



# Tutorial de uso do Quartus-Prime

Prof. Marcus Vinicius Lamar, v2.2

**OBJETIVO:** Este tutorial visa ensinar o passo a passo de uso básico da ferramenta Quartus Prime v.18.1 para a simulação de circuitos digitais e sua síntese no kit de desenvolvimento FPGA Intel DE1-SoC.

## 1. INTRODUÇÃO

Neste tutorial apresentaremos o software Quartus Prime para a realização de simulações e síntese de circuitos digitais em FPGA. Vários softwares de simulação de circuitos digitais existem gratuitamente na internet, tanto para download (instalar) quanto para execução no navegador (Google it!!!).

O Quartus Prime é a ferramenta de desenvolvimento desenvolvido pela empresa Intel para seus produtos FPGA. Esta ferramenta permite a síntese de circuitos descritos em diagramas esquemáticos ou nas Linguagens de Descrição de Hardware Verilog, System Verilog ou VHDL, para fins de implementação no dispositivo FPGA ou para simulação.

No LINF, use a versão 18.1 do Quartus Prime. (Está instalada também a versão Quartus-II 13.0!)

Instale o Quartus Prime v18.1 (ou a versão mais nova) na sua máquina de casa ou no seu notebook. Faça o download da versão **Lite Edition** a partir do site da Intel :

<http://fpgasoftware.intel.com/18.1/?edition=lite>

Use a aba *Combinated Files* para a instalação de todas as ferramentas!

Combined Files Individual Files Additional Software Updates

**Download and install instructions:** [More](#)  
[Read Intel FPGA Software v18.1 Installation FAQ](#)  
[Quick Start Guide](#)

The **Combined Files** download for the Quartus Prime Design Software includes a number of additional software components. A list of files included in each download can be viewed in the tool tip (i icon) to the right of the description. The **Complete Download** includes all available device families. To achieve a smaller download and installation footprint, you can select device support in the **Multiple File Download** section, or select components in the **Individual Files** and **Additional Software** tabs.

☒ **Quartus Prime Lite Edition**  
☒ **Quartus Prime Lite Edition Software (Device support included)**   
Size: 5.8 GB MD5: 9A217783FCBAED37AE0B7DE11AEA53A8 [Updates Available](#)

**Download Selected Files** [Troubleshoot download problems](#)

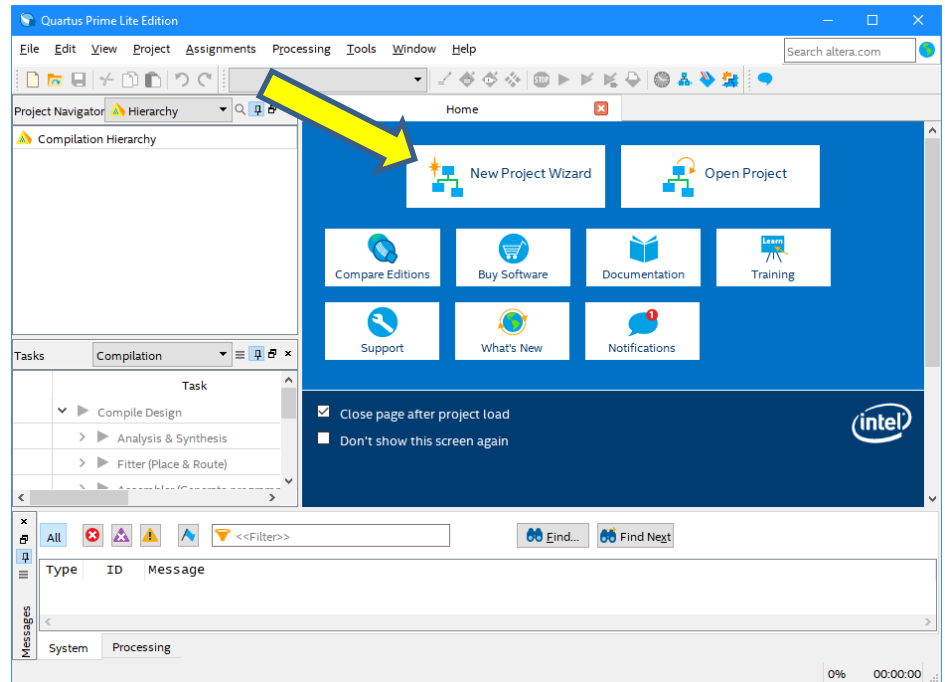
Note: The Quartus Prime software is a full-featured EDA product. Depending on your download speed, download times may be lengthy.



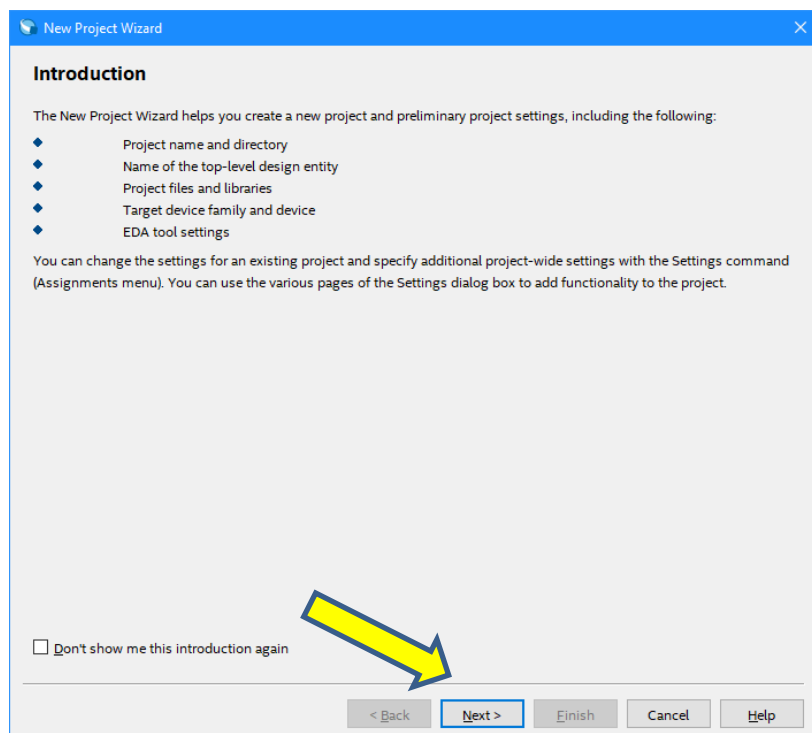
## 2. INICIANDO UM NOVO PROJETO

- Iniciar o programa Quartus (Quartus Prime 18.1) através do Menu Iniciar do Windows, submenu Intel FPGA 18.0.0 Lite Edition.

-Na tela de abertura:  
escolha criar um novo projeto:  
→ clique em New Project Wizard



- Tela: **Introduction:**  
Fornece informações sobre os próximos passos.  
→ clique Next>





- Tela: **Directory, Name and Top-Level Entity**

**Crie um novo diretório no seu Pendrive** onde estará o projeto, ou crie um diretório da área de trabalho.

Defina o nome do Projeto e, automaticamente, o nome do arquivo Top-Level.

→clique Next>

New Project Wizard

**Directory, Name, Top-Level Entity**

What is the working directory for this project?

C:/Users/lamar/Dropbox/kiko/Disiplinas/UnB/Oficina\_Verilog/Exemplo1

What is the name of this project?

exemplo1

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.

exemplo1

Use Existing Project Settings...

< Back Next > Finish Cancel Help

**Importante:** Sempre use nomes que não contenham caracteres especiais !@#%\$`~&()-[]{}.,;<>/?| etc.

- Tela **Project Type:**

Escolha Empty project

→clique Next>

New Project Wizard

**Project Type**

Select the type of project:

☒ Empty project

Create new project by specifying project files and libraries, target device family and device, and EDA tool settings.

☐ Project template

Create a project from an existing design template. You can choose from design templates installed with the Quartus Prime software, or download design templates from the [Design Store](#).

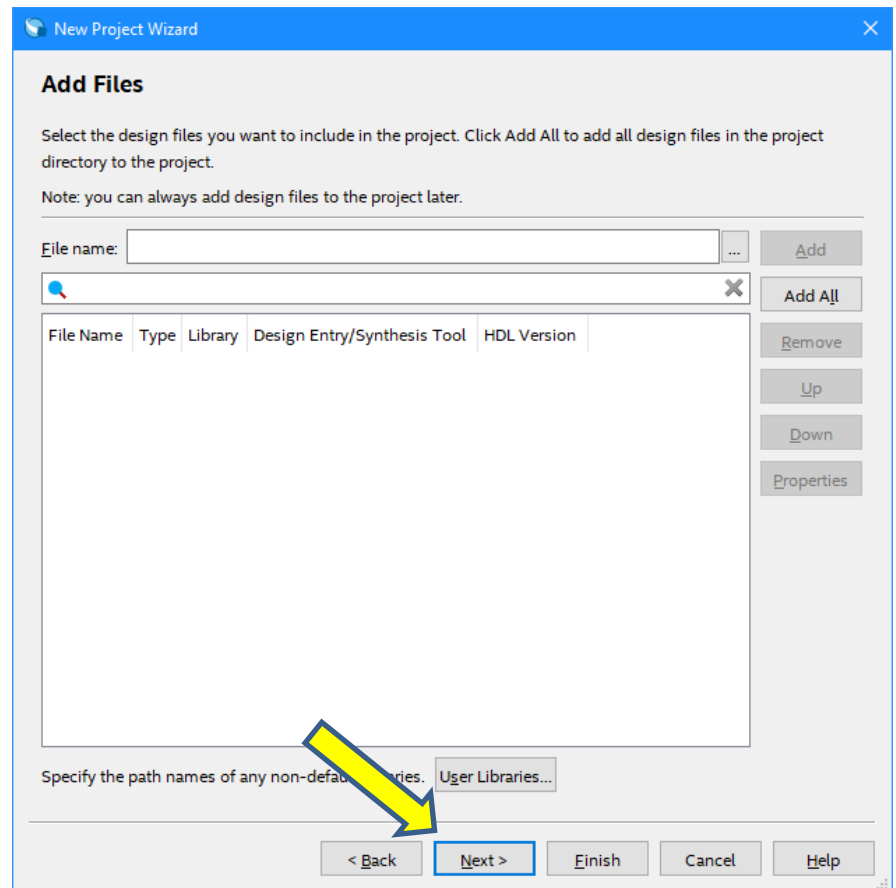
< Back Next > Finish Cancel Help



- Tela: **Add Files:**

Não precisamos neste momento  
adicionar nenhum arquivo.

→clique Next

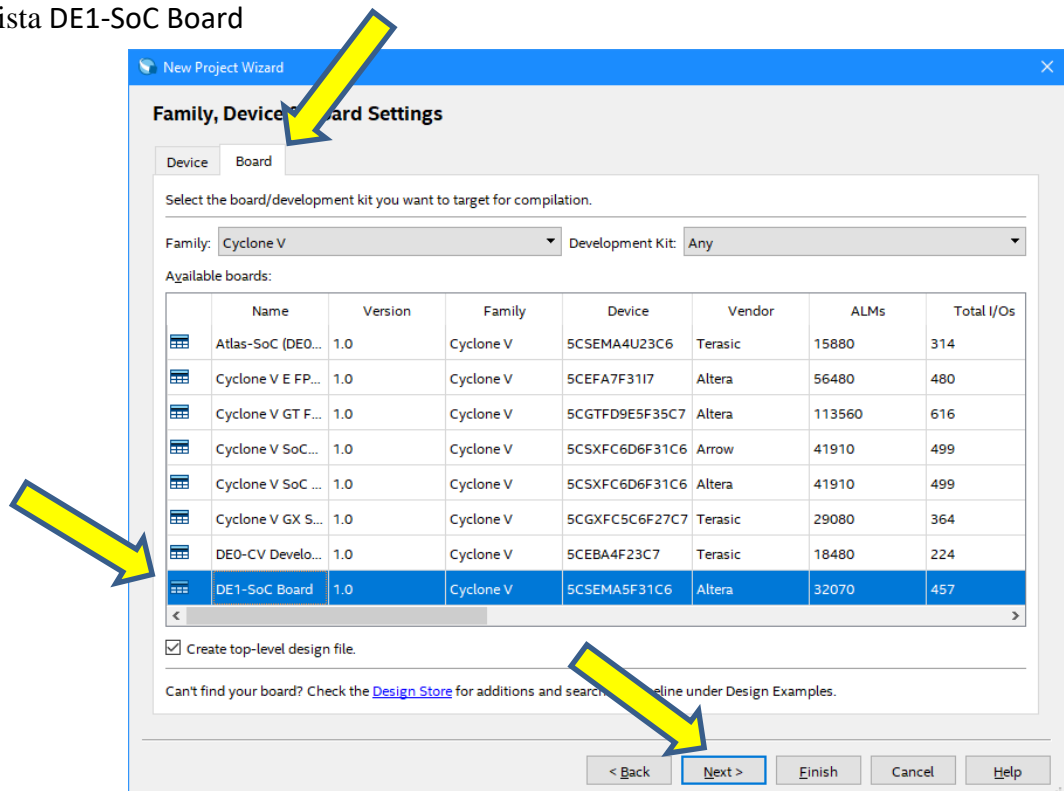


- Tela: **Family, Device & Board Settings:**

→Selecione a aba Board

→Selecione na lista DE1-SoC Board

→clique Next>





-Tela: **EDA Tool Settings:**  
Usaremos as ferramentas padrões.  
→clique Next>

**New Project Wizard**

**EDA Tool Settings**

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

EDA tools:

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

< Back **Next >** Finish Cancel Help

-Tela: **Summary:**  
→clique Finish

**New Project Wizard**

**Summary**

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

Project directory: C:/Users/lamar/Dropbox/kiko/Disciplinas/UnB/Oficina\_Verilog/Exemplo1

Project name: exemplo1

Top-level design entity: exemplo1

Number of files added: 0

Number of user libraries added: 0

Device assignments:

Design template: n/a

Family name: Cyclone V

Device: 5CSEMA5F31C6

Board: DE1-SoC Board

EDA tools:

Design entry/synthesis: <None> (<None>)

Simulation: <None> (<None>)

Timing analysis: 0

Operating conditions:

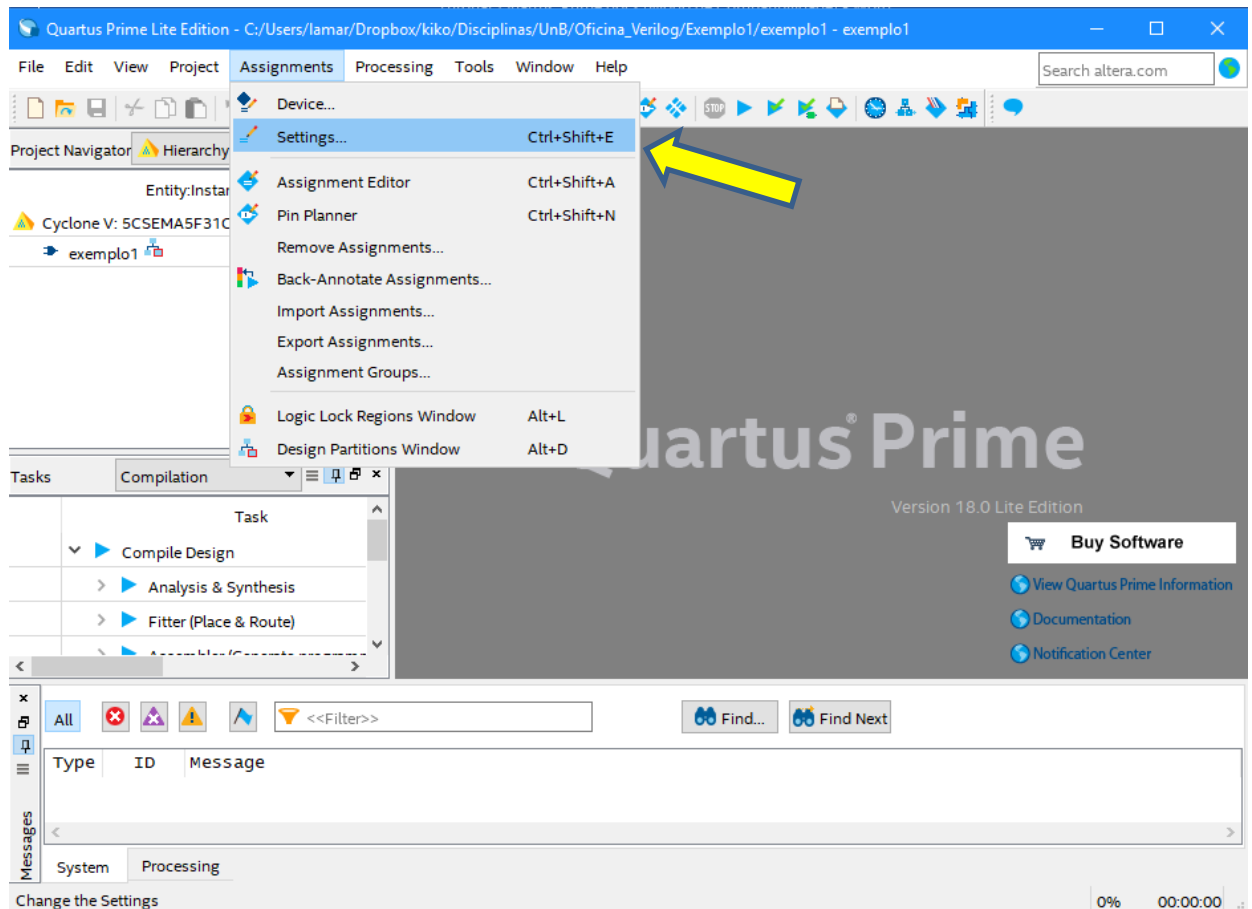
Core voltage: 1.1V

Junction temperature range: 0-85 °C

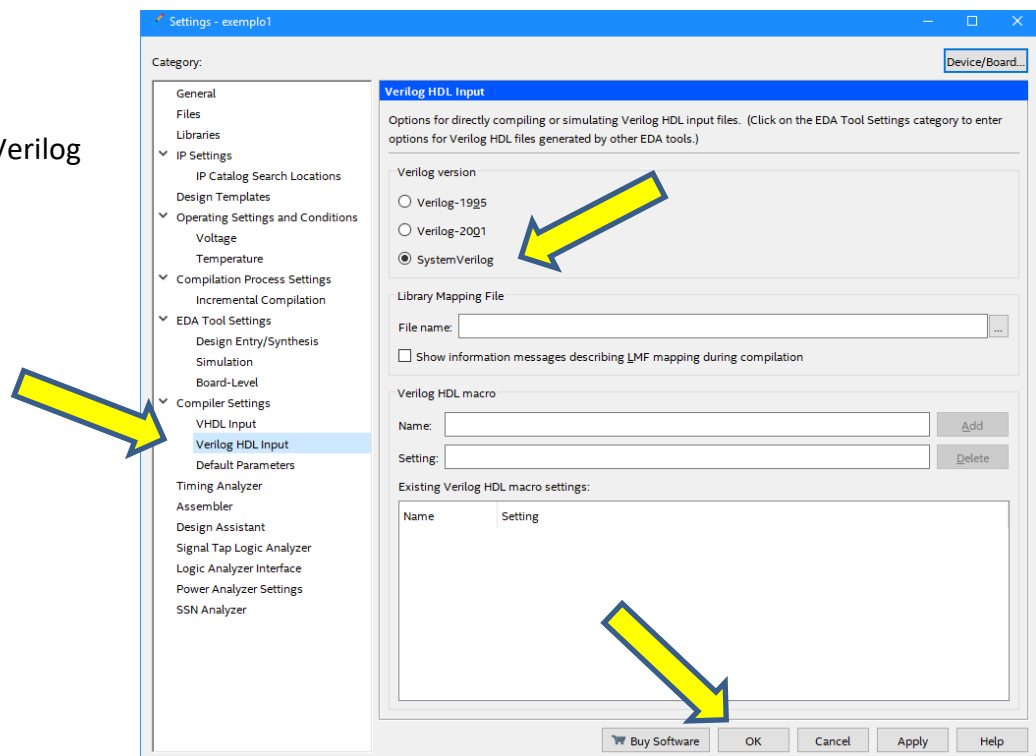
< Back Next > **Finish** Cancel Help



- A seguir configure a HDL a ser usada como SystemVerilog:
- abra o menu Assignments/Settings/Analysis & Synthesis Settings/ Verilog HDL input



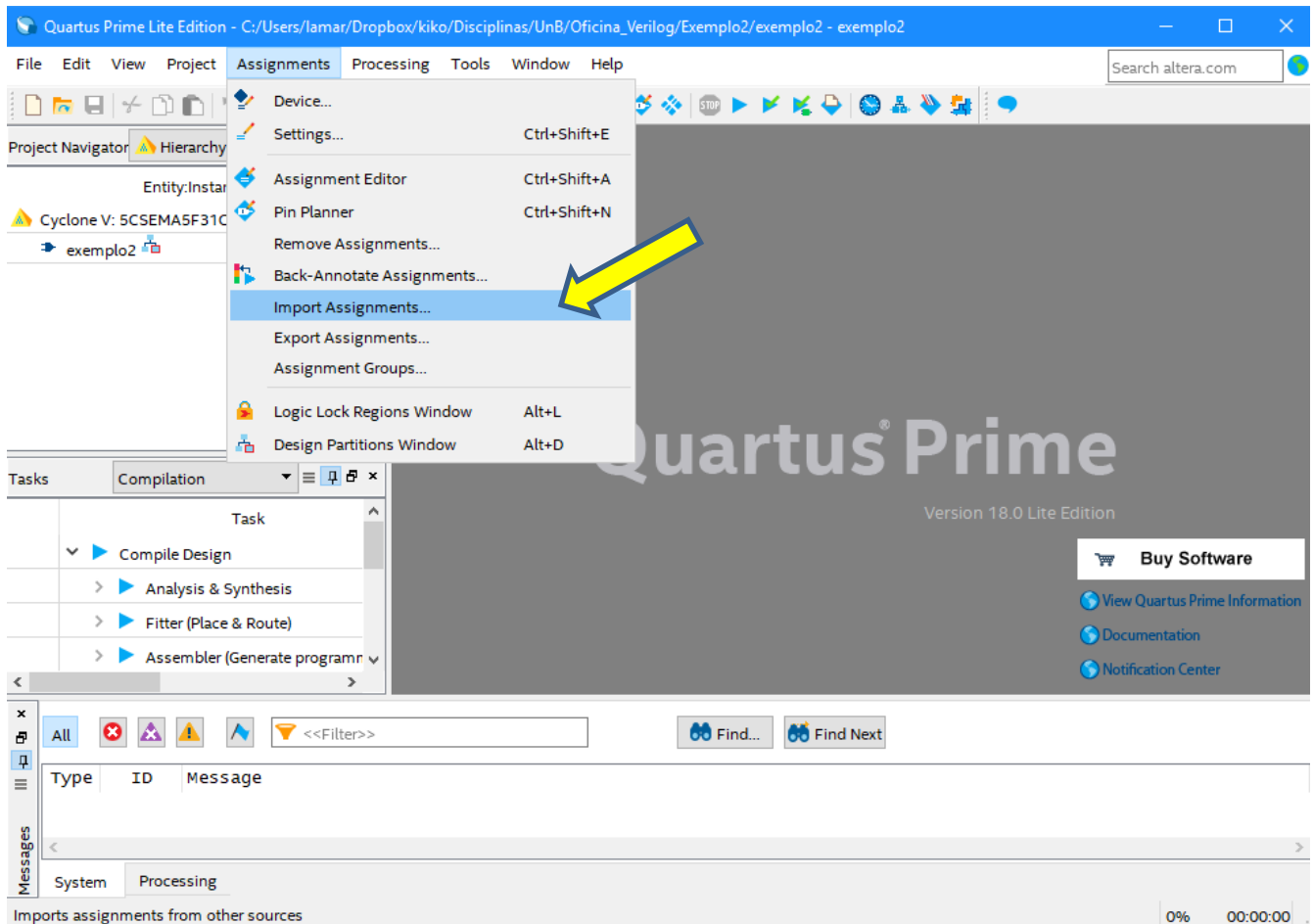
- selecione a linguagem SystemVerilog
- clique OK



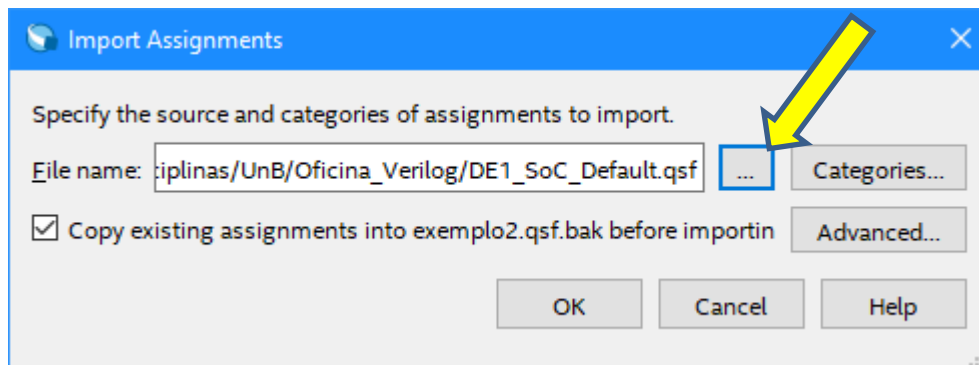


- Configure a localização e os nomes dos pinos do FPGA Cyclone-V

→ clique em Assignments / Import Assignments...



- Faça o download e escolha o arquivo DE1\_SoC\_Default.qsf disponível no Moodle.

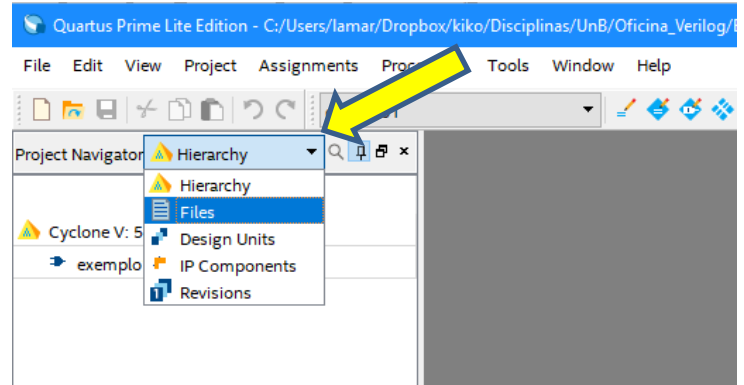



→ clique em OK

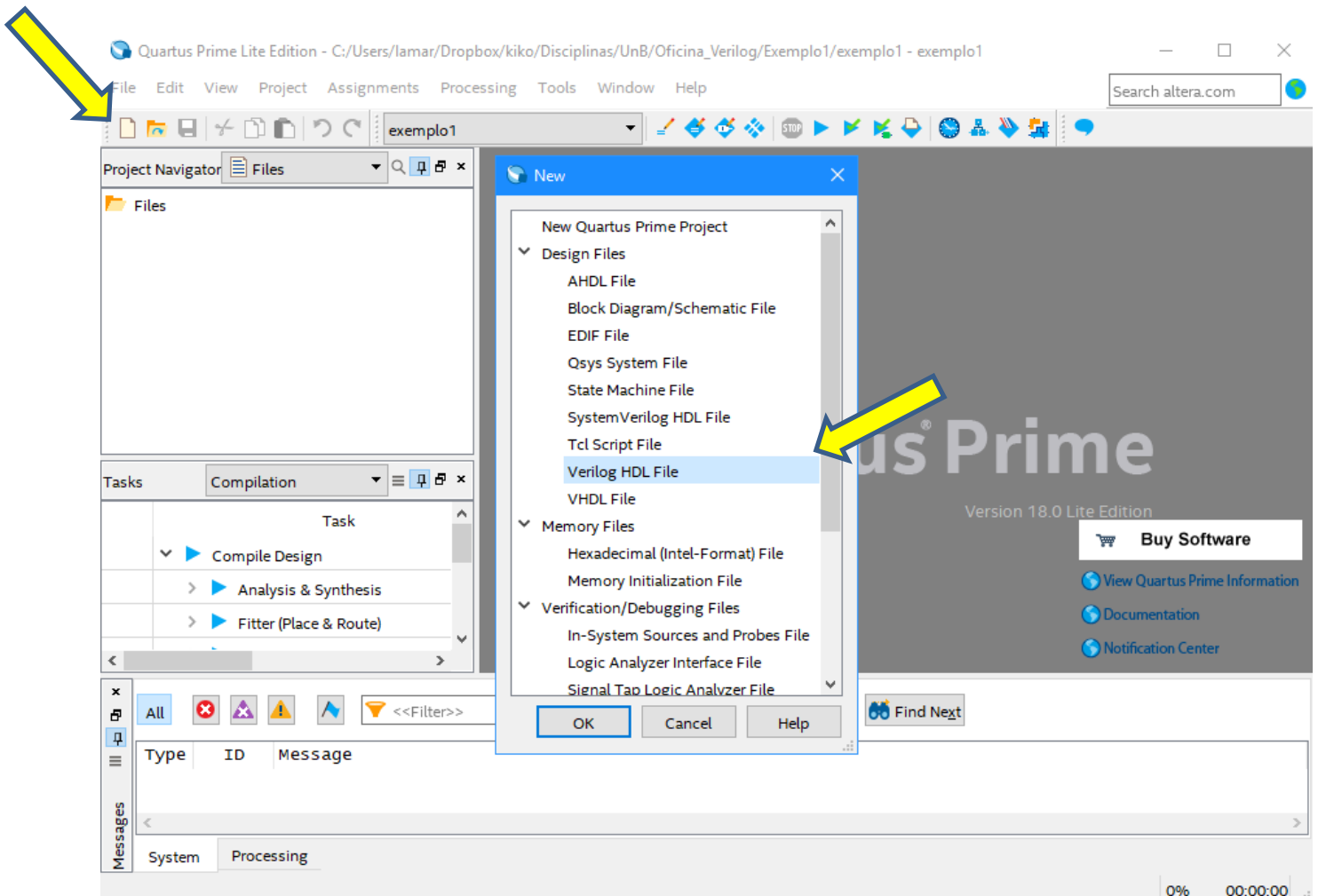


### 3. CRIANDO UMA NOVA DESCRIÇÃO VERILOG

- Inicialmente coloque o Project Navigator no modo Files
- Selecione o modo na janela indicada na figura.



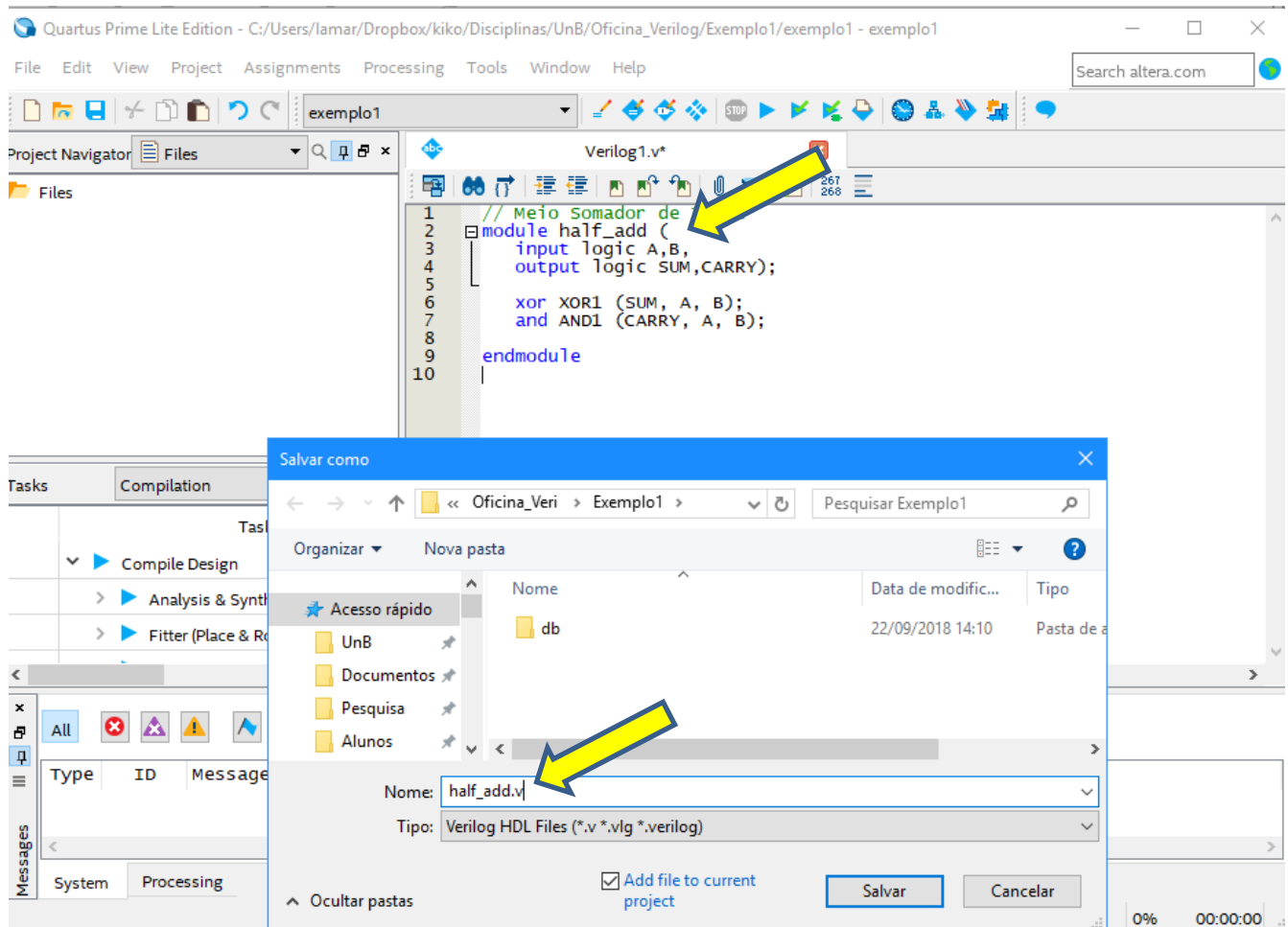
- Crie um novo arquivo Verilog
- clique no ícone new 
- selecione Verilog HDL File



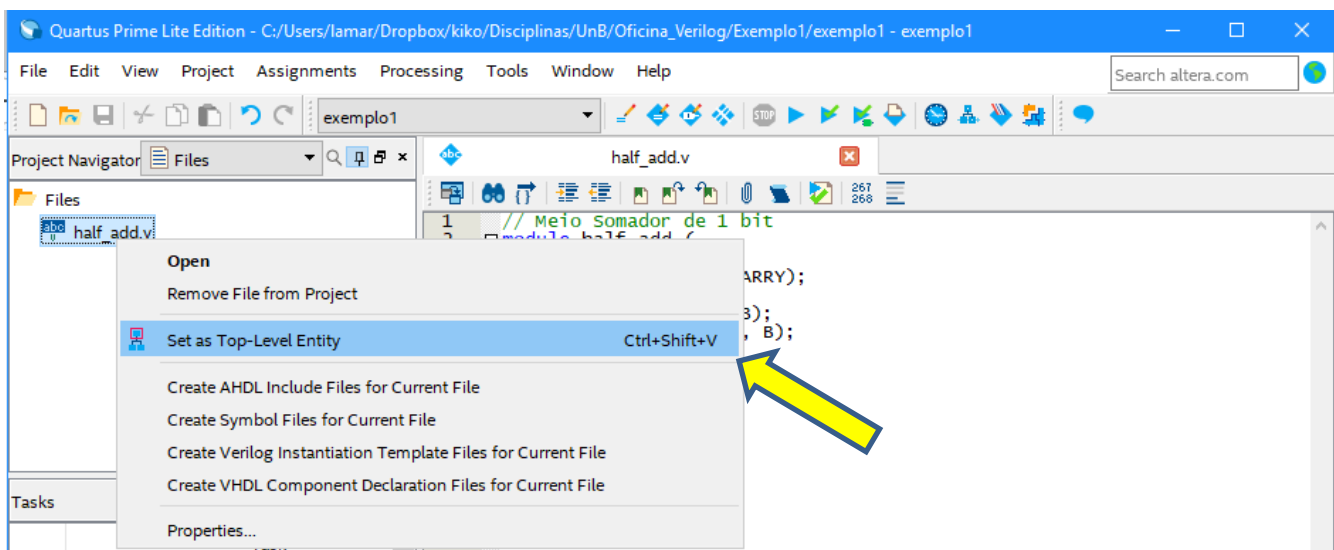




- Um arquivo em branco será criado na área de Edição com o nome Verilog1.v
- Edite o arquivo e salve com o nome do módulo .v



- Para testar apenas este módulo defina-o como o arquivo Top Level!
- Clique com o botão esquerdo do mouse sobre o nome do arquivo no Project Navigator.



#### 4. COMPILANDO O PROJETO

Uma vez salvo o arquivo Verilog com o módulo a ser testado, devemos compila-lo, isto é obter a representação netlist interna do Quartus e os arquivos para programação do chip FPGA a partir da descrição Verilog.

Lembre-se que o arquivo principal do projeto, que será compilado e sintetizado, é definido pelo Top-Level!

→ Inicie a compilação clicando no ícone Start Compilation

- Os passos de Analysis & Synthesis, Fitter (Place & Route), Assembler (Generate programming files) e Time Analysis serão executados e podem ser acompanhados pela janela Tasks.

- A compilação pode demorar de acordo com a complexidade do projeto.

Quartus Prime Lite Edition - C:/Users/lamar/Dropbox/kiko/Disiplinas/UnB/Oficina\_Verilog/Exemplo1/exemplo1

File Edit View Project Assignments Processing Tools Window Help

Project Navigator Files Files half\_add.v

Tasks Compilation Task

- ✓ Compile Design
- ✓ Analysis & Synthesis
- ✓ Fitter (Place & Route)
- ✓ Assembler (Generate programming files)
- ✓ Timing Analysis
- EDA Netlist Writer
- Edit Settings
- Program Device (Open Programmer)

Table of Contents

- Flow Summary
- Flow Settings
- Flow Non-Default Global Settings
- Flow Elapsed Time
- Flow OS Summary
- Flow Log
- Analysis & Synthesis
- Fitter
- Flow Messages
- Flow Suppressed Messages
- Assembler
- Timing Analyzer

Flow Summary

Flow Status Successful - Sat Sep 22 15:33:05 2018

Quartus Prime Version 18.0.0 Build 614 04/24/2018 SJ Lite Edition

Revision Name exemplo1

Top-level Entity Name half\_add

Family Cyclone V

Device 5CSEMA5F31C6

Timing Models Final

Logic utilization (in ALMs) 2 / 32,070 (< 1 %)

Total registers 0

Total pins 4 / 457 (< 1 %)

Total virtual pins 0

Total block memory bits 0 / 4,065,280 (0 %)

Total DSP Blocks 0 / 87 (0 %)

Total HSSI RX PCSs 0

Total HSSI PMA RX Deserializers 0

Total HSSI TX PCSs 0

Messages

Type	ID	Message
Warning	332102	Design is not fully constrained for setup requirements
Warning	332102	Design is not fully constrained for hold requirements
Info		Quartus Prime Timing Analyzer was successful. 0 errors, 6 warnings
Info	293000	Quartus Prime Full Compilation was successful. 0 errors, 14 warnings

System (1) Processing (125)

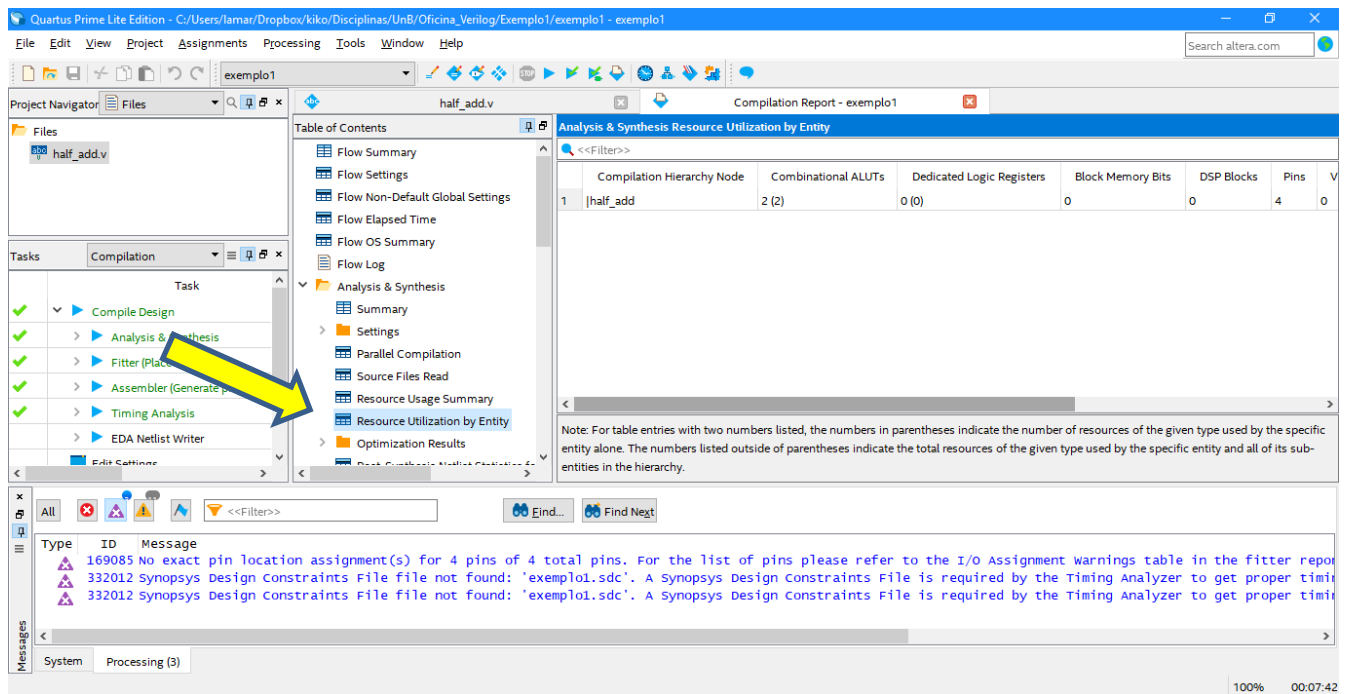
100% 00:07:42

- Corrija os **erros** encontrados e “passe os olhos” pelos **Critical Warnings** e **Warnings** , muitas vezes possuem pistas sobre algum possível bug que esteja ocorrendo.

## - Análise dos requerimentos do Projeto

- Os **requerimentos físicos** do projeto (número de Elementos Lógicos (ALMs), registradores, quantidade de bits de memória e número de multiplicadores (DSP) ) podem ser obtidos :

- De maneira geral pela própria tela de Flow Summary resultante da compilação (figura anterior)
- E em detalhes visualizando o relatório Analysis & Synthesis / Resource Utilization by Entity

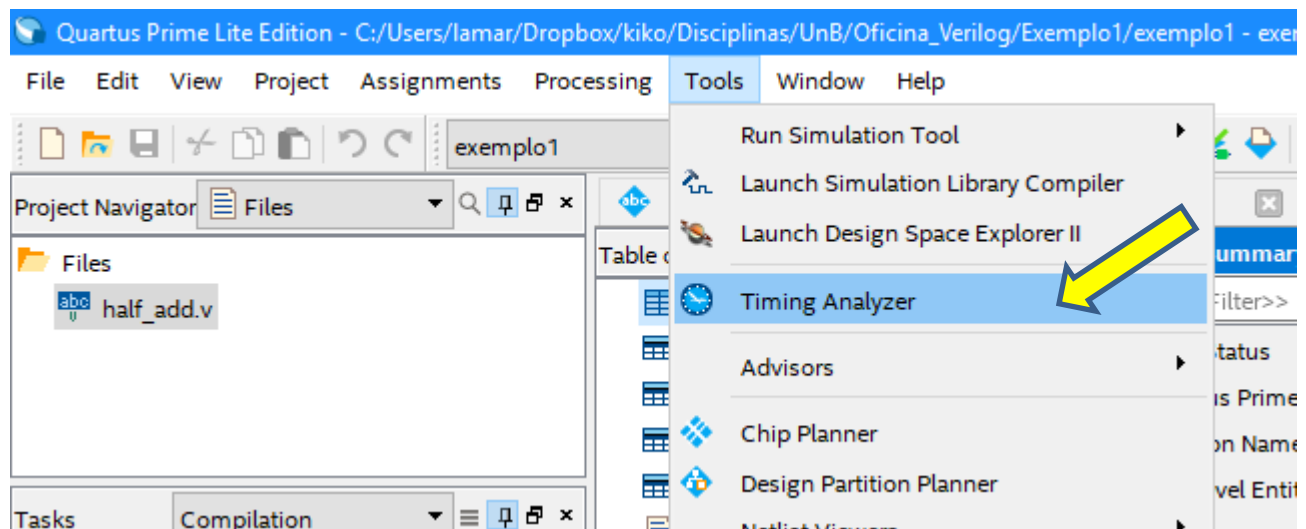


The screenshot shows the Quartus Prime Lite Edition interface. The 'Table of Contents' pane on the left has a yellow arrow pointing to 'Resource Utilization by Entity' under the 'Analysis & Synthesis' section. The main window displays the 'Analysis & Synthesis Resource Utilization by Entity' report for the 'half\_add.v' entity.

Compilation Hierarchy Node	Combinational ALUTs	Dedicated Logic Registers	Block Memory Bits	DSP Blocks	Pins	V
1  half_add	2 (2)	0 (0)	0	0	4	0

Note: For table entries with two numbers listed, the numbers in parentheses indicate the number of resources of the given type used by the specific entity alone. The numbers listed outside of parentheses indicate the total resources of the given type used by the specific entity and all of its sub-entities in the hierarchy.

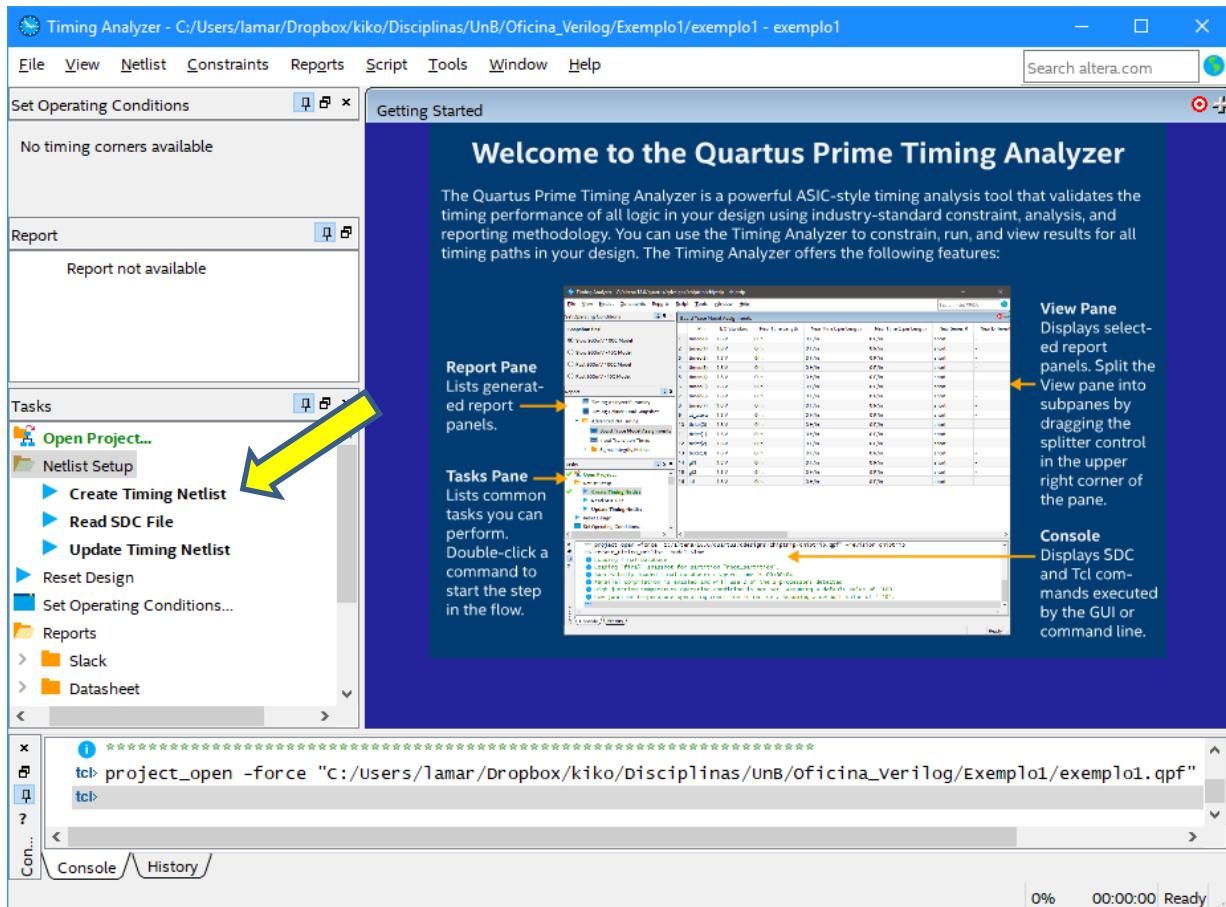
-Os **requerimentos temporais** tpd, tsu, tco, th **devem** ser obtidos através da execução da ferramenta Timing Analyzer.



The screenshot shows the Quartus Prime Lite Edition interface with the 'Tools' menu open. A yellow arrow points to the 'Timing Analyzer' option in the menu.



- Para usar o Timing Analyzer
- clique em Tool/Time Analyzer

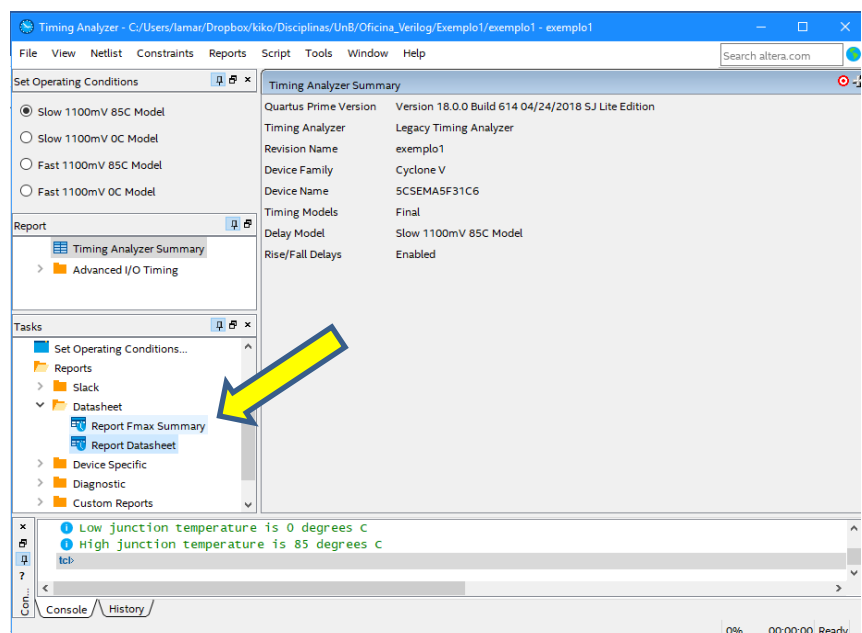


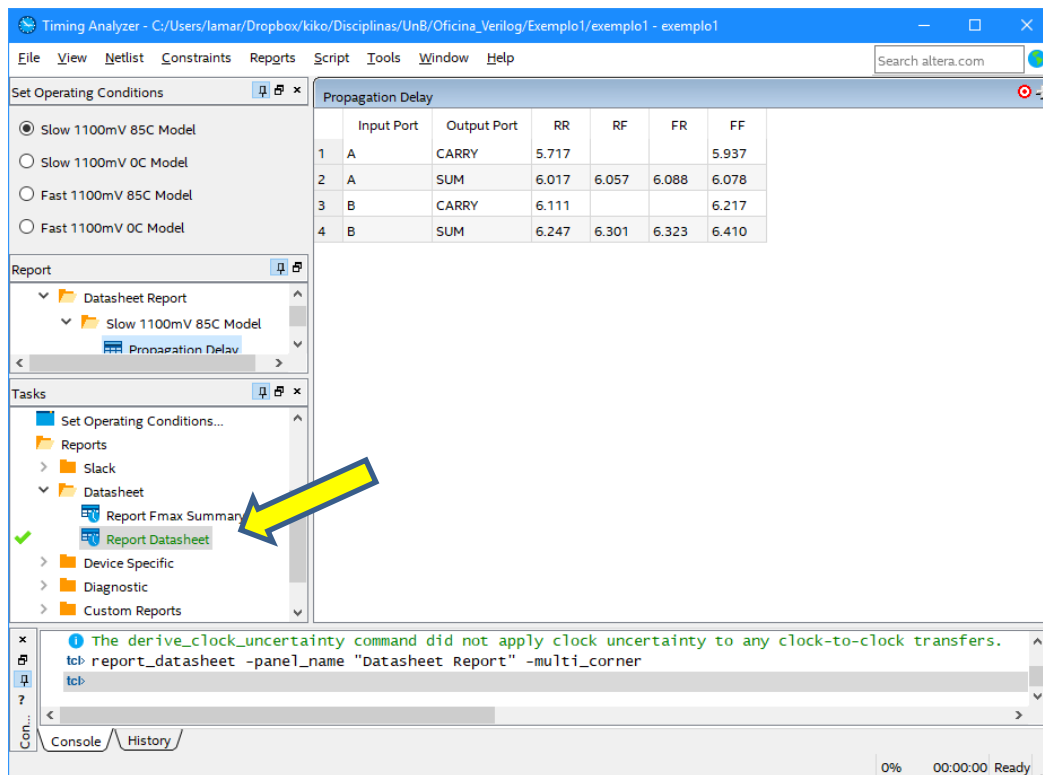
A ferramenta se baseia nas tarefas (Tasks) e geração de relatórios (Reports).

→ Duplo clique em Create Timing Netlists – para a geração das netlists

→ clique em Reports/Datasheet e escolha os relatórios

- Fmax Summary
- e/ou
- Datasheet

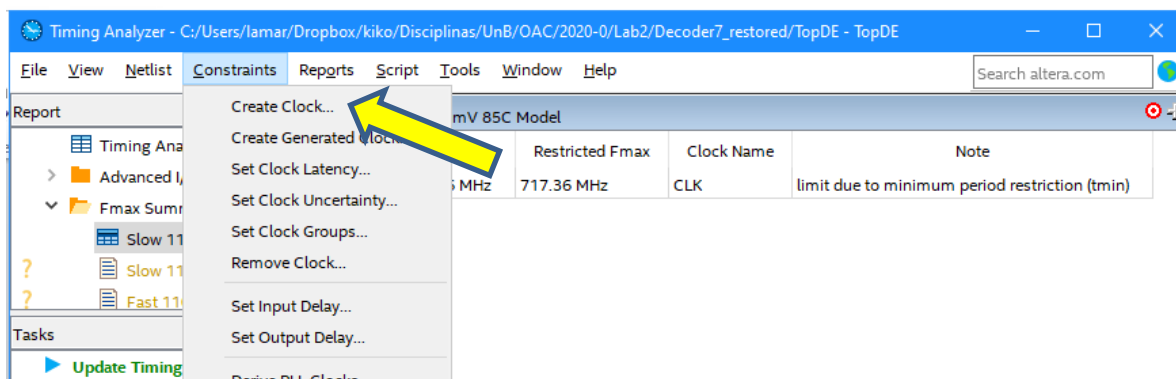




- No exemplo acima, clicando no Report Datasheet, obtém-se os tempos de propagação de um circuito combinacional para as variações da entrada Rise to Rise (RR), Rise to Fall(RF), Fall to Rise(FR) e Fall to Fall(FF). O tpd (propagation delay time) é definido como o **maior** desses tempos.

- Caso seu projeto seja **síncrono**, isto é, use um sinal de clock (circuito sequencial), é necessário definir a frequência utilizada no projeto. Para tanto:

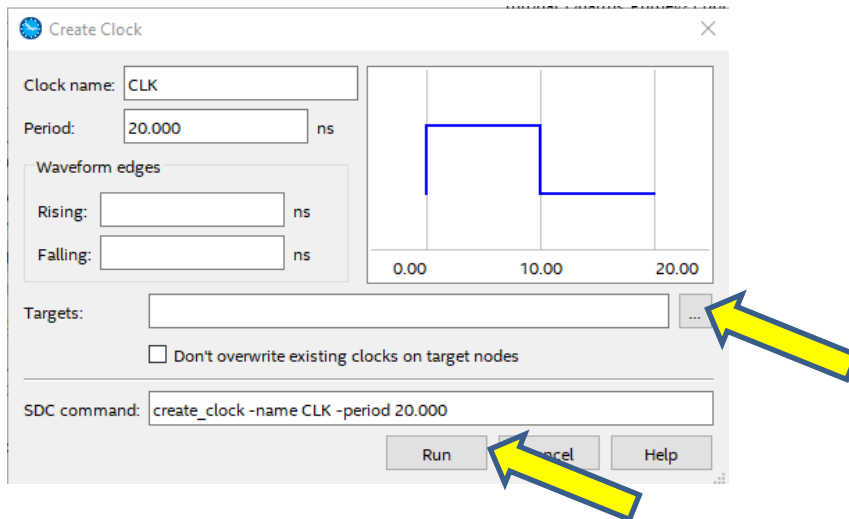
→ Clique em Constraints/Create Clock...



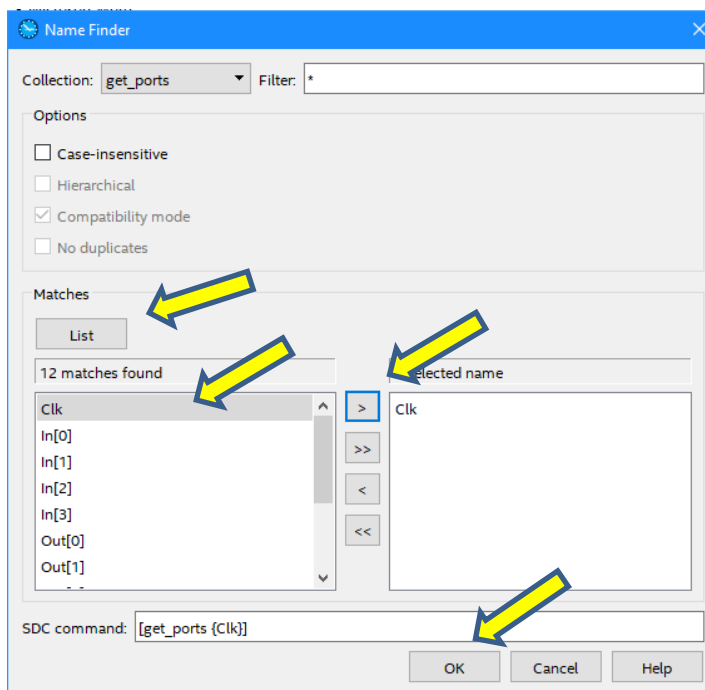
- Irá abrir a janela de definição do sinal de clock.



- Defina o Clock name: CLOCK ou CLK ou qualquer outro genérico (não utilizado no seu projeto)
- Defina o período como 20.000 ns para uma frequência de 50MHz



- Clique no ícone [...] em Targets:
- Clique em List
- Clique no nome do pino de clock do seu circuito
- Clique em > e em OK



- Clique em RUN
- Clique em  Update Timing Netlist



Clicando no Report Fmax Summary, se obtém a frequência máxima utilizável considerando os tempos de atrasos dos elementos.

Clicando em Report Datasheet, se obtém os tempos tsu (setup time), th (hold time) e tco (clock to output time). O th e o tco devem ser sempre positivos, o tsu deve ser sempre negativo para que o circuito esteja OK em termos de temporização, uma vez que é adotada a borda de subida do clock como referência para as medidas desses tempos.

Timing Analyzer - C:/Users/lamar/Dropbox/kiko/Disiplinas/UnB/OAC/2020-0/Lab2/Decoder7\_restored/TopDE - To...

File View Netlist Constraints Reports Script Tools Window Help

Search altera.com

Report

- Slow 1100mV 85C Model
  - Setup Times
  - Hold Times
  - Clock to Output Times
  - Minimum Clock to Output

Tasks

- Read SDC File
- Update Timing Netlist
- Reset Design
- Set Operating Conditions...
- Reports
  - Slack
  - Datasheet
    - Report Fmax Summary
    - Report Datasheet

	Data Port	Clock Port	Rise	Fall	Clock Edge	Clock Reference
1	In[*]	CLK	-2.826	-2.805	Rise	CLK
1	In[0]	CLK	-2.826	-2.806	Rise	CLK
2	In[1]	CLK	-2.831	-2.824	Rise	CLK
3	In[2]	CLK	-2.830	-2.805	Rise	CLK
4	In[3]	CLK	-2.831	-2.825	Rise	CLK

Deriving Clock Uncertainty. Please refer to report\_sdc in the Timing Analyzer to see c

```
tc> report_clock_fmax_summary -panel_name "Fmax Summary" -multi_corner
tc> report_datasheet -panel_name "Datasheet Report" -multi_corner
tc>
```

Console History

0% 00:00:00 Ready

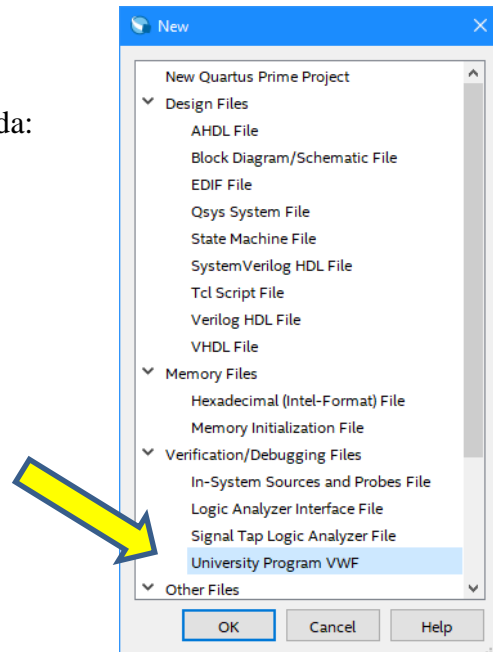


## 5. SIMULANDO O PROJETO

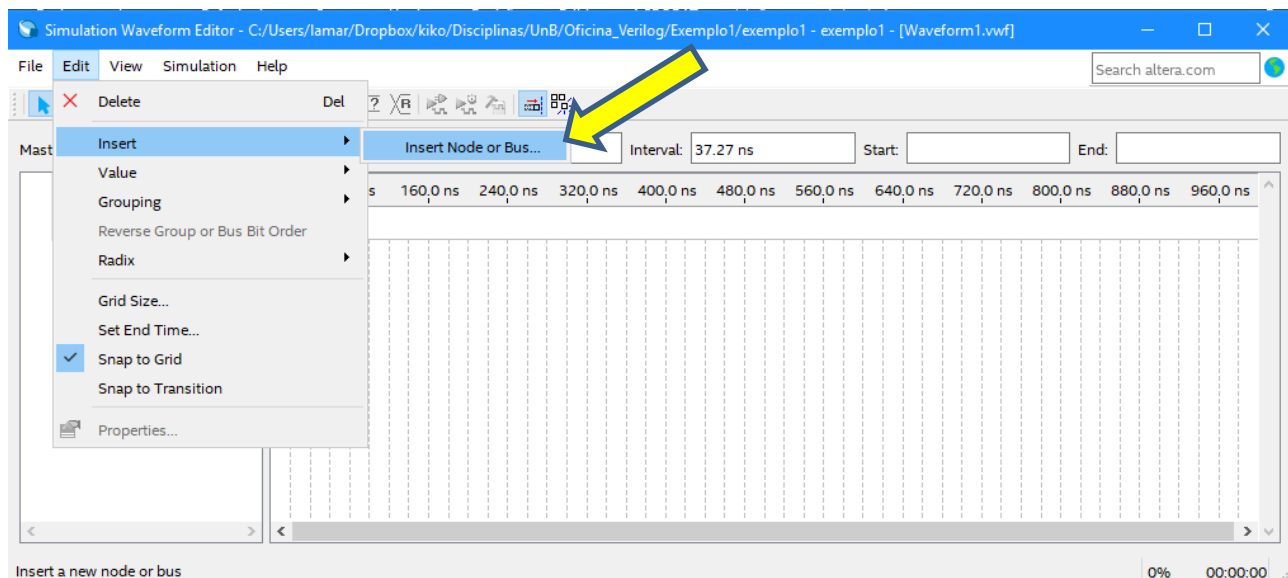
O Quartus Prime realiza apenas a simulação funcional por forma de onda do FPGA da família Cyclone V usada na DE1-SoC.

Caso uma simulação mais complexa seja necessária pode-se executar a ferramenta de simulação Modelsim (outro Tutorial).

- Para a Simulação Funcional por Forma de Onda
- Crie o arquivo com a forma de onda dos sinais de entrada:  
→ clique em File > New > University Program (VWF)



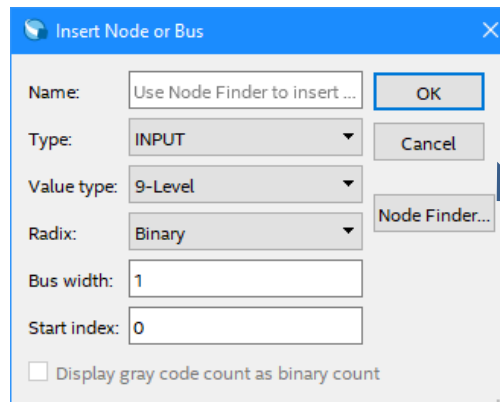
- Defina os pinos a serem considerados na simulação.  
→ Clique em > Insert Node or Bus > Node Finder > Selecione Filter: Pins All > List





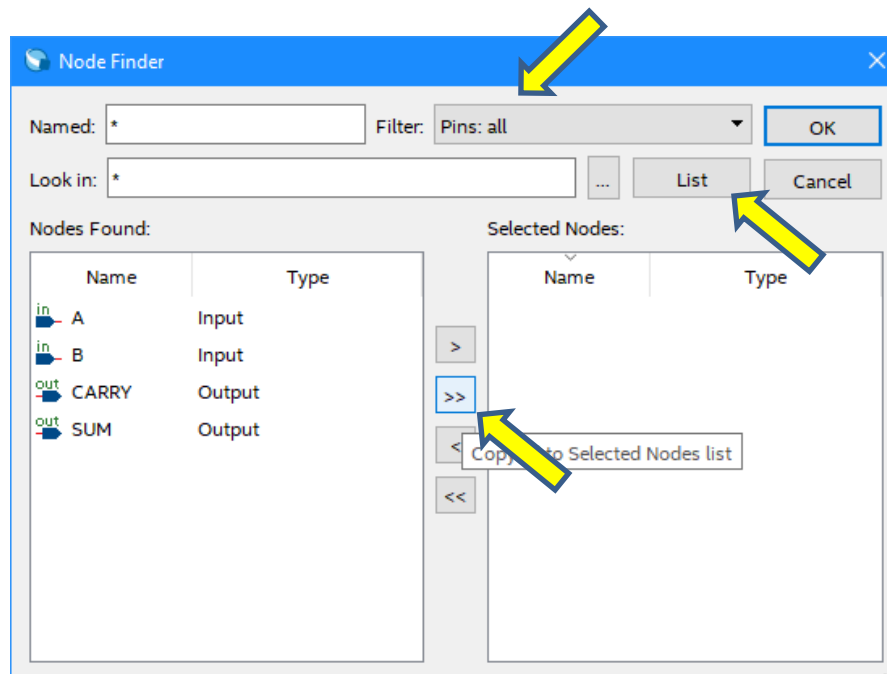


→ Clique em Node Finder



→ Selecione Filter: Pins all e em seguida clique em List

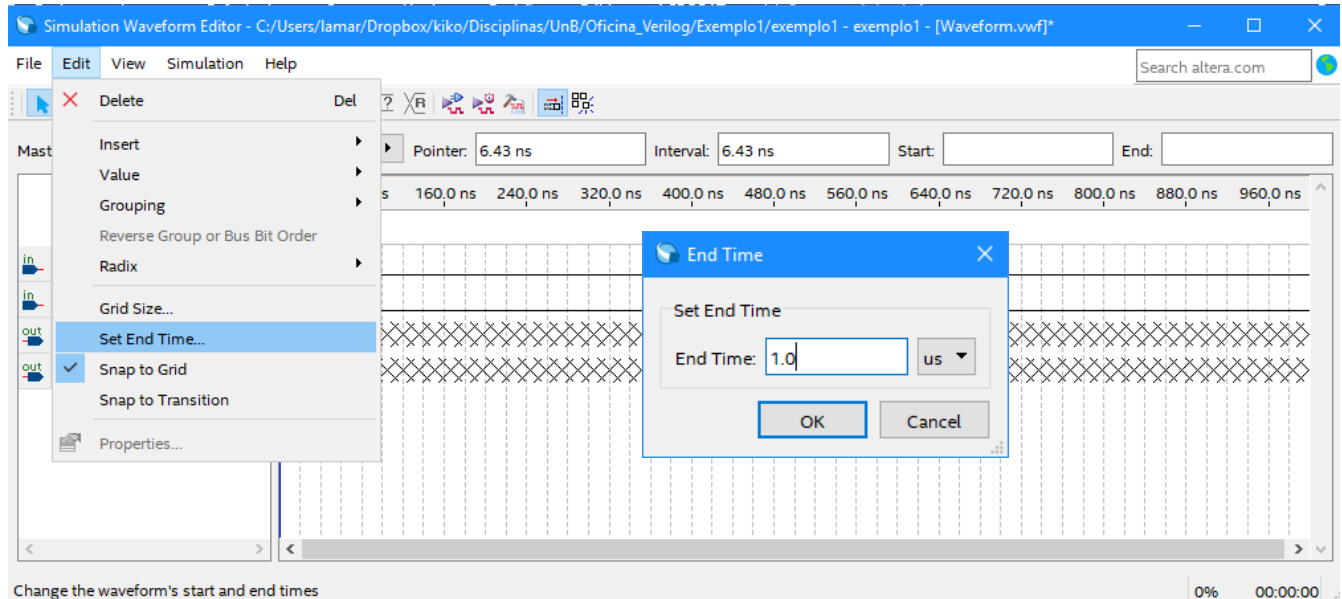
→ Clique em >> para adicionar todos os nós encontrados.





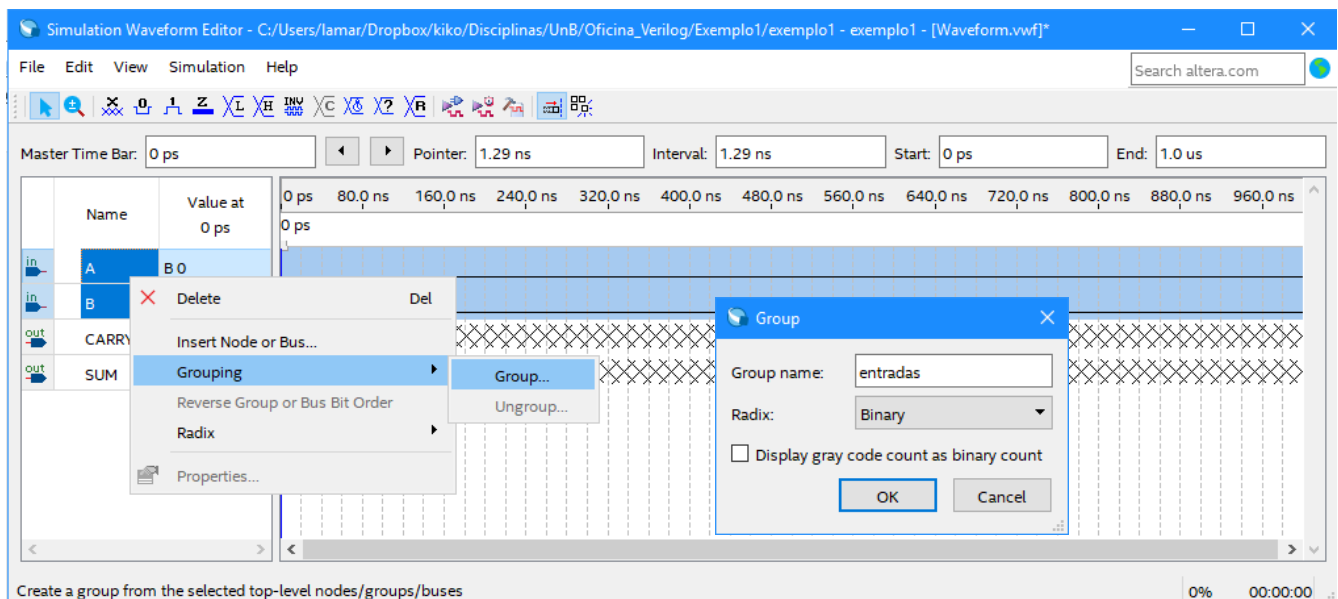
- Defina o tempo total de simulação (de 1 $\mu$ s a, no máximo, 100 $\mu$ s) e o período de contagem de acordo com o experimento e tamanho da tabela verdade a ser verificada.

→ clique em Edit/Set End Time... e defina o tempo total da simulação. Default é 1 $\mu$ s.



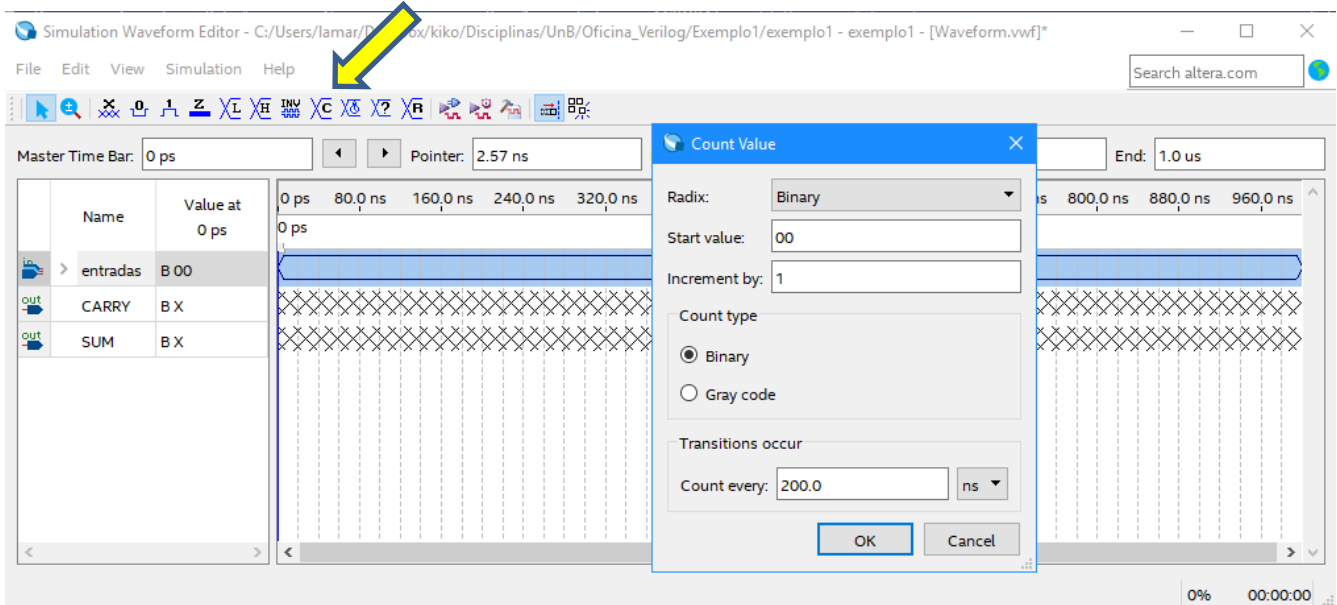
→ Selecione e agrupe os sinais de entrada Grouping/Group...

→ Defina como nome 'entradas' (ou qualquer outro)





→ clique em Count Value ícone



-Caso o Start value apareça como 00, 00, apague a vírgula e o espaço, passando então a ser 0000

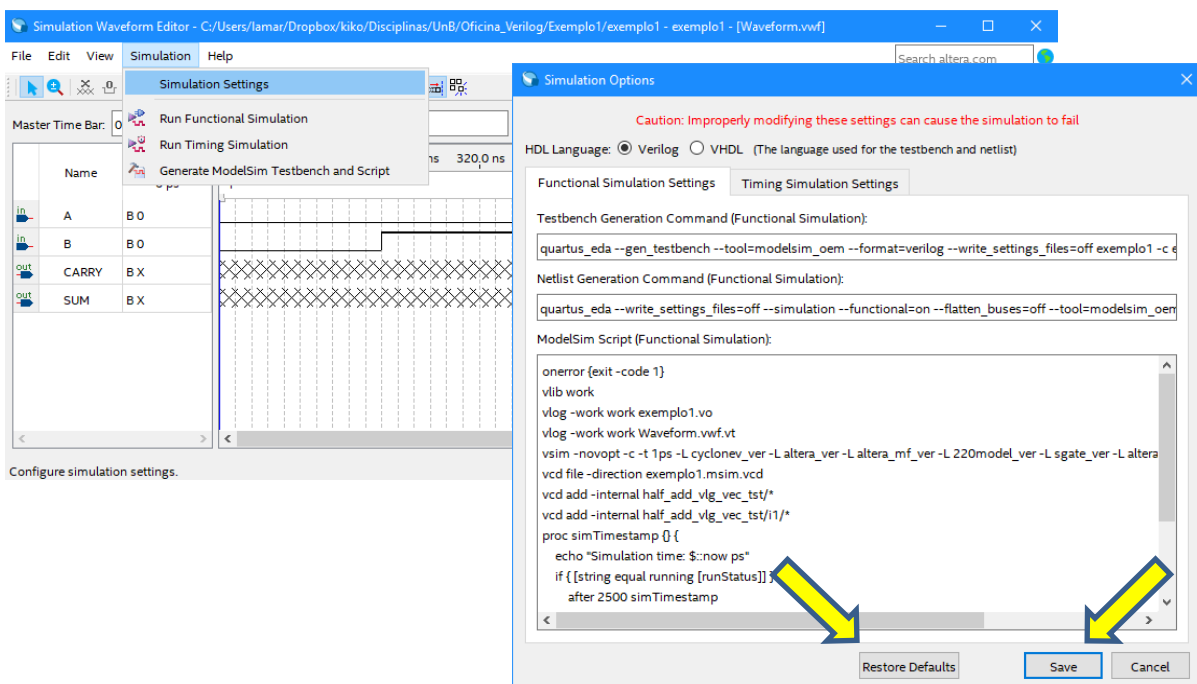
→ Defina o período de cada contagem em Count every. Ex.: 200.0 ns. (O default é 10ns)

→ Desagrupe (Ungroup) a Entrada de modo a voltar ao modo normal

→ Salve como arquivo .vwf

- Defina o Simulation Setup para garantir que o arquivo correto que será simulado.

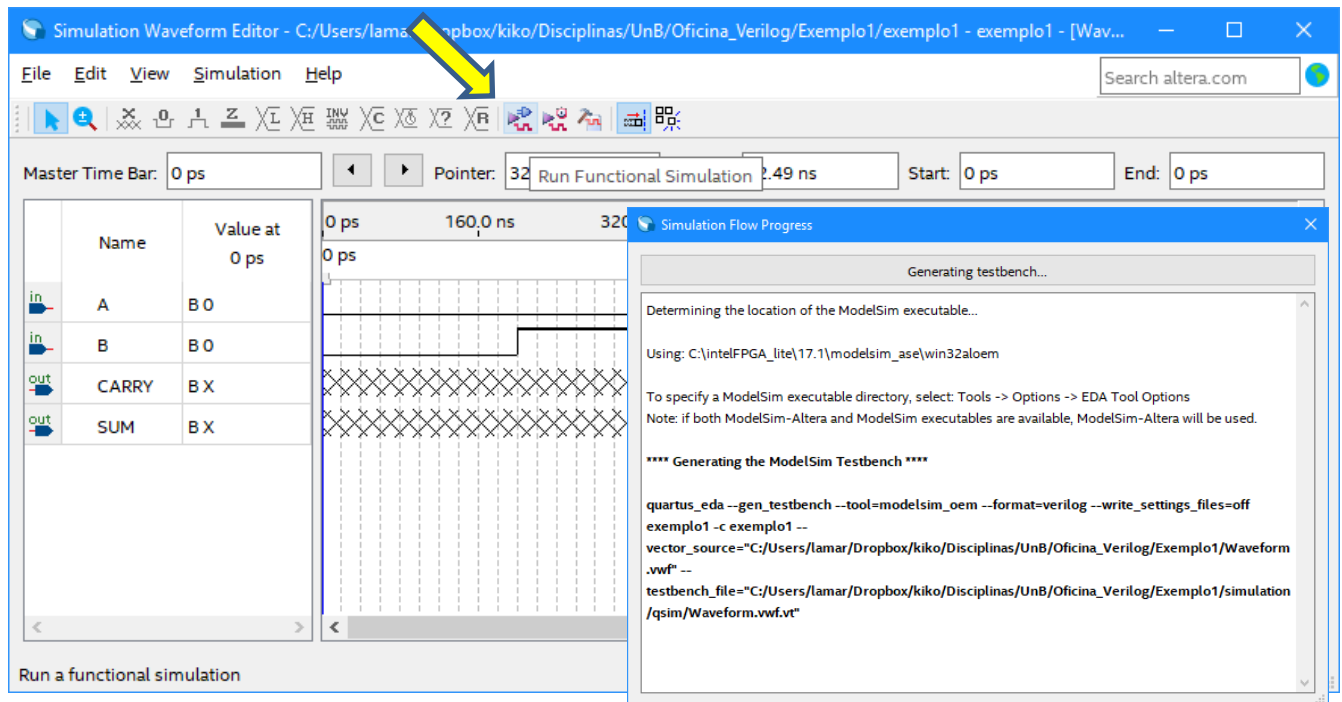
→ clique em Restore Defaults e em seguida em Save



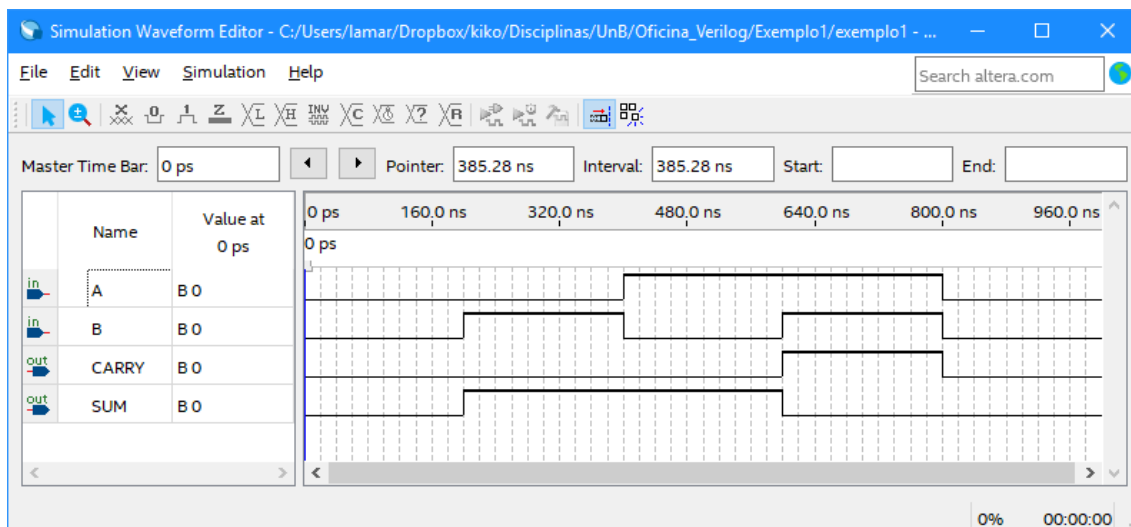


- Execute a Simulação Funcional, onde todas as portas e elementos lógicos serão considerados ideais (sem atraso de propagação) para verificar a corretude do seu projeto.

→ clique no ícone Run Functional Simulation:



Uma nova janela do Simulador por forma de onda irá abrir com o resultado da simulação.

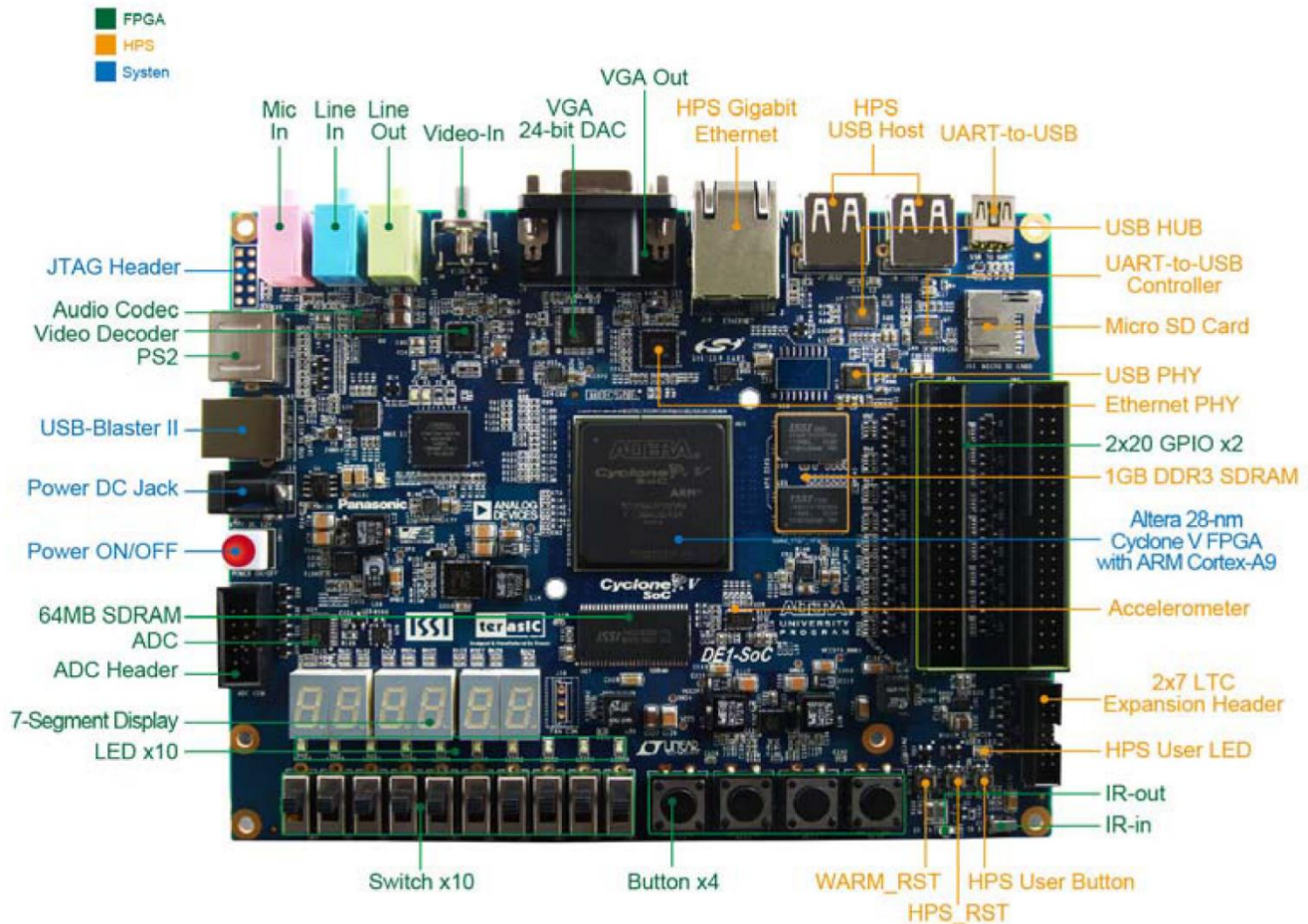


Obs.: Caso o Quartus indique um erro que não encontrou o simulador, defina manualmente a sua localização. Para tanto: Abra o menu Tools/Options/EDA Tools Options e no campo ModelSim-Altera defina a path C:\intelFPGA\_lite\18.0\modelsim\_ase\win32aloem

## 6. A PLATAFORMA DE DESENVOLVIMENTO INTEL DE1-SOC2

Um chip FPGA (*Field Programmable Gate Array*) é um circuito eletrônico integrado capaz de sintetizar qualquer sistema digital através das interconexões de seus componentes elementares (Elementos Lógicos).

Nos experimentos, iremos utilizar kits de desenvolvimento FPGA DE1-SoC da Intel.



**Kit de desenvolvimento Intel DE1-SoC**

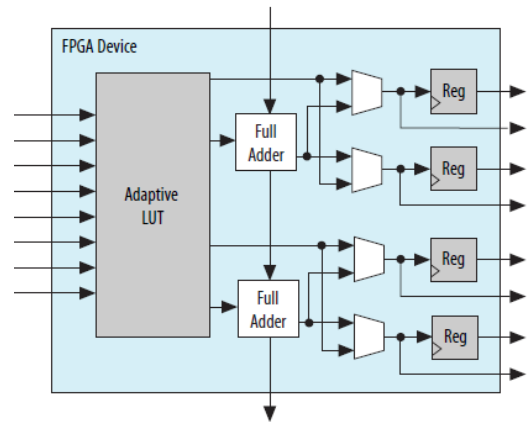
O kit de desenvolvimento possui além do chip FPGA, um processador ARM dual-core A9 e diversas interfaces, tais como Chaves, Botões, LEDs, Displays, Interface VGA, PS2, Áudio, acelerômetro, conversores Analógico-Digital, interface IR, entrada de vídeo e GPIOs.



O kit DE1-SoC possui um chip FPGA da família Cyclone V modelo 5CSEMA5F31C6, que possui 896 pinos e contém 32.070 Elementos Lógicos (ALM – *Adaptive Logic Module*).

Cada Elemento Lógico (ALM), mostrado na figura ao lado, é capaz de implementar uma função lógica booleana de 8 entradas e 4 saídas, possui 4 registradores programáveis (*Flip-Flop*), são encadeáveis (*carry*) e podem ser usados para roteamento (ligação de um ponto a outro no chip FPGA).

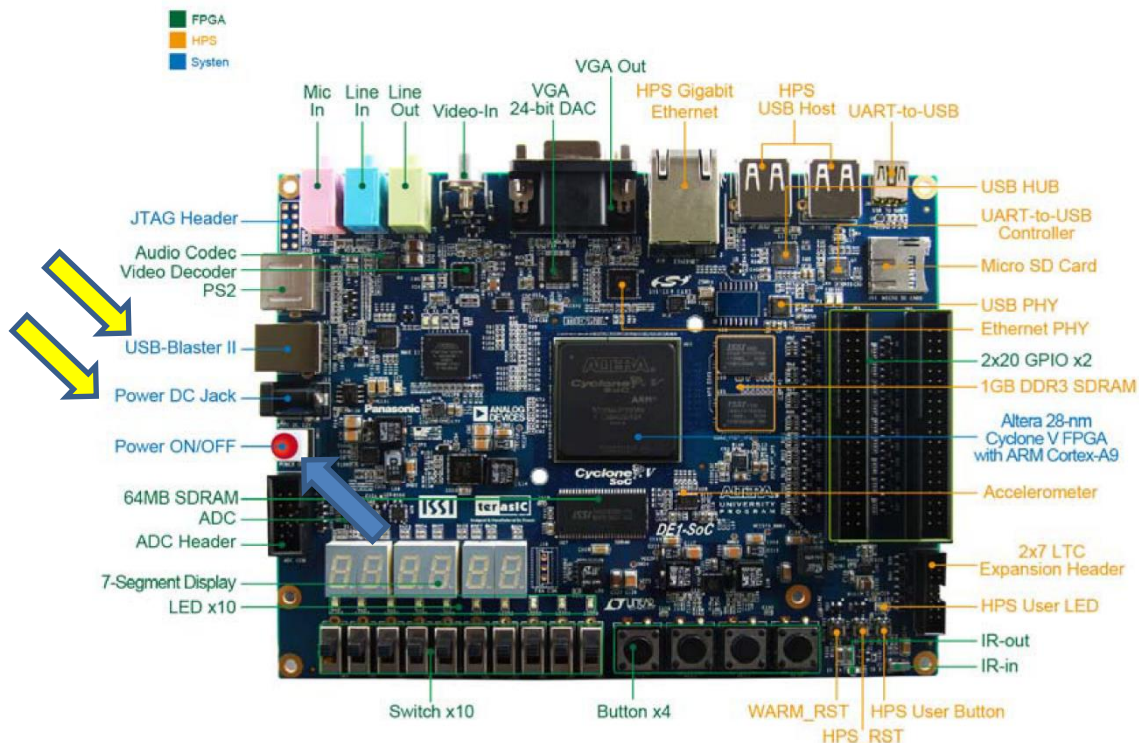
Figure 8: ALM for Cyclone V Devices



### 1) Ligando o kit DE1-SoC:

→Coloque a fonte de alimentação na tomada e conecte-a a placa.

→Conecte o cabo USB, uma extremidade no computador que possui o Quartus Prime instalado, e a outra extremidade na porta USB denominada USB BLASTER-II (escrito na placa).



→Ligue a placa pressionando, *cuidadosamente*, o botão vermelho.

Se os passos forem seguidos de maneira correta, a placa deve ligar, os LEDs devem piscar e os displays mostram uma contagem em hexadecimal.



## 7. SINTETIZANDO O PROJETO NO CHIP FPGA DA DE1-SOC

Após o projeto a ser implementado ter sido corretamente simulado por forma de onda (Funcional) e os requerimentos temporais (TimeQuest) atendidos, podemos sintetizá-lo no chip FPGA do kit de desenvolvimento.

Para tanto:

→ Crie o arquivo Top Level exemplo1.v do seu projeto, que define a interface entre seu projeto e os recursos disponíveis no kit DE1-SOC. Caso você tenha alterado, lembre-se de redefinir o exemplo1.v como o arquivo Top Level!

Os recursos mais básicos são:

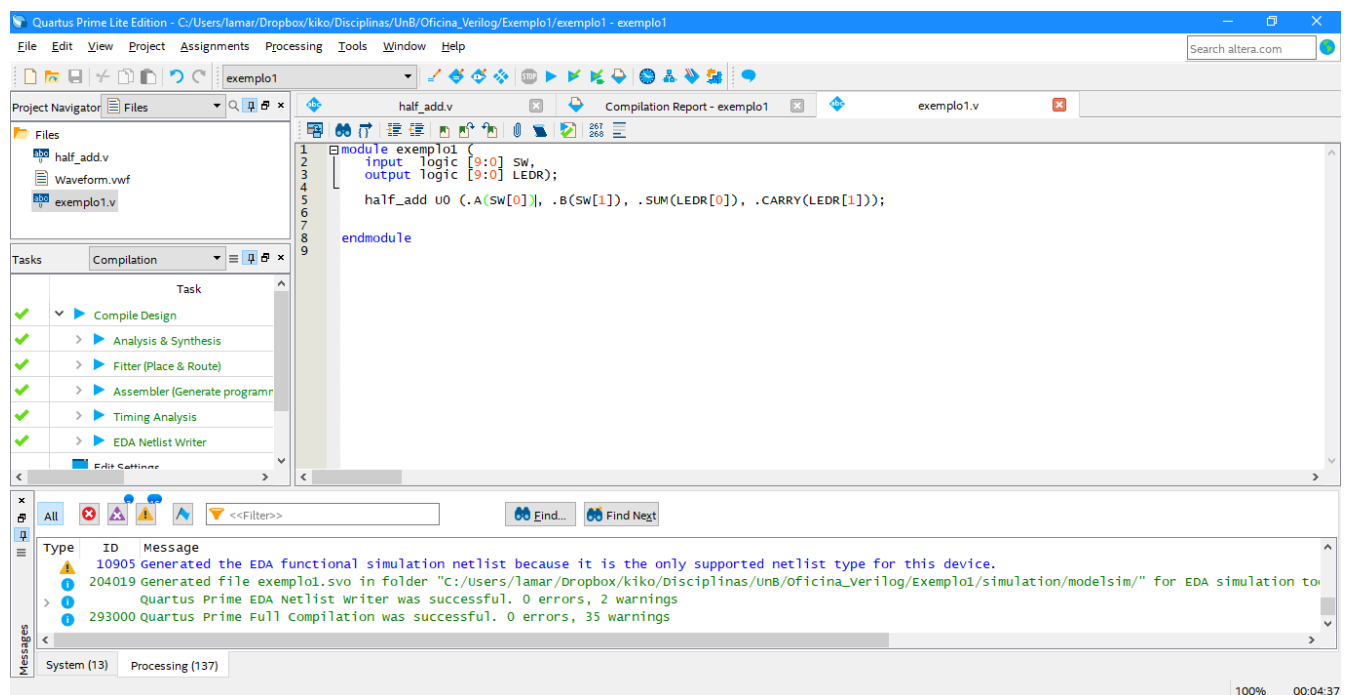
10 Chaves H-H: SW[9:0]

10 LEDs Vermelhos: LEDR[9:0]

4 Push-bottoms: KEY[3:0]

6 Displays de 7 segmentos: HEX0[6:0] a HEX5[6:0]

1 Clock de 50MHz: CLOCK\_50



→ Compile novamente seu projeto para gerar o arquivo .sof de configuração da FPGA  
- Corrija eventuais erros.

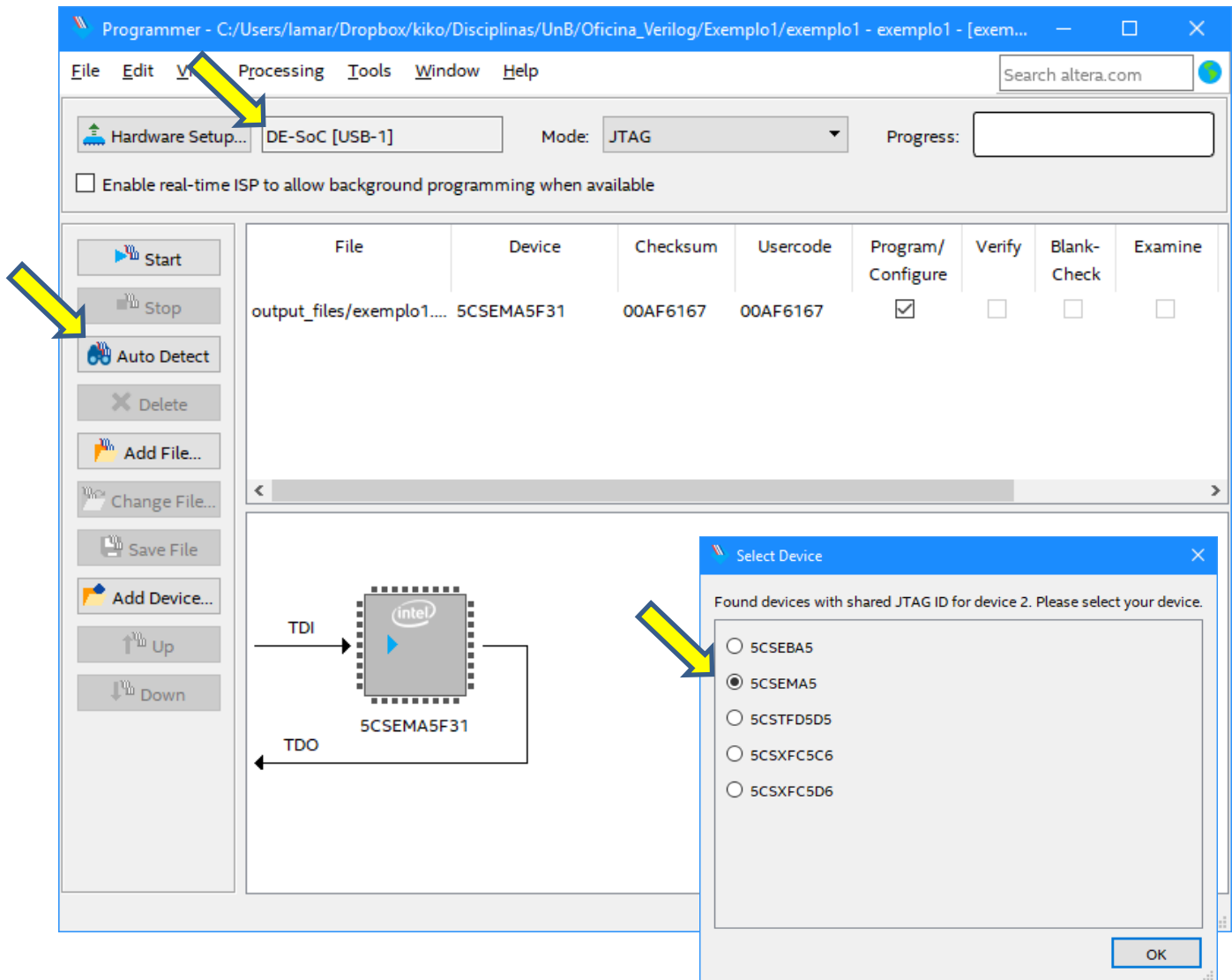
- Passando o seu projeto compilado para o chip FPGA:

→ Abra a janela do programador do chip FPGA através do ícone Programmer

→ Verifique se o cabo USB está corretamente conectado na placa DE1-SoC na entrada USB Blaster-II

- Verifique se o driver USB Blaster-II foi corretamente reconhecido:

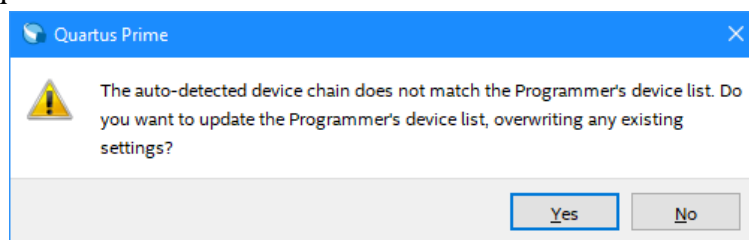
→ clique em Hardware Setup > Currently selected hardware > DE-SoC [USB-1]



→ clique em Auto Detect – para definir corretamente o dispositivo a ser usado

→ Selecione o device 5CSEMA5 e clique OK

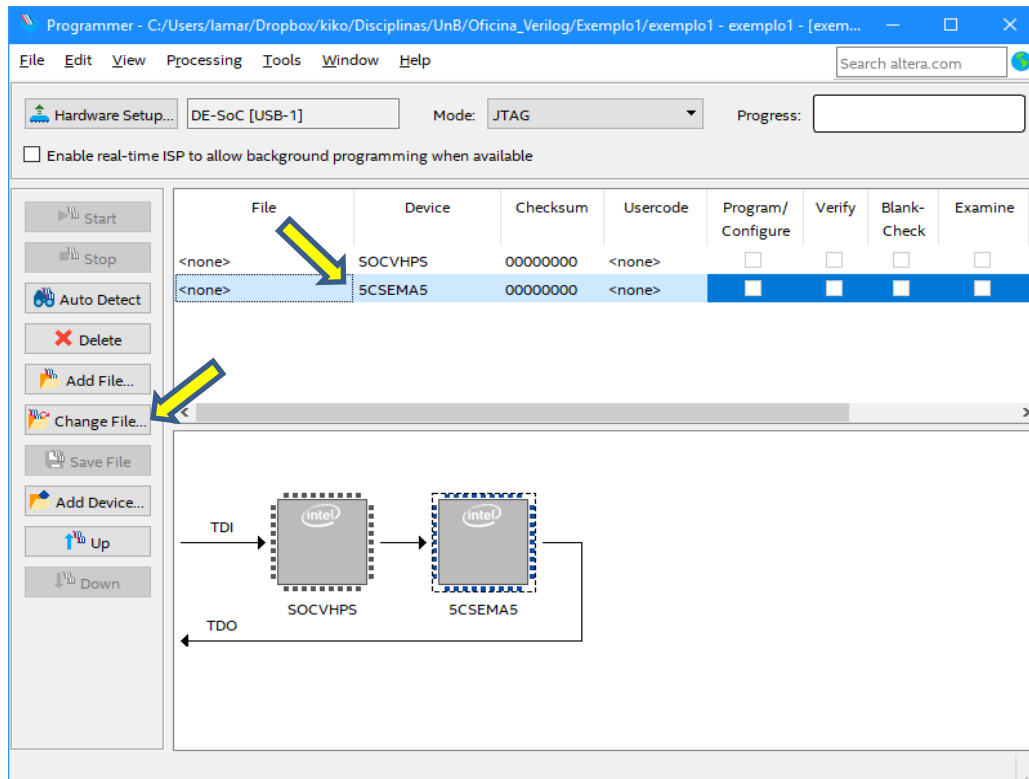
→ Confirme clicando YES







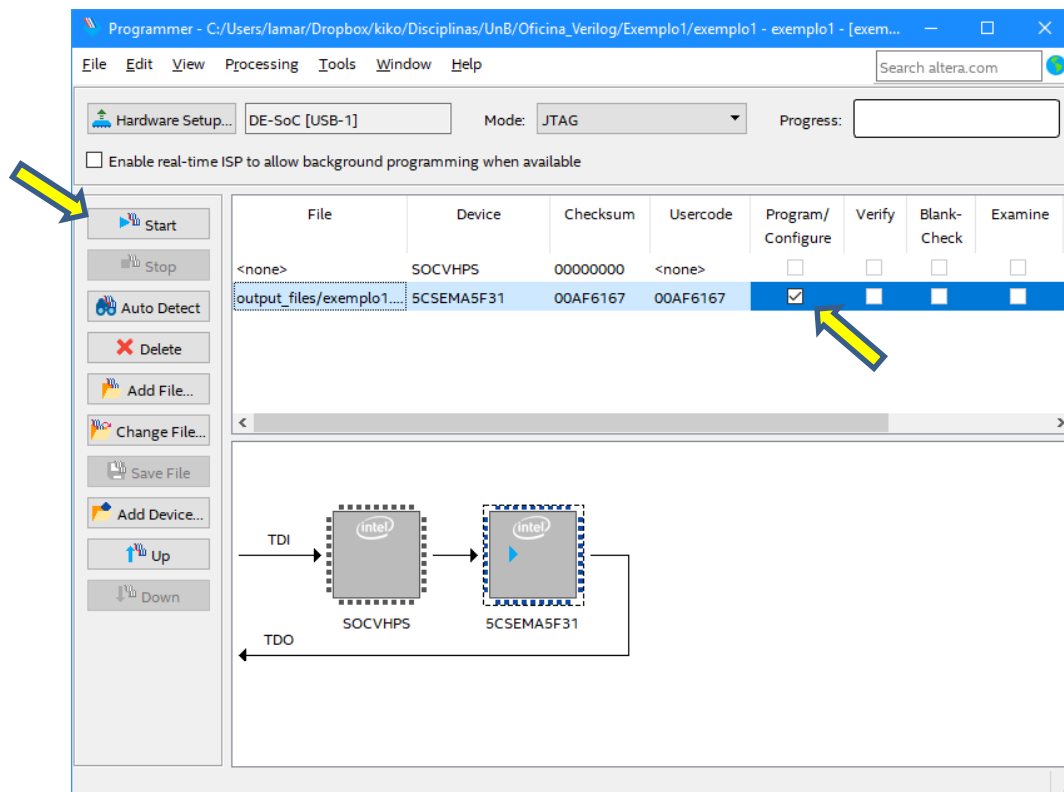
→ Selecione o Device 5CSEMA5 e clique em Change File...



→ Selecione o arquivo exemplo1.sof que deve estar no diretório output\_files.

→ Check a caixa Program/Configure.

→ Clique em Start





O kit DE1-SoC deverá apagar e instantes depois o seu circuito deverá estar carregado no FPGA. ☺

→Teste o circuito implementado usando as chaves(SW), botões(KEY), displays (HEX) e LEDs (LEDR) definidos no seu Projeto. Grave um vídeo mostrando o correto funcionamento do seu Projeto.

## **8. TERMINANDO O EXPERIMENTO**

Ao finalizar o experimento, guarde **corretamente** os cabos, a fonte e a placa na caixa e devolva ao professor.

Lembre-se de reconectar os computadores do LINF na tomada!!!



## F.A.Q

### Instalação

- Preciso instalar todos os devices que vêm no instalador do Quartus?

Não, somente é necessário instalar o da DE1-SOC (Cyclone V).

- Não consigo instalar o Quartus no Linux (instalador trava)

Durante o processo de instalação não instale nenhum componente não necessário, após concluir a instalação você pode manualmente instalar as outras partes que podem vir a travar o instalador (normalmente o modelsim e o Help são os responsáveis)

### Compilando o Projeto

- Preciso sempre recompilar todo o projeto após alguma alteração?

Não, para o TimeQuest e os simuladores basta completar a etapa de análise e síntese, apenas clique duas vezes em “Analysis & Synthesis” na janela abaixo da lista de arquivos. Para transferir para a placa é necessária uma compilação completa.

### TimeQuest

- Os valores estão estranhos:

Vá em Diagnostic > Report Clocks, confira se o clock presente é o que você criou, caso um clock não seja criado ou não haja nenhum clock atribuído a uma entrada de clock do circuito, o Quartus automaticamente atribui um clock padrão o que pode levar a valores fora do esperado.

### ModelSim

- Modelsim não abre no Linux:

O Quartus utiliza algumas bibliotecas extremamente desatualizadas, e que não são mais disponibilizadas pela maioria das distribuições, você deve utilizar o Red Hat Enterprise Linux 7 ou Ubuntu anterior a 13.10 para que funcione. Há outra forma, editar os scripts que lançam o modelsim de forma que seja executado um “export LD\_LIBRARY\_PATH” apontando para a versão correta das bibliotecas desatualizadas ([Arch Wiki](#)).



- Modelsim acusa que tem algum wire/reg não declarado/já declarado:

Deixe o código em Verilog estruturado corretamente (como você faria em C, C++, Java etc.) com as variáveis declaradas antes do primeiro uso.

## Transferindo o projeto para o chip FPGA

- A placa não é reconhecida pelo Quartus:

Windows:

Confira no gerenciador de dispositivos se a placa está sendo reconhecida pelo Windows, caso não, tente outra porta usb e/ou reinicie o computador. Tenha certeza que foi instalado o suporte para o dispositivo Cyclone V.

Linux:

[Siga estas instruções](#)

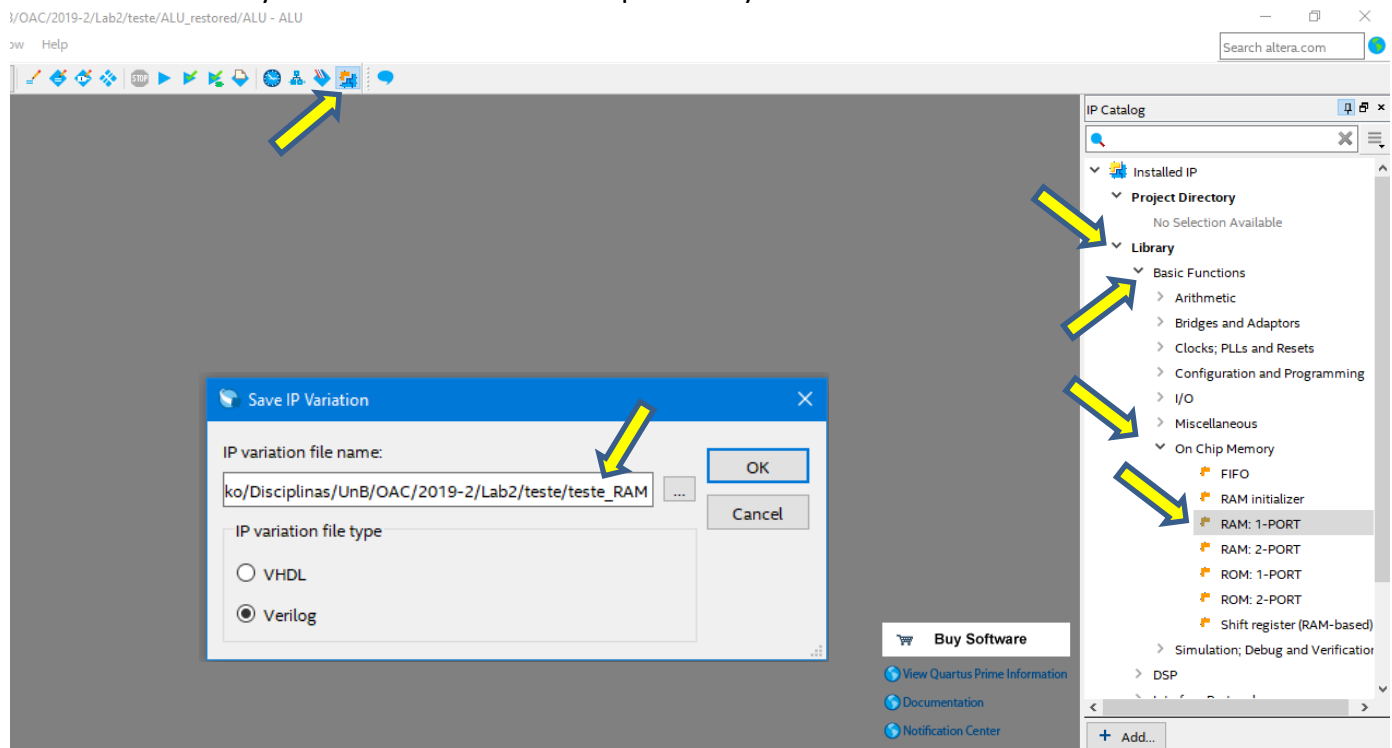


## ANEXO A

### Trabalhando com a Memória Interna do FPGA Cyclone-V

#### A.1. CRIANDO UM BLOCO DE MEMÓRIA RAM

- Iniciar a ferramenta IP Catalog
- Clique no ícone Ip Catalog
- Selecione Library > Basic Functions > On Chip Memory > RAM: 1-PORT



→ Defina o nome da memória: IP variation file name (Ex.: teste\_RAM) e a linguagem a ser usada: Verilog.

→ Escolha os parâmetros da memória: Tamanho da palavra (ex.:32 bits) e o tamanho da memória (ex.: 1024 words). Deixe os demais parâmetros no default.

→ Clique em Next >



MegaWizard Plug-In Manager [page 1 of 6]

**RAM: 1-PORT**

Parameter Settings | EDA | Summary

Widths/Blk Type/Cks | Regs/Cken/Byte Enable/Acds | Read During Write Option | Mem Init

Currently selected device family: Cyclone V

☒ Match previous default

How wide should the 'q' output bus be? 32 bits

How many 32-bit words of memory? 1024 words

Note: You could enter arbitrary values for width and depth.

What should the memory block type be?

☒ Auto ☐ MLAB ☐ M10K

☐ M-RAM ☐ LCs

Options...

Set the maximum block depth to Auto words

What docking method would you like to use?

☒ Single clock

☐ Dual clock: use separate 'input' and 'output' clocks

Resource Usage

4 M10K

Cancel < Back Next > Finish

→ Desselecione a opção de registrar a saída.

→ Clique em Next >

MegaWizard Plug-In Manager [page 2 of 6]

**RAM: 1-PORT**

Parameter Settings | EDA | Summary

Widths/Blk Type/Cks | Regs/Cken/Byte Enable/Acds | Read During Write Option | Mem Init

Which ports should be registered?

☒ 'data' and 'wren' input ports

☒ 'address' input port

☐ 'q' output port

☐ Create one clock enable signal for each clock signal.

Note: All registered ports are controlled by the enable signal(s).

More Options...

☐ Create byte enable for port A

What is the width of a byte for byte enables? 8 bits

☐ Create an 'acl' asynchronous clear for the registered ports

More Options...

☐ Create a 'rden' read enable signal

Resource Usage

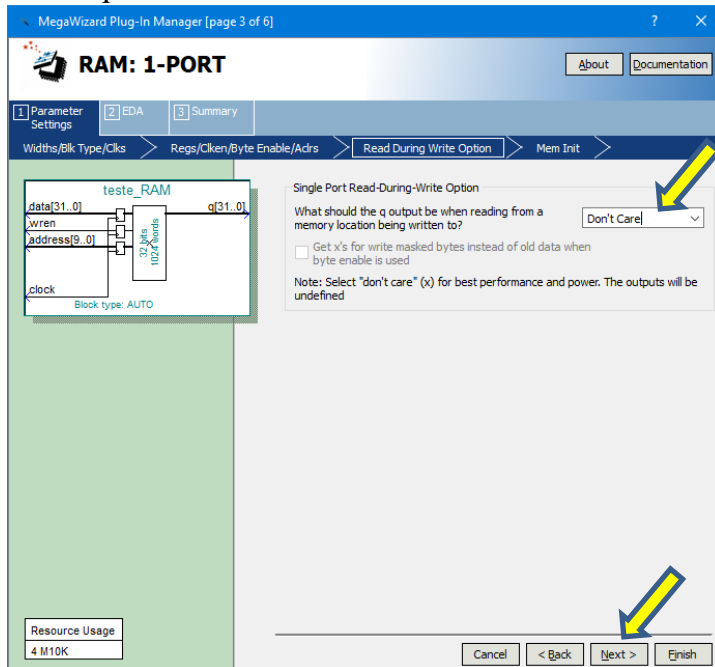
4 M10K

Cancel < Back Next > Finish



→ Selecione Don't Care para Single Port Read-During-Write.

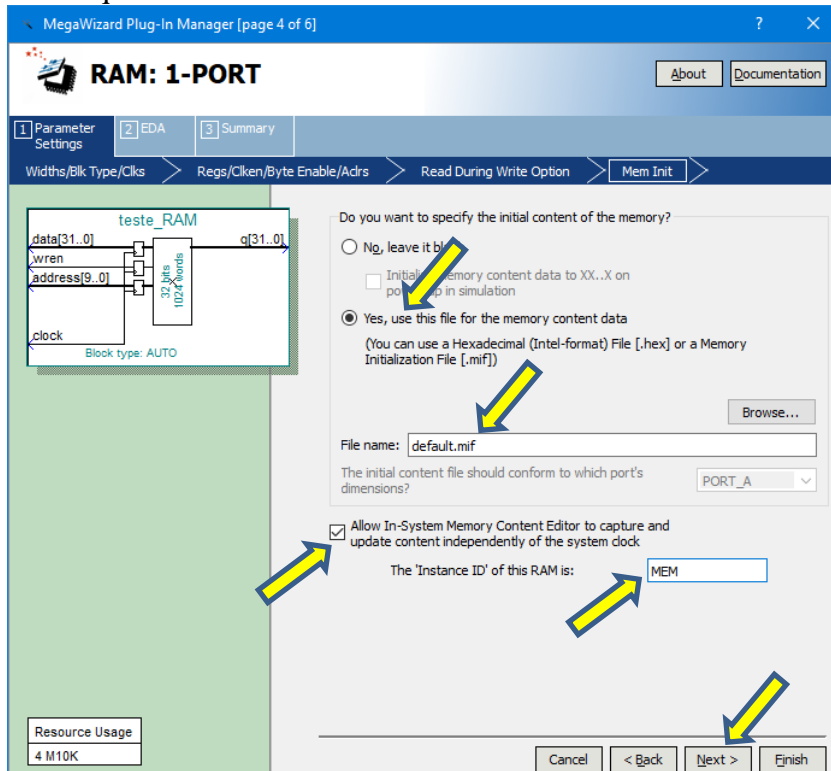
→ Clique em Next >



→ Selecione Yes, use this file for memory content data e coloque o nome do arquivo default.mif (ou o nome que vc desejar)

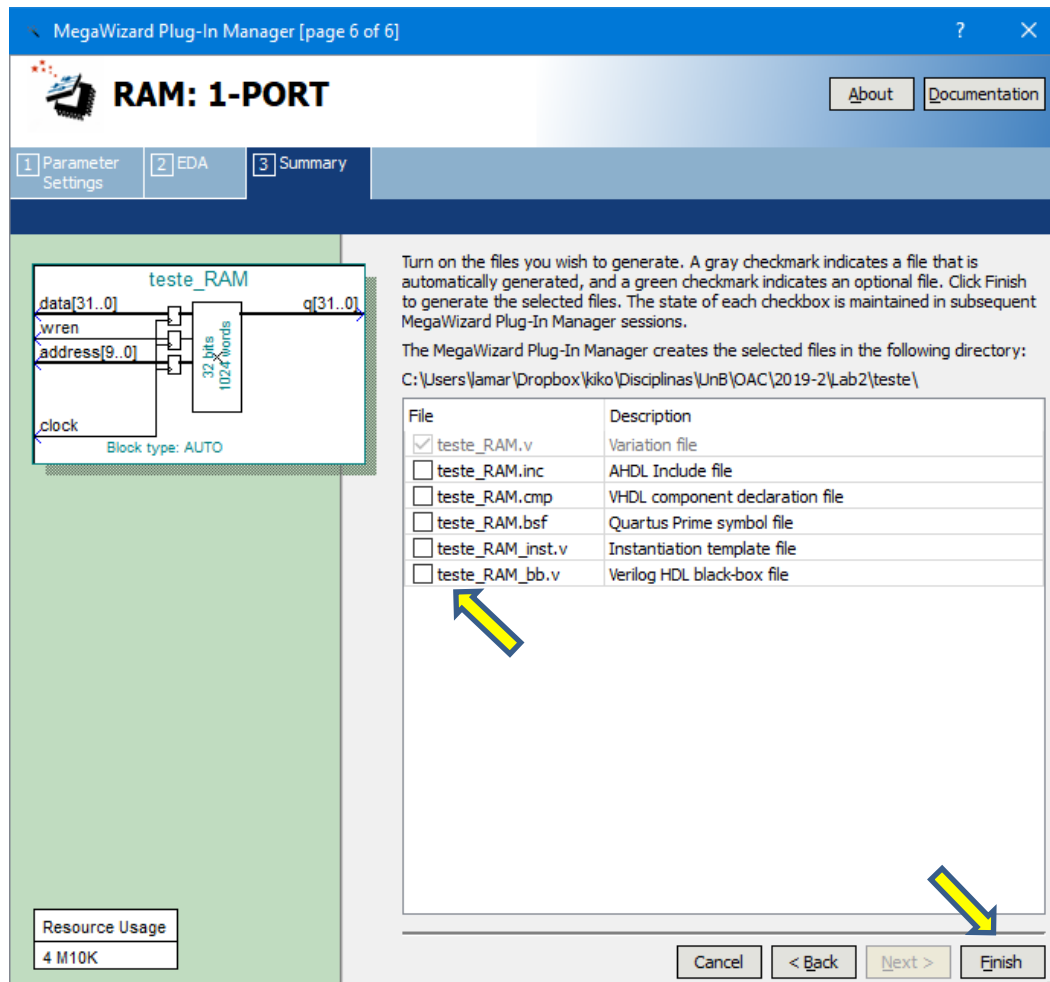
→ Selecione Allow In-System Memory Content Editor e defina o nome do bloco (Ex.: MEM)

→ Clique em Next >





- Desselecione a opção de criar o blackbox file \_bb.v .
- Clique em Finish



Serão criados os arquivos teste\_RAM.qip e o teste\_RAM.v para ser usado no seu projeto.

```
module teste_RAM (  
    input [9:0] address,  
    input      clock,  
    input [31:0] data,  
    input      wren,  
    output [31:0] q  
);
```

*Nesta memória a escrita é realizada em 1 ciclo de clock, porém a leitura necessita 2 ciclos de clock para ser finalizada devido aos registradores de entrada .*

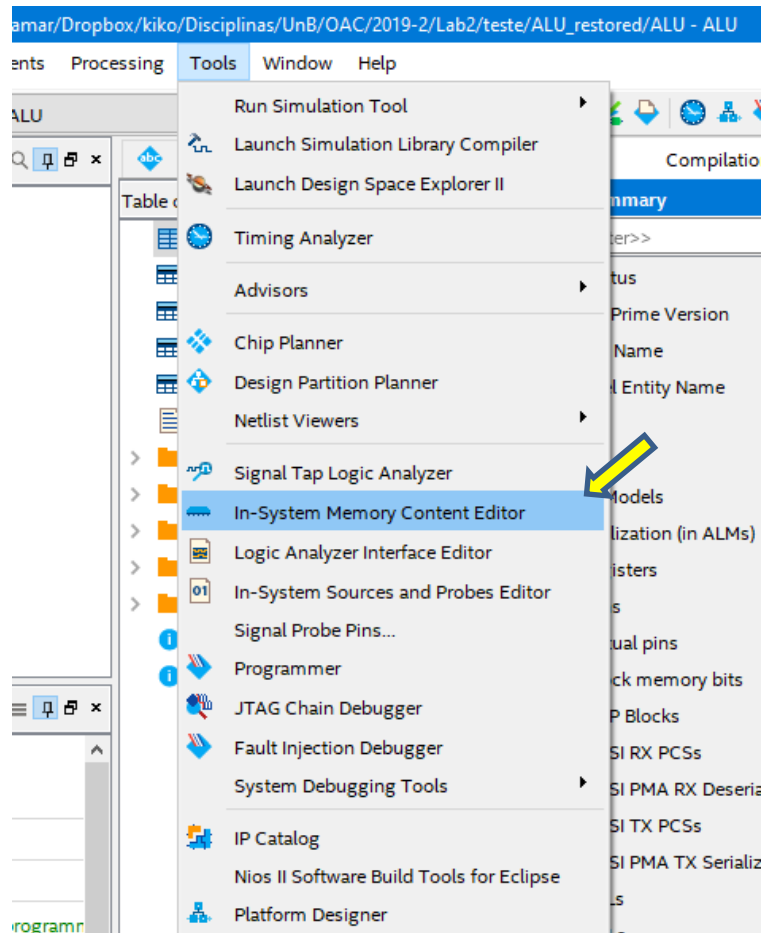




## A.2. PROGRAMANDO O BLOCO DE MEMÓRIA EM TEMPO DE EXECUÇÃO

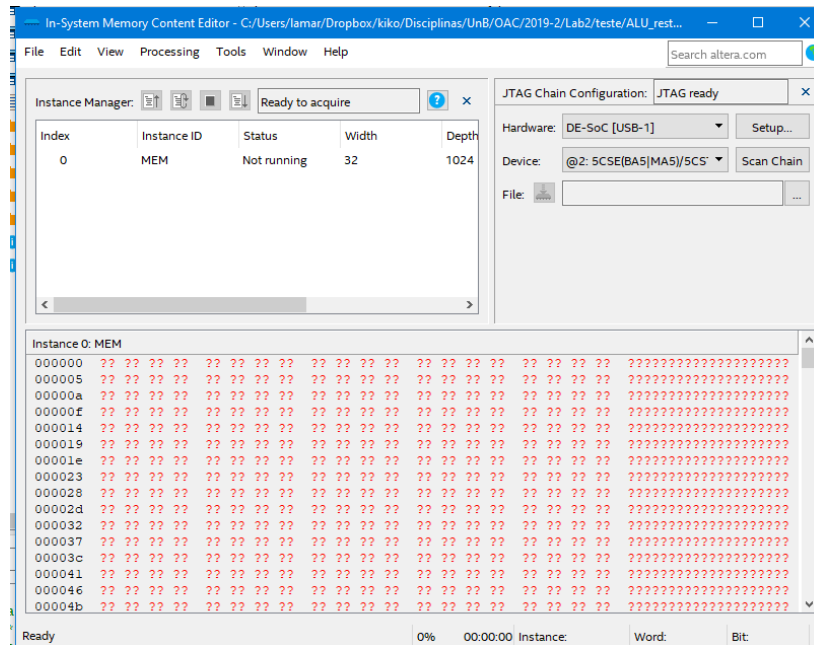
O block de memória criado é inicializado de acordo com o arquivo default.mif definido previamente. Porém, podemos visualizar e alterar seu conteúdo após o upload do projeto para o FPGA usando a ferramenta In-System-Memory Content Editor.

→ Abra a ferramenta Tools/In-System Memory Content Editor

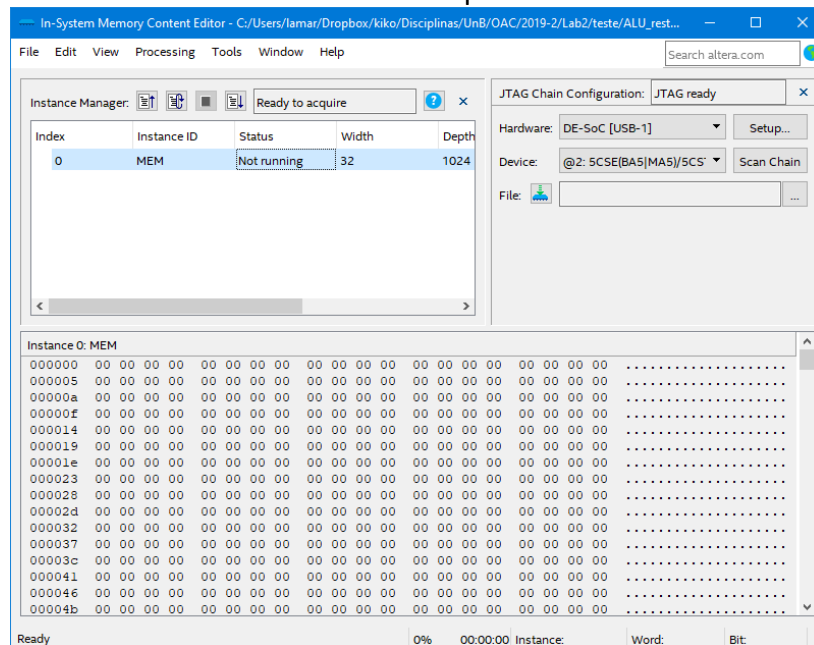




A janela da ferramenta irá aparecer mostrando a instância de memória RAM que foi criada, nome MEM, width 32, Depth 1024



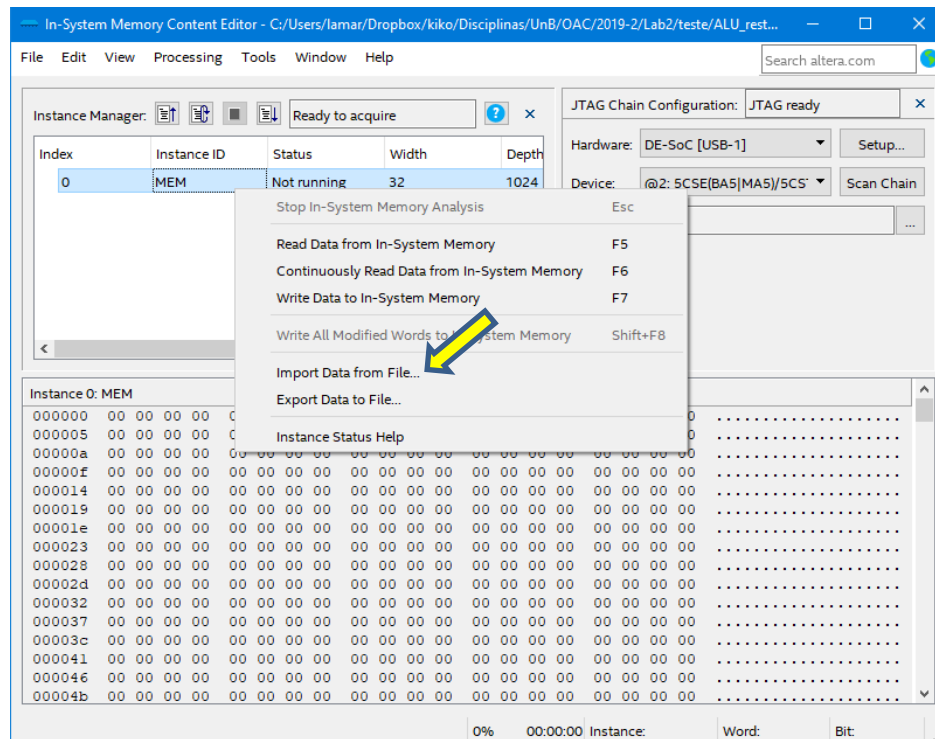
→ Pressione F5 para ler o conteúdo atual da memória presente no FPGA



Neste exemplo a memória está completamente vazia (com 00s)

Você pode editar **manualmente** seu conteúdo ou **importar** um arquivo mif clicando no botão direito e selecionando a opção Import Data from File...

→ Selecione o formato de arquivo mif e o arquivo que deseja importar.



**IMPORTANTE:** Após a importação é necessário escrever o novo conteúdo da memória no FPGA, pressionando F7 (Write Data to In-System Memory)!

