

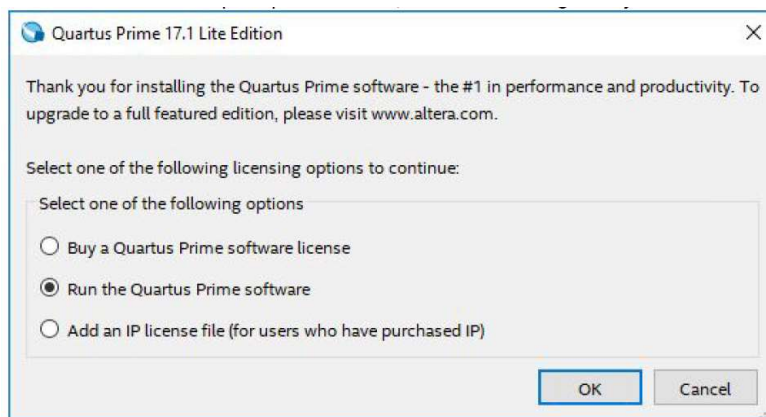
Laboratório 01

Instruções:

- Este laboratório é um tutorial para familiarização com a ferramenta. Leia todas as instruções com atenção e faça todos os passos.
- Os nomes dos arquivos devem ser seguidos, e isso faz parte da avaliação.

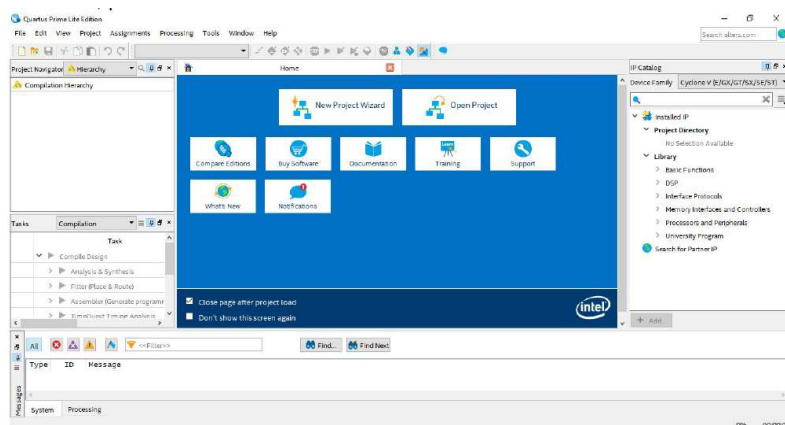
Parte I - Iniciando o Quartus pela primeira vez

I-1. Ao abrir o Quartus pela primeira vez, você verá a seguinte janela:



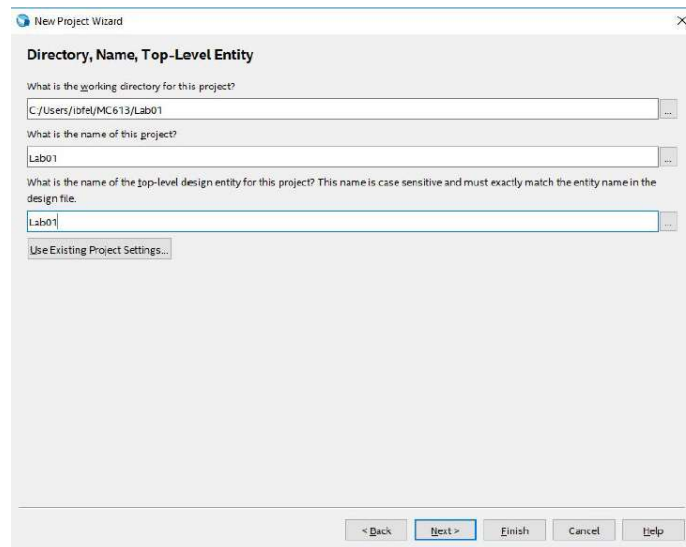
Selecione **Run the Quartus Prime software** e clique **OK**.

I-2. Você verá a janela inicial do Quartus. Se ela não aparecer dentro de alguns instantes, pode ser necessário executar o Quartus novamente.

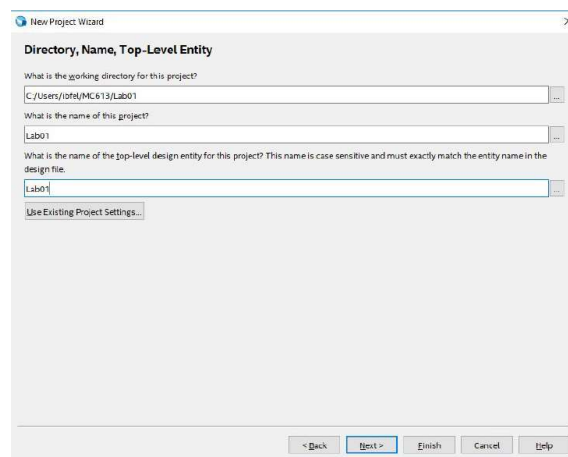


Parte II - Criando um projeto

II-1. Da janela principal do Quartus, selecione **File > New Project Wizard...**



II-2. Na janela **New Project Wizard**, parte **Introduction**, pressione **Next >**.



II-3. Em **Directory, Name, Top-Level Entity**, indique o caminho onde o projeto será salvo, o nome do projeto (por exemplo, *Lab01*) e o nome da entidade top-level (por exemplo, *Lab01*). A entidade top-level é a entidade mais acima na hierarquia do projeto e aquela que se comunicará com os pinos de entrada e saída da placa DE1-SoC.

New Project Wizard

Directory, Name, Top-Level Entity

What is the working directory for this project?

C:/Users/lofel/MC613/Lab01

What is the name of this project?

Lab01

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.

Lab01

Use Existing Project Settings...

< Back Next > Finish Cancel Help

II-4. Em **Project Type** selecione **Empty project** e deixe a janela seguinte, **Add Files**, em branco.

New Project Wizard

Project Type

Select the type of project to create

☒ 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

New Project Wizard

Add Files

Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project. Note you can always add design files to the project later.

File name: Add

Add All

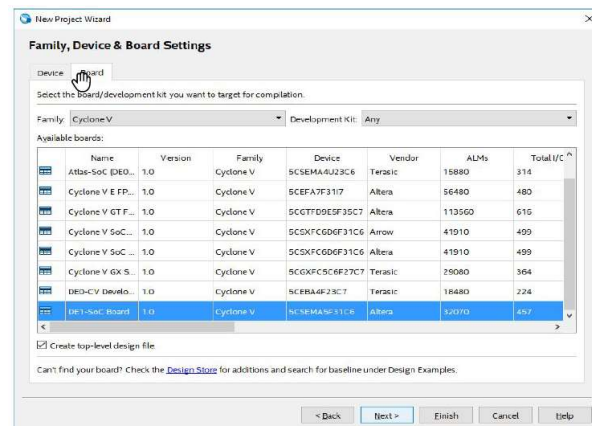
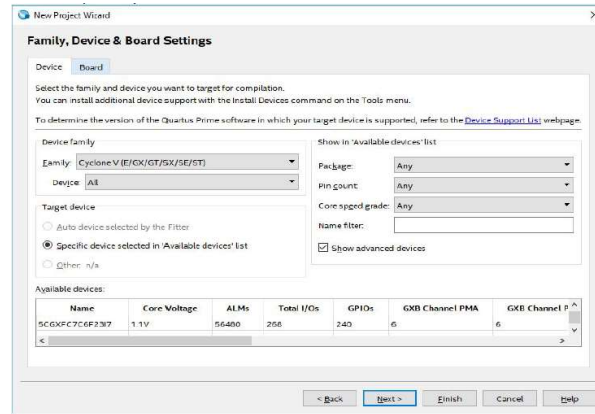
File Name	Type	Library	Design Entry/Synthesis Tool	HDL Version

Remove Up Down Properties

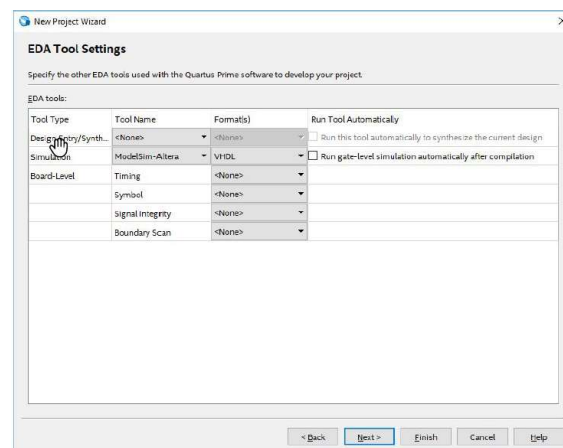
Specify the path names of any non-default libraries. Use Libraries...

< Back Next > Finish Cancel Help

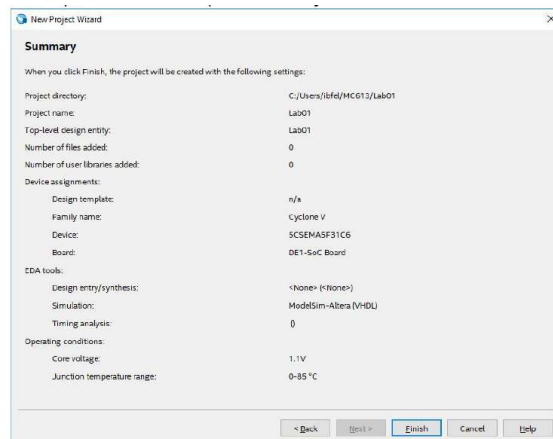
II-5. Em Family, Device & Board Settings selecione a aba **Board** e procure na lista a placa que usaremos, **DE1-SoC Board**.



II-6. Em EDA Tool Settings configure a ferramenta de simulação como **ModelSim-Altera** e o formato como **VHDL**, conforme a imagem abaixo.

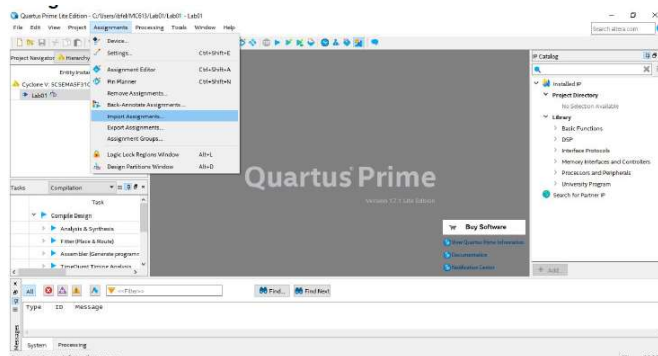


II-7. Clique em **Finish** na etapa de **Summary**.

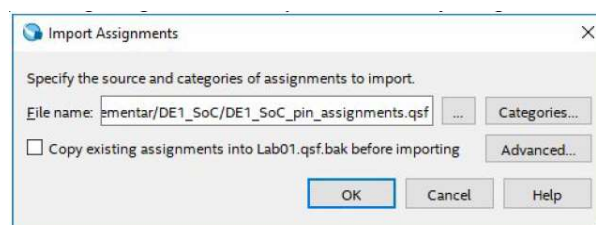


II-8. Acesse a pasta no CANVAS da disciplina e faça o download do arquivo **DE1_SoC_pin_assignments.qsf**, dentro da pasta **DE1-SoC**.

II-9. Da janela principal do Quartus, vá em **Assignments > Import Assignments...**



II-10. Selecione o arquivo que você baixou no passo 8, desmarque a caixa **Copy existing assignments into *.qsf.bak before importing** e confirme.



II-11. De volta à janela principal, vá em **Assignments > Assignment Editor**. Você

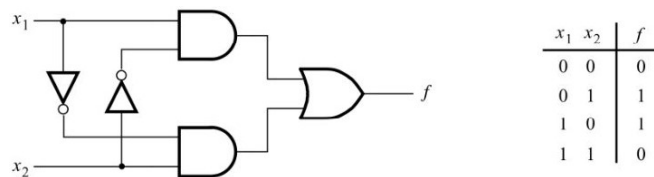
verá uma lista de assignments semelhante à da imagem abaixo.

tabs	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	ADC_CS_N	Location	PIN_A14	3.3-V LVTTTL	Yes	DE1_SoC		
2	ADC_DIN	Location	PIN_AK4	3.3-V LVTTTL	Yes	DE1_SoC		
3	ADC_DOUT	Location	PIN_AK3	3.3-V LVTTTL	Yes	DE1_SoC		
4	ADC_SCLK	Location	PIN_AK2	3.3-V LVTTTL	Yes	DE1_SoC		
5	ADC_CS_N	I/O Standard	3.3-V LVTTTL	3.3-V LVTTTL	Yes	DE1_SoC		
6	ADC_DIN	I/O Standard	3.3-V LVTTTL	3.3-V LVTTTL	Yes	DE1_SoC		
7	ADC_DOUT	I/O Standard	3.3-V LVTTTL	3.3-V LVTTTL	Yes	DE1_SoC		
8	ADC_SCLK	I/O Standard	3.3-V LVTTTL	3.3-V LVTTTL	Yes	DE1_SoC		
9	AUD_CDAT	Location	PIN_K7	3.3-V LVTTTL	Yes	DE1_SoC		
10	AUD_LFCLK	Location	PIN_K0	3.3-V LVTTTL	Yes	DE1_SoC		
11	AUD_BCLK	Location	PIN_H7	3.3-V LVTTTL	Yes	DE1_SoC		
12	AUD_CDAT	Location	PIN_J7	3.3-V LVTTTL	Yes	DE1_SoC		
13	AUD_LFCLK	Location	PIN_H8	3.3-V LVTTTL	Yes	DE1_SoC		
14	AUD_XCK	Location	PIN_G7	3.3-V LVTTTL	Yes	DE1_SoC		
15	AUD_CDAT	I/O Standard	3.3-V LVTTTL	3.3-V LVTTTL	Yes	DE1_SoC		
16	AUD_LFCLK	I/O Standard	3.3-V LVTTTL	3.3-V LVTTTL	Yes	DE1_SoC		
17	AUD_BCLK	I/O Standard	3.3-V LVTTTL	3.3-V LVTTTL	Yes	DE1_SoC		
18	AUD_CDAT	I/O Standard	3.3-V LVTTTL	3.3-V LVTTTL	Yes	DE1_SoC		
19	AUD_LFCLK	I/O Standard	3.3-V LVTTTL	3.3-V LVTTTL	Yes	DE1_SoC		
20	AUD_XCK	I/O Standard	3.3-V LVTTTL	3.3-V LVTTTL	Yes	DE1_SoC		
21	CLOCK_50	Location	PIN_AF14	3.3-V LVTTTL	Yes	DE1_SoC		
22	CLOCK_50	I/O Standard	3.3-V LVTTTL	3.3-V LVTTTL	Yes	DE1_SoC		

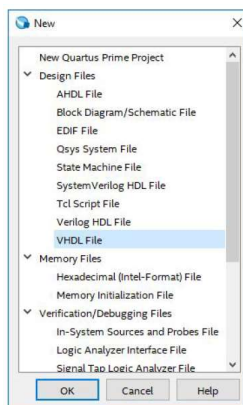
Vamos focar apenas nos itens da lista que possuem a coluna **Assignment Name** como **Location**, que indicam a correspondência entre os pinos da placa a os sinais no circuito. A coluna **To** indica o nome de um possível sinal no circuito (no código VHDL), de acordo com as convenções definidas no manual da placa (consulte o manual na pasta de Material Complementar). A coluna **Value** contém o código de um pino na placa para fazer a conexão. Por exemplo, o sinal **SW[0]** é o primeiro switch-button da placa (**SW0**) e será conectado ao pino **PIN_AB12**.

Parte III - Escrevendo o primeiro código VHDL III-1.

III-1. Nosso circuito de exemplo será um circuito de controle de interruptores do tipo “chave-hotel”, em que dois interruptores controlam uma mesma lâmpada, permitindo acendê-la em um interruptor e apagá-la em outro. Veja o diagrama do circuito lógico e a tabela-verdade abaixo, em que x_1 e x_2 são os interruptores e f é a lâmpada.



III-2. Crie um novo arquivo a partir da janela principal do Quartus, dentro de um projeto, em **File > New....** Selecione um arquivo do tipo **Design Files > VHDL File**.



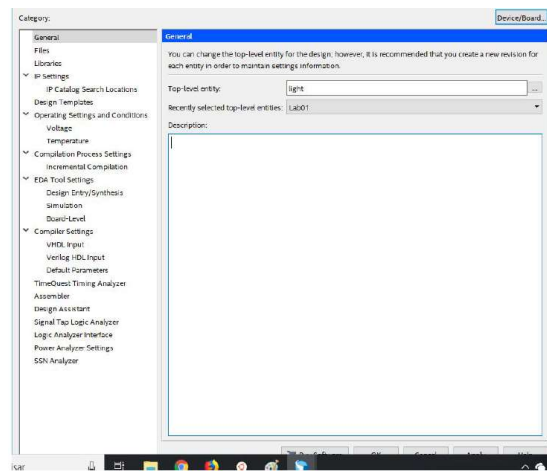
III-3. Escreva no novo arquivo o código em VHDL que representa o circuito:

```
Text Editor - C:/Users/ibfel/MC613/Lab01/Lab01 - Lab01 - [Vhdl1.vhd]*
File Edit View Project Processing Tools Window Help Search altera.com
1  LIBRARY ieee;
2  USE ieee.std_logic_1164.all;
3
4  ENTITY light IS
5  PORT(
6      x1, x2 : IN STD_LOGIC;
7      f : OUT STD_LOGIC
8  );
9  END light;
10
11 ARCHITECTURE LogicFunction OF light IS
12 BEGIN
13     f <= (x1 AND NOT x2) OR (NOT x1 AND x2);
14 END LogicFunction;
```


III-4. Salve o arquivo (**File > Save**) com o nome **light.vhd**. Certifique-se de marcar a caixa **Add file to current project**.

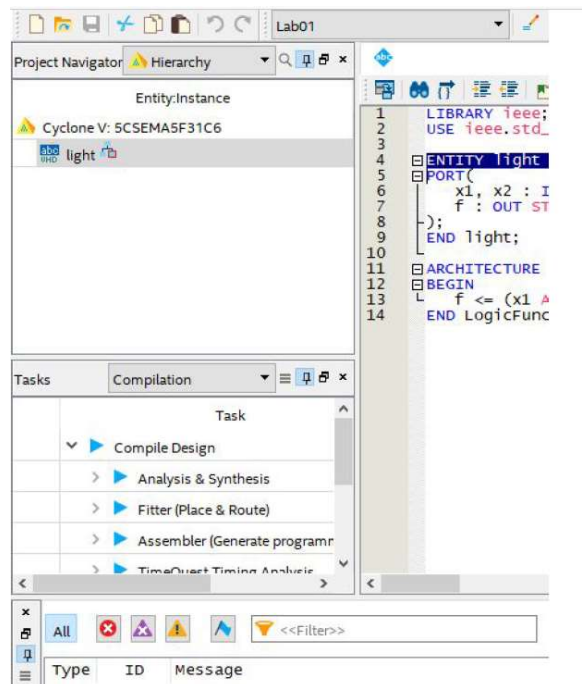
ENTREGA: arquivo **light.vhd**.

III-5. No código VHDL, nós chamamos a entidade principal de **light**. Como essa será a entidade top-level do projeto, temos que instruir o Quartus a buscar a entidade certa. Para isso, vá em **Assignments > Settings**, na aba **General** e escreva o nome da entidade top-level como **light**.

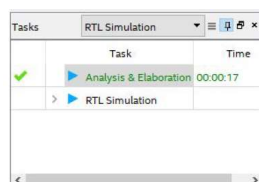


Parte IV - Simulando o circuito

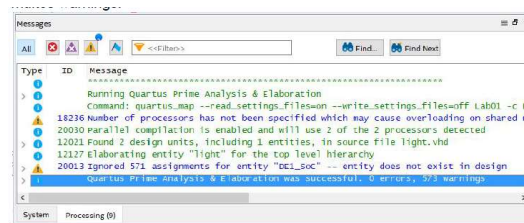
IV-1. Para iniciar a simulação, vamos utilizar as tarefas disponíveis no painel **Tasks**, à esquerda da tela. Se o painel não estiver visível, exiba-o pelo menu **View > Utility Windows > Tasks**.



IV-2. Selecione **RTL Simulation** no menu drop-down do painel **Tasks**. Você verá duas tarefas disponíveis: **Analysis & Elaboration** e **RTL Simulation**. Execute primeiro **Analysis & Elaboration** clicando duas vezes sobre ela. Se tudo estiver certo com o seu código, a tarefa terminará de executar com sucesso.

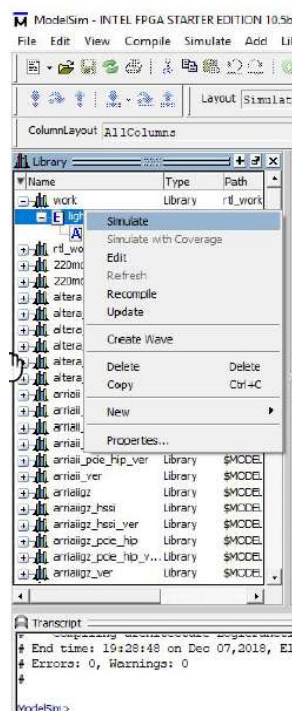


IV-3. Na parte inferior da janela principal do Quartus você verá o painel de log, **Messages** (se não estiver visível, exiba-o em **View > Utility Windows > Messages**). Se tudo estiver certo, você verá ao fim uma mensagem indicando que Analysis & Elaboration executou com sucesso, sem nenhum erro e com muitos warnings.

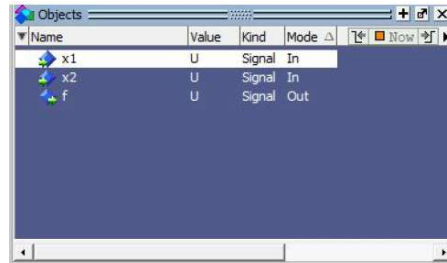


É normal que o código VHDL compile com muitos warnings. Apesar disso, vale a pena reservar um tempo para analisar os warnings, pois muitos erros comuns podem ser mais facilmente identificados a partir deles.

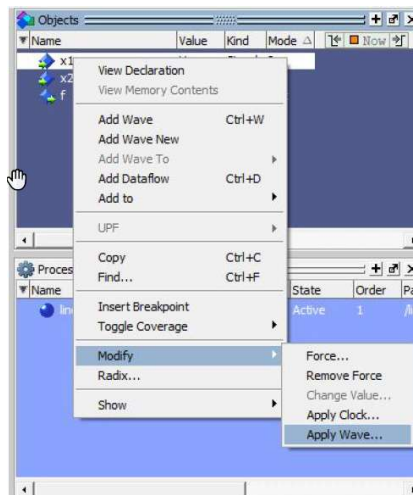
IV-4. De volta ao painel **Tasks**, dê um clique duplo na tarefa **RTL Simulation**. O Quartus executará a aplicação ModelSim. Dentro do ModelSim, você verá o painel **Library** (se não estiver visível, exiba-o em **View > Library**), listando várias bibliotecas. Dentro da biblioteca **work**, você verá a entidade **light** que acabou de criar. Clique com o botão direito sobre a entidade **light** e selecione **Simulate**.



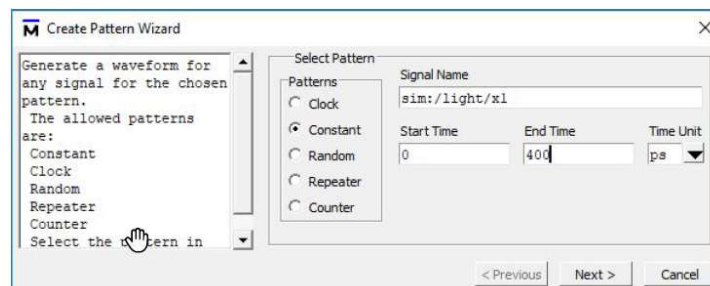
IV-5. O painel **Objects** (se não estiver visível, exiba-o em **View > Objects**) será populado com os elementos da entidade escolhida, no caso as entradas **x₁** e **x₂** e a saída **f**.



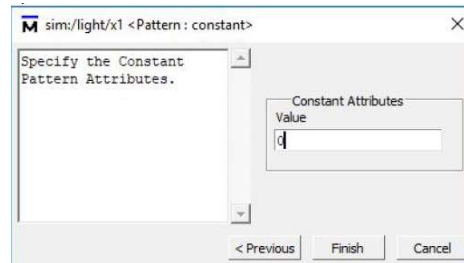
IV-6. Clique com o botão direito sobre o sinal **x₁**, selecione **Modify > Apply Wave...** para começar a criar uma onda de entrada para este sinal.



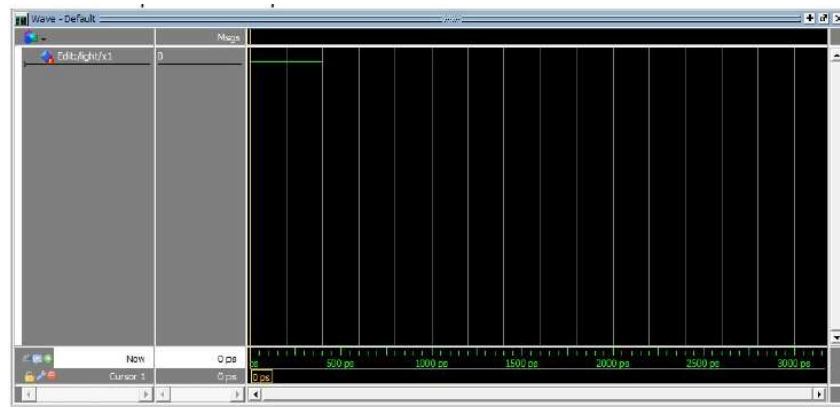
IV-7. Na janela **Create Pattern Wizard**, selecione o **Pattern** como **Constant**, e configure **Start Time** e **End Time** como **0** e **400 ps**. Clique **Next >**.



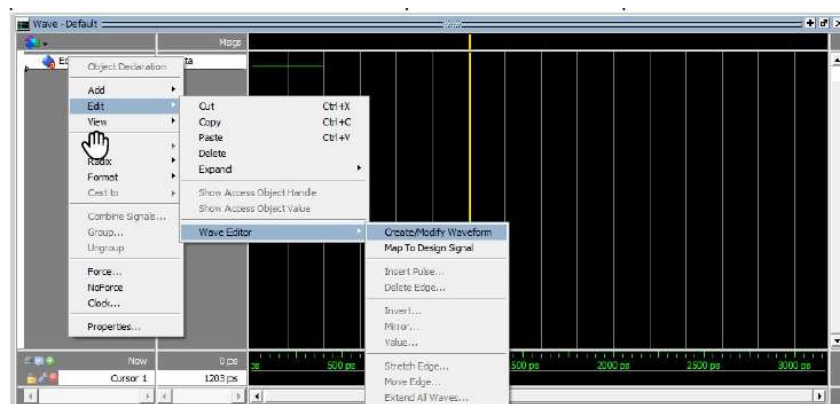
IV-8. Na janela seguinte, dê o valor **0** para o período especificado da onda de entrada e clique **Finish**.



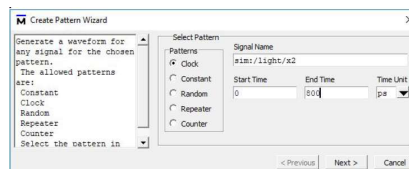
IV-9. O ModelSim exibirá o painel **Wave** com a onda de entrada **x₁** e o valor 0 entre os tempos 0 e 400 ps.



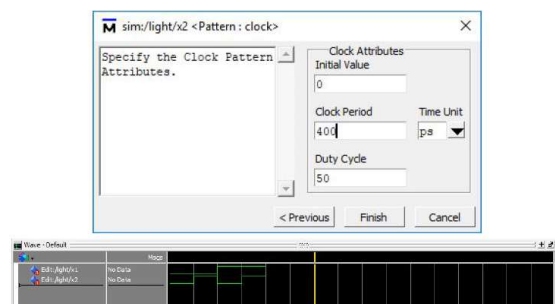
IV-10. Para adicionar um segundo valor à onda, clique com o botão direito sobre ela no painel **Wave** e selecione **Edit > Wave Editor > Create/Modify Waveform**. Você verá novamente a janela **Create Pattern Wizard**. Utilize-a para incluir na onda o valor **1** nos tempos entre 400 e 800 ps.



IV-11. Uma outra maneira de definir ondas que oscilam entre 0 e 1 é definindo a elas um formato de sinal de relógio (clock). Para isso, clique com o botão direito no sinal **x₂** no painel **Objects** e selecione **Modify > Apply Wave...** novamente. Na janela **Create Pattern Wizard** selecione o **Pattern** como **Clock** e **Start Time** e **End Time** como os tempos totais para a onda, no caso, **0** e **800 ps**.



IV-12. Na janela seguinte, marque o valor inicial (**Initial Value**) da onda como **0** e o período (**Clock Period**) como **400 ps**. Observe a forma de onda resultante no painel **Wave**.



IV-13. Adicione o sinal de saída **f** no painel **Wave** simplesmente clicando e arrastando ele desde o painel **Objects**.

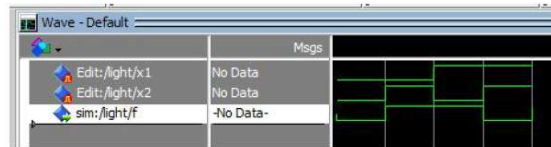
IV-14. Na barra de tarefas do ModelSim, procure pelas opções de simulação, que têm a seguinte aparência:



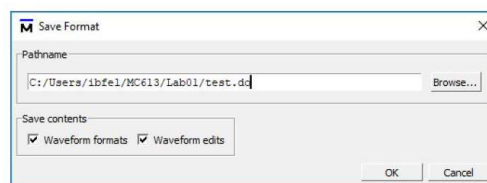
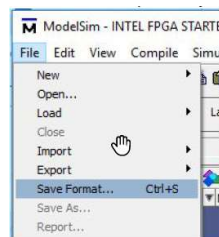
Caso não estejam visíveis, exiba-as em **Window > Toolbars > Simulate**.

IV-15. Configure o tempo de simulação como **800 ps** () e inicie a simulação clicando em **Run** (). O simulador completará as formas de onda do painel **Wave** com a onda de saída do sinal **f**. Compare a saída com a tabela-verdade esperada para atestar sua corretude.

ENTREGA: uma screenshot desta simulação, onde apareçam os sinais **x₁**, **x₂** e **f**, no arquivo **simulation.png**.



IV-16. Agora que você criou sua forma de onda de teste, salve-a para uso futuro, para que não precise redesenhá-la novamente. Para isso, selecione **File > Save Format...** e escolha o nome do arquivo na janela **Save Format**.



IV-17. Para recuperar a forma de onda salva após fechar o ModelSim, inicie uma nova simulação (passos 1-4) e selecione **File > Load > Macro File....** e escolha o arquivo ***.do** salvo.

