Topics

- Introduction to schematic capture and behavioral simulation based on computer aided design tools
- Familiarization with the design tools that will be used in the practical classes.
- Capture of schematic diagrams (logic schemes) and behavioral simulation of elementary components.

Summary

This practical work is divided into 2 parts. In Part I we intend to introduce the basic aspects of the tools that will be used in the practical classes of Introduction to Digital Systems (ISD), based on a simple project using schematic capture and behavioral simulation. Part II discusses an example of a hierarchical project based on an equality comparator.

Summary of the Design Flow

Figure 1 illustrates the typical design flow of digital systems, based on reprogrammable logic devices, such as Field Programmable Gate Arrays (FPGAs). In ISD only the initial steps are covered. The rest will be the subject of study in the subsequent course "Digital Systems Laboratory - LSD".

The design entry stage consists of modeling, coding, or introducing the intended functionality. For such purposes the designer may use hardware description languages, schematic diagrams, state transition diagrams, or other methods. In the case of ISD, the capture of schematic diagrams and/or state transition diagrams will be used, depending on which one is more appropriate for each system. For convenience, the Integrated Development Environment (IDE) "Quartus Prime" will be used, which has schematic and state diagram editors, as well as other tools needed to perform all steps of the project flow.

Once the system has been modeled, the next step is its synthesis (logic), i.e. the compilation of the model to create a netlist (i.e. a set of logic gates, flip-flops, multiplexers, other components and their interconnections) that implements the desired functionality. This step is performed by software tools commonly developed by the manufacturer of the implementation technology used.

System validation can be performed at various stages of the project flow, either by simulation or real hardware verification. Various tools for time analysis, energy analysis, logic resources usage, etc. are also normally available from the manufacturer. In this practical work only behavioral simulation will be performed.

Important Notes: Due to network speed limitations in accessing the personal directory on arca.ua.pt, on classroom PCs it is recommended to use a USB stick to store your projects and files. If this is not possible, save and access your work from a directory of drive Z :.

In either case do not use spaces or special characters (e.g. accents) in the project and file paths as this will cause problems with the use of the tools (this means you should not save your projects into sub directories of "Desktop" or "My Documents").

Tip: In Windows create, for example in C:\User>\<User>, a directory structure to save your ISD projects (e.g. C:\Users\<User>\ISD\P5\Part1\<Project Name>); on Linux you can do so at /home/<user>/ISD/P5/Part1<<Project Name>).

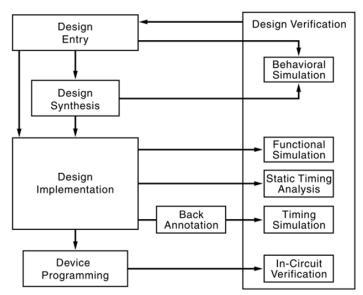


Figure 1 - Design flow for FPGA-based digital systems (source: <u>www.xilinx.com</u>).

Part I Demonstration of key steps in project flow based on schematic capture and behavioural simulation

1. Open the "Quartus Prime" application and create a new project (menu "File \rightarrow New Project Wizard") according to the following steps (Figures 2 to 9).



Figure 2 - Initial aspect of the "Quartus Prime" application (without any project opened).

2022-2023

New Proj	ect Wizard
Introdu	uction
The New F	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 too settings
	hange the settings for an existing project and specify additional project-wide settings with the Settings (Assignments menu). You can use the various pages of the Settings dialog box to add functionality to the
Don't s	how me this introduction again
	< <u>B</u> ack <u>N</u> ext > Einish Cancel <u>H</u> elp

Figure 3 - Introductory initial step (can be disabled).

Directory, Name, To	p-Level Enti	ty			
What is the working directory for	or this project?				
C:\Users\asroliveira\ISDig\GP4\	Parte1				
What is the name of this project	?				
GateDemo					
What is the name of the top-leve the entity name in the design file		this project? This	name is case sen	sitive and must ex	actly match
GateDemo					
Use Existing Project Settings]				

Figure 4 - Step 1 - Identification and location of the project in the file system - adapt according to the directory used, which cannot contain spaces or special characters, e.g. accents.

Project Type					
Select the type of projec	t to create.				
Empty project					
Create new project settings.	by specifying project fi	les and libraries, tar	get device family	and device, and E	DA tool
Project template					
	m an existing design ter vare, or download desi			emplates installed	with the

Figure 5 - Step 2 - Select the type of project to create (empty project in this case).

						Add
•					×	Add All
File Name	Туре	Library	Design Entry/Synthesis Tool	HDL Version		<u>R</u> emove
					[<u>U</u> p
					[<u>D</u> own
					[<u>P</u> roperties

Figure 6 - Step 2 - Adding pre-existing files (not used in this project).

You can install a	and device you want t dditional device suppor version of the Quartus	rt with the In	stall Devices com				to the [Device Support List wel	opage	
Device family				Show in !	Available (levices' list				
Eamily: Cyclor	ne IV E		•	Package:		Any		•		
Devices: A						Any	_		-	
					ed grade:	_			•	
Target device				Core Spe	ed grade.	Any			_	
Other: n/a										
_		LEs	Total VOs	GPIOs	Mer	norv Bits		Embedded multiplier 9	-bi ^	
A <u>v</u> ailable devices Name	Core Voltage	LEs 6272	Total VOs 92	GPIOs 92	Mer 276480	nory Bits	30	Embedded multiplier S	-bi ^	
Available devices Name EP4CE6E22A7	Core Voltage						30	Embedded multiplier S	-b ^	
Available devices Name EP4CE6E22A7 EP4CE6E22C6	Core Voltage 1.2V	6272	92	92	276480			Embedded multiplier 9	-b ^	
Available devices Name EP4CE6E22A7 EP4CE6E22C6 EP4CE6E22C7	Core Voltage 1.2V 1.2V	6272 6272	92 92	92 92	276480 276480	•	30	Embedded multiplier S	-b ^	
Available devices Name EP4CE6E22A7 EP4CE6E22C6 EP4CE6E22C7 EP4CE6E22C8	Core Voltage 1.2V 1.2V 1.2V	6272 6272 6272	92 92 92 92	92 92 92 92	276480 276480 276480		30 30	Embedded multiplier S	-b ^	
Available devices Name EP4CE6E22A7 EP4CE6E22C6 EP4CE6E22C7 EP4CE6E22C8 EP4CE6E22C8	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V	6272 6272 6272 6272 6272	92 92 92 92 92	92 92 92 92 92	276480 276480 276480 276480		30 30 30	Embedded multiplier S	-b ^	
Available devices Name EP4CE6E22A7 EP4CE6E22C6 EP4CE6E22C7 EP4CE6E22C8 EP4CE6E22C8L EP4CE6E22C9L	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.2V 1.0V	6272 6272 6272 6272 6272 6272	92 92 92 92 92 92 92	92 92 92 92 92 92 92	276480 276480 276480 276480 276480 276480		30 30 30 30	Embedded multiplier S	-b ^	
Available devices Name EP4CE6E22A7 EP4CE6E22C6 EP4CE6E22C7 EP4CE6E22C8	Core Voltage 1.2V 1.2V 1.2V 1.2V 1.2V 1.0V 1.0V 1.0V 1.0V	6272 6272 6272 6272 6272 6272 6272	92 92 92 92 92 92 92 92 92	92 92 92 92 92 92 92 92 92	276480 276480 276480 276480 276480 276480 276480		30 30 30 30 30 30	Embedded multiplier S	-b ^	

Figure 7 - Step 3 - Selection of the device (FPGA) used in the implementation (as the only behavioural simulation is intended, the selection of a concrete device can be omitted).

	DA tools:				
Simulation ModelSim-Altera VHDL Run gate-level simulation automatically after compilate Formal Verification <none> Board-Level Timing <none> Symbol <none> Signal Integrity <none></none></none></none></none>			Format(s)		Run Tool Automatically
Formal Verification Alone>	Design Entry/Synthesis	<none></none>	<none></none>		Run this tool automatically to synthesize the current d
Board-Level Timing <none> Symbol <none> Signal Integrity <none></none></none></none>	Simulation	ModelSim-Altera 🔻	VHDL		Run gate-level simulation automatically after compilation
Symbol <none> Signal Integrity <none></none></none>	Formal Verification	<none> -</none>]		
Signal Integrity	Board-Level	Timing	<none></none>	•	
		Symbol	<none></none>	-	
Boundary Scan		Signal Integrity	<none></none>	•	
		Boundary Scan	<none></none>	-	
			<u>, </u>		
			Δ	1	

Figure 8 - Step 4 - Selection of the tools and languages used in the project flow (use default values).

Summary When you click Finish, the project will be created with the followi	
When you click Einish, the project will be created with the followi	
when you click rimsh, the project will be created with the followi	ing settings:
Project directory:	C:\Users\asroliveira\ISDig\GP4\Parte1
Project name:	GateDemo
Top-level design entity:	GateDemo
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Design template:	n/a
Family name:	Cyclone IV E
Device:	AUTO
EDA tools:	
Design entry/synthesis:	<none> (<none>)</none></none>
Simulation:	ModelSim-Altera (VHDL)
Timing analysis:	0
Operating conditions:	
Core voltage:	n/a
Junction temperature range:	n/a
	Back Next > Einish Cancel Help

Figure 9 - Step 5 - Final summary of project creation.

2. After pressing "Finish" the "Quartus Prime" IDE should look like Figure 10.



Figure 10 - Aspect of the "Quartus Prime" IDE after project creation.

3. Create a new file for a schematic diagram ("File \rightarrow New" menu) according to Figure 11.

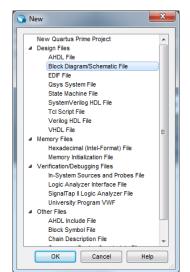


Figure 11 - Selection of the file type to create (Block Diagram / Schematic File).

4. Add a 2-input AND logic gate using the "Symbol Tool" toolbar button (Figure 12 a)) and choosing the component according to Figure 12 c).

5. Add two input ports and one output port using the "Pin Tool" toolbar button (Figure 12 b)).

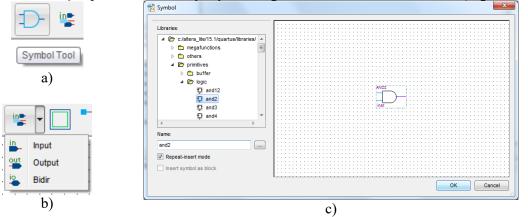


Figure 12 - "Tool Tool" and "Pin Tool" toolbar buttons and AND logic port selection in the Quartus Prime component library.

6. Link the input, output, and logical ports according to Figure 13. Identify each of the elements according to the names shown in Figure 13.

le <u>E</u> dit <u>V</u> iew <u>P</u> roject <u>A</u> ssignments P <u>r</u> ocessin	ig/GP4/Parte1/GateDemo - GateDemo		Search altera.com
D 🗖 🖶 🗲 D 💼 🤈	C GateDemo 🖌 🎸 🎸 🗞	⑩ ⊳ ≁ 🗲 🔶 🚫 🛔 🌺	
nject Navigator 🔥 Hierarchy 🔻 🗏 🛱 🗴	Block1.bdf*		
Entity:Instance	🚰 💽 🔍 👋 A 🕩 🖷 - 🔲 "ר "	$] \land \land \land \\ \blacksquare \bigcirc \land \land $	₽ ₩ ▲
GateDemo			
sks Compilation ▼ ≡ I I I × Task		AND2	
Compile Design Analysis & Synthesis	InPort0 INPUT		⊃ outPort
	VCC	AND2_1	
Edit Settings			
Program Device (Open Programmer)	· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·	
	💏 Find 🥅 Find Next		4
Type ID Message			
System Processing			

Figure 13 - Interconnection of AND logic port and input and output ports and identification of the various circuit elements (to rename a component or port, double click on the current name).

7. Save the file, whose name should be "GateDemo.bdf" (Figure 14).

					×
Save As					
🔾 🖉 🖉 🖉 🖉	Dig 🕨	GP4 ► Parte	1 •	✓ Search Pail	rte1 🔎
Organize 🔻 🛛 Ne	w fold	ler			≣ ▾ 🔞
🔆 Favorites	-	Name	<u>^</u>	Date modified	Туре
E Desktop Downloads CloudStation CloudStation Dropbox Coogle Drive MEOCloud ownCloud C OneDrive for		🄑 db		17-10-2016 09:43	File folder
🝊 OneDrive	-	•	III		•
File <u>n</u> ame:	Gate	Demo.bdf			•
Save as <u>t</u> ype:	Block	c Diagram/Sche	ematic Files (*.bdf)		•
lide Folders			Add file to cu project	rrent Save	Cancel

Figure 14 - Dialog for saving the GateDemo.bdf file.

8. Now the behaviour of the logic gate used will be validated by simulation. However, before performing the simulation, execute the "Analysis & Synthesis" option to analyse the structural correction of the project. After running Analysis & Synthesis the IDE should look like Figure 15.

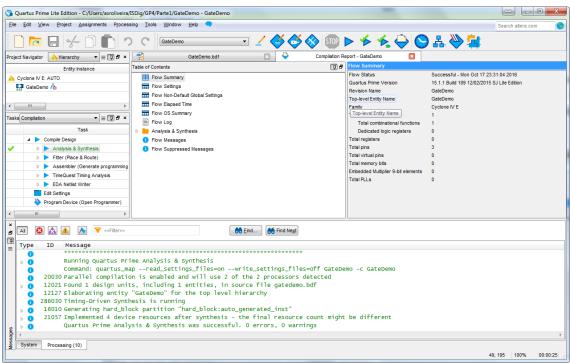


Figure 15 - "Quartus Prime" IDE after running "Analysis & Synthesis".

9. Simulate the behaviour of the logical port by creating a VWF file ("File \rightarrow New" menu) according to the steps described in the following points (described in Figures 16 to 18).

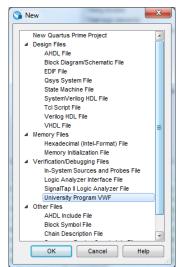


Figure 16 - Selection of file type to create ("University Program VWF").

10. After pressing "OK" the next window should open, where the signals to be used in the simulation should be indicated (Figure 17).

11. Through the "Edit \rightarrow Insert \rightarrow Insert Node or Bus" menu, then pressing the "Node Finder" and "List" buttons, select all input and output ports of the circuit (Figures 17 and 18).

🕥 Sin	nulation Wav	eform	Editor - C:/	Users/asi	oliveira/I	SDig/GP4	/Parte1/Gat	eDemo - G	steDemo - [Waveform1	.vwf]					- O <mark>- X</mark>				
-	Edit ⊻iew	-												s	earch altera.	com	• ••••			
	9 🔍 🖄 🖞	5 Å) <u>8</u>) <u>?</u>	$\langle \overline{B} \mid \mathbb{R}^{D}_{n}$	R 2a (= B <u>K</u>									🕥 Insert No	de or Bus		
Maste	r Time Bar: 0) ps				Pointer: 0	ps		Interval: 0 p	s		Start:		En	d:		Name:			ОК
	Name		Value at	1	80.0 ns	160.0 ns	240.0 ns	320.0 ns	400.0 ns	480.0 ns	560.0 ns	640.0 ns	720.0 ns	800.0 ns	880.0 ns	960.0 ns	Туре:	INPUT	-	Cancel
			0 ps	0 ps													Value type:	9-Level	•	Canoor
																	Radix:	Binary	•	Node Finder
		×	Delete			Del	1										Bus width:	1		
			Insert Nod	e or Bus													Start index:			
			Grouping Reverse G		- 83.0-4															
			Reverse G	roup of B	is bit Uro	•											Display g	gray code count as l	omary count	a
•		P	Properties							III						- + v				
Insert	a new node o	r bus													0%	00:00:00				

Figure 17 - Simulator window before specifying the input and output signals to be used in the simulation.

🕤 Node Finder			X	Node Finder				×
Named: *	Filter: Pins	all	• ОК	Named: *	Filte	r: Pins	all	• ОК
Look in: *		List	Cancel	Look in: *			List	Cancel
Nodes Found:	:	Selected Nodes:		Nodes Found:			Selected Nodes:	
Name	Туре	Name	Туре	Name	Туре	>	Name	Туре
inPort0 Inp	out 📃			ininPort0	Input	\equiv	inPort0	Input
inPort1 Inp	out >>			in inPort1	Input	>>	inPort1	Input
outPort Ou	itput <			out outPort	Output	<	outPort	Output
	<<					<<		
٠ III	•	٠ III	▶	•	4 III		•	4

Figure 18 - Specification of input and output signals to be used in the simulation.

12. After pressing "OK" you can specify the desired values for the circuit inputs over time - the output value (s) will be determined during the simulation. Use the mouse to select the time-frame sections of the desired signal with the logical value you want it to assume (Figure 19).

ile	<u>E</u> dit <u>V</u> iew	Simulation <u>H</u> elp	o 🤜				Search alte	era.com
	e 🐹 e		E 133 XC X3 X7 X8 I	🗞 式 🚈 📾 🐘				
_	Time Bar: 0		Pointer: 992		992.02 ns	Start: 0 ps	End: 0	ps
	Name	Value at 0 ps	0 ps 160.0 ns 0 ps	320.0 ns	480.0 ns	640.0 ns	800.0 ns	960.0 ns
in_	inPort0	В 0						
in	inPort1	в 0						
out ●	outPort	BX	*****	******	******	******	*******	******

Figure 19 - Simulator window after specifying the input signal waveforms (vectors).

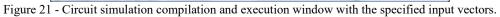
13. After specifying the simulation values (vectors), save the file with the name "GateDemo.vwf" (Figure 20).

🖉 🖉 🖌 🖉 SDig] ► GP4 ► Parte1 ►	earch Parte1	Q
Organize 🔻 New	folder		0
🔆 Favorites	A Name	Date modified	Туре
🧮 Desktop	🔒 db	17-10-2016 09:52	File folde
鷆 Downloads	incremental_db	17-10-2016 09:52	File folde
Recent Places CloudStation	output_files	17-10-2016 09:52	File folde
😌 Dropbox			
👠 Google Drive			
👗 MEOCloud			
鷆 ownCloud			
🐔 OneDrive for Bu	s		
🝊 OneDrive	* • III		Þ
File <u>n</u> ame:	GateDemo.vwf		•
Save as <u>t</u> ype:	Iniversity Program VWF (*.vwf)		•
	Add file to current		

Figure 20 - Dialog for saving the file "GateDemo.vwf".

14. Run the simulation through the "Simulation—Run Functional Simulation" menu, which should open a window similar to Figure 21. After the simulation you should get the value of the logic gate output corresponding to the inputs you specified (Figure 22).

Completed successfully.	
**** Generating the ModelSim .do script ****	
C:/Users/asroliveira/ISDig/GP4/Parte1/simulation/qsim/GateDemo.do generated.	
Completed successfully.	
**** Running the ModelSim simulation ****	
c:/altera_lite/15.1/modelsim_ase/win32aloem//vsim -c -do GateDemo.do	
Reading C:/altera_ite/15.1/modelsim_ase/tcl/vsim/pref.tcl	
# 10.4b	
# do GateDemo.do	
# Model Technology ModelSim ALTERA vlog 10.4b Compiler 2015.05 May 27 2015 # Start time: 09:57:13 on Oct 17.2016	
# start time: 05:57:15 01:00:17;2010 # vlog-work work GateDemo.vo # - Compiling module GateDemo	
#	
# Top level modules:	-
# GateDemo #End time: 09:57:13 on Oct 17,2016, Elapsed time: 0:00:00	
# End time: 09.57.13 on Oct 17,2010, Elapsed time: 0.00.00 # Errors: 0, Warnings: 0	
# Model Technology ModelSim ALTERA vlog 10.4b Compiler 2015.05 May 27 2015	
# Start time: 09:57:13 on Oct 17,2016	
# vlog -work work GateDemo.vvf.vt # Compiling module GateDemo_vlg_vec_tst	
# Company module Galebenio_vig_vec_tsi	
# Top level modules:	
# GateDemo_vig_vec_tst	
# End time: 09:57:14 on Oct 17,2016, Elapsed time: 0:00:01	
#Errors: 0, Warnings: 0	



			H W XC	(5) X? XB ≥ ≥	는 🔁 🐨				
laster Tin	ne Bar: 0 p	08		Pointer: 968.07 ns	Interval:	968.07 ns	Start:	End:	
	Name	Value at 0 ps	0 ps 0 ps	160.0 ns	320.0 ns	480.0 ns	640.0 ns	800.0 ns	960.0 ns
I	inPort0	В 0							
in	inPort1	в 0							
out	outPort	В 0							

Figure 22 - Simulator window with output signal waveform as a function of the specified input vectors and circuit behaviour.

15. Add to the "GateDemo.bdf" file the instantiation of an inverter to construct a NAND logical port (instantiating a component from a library means using it in the context of a project). Perform the behavioral simulation of the set.

Part II Hierarchical Design Demonstration

1. Create a new project by replicating the steps in part 1 of this guide. Designate the new project and its top-level entity as "EqCmpDemo".

2. Create a new file for a schematic diagram called "EqCmp4.bdf" to implement a 4-bit word equality comparator (Figure 23). At the end of editing the logic scheme, the circuit should look similar to Figure 24. Use the "xnor" and "and4" components of the Quartus Prime library accessible through the Symbol Tool.

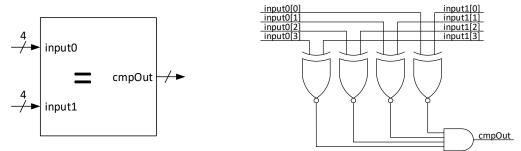


Figure 23 - Logical diagram of the 4-bit word equality comparator.

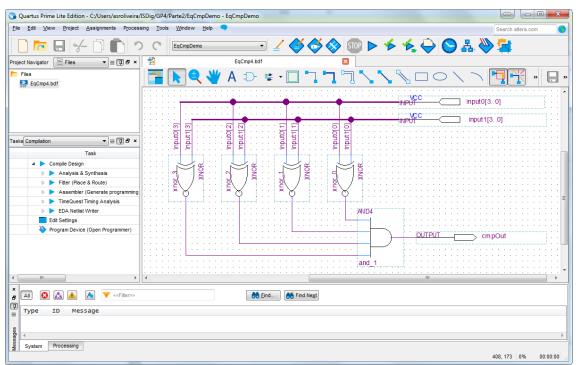


Figure 24 - Logical diagram of the 4-bit word equality comparator with interconnection and identification of the various circuit elements.

3. Create a symbol for the "EqCmp4" module so that it can be used in a schematic diagram as shown in Figure 25 and write it with the name "EqCmp4.bsf" (Figure 26).

) Qu File	uartus Prime Lite Edition - C:/U: Edit View Project Assign		ra/ISDig/GP4/Parte2/EqCmpDemo - EqCmpDemo	⇔ = = ×
	New Ctr Open Ctr	trl+N trl+O trl+F4	○ C EqCmpDemo × EqCmpLedd EqCmp4.bdf EqCmp4.bdf	
£	New Project Wizard Open Project Cti Save Project Close Project	tri+J		
æ	Save As Save All Ctr	trl+S trl+Shift+S	×	0]
	File Properties Create / Update	•	Create HDL Design File from Current File	· · · · · · · · · · · · · · · · · · ·
Ħ	Export Convert Programming Files Page Setup Print Preview		Create Symbol Files for Current File Create Verilog Instantiation Template Files for Current File Create VHDL Component Declaration Files for Current File AND4	
	Print Ctr Recent Files	trl+P	Create Design File from Selected Block	Dut
	Recent Projects Exit All	► It+F4	Create SignalTap II File from Design Instance(s) Create JAM, JBC, SVF, or ISC File CreateIVpdate PS File III Create Board-Level Boundary-Scan File III	
	AII 🙆 🚵 🛕 📐	Y < <filter< td=""><td>Create Top-Level Design File From Pin Planner</td><td></td></filter<>	Create Top-Level Design File From Pin Planner	
seges)
	System Processing e symbol files for current file		4	06, 232 0% 00:00:00

Figure 25 - Creation of a symbol for the "EqCmp4" module.

🕤 Create Syn	nbol File			8	- 1				x
Look in:	C:\	Users\asrolivei	ra\ISDig\GP4\Pa	arte2		- G	00	📑 🖽	
🔊 My Cor	mputer	Name	~			Size	Туре	Date Modif	fied
k asroliv	reira	i db					Filder	17-10-20	.23:
		•			III				×
File name:	EqCmp	4.bsf						Save	
Files of type:	Symbol	File (*.bsf)					•	Cancel	

Figure 26 - Recording of file "EqCmp4.bsf" relative to module symbol "EqCmp4".

4. Create a new file for a schematic diagram called "EqCmpDemo.bdf" to instantiate the equality comparator built in the previous point (Figure 27) and link its ports as illustrated in Figure 28.

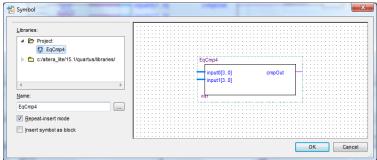


Figure 27 - Instantiation in a schematic diagram of the module "EqCmp4".

Quartus Prime Lite Edition - C:/Users/asroliveira/	ISDig/GP4/Parte2/EqCmpDemo - E	qCmpDemo			
Eile Edit View Project Assignments Process	sing <u>T</u> ools <u>W</u> indow <u>H</u> elp 🤜				Search altera.com
🗋 🗖 🗔 😽 🗇 💼 💈	EqCmpDemo		🗞 💷 🕨 🤸 א	k 🔶 🚫 🕹 k	👋 🚘
Project Navigator Files 💌 🗐 🗗 🗙	EqCmp4.bdf	🛛 🔁 Bk	ick1.bdf 🛛 🔀	EqCmpDemo.bdf*	
Files	🔁 💽 🔍 👋 .	A 🕩 📽 🗖 🎝	$\neg \neg \neg \land \land$,00/>	ヽ <mark>ヽ゚ヿ゚ヽ゚ヽ゚</mark> ∧
EqCmp4.bdf					***************************************
Tasks Compilation					· · · · · · · · · · · · · · · · · · ·
Tasks Compliation			EqCmp4		
Compile Design Analysis & Synthesis	inPort0[30]		input0[30]	cmpOut	OUTPUT cmpPort =
Fitter (Place & Route) Assembler (Generate programming	inPort1[30]		input1[30]		
TimeQuest Timing Analysis			eg cmp 4 inst		
EDA Netlist Writer Edit Settings					
Program Device (Open Programmer)					
۱	•			III	•
× AII 😢 🖄 🔺 📐 💙 < <fiter>></fiter>		Eind 65 Find Negt			
Type ID Message					
sole -					•
System Processing					843, 218 0% 00:00:00

Figure 28 - Interconnection of the "EqCmp4" module and the input and output ports and identification of the various circuit elements in the "EqCmpDemo" module.

5. Perform the behavioural simulation of the comparator.

6. Build and simulate a 16-bit comparator from the instantiation of 4-bit comparators and in as few elementary logic gates as possible.