

## Introdução aos Sistemas Digitais

### Introdução à captura de diagramas esquemáticos e simulação comportamental baseadas em ferramentas de projeto assistido por computador

#### Objetivos

- Familiarização com as ferramentas de projeto que vão ser usadas nas aulas práticas.
- Captura de diagramas esquemáticos (esquemas lógicos) e simulação comportamental de componentes elementares.

#### Sumário

Este trabalho prático está dividido em 2 partes. Na parte I pretende-se introduzir os aspetos básicos das ferramentas que vão ser usadas nas aulas práticas de Introdução aos Sistemas Digitais (ISDig) com base num projeto simples recorrendo à captura de diagrama esquemático e simulação comportamental. Na parte II é abordado um exemplo de projeto hierárquico, baseado num comparador de igualdade.

#### Resumo do Fluxo de Projeto

A Figura 1 ilustra o fluxo de projeto típico de sistemas digitais baseados em hardware reprogramável como é o caso das FPGA (*Field Programmable Gate Array*). Em ISDig apenas são abordadas as etapas iniciais. As restantes serão objeto de estudo na disciplina subsequente “Laboratórios de Sistemas Digitais”.

A etapa de *design entry* consiste na modelação, codificação ou introdução da funcionalidade pretendida, podendo para tal o projetista usar linguagens de descrição de hardware, diagramas esquemáticos, diagramas de transição de estado ou outros métodos. No caso de ISDig vai ser utilizada a captura de diagramas esquemáticos e/ou diagramas de transição de estado, consoante o que for mais adequado para cada sistema. Por conveniência vai ser usado o ambiente de desenvolvimento (*Integrated Development Environment – IDE*) “Altera Quartus Prime”, o qual possui editores de diagramas esquemáticos e de diagramas de estados, além de outras ferramentas necessárias para realizar todos os passos do fluxo de projeto.

Depois de modelado o sistema, o passo seguinte é a sua síntese (lógica), isto é, a compilação do modelo de forma a criar uma *netlist* (i.e. um conjunto de portas lógicas, flip-flops, multiplexadores, outros componentes e respetivas interligações) que implementa a funcionalidade pretendida. Este passo é realizado por ferramentas de software normalmente desenvolvidas pelo fabricante da tecnologia de implementação usada.

A validação do sistema pode ser realizada em diversas etapas do fluxo de projeto, por simulação ou verificação em hardware real. São também normalmente disponibilizadas pelo fabricante diversas ferramentas de análise temporal, energética, recursos lógicos utilizados, etc. Neste trabalho prático vai ser realizada apenas a simulação comportamental.

**Notas importantes:** devido a limitações de velocidade da rede no acesso ao diretório pessoal em arca.ua.pt, nos PCs das salas de aula recomenda-se a utilização de uma *pendrive* para guardar os seus projetos e ficheiros. Caso isso não seja possível, guarde e aceda ao seu trabalho a partir de um diretório da drive Z:

Em qualquer dos casos não utilize espaços nem caracteres especiais (e.g. acentos) nas *paths* dos projetos e ficheiros, uma vez que isso causará problemas na utilização das ferramentas (isto significa que não deve gravar os seus projetos em sub-diretórios do “Ambiente de Trabalho” ou dos “Meus Documentos”).

**Sugestão:** Em Windows crie, por exemplo em C:\Users\<Utilizador>, uma estrutura de diretórios para gravar os seus projetos de ISDig (e.g. C:\Users\<Utilizador>\ISDig\Aula1\Parte1\<Nome do Projeto>); em Linux poderá fazê-lo em /home/<utilizador>/ISDig/Aula1/Parte1/<Nome do Projeto>).

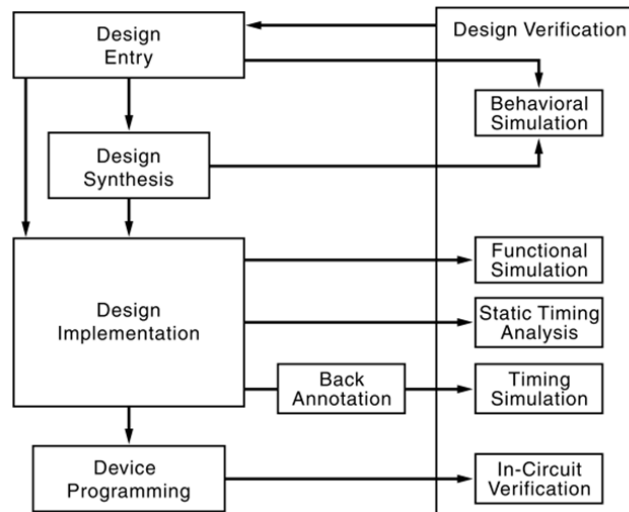


Figura 1 – Fluxo de projeto para sistemas digitais baseados em FPGA (fonte: www.xilinx.com).

### Parte I

*Demonstração das etapas fundamentais do fluxo de projeto com base em captura de diagrama esquemático e simulação comportamental*

1. Abra a aplicação “Altera Quartus Prime” e crie um novo projeto (menu “File→New Project Wizard”) de acordo com os passos seguintes (Figuras 2 a 9).

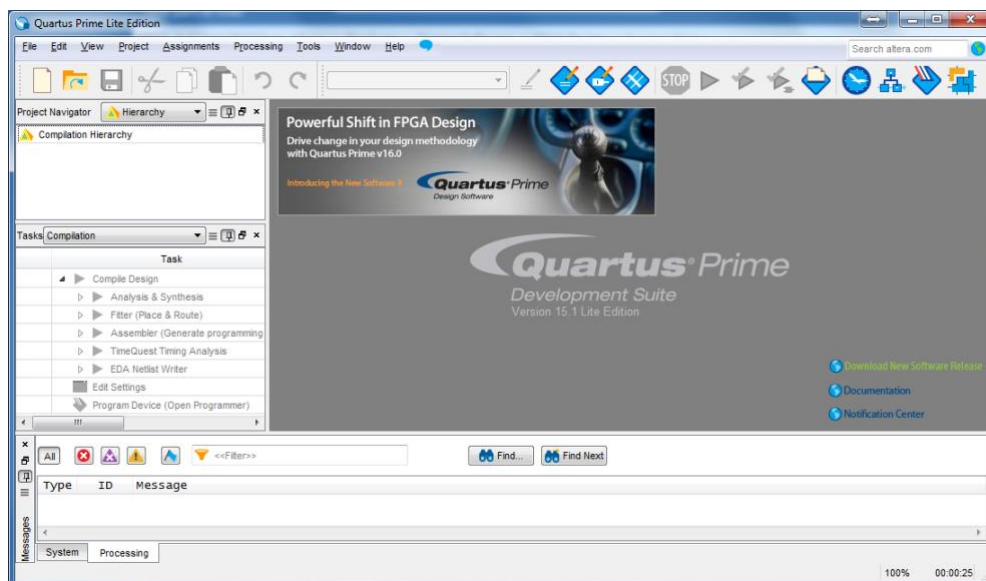


Figura 2 – Aspeto inicial da aplicação “Altera Quartus Prime” (sem qualquer projeto aberto).

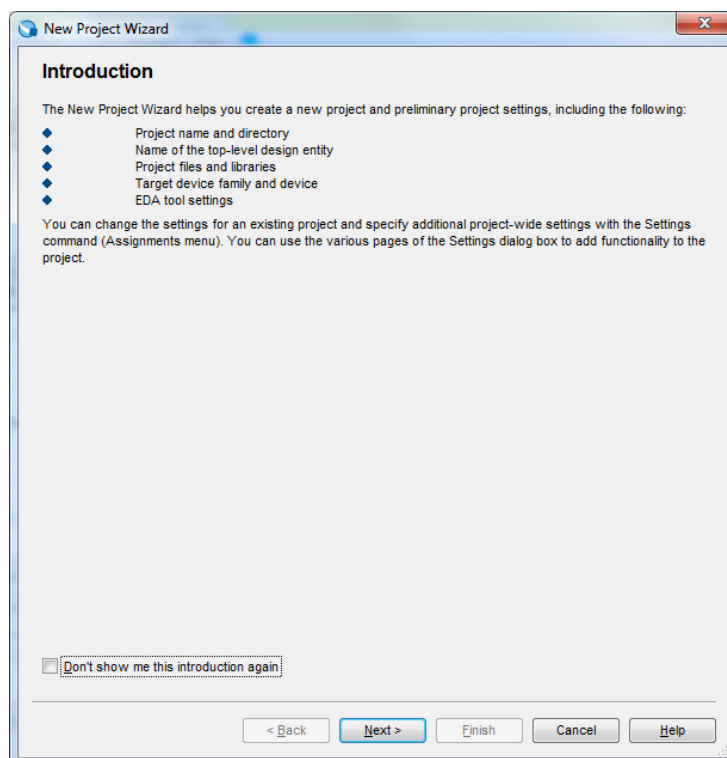


Figura 3 – Passo inicial introdutório (pode ser desativado).

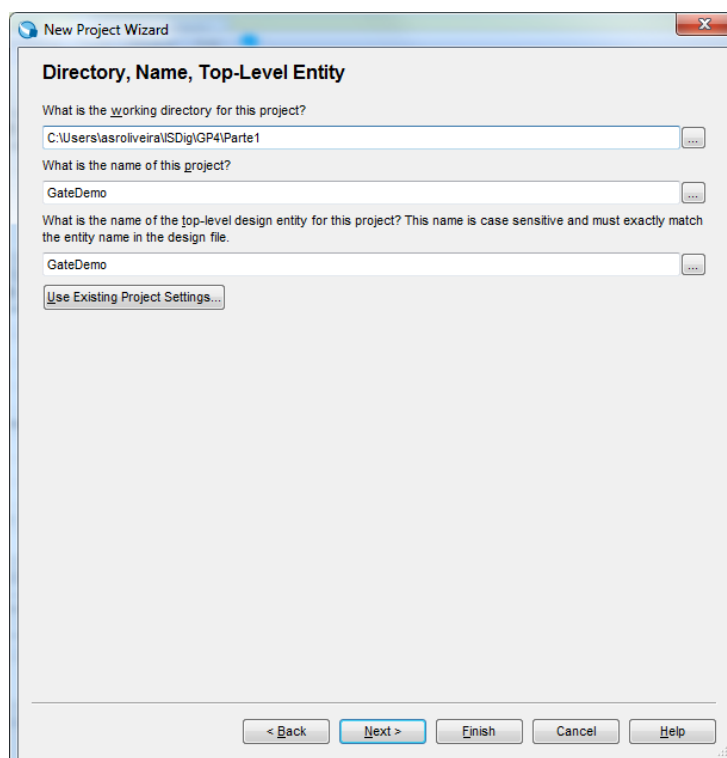


Figura 4 – Passo 1 – identificação e localização do projeto no sistema de ficheiros – adaptar de acordo com o diretório usado, o qual não pode conter no caminho (*path*) espaços nem caracteres especiais, e.g. acentos.

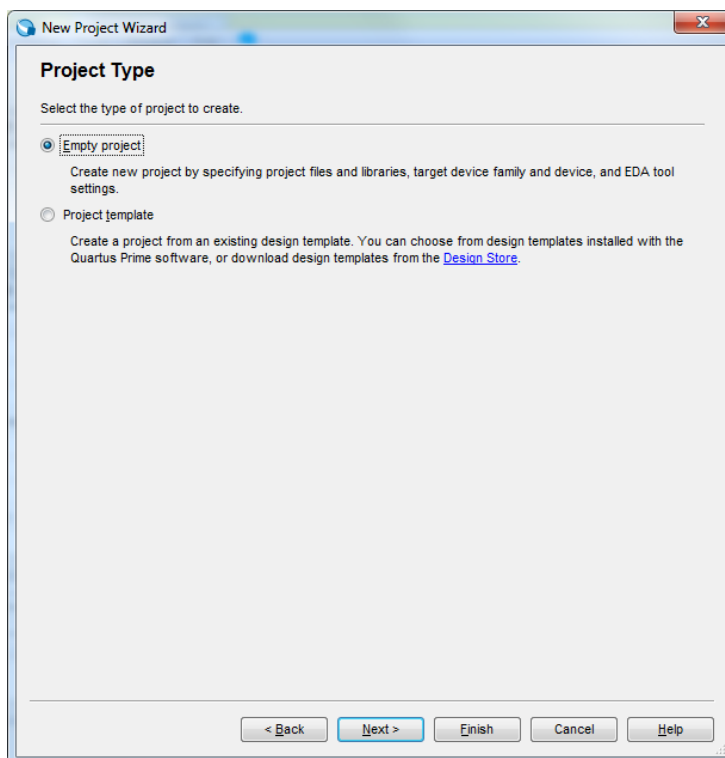


Figura 5 – Passo 2 – seleção do tipo de projeto a criar (projeto vazio neste caso).

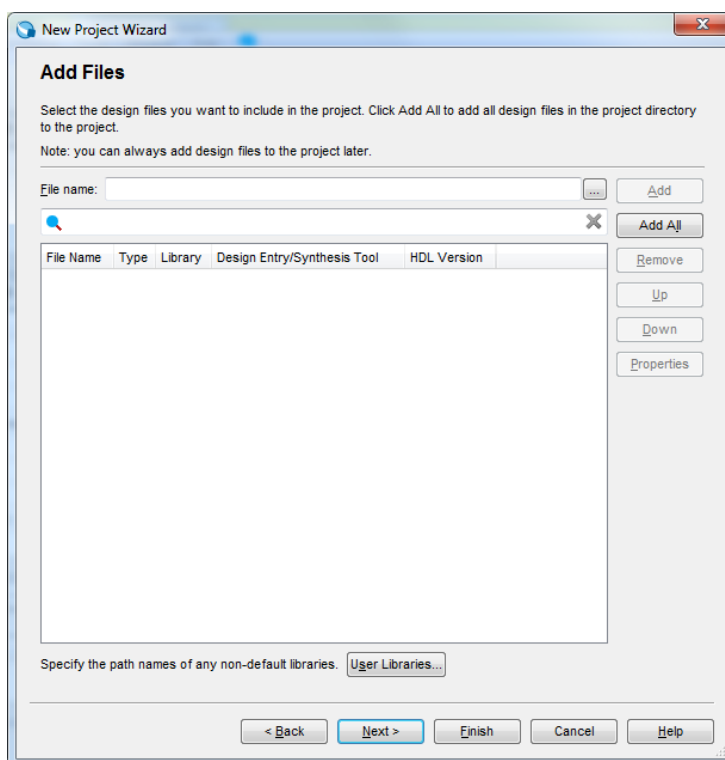


Figura 6 – Passo 2 – adição de ficheiros pré-existentes (não usado neste projeto).

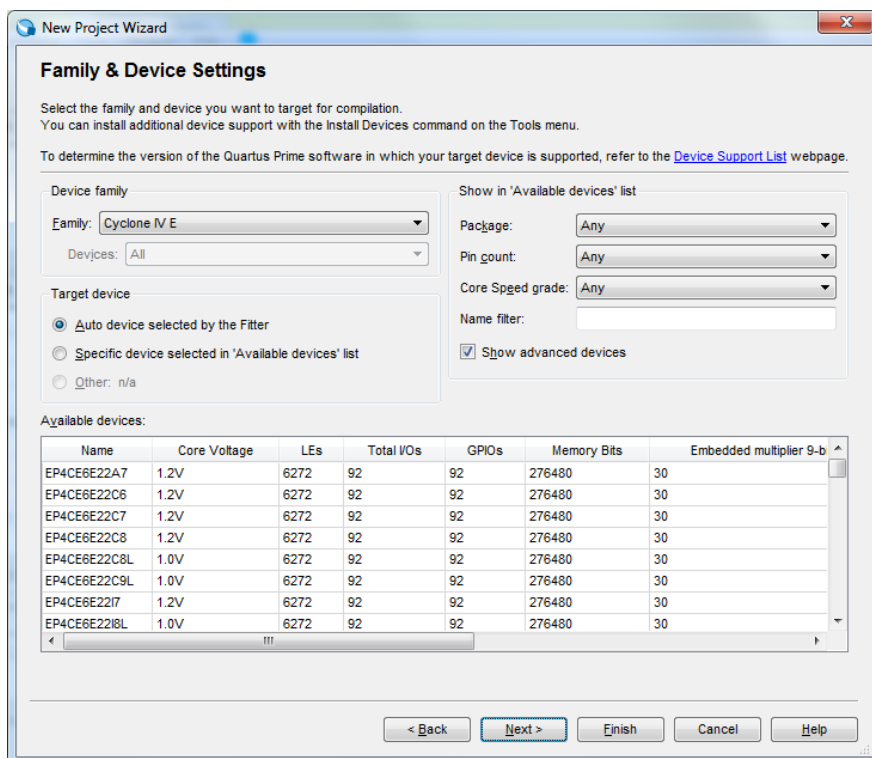


Figura 7 – Passo 3 – seleção do dispositivo (FPGA) usado na implementação (como se pretende apenas realizar a simulação comportamental, pode ser omitida a seleção de um dispositivo concreto).

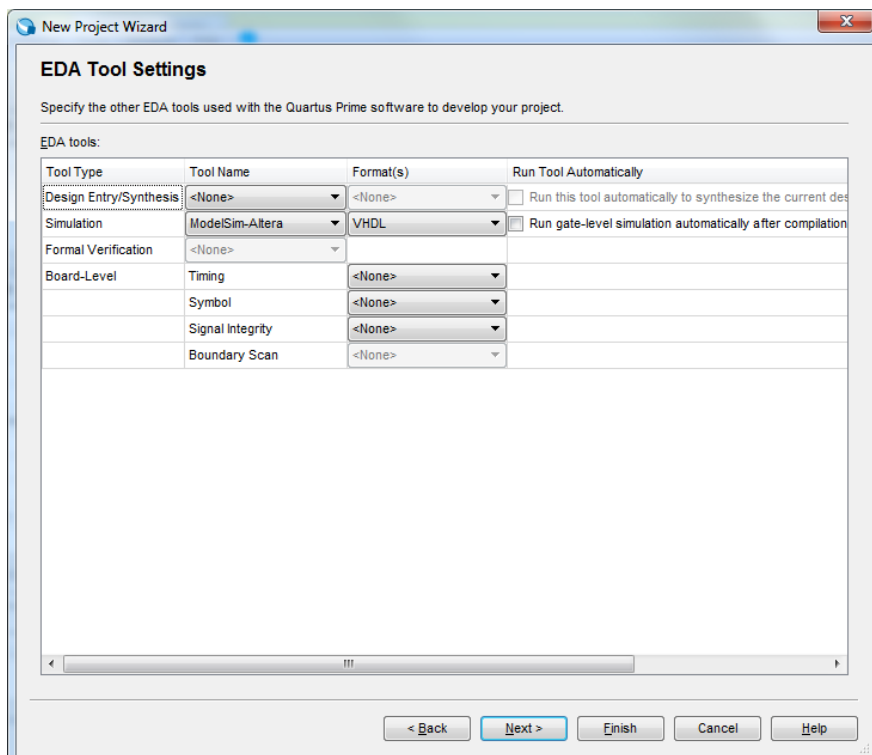


Figura 8 – Passo 4 – seleção das ferramentas e linguagens usadas no fluxo de projeto (valores por omissão).

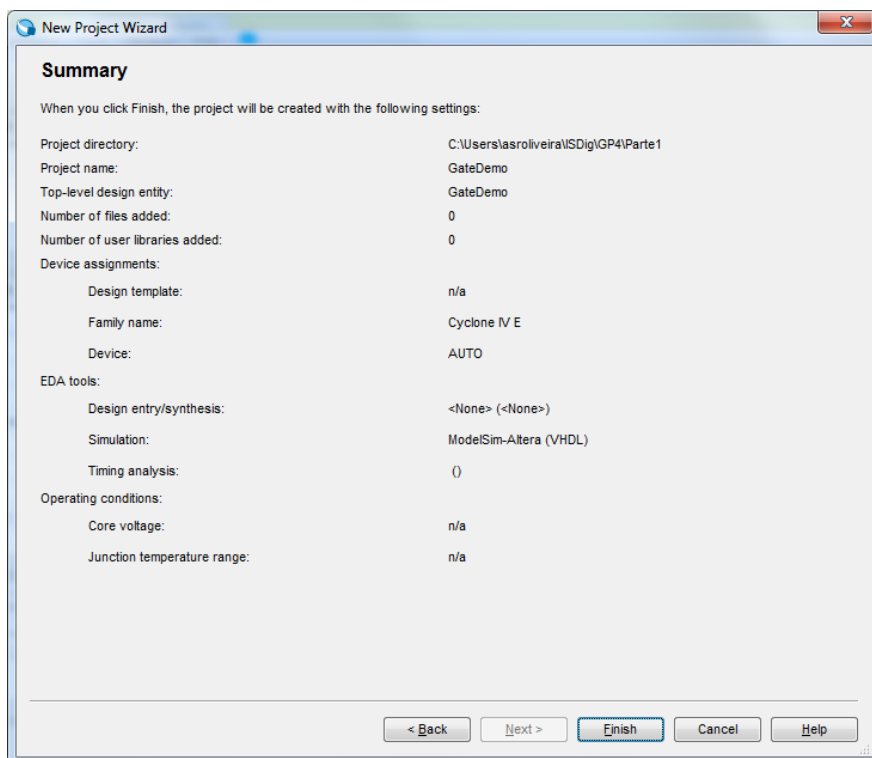


Figura 9 – Passo 5 – sumário final da criação do projeto.

2. Após premir “Finish” o IDE “Altera Quartus Prime” deve apresentar o aspeto da Figura 10.

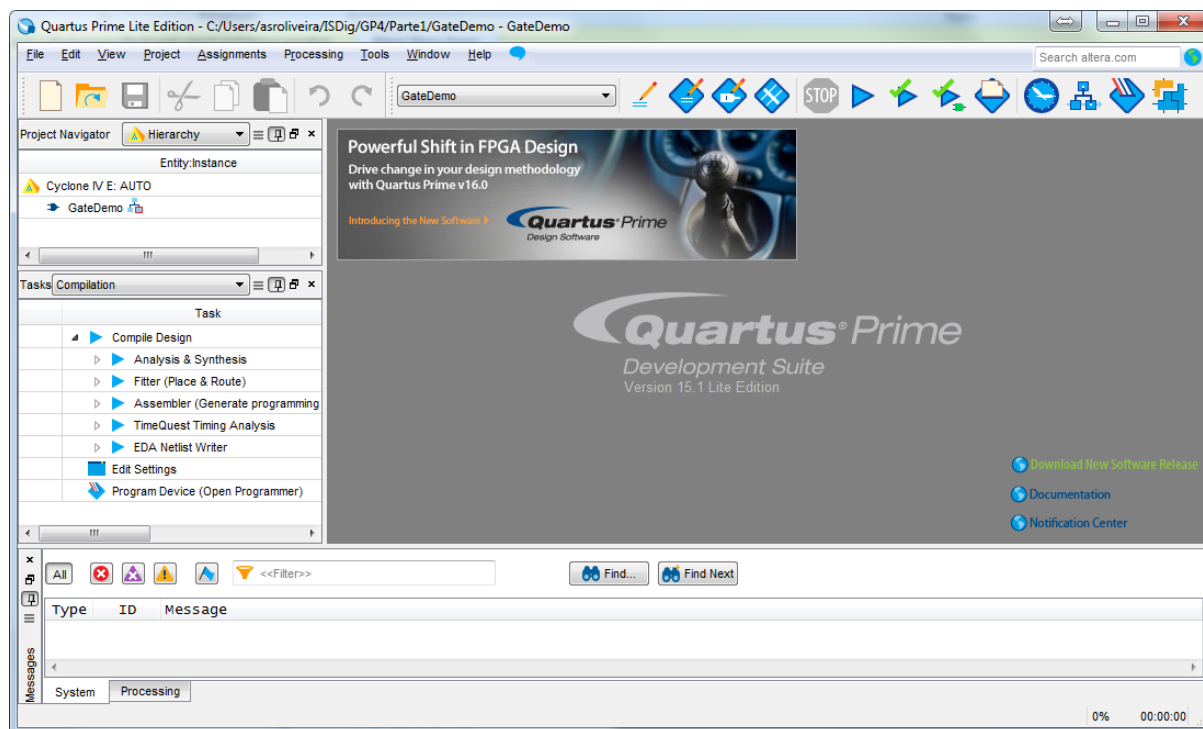


Figura 10 – Aspeto do “Altera Quartus Prime” IDE após a criação do projeto.

3. Crie um novo ficheiro para um diagrama esquemático (menu “File→New”), de acordo com a Figura 11.

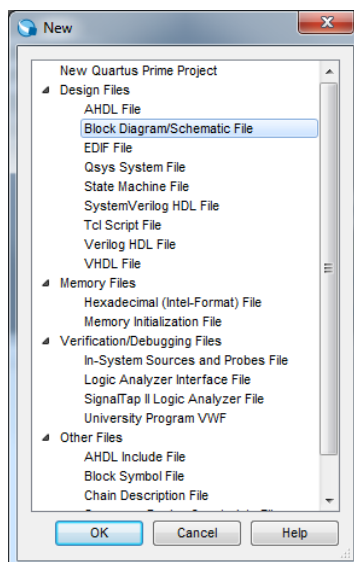


Figura 11 – Seleção do tipo de ficheiro a criar (*Block Diagram / Schematic File*).

4. Adicione uma porta lógica AND de 2 entradas usando o botão “*Symbol Tool*” da barra de ferramentas (Figura 12 a)) e escolhendo o componente de acordo com a Figura 12 c).

5. Adicione dois portos de entrada e um de saída usando o botão “*Pin Tool*” da barra de ferramentas (Figura 12 b)).

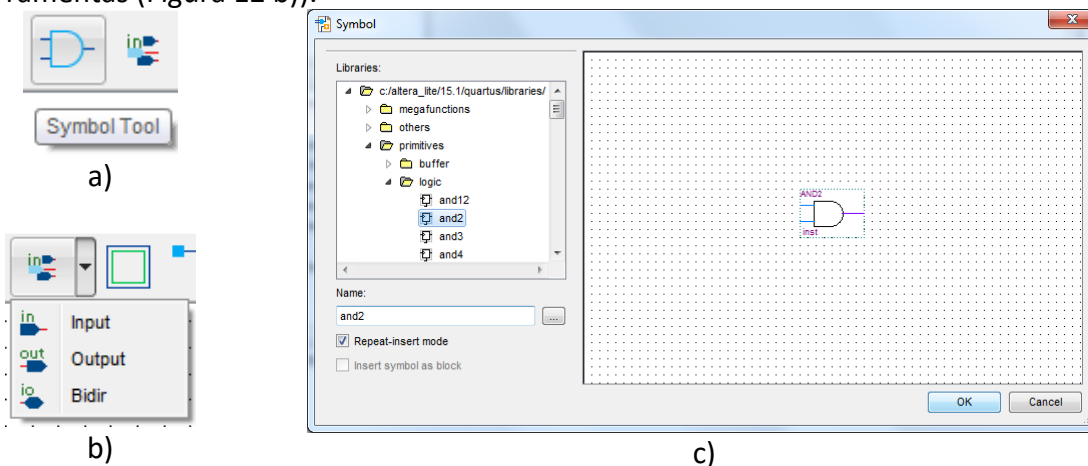


Figura 12 – Botões “*Symbol Tool*” e “*Pin Tool*” da barra de ferramentas e seleção da porta lógica AND na biblioteca de componentes do “*Altera Quartus Prime*”.

6. Interligue os portos de entrada, de saída e a porta lógica de acordo com a Figura 13. Identifique cada um dos elementos de acordo com os nomes apresentados na Figura 13.

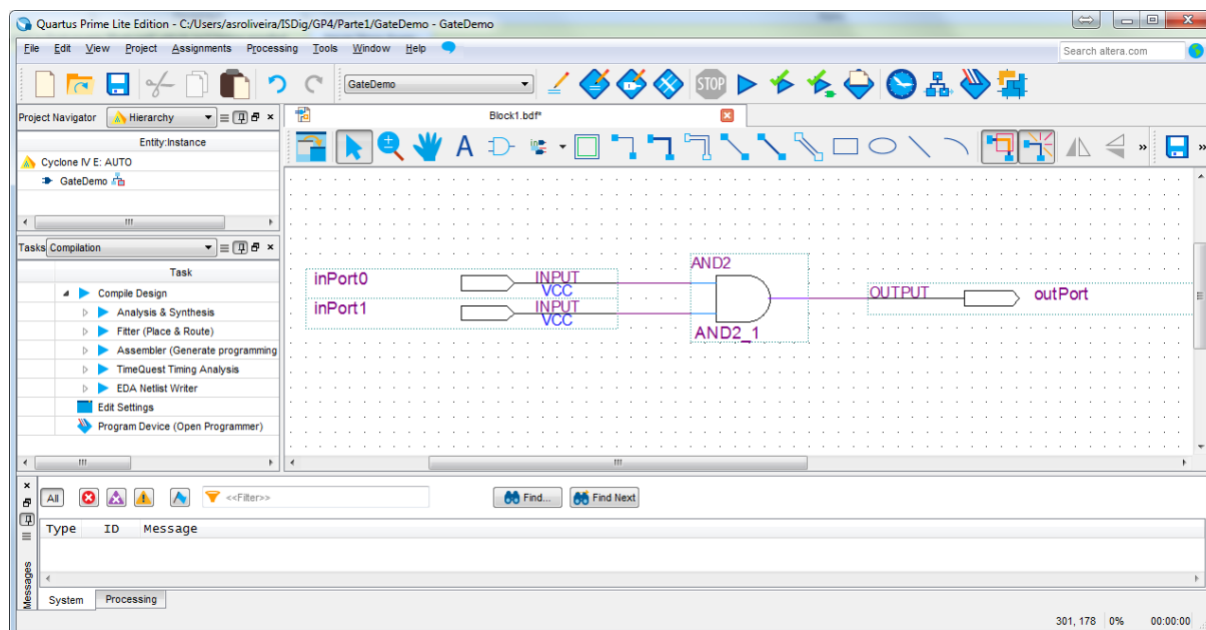


Figura 13 – Interligação da porta lógica AND e dos portos de entrada e de saída e identificação dos diversos elementos do circuito (para mudar o nome de um componente ou porto faça duplo clique sobre o nome atual).

7. Grave o ficheiro, cujo nome deverá ser “GateDemo.bdf” (Figura 14).

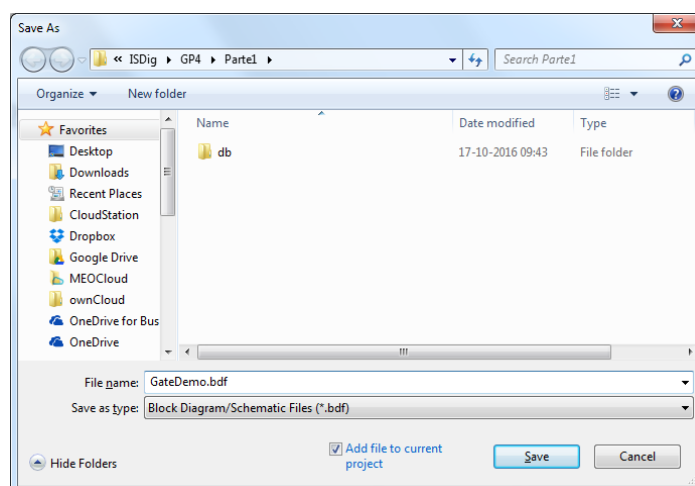


Figura 14 – Caixa de diálogo para gravação do ficheiro “GateDemo.bdf”.

8. Seguidamente vai ser validado por simulação o comportamento da porta lógica utilizada. No entanto, antes de efetuar a simulação, execute a opção “Analysis & Synthesis” para que seja analisada a correção estrutural do projeto. Após a execução da “Analysis & Synthesis” o IDE deve apresentar o aspeto da Figura 15.



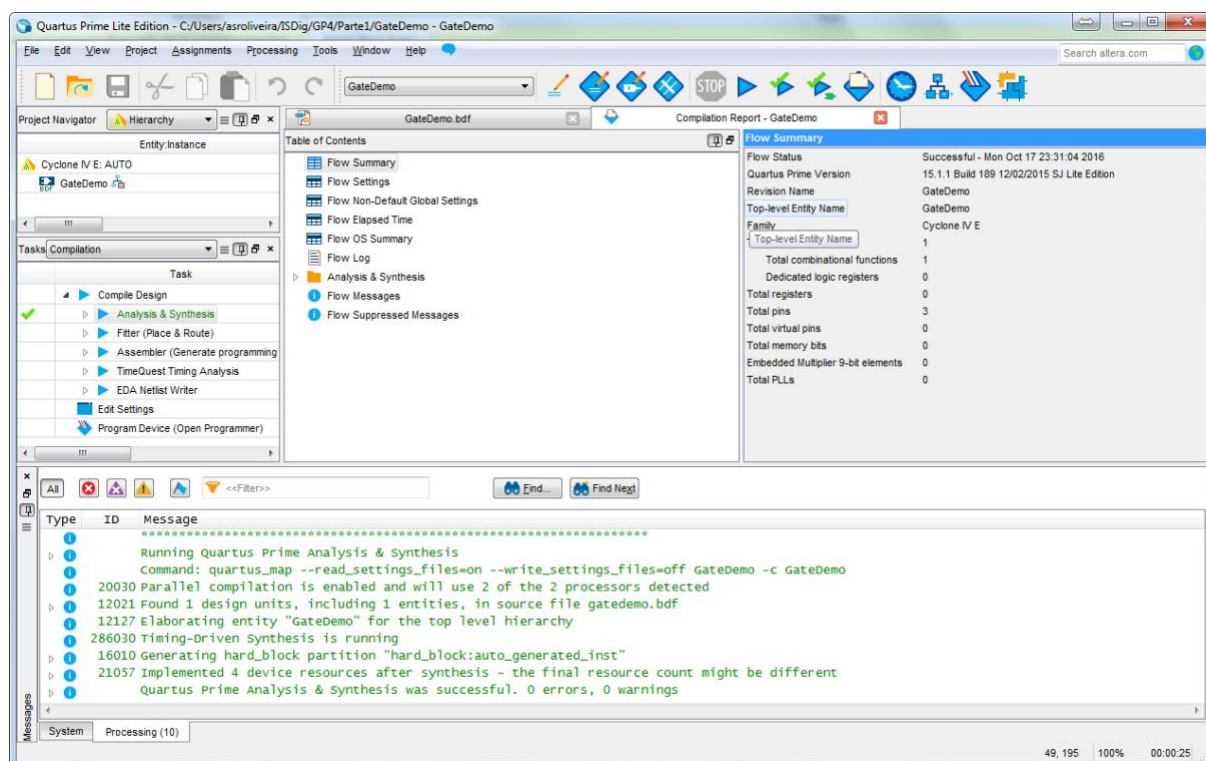


Figura 15 – “Altera Quartus Prime” IDE após a execução da “Analysis & Synthesis”.

9. Simule o comportamento da porta lógica, criando para tal um ficheiro VWF (menu “File→New”) de acordo com os passos descritos nos pontos seguintes (descritos nas Figuras 16 a 18).

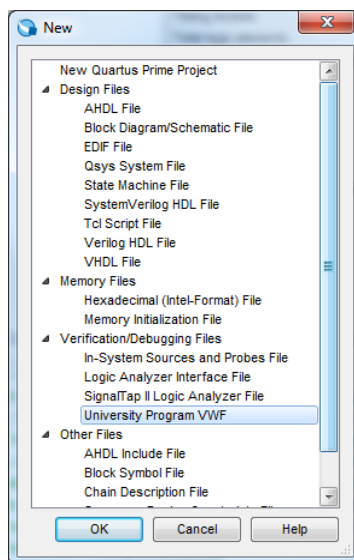


Figura 16 – Seleção do tipo de ficheiro a criar (“University Program VWF”).

10. Após premir “OK” a janela seguinte deverá abrir, onde deverão ser indicados os sinais a usar na simulação (Figura 17).

11. Através do menu “Edit→Insert→Insert Node or Bus”, premindo de seguida os botões “Node Finder” e “List”, selecione todos os portos de entrada e saída do circuito (Figuras 17 e 18).

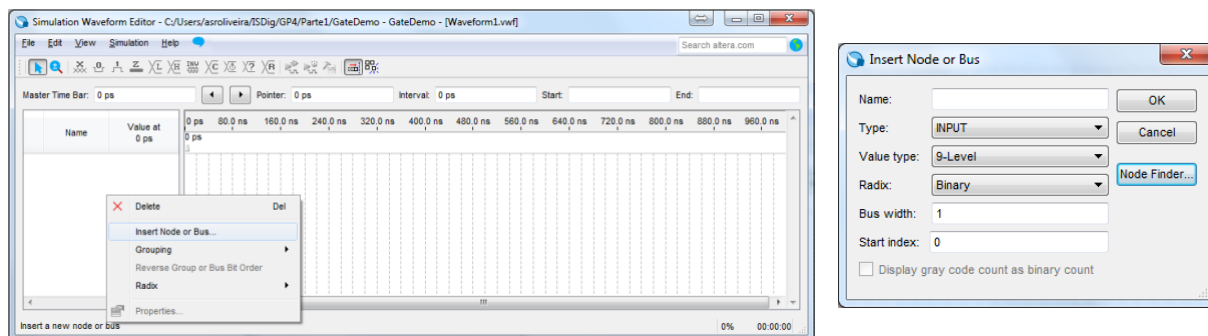


Figura 17 – Janela do simulador antes da especificação dos sinais de entrada e de saída a usar na simulação.

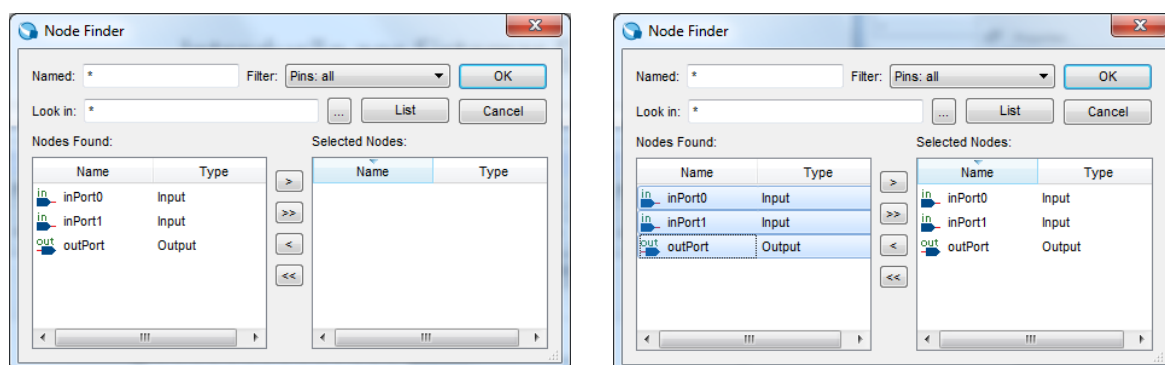


Figura 18 – Especificação dos sinais de entrada e de saída a usar na simulação.

**12.** Após premir “OK” poderá especificar os valores pretendidos para as entradas do circuito ao longo do tempo – o(s) valor(es) da(s) saída(s) será(ão) determinado(s) durante a simulação. Utilize para tal o rato, seleccionando os troços do diagrama temporal do sinal pretendido com o valor lógico que deseja que ele assuma (Figura 19).

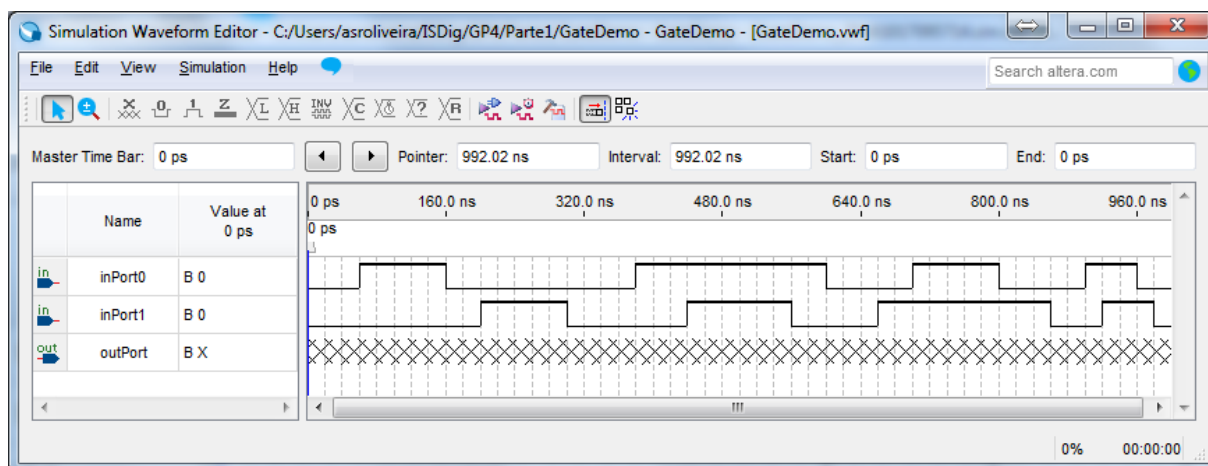


Figura 19 – Janela do simulador após a especificação das formas de onda dos sinais (vetores) de entrada.

**13.** Após especificar os valores (vetores) de simulação, grave o ficheiro com o nome “GateDemo.vwf” (Figura 20).

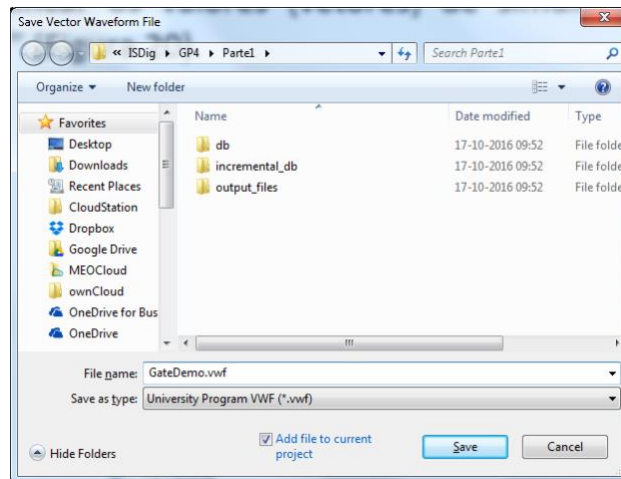


Figura 20 – Caixa de diálogo para gravação do ficheiro “GateDemo.vwf”.

14. Execute a simulação através do menu “*Simulation → Run Functional Simulation*”, o qual deve abrir uma janela semelhante à Figura 21. Após a simulação deve obter o valor da saída da porta lógica correspondente às entradas que especificou (Figura 22).

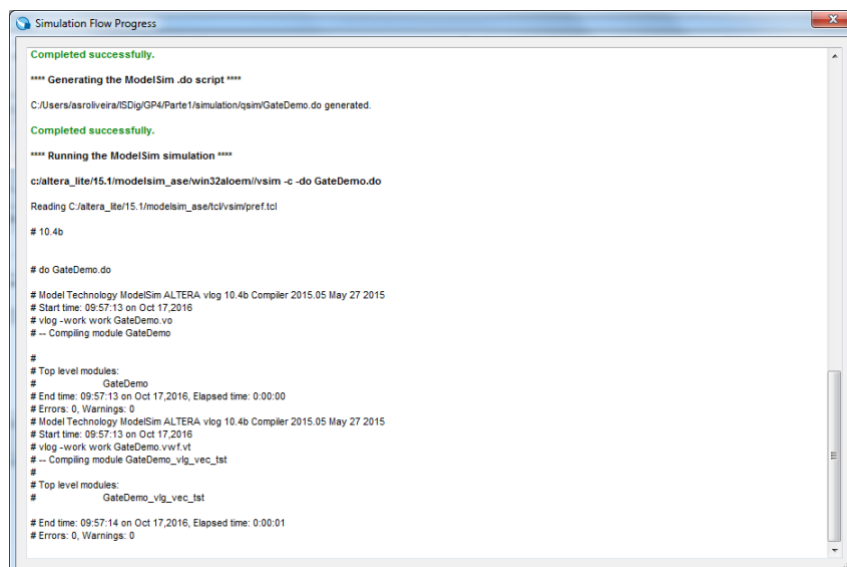


Figura 21 – Janela de compilação e execução da simulação do circuito com os vetores de entrada especificados.

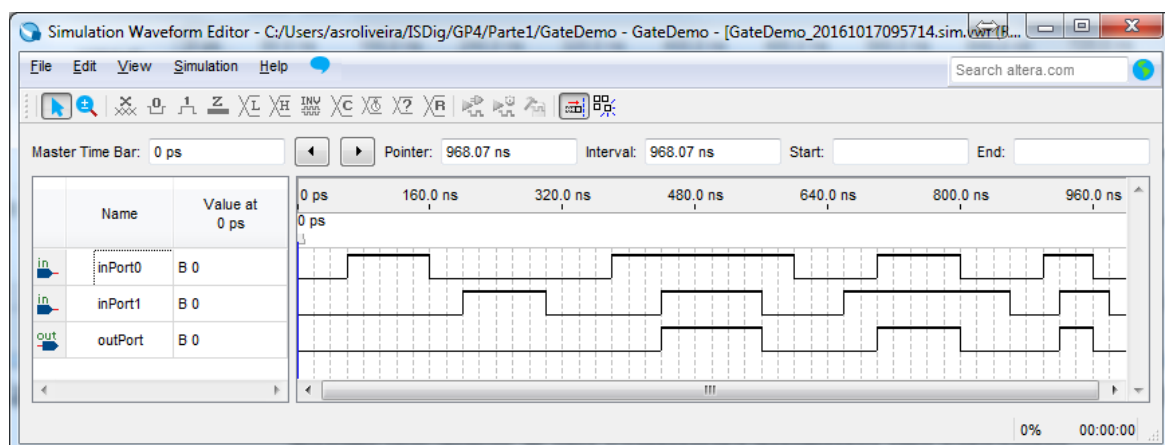


Figura 22 – Janela do simulador com a forma de onda do sinal de saída em função dos vetores de entrada especificados e do comportamento do circuito.

**15.** Adicione ao ficheiro “GateDemo.bdf” a instanciação de um inversor de forma construir uma porta lógica NAND (instanciar um componente, de uma biblioteca, significa usá-lo no contexto de um projeto). Realize a simulação comportamental do conjunto.

## Parte II

### Demonstração do projeto hierárquico

**1.** Crie um novo projeto, replicando os passos do ponto 1 da parte I deste guião. Designe o novo projeto e a respetiva entidade *top-level* como “EqCmpDemo”.

**2.** Crie um novo ficheiro para um diagrama esquemático, chamado “EqCmp4.bdf”, para implementar um comparador de igualdade de palavras de 4 bits (Figura 23). No final da edição do esquema lógico, o circuito deve ser semelhante ao da Figura 24. Utilize os componentes “xnor” e “and4” da biblioteca do “Altera Quartus Prime” acessível através do “Symbol Tool”.

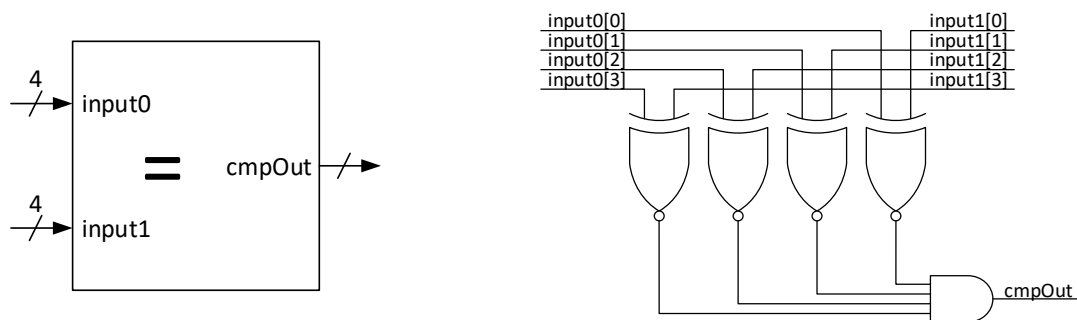


Figura 23 – Diagrama lógico do comparador de igualdade de palavras de 4 bits.

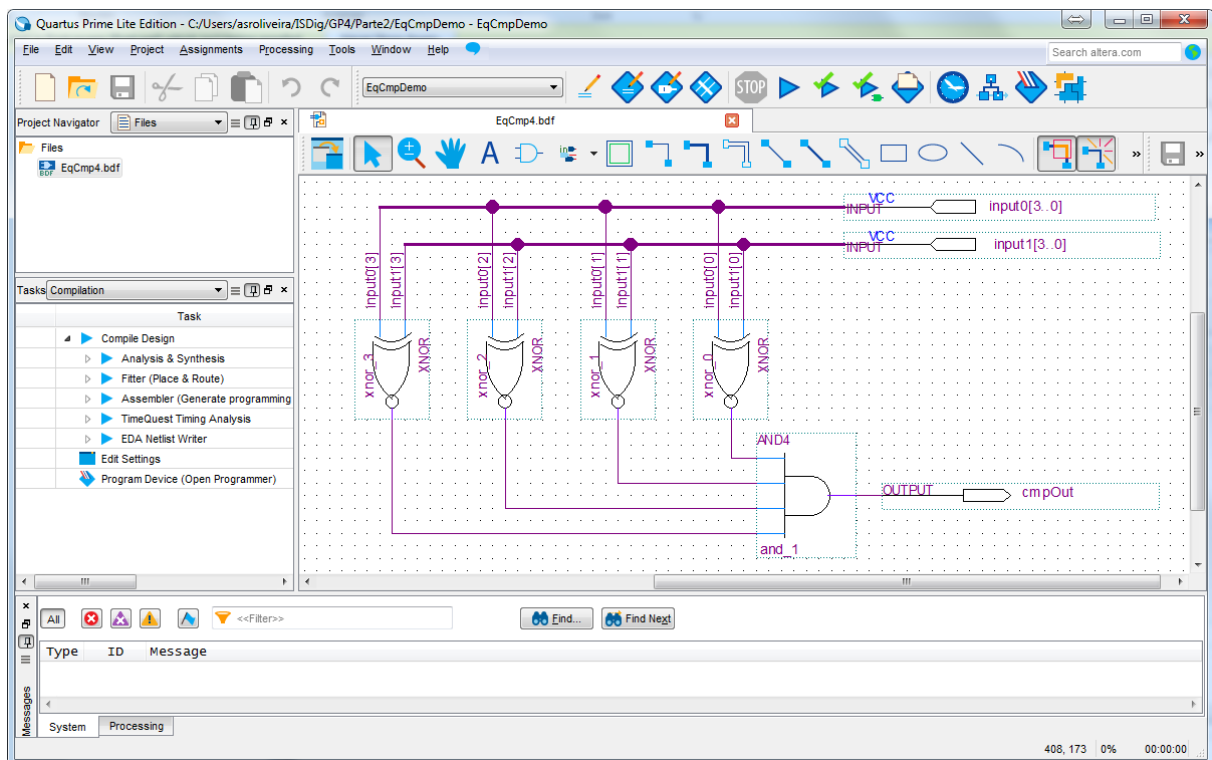


Figura 24 – Diagrama lógico do comparador de igualdade de palavras de 4 bits com interligação e identificação dos diversos elementos do circuito.

3. Crie um símbolo para o módulo “EqCmp4”, de forma a poder ser usado num diagrama esquemático, tal como ilustrado na Figura 25, e grave-o com o nome “EqCmp4.bsf” (Figura 26).

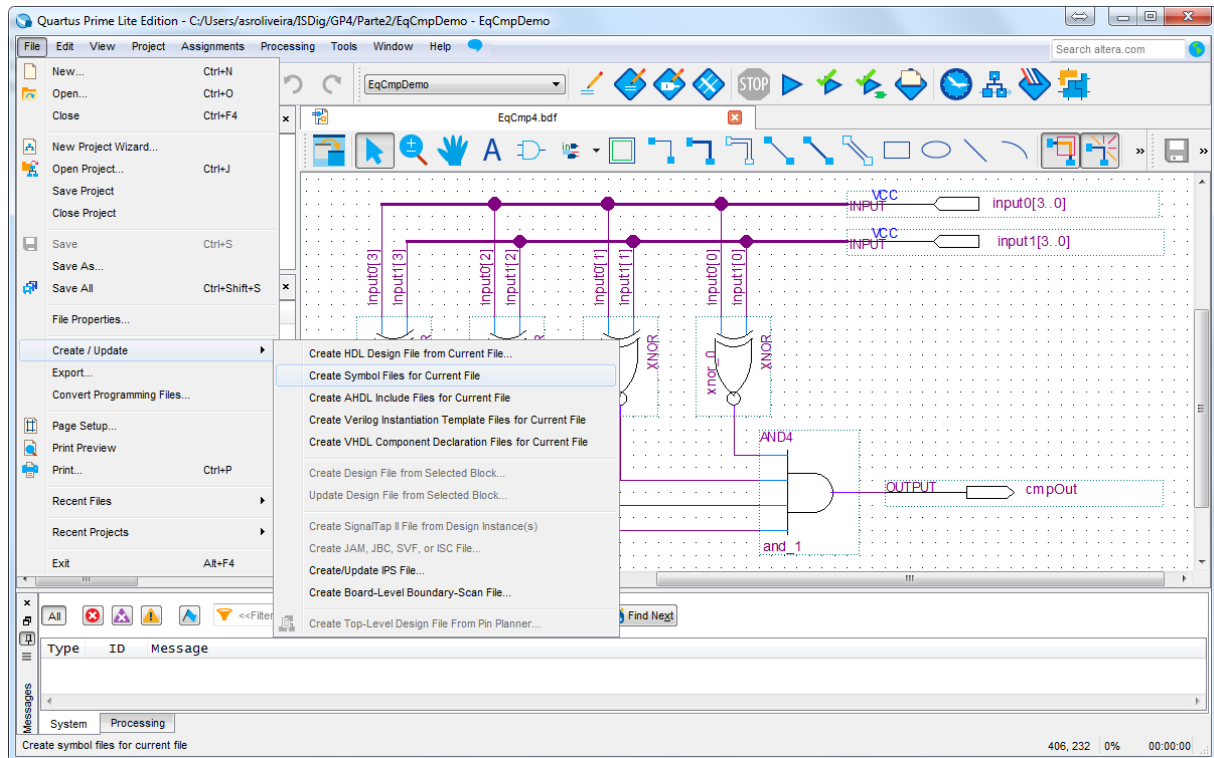


Figura 25 – Criação de um símbolo para o módulo “EqCmp4”.

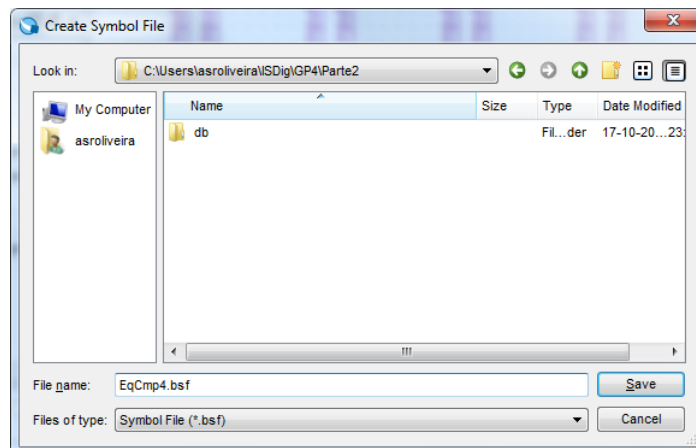


Figura 26 – Gravação do ficheiro “EqCmp4.bsf” relativo ao símbolo do módulo “EqCmp4”.

4. Crie um novo ficheiro para um diagrama esquemático, chamado “EqCmpDemo.bdf”, para instanciar o comparador de igualdade construído no ponto anterior (Figura 27) e efetuar a ligação dos seus portos, tal como ilustrado na Figura 28.

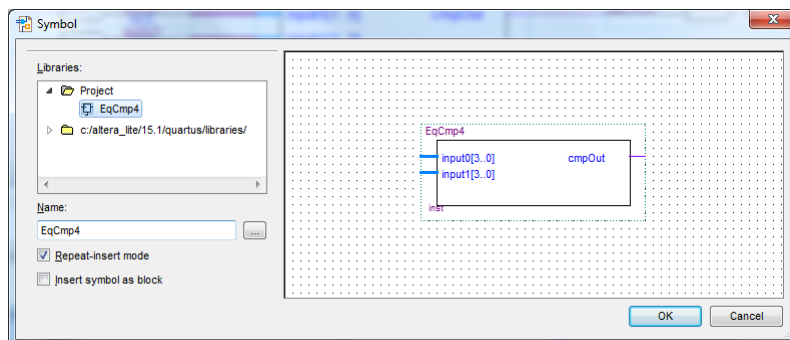


Figura 27 – Instanciação num diagrama esquemático do módulo “EqCmp4”.

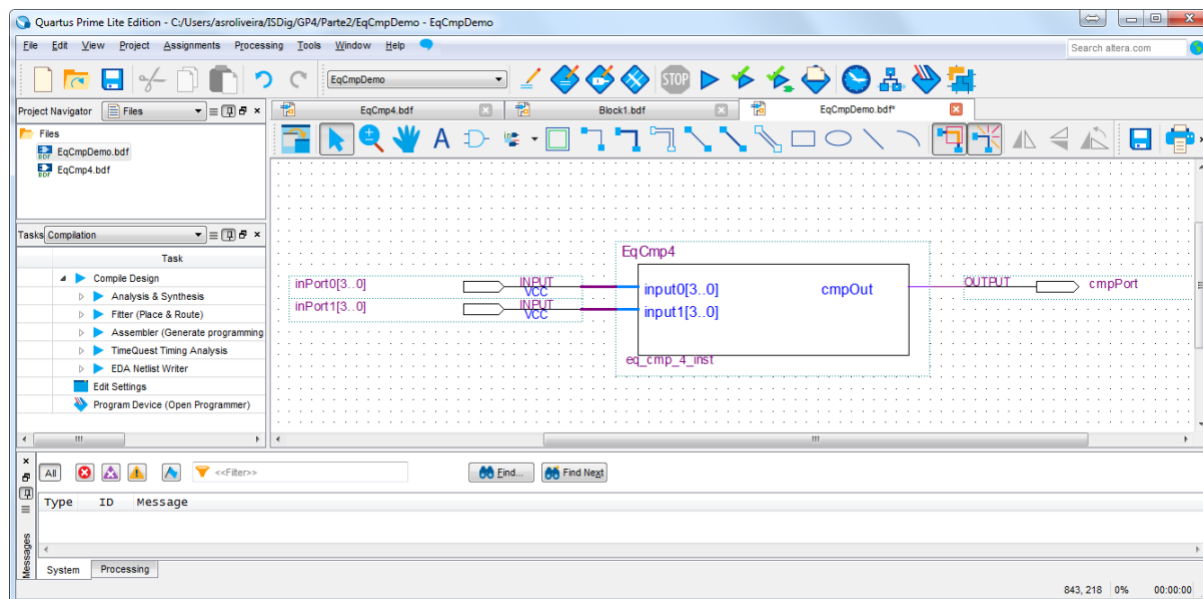


Figura 28 – Interligação do módulo “EqCmp4” e dos portos de entrada e de saída e identificação dos diversos elementos do circuito no módulo “EqCmpDemo”.

5. Efetue a simulação comportamental do comparador.

6. Contrua e simule um comparador de 16 bits a partir da instanciação de comparadores de 4 bits e no menor número possível de portas lógicas elementares.