

Software Quartus II

1. Crie diretório ou pasta **"work"** na área de trabalho.
2. Crie os seguintes subdiretórios dentro do diretório **"work"**:
 - a) **"and_2"**
 - b) **"or_2"**
 - c) **"nand_2"**
 - d) **"xnor_2"**
 - e) **"portas_2"**
3. Inicialize o Software **Quartus II**

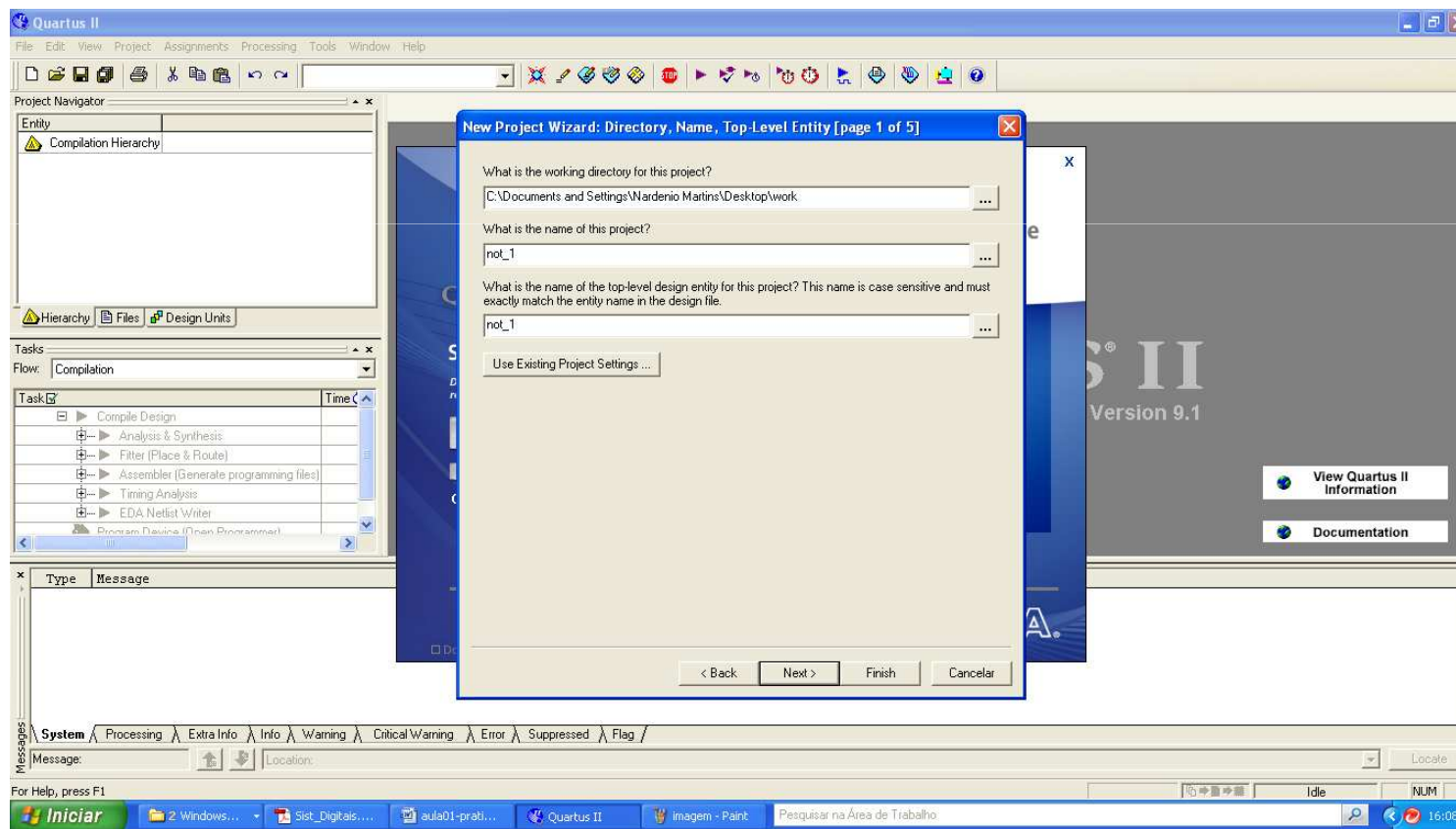
Software Quartus II

4. Crie um novo projeto: selecione "File > New Project Wizard"



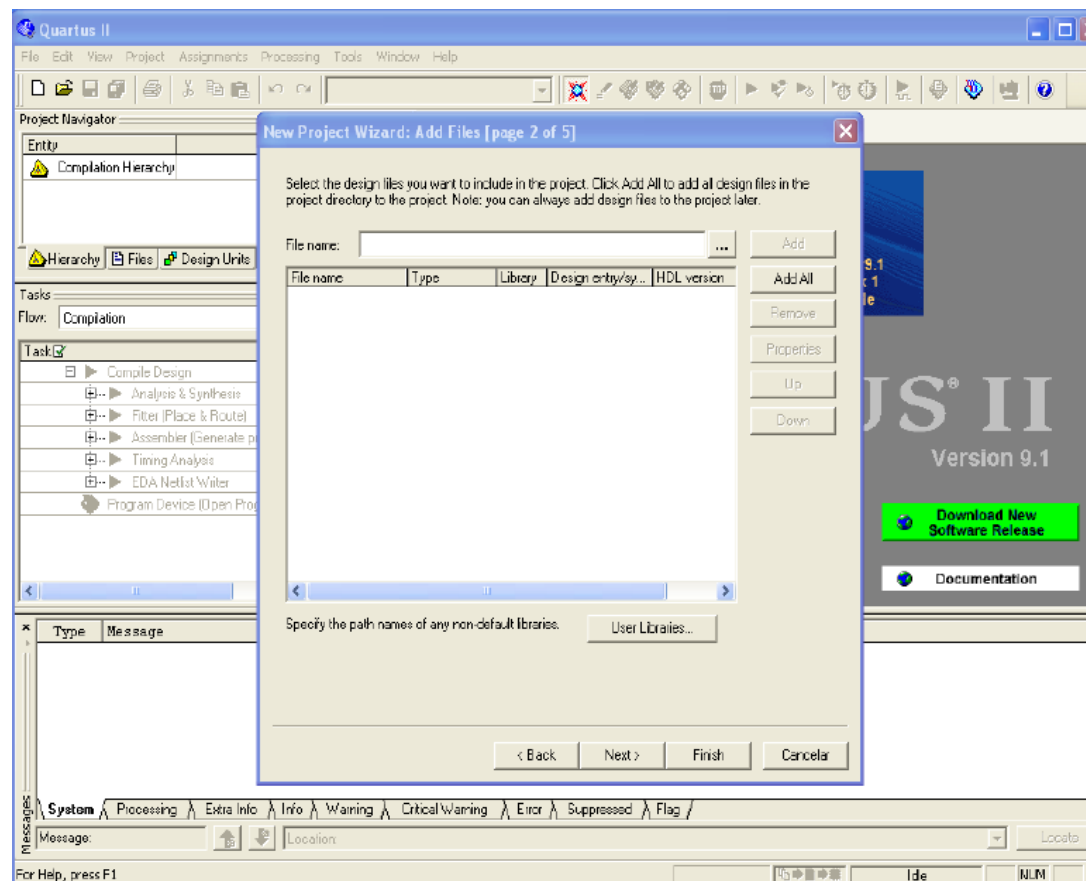
Software Quartus II

5. Na primeira linha da janela, insira o caminho e o nome do diretório do projeto → **"work"**. Na segunda linha insira o nome do projeto → **"and_2"**.



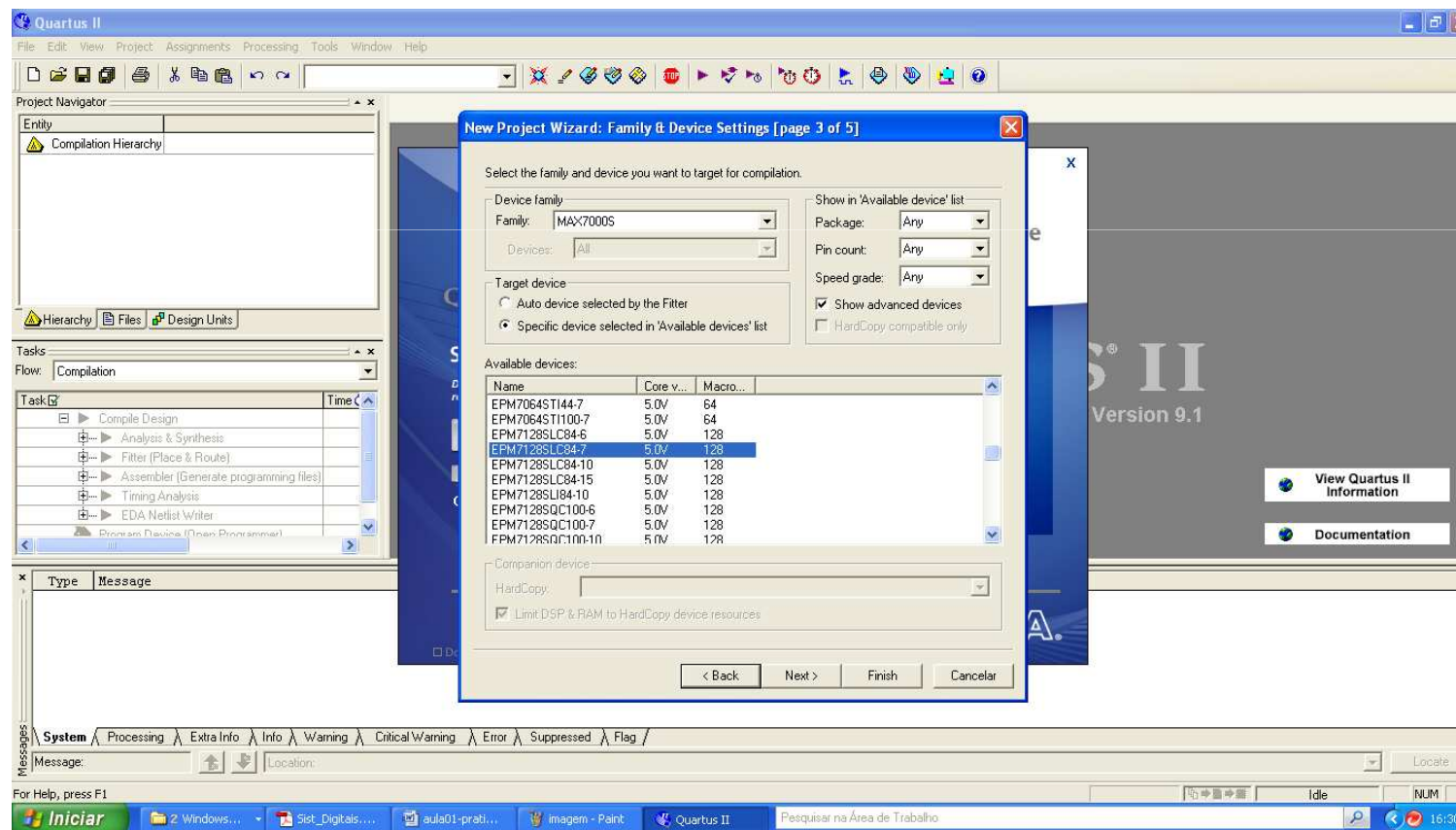
Software Quartus II

6. Pressione "**Next**". O projetista pode incluir arquivos de outros projetos, ou mesmo aqueles que estão nas "*Libraries*" do software Quartus II.



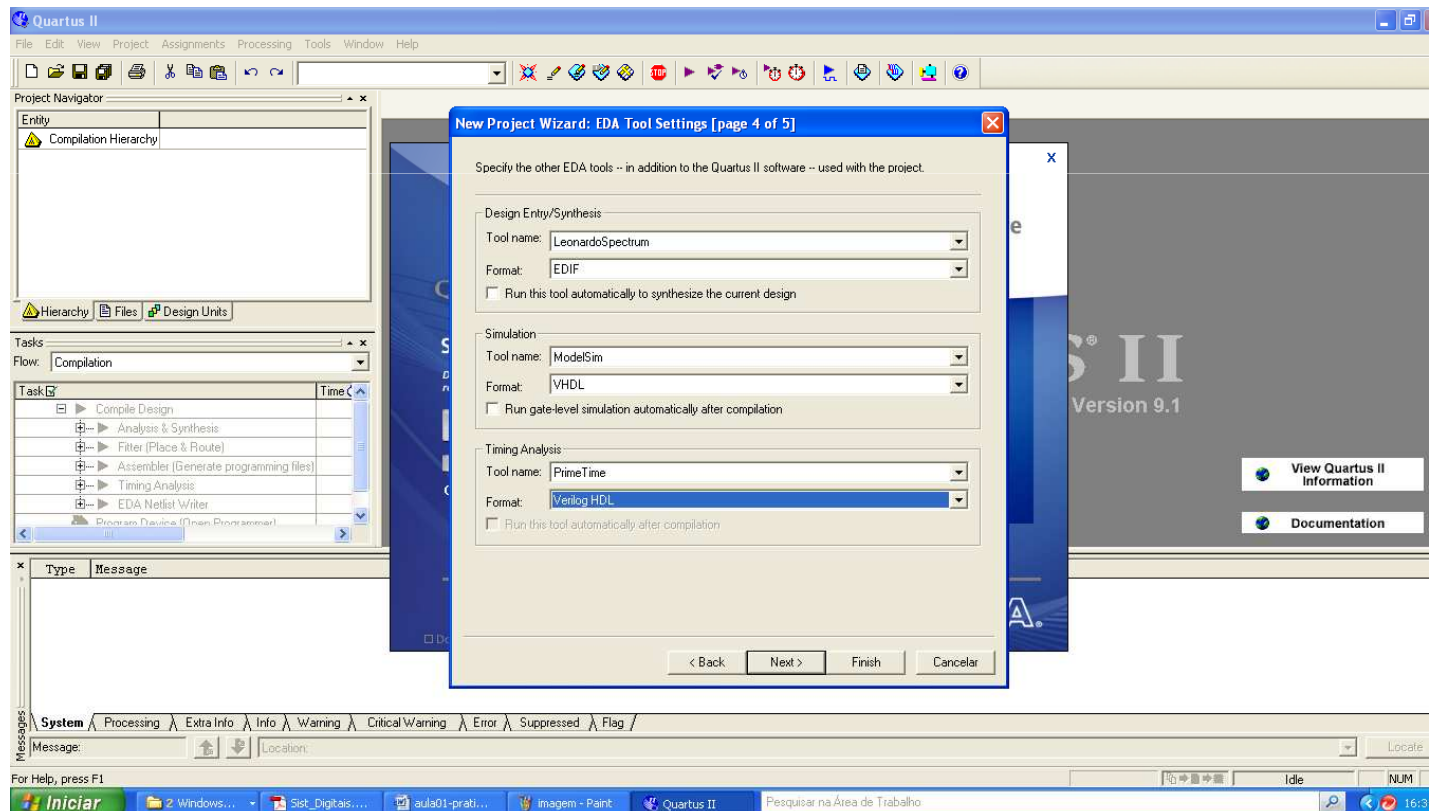
Software Quartus II

7. Selecione o dispositivo lógico programável a ser utilizado. Neste caso é usado o CPLD da família "MAX7000S", denominado "EPM7128SLC84-7".



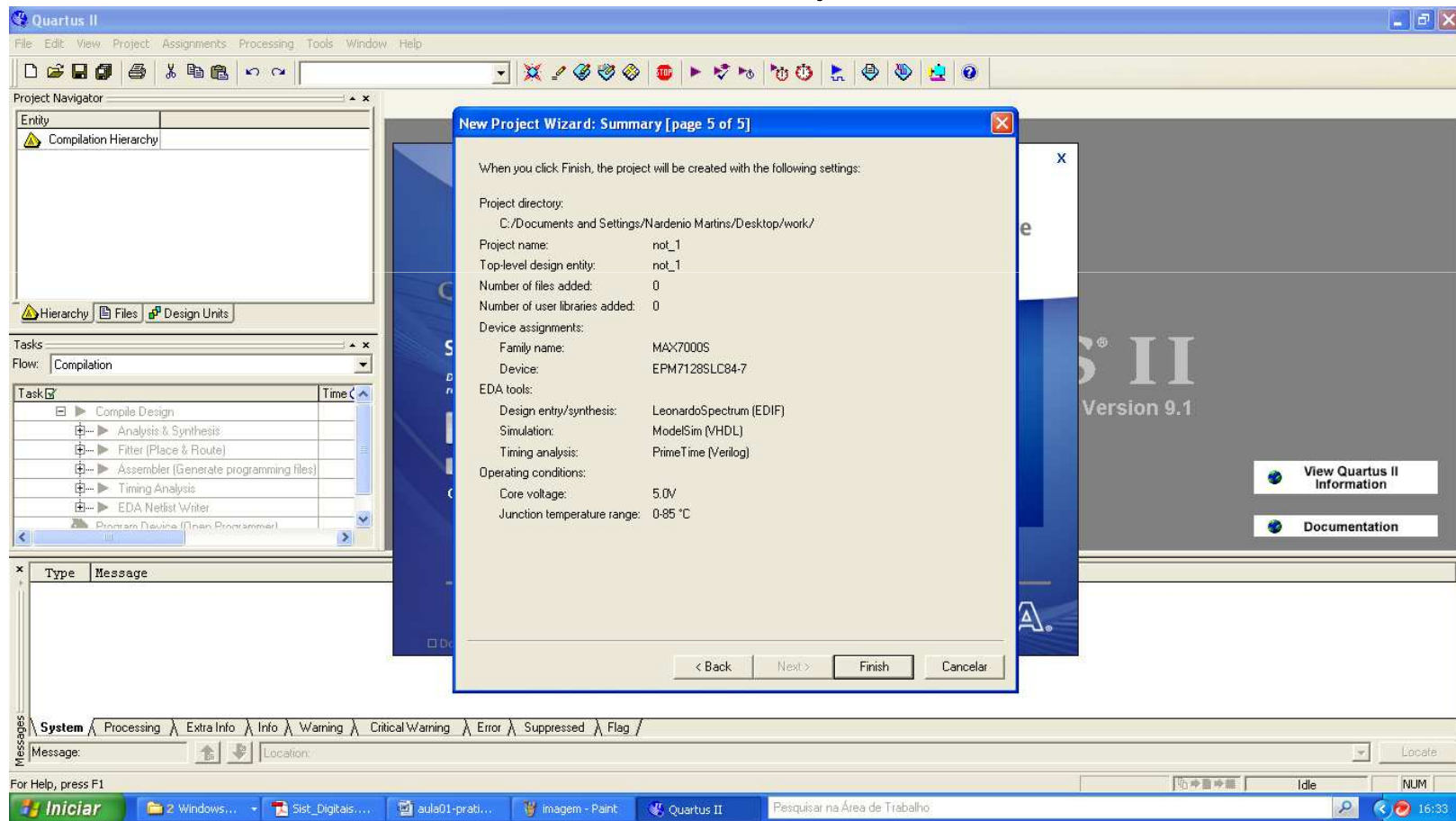
Software Quartus II

8. O próximo passo permite a adição de outras ferramentas como "LeonardoSpectrum" e "EDIF", "ModelSim" e "VHDL", "PrimeTime" e "Verilog HDL" que possibilita a interação entre FPGA e ASIC.



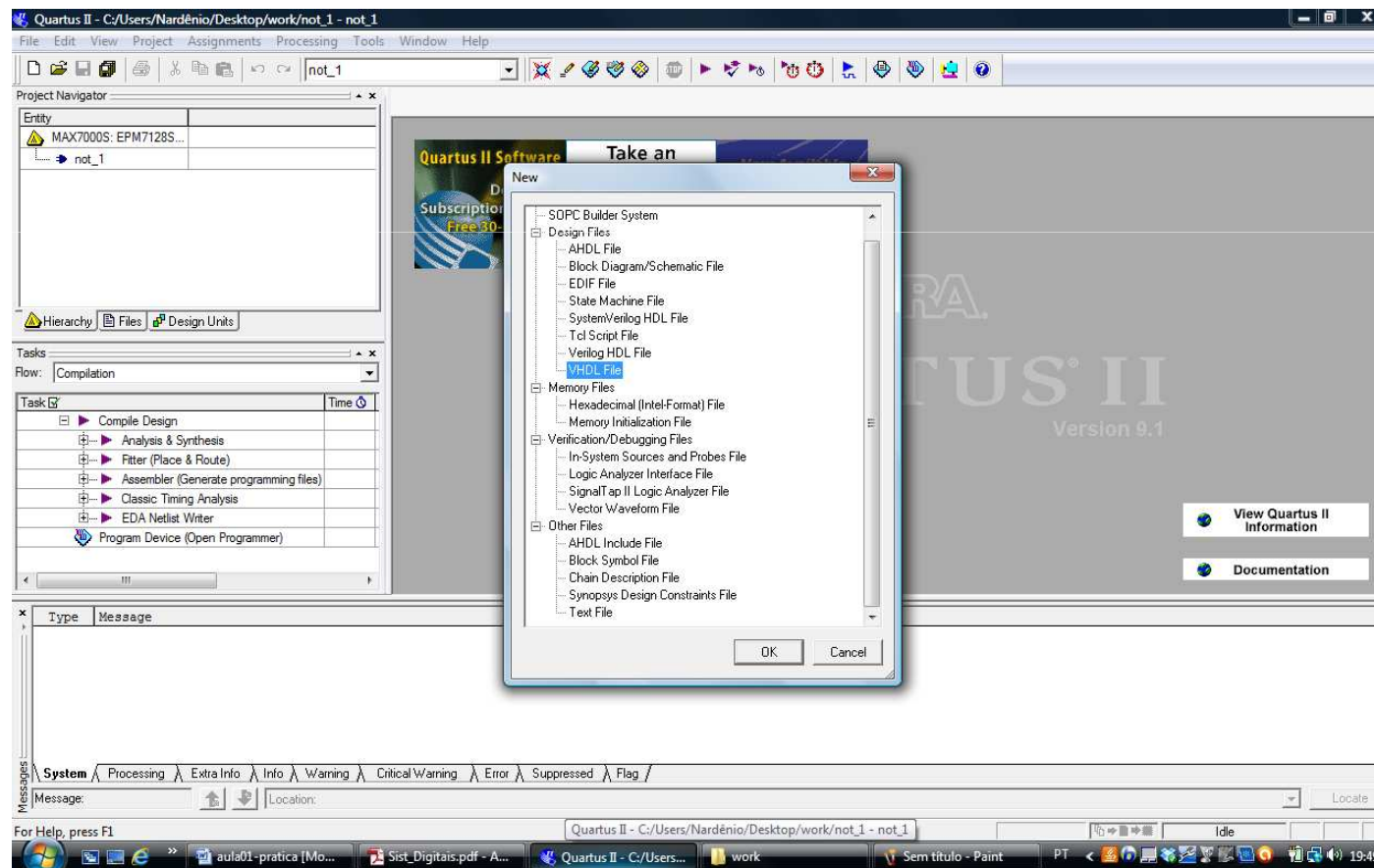
Software Quartus II

9. O último passo apresenta um resumo do projeto a ser executado. Posteriormente, clique em **Finish**.



Software Quartus II

10. Defina o modo a ser utilizado para desenvolver o projeto: AHDL, VHDL ou Block Diagram/Schematic File. Selecione "File > New" e escolha "VHDL file".



Software Quartus II

11. Escreva o código em VHDL da função AND.

-- Modelagem Comportamental em VHDL:

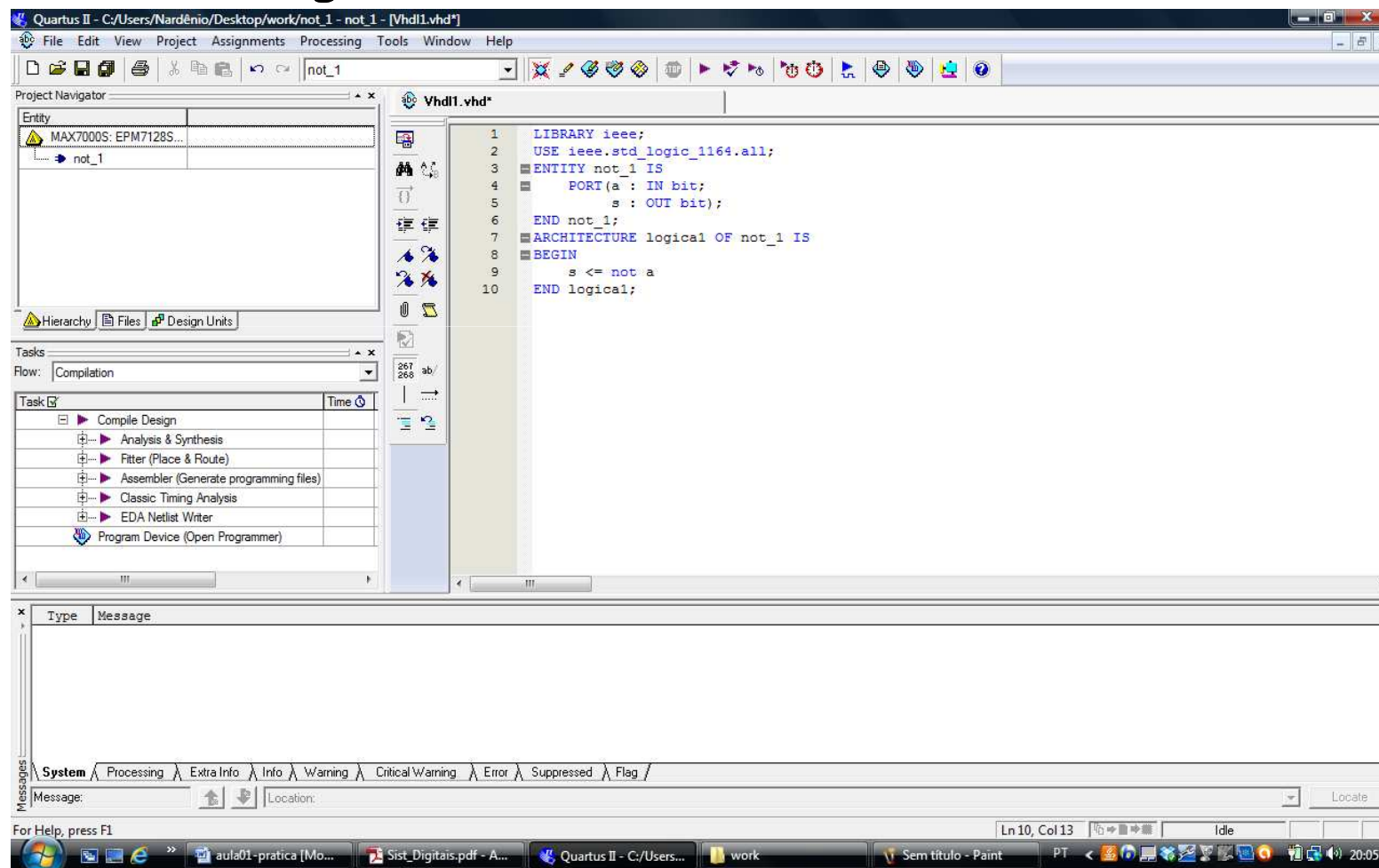
-- Função AND

```
LIBRARY ieee;  
USE ieee.std_logic_1164.all;  
  
ENTITY and_2 IS  
    PORT (a, b : IN BIT;  
          f : OUT BIT);  
END and_2;
```

```
ARCHITECTURE comportamental OF and_2 IS  
BEGIN  
    PROCESS (a, b)  
    BEGIN  
        IF (a='1') AND (b='1') THEN  
            f <= '1';  
        ELSE  
            f <= '0';  
        END IF;  
    END PROCESS;  
END comportamental;
```

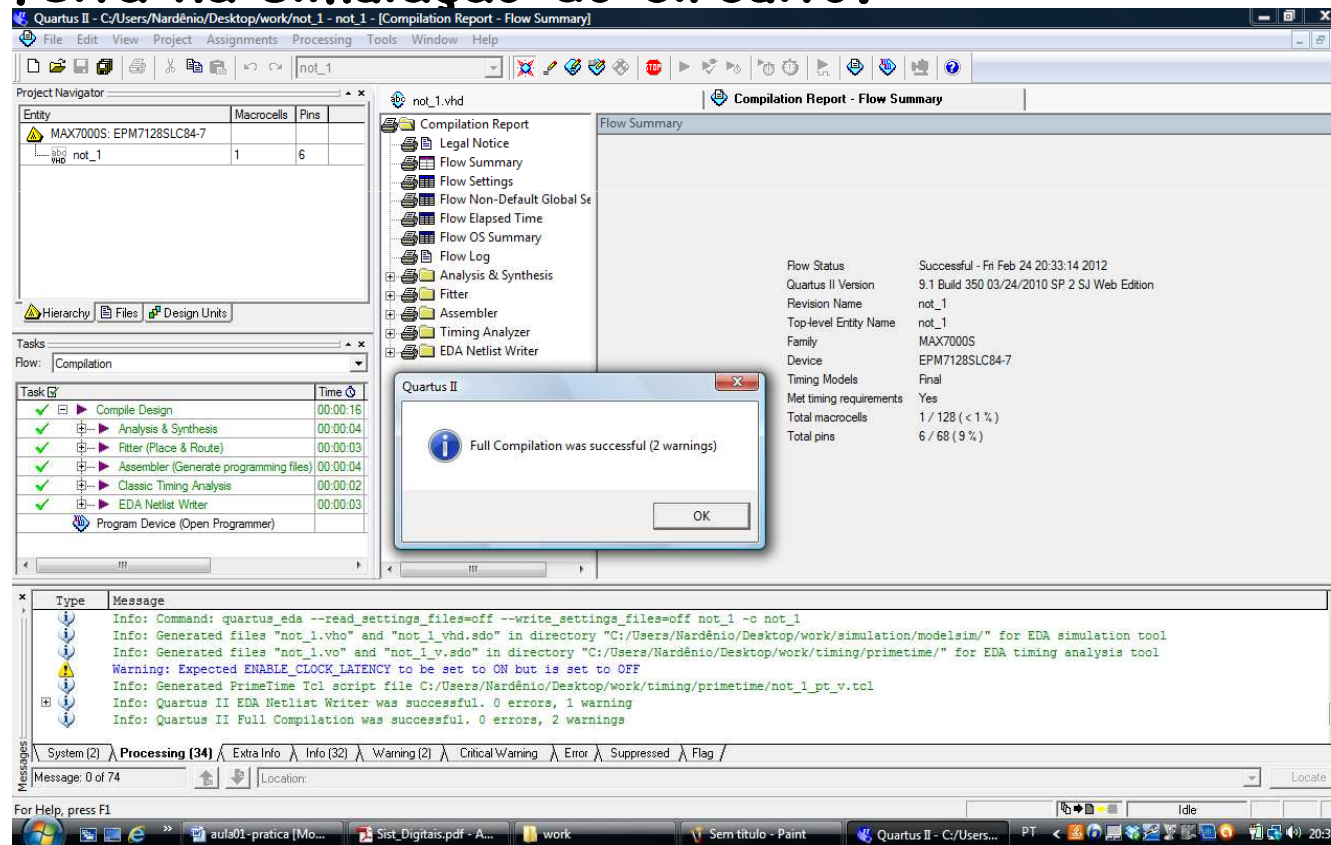
Software Quartus II

12. Salve o código em VHDL com extensão **".vhd"**.



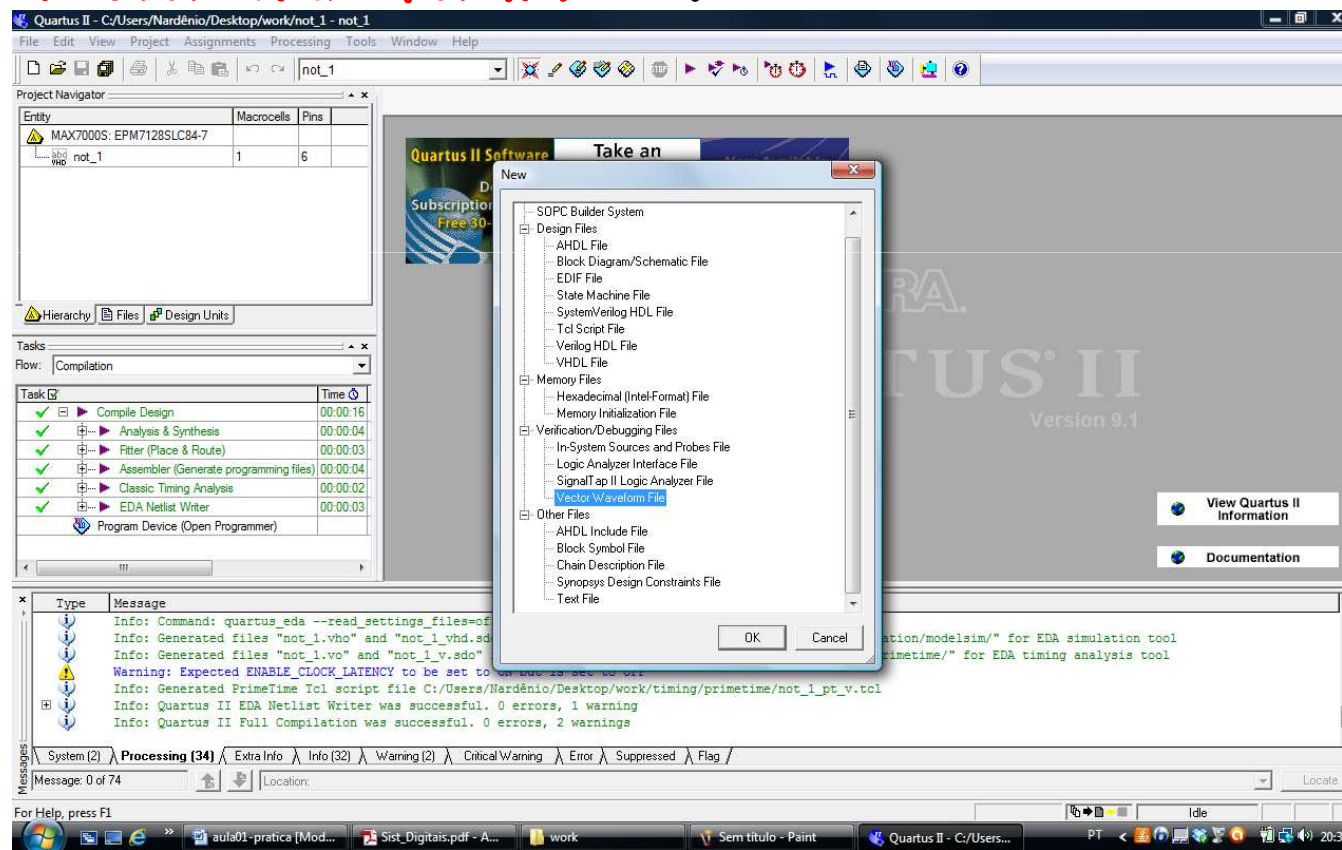
Software Quartus II

13. Compilação: **"Processing > Start Compilation"**. A compilação é a verificação da construção. Nesta etapa, erros lógicos não são detectados. Esta verificação é feita na simulação do circuito.



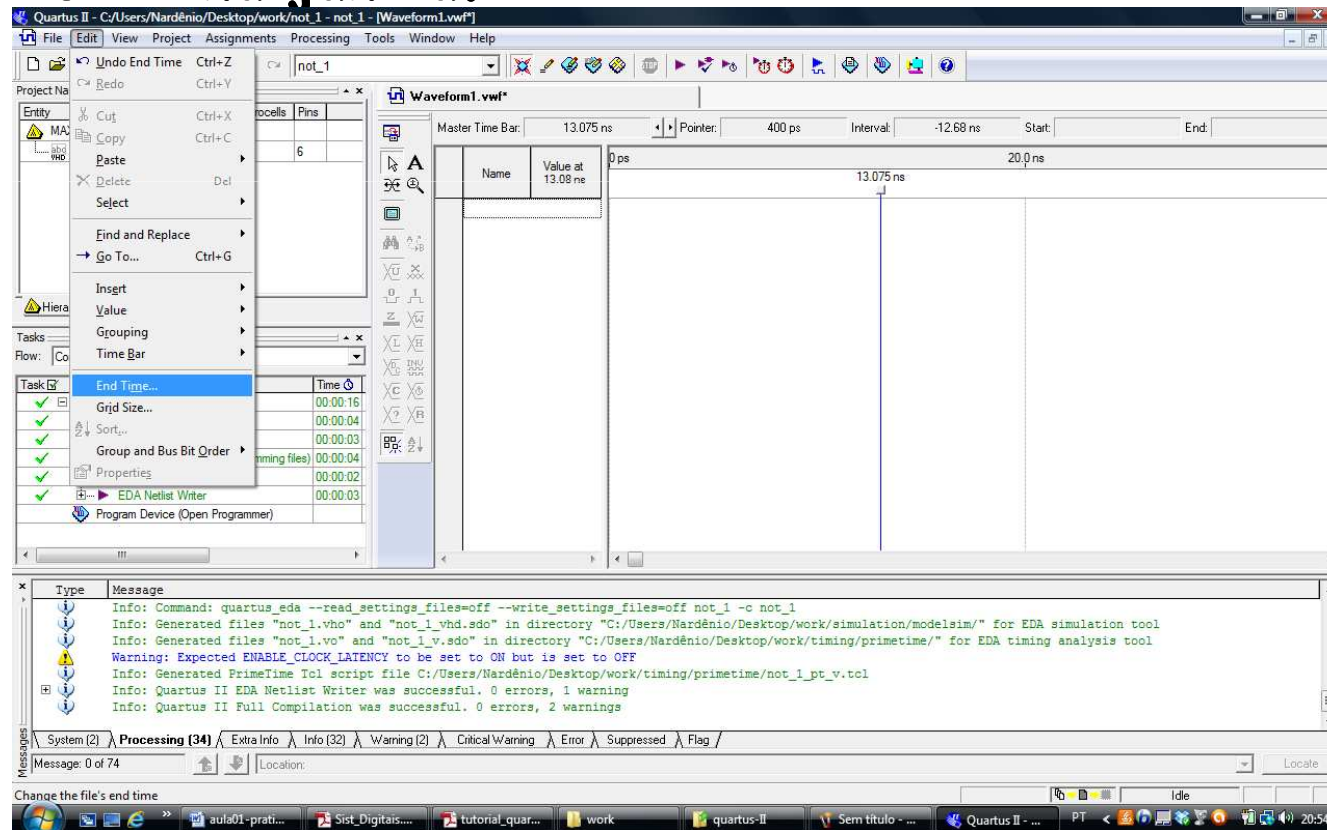
Software Quartus II

14.A verificação de erros lógicos é feita na simulação do circuito. Para isto selecione "File > New" e escolha "Vector Waveform File".



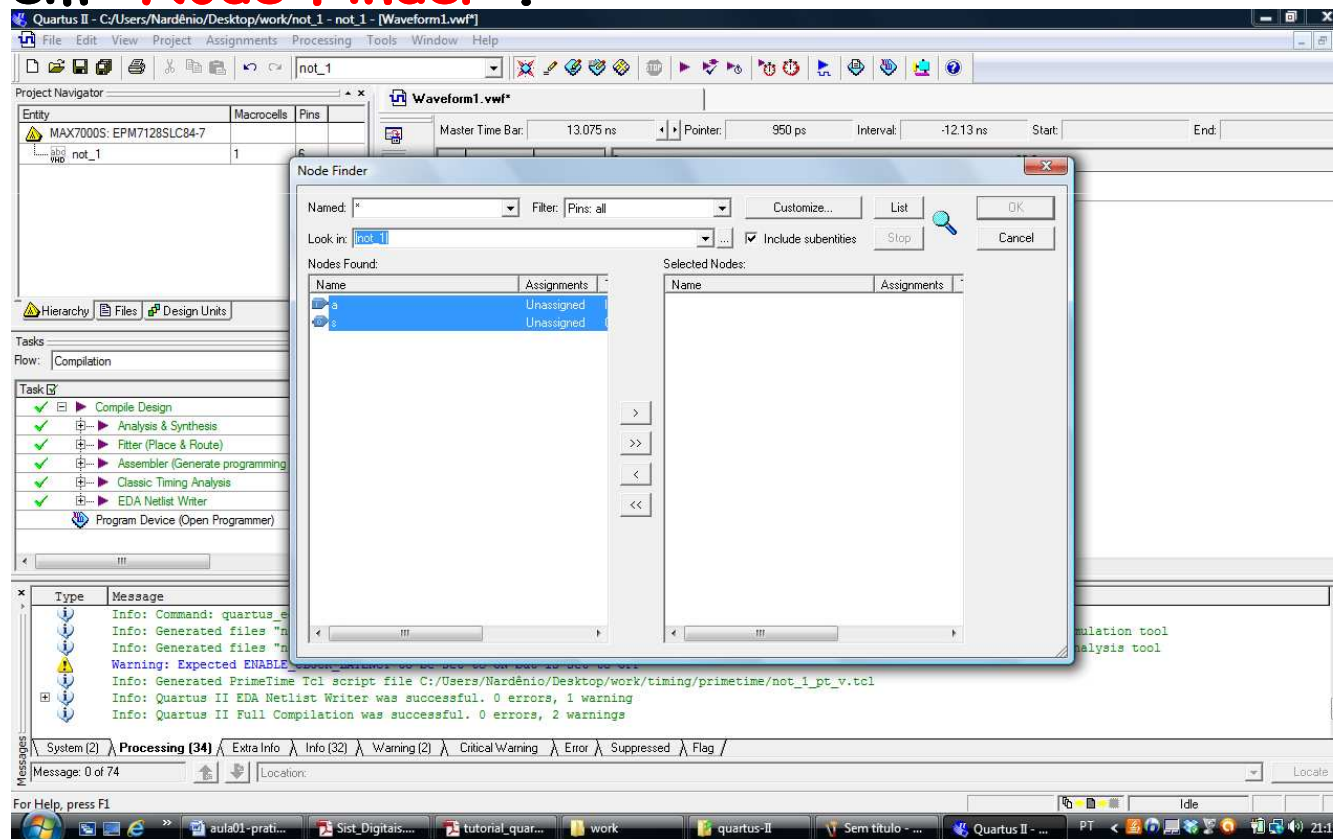
Software Quartus II

15. Ajuste o tempo de simulação: "Edit > End Time" e coloque 40 ns para simulação. Clique "View > Fit in Window" para que todo o tempo de simulação fique visível na janela.



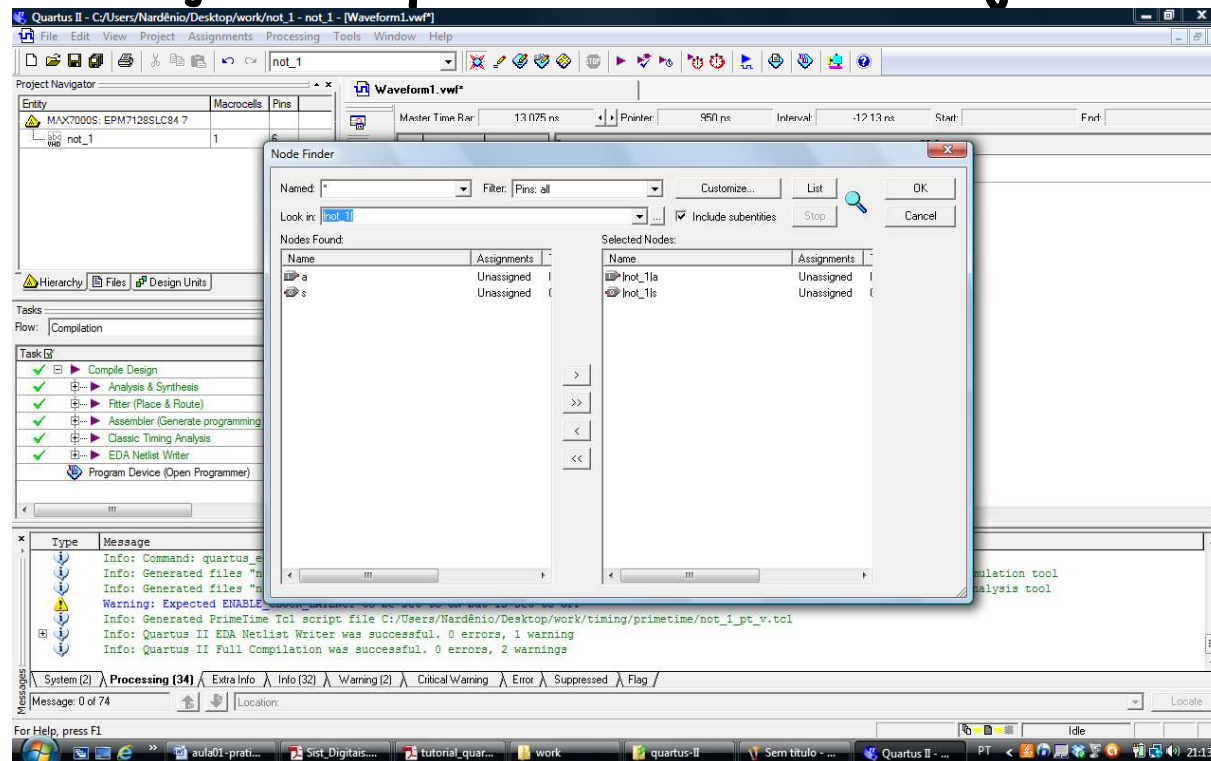
Software Quartus II

16. Selecione os vetores de entrada e saída a serem incluídos na simulação. Para isto clique em **Edit > Insert > Node or Bus**. Na janela que aparece, clique em **Node Finder**.



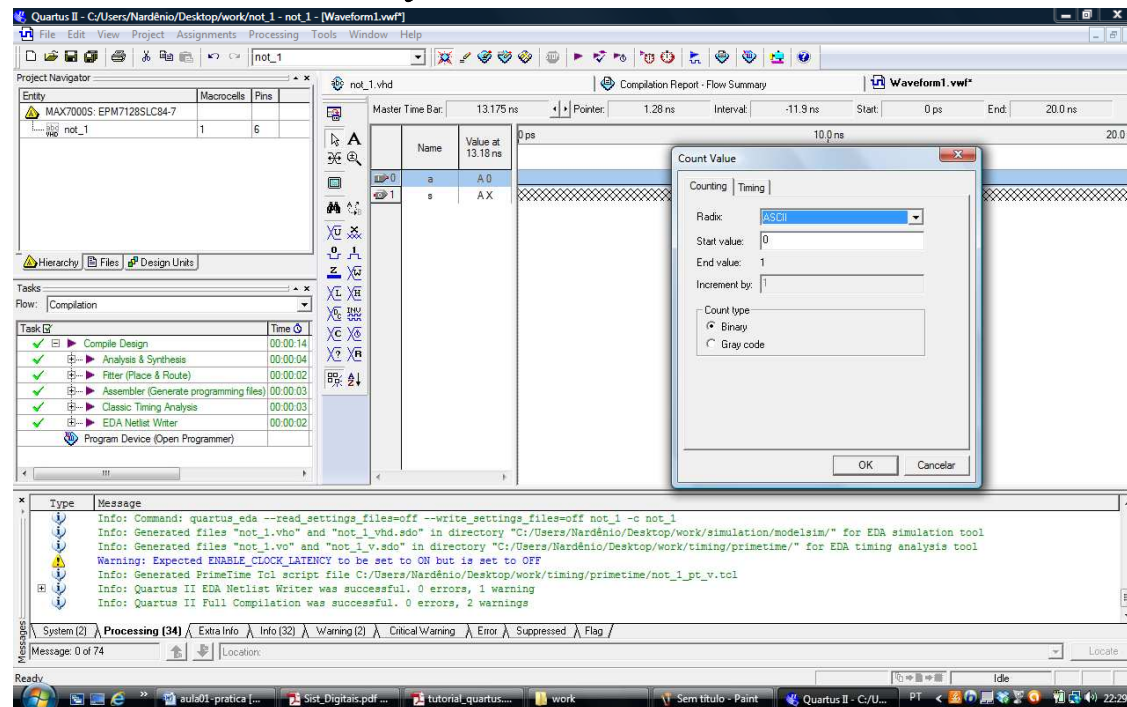
Software Quartus II

17. Na próxima janela, selecione "**Pins: All**" e, em seguida, clique em "**List**" (a função *List* amostra os vetores de entrada e saída). Em seguida, utilizando o botão ">>", transferir as entradas e saídas para a ferramenta de simulação. Clique em **Ok** nas duas janelas subsequentes.



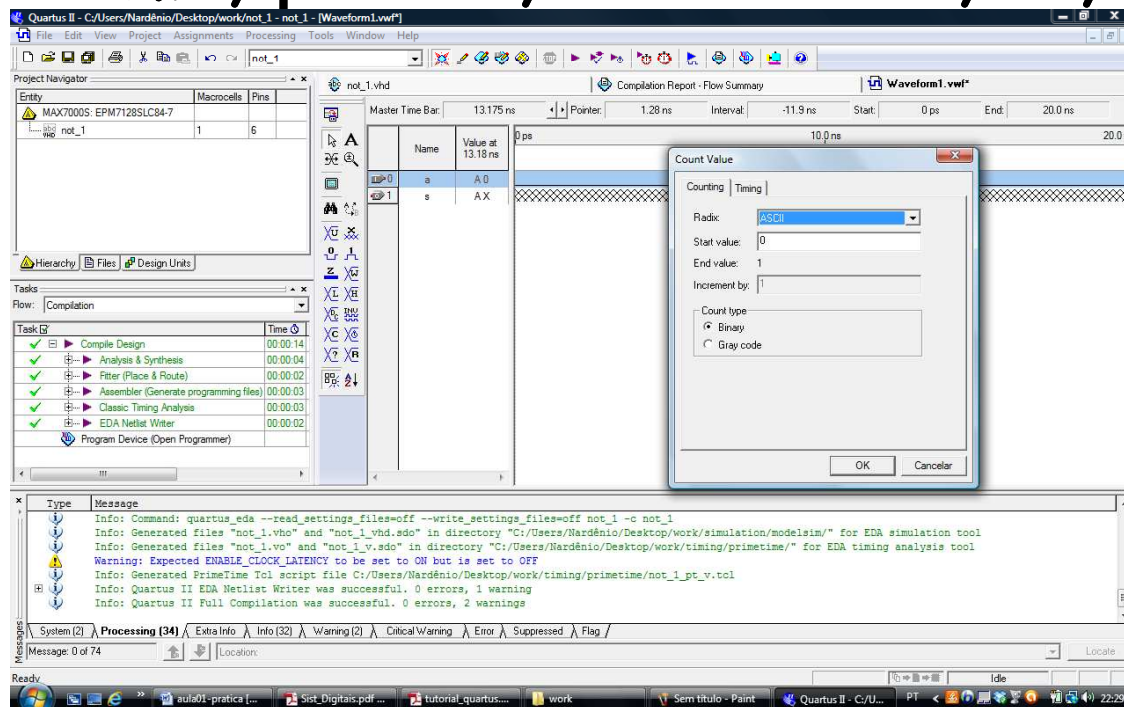
Software Quartus II

18. Insira as formas de onda de entrada para testar todas as possibilidades para a(s) entrada(s) do projeto. Marque toda(s) a(s) entrada(s) do projeto, clique com o botão direito e selecione "**Grouping > Group**". Insira um nome para o grupo de entradas (por exemplo, "**inputs**" ou "**entradas**").



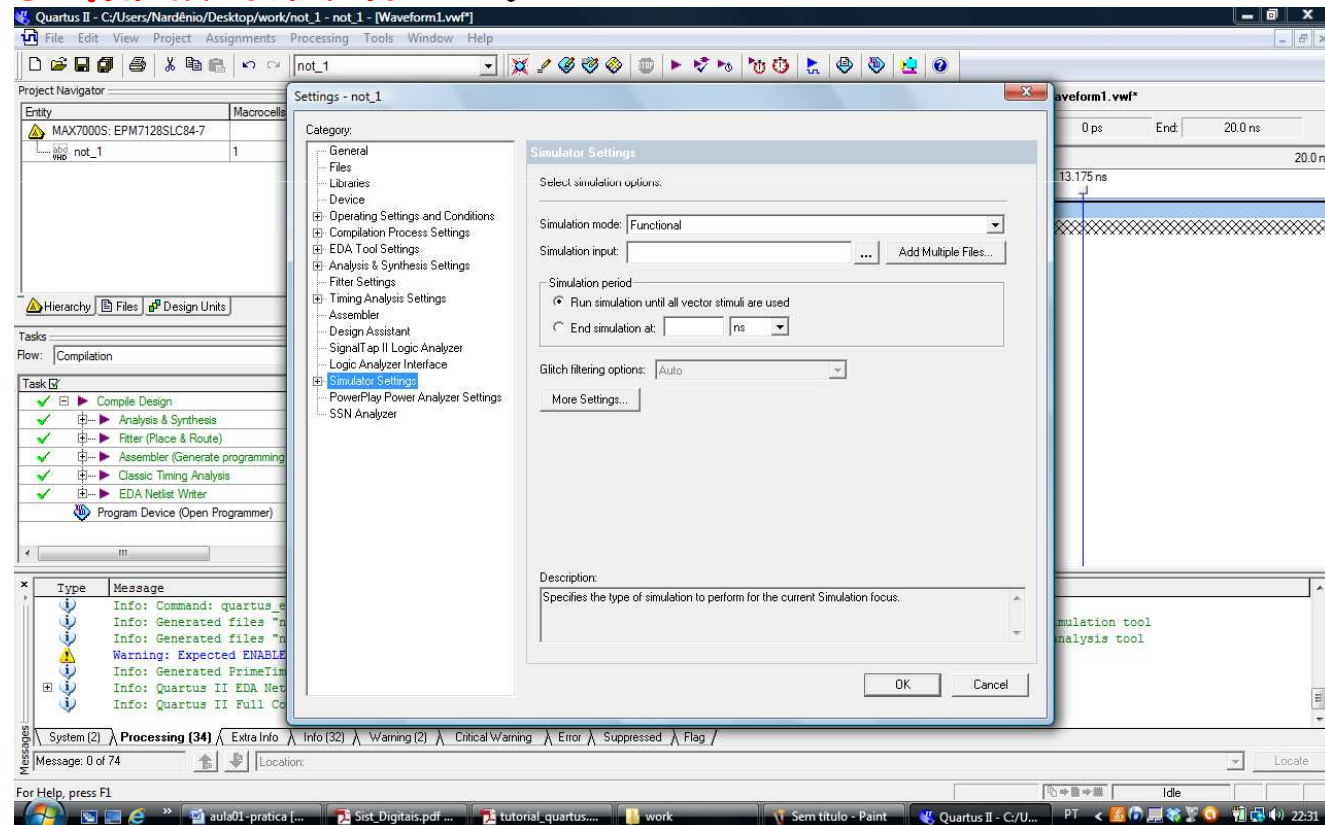
Software Quartus II

19. Clique com o botão direito sobre a(s) entrada(s) e selecione "**Value > Count Value**". Verifique que o campo "**Start Value**" tenha o valor [0] e o "**End Value**" tenha o valor [3] (na realidade, pode-se ver que os bits de entrada como duas entradas de 1 *bit*, que pode assumir, portanto, valores de 00, 01, 10 a 11).



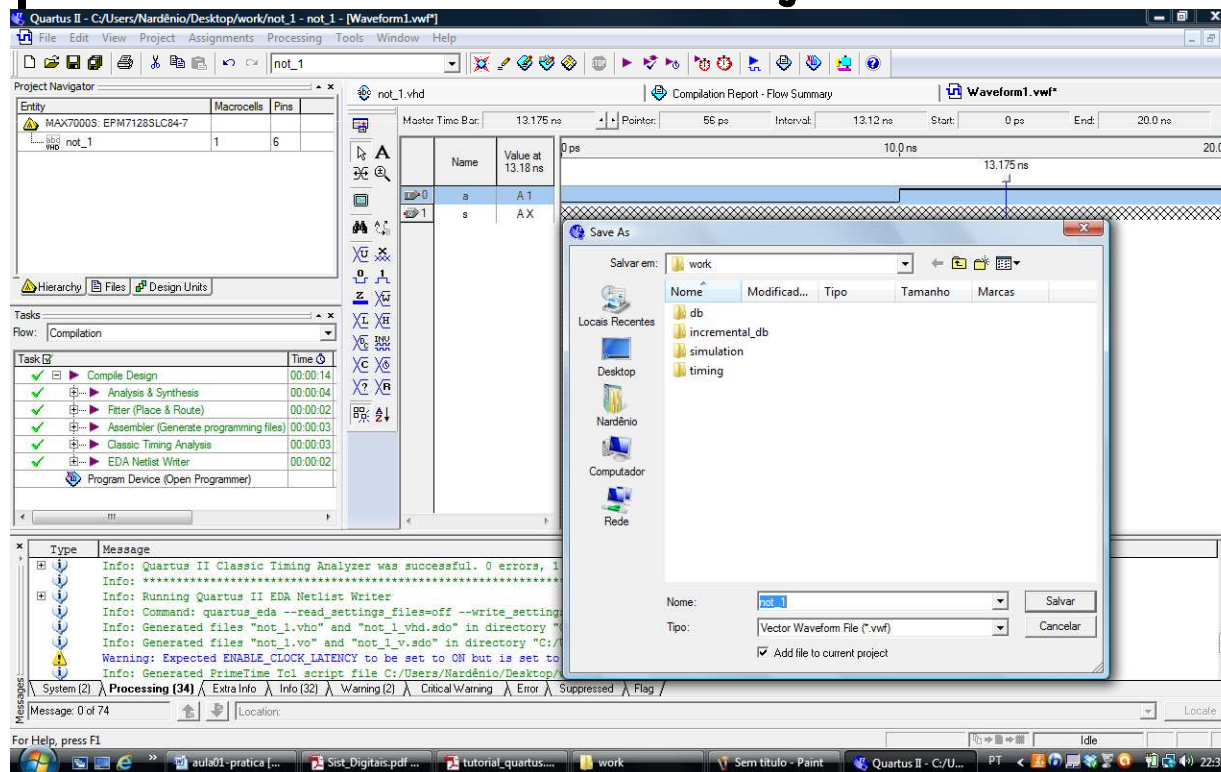
Software Quartus II

20. O passo seguinte é a simulação do circuito projetado. Clique em **"Assignments > Settings"**, selecione **"Simulator Settings"** e escolha **"Functional em Simulation Mode"**.



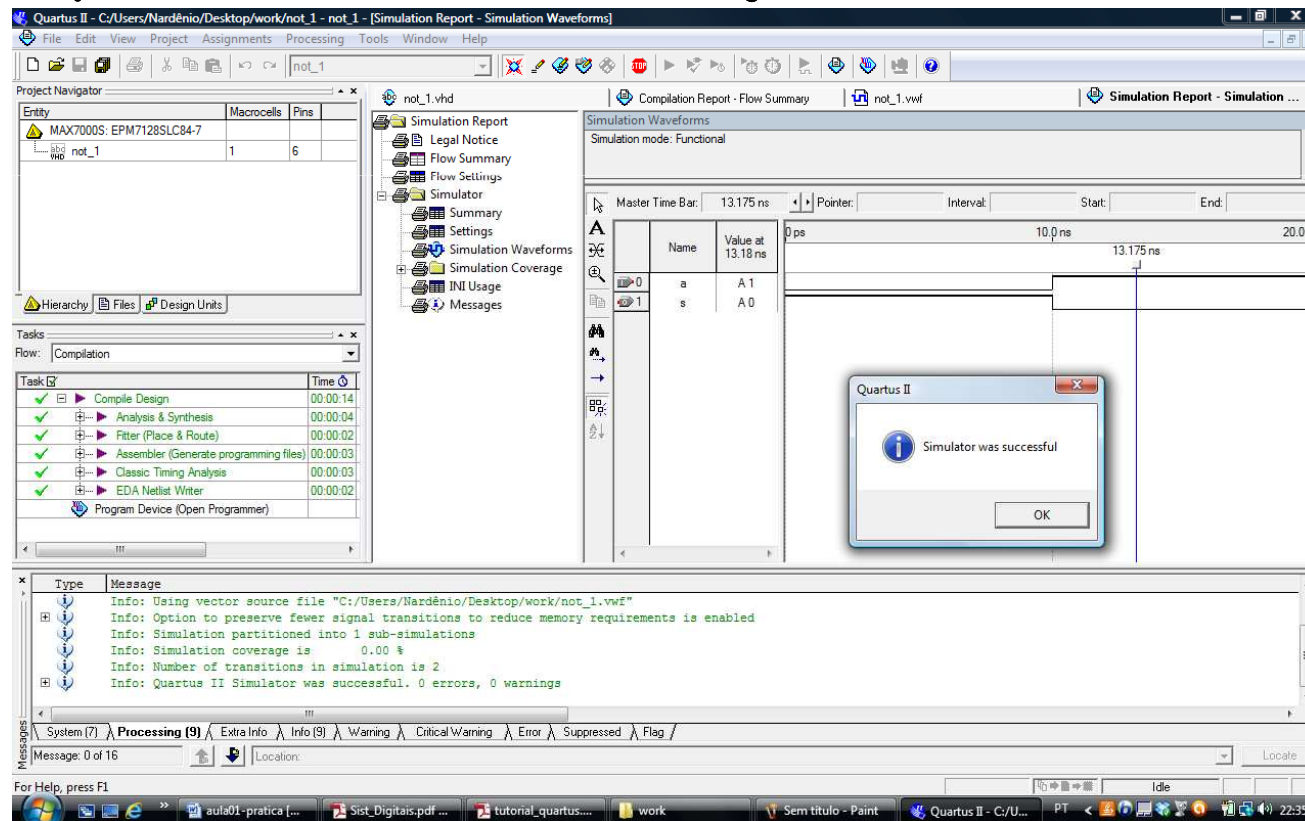
Software Quartus II

21. Clique em "**Processing > Generate Functional Simulation Netlist**". Antes de executar a simulação é necessário salvar o arquivo que deve conter o mesmo nome dado ao código em VHDL. Neste caso, "**and_2.vwf**". Esses passos definem uma simulação funcional.



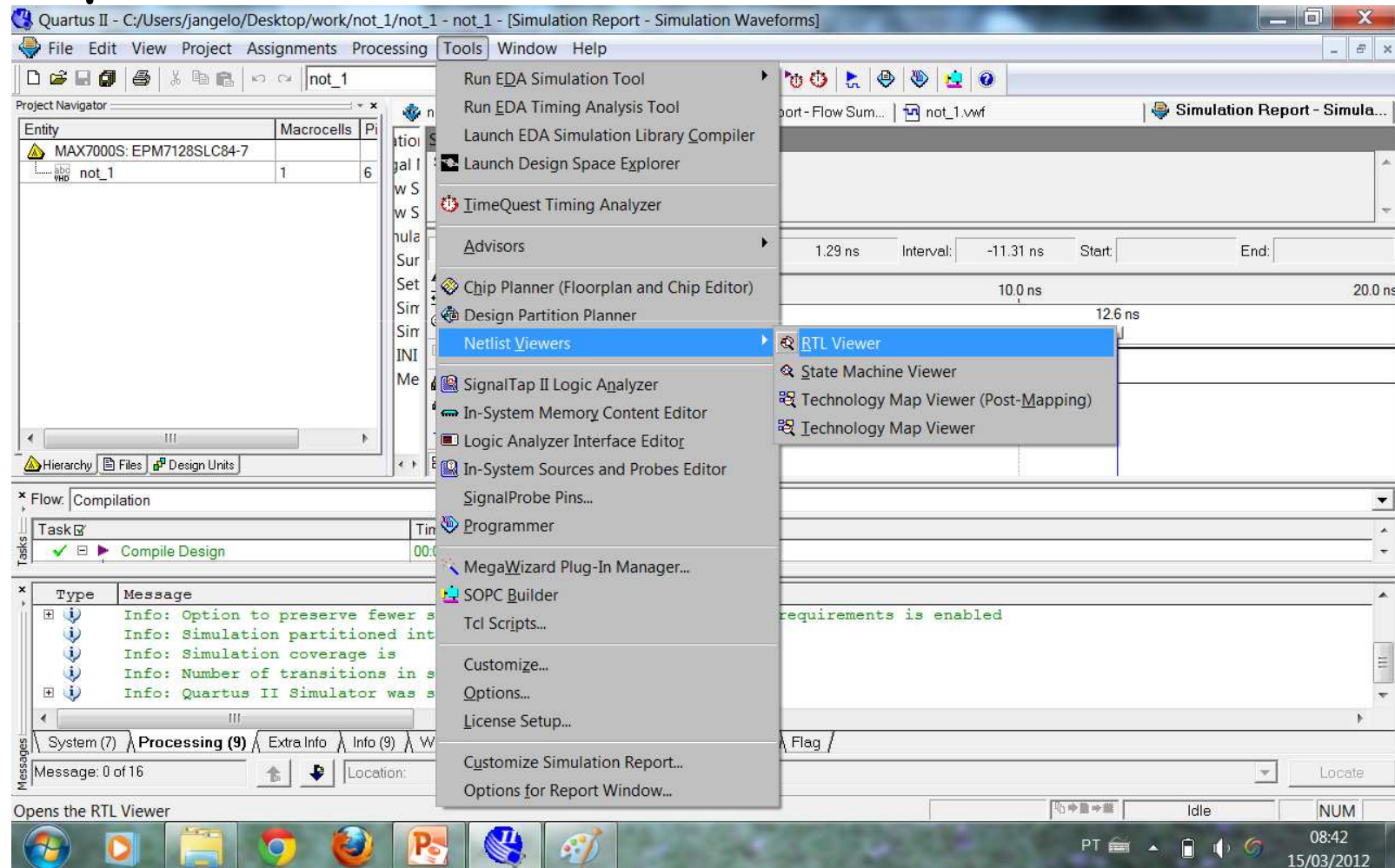
Software Quartus II

22. Clique em "**Processing > Start Simulation**". Verifique o valor da saída para cada entrada e veja que o circuito sintetizado a partir do código em VHDL de fato implementa a função desejada.



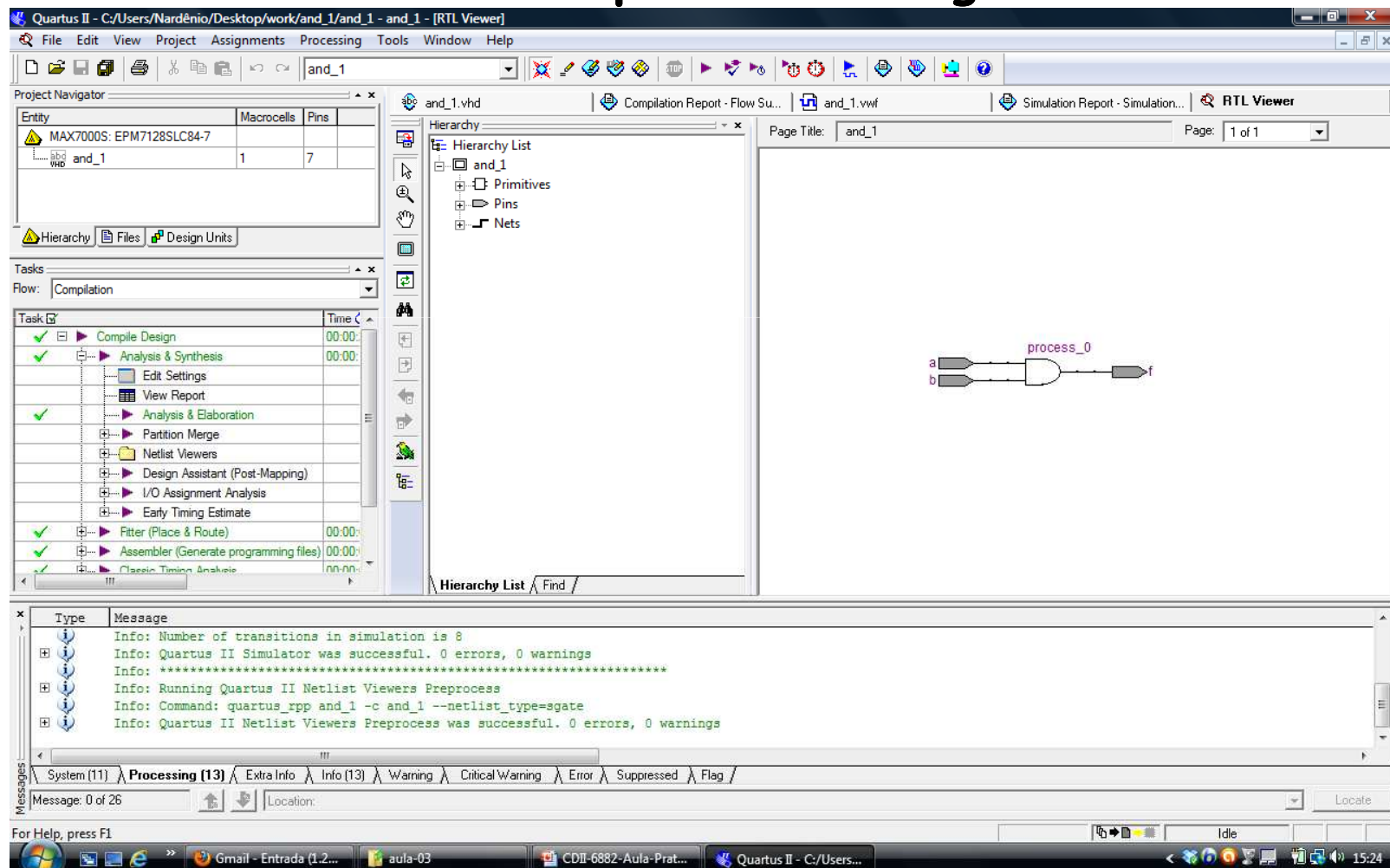
Software Quartus II

23. Clique em **"Tools > Netlist Viewers > RTL Viewer"**.



Software Quartus II

24. O circuito sintetizado a partir do código em VHDL.



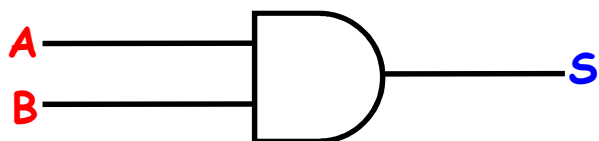
Aula de Hoje

Repita os procedimentos para as implementações das demais portas lógicas

HDL - Linguagem de Descrição de Hardware

Tabela Verdade

Símbolo da Porta AND



TV da Porta AND

Entradas		Saída
A	B	S
0	0	0
0	1	0
1	0	0
1	1	1

MODELAGEM COMPORTAMENTAL EM VHDL

...

```
PROCESS (a, b)
```

```
BEGIN
```

```
    IF (a='1') and (b='1') THEN
```

```
        f<='1';
```

```
    ELSE
```

```
        f<='0';
```

```
    END IF;
```

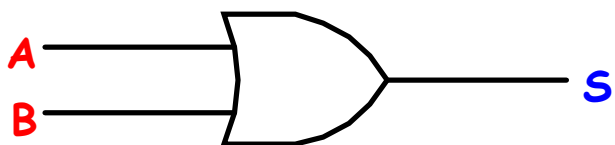
```
END PROCESS;
```

...

HDL - Linguagem de Descrição de Hardware

Tabela Verdade

Símbolo da Porta OR



TV da Porta OR

Entradas Saída

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1

MODELAGEM COMPORTAMENTAL EM VHDL

...

PROCESS (a, b)

BEGIN

IF (a='0') and (b='0') THEN

f<='0';

ELSE

f<='1';

END IF;

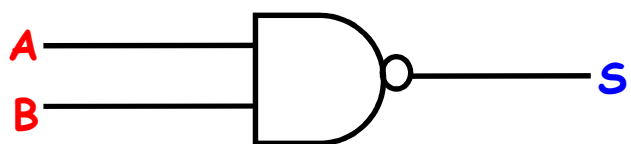
END PROCESS;

...

HDL - Linguagem de Descrição de Hardware

Tabela Verdade

Símbolo da Porta NAND



TV da Porta NAND

Entradas		Saída
A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

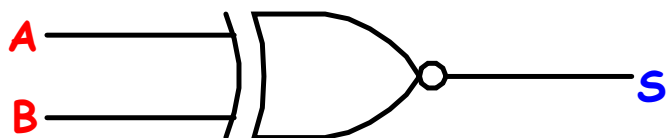
MODELAGEM COMPORTAMENTAL EM VHDL

```
...  
PROCESS (a, b)  
  BEGIN  
    IF (a='1') and (b='1') THEN  
      f<='0';  
    ELSE  
      f<='1';  
    END IF;  
  END PROCESS;  
...
```

HDL - Linguagem de Descrição de Hardware

Tabela Verdade

Símbolo da Porta XNOR



TV da Porta XNOR

Entradas		Saída
A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

MODELAGEM COMPORTAMENTAL EM VHDL

...

```
PROCESS (a, b)
```

```
BEGIN
```

```
    IF (a=b) THEN
```

```
        f<='1';
```

```
    ELSE
```

```
        f<='0';
```

```
    END IF;
```

```
END PROCESS;
```

...