

New Project Wizard

Introduction

The New Project Wizard helps you create a new project and preliminary project settings, including the following:

- Project name and directory
- Name of the top-level design entity
- Project files and libraries
- Target device family and device
- EDA tool settings

You can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the project.

☐ Don't show me this introduction again

< Back Next > Finish Cancel Help

New Project Wizard

Directory, Name, Top-Level Entity [page 1 of 5]

What is the working directory for this project?

E:\exemplo_vhdl\cont4 ...

What is the name of this project?

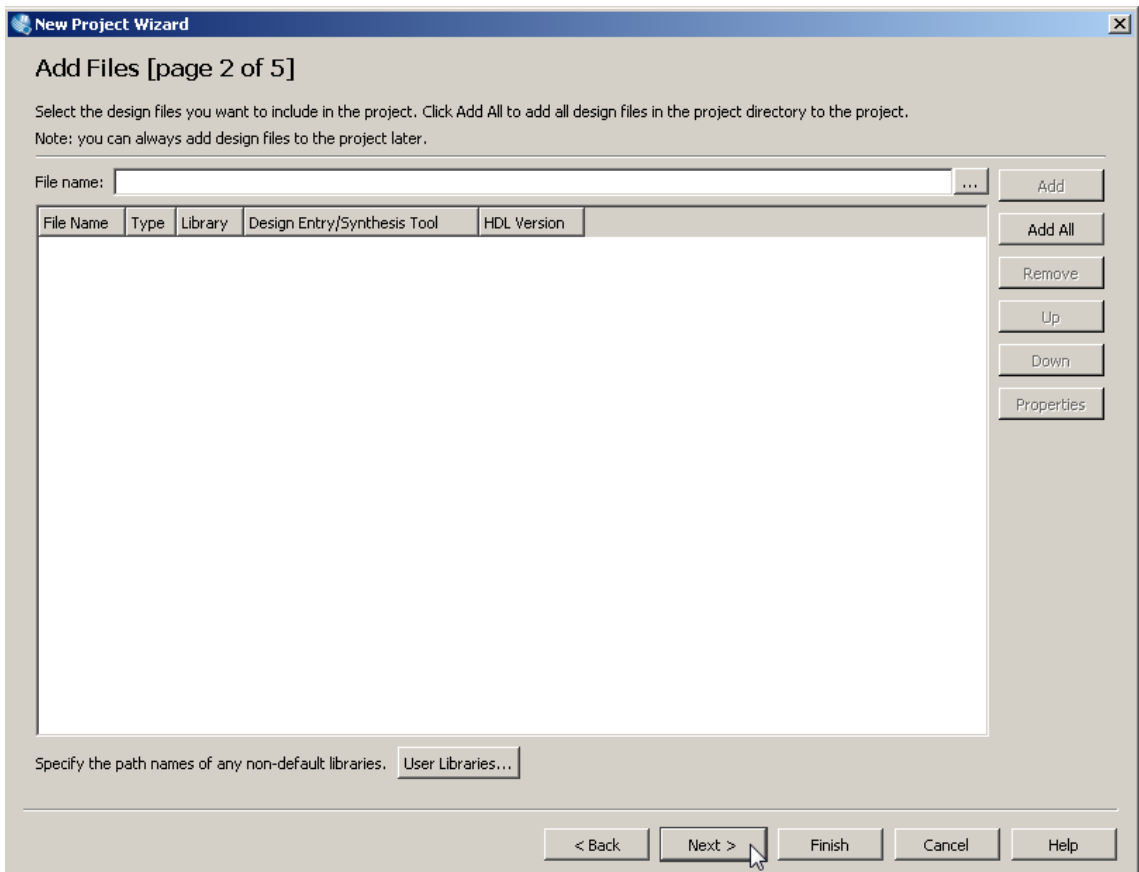
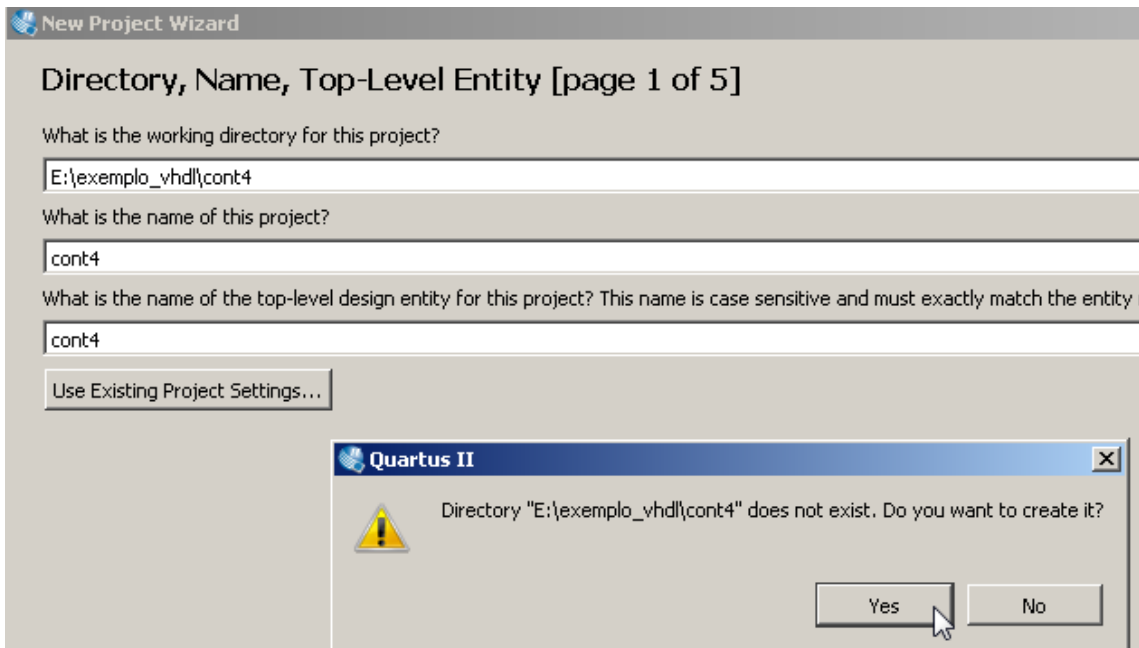
cont4 ...

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.

cont4 ...

Use Existing Project Settings...

< Back Next > Finish Cancel Help



New Project Wizard

Family & Device Settings [page 3 of 5]

Select the family and device you want to target for compilation.
You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus II software in which your target device is supported, refer to the [Device Support List](#) webpage.

Device family

Family: Cyclone IV E

Devices: All

Show in 'Available devices' list

Package: Any

Pin count: Any

Core Speed grade: Any

Name filter:

☒ Show advanced devices

Target device

☐ Auto device selected by the Filter

☒ Specific device selected in 'Available devices' list

☐ Other: n/a

Available devices:

Name	Core Voltage	LEs	User I/Os	Memory Bits	Embedded multiplier 9-bit element
EP4CE22F17A7	1.2V	22320	154	608256	132
EP4CE22F17C6	1.2V	22320	154	608256	132
EP4CE22F17C7	1.2V	22320	154	608256	132
EP4CE22F17C8	1.2V	22320	154	608256	132
EP4CE22F17C8L	1.0V	22320	154	608256	132
EP4CE22F17C9L	1.0V	22320	154	608256	132
EP4CE22F17I7	1.2V	22320	154	608256	132
EP4CE22F17I8L	1.0V	22320	154	608256	132

< Back Next > Finish Cancel Help

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

New Project Wizard

Summary [page 5 of 5]

When you click Finish, the project will be created with the following settings:

Project directory:	E:\exemplo_vhdl\cont4
Project name:	cont4
Top-level design entity:	cont4
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Family name:	Cyclone IV E
Device:	EP4CE22F17C6
EDA tools:	
Design entry/synthesis:	<None> (<None>)
Simulation:	ModelSim-Altera (VHDL)
Timing analysis:	()
Operating conditions:	
VCCINT voltage:	1.2V
Junction temperature range:	0-85 °C

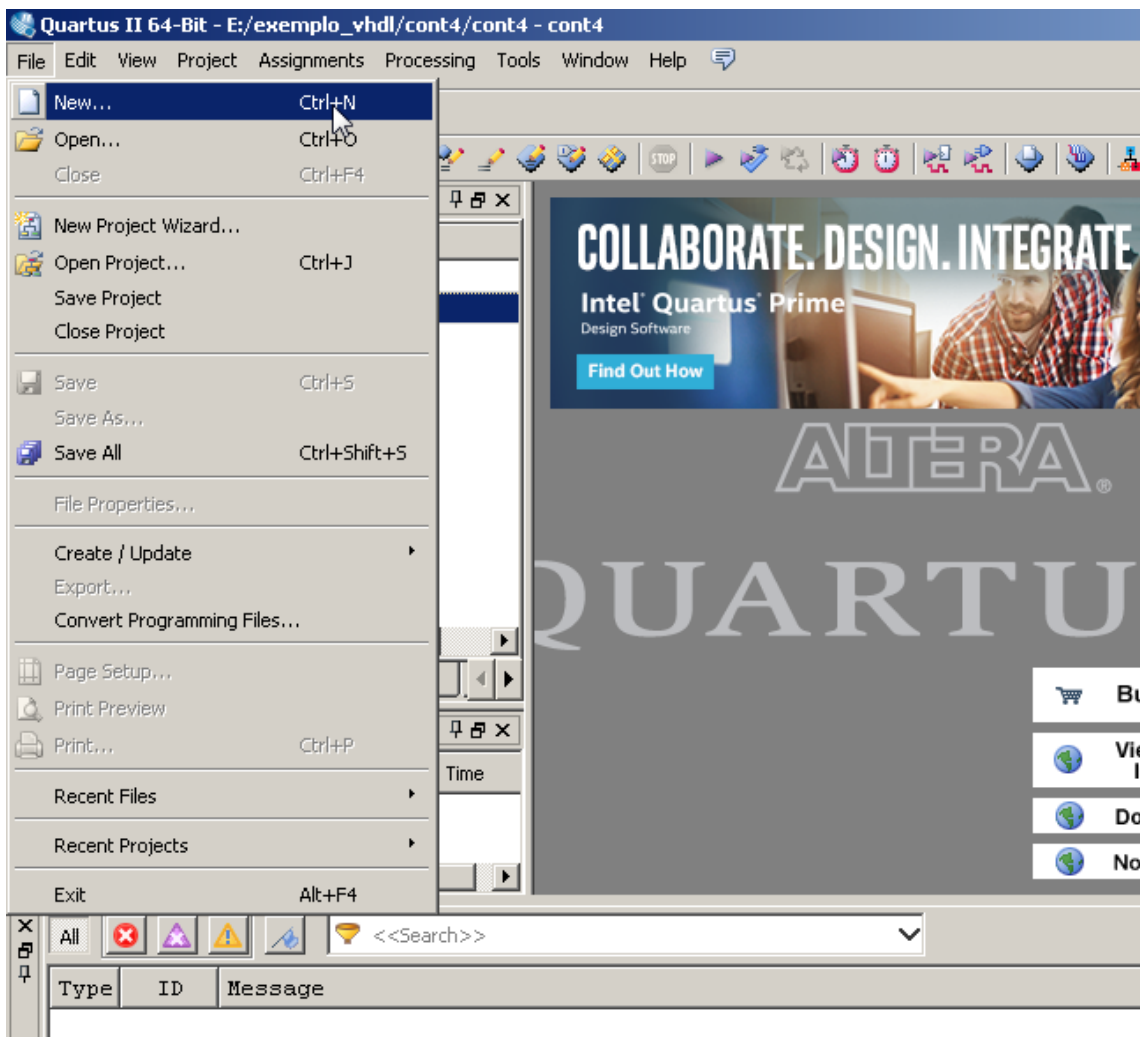
< Back

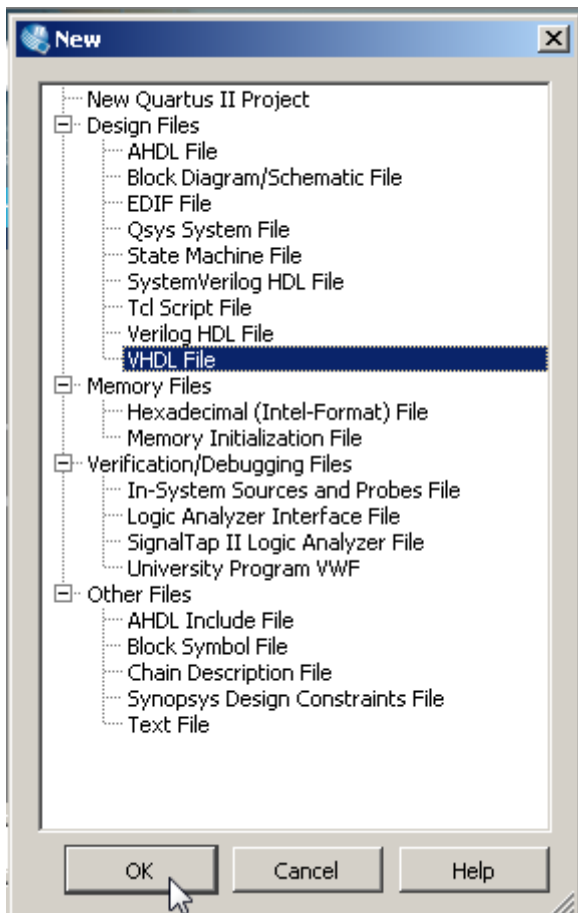
Next >

Finish

Cancel

Help





Copiar o código abaixo para o novo arquivo vhd.

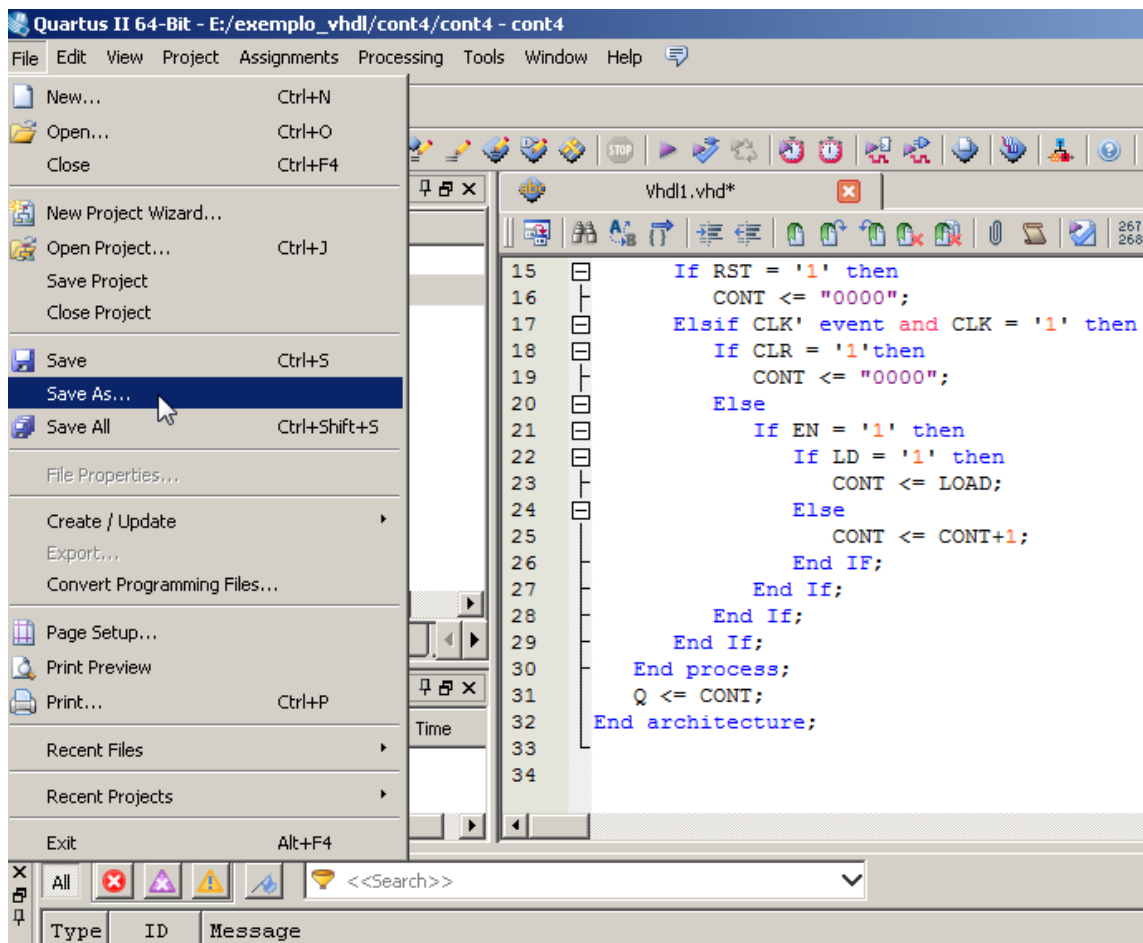
```
Library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;
```

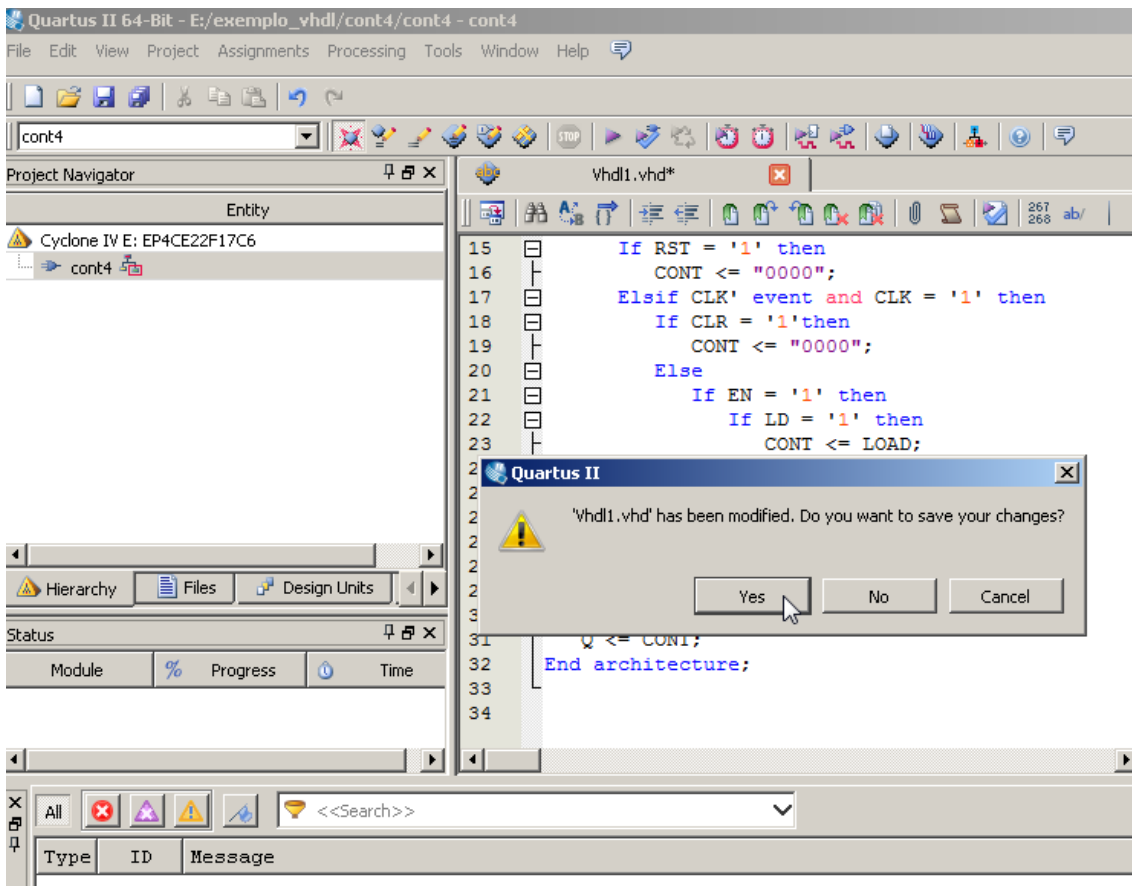
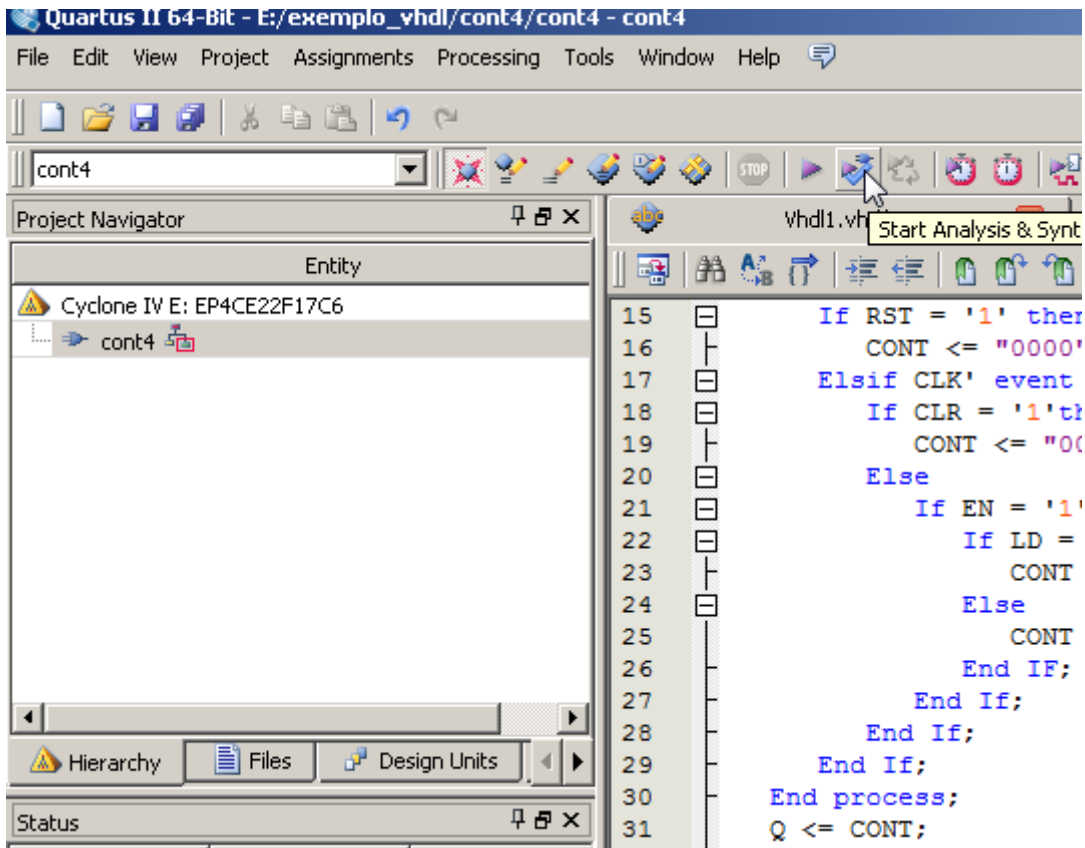
```
Entity Cont4_completo is
    PORT(RST: in std_logic;
          CLK: in std_logic;
          Q: out unsigned(3 downto 0);
          EN: in std_logic;
          CLR: in std_logic;
          LD: in std_logic;
          LOAD: in unsigned (3 downto 0));
end entity;
Architecture X of Cont4_completo is
    Signal CONT: unsigned (3 downto 0);
Begin
    Process (CLK, RST)
    Begin
        If RST = '1' then
            CONT <= "0000";
```

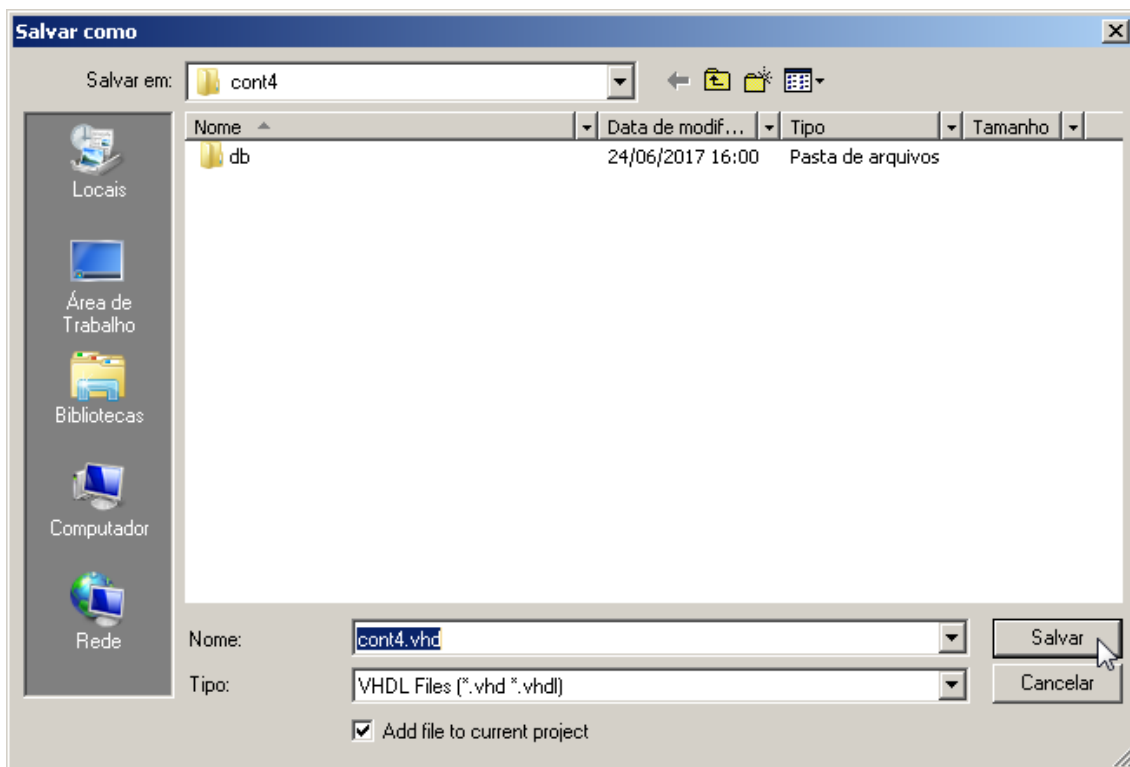
```

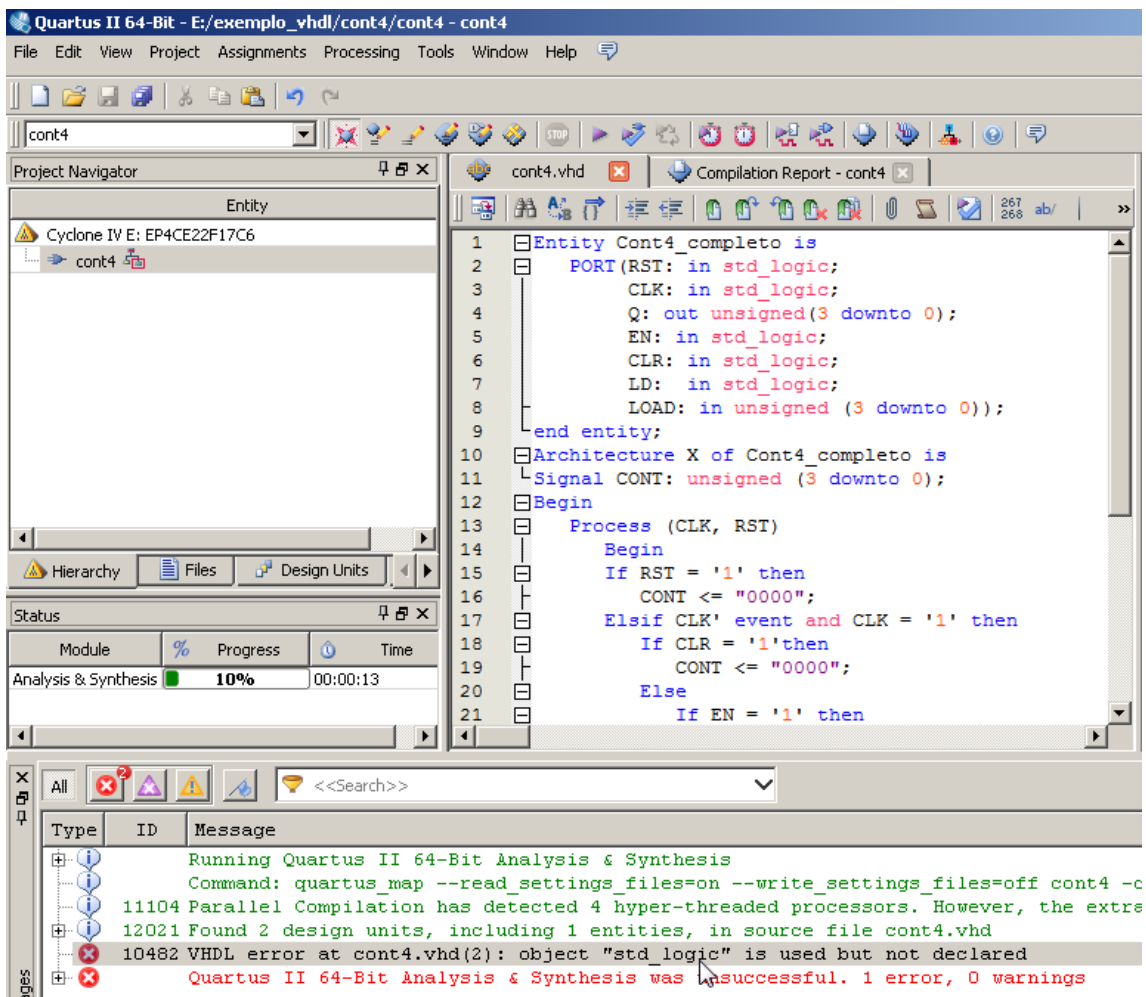
    Elsif CLK' event and CLK = '1' then
        If CLR = '1'then
            CONT <= "0000";
        Else
            If EN = '1' then
                If LD = '1' then
                    CONT <= LOAD;
                Else
                    CONT <= CONT+1;
                End IF;
            End If;
        End If;
    End If;
End process;
Q <= CONT;
End architecture;

```







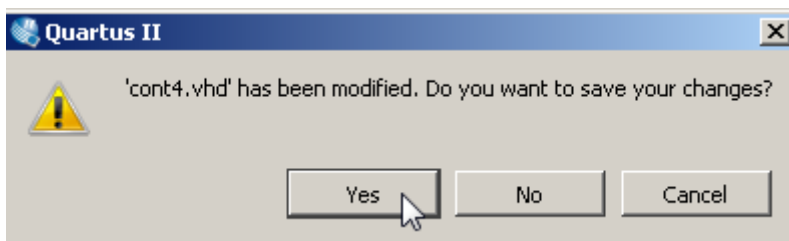
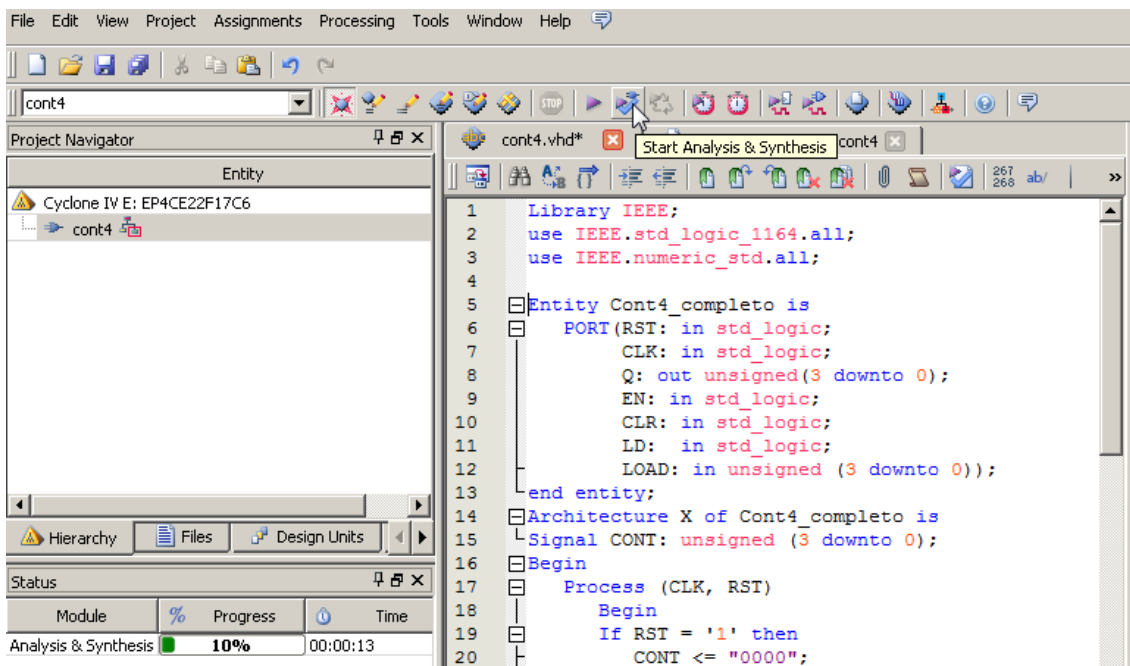


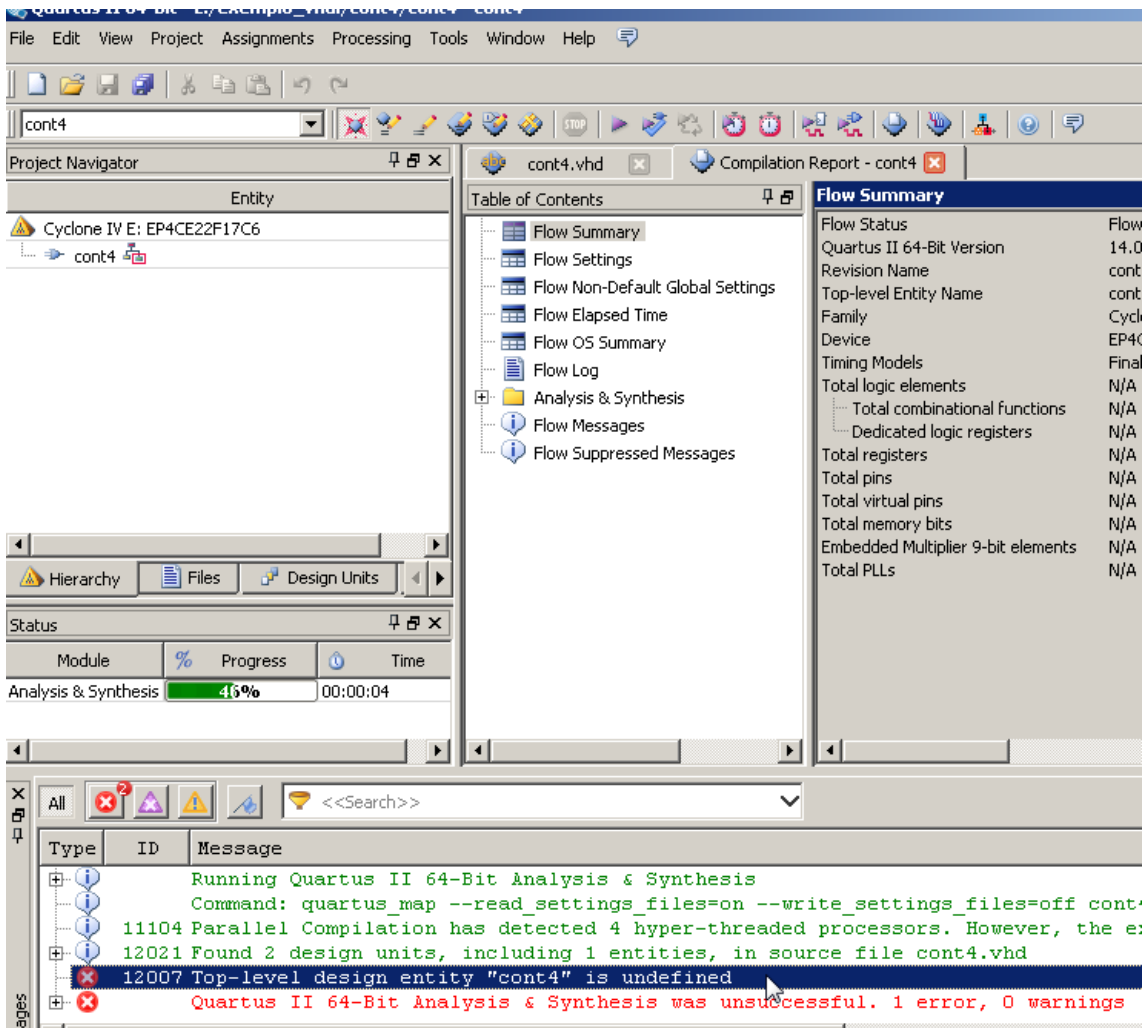
Faltaram as linhas relativas às bibliotecas

Library IEEE;

use IEEE.std_logic_1164.all;

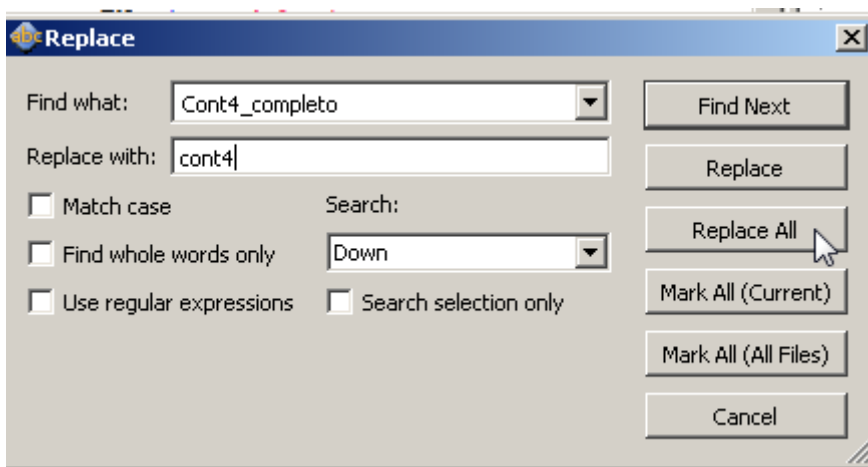
use IEEE.numeric_std.all;

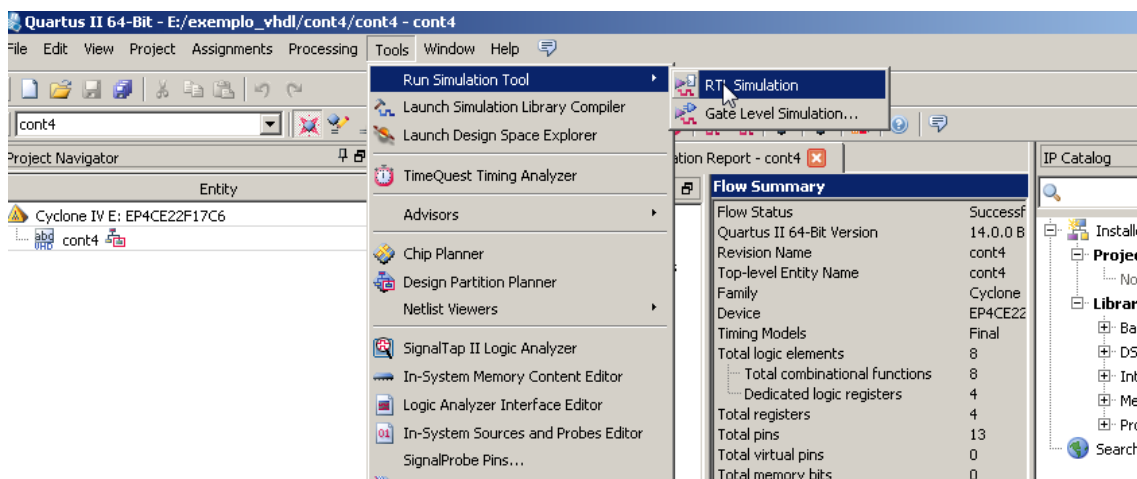
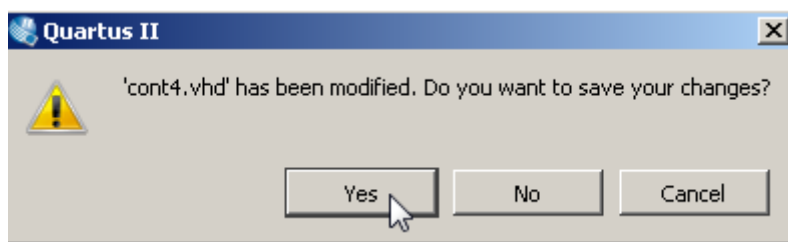
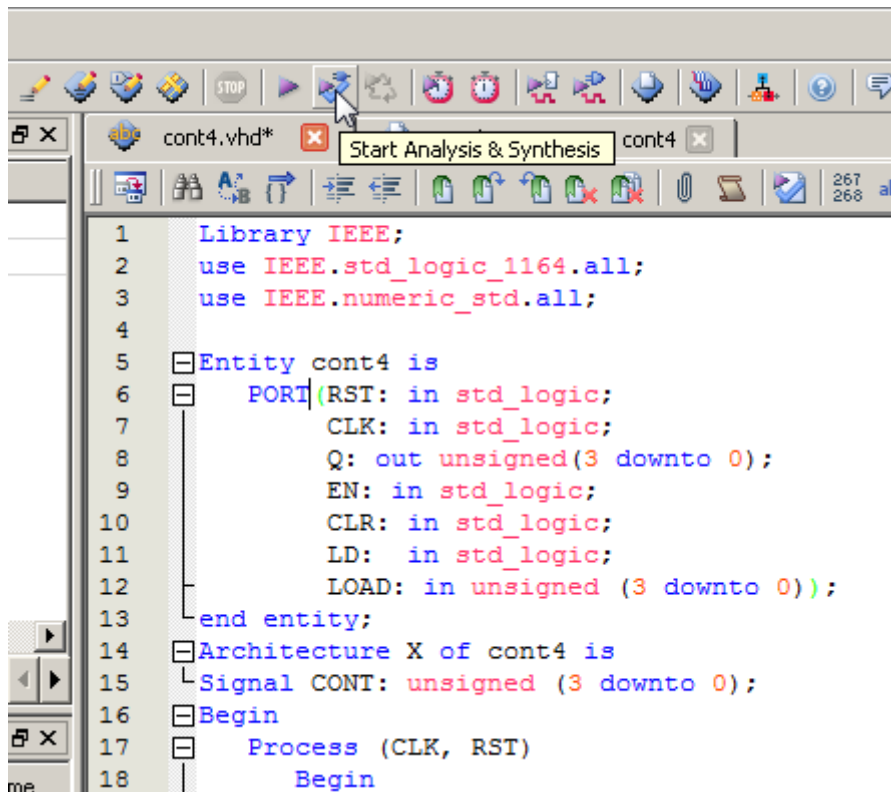


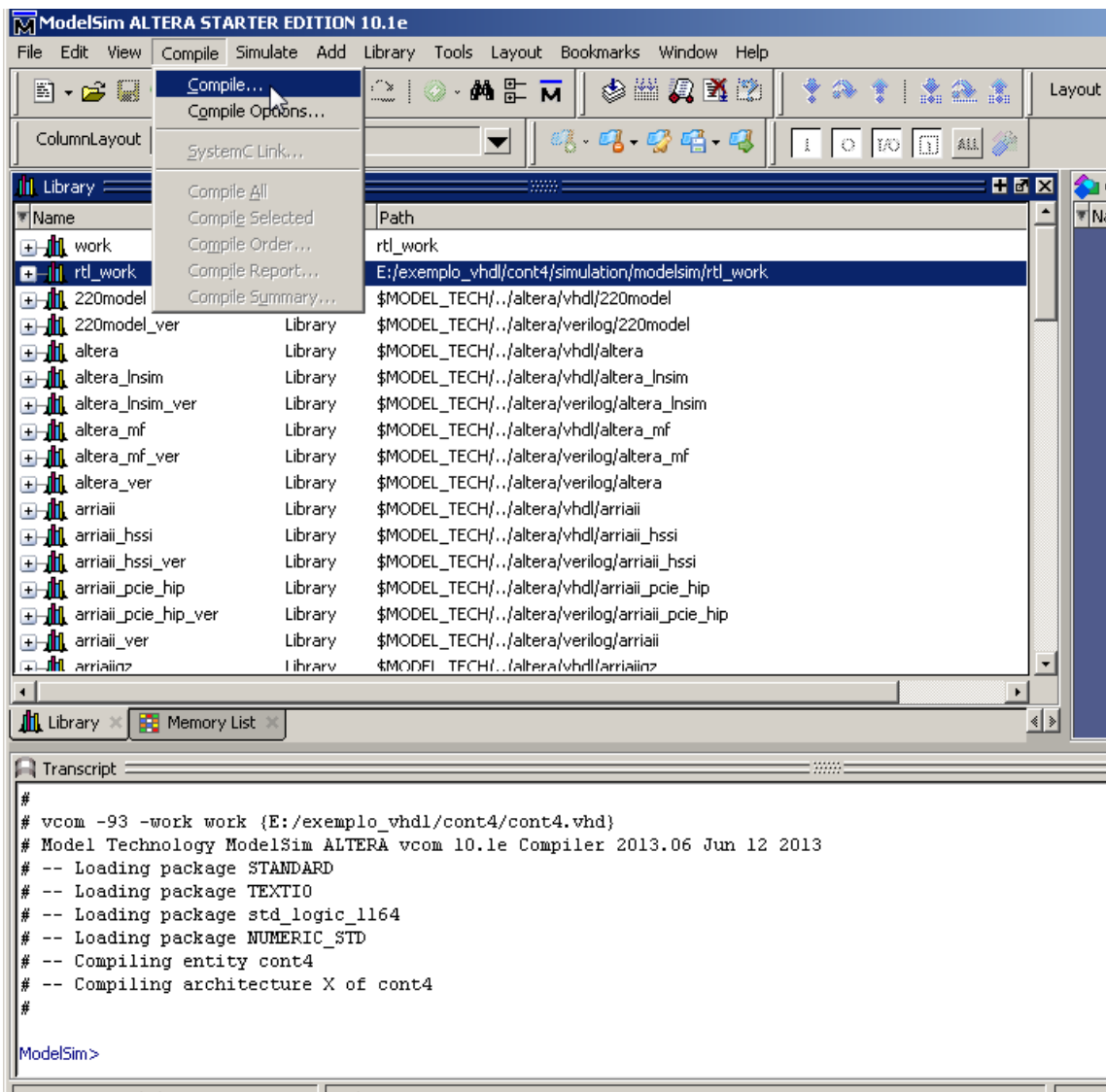


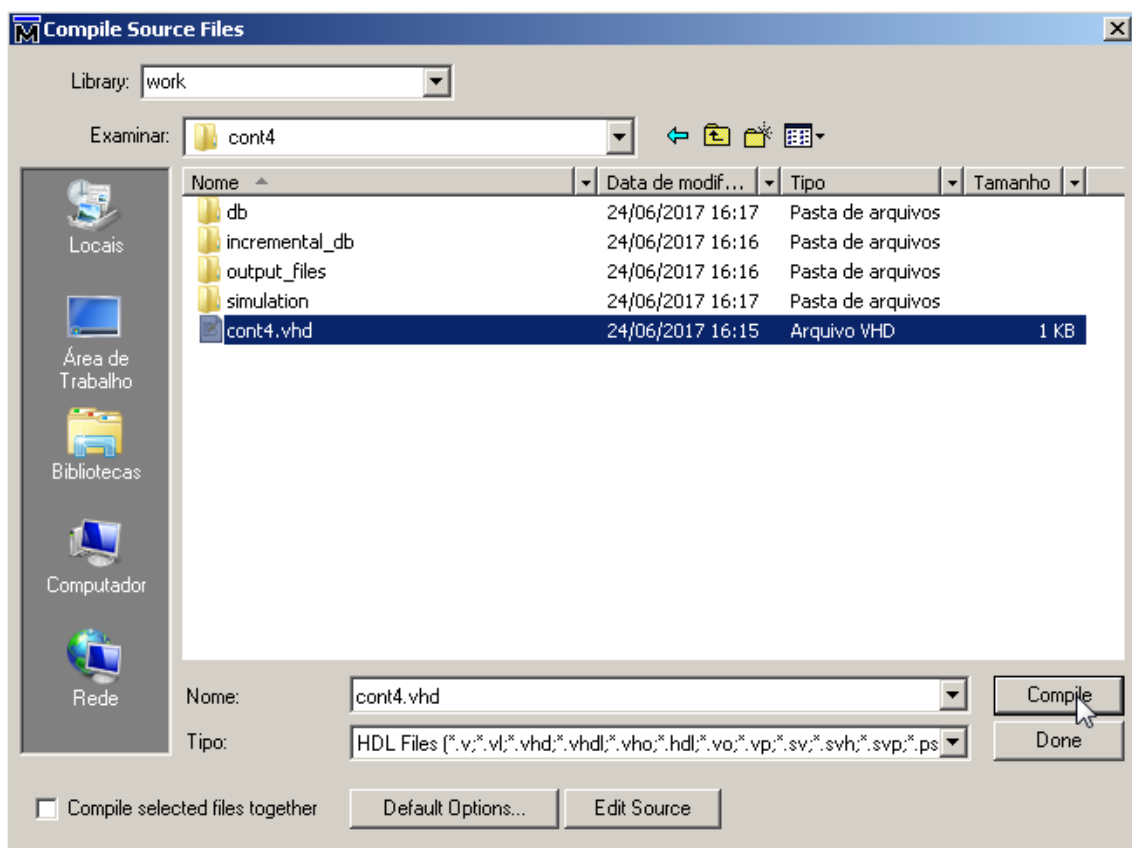
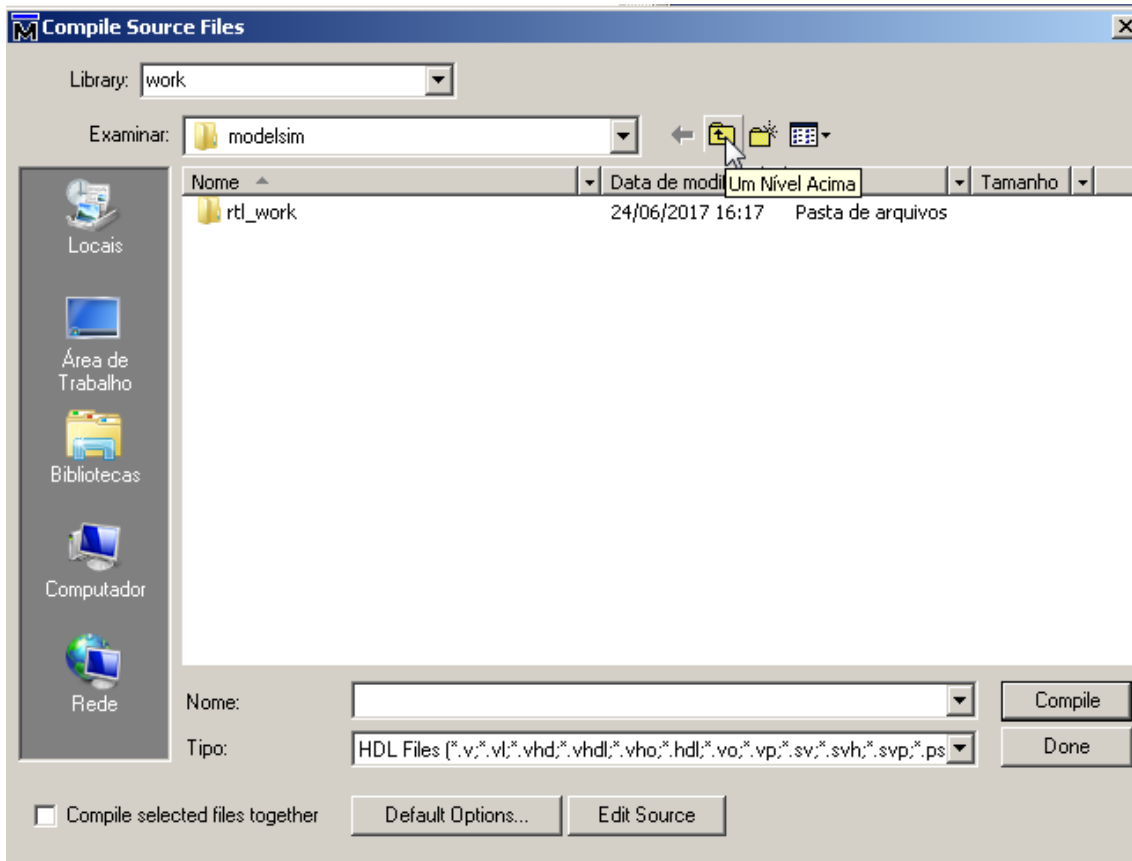
O problema é que o projeto chama-se “cont4”, mas a entidade e arquitetura chama-se “cont4_completo”. Substituir e recompilar.

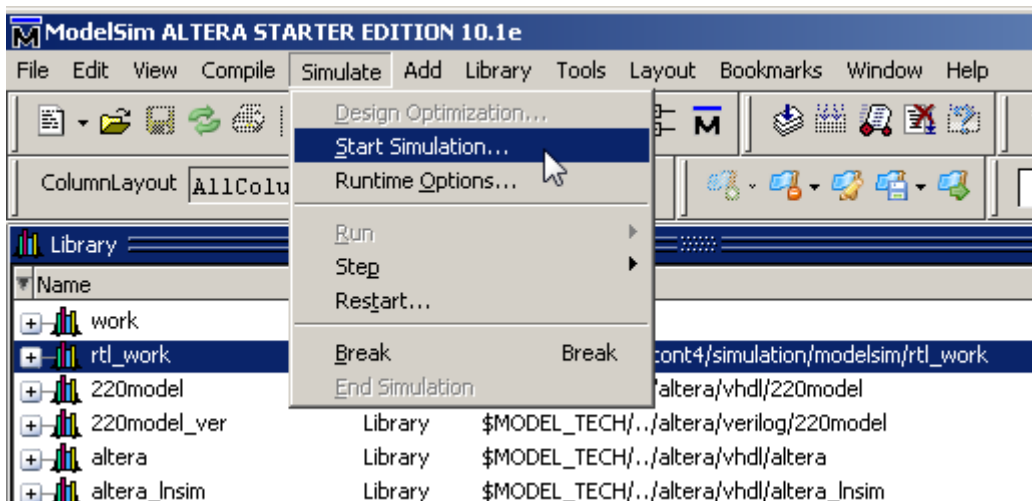
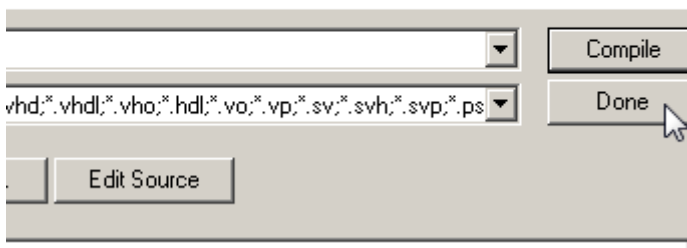
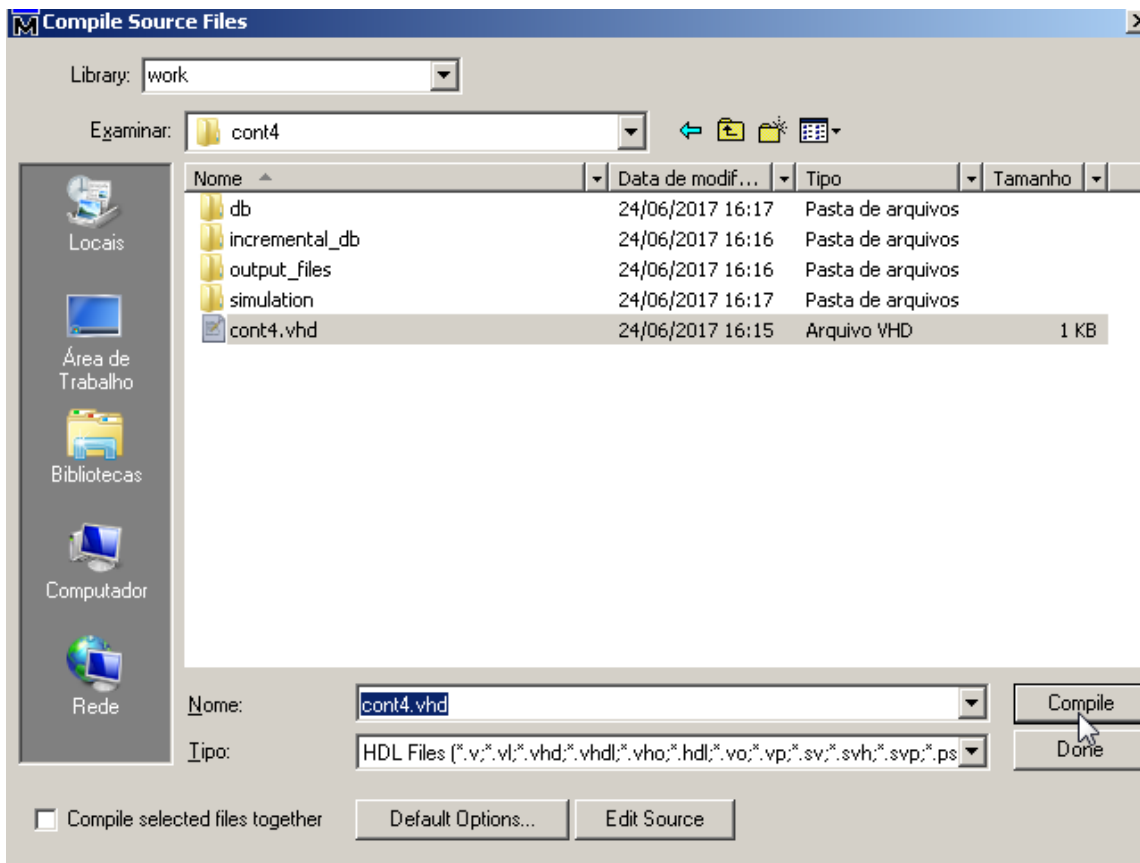
CONTROL H (replace)

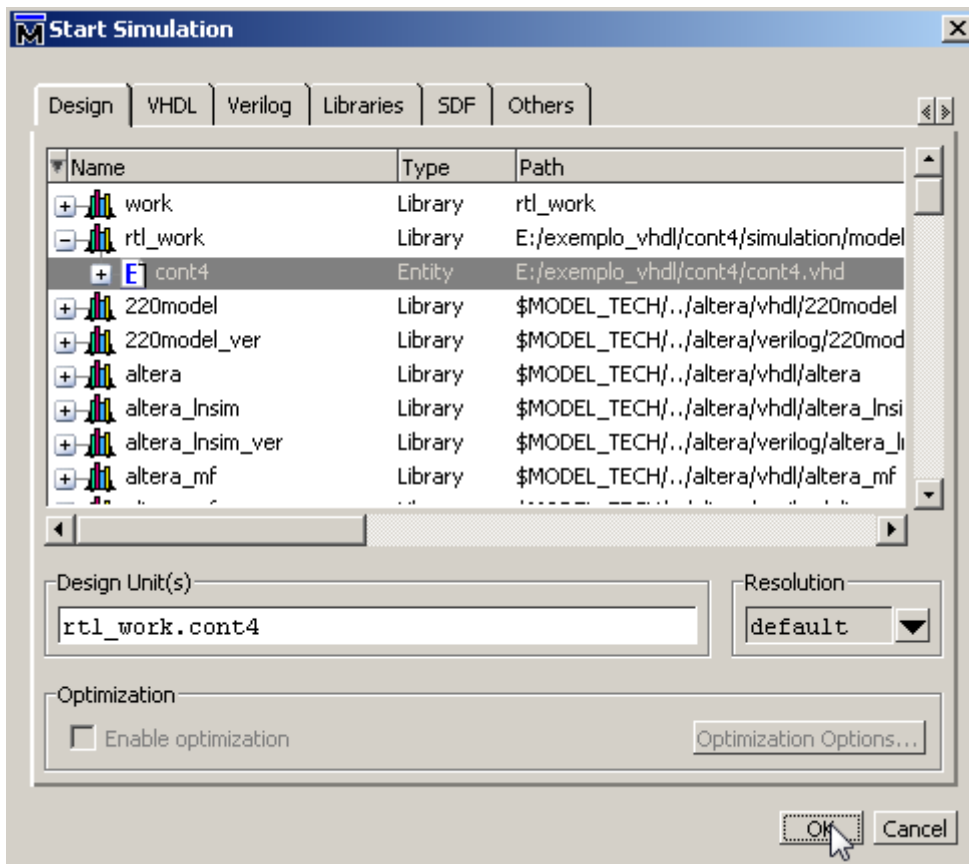


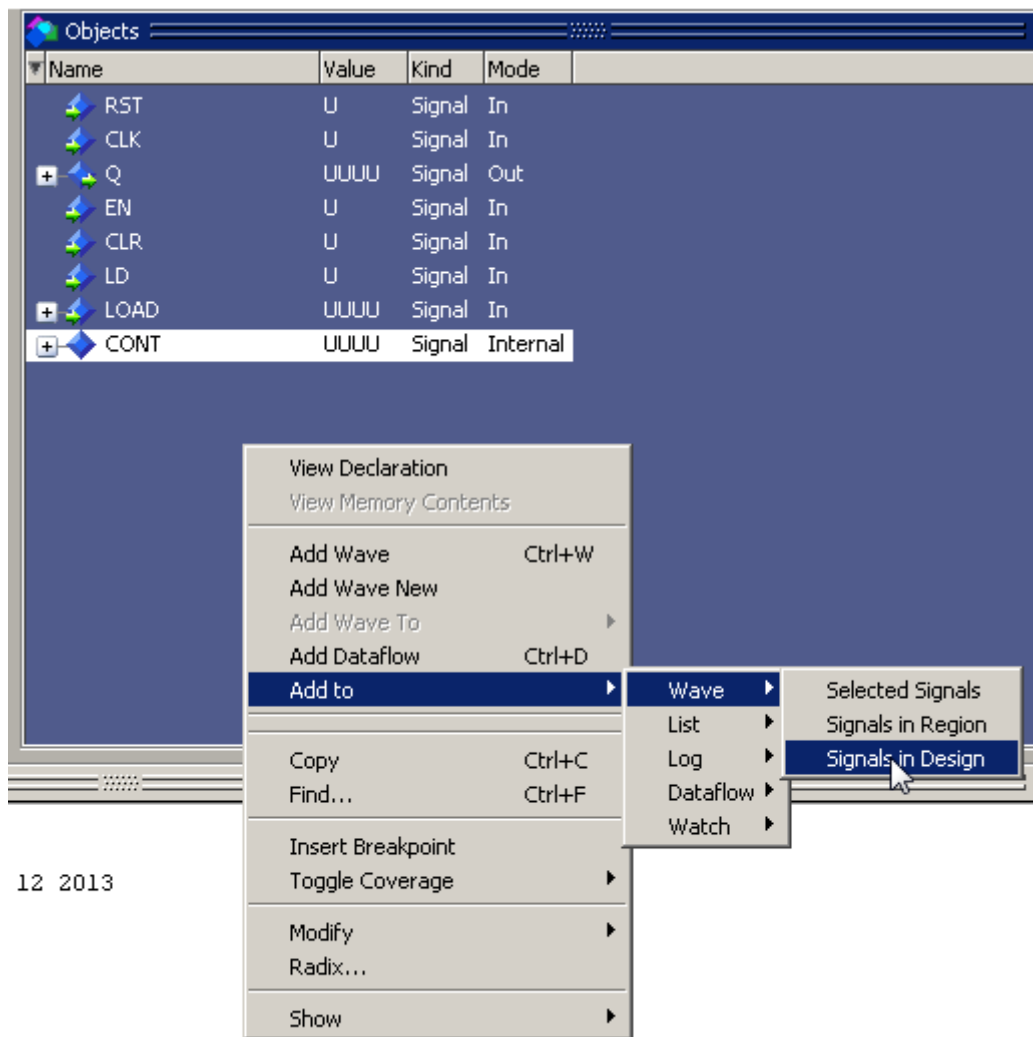












12 2013

Instance	Design unit	Design
cont4	cont4(x)	Archit
line__17	cont4(x)	Proces
line__35	cont4(x)	Proces
standard	standard	Packa
textio	textio	Packa
std_logic_1164	std_logic_1164	Packa
numeric_std	numeric_std	Packa

Library
Memory List
sim

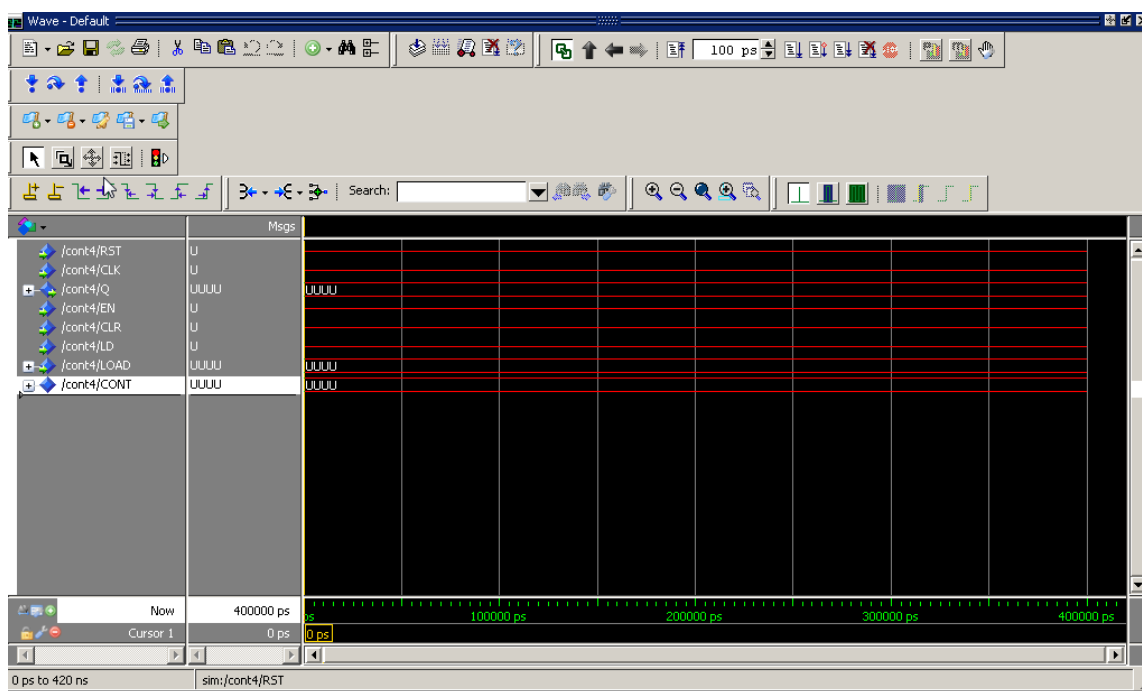
Transcript

```

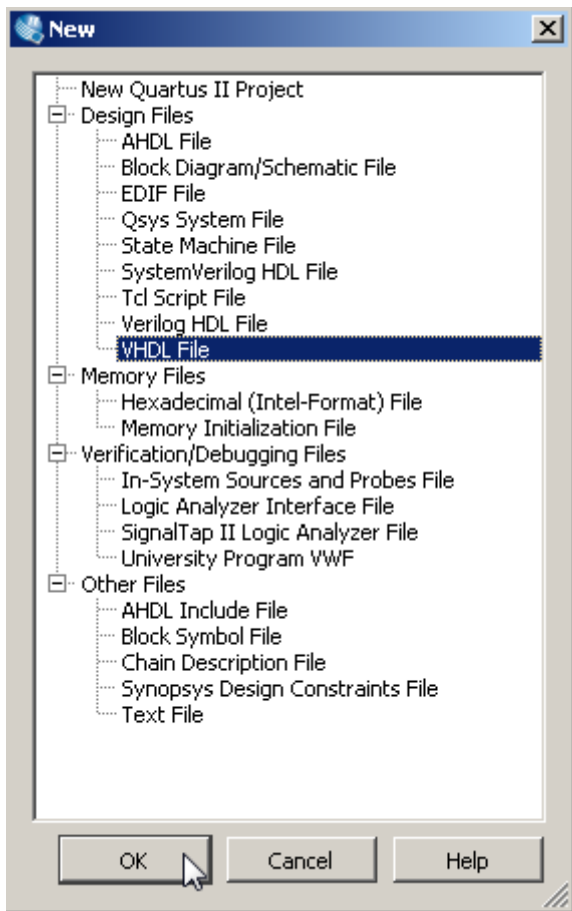
# vcom -93 -work work {E:/exemplo_vhdl/cont4/cont4.
# Model Technology ModelSim ALTERA vcom 10.1e Comp:
# -- Loading package STANDARD
# -- Loading package TEXTIO
# Loading package std_logic_1164
run 400 cont4_run_msim_rtl_vhdl.do
run 400 modelsim.ini
run 400 msim_transcript
run 400 rtl_work

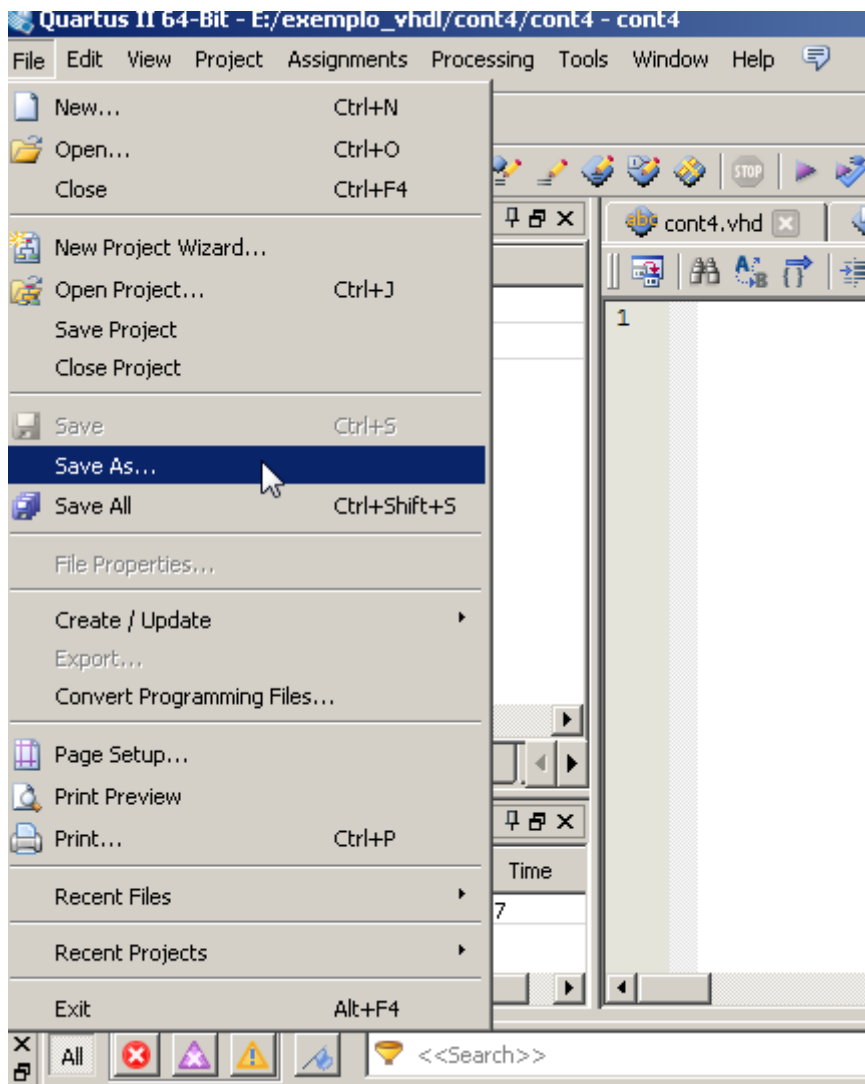
VSIM 6> run 400 ns

```



Construir o testbench





Salvar como cont4_tb.vhd

Cabeçalho do arquivo

```
Library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;
```

```
Entity cont4_tb is
end entity;
```

```
architecture nome_qualquer of cont4_tb is
```

```
component cont4 is
    PORT(RST: in std_logic;
          CLK: in std_logic;
          Q: out unsigned(3 downto 0);
          EN: in std_logic;
          CLR: in std_logic;
```

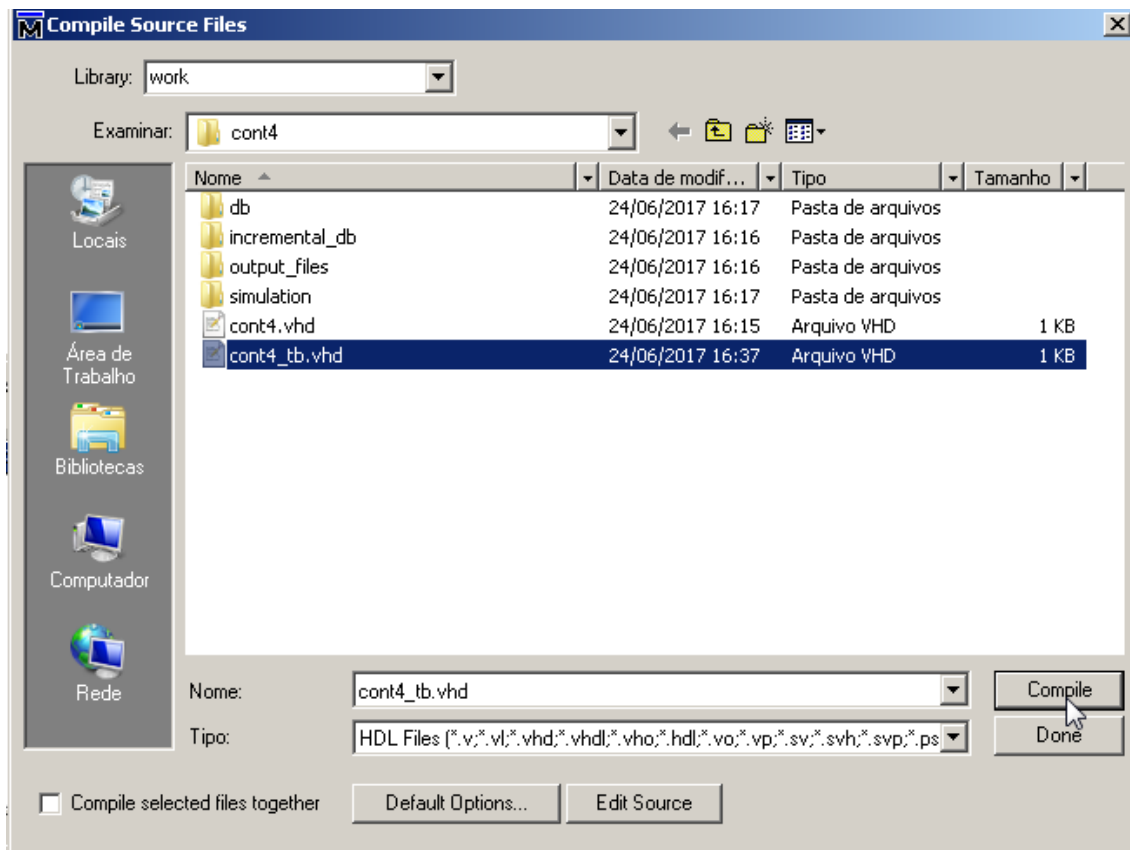
```

        LD: in std_logic;
        LOAD: in unsigned (3 downto 0));
end component;

begin
    DUT: cont4
        port map
            (CLK => clk,
             Q  => Q,
             EN => EN,
             CLR=> clr
             LD  => ld
             LOAD:load);

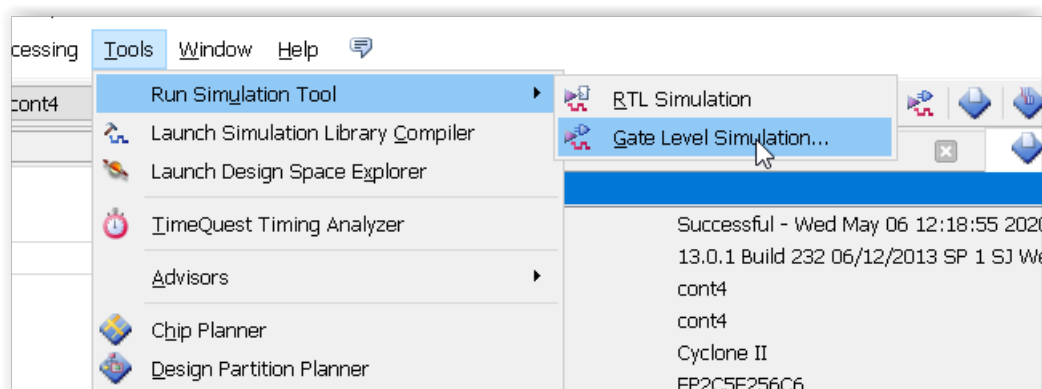
    process
        begin
            clk <= '0';
            wait for 15 ns;
            clk <= '1';
            wait for 15 ns;
        end process;
end architecture;

```

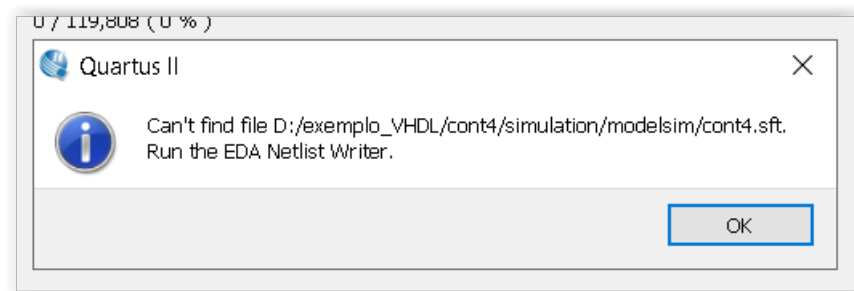


Simulação GATE LEVEL

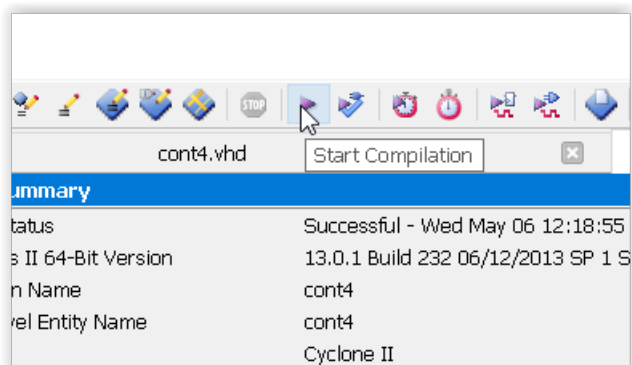
De maneira similar à simulação RTL, invocar a simulação como indicado abaixo



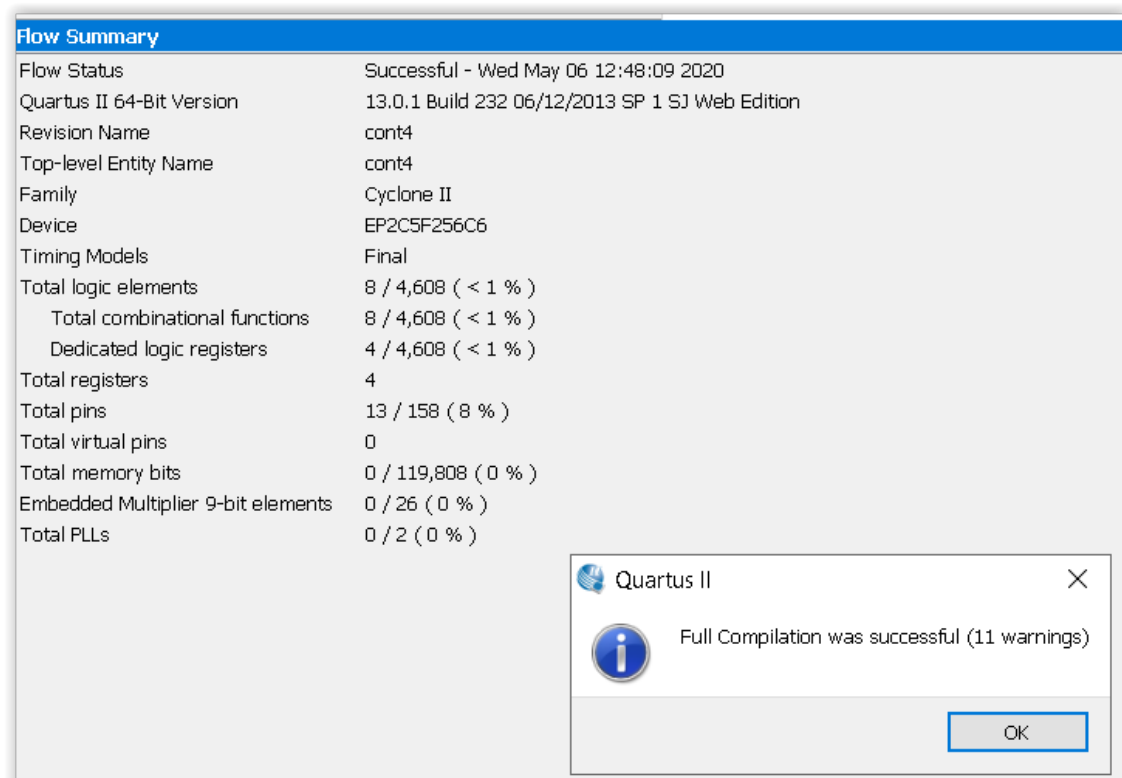
Ao invocar a simulação GATE LEVEL, pode aparecer a seguinte mensagem: “Can’t find <filename>.sft”. Este arquivo com extensão sft é gerado pelo “netlist writer”. Como o próprio nome indica é a ferramenta que gera a lista de conexões entre os elementos da fpga.



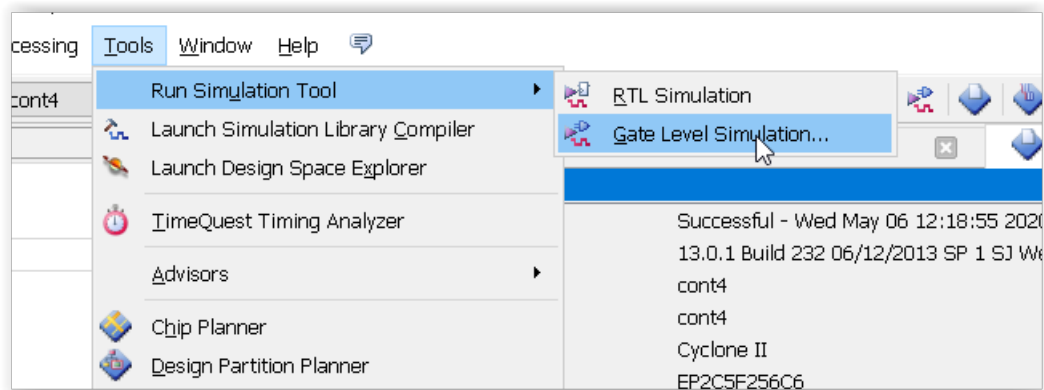
Esta ferramenta é invocada quando a compilação completa é executada, conforme a figura abaixo.



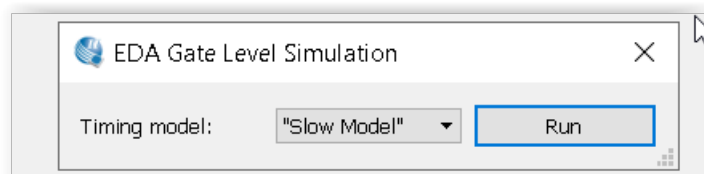
Se a compilação for bem sucedida, a seguinte tela é apresentada.



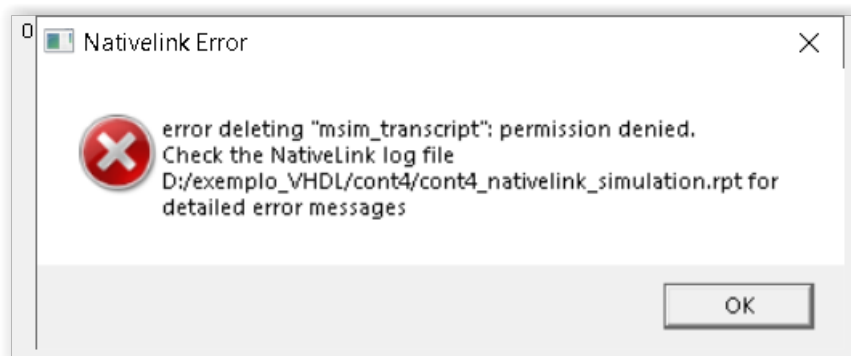
E pode ser invocada novamente a simulação GATE LEVEL:



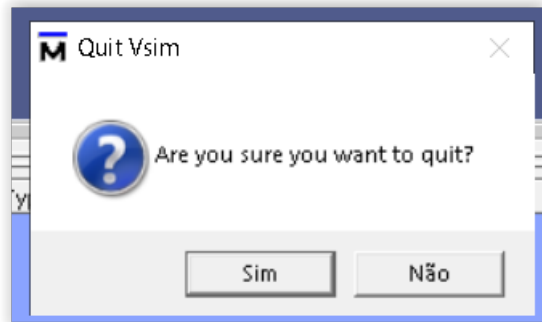
A seguir é mostrada a figura onde se escolhe o tipo de atraso que se deseja. “Slow Model” se refere às condições onde a propagação dos sinais através da fpga leva mais tempo, em geral devido a temperaturas de operação mais alta. “Fast Model” está associado ao funcionamento em temperaturas mais baixas, onde os tempos de propagação são menores. Algumas vezes aparecem as expressões 0C ou 85C que se referem, respectivamente, a 0 grau Celsius e 85 graus Celsius.



Neste ponto o Modelsim deve ser invocado, mas pode aparecer a mensagem abaixo. Ela indica que um elemento do Modelsim não pode ser apagado. Normalmente isso ocorre devido ao fato de o Modelsim estar aberto. Solução: fechat o Modelsim e reinvocar o comando.



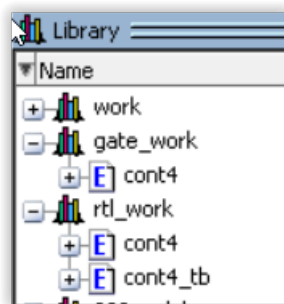
Fechamento do Modesim:



Pressionar “Sim”.

Simulação GATE LEVEL no Modelsim.

Na janela “Library” do Modelsim devem aparecer as janelas “rtl_work” e “gate_work”. Para que o simulador funcione, uma destas bibliotecas deve ser selecionada.



A figura acima ilustra como as bibliotecas são organizadas pelo Modelsim.

Para escolher a biblioteca, existe um comando do Modelsim: “vlib”, da mesma maneira que existe um para o compilador: “vcom” e para o simulador: “vsim”.

_ Para podermos entender a simulação GATE LEVEL, precisamos ir a um nível mais detalhado na simulação RTL.

Quando a simulação RTL é invocada, o “vlib” seleciona a biblioteca de trabalho “rtl_work”. Isto é feito através de scripts.

Já a simulação GATE LEVEL produz um script que chama o comando “vlib” é invocado e seleciona a biblioteca “gate_work”.

Na simulação RTL, o próprio arquivo fonte VHDL, no nosso exemplo cont4.vhd é utilizado em conjunto com cont4_tb.vhd. O arquivo cont4.vhd é compilado através do script:

```
cont4_run_msim_rtl_vhdl.do
```

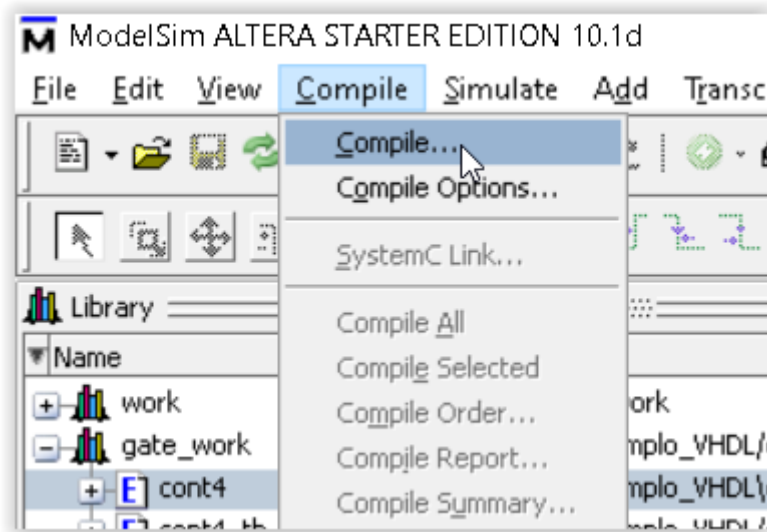
Este script é gerado e colocado para rodar pelo Quartus quando a simulação RTL é invocada.

```

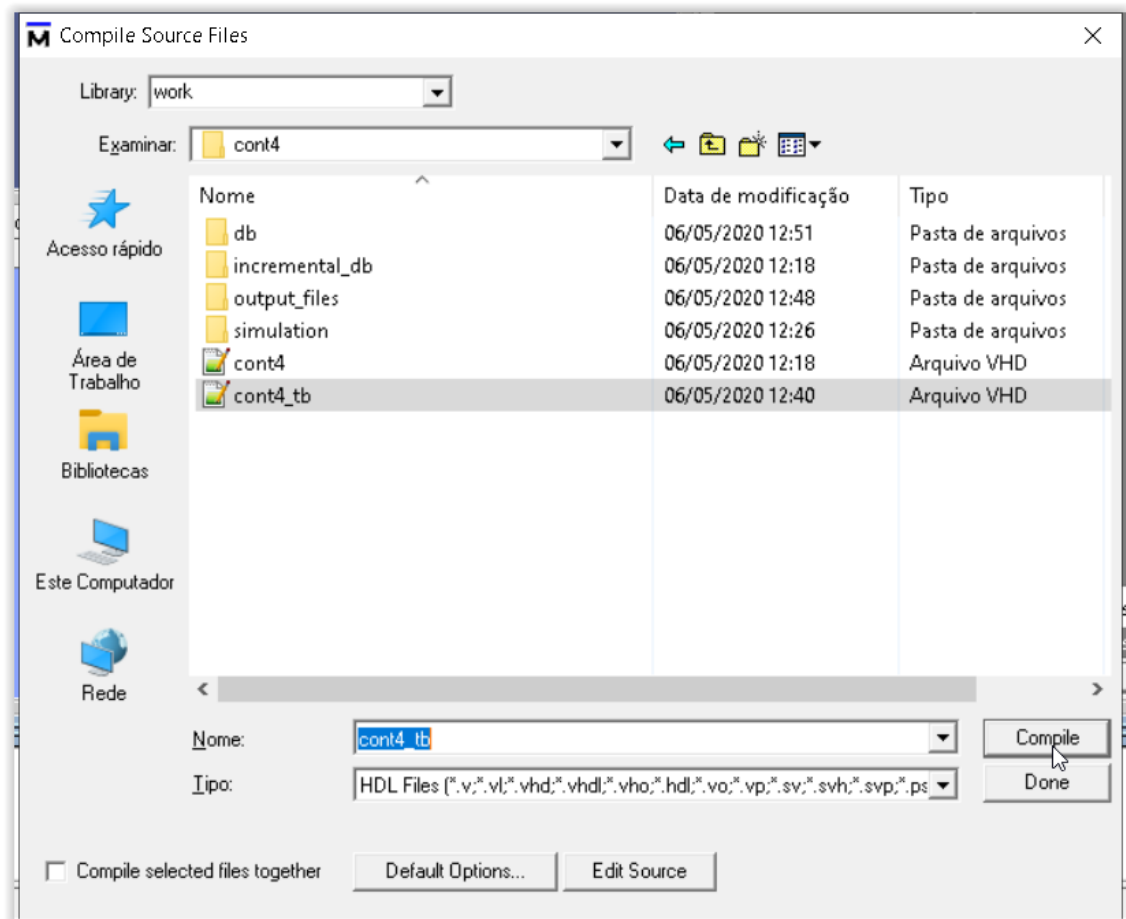
transcript on
if {[file exists rtl_work]} {
    vdel -lib rtl_work -all
}
vlib rtl_work
vmap work rtl_work
vcom -93 -work work {D:/exemplo_VHDL/cont4/cont4.vhd}

```

Mas essa simulação não pode ser completada se o arquivo cont4_tb.vhd não for compilado. Isso é feito através da interface gráfica :



Escolher o diretório onde o arquivo “cont4_tb.vhd” se encontra e compilar o arquivo



Para isso pressionar o botão “compile”.

A simulação efetivamente é feita através da opção “Simulate” do Modelsim. Os demais passos estão descritos em seções anteriores deste documento.

__ Simulação GATE_LEVEL propriamente dita: O Quartus gera o seguinte script:

```
transcript on
if {[file exists gate_work]} {
    vdel -lib gate_work -all
}
vlib gate_work
vmap work gate_work
vcom -93 -work work {cont4.vho}
```

Notar que a biblioteca agora é “gate_work” e o arquivo compilado é “cont4.vho”. Esse arquivo é baseado no vhd original e complementado com os atrasos intrínsecos de cada dispositivo.

De forma similar à simulação RTL, “comp4_tb.vhd” também precisa ser compilado. Se a simulação GATE_LEVEL for feita em seguida à RTL, o tetbench PRECISA ser compilado obrigatoriamente. Na primeira simulação (RTL) ele é compilado em uma biblioteca e na segunda (GATE_LEVEL) em outra.

Para a simulação seguir os mesmos passos anteriores

ANEXO: configurações adicionais necessárias para a simulação GATE LEVEL

1) Identificar a instância a quem deveremos aplicar as anotações quanto ao atraso intrínseco dos diversos elementos dentro da FPGA. Estas anotações são feitas pelo Quartus e ficam dentro de um arquivo com extensão SDF.

Caso o arquivo SDF seja associado a tdo o ambiente de simulação, não haverá anotações para os sinais do testbench. Neste caso a simulação falhará. No exemplo abaixo, o arquivo SDF deverá ser aplicado unicamente à instância DUT.

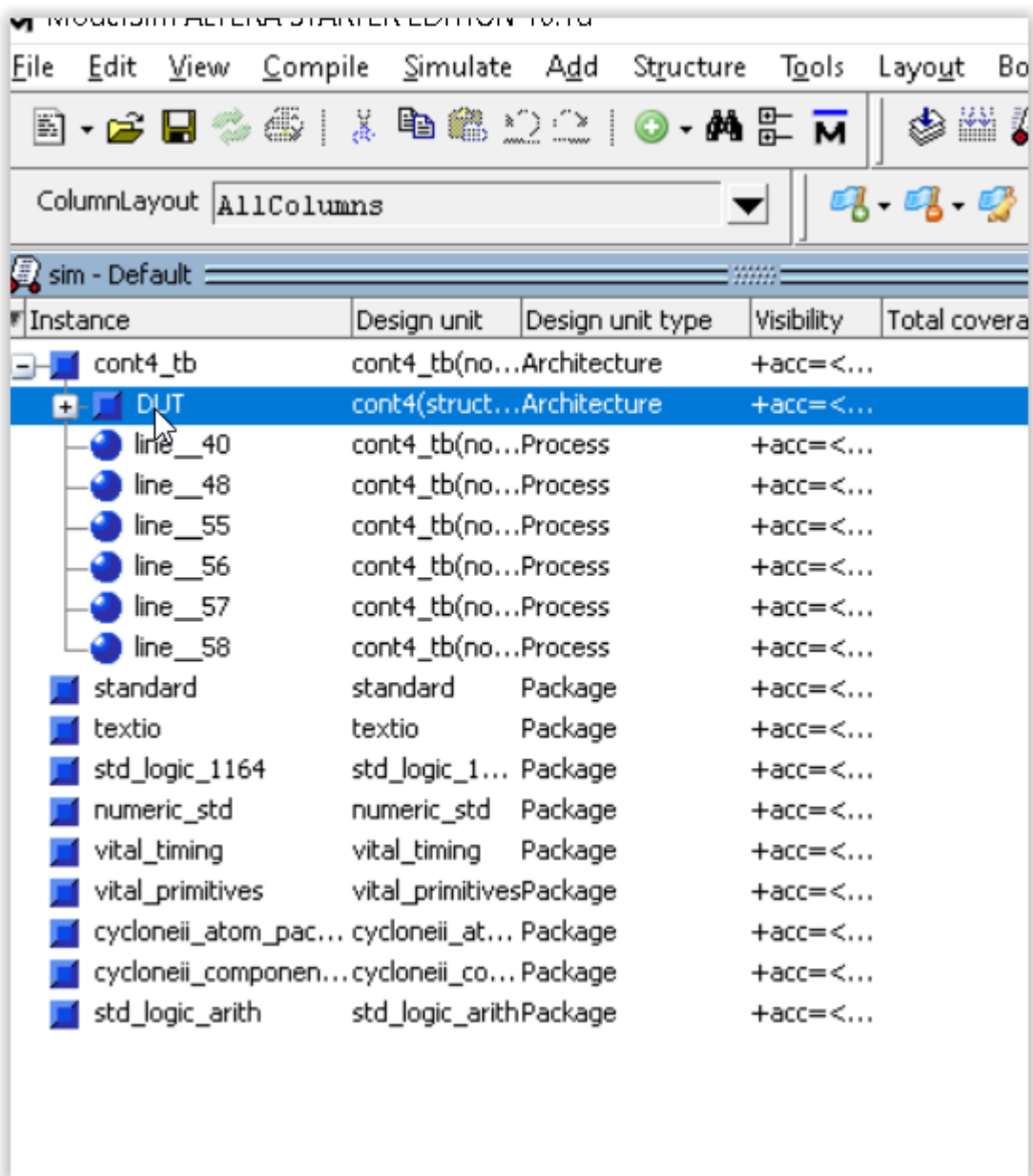


FIG 1

No exemplo em questão esse nome “DUT” é o nome da architecture da entidade que estamos simulando.

```

component cont4 is
  PORT(RST: in std_logic;
        CLK: in std_logic;
        Q: out std_logic_vector(4 downto 0);
        EN: in std_logic;
        CLR: in std_logic;
        LD: in std_logic;
        LOAD: in std_logic_vector (4 downto 0));
end component;
begin
  DUT: cont4
  port map
  (RST => RST,
   CLK => clk,
   Q   => Q,
   EN  => EN,
   CLR => clr,
   LD  => ld,
   LOAD=> load);

```

FIG 2

2) Configuração da simulação com o arquivo SDF. No menu “Simulate” escolher a opção “Start Simulation”.

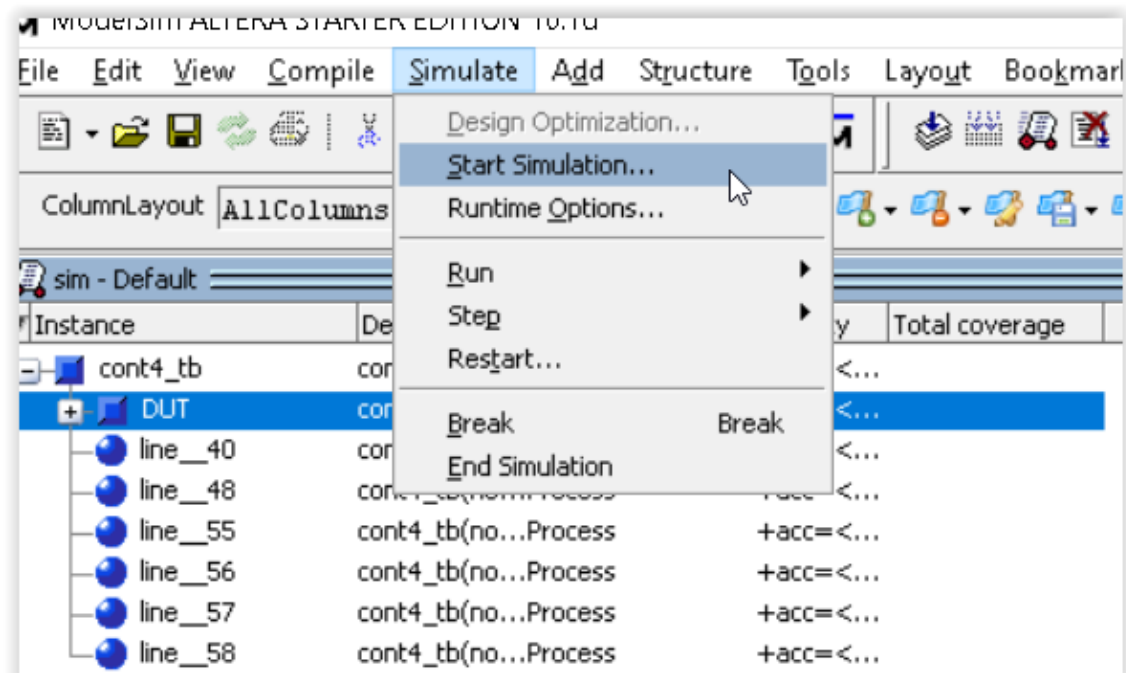


FIG 3

Na aba “SDF”, pressionar Add... Na figura abaixo é mostrada a janela de escolha dos parâmetros. A caixa de diálogo “Apply to Region” contém o valor “/” que significa a aplicação de todas as anotações do arquivo SDF a todo o ambiente de simulação, aí incluso o testbench. O caminho correto, consultando a FIG 1, vemos que o nosso objeto sob teste é “/cont4_tb/DUT”, que deve ser colocado nesta caixa de diálogo.

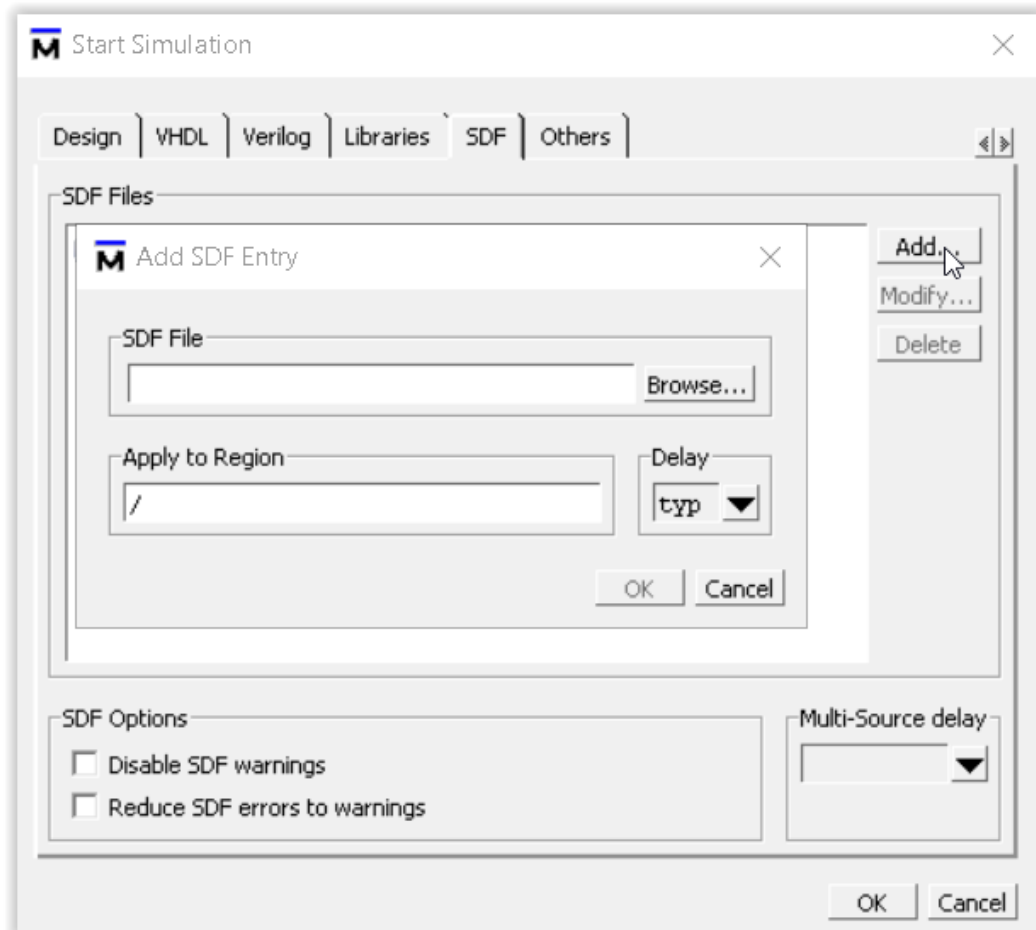


FIG 4

“Browse” permite escolher o arquivo SDF, no exemplo com dois perfis de timing, fast e normal. Escolher aquele adequado à simulação que se deseja fazer -> FIG 5.

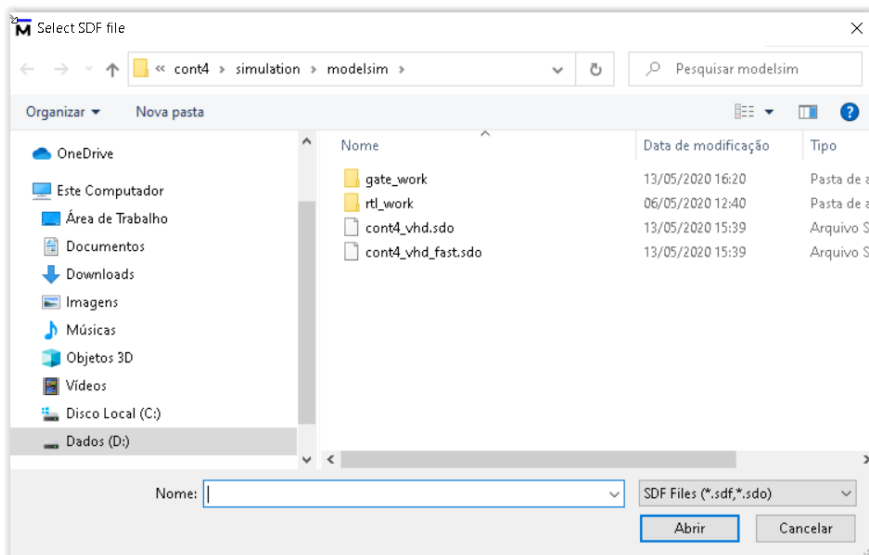


FIG 5

Após a escolha do arquivo (FIG 5) e do correto preenchimento do campo “Apply to Region”, a simulação pode ser invocada com a seleção do testbench e pressionando-se “OK” (FIG 6)

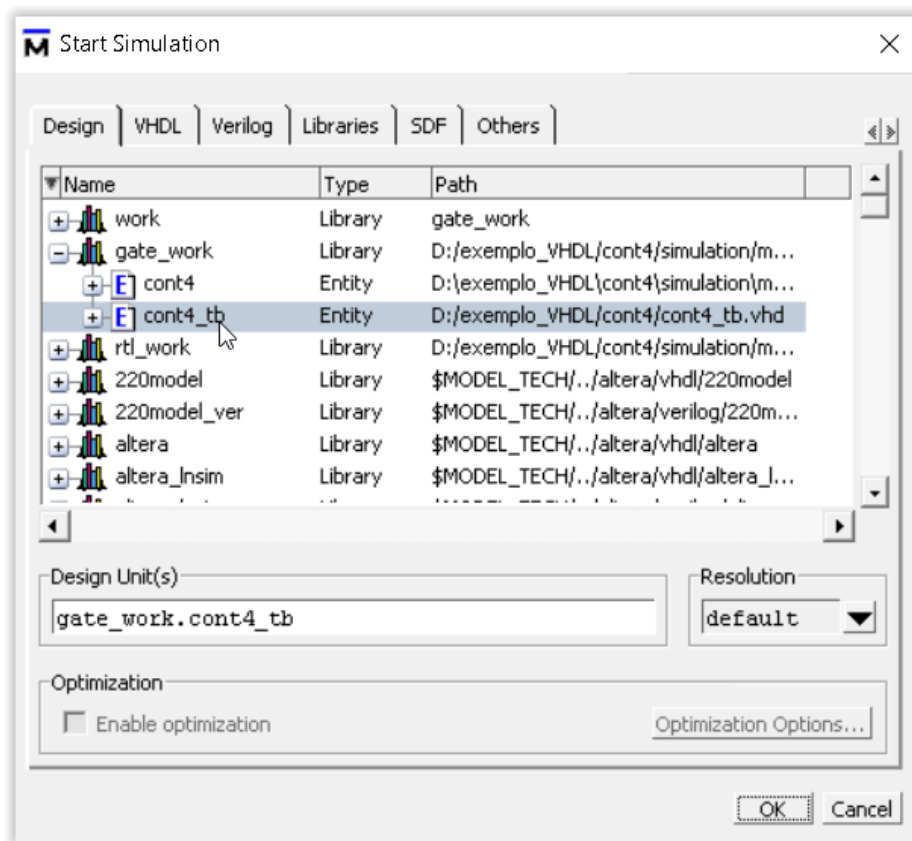


FIG 6

Configurações adicionais para o simulador:

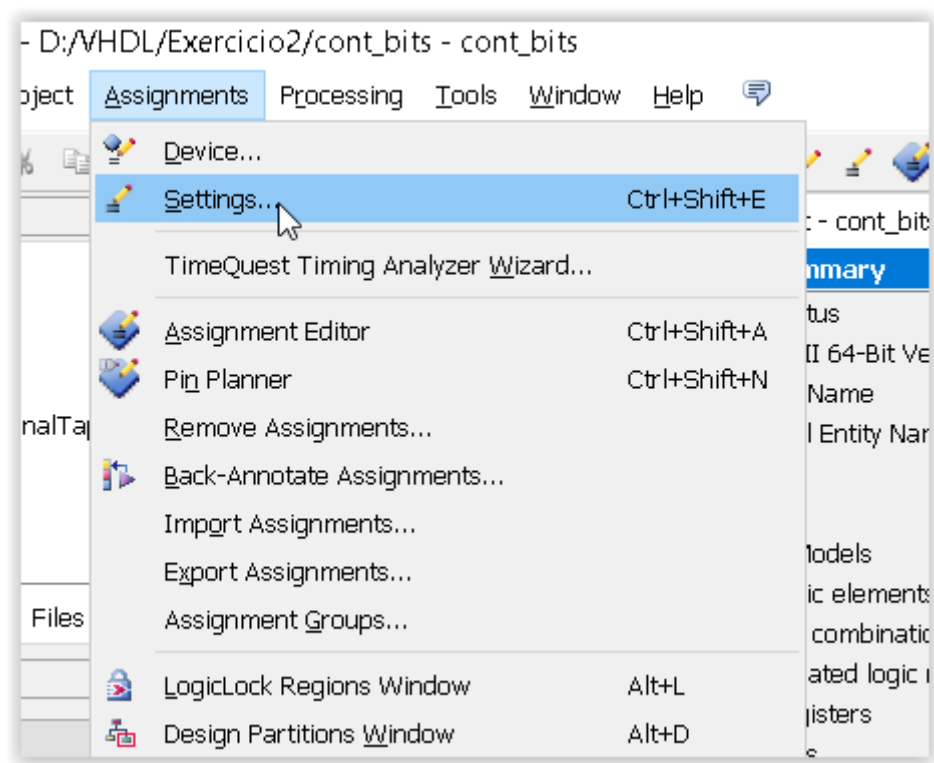


FIG 7

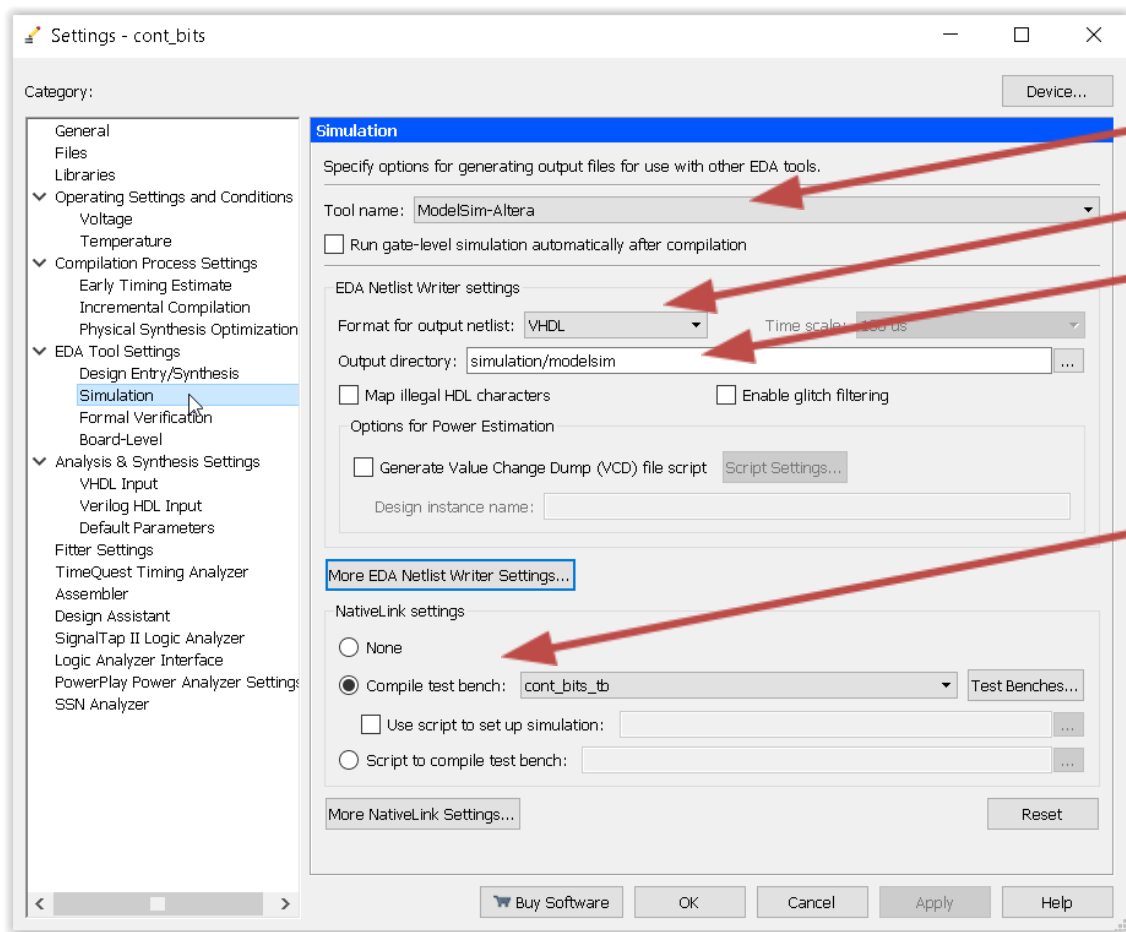


FIG 8

A figura 8 mostra configurações adicionais para a simulação gate level.

Adicionalmente o documento disponível em

https://www.uio.no/studier/emner/matnat/fys/FYS4220/h12/documentation/quartus_models_im_setup.pdf

Pode ser muito útil.