# Laboratório de Sistemas Digitais

# Trabalho Prático nº 1

# Introdução às FPGAs, VHDL, ferramentas de projeto e kit de desenvolvimento

### Objetivos

- Familiarização com as ferramentas de projeto e com o kit de desenvolvimento com FPGA que vai ser usado nas aulas práticas.
- Captura de diagramas esquemáticos (esquemas lógicos), modelação em VHDL, simulação, implementação em FPGA e teste de componentes elementares.

# Sumário

Este trabalho prático está dividido em 4 partes. Na parte I pretende-se introduzir os aspetos básicos das ferramentas e do *kit* de desenvolvimento que vai ser usado nas aulas práticas de Laboratório de Sistemas Digitais (LSD), com base num projeto simples baseado em captura de diagrama esquemático. São abordadas as diversas fases de projeto, desde a descrição até ao teste, passando pela simulação e implementação. Na parte II é ilustrado o projeto hierárquico baseado inteiramente na linguagem de descrição de hardware VHDL, através da construção gradual de uma porta lógica NAND a partir de uma AND e de um inversor. A parte III é dedicada ao projeto híbrido baseado em diagramas esquemáticos e descrições VHDL para diferentes partes do sistema e introduz-se os operadores lógicos em VHDL. Na parte IV é abordado um exemplo, baseado num comparador de igualdade, que permite ilustrar algumas das vantagens da linguagem VHDL, ao nível da abstração e produtividade.

# Advertências muito importantes:

- A placa de desenvolvimento usada nas aulas práticas de Laboratório de Sistemas Digitais possui uma FPGA e diversos componentes que se podem danificar devido a descargas de eletricidade estática, pelo que deve ser manuseada com cuidado. Em particular, não deve tocar com qualquer parte do corpo ou objetos (incluindo vestuário) nos seus contatos elétricos e conetores.
- No final da aula, desligue o kit e arrume-o adequadamente juntamente com os cabos e alimentador na respetiva caixa.
- Na realização deste guião respeite escrupulosamente os nomes indicados para todos os projetos, ficheiros, módulos e identificadores (portos, sinais, etc.).

# Resumo do Fluxo de Projeto para Sistemas Baseados em FPGA

A Figura 1 ilustra o fluxo de projeto para sistemas baseados em FPGA. Os diversos passos são resumidos de seguida.

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 LSD vai ser utilizada captura de diagramas esquemáticos e/ou a linguagem de descrição de hardware VHDL consoante o que for mais adequado para cada sistema. No caso das descrições em VHDL pode ser utilizado qualquer editor de texto para este efeito, embora por conveniência seja usado o editor integrado no ambiente de desenvolvimento (*Integrated Development Environment* – IDE) que vamos adotar (*"Quartus Prime"*), o qual possui também

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 FPGA usada.



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

Seguidamente, a *netlist* vai ser implementada (compilada) para uma determinada família de FPGAs por uma ferramenta de software (desenvolvida pelo fabricante da FPGA) que realiza os seguintes passos:

- Mapeamento da *netlist* em primitivas da FPGA (tabelas de verdade, portas lógicas, multiplexadores, registos, etc.);
- Posicionamento das primitivas em localizações específicas da FPGA;
- Encaminhamento das interconexões (estabelecimento das ligações) entre as primitivas da FPGA.

O resultado da implementação é um ficheiro de configuração da FPGA que deve ser usado para a sua programação através de software e um cabo de programação adequado.

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. A maior parte destes passos vai ser ilustrada ao longo deste trabalho prático.

**Nota importante:** devido a limitações de velocidade da rede no acesso ao diretório pessoal em arca.ua.pt, nos PCs das salas de aula <u>recomenda-se a utilização de uma *pendrive* para guardar os seus projetos e ficheiros. <u>Caso isso não seja possível, guarde e aceda ao seu</u> <u>trabalho a partir de um diretório da drive Z</u>:. Em qualquer dos casos <u>não utilize espaços nem</u></u>

<u>carateres especiais</u> (e.g. acentos) nas *paths* dos projetos e ficheiros, uma vez que isso causará problemas na utilização das ferramentas (isto significa que <u>não deve gravar</u> 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 LSD (e.g. C:\Users\<Utilizador>\LSD\Aula1\Parte1\Projeto); em Linux poderá fazê-lo em /home/<utilizador>/LSD/Aula1/Parte1/Projeto).

# **Parte I** Demonstração das etapas fundamentais do fluxo de projeto com base em captura de diagrama esquemático

**1.** Abra a aplicação "Quartus Prime" e crie um novo projeto (menu "File $\rightarrow$ New Project Wizard") de acordo com os passos seguintes (Figuras 2 a 9).



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

🕞 Nev	v Project Wizard	×
Int	roduction	
The	New Project Wizard helps you create a new project and preliminary project settings, including the following:	
٠	Project name and directory	
:	Name of the top-level design entity Project files and libraries	
•	Target device family and device	
You	can change the settings for an existing project and specify additional project-wide settings with the Settings command (Assignments menu).	You can
use	the various pages of the Settings dialog box to add functionality to the project.	
E r	Don't show me this introduction again	
	gent ono ti me uno antioosocioni againi	
		11-1-
	< Back Next> Einish Cancel	<u>n</u> etb

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

New Project Wizard	<u> </u>
Directory, Name, Top-Level Entity	
What is the working directory for this project?	
C:\Users\asroliveira\LSDig\Aula1\Parte1	
What is the name of this project?	
GateDemo	
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 flik	ə.
GateDemo	
Use Existing Project Settings	
< Back Next > Einish Cancel	<u>H</u> elp

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 carateres especiais, e.g. acentos.

Project	Туре							
Select the	type of project to c	create.						
Empty	project							
Creat	e new project by s	specifying project files	and libraries, targ	et device family a	nd device, and EDA	tool settings.		
Project	template							
Creat templ	e a project from an ates from the <u>Desi</u>	i existing design templ <u>an Store</u> .	late. You can cho	ose from design te	mplates installed w	ith the Quartus Prir	me software, or do	wnload design

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

•			IļA bbA
File Name	Type Library	Design Entry/Synthesis Tool HDL Version	Remove
			<u>U</u> p
			Down
			Properties

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

						to the <u>beride ouppoint the</u> heepinge.	
Device family					Show in 'Available of	devices' list	
Family: Cyclone	IV E			•	Package:	Any	•
Devices: All				-	Pin count:	Any	•
					Core Speed grade:	Any	-
Target device					core op <u>e</u> ca grade.		_
Auto device s	elected by the Fitter				Name filter:	115F29	
O Other: n/a					Snow advanced	a devices	
O Other: n/a	Core Voltage	I Fe	Total I/Os	GPIO	<ul> <li>Memory Bit</li> </ul>	a devices	
Other: n/a     Agailable devices:     Name EP40E115E2907	Core Voltage	LEs	Total I/Os	GPIO:	s Memory Bit	s Embedded multiplier 9-bit elements	4
Other: n/a	Core Voltage 1.2V 1.2V	LEs 114480 114480	Total I/Os 529 529	GPIO: 529 529	Memory Bit     3981312     3981312	s Embedded multiplier 9-bit elements 532 532	4
<u>Q</u> ther: n/a <u>Name</u> EP4CE115F29C7 EP4CE115F29C8 EP4CE115F29C8L	Core Voltage 1.2V 1.2V 1.0V	LEs 114480 114480 114480	Total I/Os 529 529 529	GPIO: 529 529 529	<ul> <li>Memory Bit 3981312 3981312 3981312</li> </ul>	s Embedded multiplier 9-bit elements 532 532 532	4 4 4
<u>Q</u> ther: n/a Agailable devices: Name EP4CE115F29C7 EP4CE115F29C8 EP4CE115F29C8L EP4CE115F29C9L	Core Voltage 1.2V 1.2V 1.0V 1.0V	LEs 114480 114480 114480 114480	Total I/Os 529 529 529 529 529	GPIO: 529 529 529 529 529	<ul> <li>Snow advanced</li> <li>Snow advanced</li> <li>3981312</li> <li>3981312</li> <li>3981312</li> <li>3981312</li> </ul>	s Embedded multiplier 9-bit elements 50 532 532 532 532 532	4444
<u>Q</u> ther: n/a Agailable devices: Name EP4CE115F29C7 EP4CE115F29C8 EP4CE115F29C8L EP4CE115F29C9L EP4CE115F29I7	Core Voltage 1.2V 1.2V 1.0V 1.0V 1.2V	LEs 114480 114480 114480 114480 114480 114480	Total I/Os 529 529 529 529 529 529	GPIO: 529 529 529 529 529 529 529	<ul> <li>Snow advanced</li> <li>Snow adv</li></ul>	s Embedded multiplier 9-bit elements 532 532 532 532 532 532 532 532	44444
Qther: n/a Agailable devices: Name EP4CE115F29C7 EP4CE115F29C8 EP4CE115F29C8 EP4CE115F2917 EP4CE115F2918	Core Voltage 1.2V 1.2V 1.0V 1.0V 1.2V 1.0V 1.0V	LEs 114480 114480 114480 114480 114480 114480 114480	Total I/Os 529 529 529 529 529 529 529 529	GPIO: 529 529 529 529 529 529 529 529 529	<ul> <li>Snow advanced</li> <li>Snow advanced</li> <li>3981312</li> <li>3981312</li> <li>3981312</li> <li>3981312</li> <li>3981312</li> <li>3981312</li> <li>3981312</li> </ul>	s Embedded multiplier 9-bit elements 532 532 532 532 532 532 532 532 532 532	444444
© ther: n/a valable devices: Name EP4CE115F29C7 EP4CE115F29C8 EP4CE115F29C9L EP4CE115F2917 EP4CE115F2918L	Core Voltage 1.2V 1.2V 1.0V 1.0V 1.2V 1.0V 1.2V 1.0V	LEs 114480 114480 114480 114480 114480 114480	Total I/Os 529 529 529 529 529 529 529 529	GPI0: 529 529 529 529 529 529 529	<ul> <li>Snow advances</li> <li>Memory Bit</li> <li>3981312</li> <li>3981312</li> <li>3981312</li> <li>3981312</li> <li>3981312</li> <li>3981312</li> <li>3981312</li> </ul>	s Embedded multiplier 9-bit elements 532 532 532 532 532 532 532 532 532	4 4 4 4 4

Figura 7 – Passo 3 – seleção da FPGA usada (**Cyclone IV EP4CE115F29C7**) – a especificação do filtro "115F29" (campo *"Name filter"*) facilita a seleção da FPGA correta (primeira da lista).

Tool Type	Tool Name	Format(s)		Run Tool Automatically
Design Entry/Synthesis	<none></none>	<none></none>	Ψ	Run this tool automatically to synthesize the current design
Simulation	ModelSim-Altera	▼ VHDL	•	Run gate-level simulation automatically after compilation
Formal Verification	<none></none>	¥		
Board-Level	Timing	<none></none>	-	
	Symbol	<none></none>	-	
	Signal Integrity	<none></none>	-	
	Boundary Scan	<none></none>	Ŧ	

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

S New Project Wizard	×
Summary	
When you click Finish, the project will be created with the following settings:	
Project directory:	C:\Users\asroliveira\LSDig\Aula1\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:	EP4CE115F29C7
EDA tools:	
Design entry/synthesis:	<none> (<none>)</none></none>
Simulation:	ModelSim-Altera (VHDL)
Timing analysis:	0
Operating conditions:	
VCCINT voltage:	1.2V
Junction temperature range:	0-85 °C
	< Back Next> Einish Cancel Help
l	

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

2. Após premir "Finish" o IDE "Quartus Prime" deve apresentar o aspeto da Figura 10.



Figura 10 – Aspeto do "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.

<b>N</b>	ew	x
	New Quartus Prime Project	
4	Design Files	
	AHDL File	
	Block Diagram/Schematic File	
	EDIF File	
	Qsys System File	
	State Machine File	
	SystemVerilog HDL File	
	Tcl Script File	
	Verilog HDL File	
	VHDL File	=
4	Memory Files	
	Hexadecimal (Intel-Format) File	
	Memory Initialization File	
4	Verification/Debugging Files	
	In-System Sources and Probes File	
	Logic Analyzer Interface File	
	SignalTap II Logic Analyzer File	
	University Program VWF	
4	Other Files	
	AHDL Include File	
	Block Symbol File	
	Chain Description File	-
	OK Cancel He	p

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)).

- in	Symbol		×
	Libraries:		
	▲ D c:/altera_lite/15.1/quartus/libraries/ ▲		
	megafunctions		
Symbol Tool	others		
	<ul> <li>Primitives</li> </ul>		
- )	b buffer		
a)	🖌 🗁 logic		
•	다 and12	AND2	
	t∰ and2		
	t∰ and3	inst	
in <b>B</b>	ट्री and4 र		
	٠		
	Name:		
in in i	and2		
Input			
out a contract of	Repeat-Insert mode		
Output	Insert symbol as block		
io para		·····	
			OK Cancel
· · · · · · · · · ·			
b)		c)	

Figura 12 – Botões "Symbol Tool" e "Pin Tool" da barra de ferramentas e seleção da porta lógica AND na biblioteca de componentes do "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).

Organize 🔻 Nev	w folder				855	• 🔞
🔆 Favorites	-	Name	^		Date modified	Туре
E Desktop		퉬 db			05-01-2016 12:06	File fold
〕 Downloads	Ξ					
🔛 Recent Places						
🚺 CloudStation						
🐔 OneDrive						
😌 Dropbox						
诸 Google Drive						
퉬 ownCloud						
🚝 Librarier	-	•				
File <u>n</u> ame:	GateDe	mo.bdf				•
Save as type:	Block D	iagram/Sc	hematic Files (*.	bdf)		-

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.

Quartus Prime Lite Edition - C:/Users/asroliveira	ra/LSDig/Aula1/Parte1/GateDemo - GateDemo	10. 1.010 1.00	the Barrar Manual. North		- 0 <mark>- X</mark> -
Eile Edit View Project Assignments Proce	essing <u>T</u> ools <u>W</u> indow <u>H</u> elp 🤜			Search altera	.com 🌖
🗋 🗖 🖯 🗖 🗖 '	っ c GateDemo 💽 🗹 🎸	🍜 🐟 💿 🕨 🎸	• 🗲 🍚 😂 🚠 📎 🔝		
Project Navigator 🔥 Hierarchy 🔹 🗐 🗗 🗙	🛪 🔂 GateDemo.bdf 🗵 👇	Compilation Report - GateDemo		IP	Catalog 📮 🗗 🗙
Entity:Instance	Table of Contents	Flow Summary			× ≍
Cyclone IV E: EP4CE115F29C7	Flow Summary	Flow Status	Successful - Tue Jan 05 12:27:20 2016		installed IP
GateDemo 📩	E Flow Settings	Revision Name	GateDemo		
	Flow Non-Default Global Settings	Top-level Entity Name	GateDemo		No Sele
	Flow Elapsed Time	Family	Cyclone IV E		⊿ Library
	Flow OS Summary	Device	EP4CE115F29C7		Basic F
4	Flow Log	Timing Models	Final		DSP
	Elow Marsanar	Total combinational functions	1		Interfac
Tasks Compilation	Flow Suppressed Messages	Dedicated logic registers	0		Memory
Task		Total registers	0		Process
4 🕨 Compile Design		Total pins	3		Univers
Analysis & Synthesis		Total virtual pins	0		Search for
Pitter (Place & Route)		Finhedded Multiniar 9 bit elements	0		
? D Assembler (Generate programming	19	Total PLLs	0		
ImeQuest Iming Analysis	_				
EDA Netist Writer					
Program Device (Open Programmer)	-				F F
	-				+ Add
4	*				+ Aug
* Type ID Message					
Running Quartus Pr	rime Analycic & Synthesic				
E Command: guartus_n	mapread_settings_files=onwrite_settings_f	files=off GateDemo -c G	ateDemo		
20030 Parallel compilation	ion is enabled and will use 2 of the 2 process	ors detected			
12021 Found 1 design uni	its, including 1 entities, in source file gater	demo.bdf			
12127 Elaborating entity	y "GateDemo" for the top level hierarchy				
16010 Cenerating bard b	thesis is running				
21057 Implemented 4 devi	vice resources after synthesis - the final resor	urce count miaht be dif	ferent		
Quartus Prime Anal	lysis & Synthesis was successful. 0 errors, 0 v	warnings			
w					
					F
System (9) Processing (10)					
			8	6, 137 100%	00:00:17

Figura 15 – "Quartus Prime" IDE após a execução da "Analysis & Synthesis".

**9.** Simule o comportamento da porta lógica, criando para tal um ficheiro VWF (menu "*File* $\rightarrow$ *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).

X 0 1 7 VT			.a.s. 16	-100,						Sea	rch altera.	com 🧿
ne Bar: Ops		2 X∞ X2 XB R5	RATINA ( 38 ns		Interval: 1.3	lô ns		Start:		End:		
Name Value at 0 ps	0 ps a 0 ps	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	300.0 ns	880.0 ns	960.0 ns
× Dek	te	De	l									
Gro	rt Node or Bu uping	US	,									
Rev Rad	erse Group o ix	or Bus Bit Order										
Proj	erties											

Name:		ок
Туре:	NPUT -	Cancel
Value type:	9-Level 🔻	
Radix:	Binary 💌	Node Finder
Bus width:	1	
Start index:	0	

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

🕥 Node Finder	< 6 (ER )			x	🕥 Node Finder	< 6 (EA )			X
Named: *		Filter: Pir	is: all	• ОК	Named: *		Filter: F	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
LEDR	Output Group				LEDR	Output Group		SW[0]	Input
SW	Input Group	>			SW	Input Group	>	LEDR[0]	Output
in SW[0]	Input	>>			in_ sw[0]	Input	>>	-	
🖳 SW[1]	Input	<			🖺_ SW[1]	Input	<		
		<					<		

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 <u>entradas</u> do circuito ao longo do tempo – o(s) valor(es) da(s) <u>saída(s)</u> será(ão) determinado(s) durante a simulação. Utilize para tal o rato, selecionando os troços do diagrama temporal do sinal pretendido com o valor lógico que deseja que ele assuma (Figura 19).

Simu	ulation Wave	eform Editor - Ca	Users/asroliveira/LSDig/Aula1/Parte1/GateDemo - GateDemo - [Waveform.vwf]*	
ile <u>E</u>	dit <u>V</u> iew	Simulation Hel	Sea	rch altera.com
	् 🗼 🕹	Λ <u>z</u> χīχ	· ※ Xo Xo Xo Xe 😪 😪 🐴 🔳 联	
laster	Time Bar: 0	ps	Pointer: 990.23 ns Interval: 990.23 ns Start: End:	
	Name	Value at 0 ps	0 ps 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 8 0 ps	380.0 ns 960.0 ns
in	SW[0]	в 0		
n	SW[1]	в 0		
ut	LEDR[0]	вх		*******
(		÷	<u>د</u> ا	- F
				0% 00:00:0

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).

ave vector waveform Fi	e g → Aula1 → Parte1 → 🗸 😽 So	earch Parte1	م
Organize 🔻 New	folder		• 🕡
👉 Favorites	A Name	Date modified	Туре
E Desktop	📕 db	05-01-2016 12:14	File folde
🐌 Downloads	incremental_db	05-01-2016 12:13	File folde
🔛 Recent Places	output_files	05-01-2016 12:14	File folde
隘 CloudStation	simulation	05-01-2016 12:14	File fold
🐔 OneDrive			
😌 Dropbox			
👠 Google Drive			
鷆 ownCloud			
🛤 Librarier	• • III		,
File <u>n</u> ame:	ateDemo.vwf		•
Save as <u>t</u> ype: U	niversity Program VWF (*.vwf)		•
Hide Folders	Add file to current project	Save	ncel

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.

ile ;	dit <u>V</u> iew	Simulation He	ip 🤜											Search altera.	com
	<u>२</u> 🔬 🛽	h Z XI >	<b>王 昭)</b>	(C)	B	À 🔳 🎼									
Master	Time Bar: 0	ps		• •	Pointer: 52	5.4 ns		Interval: 525.	4 ns	St	art:		End:		
	Name	Value at 0 ps	0 ps 0 ps	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
in_	SW[0]	B 0													
in	SW[1]	B 0													
out	LEDR[0]	B 0													

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.** Para implementar o circuito em hardware real (FPGA), importe agora as definições de pinos da FPGA da placa de desenvolvimento que vai usar (localização física dos pinos aos quais estão ligados à FPGA os vários dispositivos da placa – e.g. LEDs, interruptores, displays, etc.). Para tal use o menu "Assignments →Import Assignments" (Figura 23).

O ficheiro que contém todas as definições de pinos da FPGA da placa DE2-115 é o "master.qsf". Este ficheiro está disponível para *download* no site da UC (em elearning.ua.pt) e não deve ser alterado. Este ficheiro será usado ao longo do semestre, pelo que deve ser colocado num diretório base e comum a todos os projetos.

Import Assignments	Select File		×
Specify the source and categories of assignments to import.	C C → C LSDig → Aula1 → Parte1 →	✓ Search Parte1	٩
Ele name: Categories	Organize 🔻 New folder	8= -	
Copy existing assignments into Altera15Test.qsf.bak before importing Advanced	★ Favorites Name	Date modified	Туре
	📃 Desktop 🛛 🎍 db	20-02-2017 10:52	File folder
OK Cancel Help	📕 Downloads 💡 🔒 incremental_db	20-02-2017 10:51	File folder
h.	💯 Recent Places 🥼 output_files	20-02-2017 10:51	File folder
	🖟 CloudStation 🔋 🕌 simulation	20-02-2017 10:51	File folder
	💱 Dropbox 📄 GateDemo.qsf	08-01-2016 15:24	QSF File
Import Assignments	🔒 Google Drive 📄 master.qsf	02-05-2016 00:48	QSF File
	Le MEOCloud		
Specify the source and categories of assignments to import.	ownCloud		
Elle name: C:/Users/asroliveira/LSDig/Aula1/Parte1/master.qsf Categories	ConeDrive for Bus		
Copy existing assignments into GateDemo.qsf.bak before importing	ConeDrive		F
OK Cancel Help	File name: master.qsf	✓ Import Files (*.qsf *.esf	*.acf *.c 🔻
		<u>O</u> pen <b> </b> ▼	Cancel

Figura 23 – Seleção e importação do ficheiro "master.qsf" com as definições dos pinos da FPGA ligados aos dispositivos do *kit*.

**16.** Efetue a síntese e implementação do projeto através do comando *"Compile Design"*. No final da compilação o IDE deve apresentar o aspeto da Figura 24.



Figura 24 – "Quartus Prime" IDE após compilação (implementação) completa do projeto.

**17.** No final do processo de compilação, programe a FPGA através do comando *"Program Device"* que deverá abrir a janela da Figura 25.

<u>E</u> dit <u>V</u> iew P <u>r</u> oces	sing <u>T</u> ools <u>W</u> indo	w <u>H</u> elp 🤜				Search at	tera.com	
Hardware Setup No	Hardware	Mode:	FAG	•	Progress:			
Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Exar
Ma Stop								
Auto Detect								
Add File								
Change File								
Add Device								
1 <sup>10</sup> Up								
<u> Մա</u> Down								

Figura 25 – Janela inicial da aplicação de programação da FPGA.

**18.** Configure a interface usada para programação da FPGA, premindo o botão *"Hardware Setup"* e selecionando a opção *"USB-Blaster"* (Figura 26). Caso esta opção não esteja disponível, verifique se a placa possui a alimentação ligada e está conetada ao computador através do cabo USB.

	JTAG S	ettings			
elect a programmin ardware setup app	g hardware lies only to	e setup to the currer	use when pro it programmer	gramming devices window.	s. This programming
urrently selected ha	ardware:	No Hard	vare		
Available hardward	e items				
Hardware			Server	Port	Add Hardware
USE-blaster			Local	058-0	Remove Hardware

Figura 26 – Configuração da interface de programação da FPGA (USB-Blaster).

**19.** Premindo o botão *"Add File"*, abra o ficheiro de programação da FPGA com a configuração relativa ao circuito construído. O ficheiro deverá possuir o nome "GateDemo.sof" e encontrarse no sub-diretório "output\_files" do projeto (Figura 27).

**Nota:** se ao abrir a aplicação de programação, esta já apresentar o aspeto da Figura 28, não necessita de realizar a ação *"Add File"* descrita neste ponto.

👋 Select Progra	amming File
Look in:	🔓 C:\Users\asroliveira\LSla1\Parte1\output_files 🔹 🧿 🗿 📑 🔃 🗐
My Compu asroliveira	uter List Ve
File name: G	ateDemo.sof Open
Files of type: Pr	rogramming Files (*.sof *.pof *.jam *.jbc *.ekp *.jic)   Cancel

Figura 27 – Seleção do ficheiro de programação da FPGA (ficheiro "GateDemo.sof" no sub-diretório "output\_files" do projeto).

**20.** Para programar a FPGA prima o botão *"Start"*. Quando a programação da FPGA estiver concluída (com sucesso), a aplicação deve apresentar o aspeto da Figura 29.



Figura 28 – Janela da aplicação antes da programação da FPGA (com indicação da FPGA usada e ficheiro de programação selecionado).

Processing Tools Window	w <u>H</u> elp 🤜							Search	altera.com	
USB-Blaster (USB-0)	mmino when available		Mode:	JTAG		•	Progress:	100%	Successf	'ul)
File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMF
output_files/GateDemo.sof	EP4CE115F29	00562D95	00562D95							
•			m							
ED4CE116	520									
TDO	20									

Figura 29 – Janela da aplicação após programação da FPGA.

**21.** Teste a porta lógica AND no *kit* de desenvolvimento aplicando diversos vetores de teste através dos interruptores usados e observando no LED o valor da saída.

**22.** Feche a aplicação de programação da FPGA. Recomenda-se que grave as configurações da aplicação de programação num ficheiro com o mesmo nome do projeto (neste caso "GateDemo.cdf"). Desta forma, quando voltar a abrir este ficheiro através da janela principal do "Quartus Prime" a aplicação de programação já estará corretamente configurada.

23. Finalmente feche o projeto (menu "File→Close Project").

#### Parte II

# Demonstração do projeto hierárquico com base na linguagem de descrição de hardware VHDL

**1.** Crie para a FPGA do *kit* DE2-115 (**Cyclone IV EP4CE115F29C7**) um novo projeto, seguindo os mesmos passos do ponto 1 da parte I deste guião. Considere a informação de identificação do projeto apresentada na Figura 30. O sumário da descrição do projeto deve ser semelhante ao da Figura 31.

Directory, Name, Top-Level Entity			
Vhat is the working directory for this project?			
C:\Users\asroliveira\LSDig\Aula1\Parte2			
Vhat is the name of this project?			
VHDLDemo			
Vhat is the name of the top-level design entity for this project? This r	name is case sensitive and must exactly	match the entity name in the design file.	
AND2Gate			
Use Existing Project Settings			

Figura 30 – Identificação e localização do projeto no sistema de ficheiros – adaptar de acordo com o diretório usado, o qual não deve conter no caminho (*path*) espaços nem carateres especiais, e.g. acentos.

Summary	
When you click Finish, the project will be created with the follow	ing settings:
Project directory:	C:\Users\asroliveira\LSDig\Aula1\Parte2
Project name:	VHDLDemo
Top-level design entity:	AND2Gate
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Design template:	n/a
Family name:	Cyclone IV E
Device:	EP4CE115F29C7
EDA tools:	
Design entry/synthesis:	<none> (<none>)</none></none>
Simulation:	ModelSim-Altera (VHDL)
Timing analysis:	0
Operating conditions:	
VCCINT voltage:	1.2V
Junction temperature range:	0-85 °C

Figura 31 – Sumário final da criação do projeto.

2. Após premir "Finish" o IDE "Quartus Prime" deve apresentar o aspeto da Figura 32.

Quartus Prime Lite Edition - C:/Users/asroliveira/LSD	ig/Aula1/Parte2/VHDLDemo - AND2Gate		
Eile Edit View Project Assignments Processing	Tools Window Help 🤜	Se	arch altera.com 🔵
🗋 🗖 🖶 🤟 🗋 🤊	C AND2Gate 2 4 4 4 500 > 4 4 5	🔶 🛇 🛦 ờ 🚰	
Project Navigator A Hierarchy → ≡ (1) Ø × Entty:Instance Cyclone IV E: EP4CE115F29C7 AND2Gete r <sup>®</sup> <sub>10</sub>	Primed to Perform Go faster on next generation devices with the Spectra-Q <sup>2</sup> engine. Introducing the Hew Software Design Software	P Catalog  C  A the second sec	tory     ton Available     nettons     Protocols     Interfaces and Controllers
	<b>Couartus</b> <sup>®</sup> Prime	Procession Processi	ors and Peripherals
Tasks Compilation	Development Suite	Search for P	Partner IP
Task	Version 15 1 Lite Edition		
Compile Design			
Analysis & Synthesis			
Fitter (Place & Route)			
Assembler (Generate programming     TimeQuest Timing Analysis	'₩ B	uy Software	
EDA Netist Writer		a di Manana di Ananana Pantanana	
Edit Settings		au new software nerease	
Program Device (Open Programmer)		entation	
< III >	<ul> <li>Notificat</li> </ul>	tion Center + Add	
* Type ID Message			
€ () () () () () () () () () ()			
Gess -			4
System Processing			0% 00:00:00

Figura 32 – "Quartus Prime" IDE após a criação do projeto.

3. Crie um novo ficheiro para código fonte VHDL (menu *"File→New"*), de acordo com a Figura
 33.



Figura 33 – Seleção do tipo de ficheiro a criar (VHDL File).

**4.** Introduza no ficheiro que acabou de criar o código VHDL correspondente a uma porta lógica AND de 2 entradas (mostrado na Figura 34).



Figura 34 – Código fonte da porta lógica AND de 2 entradas ("AND2Gate.vhd").

5. Grave o ficheiro, cujo nome deverá ser "AND2Gate.vhd" (Figura 35).



Figura 35 – Caixa de diálogo para gravação do ficheiro "AND2Gate.vhd".

**6.** Seguidamente vai ser validado por simulação o comportamento da porta lógica modelada. No entanto, antes de efetuar a simulação, execute a opção "*Analysis & Synthesis*" para que a correção sintática e estrutura do projeto sejam analisadas. Após a execução da "*Analysis & Synthesis*" o IDE deve apresentar o aspeto da Figura 36.



Figura 36 – "Quartus Prime" IDE após a execução da "Analysis & Synthesis".

**7.** Simule o comportamento da porta lógica que acabou de descrever, criando para tal um ficheiro VWF (menu "*File* $\rightarrow$ *New*") (Figura 37).



Figura 37 – Seleção do tipo de ficheiro a criar (University Program VWF).

8. Após premir "OK" abrirá uma janela onde deverão ser indicados os sinais usados na simulação. À semelhança da simulação realizada na primeira parte do guião, 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 da "AND2Gate". Após premir "OK" poderá especificar os valores pretendidos ao longo do tempo para as entradas da "AND2Gate". Utilize para tal o rato, selecionando os troços do diagrama temporal do sinal pretendido com o valor lógico que deseja que ele assuma (Figura 38).

Simu	lation Wave	eform Editor - C:	/Users/asroliveira/LSDig/Aula1/Parte2/VHDLDemo - AND2Gate - [Waveform.vwf]*	
e <u>E</u>	dit <u>V</u> iew	Simulation <u>H</u> e	lp 🤜	Search altera.com
	<u>२</u> 👗 🖞	r it z XI >	E 器 /C /Z /E 💦 👯 🐜 📾 職	
aster	Time Bar: 0	ps	Pointer: 705.89 ns Interval: 705.89 ns Start:	End:
	Name	Value at 0 ps	0 ps 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 0 ps	ns 880.0 ns 960.0 ns
-	inPort0	В 0		
-	inPort1	B 0		
5	outPort	вх		*******
_		,		0% 00:00:

Figura 38 – Janela do simulador após a especificação das formas de onda (vetores) de entrada.

**9.** Após especificar os valores (vetores) de simulação, grave o ficheiro com o nome "AND2Gate.vwf".

**10.** Execute a simulação através do menu "Simulation  $\rightarrow$  Run Functional Simulation". Após a simulação deve obter o valor da saída da porta lógica AND correspondente às entradas que especificou (Figura 39).



Figura 39 – Janela do simulador após a simulação.

**11.** Crie um novo ficheiro VHDL, chamado "GateDemo.vhd", que irá servir para instanciar a porta lógica que acabou de descrever nos pontos anteriores e associá-la a pinos concretos da FPGA do *kit* de desenvolvimento que vai usar para a testar. Introduza o código VHDL mostrado na Figura 40 no ficheiro que acabou de criar, para instanciar a porta lógica e ligá-la a pinos adequados da FPGA (neste caso as entradas vão ser ligadas a interruptores e a saída ligada a um LED). Após editar, grave o ficheiro com o nome "GateDemo.vhd".



Figura 40 – Código fonte do módulo top-level ("GateDemo.vhd").

**12.** Altere o ficheiro *top-level* do projeto (ficheiro "principal" que inclui todos os outros e portanto está no nível hierárquico mais elevado do projeto) de acordo com a Figura 41. O *top-level* deixa de ser o "AND2Gate.vhd" e passa a ser o "GateDemo.vhd".

🕥 Quartus Prime Lite Edition - C:/Users/asroliveira/LSDig/Aula1/Parte2/VHDLDemo - AND2Gate	
File Edit View Project Assignments Processing Tools Window Help 🤜	Search altera.com
🗋 🔂 🚽 🤟 🗋 🎓 🗘 🗚 🖓 🛇 💷 🕨	🔸 🛧 🔶 🚫 👒
Project Navigator 📄 Files 🔹 🖶 🗗 🛪 🔷 AND2Gate.vhd 🛛 🕹 Compilation Report - AND2Gate 😒 🔅 GateDemo.vhd 🔀	IP Catalog 🗍 🗗 🗙
Fies       Importing       Importing	K      Instalad P      Project Directory     No Section Available      Library     Basic Functions     DSP     Interface Protocols     Memory interfaces and Contro     Processors and Peripherals     University Program     Search for Partner P
Edt Settings	۰ ( الس
	+ Add
<pre>x Type ID Message 0 286030 Timing-oriven Synthesis is running 0 &gt; 0 16010 Generating hard_block partition "hard_block:auto_generated_inst" 0 0 21057 Implemented 4 device resources after synthesis - the final resource count might be 0 0 Quartus Prime Analysis &amp; Synthesis was successful. 0 errors, 0 warnings 0 0 Syntm(20) Processing (10)</pre>	e different
Sets the current file entity as the top-level entity for the next complation	100% 00:00:20

Figura 41 – Alteração do módulo top-level para o "GateDemo.vhd".

**13.** Importe as definições de pinos da FPGA da placa de desenvolvimento, usando o menu "Assignments —>Import Assignments" (Figura 42).

🕞 Import A	Assignments
Specify the	e source and categories of assignments to import.
<u>F</u> ile name:	C:/Users/asroliveira/LSDig/Aula1/Parte2/master.qsf Categories
🔽 Copy e	xisting assignments into GateDemo.qsf.bak before importing Advanced
	OK Cancel Help

Figura 42 – Importação do ficheiro "master.qsf" com as definições dos pinos da FPGA ligados aos dispositivos do *kit*.

**14.** Efetue a síntese e implementação do projeto através do comando "*Compile Design*". No final da compilação o IDE deve apresentar o aspeto da Figura 43.



Figura 43 – "Quartus Prime" IDE após compilação (implementação) completa do projeto.

**15.** No final do processo de compilação, programe a FPGA. Quando estiver concluída (com sucesso) a programação da FPGA, a aplicação de programação deve apresentar o aspeto da Figura 44.



Figura 44 – Janela da aplicação após programação da FPGA.

**16.** Teste a porta lógica AND no *kit* de desenvolvimento aplicando diversos vetores de teste através dos interruptores usados e observando no LED o valor da saída.

**17.** Crie um novo ficheiro VHDL, contendo o código fonte correspondente a um inversor e no final grave com o nome "NOTGate.vhd" (Figura 45).



Figura 45 – Código fonte do inversor ("NOTGate.vhd").

**18.** Crie um novo ficheiro VHDL, contendo o código correspondente a uma porta lógica NAND de 2 entradas, construída a partir das duas portas lógicas (AND e inversor) implementadas nos pontos anteriores, instanciadas e interligadas de acordo com a estrutura ilustrada na Figura 46 e o código da Figura 47.

**Nota:** este ponto pretende ilustrar apenas a instanciação e interligação de componentes descritos em VHDL, com base num exemplo simples, não sendo a forma mais eficaz de criar uma porta lógica NAND, uma vez que a linguagem VHDL também disponibiliza o operador "nand". No entanto, esta forma tem a vantagem de ilustrar o projeto hierárquico em VHDL, com um exemplo simples, a instanciação e interligação textual de componentes, de forma análoga aos métodos de captura de diagrama esquemático.



de saída.

Quartus Prime Lite Edition - C:/Users/asroliveira/	SDig/Aula1/Parte2/VHDLDemo - AND2Gate	
<u>File Edit View Project Assignments Process</u>	ng <u>T</u> ools <u>W</u> indow <u>H</u> elp 🤜	Search altera.com
🗋 🗖 🖯 🦫 🖓	C AND2Gate 🖌 🎸 🎸 🔊 510P 🕨	> 🇲 🌴 🍚 🔗 🔹
Project Navigator 📄 Files 🔹 🗐 🗗 🗙	vhd 🖾 🛛 🗇 NOTGate.vhd 🖾 🔄 🧇 NOR2Gate.vhd 🗵 🔹 🍫 NAND2Gate.vhd 🔯 🔺	P Catalog 🛛 🖓 🗗 ×
Files  AND2Gate.vhd  AND2Gate.vwf  CateDame.vbd	Image: State of the s	<ul> <li>X =</li> <li>Installed IP</li> <li>Project Directory</li> </ul>
GateLemo.vhd       Image: NotTgate.vhd       Tasks       Compile Design       ▷       Compile Design       ▷       ▲       Compile Design       ▷       ▲       ▶       Filter (Place & Route)	<pre>3 4</pre>	No Selection Available  Library  Basic Functions  DSP  Interface Protocols  Memory Interfaces and Control Processors and Peripherals University Program Search for Partner IP
Assembler (Generate programming      Assembler (Generate programming      Dentity Uniter      Edd Settings      Program Device (Open Programmer)      III	<pre>17 17 18 19 20 20 20 20 21 22 23 24 4 10 10 10 10 10 10 10 10 10 10</pre>	< <u> </u>
X Type ID Message		► 0% 00.00.00

Figura 47 – Código fonte da porta lógica NAND de 2 entradas ("NAND2Gate.vhd").

19. Grave o ficheiro com o nome "NAND2Gate.vhd".

**20.** Edite o ficheiro *top-level* "GateDemo.vhd" de forma a que seja usado o componente NAND2Gate (arquitetura <u>Structural</u>) em vez do AND2Gate (arquitetura <u>Behavioral</u>) usado anteriormente – Figuras 48 e 49.



Figura 48 – Abertura do módulo top-level "GateDemo.vhd".

Quartus Prime Lite Edition - C:/Users/asroliveira,	SDig/Aula1/Parte2/VHDLDemo - AND2Gate	
<u>File Edit View Project Assignments Proces</u>	ing <u>T</u> ools <u>W</u> indow <u>H</u> elp <b>9</b>	Search altera.com
🗋 🗖 🖯 🤟 🗂	AND2Gate 🗸 🎸 🎸 🔊 📭 🕨	🄸 🛧 🤤 🚫 👒
Project Navigator Files = 🗐 🗗 ×	💠 GateDemo.vhd 🛛 🔹 NOTGate.vhd 🖂 💠 NOTGate.vhd 🖾 🔶 NOR2 4 🕨 P	Catalog 🔲 🗗 🗙
Files	📑 💏 🗄 📰 📰 🚺 🍢 🔥 🖉 🍒 🔹	X I
AND2Gate vwf GateDemo.vhd NOTGate.vhd NAND2Gate.vhd TasksCompilation Task Compile Design	<pre>1 library IEEE; 2 use IEEE.STD_LOGIC_1164.all; 3 4 Bentity GateDemo is 5 D port(SW : in std_logic_vector(1 downto 0); 6 LEDR : out std_logic_vector(0 downto 0)); 7 Lend GateDemo; 9 Barchitecture shell of GateDemo is 10 Bbegin 11 B system_core : entity work. WAND2Gate(Structural) 12 port map(inPort0 =&gt; SW(0); 13 L</pre>	American and a project Directory     No Selection Available     Library     DSP     Interface Protocols     Memory Interfaces and Perinherals
Analysis & Synthesis      Analysis & Synthesis      Fitter (Place & Route)      Assembler (Generate programming      TimeQuest Timing Analysis      EDA Netlist Writer      Edt Settings	<pre>infort1 =&gt; Sw(1); if outPort =&gt; LEDR(0)); if outPort =&gt; LEDR(0); if outPort =&gt; LEDR(0)</pre>	University Program     Search for Partner IP
Program Device (Open Programmer)	*	· · · · · · · · · · · · · · · · · · ·
Type ID Message	< >	+ Add
System (47) Processing	Ln 11 Col 51 VHDL File	0% 00:00:00

Figura 49 – Edição e alteração do módulo top-level "GateDemo.vhd".

**21.** Volte a importar o ficheiro "master.qsf" com as definições dos pinos da FPGA na placa de desenvolvimento.

**22.** Efetue a síntese e implementação do sistema através do comando "*Compile Design*". No final da compilação o IDE deve apresentar o aspeto da Figura 50.



Figura 50 – "Quartus Prime" IDE após compilação completa do projeto.

23. Programe a FPGA e teste no kit o funcionamento da porta lógica NAND implementada.

24. Feche a aplicação de programação da FPGA e seguidamente o projeto.

### Parte III

Demonstração do projeto híbrido baseado em diagramas esquemáticos e descrições VHDL

**1.** Crie para a FPGA do *kit* DE2-115 (**Cyclone IV EP4CE115F29C7**) um novo projeto, seguindo os mesmos passos do ponto 1 da parte I deste guião. Considere a informação de identificação do projeto apresentada na Figura 51. O sumário da descrição do projeto deve ser semelhante à Figura 52.

New Project Wizard	<b>X</b>
Directory, Name, Top-Level Entity	
Vhat is the working directory for this project?	
C:/Users/asroliveira/CloudStation/LSDig/Aula1/Parte3	
Vhat is the name of this project?	
LogicDemo	
Vhat 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.	
LogicTop	
Use Existing Project Settings	
< Back Next > Finish Cancel	Help

Figura 51 – Identificação e localização do projeto no sistema de ficheiros.

S New Project Wizard		
Summary		
When you click Finish, the project will be created with the following settings:		
Project directory:	C:/Users/asroliveira/CloudStation/LSDig2016/LSDig/Aula1/Parte3	
Project name:	LogicDemo	
Top-level design entity:	LogicTop	
Number of files added:	0	
Number of user libraries added:	0	
Device assignments:		
Design template:	n/a	
Family name:	Cyclone IV E	
Device:	EP4CE115F29C7	
EDA tools:		
Design entry/synthesis:	<none> (<none>)</none></none>	
Simulation:	ModelSim-Altera (VHDL)	
Timing analysis:	0	
Operating conditions:		
VCCINT voltage:	1.2V	
Junction temperature range:	0-85 °C	
	< <u>Back</u> <u>N</u> ext > <u>Finish</u> Cancel <u>H</u> elp	

Figura 52 – Sumário final da criação do projeto.

2. Crie um novo ficheiro para código fonte VHDL (menu "File→New").

**3.** Introduza no ficheiro que acabou de criar o código VHDL da Figura 53, correspondente a uma unidade que realiza diversas operações lógicas.

4. Grave o ficheiro, cujo nome deverá ser "LogicUnit.vhd".

```
library IEEE;
use IEEE.STD LOGIC 1164.all;
entity LogicUnit is
    port(input0 : in std logic;
         input1 : in std logic;
         invOut : out std logic;
         andOut : out std logic;
         orOut : out std logic;
         xorOut : out std logic;
         nandOut : out std logic;
         norOut : out std logic);
end LogicUnit;
architecture Behavioral of LogicUnit is
begin
    invOut <= not input0;</pre>
    andOut <= input0 and input1;</pre>
    orOut <= input0 or
                            input1;
    xorOut <= input0 xor input1;</pre>
    nandOut <= input0 nand input1;</pre>
    norOut <= input0 nor input1;</pre>
end Behavioral;
```

Figura 53 – Código fonte do módulo "LogicUnit" (exemplo em VHDL com operadores lógicos).

**5.** Para poder instanciar num diagrama esquemático o módulo "LogicUnit.vhd" escrito em VHDL é necessário criar um símbolo tal como ilustrado na Figura 54.



Figura 54 – Criação de um símbolo para um módulo de forma a poder ser utilizado num diagrama esquemático.

**6.** Crie um novo ficheiro para um diagrama esquemático (menu "File  $\rightarrow$ New"), que irá servir para instanciar a entidade "LogicUnit" e associá-la a pinos adequados da FPGA (entradas ligadas aos interruptores e saídas ligada a LEDs) do *kit* de desenvolvimento que vai usar para a testar.

**7.** Uma vez criado o símbolo, um módulo definido pelo utilizador pode ser usado num diagrama esquemático da mesma forma que qualquer bloco predefinido, usando o botão *"Symbol Tool"* da barra de ferramentas. Crie uma instância do módulo *"LogicUnit"* (Figura 55).



Figura 55 – Instanciação num diagrama esquemático do módulo "LogicUnit".

**8.** Ligue as entradas e as saídas do módulo "LogicUnit" a portos de entrada e de saída tal como ilustrado na Figura 56 (entradas ligadas a interruptores e saída a LEDs do *kit*).

🤉 Quartus Prime Lite Edition - C/Users/asroliveira/CloudStation/LSDig2016/LSDig/Aula1/Parte3/LogicDemo - LogicTop						
<u>File Edit View Project Assignments Process</u>	ng <u>T</u> ools <u>W</u> indow <u>H</u> elp 🤜	Search altera.com				
🗋 🗖 🗔 🤟 🗇 💼 🔿	C LogicTop	> 🚘				
Project Navigator 🔥 Hierarchy 💌 🚍 🗗 🗙	🔁 LogicTop.bdf* 🗵					
Entity:Instance	The second s	• • • • • • • • • • • • • • • • • • •				
➤ LogicTop r <sup>th</sup>	LogicUnit					
۲asks[Compilation ▼]≡ ₪ ♂ ×	SW[0]         INPUT         Input0         InvOut         OUTPUT           SW[1]         INPUT         Input1         andOut         OUTPUT         Input1	LEDG[0] LEDG[1]				
Task Compile Design Cacheric & Synthesis	vorout vorout	LEDG[2]				
Anarysis & Synthesis     Fitter (Place & Route)     Assembler (Generate programming	nandOut OUTFUT	LEDG[5]				
FireQuest Timing Analysis     EDA Netlist Writer  Edit Settings	inst					
Program Device (Open Programmer)		· · · · · · · · · · · · · · · · · · ·				
x Type ID Message		90.161 0% 00:000				

Figura 56 – Interligação do módulo "LogicUnit" e dos portos de entrada e de saída e identificação dos diversos elementos do circuito no módulo "LogicDemo".

9. Grave o ficheiro, cujo nome deverá ser "LogicTop.bdf".

**10.** Importe o ficheiro "master.qsf" com as definições dos pinos da FPGA na placa de desenvolvimento. Após a importação, o diagrama esquemático deverá apresentar o aspeto da Figura 57.

Quartus Prime Lite Edition - C:/Users/asroliveira	a/CloudStation/LSDig2016/LS	Dig/Aula1/Parte3/LogicDemo - LogicTop			
Eile Edit View Project Assignments Proce	essing <u>T</u> ools <u>W</u> indow <u>H</u> el	p 🗢			Search altera.com
- 🗋 🗖 🖯 👉	C LogicTop		100 🕨 🗲 🤸 🕒	& 🔶 😫	
Project Navigator 🔥 Hierarchy 💌 🚍 🗗 🛪	< 🔁	LogicTop.bdf			
Entity:Instance	🔁 📐 🍳 👌	🍟 A 🕩 📽 • 🔲 ٦ 🦳	$  \mathbf{N}   \mathbf{O}  $		🖶 🗲 🗋 🚺 🔈 🔹
LogicTop de LogicTop de		SW[0]	Logic U nit input0 inv Out		
< m >	PIN_AB28	SW[1] INPUT	input1 andOut	OUTPUT LEDG[1]	PIN_E21
Tasks Compliation	PIN_AC28		orOut	LEDG[2]	PIN_E22
Task			xorOut	OUTPUT LEDG[3]	PIN_E25 ≡
▲ ► Compile Design			nandOut	OUTPUT LEDG[4]	PIN_E24
Analysis & Synthesis			norOut	OUTPUT LEDG[5]	PIN_H21
Fitter (Place & Route)					PIN G20
Assembler (Generate programming)	9		Inst		
TimeQuest Timing Analysis					
EDA Netlist Writer					
Edit Settings					
Program Device (Open Programmer)					· · · · · · · · · · · · · · · · · · ·
< III )	• •				E.
Yype TD Message     140120 Import completed.     System(1) Processing	1065 assignments w	ere written (out of 1072 read). O no	n-global assignments were skipp	ped because of entity name mismatch.	» 180, 58 0% 00.00 00

Figura 57 – Módulo "LogicUnit" e portos de entrada e de saída com identificação dos pinos usados da FPGA.

**11.** Efetue a síntese e implementação do projeto através do comando *"Compile Design"*. No final da compilação o IDE deve apresentar o aspeto da Figura 58.

Quartus Prime Lite Edition - C:/Users/asroliveira	/CloudStation/LSDig2016/LSDig/Aula1/Parte3/LogicDemo - LogicTi	op					
Elle Edit Yew Boyert Assignments Pyrcessing Iools Window Help 📍							
	C LogicTop 🖌 🖌 🎸	🗳 🔇 💷 🕨 🔸	<b>★</b>				
Project Navigator	LogicTop.bdf		Complation Report - LogicTop				
EntryPatience           > Consol + E for (1972c7)           > Que of the form           > Tanit (Companies           > (Consent in the form)           > D = Consta in the form           > > > Consta in the form           > > > > > > Consta in the form           > > > > > > > > > Consta in the form           > > > > > > > > > > > > > > > >            > > > > > > > > > > > > > >            > > > > > > > > > > > > >            > > > > > > > > > > > > >            > > > > > > > > > > > >            > > > > > > > > > > > >            > > > > > > > > > > > >            > > > > > > > > > >            > > > > > > > > >            > > > > > > >            > > > > > > >            > > > > > >            > > > > >            > > > > >            > > >            > > >            > > >            > > >            > > >            > >            > >            > >            > >            > >            > >            > >            > >            > >            > >            > >            > >	Title of Contents     ()) of       The exclusions     ()) of       The exclus	Circle Annual / Area Status Area Status Response for your was Provide Status Provide Status Provide Status Test Synch Status Test Synch Status Test Synch Status Test Status Tes	Sevenda / Francis 11: 1988 2000 Lapito Uspito Sevenda / Sevenda /				
Type         ID         Message           Ø         0         204019 Generated file Log           Ø         0         204019 Generated file Log           Ø         0         Quartus Prime EDA           Ø         293000 Quartus Prime Full	icTop_min_1200mv_0c_vhd_fast.sdo in folder "C icTop_vhd.sdo in folder "C://wsers/asroliveira Netlist Writer was successful. 0 errors, 0 wa Compilation was successful. 0 errors, 528 wa 	:/Users/asroliveira/Clo a/CloudStation/LSDig2016 arnings arnings	udstation/LSO1g2016/LSO1g/Aulal/Parte3/simulation/modelsim/ /LSD1g/Aulal/Parte3/simulation/modelsim/" for EDA simulatio	for EDA simulation tool			
System (1) Processing (126)				.63.70 102% 00:01:04			

Figura 58 – "Quartus Prime" IDE após compilação (implementação) completa do projeto.

**12.** No final do processo de compilação, programe a FPGA através do comando "*Program Device*".

**13.** Teste o projeto no *kit* de desenvolvimento aplicando diversos vetores de teste através dos interruptores usados e observando nos LEDs os valores das saídas para as diversas funções lógicas.

**14.** Feche a aplicação de programação da FPGA e seguidamente o projeto.

# Parte IV

# Demonstração das vantagens de VHDL na modelação comportamental de componentes

**1.** Crie para a FPGA do *kit* DE2-115 (**Cyclone IV EP4CE115F29C7**) um novo projeto, chamado "EqCmpDemo", cuja entidade *top-level* deverá chamar-se "EqCmpDemo" e seguindo os mesmos passos do ponto 1 da parte I deste guião.

**2.** Crie um novo ficheiro para um diagrama esquemático, chamado "EqCmp4.bdf", para implementar um comparador de igualdade de palavras de 4 bits (Figura 59). No final da edição do esquema lógico, o circuito deve ser semelhante ao da Figura 60. Utilize os componentes "xnor" e "and4" da biblioteca do *"Quartus Prime"* acessível através do *"Symbol Tool"*.



Figura 59 – Diagrama lógico do comparador de igualdade de palavras de 4 bits.



Figura 60 – 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 61, e grave-o com o nome "EqCmp4.bsf" (Figura 62).

0	🕽 Quartus Prime Lite Edition - C./Users/asroliveira/CloudStation/LSDig2016/LSDig/Aula1/Parte4/EqCmpDemo - EqCmpDemo							
File	Edit View Project Assignment	s Proce	ssing Tools Window Help 🤜		Search altera.com			
	New Ctrl+N			🚜 🙈 📖 🕨 🔸 🝝 🖾 🖎 😫 🚵 🖶	F			
	Open Ctrl+O	L			•			
	Close Ctrl+F4	×	Compilation Report - EqCmpDemo	EqCmp4.bdf				
•	New Project Wizard		] 🔁 🕟 😑 👑 A 🕁 👳 🗸	<b>n h h h l l l l l l l l l l l l l l l l </b>	😽 🕼 🐘 🔲 🗴			
1	Open Project Ctrl+J							
	Save Project			· · · · · · · · · · · · · · · · · · ·	· · · · · · · · · · · · · · · · · · ·			
	Close Project		••••••••••••••••••••••••••••••••••••••	INPÚŤ (input0[3.	.0]			
	Save Ctriss			input 1/2	01			
10	Save As							
1.08	Save All Ctrl+Shi	iff+S	0[3]	. 킹 킹 킹 및				
<b>.</b>	001070		the second secon					
	File Properties	Ŀ,						
	Create / Update	•	Create HDL Design File from Current File					
	Export		Create Symbol Files for Current File	1. /×				
	Convert Programming Files		Create AHDL Include Files for Current File	I\$∕				
F	Dage Setup		Create Verilog Instantiation Template Files for Current File					
	Print Preview		Create VHDL Component Declaration Files for Current File	AND4				
	Print Ctrl+P		Create Design File from Colonied Black					
			Undate Design File from Selected Block		Out			
	Recent Files	•	opulie boogn the non-booked block.					
	Recent Projects	+	Create SignalTap II File from Design Instance(s)					
	Evê AB4E4		Create JAM, JBC, SVF, or ISC File	and_1				
1		_	Create/Update IPS File		••••••••			
×	Time TD Message	_	Create Board-Level Boundary-Scan File					
8	The the wessage	1	Create Top-Level Design File From Pin Planner					
₽		-		-				
≡								
des								
essa	System (14) Processing				,			
Crea	2 Updamitry Processing Create survey files for surgers files							
orcu	areas aynor nor on current no							

Figura 61 – Criação de um símbolo para o módulo "EqCmp4".

Create Syn	nbol File	×
Look in:	CUBers/asrokreira/CoudStationLSDg2016ULSDg2Aula1Parte4 CUBers/asrokreira/CoudStationLSDg2016ULSDg2Aula1Parte4 C C C C C C CUBErs/asrokreira/CoudStationLSDg2016ULSDg2Aula1Parte4 C C C C C C C C C C C C C C C C C C C	
File name: Files of type:	EqCmp4.bsf Symbol File (*.bsf)	Save Cancel

Figura 62 – 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 63) e efetuar a sua interligação a pinos da FPGA. Ligue a entrada "input0[3..0]" aos interruptores SW[3..0], a entrada "input1[3..0]" aos interruptores SW[7..4] e a saída "cmpOut" a LEDG[0] tal como ilustrado na Figura 64.



Figura 63 – Instanciação num diagrama esquemático do módulo "EqCmp4".

Ele Ed Yew Boyet Assymmets Pocessing Too Widew Beb   Seric Atera com  Comptition  Fies  Ecomption  Fies  F	🕥 Quartus Prime Lite Edition - C/Users/asroliveira/CloudStation/LSDig2016/LSDig/Aula1/Parte4/EqCmpDemo - EqCmpDemo										
Total     Test     Product Navigator     Test     Product Navigator     Test     Product Navigator     Product Navigator <th>Eile Edit View Project Assignments Process</th> <th>Search altera.com</th>	Eile Edit View Project Assignments Process	Search altera.com									
Project Navigator Preset Navig	□ た										
Fees   Pres	Project Navigator 📄 Files 🔹 🗐 🖲 x 🖌 😜 Completion Report - EqCinpOlemo 🔯 👘 EqCinpOlemo Adf 🔯										
Task       SV(3.0)       U201       input0(3.0)       cmpOut       OUTPUT       LED G(0)         ?       >       Assession       SV(7.4)       NCC       input1(3.0)       cmpOut	Files	<b>™ Q W</b> A D <b>W O N N W O O N N</b>	<b>⊒ ™</b> ∧								
Taska Taska Taska P by Anaysia 6 Synthesia SW(7.4) DUTRUT SW(7.4) DUTRUT SW(7.4) DUTRUT DUTR	EqCmpDemo.bdf		^ 								
Task     Swi[3:0]     WC       ?     >>     Anayaa 5 Syntesis       ?     >>     Anayaa 5 Syntesis       ?     >>     Fder (Poce 3 Roule)       ?     >>     >       ?     >>       ?     ?       ?     ?       ?     ?       ?     ?       ?     ?       ?     ?       ?     ?       ?     ?       ?     ?       ?     ?       ?     ?       ?     ? <th>Tasks Compilation</th> <th>Fa Cmn4</th> <th></th>	Tasks Compilation	Fa Cmn4									
?     ▲     Compte Desagn     SW(3,0)     WC     input(3,0)     cmpOut     DUIPUL     LEDG(0)       ?     ▶     > Assentiation (Generate programme)     SW(7,4)     WC     input(13,0)     cmpOut     cmpOut     cd(2,0)       ?     ▶     > Assentiation (Generate programme)     SW(7,4)     WC     cd(2,0)     cd(2,0)     cd(2,0)       ?     ▶     > EAA steating open Programme)     SW(7,4)     SW(7,4)     SW(7,4)     SW(7,4)       ?     ▶     > Assentiation (Generate programme)     SW(7,4)     SW(7,4)     SW(7,4)     SW(7,4)       ?     ▶     > EAA steating open Programme)     SW(7,4)     SW(7,4)     SW(7,4)     SW(7,4)       ?     ▶     > EAA steating open Programme)     SW(7,4)     SW(7,4)     SW(7,4)     SW(7,4)       *     T     *     *     **     **     **       *     T     *     **     **     **       *     T     *     **     **     **       *     **     **     **     **	Task										
?     >     >     Anaysis Symbols       ?     >     >     Fetr (Pace Shoule)       ?     >     >     Assember (Generate programme)       ?     >     >     >       ?     >     >       ?     >     > <th>? 🔺 🕨 Compile Design</th> <th>SW[30] INPUT input0[30] cmpOut OUTPUT</th> <th> LEDG[0]</th>	? 🔺 🕨 Compile Design	SW[30] INPUT input0[30] cmpOut OUTPUT	LEDG[0]								
	?  Analysis & Synthesis	SW[74] input1[30]									
?     >     Treduest Timing Analysis     ed_Cfmp_4_inSt       ?     >     >     ECA Healst Weer       Image: Comp_4_inSt     ed_Cfmp_4_inSt       Image: Comp_4_inSt     image: Comp_4_inSt       Image: Comp_4_in	Priter (Place & Roule)     Assembler (Generate programming										
?     >     ECA Heles Weer       Eda Settings	? > TimeQuest Timing Analysis	eq_cmp_4_inst									
Edi Settings Program Device (Open Programmer)	? D DA Netlist Writer										
Pogram Device (Open Pogrammer)         """"""""""""""""""""""""""""""""""""	Edit Settings										
Type         ID         Message           0	Program Device (Open Programmer)		•								
x Type ID Message			•								
2007 4 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	f f (?) ≣										
2 Jack 00 0000	Burtem (14) Brossesies		•								
041.003 100/8 00.00.03	System (17) Processing		641, 303 100% 00:00:05								

Figura 64 – Interligação do módulo "EqCmp4" e dos portos de entrada e de saída e identificação dos diversos elementos do circuito no módulo "EqCmpDemo".

**5.** Importe o ficheiro "master.qsf" com as definições dos pinos da FPGA na placa de desenvolvimento.

6. Efetue a síntese e implementação do projeto através do comando "Compile Design".

**7.** No final do processo de compilação, programe a FPGA e teste o comparador no *kit* de desenvolvimento aplicando diversos vetores de teste através dos interruptores usados e observando no LED o valor da saída.

8. Feche a aplicação de programação da FPGA.

**9.** Crie um novo ficheiro VHDL, chamado "EqCmp8.vhd" com o código VHDL da Figura 65, correspondente a um comparador de igualdade de duas entradas de 8 bits.

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity EqCmp8 is
    port(input0 : in std_logic_vector(7 downto 0);
        input1 : in std_logic_vector(7 downto 0);
        cmpOut : out std_logic);
end EqCmp8;
architecture Behavioral of EqCmp8 is
begin
    cmpOut <= '1' when (input0 = input1) else
        '0';
end Behavioral;
```

Figura 65 – Código fonte do módulo "EqCmp8" (comparador de igualdade de 8 bits).

**10.** Crie um símbolo para o módulo descrito no ficheiro "EqCmp8.vhd", tal como ilustrado na Figura 66, de forma a poder instanciá-lo num diagrama esquemático.



Figura 66 – Criação de um símbolo para o módulo "EqCmp8".

**11.** Troque no ficheiro *top-level* "EqCmpDemo" a instanciação do módulo "EqCmp4" pelo módulo "EqCmp8" (Figura 67) e ligue-o tal como ilustrado na Figura 68.



Figura 67 – Instanciação num diagrama esquemático do módulo "EqCmp8".

🕥 Quartus Prime Lite Edition - C./Users/szroliveira/CloudStation/LSDig2016/LSDig/Aula1/Parte4/EqCmpDemo - EqCmpDemo											
Eile Edit View Project Assignments Proces		Search altera.com									
□ 🚾 🖯 🦟 Ū 🗊 つ 🤉 Econolemo 🔹 🗹 🏈 🚭 💷 🕨 🍐 🌺 🌺 😫											
Project Navigator Files	Compilation Report - EqCmpD	emo 🖂  🔂 EqCmp4.b	if 🗵 💠 EqC	mp8.vhd 🛛	🔁 EqCmpDemo.bdf 🛛						
Files	🔁 🕟 🍳 🕊	A 🕩 📽 🕻	<u>ר</u> רו	$\backslash \backslash \mathbb{N}$	$\Box \bigcirc \land \land \land$	••••••••••••••••••••••••••••••••••••••					
EqCmpDemo.bdf						· · · · · · · · · · · · · · · · · · ·					
EqCmp8.vhd											
Tasks Compilation ▼ = 🗊 🗗 ×			EqCmp8								
Task											
? 🔺 🕨 Compile Design	SW[70]	VCC	input0[70]	cmpOut	OUTPUT						
? Analysis & Synthesis	SW[158]		input1[70]								
? D Fitter (Place & Route)			en cmn 8 inst								
? > Assembler (Generate programming			od_oub_o_ust		l						
7 D TimeQuest Timing Analysis											
/ P EDA Netlist Writer											
Edit Settings											
	4					• • • • • • • • • • • • •					
×											
B ID Message											
<b>P</b>											
<u>¥</u> System (27) Processing											
						0% 00:00:00					

Figura 68 – Interligação do módulo "EqCmp8" e dos portos de entrada e de saída e identificação dos diversos elementos do circuito no módulo "EqCmpDemo".

**12.** Volte a importar o ficheiro "master.qsf" com as definições dos pinos da FPGA na placa de desenvolvimento.

13. Efetue a síntese e implementação do projeto através do comando "Compile Design".

14. Programe a FPGA através do comando "Program Device".

**15.** Teste o comparador no *kit* de desenvolvimento aplicando diversos vetores de teste através dos interruptores usados e observando no LED o valor da saída.

**16.** Feche a aplicação de programação da FPGA e seguidamente o projeto.

# Para refletir

De acordo com a experiência adquirida com a realização deste guião, em que circunstâncias prefere usar VHDL e em que casos prefere a captura de diagramas esquemáticos (esquemas lógicos)?

# ТРС

**1.** Acrescente ao projeto da parte II deste guião um ficheiro que permita realizar na forma de um diagrama esquemático a instanciação e interligação da porta lógica AND e do inversor de forma a construir a porta NAND. Compile e teste o projeto resultante.

2. Valide por simulação o módulo "LogicUnit" da parte III deste guião.

**3.** Valide por simulação os módulos "EqCmp4" e "EqCmp8" da parte IV deste guião.

PDF criado em 10/02/2023 às 10:40:20