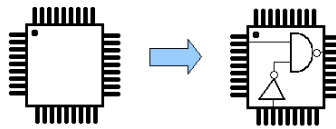


## Capítulo 01 – Projeto em VHDL

### 1 Estrutura de Projeto

O que se espera com o uso de linguagem VHDL é: a partir de uma especificação de projeto efetuar a implementação de um circuito digital e em seguida “gravar” o circuito “sintetizado” com a linguagem VHDL em um Dispositivo Lógico Programável ( Figura 1.1).



(Figura 1.1 – implementação de circuito digital em Lógica Programável)

Para chegar a esta implementação são necessárias as etapas de desenvolvimento apresentadas a seguir no fluxograma de projeto (Figura 1.2).



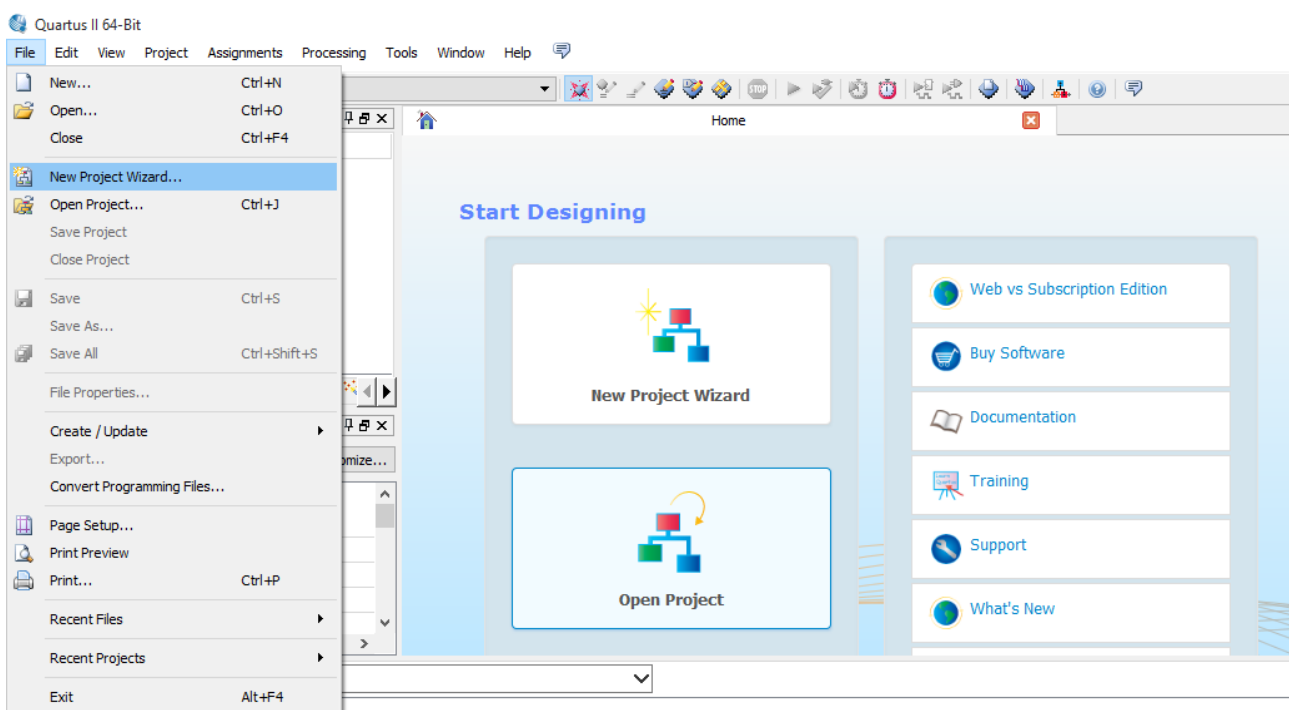
(Figura 1.2 – fluxograma de projeto)

Na fase de “**Escrita do código VHDL**” deve-se descrever o comportamento do circuito digital desejado. Em seguida, é conveniente “**Simular**” o ambiente no qual o circuito será empregado, validando assim o projeto. Para isto é gerado um ou mais arquivos de simulação.

Após a fase de validação (através de simulação) é efetuada a “**Implementação**” do circuito digital no Dispositivo Lógico Programável. Um arquivo contendo as especificações de quais pinos devem ser conectados no projeto deve ter sido gerado previamente. É então gerado um arquivo que configurará o Dispositivo Lógico Programável.

## 2 - Criando um novo Projeto em VHDL

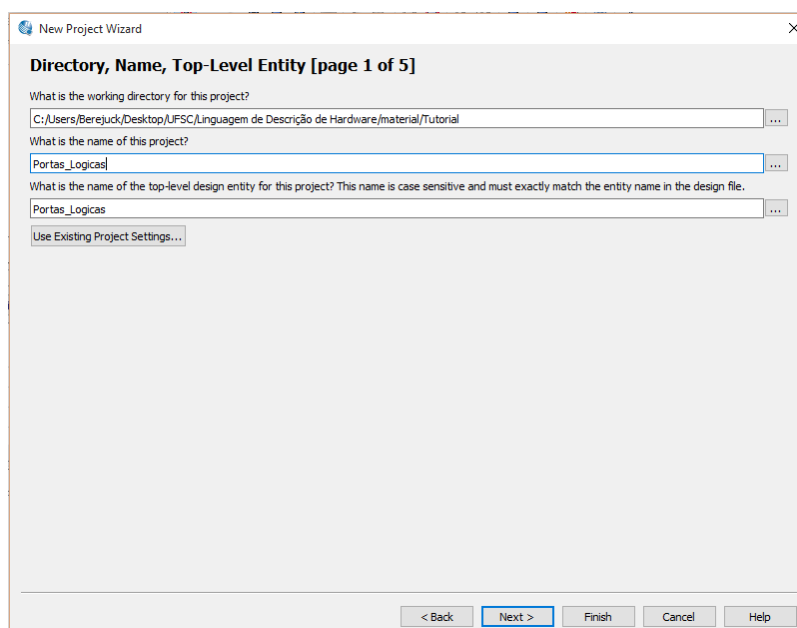
No menu principal selecione: **File** → **New Project Wizard**



Aparecerá uma tela de Introdução. Clique em: **Next >**

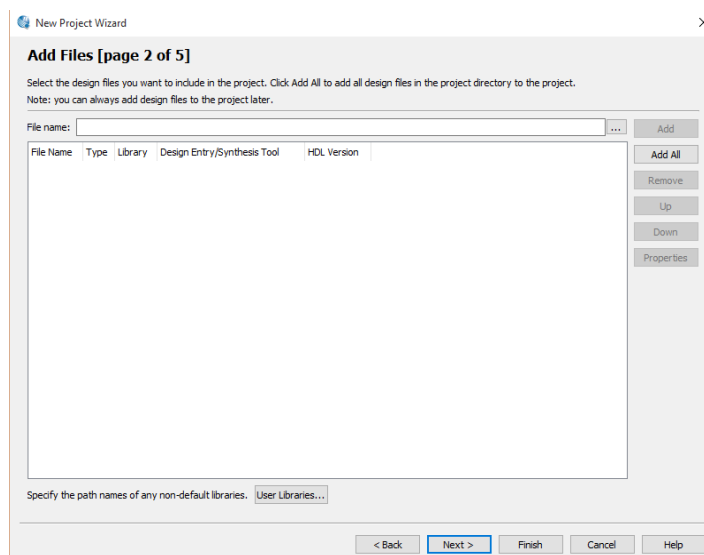
Nos três campos que aparecem a seguir, coloque o caminho (Diretório) onde o seu novo projeto será instalado, o nome do projeto e um nome de arquivo que será usado

como arquivo de referência (top) para o projeto (ver figura a seguir).



Em seguida, clique em: **Next >**

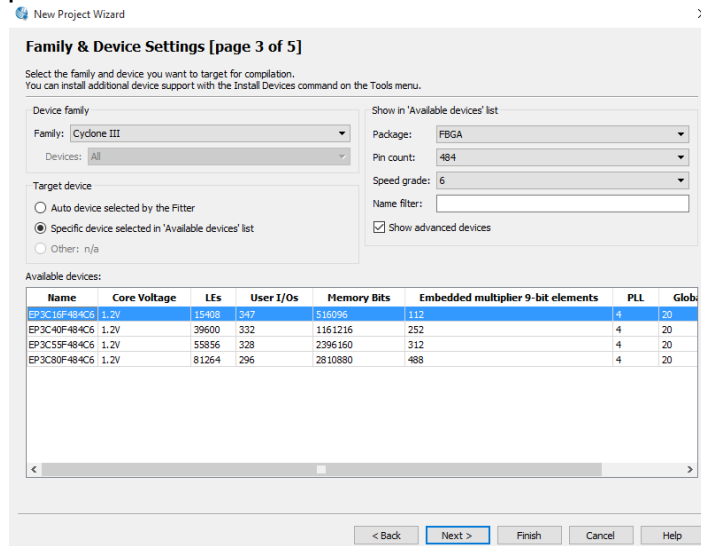
Aparecerá uma tela onde é possível fazer a inclusão de arquivos pré-existent no projeto. Neste tutorial tais arquivos não serão incluídos. Clique em: **Next >**.



A seguir você terá que selecionar o tipo (família) de FPGA que será utilizado no projeto. Na janela de opções selecione:

- Device family: **Cyclone III**
- Package: **FBGA**
- Pin count: **484**
- Speed grade: **6**

Em seguida, clique em: **Next >**



**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.

Device family  
Family: Cyclone III  
Devices: All

Package: FBGA  
Pin count: 484  
Speed grade: 6  
Name filter:   
☒ Show advanced devices

Target device  
☐ Auto device selected by the Fitter  
☒ Specific device selected in 'Available devices' list  
☐ Other: n/a

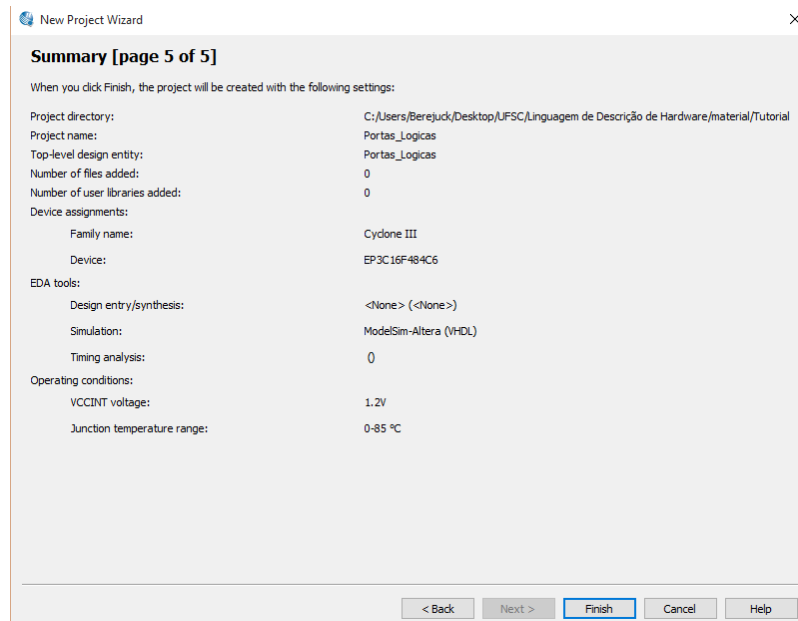
Available devices:

Name	Core Voltage	LEs	User I/Os	Memory Bits	Embedded multiplier 9-bit elements	PLL	Global
EP3C10F484C6	1.2V	15408	347	516096	112	4	20
EP3C40F484C6	1.2V	39600	332	1161216	252	4	20
EP3C59F484C6	1.2V	55856	328	2396160	312	4	20
EP3C80F484C6	1.2V	81264	296	2810880	488	4	20

< Back Next > Finish Cancel Help

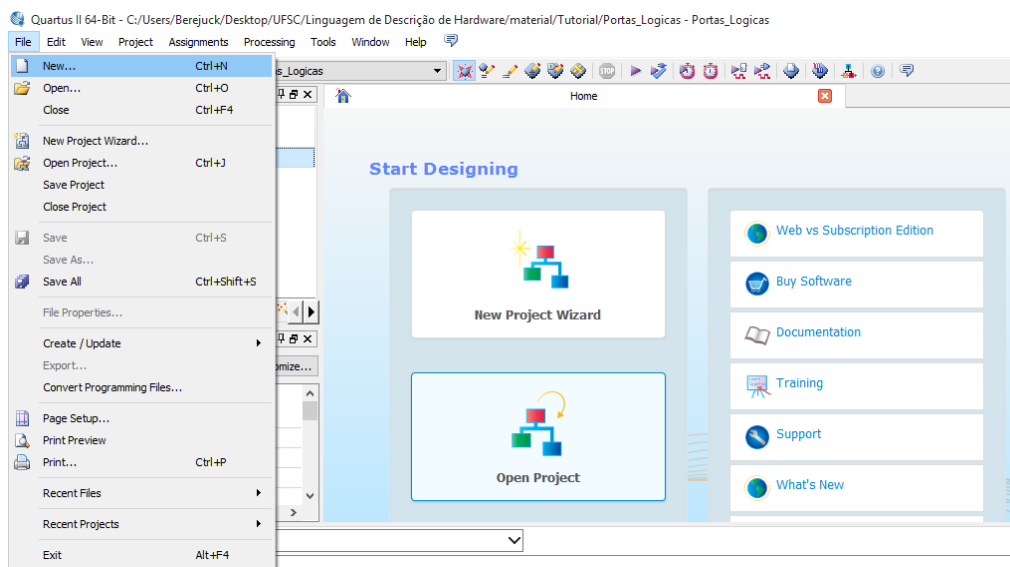
Na janela **EDA Tool Settings** mantenha os parâmetros de *Simulation* como **ModelSim Altera, VHDL**. Em seguida clique em **Next >**

Será apresentada uma janela com um resumo das opções do projeto. Verifique se são os parâmetros selecionados anteriormente e clique em **Finish** (ver figura a seguir)

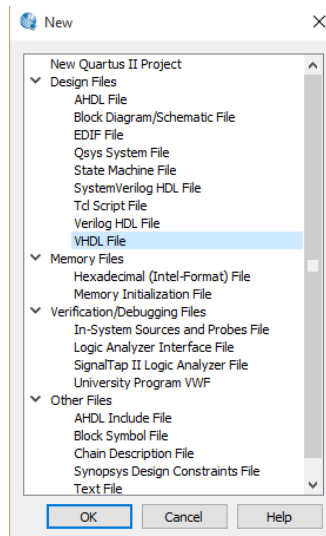


Uma vez que o projeto foi criado, é necessário incluir o arquivo com as implementações do projeto (ou arquivos, no caso de um projeto maior).

No menu de opções selecione: **File** → **New**



Selecione a opção **VHDL File** (ver figura a seguir)



No novo arquivo, escreva o seguinte código VHDL:

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_UNSIGNED.ALL;
4  use IEEE.STD_LOGIC_ARITH.ALL;
5
6
7  entity Portas_Logicas is
8  Port ( i_PINO_A : in STD_LOGIC;
9        i_PINO_B : in STD_LOGIC;
10       o_PINO_C : in STD_LOGIC
11       );
12  end Portas_Logicas;
13
14  architecture Behavioral of Portas_Logicas is
15
16  begin
17     o_PINO_C <= i_PINO_A and i_PINO_B;
18
19  end Behavioral;
20
```

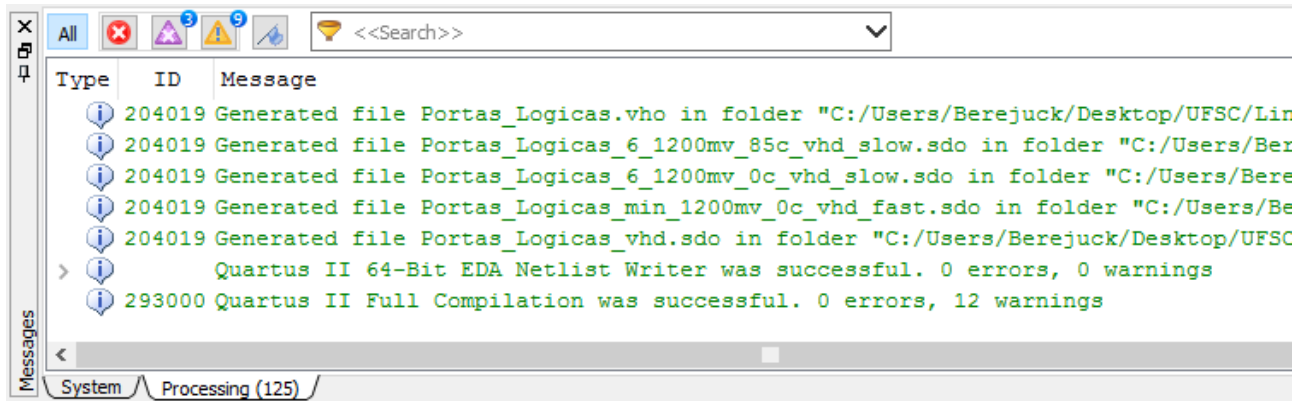
Em seguida, salve o arquivo criado (ver figura a seguir). Ele deverá ter o nome indicado no início da preparação do projeto (Portas\_Logicas):



Clique em **compilar o projeto** no menu de ícones:



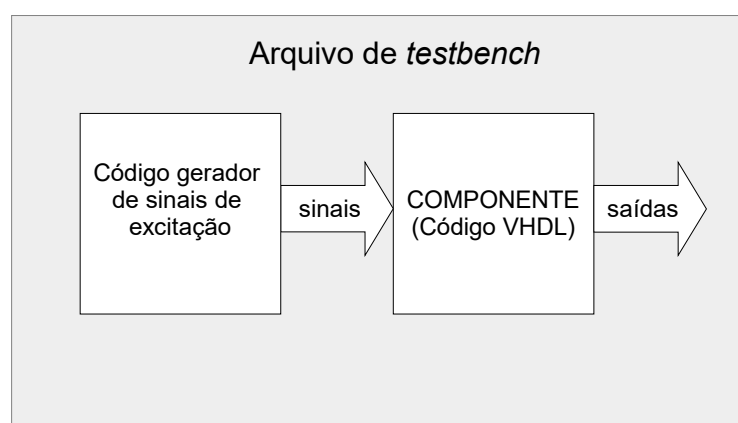
Se não houver erros de digitação, você receberá uma mensagem de *0 errors* na parte inferior da ferramenta:



### 3 Simulação com uso da ferramenta ModelSim

A simulação de projetos em VHDL para os dispositivos lógicos programáveis do fabricante Altera é feita através de uma ferramenta de simulação denominada de ModelSim, do fabricante Mentor Graphyics®.

O processo de simulação está baseado na escrita de um arquivo em VHDL específico para gerar sinais de excitação, que irão ser conectados à entrada do componente VHDL que queremos simular. Assim, o código VHDL do nosso projeto é instanciado como um componente dentro do arquivo de *testbench*, conforme mostra a Figura a seguir.



### 3.1 Configurando o Quartus II para acesso direto ao ModelSim

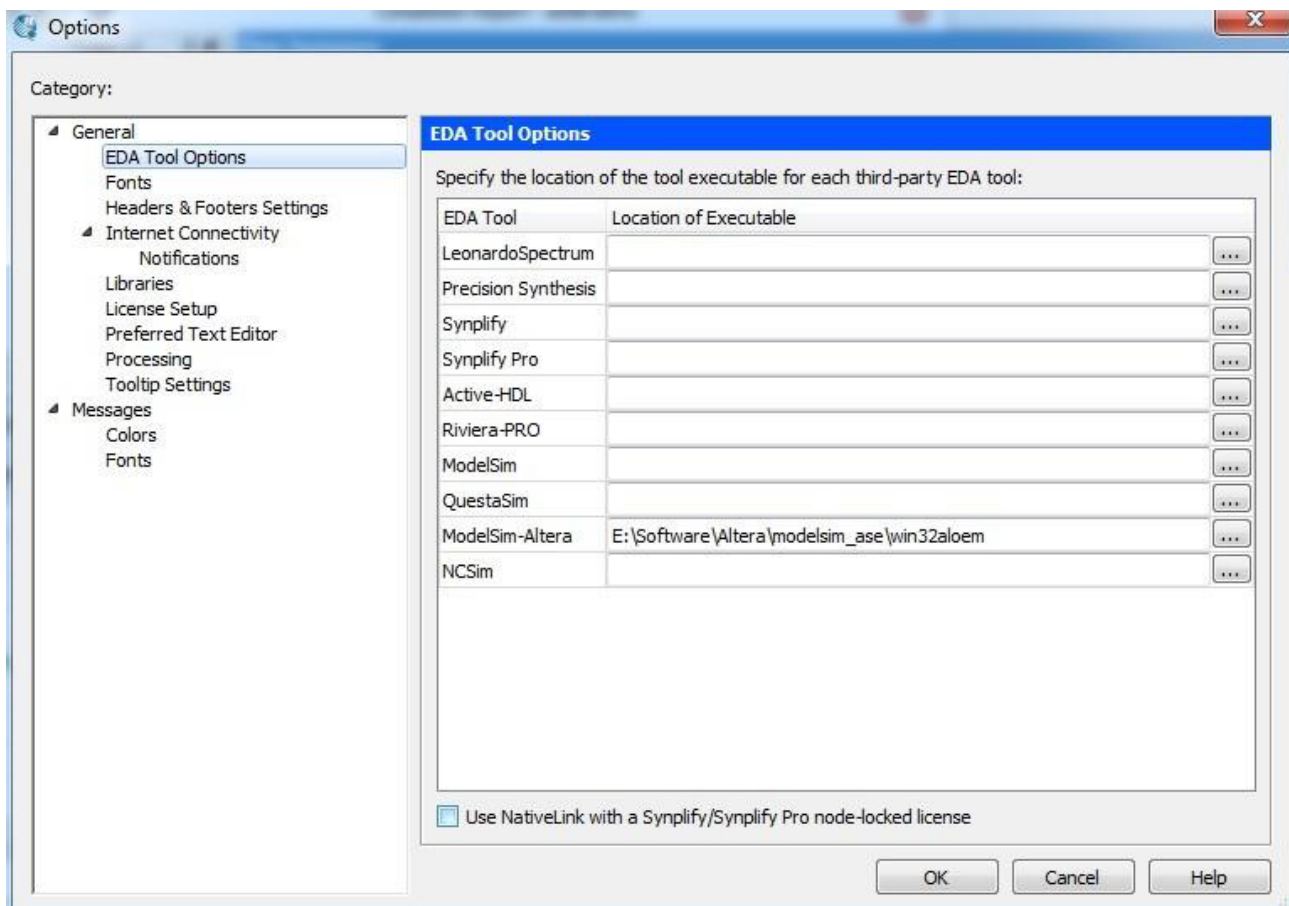
É possível acessar a ferramenta ModelSim a partir da ferramenta Quartus II. Para isso é necessário alterar os seguintes parâmetros de configuração:

#### 3.1.1. Set up do EDA tool options

No menu principal da ferramenta Quartus II Selecione **Tools** → **Options** para abrir a janela de opções. Em **General**, selecione **EDA Tool Option** à esquerda como mostra a Figura a seguir. Clique no botão de **Procurar** à direita:



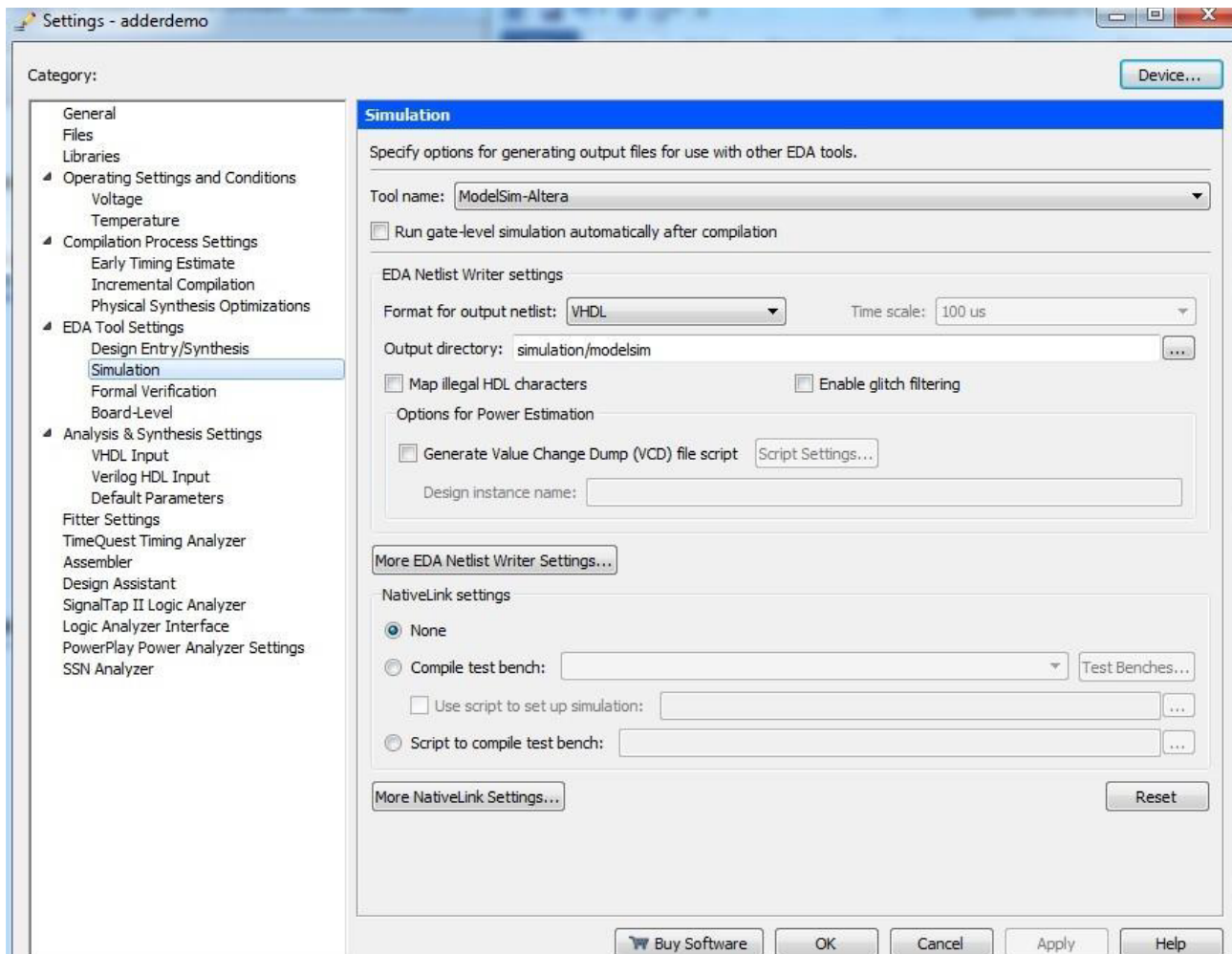
para a opção **ModelSim-Altera** na lista EDA ferramenta para especificar a localização do executável ModelSim-Altera. Finalmente, clique em OK.





### 3.1.2. Set up de Simulação

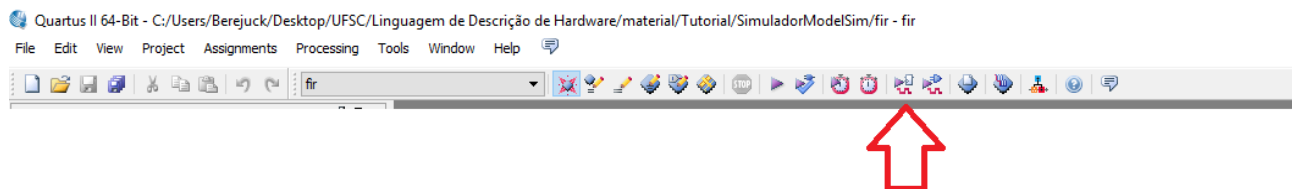
Selecione **Assignment** → **Settings**. Na caixa de diálogo **Category**, em **EDA Tool Settings**, selecione **Simulation**, como mostrado na Figura a seguir.



Na lista **Tool name**, selecione ModelSim-Altera. Certifique-se de que a opção **Run gate level simulation automatically**, abaixo de **Tool name**, está desligada. Em **EDA Netlistt Writer settings**, na opção **Format for output netlist**, selecione VHDL.

Certifique-se de que as opções **Map for Power Estimation** e **Generate Value Change Dump (VCD) for file script** estão desligadas. Em **NativeLink**, selecione **None**. Finalmente, clique em **Apply** e **OK**.

Para que o ModelSim seja chamado de dentro da ferramenta Quartus II, selecione a opção **Tools** → **Run Simulation Tool** → **RTL simulation** ou então use o ícone conforme indica a Figura a seguir:



### 3.2 Exemplo de Simulação usando o ModelSim

Crie um novo projeto e inclua um arquivo VHDL com o nome FIR, contendo o seguinte código VHDL:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use IEEE.STD_LOGIC_SIGNED.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;

entity FIR is
    generic
    (
        p_DATA_WIDTH : natural := 8
    );
    port
    (
        i_CLK          : in std_logic;
        i_RST          : in std_logic;
        i_SAMPLE       : in std_logic_vector ((p_DATA_WIDTH-1) downto 0);
        i_COEF         : in std_logic_vector ((p_DATA_WIDTH-1) downto 0);
        i_CLEAR        : in std_logic;
        i_MAC           : in std_logic;
        o_RESULT       : out std_logic_vector ((p_DATA_WIDTH-1) downto 0)
    );
end entity;

architecture rtl of FIR is
    signal w_PRODUTO    : std_logic_vector((2*p_DATA_WIDTH-1) downto 0);
    signal w_RESULT     : std_logic_vector((p_DATA_WIDTH-1) downto 0);
    signal w_ZERO_MUX   : std_logic_vector((p_DATA_WIDTH-1) downto 0);
    signal w_SOMA       : std_logic_vector((p_DATA_WIDTH-1) downto 0);

begin
    process(i_CLK)
    begin
        if rising_edge(i_CLK) then
            if (i_RST = '1') then
```

```

        w_RESULT <= (OTHERS => '0');
    else
        if (i_MAC = '1') then
            w_RESULT <= w_RESULT + w_SOMA;
        end if;
    end if;
end if;
end process;

w_SOMA <= w_PRODUTO((p_DATA_WIDTH-1) downto 0) + w_ZERO_MUX;

w_ZERO_MUX <= (others => '0') when (i_CLEAR = '1') else w_RESULT;

w_PRODUTO <= i_COEF * i_SAMPLE;
o_RESULT <= w_RESULT;

end rtl;

```

O código acima é uma implementação simples de um filtro FIR (acrônimo de *Finite Impulse Response*). Para simular o seu funcionamento iremos criar um novo arquivo, chamado de FIR\_TB, o qual irá conter uma instancialização do filtro FIR e mais uma certa quantidade de código para gerar os sinais de excitação. Coloque no arquivo FIR\_TB o seguinte código VHDL:

```

LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

```

```

ENTITY FIR_TB IS
END FIR_TB;

```

```

ARCHITECTURE behavior OF FIR_TB IS

```

```

    -- Component Declaration for the Unit Under Test (UUT)

```

```

    COMPONENT FIR
    GENERIC
    (
        p_DATA_WIDTH : natural := 8
    );
    PORT
    (
        i_CLK          : in std_logic;
        i_RST          : in std_logic;
        i_SAMPLE       : in std_logic_vector ((p_DATA_WIDTH-1) downto 0);
        i_COEF         : in std_logic_vector ((p_DATA_WIDTH-1) downto 0);
        i_CLEAR        : in std_logic;
        i_MAC          : in std_logic;
        o_RESULT       : out std_logic_vector ((p_DATA_WIDTH-1) downto 0)
    );

    END COMPONENT;

```

```

--Inputs
signal i_CLK : std_logic := '0';
signal i_RST : std_logic := '0';

```

```
signal i_SAMPLE : std_logic_vector(7 downto 0) := (others => '0');

--Outputs
signal i_COEF   : std_logic_vector(7 downto 0);
signal o_RESULT : std_logic_vector(7 downto 0);
signal i_CLEAR  : std_logic;
signal i_MAC    : std_logic;

-- Clock period definitions
constant i_CLK_period : time := 10 ns;
```

#### BEGIN

```
-- Instantiate the Unit Under Test (UUT)
uut: FIR PORT MAP (
    i_CLK      => i_CLK,
    i_RST      => i_RST,
    i_SAMPLE   => i_SAMPLE,
    i_COEF     => i_COEF,
    i_CLEAR    => i_CLEAR,
    i_MAC      => i_MAC,
    o_RESULT   => o_RESULT
);

-- Clock process definitions
i_CLK_process :process
begin
    i_CLK <= '0';
    wait for i_CLK_period/2;
    i_CLK <= '1';
    wait for i_CLK_period/2;
end process;

-- Stimulus process
stim_proc: process
begin
    -- hold reset state for 100 ns.
    wait for 100 ns;

    wait for i_CLK_period*10;

    -- insert stimulus here
    i_RST <= '1';
    WAIT FOR 40 NS;
    i_RST <= '0';

    i_CLEAR <= '0';
    i_SAMPLE <= x"12";
    i_COEF   <= x"05";
    i_MAC    <= '1';
    wait for i_CLK_period;
    i_MAC    <= '0';

    i_CLEAR <= '1';
    i_SAMPLE <= x"13";
    i_COEF   <= x"06";
```

```
i_MAC    <= '1';
wait for i_CLK_period;
i_MAC    <= '0';

i_CLEAR  <= '1';
i_SAMPLE <= x"14";
i_COEF   <= x"07";
i_MAC    <= '1';
wait for i_CLK_period;
i_MAC    <= '0';

i_CLEAR  <= '1';
i_SAMPLE <= x"15";
i_COEF   <= x"06";
i_MAC    <= '1';
wait for i_CLK_period;
i_MAC    <= '0';

i_CLEAR  <= '1';
i_SAMPLE <= x"16";
i_COEF   <= x"04";
i_MAC    <= '1';
wait for i_CLK_period;
i_MAC    <= '0';

i_CLEAR  <= '0';
i_SAMPLE <= x"15";
i_COEF   <= x"05";
i_MAC    <= '1';
wait for i_CLK_period;
i_MAC    <= '0';

i_CLEAR  <= '1';
i_SAMPLE <= x"14";
i_COEF   <= x"06";
i_MAC    <= '1';
wait for i_CLK_period;
i_MAC    <= '0';

i_CLEAR  <= '1';
i_SAMPLE <= x"13";
i_COEF   <= x"07";
i_MAC    <= '1';
wait for i_CLK_period;
i_MAC    <= '0';

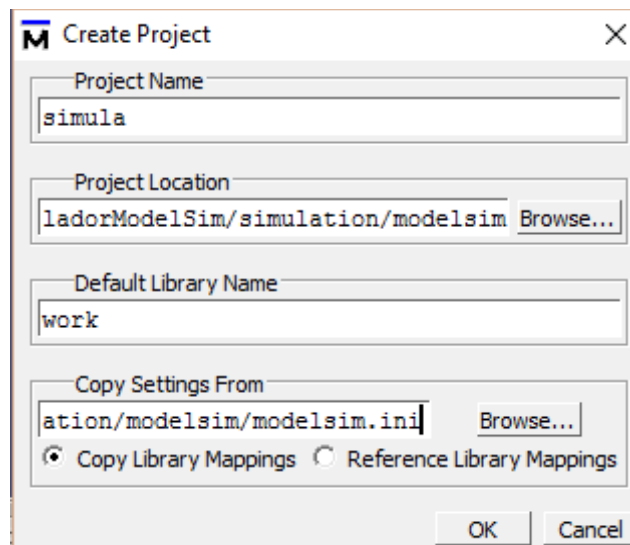
i_CLEAR  <= '1';
i_SAMPLE <= x"12";
i_COEF   <= x"06";
i_MAC    <= '1';
wait for i_CLK_period;
i_MAC    <= '0';

i_CLEAR  <= '1';
i_SAMPLE <= x"13";
i_COEF   <= x"07";
i_MAC    <= '1';
wait for i_CLK_period;
i_MAC    <= '0';
```

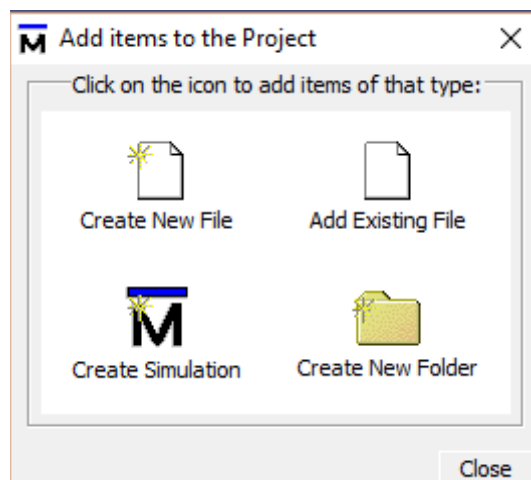
```
wait;  
end process;
```

END;

Abra a ferramenta ModelSim conforme explicado no final da Seção 3.1. No menu da ferramenta ModelSim selecione **File** → **New** → **Project**. Irá aparecer uma tela conforme mostra a figura a seguir na qual você deverá dar um nome ao projeto (no exemplo, simula) e clique em **OK**.

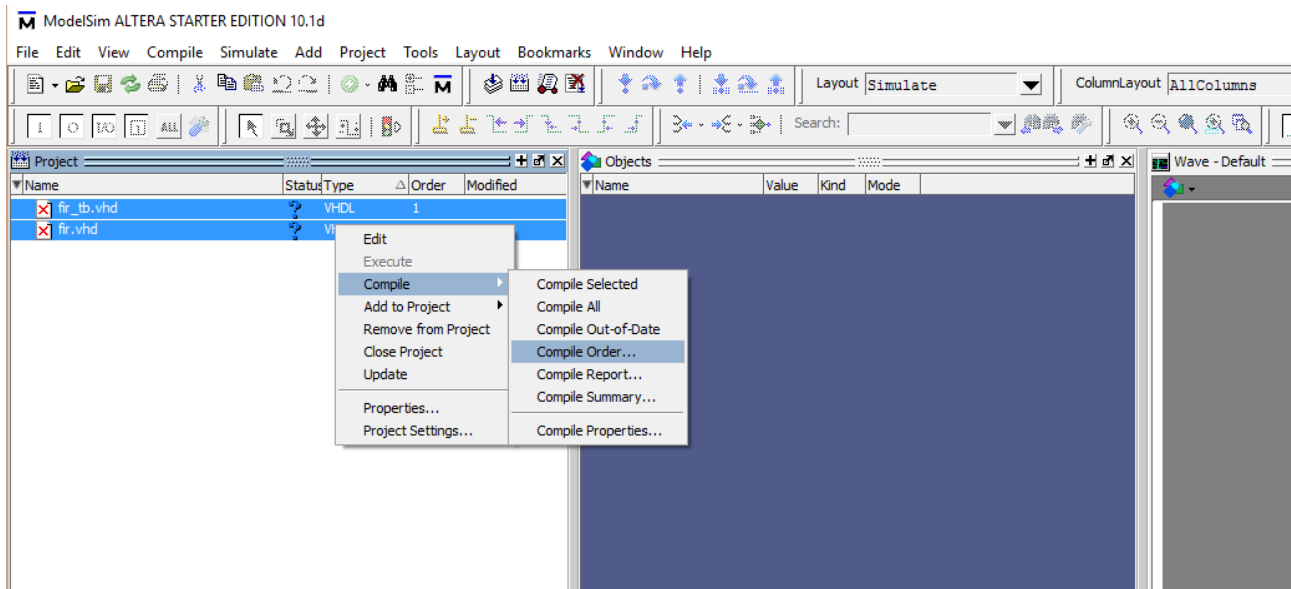


A seguir, o ModelSim irá apresentar uma janel de opções para a inclusão de arquivos neste projeto. Clique em **Add Existing File**:

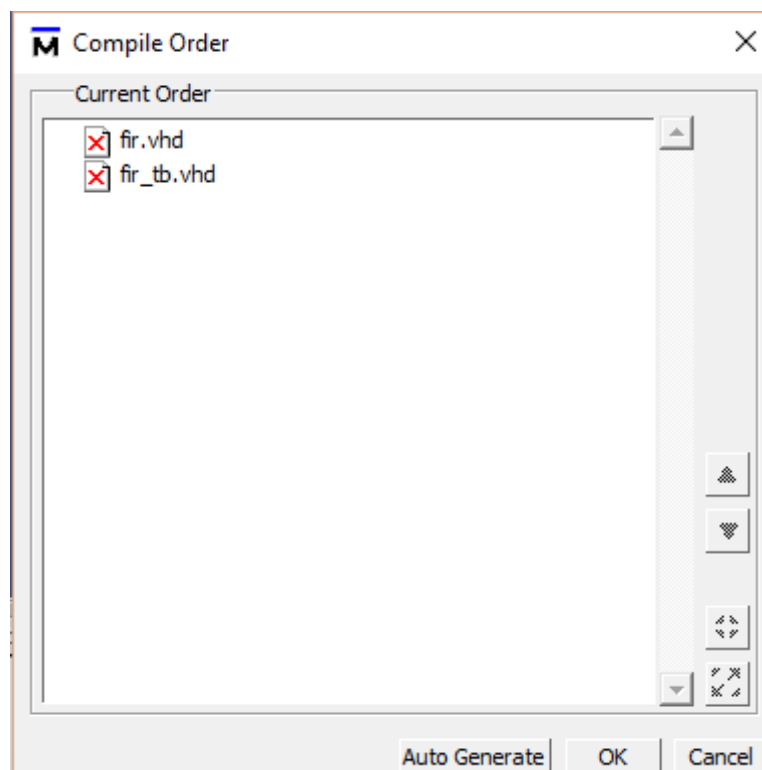


Em seguida localize os arquivos do projeto clicando em **Browse...** . Selecione o arquivo FIR.VHD e FIR\_TB.VHD e clique em **OK**. Neste ponto você pode fechar a janela **Add items to the project** clicando no botão **Close**.

Selecione os dois arquivos na janela da esquerda e clique com o botão direito em **Compile** → **Compile Order...**, como mostra a Figura a seguir:



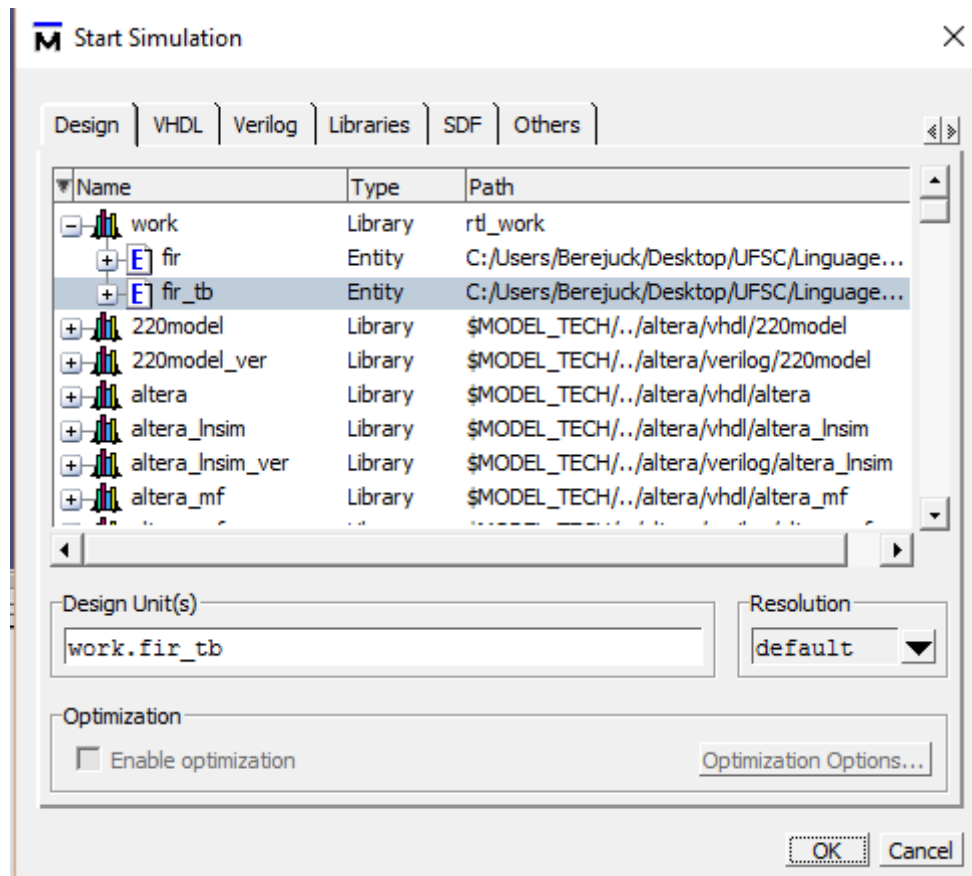
Uma janela chamada **Compile Order** irá abrir. Clique no botão **Auto Generate**:



Deverá ser apresentada uma janela de informação contendo a seguinte afirmação: **All compile devices have been resolved!** Clique em **OK**.

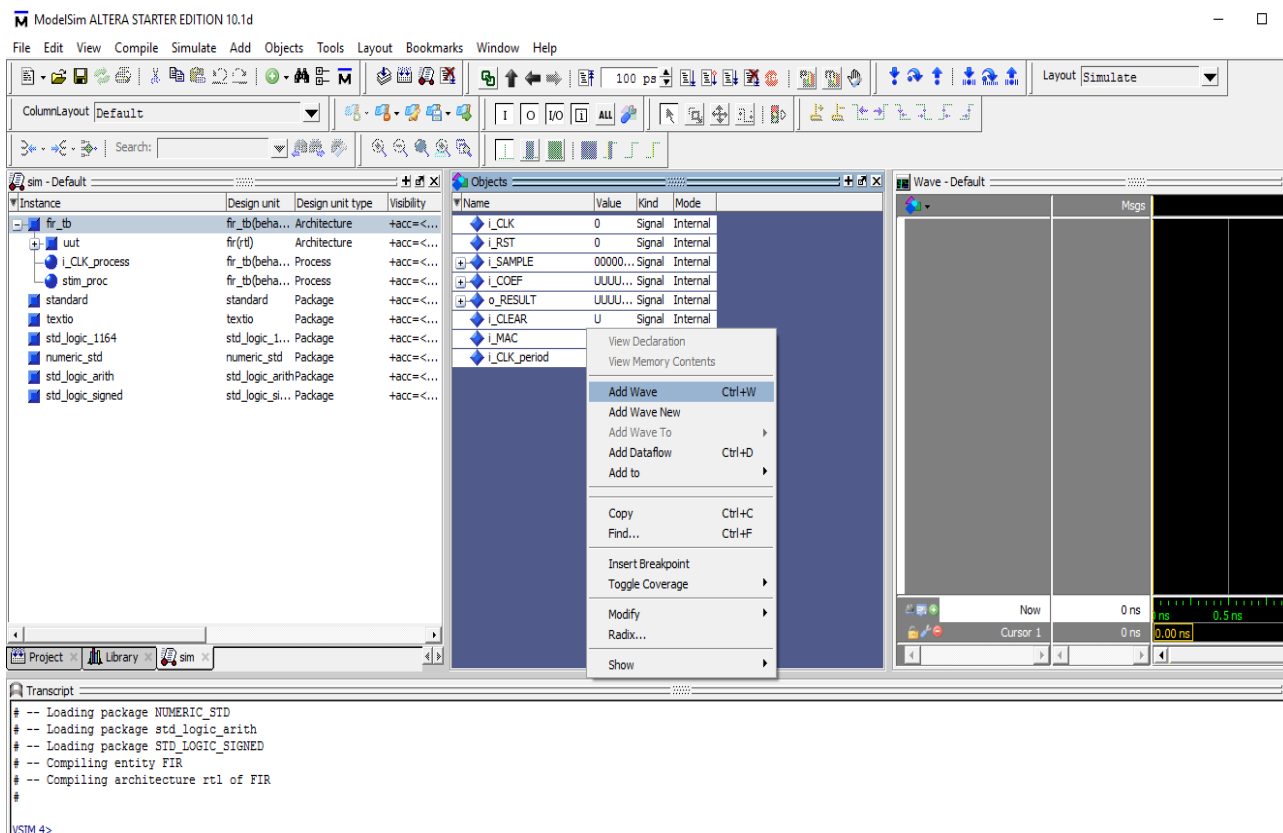
Selecionando os dois arquivos novamente, clique com o botão direito e selecione a opção **Compile** → **Compile All...** e aguarde a indicação de compilação OK sem erros.

Para iniciar a simulação, selecione **Simulation** → **Start Simulation...** no menu principal do ModelSim. Irá aparecer uma janela na qual você deve clicar no sinal + ao lado do diretório **work**, e selecionar o arquivo de testbench **FIR\_TB.vhd**, conforme mostra a figura a seguir. Clique em **OK**.

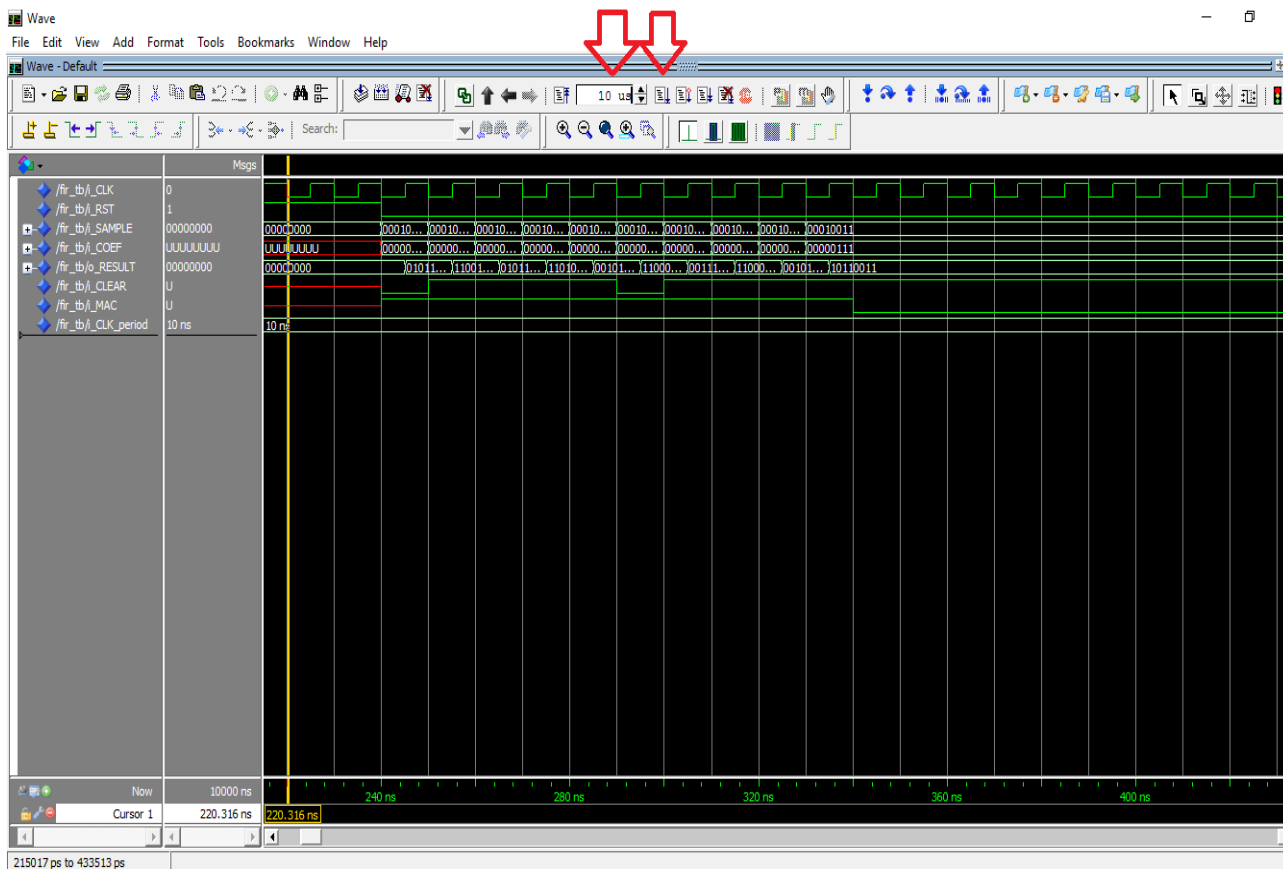


O simulador irá apresentar três janelas. Selecione os sinais mostrados na janela **Object**, clique com o botão da direita sobre eles e selecione **Add Wave...**, conforme mostra a figura a seguir.





Expanda a janela **Wave**, mais a direita na figura anterior. Na parte superior daquela janela, coloque como tempo de simulação o valor **10 us** (10 microsegundos). Em seguida clique no ícone simular, conforme mostram as setas vermelhas na figura a seguir.



## 4 Floorplanning

Sem erros de sintaxe e com uma verificação através de simulação já realizada com o ModelSim, é hora de incluir os pinos de entrada e saída do projeto. A alocação de pinos é uma das ações que podem ser feitas pelo projetista e que está vinculada ao que chamamos de *Floorplanning*. Existem muitas outras possibilidades de configurações, como alocação de componentes em áreas específicas do FPGA, ou restrições de temporizações, etc., mas estão fora do foco deste tutorial. O Dispositivo FPGA escolhido pode ter até 347 pinos de I/O para serem usados. Na janela de tarefas da ferramenta (lado esquerdo) dê um duplo click para selecionar a ferramenta **Pin Planner**:

Pin Planner - C:/Users/Berejuck/Desktop/UFSC/Linguagem de Descrição de Hardware/material/Tutorial/Portas\_Logicas - Portas\_Logicas

File Edit View Processing Tools Window Help

Report not available

Groups Report

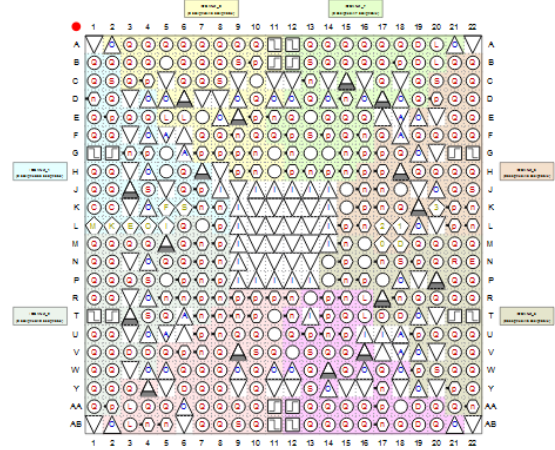
Tasks

- Early Pin
  - Early
  - Run
  - Expo
- Pin Fnde
- Highlight
  - I/O E
  - VREF
  - Frme

Pin Legend

Symbol	Pin Type
○	User I/O
●	User assigned I/O
●	Fitter assigned I/O
○	Unbonded pad
○	Reserved pin
○	Other configuration
○	DEV_OE
○	DEV_CLR
○	DIFF_n
○	DIFF_p
○	DQ
○	DQS
○	CLK_n
○	CLK_p
○	Other PLL

Top View - Wire Bond  
Cyclone III - EP3C16F484C6



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate
i_PINO_A	Input				PIN_E4	2.5 V (default)		8mA (default)	
i_PINO_B	Input				PIN_H6	2.5 V (default)		8mA (default)	
o_PINO_C	Output				PIN_J6	2.5 V (default)		8mA (default)	2 (default)

Clicando no lado direito de cada célula Location é apresentada uma listagem com todos as configurações possíveis para os pinos do FPGA. Neste exemplo foram usados os pinos G11, G13 e G14.

Report not available

Groups Report

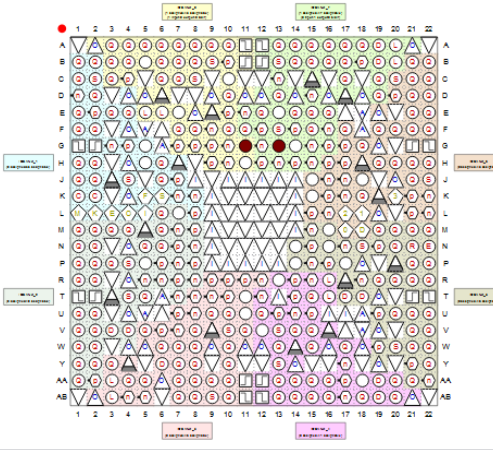
Tasks

- Early Pin
  - Early
  - Run
  - Expo
- Pin Fnde
- Highlight
  - I/O E
  - VREF
  - Frme

Pin Legend

Symbol	Pin Type
○	User I/O
●	User assigned I/O
●	Fitter assigned I/O
○	Unbonded pad
○	Reserved pin
○	Other configuration
○	DEV_OE
○	DEV_CLR
○	DIFF_n
○	DIFF_p
○	DQ
○	DQS
○	CLK_n
○	CLK_p
○	Other PLL

Top View - Wire Bond  
Cyclone III - EP3C16F484C6



Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair
i_PINO_A	Input	PIN_G10	IOBANK_8 Column I/O	DIFFIO_T8n				8mA (default)		
i_PINO_B	Input	PIN_G12	IOBANK_7 Column I/O	DIFFIO_T21n				8mA (default)		
o_PINO_C	Output	PIN_G14	IOBANK_7 Column I/O	DIFFIO_T30n				8mA (default)	2 (default)	
		PIN_G15	IOBANK_7 Column I/O	DIFFIO_T31p						
		PIN_G16	IOBANK_6 Row I/O	DIFFIO_R1p						
		PIN_G17	IOBANK_6 Row I/O	DIFFIO_R5n, PADD23						
		PIN_G21	IOBANK_6 Dedicated Clock	CLK4, DIFFCLK_2p						
		PIN_G22	IOBANK_6 Dedicated Clock	CLK5, DIFFCLK_2n						
		PIN_H1	IOBANK_1 Row I/O	DIFFIO_L11n						

Clique novamente em copilar na barra de tarefas:

