

## Digital System Design Lab Manual- 3700

COMPREHENSIVE TUTORIAL FOR SETTING UP DE10-LITE HW/SW PLATFORM

Instructor – Chris Myers Document Author - Vikas Rao

DE10-Lite reference manual - <u>https://www.terasic.com.tw/DE10-lite-user-manual</u>

## **Objective** -

This tutorial provides comprehensive information that will help you understand how to create a FPGA design and run it on your DE10-Lite development board. The following sections provide a quick overview of the design flow, explain what you need to get started, and describe what you will learn.

The standard FPGA design flow starts with design entry using schematics or a hardware description language (HDL), such as Verilog HDL or VHDL. In this step, you can create a digital circuit that is implemented inside the FPGA. The flow then proceeds through compilation, simulation, programming, and verification in the FPGA hardware.

This tutorial will not make you an expert, but at the end, you will understand basic concepts about Quartus Lite projects, entering design using HDL, compiling your design, and downloading it into the FPGA on your DE10-Lite development board.

Terasic DE10-Lite is a cost-effective Altera MAX 10 based FPGA board. The board utilizes the maximum capacity MAX 10 FPGA, which has around 50K logic elements(LEs) and on-die analog-to-digital converter (ADC). It features on-board USB-Blaster, SDRAM, accelerometer, VGA output, 2x20 GPIO expansion connector, and an Arduino UNO R3 expansion connector in a compact size. The kit provides the perfect system-level prototyping solution for industrial, automotive, consumer, and many other market applications.





## Index -

- Section 1 Download and install Altera Quartus Lite Software on Windows/Linux
- Section 2 Setup a new project in Quartus Lite Software
- Section 3 Implement a function using Verilog HDL
- Section 4 Simulate the Verilog circuit using ModelSim + Verilog test bench
- Section 5 Pin assignment and Constraint generation
- Section 6 Synthesize, Implement, Generate, and Program DE10-Lite board

## **Board Overview** –

• Go over the chapters 1 and 2 in DE10-user manual to understand the board components and the control panel setup.





## Section 1 - Download and install Altera Quartus Lite Software

 Browse to the hyperlink provided below and make sure that Lite edition and 18.1 release are selected. Also, check the appropriate OS box(Windows/Linux). Use any download method (Direct Download works faster) and make sure to download only the individual files marked in the next bullet. Create a basic account on Intel using UnID if prompted -> <u>http://fpgasoftware.intel.com/18.1/quartus\_lite</u>

#### Download Center for FPGAs

Design Software	
Embedded Software	Quartus Prime Lite Edition
Archives	Release date: September, 2018 Intel' Quartus' Prime
Licensing	Latest Release: v18.1 Design Software
Programming Software	Select edition: Lite
Drivers	Select release: 18.1 🔻
Board System Design	
Board Layout and Test	Operating System 👔 🖲 🎢 Windows 🔍 🐧 Linux
Legacy Software	
	Download Method 🍞 🔍 Akamai DLM3 Download Manager 🍞 🔍 Direct Download

2. Navigate to the Individual files tab and download the Quartus prime lite edition, ModelSim FPGA edition, along with MAX 10 FPGA device support.

ownload a	nd install instructions:More
ead Intel F	PGA Software v18.1 Installation FAQ
uick Start	Suide
Select A	11
V Qu	artus Prime Lite Edition (Free)
	Quartus Prime (includes Nios II EDS) Size: 1.7 GB MD5: F0D752D67B18C89FBC0043CEE676896D
	ModelSim-Intel FPGA Edition (includes Starter Edition) Size: 1.1 GB MD5: 7FDBE5899A9929AEDD517F410079AA35
De Yo	<b>vices</b> I must install device support for at least one device family to use the Quartus Prime software
	Arria II device support Size: 499.6 MB MD5: D87CA20C91596BC8C7BCE84253D956B7
	Cyclone IV device support Size: 466.6 MB MD5: 9E32B85F83A440604154BD729B143D5C
	Cyclone 10 LP device support Size: 266.1 MB MD5: 72AAE619D358FF6B8E42849B3BFCFADD
	Cyclone V device support Size: 1.1 GB MD5: 75F5029A9058F64F969496B016EE19D4
	MAX II, MAX V device support Size: 11.4 MB MD5: ED990775F76C35D308877F27A30B7555
~	MAX 10 FPGA device support Size: 330.9 MB MD5: E87E56DAB144529EFC515C2452F1B1FE
(	•

3. Select the QuartusLite setup file and Run as administrator for windows installation. For Linux, modify the \*.run file with executable permissions(chmod +x \*.run) and execute the \*.run on terminal as super user(sudo \.\*.run).

			Open		
DE10_Lite		•	Run as administrator		
Name	Date modifie	C	Import to Grammarly		
			Troubleshoot compatibility		
max10-18.1.0.625.qdz	12/5/2018 9:		Pin to Start		B
🗹 🐳 QuartusLiteSetup-18.1.0.625-windows.exe	12/5/2018 10		7-Zip	>	B
🐝 ModelSimSetup-18.0.0.614-windows.exe	12/6/2018 2:		CRC SHA	>	B

4. Press Yes and click Next in the QuartusLite setup installation prompt. Accept the agreement and click Next again. Select the appropriate installation folder or retain the default directory. Make sure the following boxes are marked(modelsim and MAX 10 device support) for installation. And Press Next and let the installation complete



5. The setup should also install ModelSim FPGA edition along with QuartusLite Software(If not, you can install it separately). After the installation, the setup will prompt for installation of USB Blaster 2 device driver which is required for FPGA programming. Click Next and finish the installation.



## Section 2- Setup a new project in Quartus Lite Software

You begin this section by creating a new Quartus project. A project is a set of files that maintain information about your FPGA design. The Quartus Settings File (.qsf) and Quartus Project File (.qpf) files are the primary files in a Quartus project. To compile a design or make pin assignments, you must first create a project.

1. Launch the Quartus software, select File > New Project Wizard. The Introduction page opens



- 2. Enter the following information about your project as shown in the next snapshot:
  - a. What is the working directory for this project? Enter a directory in which you will store your Quartus project files for this design.
  - b. For example, C:\intelFPGA\_lite\18.1\quartus\3700\_lab\
  - c. File names, project names, and directories in the Quartus software cannot contain spaces.
  - d. What is the name of this project? Type hexTo7Seg.
  - e. What is the name of the top-level design entity for this project? Type hexTo7Seg.
  - f. Create a directory if it isn't present.

S New Project Wizard	ł
----------------------	---

#### Directory, Name, Top-Level Entity

What is the working directory for this project?

C:\intelFPGA\_lite\18.1\3700\_lab

What is the name of this <u>p</u>roject?

#### hexTo7Seg

What is the name of the <u>t</u>op-level design entity for this project? This name is case sensitive and must exactly match the entity name in the design file.

hexTo7Seg

Use Existing Project Settings...

3. Select the Empty project template and click Next:

New Project Wizard				>
Project Type				
Select the type of pr	oject to create.			
Empty project				
Create new proje device, and EDA 1	t by specifying projection in the specifying projection is the settings.	t files and libraries, 1:	target device fami	ly and
O Project <u>t</u> emplate				
templates installe the <u>Design Store</u> .	d with the Quartus Pr	me software, or dow	vnload design ten	gu
	< <u>B</u> ack <u>N</u> ex	t > <u>F</u> inish	Cancel	Help

 $\times$ 

....

....

....

Click Next twice and navigate to the Family, Device and Board Settings. Select the specifics on the respective device target as mentioned below (Device – 10M50DAF484C7G). You can add the required Verilog files later or edit the files in the Quartus editor once the project has been setup.

Device	Board								
Select the You can in	family and dostall addition	evice you want to ta al device support w	irget for co ith the Ins	ompilation. tall Devices comn	nand on the 1	Fools menu.			
To determ	nine the version	on of the Quartus P	rime softw	are in which your	target device	e is supported, refer to	o the <u>Device Support List</u> webpage.		
Device fa	mily					Show in 'Available o	devices' list		
<u>F</u> amily:	MAX 10 (DA/	DF/DC/SA/SC)			•	Pac <u>k</u> age:	Any		•
Dev <u>i</u> ce	: MAX 10 DA	١			•	Pin <u>c</u> ount:	Any		•
Target de	evice					Core speed grade:	Any		•
O <u>A</u> uto	device selec	ted by the Fitter				Name filter:			
<u> </u>	ific device se	lected in 'Available	devices' lis	t		✓ S <u>h</u> ow advanced	d devices		
O <u>O</u> the	er: n/a								
A <u>v</u> ailable o	devices:								
N	lame	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded multiplier 9-bit elements	PLLs	^
10M50DA	AF484C6GES	1.2V	49760	360	360	1677312	288	4	2
10M50DA	AF484C7G	1.2V	49760	360	360	1677312	288	4	2
10M50DA	AF484C8G	1.2V	49760	360	360	1677312	288	4	2
10M50DA	F484C8GES	1.2V	49760	360	360	1677312	288	4	2 🗸
<									>

5. Select the simulation Tool as ModelSim-Altera(note – this is not just ModelSim but the one with Altera) and format as Verilog HDL. Retain the other Tool selections to default.

🕤 New Project Wizard					×
EDA Tool Setting	S				
Specify the other EDA too	ols used with the Quartus F	Prime software to o	deve	elop your project.	
EDA tools:					
Tool Type	Tool Name	Format(s)		Run Tool Automatically	
Design Entry/Synthesis	<none> -</none>	<none></none>	~	Run this tool automatically to synthesize the current design	
Simulation	ModelSim-Altera 🔹	Verilog HDL	-	Run gate-level simulation automatically after compilation	
Board-Level	Timing	<none></none>	•		
	Symbol	<none></none>	-		
	Signal Integrity	<none></none>	Ŧ		
	Boundary Scan	<none></none>	•		

6. Review the new project summary page and make sure all the specifics that you entered are reflected.

Summary	
When you click Finish, the project will be created with the foll	owing settings:
Project directory:	C:\intelFPGA_lite\18.1\3700_lab
Project name:	hexTo7Seg
Top-level design entity:	hexTo7Seg
Number of files added:	0
Number of user libraries added:	0
Device assignments:	
Design template:	n/a
Family name:	MAX 10 (DA/DF/DC/SA/SC)
Device:	10M50DAF484C7G
Board:	n/a
EDA tools:	
Design entry/synthesis:	<none> (<none>)</none></none>
Simulation:	ModelSim-Altera (Verilog HDL)
Timing analysis:	0
Operating conditions:	
Core voltage:	1.2V
Junction temperature range:	0-85 °C

7. Quartus Lite Project navigator will now reflect the instance as your top level design entity under the device name.



## Section 3 – Implement a function using Verilog HDL

 Once you have created and setup your project, the next step is to describe your design using a hardware description language. Go to File-->New→Select Verilog HDL File.



2. Once a new file is created, you will be able to enter your code into it. When you save it, you have to have the module name and file name match up. Remember that one of the file names must match the name of the project. Double check and make sure that the Save as type is Verilog HDL Files.



- 3. Every project in hardware needs a testbench to generate all necessary inputs and read outputs to ensure they are correct. This is very similar to writing test cases in software programming. The standard practice of naming a testbench is to add a "tb\_" in front of the name of the module you are testing. In this case, we are testing hex to seven segment display, so the name of the new Verilog HDL file is saved as "tb\_hexTo7Seg". A testbench is just another standard Verilog HDL File.
- 4. If you already have a file present in your directory, you can add/remove files to the project by navigating to "project->add/remove files in project". Select the files -> apply -> ok.

Settings - hexTo7Seg						_	
Category:						[	Device/Board
General	iles						
Files Libraries	Select the design file directory to the proje	s you want to incl ect.	ude in the	e project. Click Add All to add all	l design files ir	n the	project
IP Catalog Search Locations	File name:						Add
Design Templates							
✤ Operating Settings and Conditions	<b>`</b>	1				^	Add All
Voltage	File Name	Туре	Library	Design Entry/Synthesis Tool	HDL Versio	n	Remove
l emperature	tb_hexTo7Seg.v	Verilog HDL File		<none></none>	Default		
Incremental Compilation	hexTo7Seg.v	Verilog HDL File		<none></none>	Default		Up
✓ EDA Tool Settings							Down
Design Entry/Synthesis							Properties
Simulation							rioperaes
Board-Level							
VHDL Input							
Verilog HDL Input							
Default Parameters							
Timing Analyzer							
Assembler							
Design Assistant							
Signal Tap Logic Analyzer							
Power Analyzer Interface							
SSN Analyzer							
		W	Buy Soft	ware OK Cance	ad App	olv	Help

ī

## Section 4 - Simulate the Verilog circuit using ModelSim + Verilog test bench

 To setup the simulation Tool path for ModelSim-Altera, navigate to Assignments→ Settings and modify the simulation parameters as shown below.



2. Under EDA Tool settings, check on Simulation option and it should show the Tool name as ModelSim-Altera.

Settings - hexTo7Seg	:
Category:	Device/Board
General	Simulation
Files	Specify options for generating output files for use with other EDA tools.
Libraries	
✓ IP Settings	Tool name: ModelSim-Altera
IP Catalog Search Locatio	
Design Templates	Run gale-level simulation automatically after compilation
<ul> <li>Operating Settings and Conc</li> </ul>	FDA Netlist Writer settings
Voltage	
Temperature	Format for output netlist: Verilog HDL Time scale: 1 ps
<ul> <li>Compilation Process Setting</li> </ul>	Output directory: simulation/modelsim
Incremental Compilation	
<ul> <li>EDA Tool Settings</li> </ul>	Map illegal HDL characters     Enable glitch filtering
Design Entry/Synthesis	Options for Power Estimation
Simulation	
Board-Level	Generate Value Change Dump (VCD) file script Script Settings
<ul> <li>Compiler Settings</li> </ul>	Design instance name:
VHDL Input	
Verilog HDL Input	
Default Parameters	More EDA Netlist Writer Settings
Timing Analyzer	Nativel ink settings
Assembler	
Design Assistant	None
Signal Tap Logic Analyzer	O Compile test hench:
Logic Analyzer Interface	
Power Analyzer Settings	Use script to set up simulation:
SSN Analyzer	○ Script to compile test bench:
	More NativeLink Settings Reset
	W Buy Software OK Cancel Apply Help

- 3. **Output directory** can be e.g. /simulation/modelsim. Then these new directories are created under the working directory, and the different Modelsim output files (.vo, .sdo) are placed here.
- 4. Navigate to More EDA Netlist Writer settings and set the generate functional simulation netlist option to ON: click OK and Apply.

✓ Settings - hexTo7Seg			- U
Category:	De	vice/Board	
General Files Libraries ✔ IP Settings	More EDA Netlist Writer Settings Specify the settings for the EDA Netlist Writer options in your project.		×
IP Catalog Search Lo Design Templates	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	Show:	All
Vortage     Temperature     Vortage     Temperature     Compilation Process Se     Incremental Compila     VEDA Tool Settings     Design Entry/Synthe     Simulation     Board-Level     Vorting HDL Input     Verilog HDL Input     Verilog HDL Input     Default Parameters     Timing Analyzer     Assembler     Design Assistant     Eisend Too Looir Analyte	Name: Architecture name in VHDL output netlist Bring out device-wide set/reset signals as ports Disable detection of setup and hold time violations in the input registers of bi-directional pins Do not write top level VHDL entity Flatten buses into individual nodes Generate functional simulation netlist Generate third-party EDA tool command script for RTL functional simulation Generate third-party EDA tool command script for gate-level simulation Maintain hierarchy Truncate long hierarchy paths	Setting: Setting: Off Off Off Off Off Off Off Of	
Logic Analyzer Interfac Power Analyzer Setting SSN Analyzer	Description: Generate Verilog/VHDL netlist for functional or timing simulation with EDA simulation tools. V Netlist Wire does not generate a Standard Delay Format Output File (.sdo). If the device doe only the functional-simulation netlist is available.	Vhen this option is 'On', the EDA s not support timing simulation, OK C	then Reset All ancel Help

- 5. You can automate the modelsim execution (compile/simulate) by setting up the testbench parameters in quartus software as follows. Enter the information about the testbench file under NativeLink settings.
  - a. Select Compile test bench
  - b. Click Test Benches. The Test Benches dialog box appears.
  - c. Click New. The New Test Bench Settings dialog box appears.
  - d. Enter the test bench details from your design
    - i. The Test bench name can be any suitable name by your choice. This name will later appear in the Compile test bench list.
    - ii. The Top level module in test bench must be the name of the entity of your testbench file.
  - e. You can also limit the simulation end time by entering a time limit in terms (us/ps/ms) in the "end simulation at" dialog box.

General       Simulation         Files       Libraries         Libraries       Specify options for generating output files for use with other EDA tools.         IP Settings       IP Catalog Search Location         Design Templates       Edit Test Bench Settings for the selected test bench.         V Operating Settings and Conc       Test bench name: tb_hexTo7Seg         Voltage       Top level module in test bench: stimulus         Compilation Process       Incremental Comp         Specify       Specify         Design Entry/Synt       Existing         Simulation       Simulation period         Simulation       Simulation until all vector stimuli are used         V Compiler Settings       End simulation at:         Verilog HDL Input       Test bench and simulation files	•
Files       Specify options for generating output files for use with other EDA tools.         Libraries       IP Settings         IP Catalog Search Location       Edit Test Bench Settings         Design Templates       Edit test bench settings for the selected test bench.         Operating Settings and Conc       Test bench name: tb_hexTo7Seg         Top level module in test bench:       stimulus         Compilation Process       Incremental Components         Specify       Existing         Design Entry/Synt       Existing         Simulation       Simulation period         Simulation       Simulation until all vector stimuli are used         Compiler Settings       End simulation at:         Verilog HDL Input       Test bench and simulation files	•
<ul> <li>✓ IP Settings</li> <li>✓ Edit Test Bench Settings</li> <li>✓ Compilation Process</li> <li>✓ Test Bench name: [b_hexTo7Seg</li> <li>Top level module in test bench: stimulus</li> <li>✓ Test bench to perform VHDL timing simulation</li> <li>✓ Design Entry/Synt</li> <li>✓ Simulation</li> <li>✓ Board-Level</li> <li>✓ Compiler Settings</li> <li>✓ Hot Input</li> <li>✓ Periog HDL Input</li> <li>✓ Test bench and simulation files</li> </ul>	•
Design Templates       Edit test bench settings for the selected test bench. <ul> <li>Operating Settings and Cont</li> <li>Voltage</li> <li>Temperature</li> <li>Test bench name: tb_hexTo7Seg</li> <li>Top level module in test bench: stimulus</li> <li>Top level module in test bench: stimulus</li> <li>Use test bench to perform VHDL timing simulation</li> <li>Design Entry/Synt</li> <li>Simulation</li> <li>Board-Level</li> <li>Compiler Settings</li> <li>VHDL Input</li> <li>Test bench and simulation files</li> <li>End simulation files</li> <li>End simulation files</li> <li>Design HDL Input</li> <li>End simulation files</li> <li>End</li></ul>	
Voltage       Test bench name:       tb_hexTo7Seg         Temperature       Its tbench name:       tb_hexTo7Seg         Top level module in test bench:       stimulus         Compilation Process       Use test bench to perform VHDL timing simulation         Design Entry/Synt       Existing         Simulation       Board-Level         V Compiler Settings       WhDL Input         Verilog HDL Input       End simulation at:         Image: Set	
Temperature       Image: Test of level module in test bench: stimulus         Compilation Process       Image: Test of level module in test bench: stimulus         Incremental Componental Componentation       Specify         VEDA Tool Settings       Existing         Design Entry/Synt       Existing         Simulation       Board-Level         Compiler Settings       Image: Test bench and simulation at: Image: Section and test of tes	
Incremental Comr.       Specify       Use test bench to perform VHDL timing simulation	
Design Instance name in test bench:     NA       Design Entry/Synt     Existing       Simulation     Simulation period       Board-Level     Image: Simulation at:       Compiler Settings     End simulation at:       VHDL Input     End simulation fles	
Simulation Board-Level Compiler Settings VHDL Input Verilog HDL Input Verilog HDL Input	New
✓ Compiler Settings VHDL Input Verilog HDL Input     Test bench and simulation files	Edit
Verilog HDL Input Test bench and simulation files	Delete
Default Parameter	
Timing Analyzer File name: tb_hexTo7Seg.v Add	
Assembler     File Name     Library     HDL Version     Remove	
Signal Tap Logic Ana tb_hexTo7Seg.v Default Up	
Power Analyzer Setti Down	
Properties	Help
OK Cancel Help	

 To setup the simulation executable path, click on Tools> Options> EDA Tool Options and and set the Modelsim-Altera path to your installation directory.
 Options

Options		×								
Category:										
✓ General	EDA Tool Op	vtions								
EDA Tool Options	Specify the directory that contains the tool executable for each third-party EDA tool:									
Headers & Footers Settir	EDA Tool	Directory Containing Tool Executable								
✓ Internet Connectivity	Precision									
Notifications	Synplify									
✓ IP Settings	Synplify									
IP Catalog Search Loc	Active-HDL									
Design Templates License Setup	Riviera-P									
Preferred Text Editor	ModelSim									
Processing	QuestaSim									
Tooltip Settings ✓ Messages	ModelSi	C:\intelFPGA_lite\18.1\modelsim_ase\win32aloem\								
Colors Fonts	Use Nati	veLink with a Synplify/Synplify Pro node-locked license								
< >>		OK Cancel Help								

7. After you have finished coding up your modules, double click on Analysis & Synthesis under the Tasks pane. If you do not see Analysis & Synthesis, double check that Flow is set to Compilation. This will compile and synthesize your program(s). If there are no errors, you will see a pop-up saying the Analysis & Compilation was successful. If not, it will tell you your errors in the messages pane at the bottom of the screen. It is good habit to just review your warnings (if any) to ensure you have no latches or other design hazards.



- 8. Once the Analysis & Synthesis is successful, you can do a RTL Simulation. Go to Tools-->Run Simulation Tool-->RTL Simulation.
  - a. Now, if you have setup the testbench setup in nativelink settings as described before, the modelsim window will automatically compile/simulate the testbench mentioned there and will dump the port level signals from the top level module onto waveform.
  - b. If you have not mentioned any testbench settings, you can follow the steps from next bullet to compile/simulate the design in Modelsim.



9. Go to Compile-->Compile. Ensure that the Library is work and navigate to your project directory. Select all Verilog HDL Files that pertain to your project. This includes the testbench. Hit Compile and then Done.

ModelSim - INTEL FPGA STARTER EDITION 10.56								- 0	$\times$
File Edit View Compile Simulate Add Project Tools Layout Bookn	narks Window H	lelp 表 ※ 以 初 家			SI 🕱 🔶 i	A. 100 100	+ -	H + +	
		🖉 L2 HH 🏭 🗷	- <u>-</u>	.   100 ba 🛋 🗐 🗐	E# 12 1	M M 🐼	•••	1 nin * 2.X. ni	<u>.</u>
Layout Simulate									
ColumnLayout Default	å - 🝕								
🖽 Project				🖬 🗙 🔷 Objects 🚃				+	d' X
▼ Name △ StatutType Orde(Modified		<b>File</b> -		▼ Name	Vak	ue Kind Mo	de	[ 🖆 Now	1 ▶
	M Compile Source	Je riles			^	x Pack Ou			
	Library: work		•						
	Look in:	3710_lab	-	- 🗈 📸 🖛					
	<u>a</u> _	Name	^	Date modified	Туре				
	Quick access	db		7/27/2018 5:15 PM	File folder				
		incremental_db		7/27/2018 3:52 PM	File folder				
	Deskton	simulation		7/27/2018 3:54 PM	File folder				
		hexTo7Seg.v		7/27/2018 4:49 PM	V File				
	Libraries	tb_hexTo7Seg.v		7/27/2018 4:49 PM	V File				
A Transcript									a ×
# Errors: 0, Warnings: 0	This PC								-
<pre>vlog -reportprogress 300 -work work C:/intelFPGA_lite/18.0/qua # Model Technology ModelSim - Intel FPGA Edition vlog 10.5b Com</pre>									
<pre># Start time: 17:21:07 on Jul 27,2018 # vlog -reportprogress 300 -work work C:/intelFPGA lite/18.0/m</pre>	Network								
# Compiling module stimulus	THELWOIK	<		_	>				
# Top level modules:		Clanama: Dh. h	au Ta 72an u" "heu Ta 72an		Compile				
<pre># stimulus # End time: 17:21:07 on Jul 27,2018, Elapsed time: 0:00:00</pre>		File rafie.	Exitoraeg.v Hexitoraeg.		Dopo				
# Errors: 0, Warnings: 0		Hes of type. [HDL	Hies ( .v; .vi; .vnd; .vndi; .v	/no; .ndi; .vo; •	Done				
VSIM 7>	Compile select	ed files together D	efault Options Edit	Source					•
Now: 100 ps Delta: 0 sim:/hexTo7Seg									_
+ O Type here to search	🖨 e	🧟 🚺 🭕	ž 💽 💈	M	A	e ~ 🖷		5:21 PM 7/27/2018	

10. If you look at the bottom in the Transcript pane, you will see that it did compile and there were no errors. The work library should have all the files you just compiled. If not, repeat the previous step. Since the testbench does the signal generations and testing, double click on your testbench file.

ModelSim - INTEL FPGA STARTER EDITION 10.5b File Edit View Compile Simulate Add Project Tools Layout I	Bookmarks Window Help				- 0 ×
▶ • ☞ 🗑 🗇 香   ※ № 億 立立   ② • 林 計 🖬    🔳	o 1/0 🔲 🚛 🤌 🖄 🚟 🛺 🗎	🍇 🕒 🛊 🖛 🐳 📑	100 ps 🔶 🖹 🖹 😫 🎇	😃   🛅 🛐 😍 🕴 🏞 🍳	• ‡   <b>‡</b> - 🏤 🛔
Layout Simulate					
ColumnLayout AllColumns	🚱 🕰 - 🤹				
Project	,	+ d	🗙 斜 Objects		+ # ×
▼ Name △ Statu Type Orde Modifie	ed		▼ Name	Value Kind Mode	1 🗖 Now 🌶 🕨
			hex_input	zzzz Net In xxxxx Padk Out	
	M Start Simulation		×		
	Design VHDL Verilog Libraries SDF	Others	<u> ()</u>		
	* Name	Type Path	<b>_</b>		
	fill work     fill work     fill work     fill till till till till till till t	Library rtl_wo Module C:/nt Library \$MOD Library \$MOD	rk elFPGA_ib elFPGA_ib eLFPGA_ib eL_TECH/ eL_TECH/		
A Transcript	tera losim	Library \$MOD	EL_TECH/		<u>+</u> # ×
# Errors: 0, Warnings: 0	→ A altera_Insim_ver	Library \$MOD	EL_TECH/		<u> </u>
vlog -reportprogress 300 -work work C:/intelFPGA_lite/18.0		Library \$MOD	EL_TECH/ -		
<pre># Hodel lechnology HodelSim - Intel FFGA Edition viog 10.3 # Start time: 17:21:43 on Jul 27,2018</pre>	•		▶		
<pre># vlog -reportprogress 300 -work work C:/intelFPGA_lite/10 # Compiling module stimulus # # # # # # # # # # # # # # # # # # #</pre>	Design Unit(s)	Resolution	ult 💌		
<pre># Top Tevel modules: # stimulus</pre>					
# End time: 17:21:43 on Jul 27,2018, Elapsed time: 0:00:00			OK Cancel		
# Errors: 0, Warnings: 0					
VSIM 9>					
Now: 100 ps Delta: 0 work					-10
Type here to search	i 📄 🤤 💽 📶	<i>ø</i> 🕤 🗾	A 🕅	x <sup>e</sup> ^ 🐂 <i>(ii</i> ,	5:22 PM

11. Click Simulate> Start simulation . Under work select the module to be simulated and click ok.

ModelSim - INTEL FPGA S	TARTER EDITIO	N 10.5b										-	۵	$\times$
File Edit View Compile	Simulate A	Add Structure	Tools Layout I	Bookmarks	Window He	elp								
B • 🗃 🖬 🌤 😂   🤞	· 🐿 🖷 🔝		- M  ] [1  0	10	ALL 🎉 🛛 📚	i 🗱 🏢 👯 🖬	j 💁 🕯 '	(m m)   <u>1</u>	100 ps 😴 🖹 🕅	Ff 👿 🧰 🕴 📰 🖥	1 🎝 🔰 🚺	🗢 T 🗆	<b>.</b> • 🕅 🗄	•
Layout Simulate	-													
ColumnLayout AllColumn	5	•	<b>\$</b> . <b>\$</b> . <b>\$</b>	🖷 • 🚳										
🖉 sim - Default 🚃					,			+ a >	🔹 Objects 🚃					ar >
▼ Instance	Design unit	Design unit type	Top Category	Visibility	Total coverage	:			▼ Name	Value Kir	nd Mode		🕑 🗖 Now	
🖃 📕 stimulus	stimulus	Module	DU Instance	+acc=					🖬 🔶 x	ххххх Ра	dk Internal			
😥 🔟 de 1soc	hexTo7Seg	Module	DU Instance	+acc=					🖬 🔶 z	xxxxx Ne	t Internal			
	stimulus	Process	-	+acc=										
#vsim_capacity#	samulus	Capacity	Statistics	+acc=										
		,,												
We have a second second								4						
Project × 🖓 sim ×								<u>•</u>	P					
A Transcript														a >
* Top level modules:														-1
# stimulus														
# End time: 17:21:43 o	on Jul 27,20	018, Elapsed t	ime: 0:00:00											
<pre># Errors: 0, Warnings:</pre>	0													
VSIM 9> VSIM -gu1 -1 ms	im_transcrip	pt work.stimu	ime: 0:03:08											
# Errors: 0, Warnings:	: 0	io, Elapsea c	1202. 0103.00											
# vsim -gui -1 msim_tr	anscript wo	ork.stimulus												
# Start time: 17:22:50	) on Jul 27,	2018												
# Loading work.stimulu	15													
Loading work.nex10/2	eg													
VSIM 10>														_
Now: 0 ps Delta: 0	sim:/s	timulus												
Type here t	o search		J H	<b>_</b>	е (	2	Ø 🕤	5 🛃	M	RR /	<ul> <li>9</li> <li>6</li> </ul>	(小) 5:	22 PM	Ç

12. The next step is to add signals to the wave and show the wave if it is not already present. On the left, in the sim pane, right click on the testbench file which should be the top most file. Go to Add->To Wave->All signals in region.



13. In the Wave pane, you will see all the signals declared and used in the testbench.



Note - When dealing with signals that are many bits, it is easier to see its value as an unsigned integer/Hex rather than binary. To make this conversion, right click on the signal you want, go to Radix and choose the format you want.

14. You are now ready to simulate your program. The icons boxed in the below screenshot are used to run the testbench. The first icon is Restart which will reset the simulation as if you never ran it. This is helpful to rerun the simulation without recompiling everything. The Run Length allows you to enter a specific amount of time you want the program to run for. It defaults to pico-seconds, but nano-seconds is the best time to use. The icon Run right after the Run Length is to run your program for the amount of time specified in the Run Length. If you set Run Length to be 10 ns, each time you press Run, the program will continue for 10 ns. Continue Run will run the program until it terminates. The same is true for Run -All. All the programs in this class will terminate in less than one second. If you find yourself waiting for longer than a few seconds until the program terminates, hit the Stop button and recheck your logic. you will see the following screen once your program terminates. It shows you where the program terminated. To go back to the Wave, click on the Wave tab.

ModelSim - INTEL FPGA STARTER EDITION 10.5	b		-
<u>File Edit View Compile Simulate Add S</u>	st <u>r</u> ucture T <u>o</u> ols Layo <u>u</u> t Boo <u>k</u> marks <u>W</u> indow <u>H</u> elp		
🖹 • 🚅 🖬 🍲 🍜   🧎 🛍 🏛 🕰   (	੭ - Ѩ ╠  兩    ೫ ೫  ₪  ≰     ଝ  ミ ≪ ඵ ⊈ ६	\$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$ \$	1
🕇 🏞 🕇   🏦 - 🏤 🏦   🔲 🏢		▼ 創稿 参   ととうそうすう Layout Simulate 💌	
🛺 sim - Default 🗰 🗙	🚺 🏠 Objects 🕂 🛃 📩 📊 Wave - Default		
▼ Instance Design unit Desig	Name Value 🔀 🗖 Now 🗹 🕨	Msgs	
If stanulus stanulus Mook     If stanulus Mook     If stanulus Mook     If stanulus Proc     IntTIAL#13     stanulus Proc     If stanulus     If stanu	O → X xxxx Padk Internal	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	

15. Once you click Run, you should see something like this on your Wave with values propagated and reflected on all the intended signals in the wave.



16. If you expand or scroll through the Transcript pane, you will see the output of any \$display statements you have in your code.

C Transcript
VSIM 13> run -concinue
VSIM 14> run -all
# x=1,z=0110000
# x=2, z=1101101
# x=3, z=1111001
# x=4, z=0110011
# x=5, z=1011011
# x=6, z=1011111
# x=7, z=1110000
# x=8, z=1111111
# x=9, z=1111011
# x=a, z=1110111
# x=b, z=0011111
# x=c, z=1001110
# x=d, z=0111101
# x=e, z=1001111
# x=f,z=1000111



## Section 5 – Pin assignment and Constraint generation

In this section, you will make pin assignments for all the signals from the design which interact with outer world. Before making pin assignments, perform the following steps:

 Choose Processing > Start > Start Analysis & Synthesis in preparation for assigning pin locations. Click OK in the message window that appears after analysis and elaboration completes.

🕥 Quartus Prime Lite Edition - C:/intelFPGA	_lite/	18.1/3700_lab/hexTo7Seg - hexTo7	7Seg				
<u>E</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject <u>A</u> ssignments	P <u>r</u> oc	essing <u>T</u> ools <u>W</u> indow <u>H</u> elp					
🗋 🖪 🖌 🗇 🛍 🤊 🤊 hexTo7Seg	STOP	<u>S</u> top Processing	Ctrl+Shift+C				
Project Navigator 🖹 Files 🔹 🗸		Start <u>C</u> ompilation	Ctrl+L	mp	oilation Report - hexTo7Seg 🛛 🗵		
📂 Files		Analyze Current <u>F</u> ile					
👎 tb_hexTo7Seg.v		Start	•	Pa.	Start Hierarchy Elaboration		L
📅 hexTo7Seg.v		Update Memory Initialization File		*	Start Analysis & Elaboration		
	$\ominus$	Compilation <u>R</u> eport	Ctrl+R	۴,	Start Analysis & Synt <u>h</u> esis	Ctrl+K	
	$\ominus$	Dynamic Synthesis Report		25	Start Partition Merge		
	£	Power Analyzer Tool		1	Start <u>F</u> itter		
	Ш.	SS <u>N</u> Analyzer Tool		2	Start <u>A</u> ssembler		
		Pocoivo Compilation Status Notifia	ations	0	Start Timing Analyzer	Ctrl+Shift+T	
		• I IOM Subbicosed Hessage	Total logic elements	1	Start EDA <u>N</u> etlist Writer		
Tacks Compilation	n A	×	Total registers	10	Start <u>D</u> esign Assistant		
Tasks Compilation			Total pins	۶	Start <u>P</u> ower Analyzer	Ctrl+Shift+P	
lask	í		Total virtual pins	<b>₽</b> ₽	Start SSN Analyzer		
✓ ► Compile Design	-1		Total memory bits	M. Z.d	Start Rapid Recompile		
? Analysis & Synthesis			Total PLLs	12	Start Signal Probe Compilation	Ctrl+Shift+I	
Edit Settings			UFM blocks	► io	Start I/O Assignment Analysis	curronner	
Tiew Report	— 、		ADC blocks		Start Check & Save All Netlist Changes		
<ul> <li>Analysis &amp; Elaboration</li> </ul>	>	< >					
×					Start Equation Writer (Post-synthesis)		
🗖 All 🕴 🛕 🔺 🔽 💎 < <fil< th=""><th>ter&gt;:</th><th></th><th>Find 😽 Find Next</th><th></th><th>Start Equation Writer (Post-fitting)</th><th></th><th></th></fil<>	ter>:		Find 😽 Find Next		Start Equation Writer (Post-fitting)		
EVDE TD Message					Start Test Bench Template Writer		
• Command: quartus_	map	read_settings_files=	onwrite_sett		Start EDA Synthesis		alysis_and_elabo

2. To make pin assignments that correlate to the hex\_input[3:0] input pins and seven\_seg\_out[6:0] output pins, perform the following steps: Choose Assignments > Pin Planner, which opens the Pin Planner, a spreadsheet-like table of specific pin assignments. The Pin Planner shows the designs pins.

Report	08×		Top View - Wire Bond Pin Legend							Top View - Wire Bond		nd	Q B	
Report not availabl	e				(text		AX 10 -		DAF484	C7G		Symbol	Pin Type User I/O User assi Fitter assi Unbonde Reserved Other co DEV_OE	
Groups Report Tasks F Early Pin Plan Early Pin Plin Run I/O As Export Pin / Highlight Pins I/O Banks K	ning ∧ anning ssignm Assign				Com Ing Sel							8 € 9 0 0 0 0 0	DEV_CLR DIFF_n DIFF_p DQ DQS DQSB CLK_n CLK_p Other PLL Other du TDI	
* Named: * ~	🌜 Edit: 🔀	4										Filter	Pins: all	
Node Name           hex_input[3]           hex_input[2]           hex_input[1]           hex_input[1]           hex_input[1]           sevenout[6]           sevenout[5]           sevenout[3]           sevenout[3]           sevenout[3]	Direction Input Input Input Output Output Output Output Output Output	Location	I/O Bank	/REF Group // 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2	D Standari .5 Vault) .5 Vault) .5 Vault) .5 Vault) .5 Vault) .5 Vault) .5 Vault) .5 Vault) .5 Vault)	Reserved	rent Stren; 12mAult) 12mAult) 12mAult) 12mAult) 12mAult) 12mAult) 12mAult) 12mAult)	Slew Rate 2 (default) 2 (default) 2 (default) 2 (default) 2 (default)	lferential P.	ct Preserva				
- I and I and I and I														

3. In the Location column next to each of the node names, add the coordinates(pin numbers) as shown from the tables below for the actual values to use with your DE10-Lite board(check manual for any particular

DIGITAL SYSTEM DESIGN

pins you are looking for based on your design usage). Double-click in the Location column for any of the pins to open a drop-down list and type the pin number shown in the table. Alternatively, you can select the pin from a drop-down list. For example, if you type C12 and press the Enter key, the Quartus software fills in the full PIN\_C12 location name for you. The software also keeps track of corresponding FPGA data such as the I/O bank and VREF Group. Each bank has a distinct color, which corresponds to the top-view wire bond drawing in the upper right window.

- September 2 C:/intelFPGA lite/18.1/3700 lab/hexTo7Seg hexTo7Seg ٥ Search altera.com Eile Edit View Processing Tools Window Help Report *4 8 ×* Top View - Wire Bond Pin Legend MAX 10 - 10M50DAF484C7G Symbol Pin Type User I/O User assi. Fitter assi.. Unbonde. • Reserved ... Grou Other co... C E R n Q Q Groups Report DEV OE 48, DEV\_CLR ✓ Early Pin Planning ∧ DIFF n Early Pin Planning DIFF\_p Run I/O Assignm DQ Export Pin Assign S S DQS Pin Finder... DQSB > Filter: Pins: all ✓ S Edit: × Named: \* VREF Group Node Name Direction I/O Standard Current Strength Slew Rate Differential Pair trict Preservation I/O Bank Node Name hex\_input[3] hex\_input[2] hex\_input[0] seven\_seg\_out[6] seven\_seg\_out[5] seven\_seg\_out[3] seven\_seg\_out[3] seven\_seg\_out[2] seven\_seg\_out[2] Location Reserved I/O Standard 3.3-V LVTTL Current Stren 8mA (default) PIN\_c12 PIN\_d12 PIN\_c11 PIN\_c10 PIN\_c17 PIN\_d17 PIN\_e16 PIN\_c16 PIN\_c15 PIN\_e15 B7\_N0 Input Input Input Output Output Output Output Output Output 2 (default) 2 (default) 2 (default) 2 (default) 2 (default) 2 (default) seven\_seg\_out[1] seven\_seg\_out[0] -V LVTTI nA (default) 3.3-V LVTTL nA (default) 2 (default) 00:0 4:55 PM Type here to search 0 🖬 🧲 i 😒 📄 👩 💽 ^ 🖷 *(i*. 🕼 5 M
- 4. After filling pin numbers for all ports, close the pin planner window.

Signal Name	FPGA Pin No.	Description	I/O Standard
HEX00	PIN_C14	Seven Segment Digit 0[0]	3.3-V LVTTL
HEX01	PIN_E15	Seven Segment Digit 0[1]	3.3-V LVTTL
HEX02	PIN_C15	Seven Segment Digit 0[2]	3.3-V LVTTL
HEX03	PIN_C16	Seven Segment Digit 0[3]	3.3-V LVTTL
HEX04	PIN_E16	Seven Segment Digit 0[4]	3.3-V LVTTL
HEX05	PIN_D17	Seven Segment Digit 0[5]	3.3-V LVTTL
HEX06	PIN_C17	Seven Segment Digit 0[6]	3.3-V LVTTL
HEX07	PIN_D15	Seven Segment Digit 0[7], DP	3.3-V LVTTL
HEX10	PIN_C18	Seven Segment Digit 1[0]	3.3-V LVTTL
HEX11	PIN_D18	Seven Segment Digit 1[1]	3.3-V LVTTL
HEX12	PIN_E18	Seven Segment Digit 1[2]	3.3-V LVTTL
HEX13	PIN_B16	Seven Segment Digit 1[3]	3.3-V LVTTL

HEX14	PIN_A17	Seven Segment Digit 1[4]	3.3-V LVTTL
HEX15	PIN_A18	Seven Segment Digit 1[5]	3.3-V LVTTL
HEX16	PIN_B17	Seven Segment Digit 1[6]	3.3-V LVTTL
HEX17	PIN_A16	Seven Segment Digit 1[7], DP	3.3-V LVTTL
HEX20	PIN_B20	Seven Segment Digit 2[0]	3.3-V LVTTL
HEX21	PIN_A20	Seven Segment Digit 2[1]	3.3-V LVTTL
HEX22	PIN_B19	Seven Segment Digit 2[2]	3.3-V LVTTL
HEX23	PIN_A21	Seven Segment Digit 2[3]	3.3-V LVTTL
HEX24	PIN_B21	Seven Segment Digit 2[4]	3.3-V LVTTL
HEX25	PIN_C22	Seven Segment Digit 2[5]	3.3-V LVTTL
HEX26	PIN_B22	Seven Segment Digit 2[6]	3.3-V LVTTL
HEX27	PIN_A19	Seven Segment Digit 2[7], DP	3.3-V LVTTL
HEX30	PIN_F21	Seven Segment Digit 3[0]	3.3-V LVTTL
HEX31	PIN_E22	Seven Segment Digit 3[1]	3.3-V LVTTL
HEX32	PIN_E21	Seven Segment Digit 3[2]	3.3-V LVTTL
HEX33	PIN_C19	Seven Segment Digit 3[3]	3.3-V LVTTL
HEX34	PIN_C20	Seven Segment Digit 3[4]	3.3-V LVTTL
HEX35	PIN_D19	Seven Segment Digit 3[5]	3.3-V LVTTL
HEX36	PIN_E17	Seven Segment Digit 3[6]	3.3-V LVTTL
HEX37	PIN_D22	Seven Segment Digit 3[7], DP	3.3-V LVTTL
HEX40	PIN_F18	Seven Segment Digit 4[0]	3.3-V LVTTL
HEX41	PIN_E20	Seven Segment Digit 4[1]	3.3-V LVTTL
HEX42	PIN_E19	Seven Segment Digit 4[2]	3.3-V LVTTL
HEX43	PIN_J18	Seven Segment Digit 4[3]	3.3-V LVTTL
HEX44	PIN_H19	Seven Segment Digit 4[4]	3.3-V LVTTL
HEX45	PIN_F19	Seven Segment Digit 4[5]	3.3-V LVTTL
HEX46	PIN_F20	Seven Segment Digit 4[6]	3.3-V LVTTL
HEX47	PIN_F17	Seven Segment Digit 4[7], DP	3.3-V LVTTL
HEX50	PIN_J20	Seven Segment Digit 5[0]	3.3-V LVTTL
HEX51	PIN_K20	Seven Segment Digit 5[1]	3.3-V LVTTL
HEX52	PIN_L18	Seven Segment Digit 5[2]	3.3-V LVTTL
HEX53	PIN_N18	Seven Segment Digit 5[3]	3.3-V LVTTL
HEX54	PIN_M20	Seven Segment Digit 5[4]	3.3-V LVTTL
HEX55	PIN_N19	Seven Segment Digit 5[5]	3.3-V LVTTL
HEX56	PIN_N20	Seven Segment Digit 5[6]	3.3-V LVTTL
HEX57	PIN_L19	Seven Segment Digit 5[7], DP	3.3-V LVTTL

Signal Name	FPGA Pin No.	Description	I/O Standard
SW0	PIN_C10	Slide Switch[0]	3.3-V LVTTL
SW1	PIN_C11	Slide Switch[1]	3.3-V LVTTL
SW2	PIN_D12	Slide Switch[2]	3.3-V LVTTL
SW3	PIN_C12	Slide Switch[3]	3.3-V LVTTL
SW4	PIN_A12	Slide Switch[4]	3.3-V LVTTL
SW5	PIN_B12	Slide Switch[5]	3.3-V LVTTL
SW6	PIN_A13	Slide Switch[6]	3.3-V LVTTL
SW7	PIN_A14	Slide Switch[7]	3.3-V LVTTL
SW8	PIN_B14	Slide Switch[8]	3.3-V LVTTL
SW9	PIN_F15	Slide Switch[9]	3.3-V LVTTL

5. The <design>.qsf file should reflect these pin assignments once you have set them on pin planner.

set\_location\_assignment PIN\_C14 -to seven\_seg\_out[o] set\_location\_assignment PIN\_E15 -to seven\_seg\_out[1] set\_location\_assignment PIN\_C15 -to seven\_seg\_out[2] set\_location\_assignment PIN\_C16 -to seven\_seg\_out[3] set\_location\_assignment PIN\_E16 -to seven\_seg\_out[4] set\_location\_assignment PIN\_D17 -to seven\_seg\_out[5] set\_location\_assignment PIN\_C17 -to seven\_seg\_out[6] set\_location\_assignment PIN\_C10 -to hex\_input[0] set\_location\_assignment PIN\_C11 -to hex\_input[1] set\_location\_assignment PIN\_D12 -to hex\_input[2] set\_location\_assignment PIN\_C12 -to hex\_input[3]

6. Timing settings are critically important for a successful design. For this tutorial you will create a basic Synopsys Design Constraints File (.sdc) that the Quartus TimeQuest Timing Analyzer uses during design compilation. For more complex designs and requirements, you will need to create your .sdc files by considering the timing requirements more carefully.

To create an SDC, perform the following steps:

- a. Open the TimeQuest Timing Analyzer by choosing Tools > TimeQuest Timing Analyzer.
- b. Choose File > New SDC file. The SDC editor opens in the quartus software with a file extension as .sdc.
- c. Type in your timing constraints(clock/pll) into the file and save it as top-level file(hexTo7Seg.sdc in this case). Since the current design doesn't have a clock, we will ignore the constraints. Refer the DE10-Lite manual for creating a PLL/clock as it is explained in detail.
- d. Naming the SDC with the same name as the top-level file except for the .sdc extension causes the Quartus software to using this timing analysis file automatically by default. If you used another name, you would need to add the SDC to the assignments file list.

# Section 6 – Synthesize, Implement, Generate, and Program DE10-Lite board

After creating your design you must compile it to convert the design into a bitstream that can be downloaded into the FPGA. The most important output of compilation is an SRAM Object File (.sof), which you use to program the device. The software also generates other report files that provide information about your code as it compiles. If you want to store "\*.SOF" in memory device (such as flash or EEPROMs), you must first convert the SOF to a file type specifically for the targeted device.

Now that you have created a complete Quartus project and entered all assignments, you can compile the design.

- 1. In the task window double click compile design or
- 2. In the Processing menu, choose Start Compilation or
- 3. click the Play button on the toolbar (cntrl+L).

The Quartus Messages window displays many messages during compilation. It should not display any critical warnings; it may display a few warnings that indicate that the device timing information is preliminary or that some parameters on the I/O pins used for the LEDs were not set. The software provides the compilation results in the Compilation Report tab as shown.



- 4. After compiling and verifying your design you are ready to program the FPGA on the development board. You download the SOF you just created into the FPGA using the USB-Blaster circuitry on the board. Set up your hardware for programming using the following steps:
  - a. For the DE10-Lite board, connect the USB-Blaster (included in your development kit) to pin J3 and the other end of the USB cable to the host computer.
- 5. Program the FPGA using the following steps.
  - a. Choose Tools > Programmer. Once the Programmer window opens, click Hardware Setup. If it is not already turned on, turn on the DE10-Lite [USB-0] option under currently selected hardware.

Programmer - C:/intelFPGA_lite/18.1/3700		- [		$\times$			
<u>File Edit View Processing Tools Windo</u>	ow <u>H</u> elp			Se	arch altera.	com	9
Lardware Setup No Hardware		Mode: JTA	G <b>*</b>	Progress:			
Enable real-time (Contraction to the land)	·	. de (l'alla l'a			;	<	
Hardware Settings         Hardware Settings         Stop         Select a programming hardware setup applies	JTAG Settings nardware setup only to the curr	to use when prog ent programmer	ramming devices. This prog window.	ramming			
Add File     Currently selected hard     Available hardware ite	lware: USB-Bl	aster [USB-0]			•		
Hardware Change Fil Save File Add Device	Server Local	Port USB-0		Add Hard	<b>Iware</b> ardware		

- b. Make sure the Mode is set to JTAG.
- c. Click Auto Detect to detect all the devices on the JTAG chain. Select the device as 10M50DA which reflects the device ID for DE10-Lite MAX-10 board. Click OK after the selection.

roject Navigator 🖹 Files	Nogramme	r - C:/intelFPGA_lite/18.	1/3700_lab/hexTo7Seg - hexTo7Seg - [	hexTo7Seg.cdf]*		×
Files	Eile Edit View	v Processing Tools	Window Help	Search altera.com	m • ×	
₱ tb_hexTo7Seg.v ₱ hexTo7Seg.v	A Hardware S	birectory ection Available				
asks Compilation Task Compile Design > Analysis & Symthesis > First Place & Route) > Assembler (Generate pr > Son Netlist Writer > Do Netlist Writer	<ul> <li>→<sup>k</sup> Start</li> <li>dt Stop</li> <li>Auto Dete</li> <li>× Delete</li> <li>Add File,</li> <li>Change File</li> <li>Add Device</li> </ul>	File D	evice hecksun Jsercodi Yogram onfigur Select Device Found devices with shared JTA ToMSODA 10MSODAE 10MSODC	Yerlif, Bank, 'Damini, iecurite' Jaar, TSP Deck Bit LAM	4 × 	unctions ce Protocols y Interfaces and Controllers sors and Perpherals sity Program r Partner IP
All O A A V CVnr Th Messare Quartus Prim 233000 Quartus Prim	P-Op P-Down			ОК		

DIGITAL SYSTEM DESIGN

6. Select FPGA device and click Add File to program the FPGA with the relevant .sof file(hexTo7Seg.sof in this case).



7. click Start to program the .sof file into FPGA and make sure the progress bar reflects success and not failure.

Programmer - C/intelFPGA_lite/18.1/3700_lab/hexTo7Seg - hexTo7Seg.cdf)* Eile Edit View Processing Iools Window Help										– 🗗 Search altera.co	× m •		
🚣 Hardware	Setup USB-Blaster [USB-0]							Mode	JTAG		<ul> <li>Progre</li> </ul>	ss: 100% (Succ	essful)
Enable real-time ISP to allow background programming when available													
▶ <sup>%</sup> Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine	Security Bit	Erase	ISP CLAMP		
n Stop	output_files/hexTo7Seg.sof	10M50DAF484	00271916	00271916	<b>∠</b>								
Auto Dete													
× Delete													
🌥 Add File.													
Change Fil													
😫 Save File													
Add Device	10M50DAF484												
t <sup>®</sup> Up	•												
业 Down													
	Type here to search	Q.	di 🤇	Contraction	<b>i</b>	<i>🐠</i> 💽	N	M 🔼		A	ዮ ^ 📼 🕼	7:11 PM 12/6/2018	-

- 8. When you verify the design in hardware, observe the runtime behavior of the FPGA hardware and ensure that it is functioning appropriately.
  - a. Play around with the switches (SWo-SW<sub>3</sub>) and see the 7-segment display varying .

### "Congratulations, you have created, compiled, and programmed your first FPGA design! The compiled SRAM Object File (.sof) is loaded onto the FPGA on the development board and the design should be running."

#### References and miscellaneous links -

- Altera Quartus Lite software: <u>http://fpgasoftware.intel.com/18.1/quartus\_lite</u>
- <u>User manual/datasheets/demonstrations/control-panel/sample codes:</u>
  - Click the link below (<u>http://DE10-Lite.terasic.com/cd</u>) and download the latest system build(<u>DE10-Lite v.2.0.3 SystemCD.zip</u>) with Quartus 16.o(>) support.
- For enabling inbuilt Linux Subsytem in windows please follow the instructions mentioned in the article, it works like a charm: <u>https://www.windowscentral.com/how-install-bash-shell-command-line-windows-10</u>