

Faculty of Information Technology Computer Systems Engineering Department Digital Lab ENCS 211 EXP. No. 7

Introduction to QUARTUSII Software

7.1 Objective:

Quartus II software is dedicated to program PLDs or FPGAs, in this experiment; we will study the schematic capture and HDL on Quartus II and then download simple examples on the FPGA using the Kit DE1.

<u>7.2 Pre-lab:</u>

Read the experiment before the lab,

1-Using Quartus II, create a schematic file for the function F=AB+C'D and simulate it.

2-Using Quartus II, create Verilog file for full Adder and simulate it.

7.3 Procedure:

Part 1: schematic capture:

The aim of this part is to design the combinational function: F=ABC+A'D

- Run the QUARTUSII software: double click on the QUARTUS II item in the desktop.
- Define the project : file \rightarrow new project wizard \rightarrow then press next
- Follow the windows as shown below:

New Project Wizard: Directory, Name, Top-Level Entity [page 1 of 5]		×
what is the working directory for this project?		
What is the name of this project?		
example1		
What is the name of the top-level design entity for this project? This name is case sensitive and mu exactly match the entity name in the design file.	.st	
example1		
Use Evisting Resident Cattings		
Use Existing Project Setungs		
C Back Nevts Finish Cano	-ol	1
Carc Next Printsh Carc		

<u>File name:</u>	o the project. Not	te: you can always (add design files t	o the project late	er. Add
File name	Туре	Library Desi	gn entry/sy H	DL version	Add All
					<u>R</u> emove
					Properties
					Up
					Down
				>	
<		defends Phone in a	11 1.7	rino I	
Specify the path	names of any nor	i-derault libraries.	User Libra	illes	
Specify the path	names of any nor	i-derault libraries.	User Libra	iles	

Device family <u>F</u> amily: Cyclone II Devices: All Target device C <u>A</u> uto device selec G <u>S</u> pecific device selec	ted by the Fitter elected in 'Available devic	es' list	Show in 'Av. Package: Pin <u>c</u> ount: Speed grade Show ac HardCop	Any Any Any e: Any dvanced o ay compat	vice' list		Design Entr Tool name: Format: Run thi Simulation	ry/Synth	esis e> itomatically	to synth	esize the (current des	ign			Y
Ayailable devices: Name EP2C20AF484I8 EP2C20F256C6 EP2C20F256C7 EP2C20F256C8 EP2C20F256I8 EP2C20F484C8 EP2C20F484C7 EP2C20F484C8 EP2C20F484C8 EP2C20F484C8	Core y LEs 1.2V 18752 1.2V 18752	User I/ 315 152 152 152 152 152 315 315 315 315	Memor 239616 239616 239616 239616 239616 239616 239616 239616 239616	Embed 52 52 52 52 52 52 52 52 52 52 52	PLL 4 4 4 4 4 4 4 4 4 4 8 8		Tool name: Format: Run ga Timing Ana Tool name: Format: Run thi	ite-level : lysis (<non is tool au</non 	e> simulation { e> itomatically	automati • after co	cally after	compilation	1			• •
Companion device HardCopy:	to HardCopy device resou	rces Ne	xt > Fi	nish	Cancel	_					< Back		ext>	Finisl	n	Cance

- Change the working directory to indicate your folder that will contain your project.
- Then name this project and the top level design entity, you must notice that the name of the project and the high level design entity must have the same name as the module in your program (in the case of HDL programming), then press next.
- If you have already files to add, you can add them here, otherwise (if you want to build the design now) press next.
- The next window give us a chance to use a specific device (which we will use later), choose the Family cyclone II and the device EP2C20F484C7
- Since our project is simple, and no need to add any other Tools, we just press next.
- After that the following window appears:

C:/workspace/	
Project name:	example1
Top-level design entity:	example1
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Family name:	Cyclone II
Device:	EP2C20F484C7
EDA tools:	
Design entry/synthesis:	<none></none>
Simulation:	<none></none>
Timing analysis:	<none></none>
Operating conditions:	
Core voltage:	1.2V
Junction temperature range:	0-85 °C

• Finally press finish.

Now we will build our function a schematic file: File \rightarrow new



* X

• ×

 ~ 2

12 $\Box \circ$

To enter components of our design double click anywhere on the schematic window, or select the symbol tool

(The little and gate) as shown in the figure:

This opens the symbol window in which available libraries ,including the standard QUARTUSII library, open this library by clicking on the little plus sign next to it, then select primitives, and then select logic ,then select the gate you want in your implementation



You have to define the inputs and outputs of your implementation, and you do so by opening the symbol window \rightarrow primitive \rightarrow pin, the following figure shows all the design components that must now be connected:



- To connect the components of the previous figure , select the 90° thin line with the dots on its ends on the tool bar, this makes your cursor a wiring tool that can be used to connect you circuit. When done, disable the wiring tool by clicking the arrow on the tool bar.
- Rename input and output ports to the variable names of our design ,to name a pin, either double click it to open its pin properties window, or right –click it , and select properties from the pull-down menu that shows up.
- Save your design, and make sure it is named the same as your project, the following figure shows the completed block diagram of our design.



Compilation

After that we compile our design by either clicking on processing \rightarrow start compilation or click on

If there are errors fix them before proceeding.

Simulation

If there are no errors, File \rightarrow new , select Vector waveform File

New	\times
- SOPC Builder System	~
🖻 Design Files	
- AHDL File	
Block Diagram/Schematic File	
EDIF File	
- State Machine File	
- SystemVerilog HDL File	
- I cl Script File	
- Verilog HDL File	
Hexadecimal (Intel-Format) File	
En Verification/Debugging Files	
In-System Sources and Probes File	
CianalT an UL ania Analyzer File	
Signan ap it Logic Analyzer File	
AHDL Include File	
Plack Sumbal File	
Chain Description File	
Supersus Design Constrainte File	
Text File	
Text IIC	~
OK Conset	1

Then right-click on the leftmost side of the window (under Name), then click insert node or bus, as shown below

📷 exam	npl1/example1	.bdf	🛛 🕘 Cor	npilation Report - Flow Summary	🖸 Waveform1.vwf
.	Master Time E	Bar: 9.7 ns	Pointer:	150 ps Interval: -9.55 ns	Start: End:
ЪА Ж⊕	Nam	e Value at 9.7 ns	0 ps	10.0 ns 9.7 ns	20.0 n
■ #4 ₩ ₩ ₩ ₩		Cut Copy Paste Delete	Ctrl+X Ctrl+C Del		
NU N		Inserc Zoom Show All Hidden M	Jodes	Insert Waveform Divider Insert Copied Nodes	

Click on Node Finder and then on List (using the Filter pins: all)

-		-	A		Value at	P PS	10	U His I	20	5.0 ms
		2	Æ€	Name	9.7 ns		9.7	ns		
							T			
		Insert Node	e or Bus			×				
		Name:				ок				
		Туре:	INPUT		•	Cancel				
		Value type:	9-Level		-	Node Finder				
		Radix:	ASCII		-					
	Node Fir	ider							X	
	Named:	x		Filter: Pins:	all	•	Customize	List	OK	
	Look in:	lexample1				•	Include subentities	Stop	Cancel	
	Nodes Fo	und:				Selected Nodes	:			
	Name			Assignments	T	Name		Assignments T		
	₽A			Unassigned	lr					
	D B			Unassigned	lt –					
	D C			Unassigned	lt					
1g .	D D			Unassigned	lt C					
lin	P F			Unassigned	L					
to :er					>					

- Select all inputs and the output by clicking on (>>) (you can select one by one).
- You can select the intervals when you want the inputs to be one or zero, either by shadowing the interval, then press the one level in the tool bar, or by write clicking the name, then select value→count value, the change the start value, the end value, and the radix as shown in the following figure:
- Now save the file with the same name as your project and in the same folder.
- Simulate your file either by clicking processes \rightarrow start simulation or click on.

Routing the Project on the FPGA:

After verifying that the design works as expected we can install the cod on the hardware (FPGA).

- Double check that the selected device is the one available on the Kit DE1. This can be done using Assignments→ Device
- We need 4 Inputs and one output, so we can use 4 switches and a LED. We have to use the user manual of DE1 to figure out the location of the switches and the LEDs. the Table below show the location of the switches

Signal Name	FPGA Pin No.	Description
SW[0]	PIN_L22	Toggle Switch[0]
SW[1]	PIN_L21	Toggle Switch[1]
SW[2]	PIN_M22	Toggle Switch[2]
SW[3]	PIN_V12	Toggle Switch[3]
SW[4]	PIN_W12	Toggle Switch[4]
SW[5]	PIN_U12	Toggle Switch[5]
SW[6]	PIN_U11	Toggle Switch[6]
SW[7]	PIN_M2	Toggle Switch[7]



Assign PINs for the Inputs and Outputs. You can assign by right click the pin then choose

• Locate \rightarrow locate in assignment editor

< 🖉	Ø ,	🦻 🍪 🛛	🕮 ► 🕏 🏍 1	o 🛈 👗 🕘	🗞 🔁 🙆		
• ×	1	exampl17	example1.bdf*	🛛 🧭 Assi	gnment Editor*		
× ×		Category: Node Filter: Information: X	All Locations Pin PLL Comb. cell Register cell Comb. cell Register cell Comb. cell Register cell Comb. cell Comb. cell Register cell Comb. cell Comb. cell Comb. cell Register cell Comb.	- LI - LI. • specific nodes: • device for the curren	t node(s) and/or pin	(s).	
		×	Edit: XV	PIN_L22			
		7	То	Location	I/O Bank	I/O S	itandard Gene
		1	∎≥A	PIN_L22	▼ 5	3.3-\	LVTTL Dedic
				PIN_122	I/O Bank 5	Dedicated Clo	ock CLK4, LVD5CLK2p,
				PIN_M1 PIN_M2	I/O Bank I I/O Bank 1	Dedicated Clo	ick CLK2, LVDSCLK1P, ick CLK3 LVDSCLK1p
		<	1	PIN_M5	I/O Bank 1	Row I/O	LVDS15p, DPCLK1
				PIN_M6	I/O Bank 1	Row I/O	LVDS15n
				PIN_M18	I/O Bank 6	Row I/O	LVDS78p, DPCLK6

• Select Pin in the category and select a switch for input.

The Figure below shows the circuit with PIN assignments.



Re-compile the Project so that the new changes in the PINs take place.

0uartus II - C	·/worksnace/example1	- example1 - Fexa	mple1 cdfl								E	
Eile Edit Processir	ng <u>T</u> ools <u>W</u> indow	- evalupte t - Teva	mprenteurl									لماركار
🔔 Hardware Setup	D USB-Blaster [USB-0]				Mode	JTAG		•	Progress		0%	
Enable real-time I	SP to allow background prog	gramming (for MAX II dev	vices)									
🏓 Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP	
🛍 Stop	example1.sof	EP2C20F484	001B051B	FFFFFFF								
Auto Detect												
🗙 Delete												
🍰 Add File												
🞬 Change File												
Save File												
🗳 Add Device												
🜓 Up												
🔑 Down												
For Help, press F1	_*											

Download the Program on the FPGA Tools \rightarrow Programmer

Make sure that the currently selected hardware is USB-Blaster, and then click Start.

(Show the results to the instructor)

Part2: Verilog

Adder subtractor:

A and B are bytes. The circuit provides A + B when X = 0, and A - B if X = 1.

- 1. Create a new project as in part1
- 2. Make new verilog file: File --> New --> "Verilog HDL file"
- 3. Study, write, and compile and simulate the following program.

on Repo	t - Flow Summary 🔯 addsub.vwf
1	// begin from here
2	■ module addsub (
3	A, // fisrt input
4	B, // second input
5	X, // control input
6	res); // result output
7	// input declaration
8	input[3:0] A, B;
9	input X;
10	// ouput declaration
11	output res;
12	// By rule all the input ports should be wires
13	wire[3:0] A, B;
14	wire X;
15	// Output port can be a storage element (reg) or a wire
16	reg [4:0] res ;
17	//Code Starts Here
18	always
19	begin //begin of block
20	if (X == 0)
21	res = A + B;
22	else
23	res = A - B;
24	end // end of block
25	endmodule // End of Module

Modify the program to work as follows:

A and B are bytes. The circuit provides zero output when X=0, A + B when X = 1, and A

- B if X = 2, A*B when X=3, .

(Simulate it and show the results to the instructor)

Seven segment display decoder (driver):



Make sure that you have the file sevenseg.v in the same directory you work and it is compiled Correctly, and then create symbol by File \rightarrow create/update \rightarrow create symbol files for current file

Now create a new project and a schematic file. Insert the created symbol in this file

Knowing that the seven-segment displays are active- low. Modify the above code and then

1-Assign four switches as input

2- Assign one of the sevens segments to the output

Re-compile and then rout it on the FPGA (show the results to the instructor)



Keep the working code; you will need it in the next experiments