Department of Electrical & Electronic Engineering
Imperial College London

Information Processing

Lab 1 – Introduction to DE10-Lite and Setting up Quartus Prime Lite

Objectives

By the end of this experiment, you should have achieved:

- Setting up the Quartus Prime Lite to run on your machine;
- Create a directory structure for this and subsequent Labs;
- Create a new project in Quartus and complete a basic 7-segment LED display decoder design using Quartus and Verilog from start to finish;
- Program the Max 10 FPGA chip on the DE10-Lite board with your design;
- Create another project to perform hexadecimal to BCD decoding;
- Explore and test your decoder design.

Installing Quartus Prime Lite

You will be using Intel/Altera software known as **Quartus Prime Lite**. There are two ways you may achieve this: 1) install the virtual machine Virtual Box and run Quartus Prime Lite under Ubuntu on your laptop; 2) if you are running Windows 10 or Ubuntu, install Quartus Prime Lite from Intel/Altera natively on your machine without running any virtual machine at all. **Method 1** is platform independent but will run Quartus slower. If you are a Mac user[1], this is the best way! **Method 2** is only for those who use Windows 10 or Linux on their laptops and want to run Quartus software natively and therefore should be faster. In either case, you will need to have at least 20GB free disk space and at least 4GB of RAM (preferably 8GB of RAM) on your machine.

Method 1

Step 1: Download the following files:

a) Virtual Box from https://www.virtualbox.org/wiki/Downloads

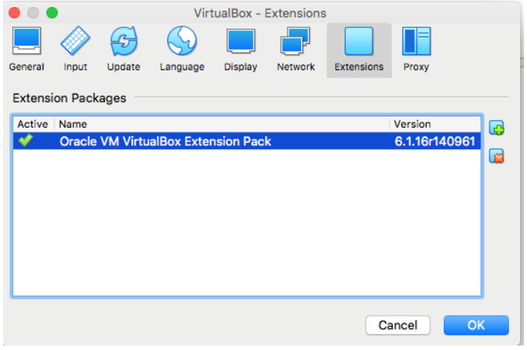
b) VirtualBox Extension Pack from: https://www.virtualbox.org/wiki/Downloads

c) Ubuntu and Quartus Prime Lite virtual machine image (.ova) from:
https://imperialcollegelondon.app.box.com/s/kg445fq4lmh393t964btnf0pq3xtgq31
(File name: EE2-Ubuntu18-2020-10-02….ova, 12.9GB)

These are large files. Make sure that you go somewhere with good WIFI connection. You only need to download these files once. If want to reduce hard disk usage, you may off load these files to a USB or external drive.

[1] If you are using Big Sur, you will not be able to use VirtualBox

Step 2: Install Virtual Box on your laptop, followed by the Extension Pack. To install the Extension Pack, which is required for USB2/USB3 interfaces, you should go to Preferences. Select Extension and then Add … Then select the Extension Pack.

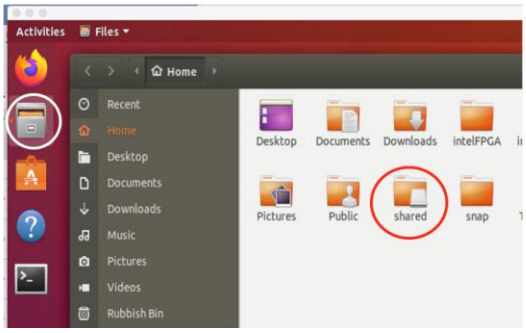


Extension Packages

Active	Name	Version
✓	Oracle VM VirtualBox Extension Pack	6.1.16r140961

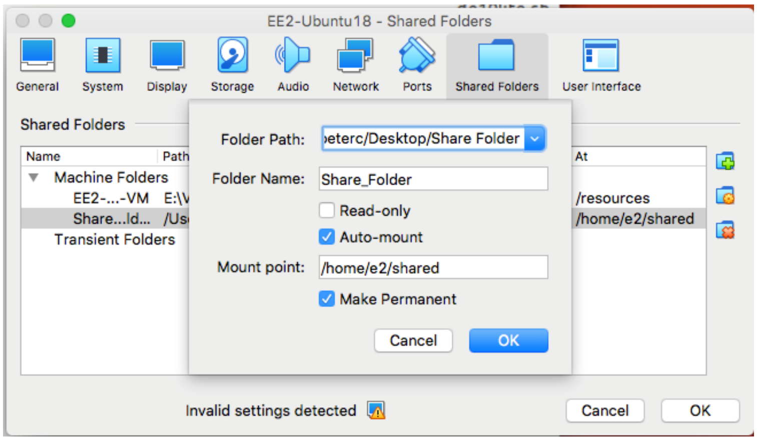
Step 3: Run the Virtual Box application and import the virtual machine image (.ova file). This may take a few minutes. Now start the virtual machine.

Step 4: Log onto Ubuntu with **user name**: e2 and **password**: e2. You should see a desktop similar to the one here.



Step 5: First make a shared directory on your host system. I suggest you put this on the host system Desktop. We now create a shared folder between your host OS and Ubuntu. Click on the Drawer icon, and then Home. Create in this directory a new folder called “shared” as shown.

Now go to Virtual Box Top menu bar and click Machine -> Settings. A window will pop up and click Shared Folders icon and complete the fields in the popup form as shown. Folder Path is where you put shared files on your laptop host. You should use /home/e2/shared as your mount point. You can now transfer files (such as screen images) between your laptop OS and Ubuntu.



Shared Folders

Name	Path	At
Machine Folders		
EE2-...-VM	E:\V	/resources
Share...ld...	/Us	/home/e2/shared
Transient Folders		

Folder Path: ›eterc/Desktop/Share Folder

Folder Name: Share_Folder

☐ Read-only

☑ Auto-mount

Mount point: /home/e2/shared

☑ Make Permanent

Invalid settings detected

Step 6: Start a terminal window on Ubuntu and enter the command: **quartus** (RETURN). Quartus should now start running and you are ready to go!

Tips: If you found that the VM window display is too small, change the viewing option with: **View > Scaled Mode**.

Depending on the resolution of your display and monitor, you may have to change the Video Memory size from 16MB to 64MB.

Method 2:

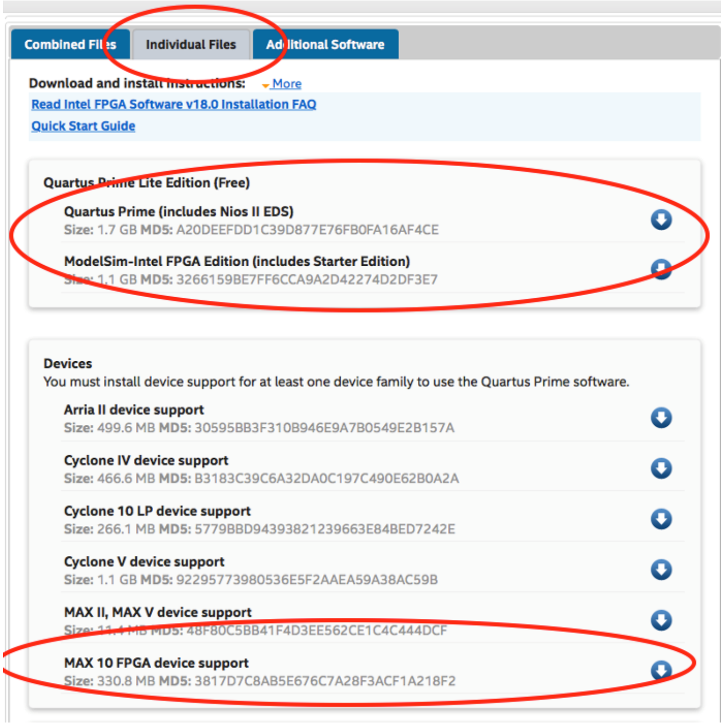
This method is only applicable if you have a Windows 10 or a Linux machine.

Step 1: Go to https://fpgasoftware.intel.com/18.0/?edition=lite&platform=windows

Do not download the entire combined Quartus Prime Lite – this will take huge amount of disk space. Instead, select the Individual Files tab, and download the three files circled in red.

Step 2: You will be asked to complete a registration form and create your own individual account. Make sure you use College's email address. You would then need to verify your identity through your College email.

Step 3: Go back to the link shown in Step 1, and now you should be able to download and install the Quartus Prime Software, the ModelSim simulator, and the Max10 device support.



Download and install instructions: More

Read Intel FPGA Software v18.0 Installation FAQ

Quick Start Guide

Quartus Prime Lite Edition (Free)

Quartus Prime (includes Nios II EDS)
Size: 1.7 GB **MD5:** A20DEEFDD1C39D877E76FB0FA16AF4CE

ModelSim-Intel FPGA Edition (includes Starter Edition)
Size: 1.1 GB **MD5:** 3266159BE7FF6CCA9A2D42274D2DF3E7

Devices
You 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:** 30595BB3F310B946E9A7B0549E2B157A

Cyclone IV device support
Size: 466.6 MB **MD5:** B3183C39C6A32DA0C197C490E62B0A2A

Cyclone 10 LP device support
Size: 266.1 MB **MD5:** 5779BBD94393821239663E84BED7242E

Cyclone V device support
Size: 1.1 GB **MD5:** 92295773980536E5F2AAEA59A38AC59B

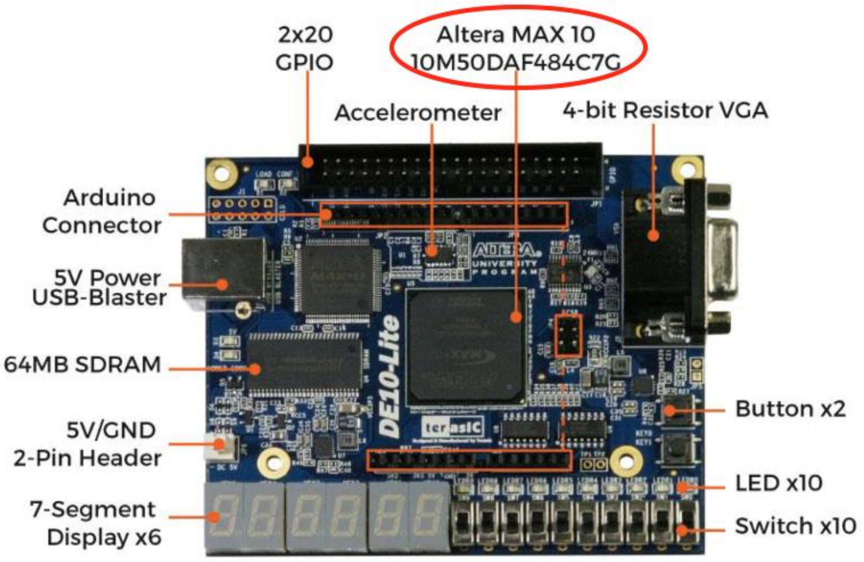
MAX II, MAX V device support
Size: 11.4 MB **MD5:** 48F80C5BB41F4D3EE562CE1C4C444DCF

MAX 10 FPGA device support
Size: 330.8 MB **MD5:** 3817D7C8AB5E676C7A28F3ACF1A218F2

Step 4: Launch Quartus Prime Lite programme by clicking the icon. The software should start. You are now ready to go!

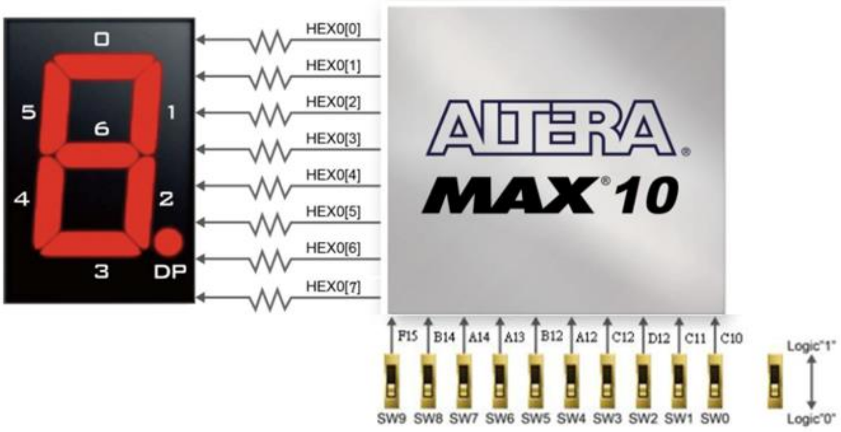
The DE10-Lite FPGA Board

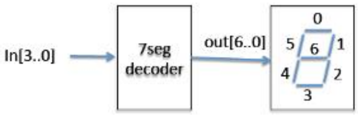
You have been loaned a DE10-Lite. Connect the DE10 board to your machine using the USB cable provided. You should see the board LED displays cycling through all digits, showing that the board is working properly. The diagram below shows the features provided on this board. Note that the FPGA chip is 10M50DAF484C7G.

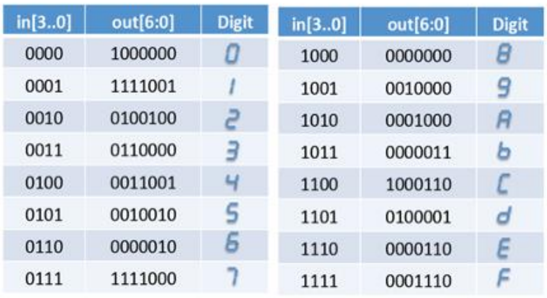


Task 0: Preparation

In task 1, you will use four slide switches on the right (SW3 to SW0) on the DE10 board as input, and display the 4-bit binary number as a hexadecimal digit on the right-most 7-segment display (HEX0). In task 0, you will check that everything is working properly.



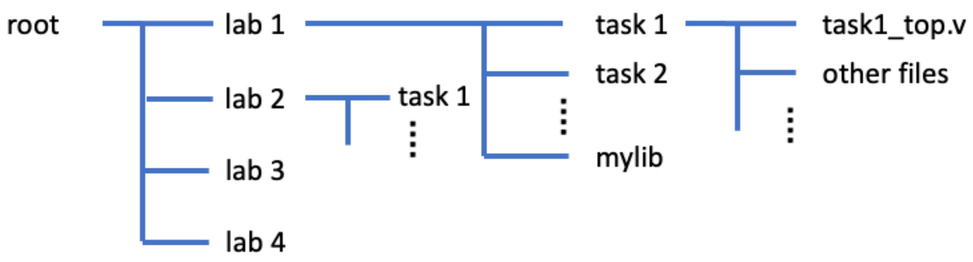




in[3..0]	out[6:0]	Digit	in[3..0]	out[6:0]	Digit
0000	1000000	0	1000	0000000	8
0001	1111001	1	1001	0010000	9
0010	0100100	2	1010	0001000	A
0011	0110000	3	1011	0000011	b
0100	0011001	4	1100	1000110	C
0101	0010010	5	1101	0100001	d
0110	0000010	6	1110	0000110	E
0111	1111000	7	1111	0001110	F

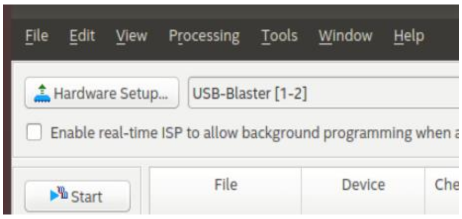
Step 1: Creating a good directory structure

Before you start carrying out any design for this Lab, it would be very helpful if you first create a directory structure for the labs. A possible directory structure is shown below:



If you are using the virtual machine, create a directory structure under /home/e2/E2_CAS. If you are using a Windows 10 machine, create this in a convenient location such as C:/E2_CAS/.

Step 2: See what you are aiming for



Go to the course webpage and download a copy of the solution for Exercise 1: “**lab1task1_sol.sof**” to your folder for Lab3 (or wherever that is). Make sure that your DE10 board is plugged in and running.

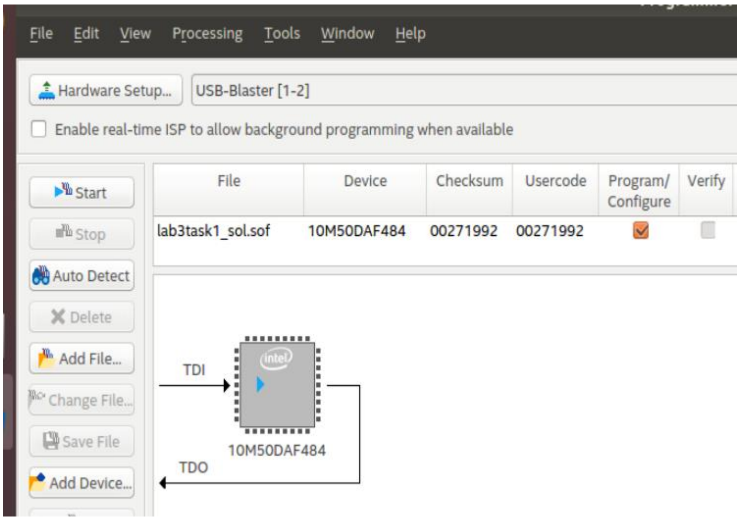
Step 3: Setting up programming hardware

Run Quartus software on your computer. Click command: **Tools > Programmer**. In the popup window, click: **Hardware Setup ….** You should see something like the diagram on the right. Then select: **USB Blaster**. This is to tell Quartus software that you are using the DE10 interface to program (or blast) the FPGA.

If you do not see USB-Blaster under Hardware Setup although the DE10-Lite is plugged in properly, it is most likely that the Device Driver had not been installed or loaded properly. In which case, please see the instruction in the Appendix.

Step 4: Blasting the FPGA

Next click the **AddFile** button. Navigate to the folder containing the **lab1task1_sol.sof** file. Select this. You should see a display like this:



USB-Blaster [1-2]

□ Enable real-time ISP to allow background programming when available

File	Device	Checksum	Usercode	Program/ Configure	Verify
lab3task1_sol.sof	10M50DAF484	00271992	00271992	☑	□

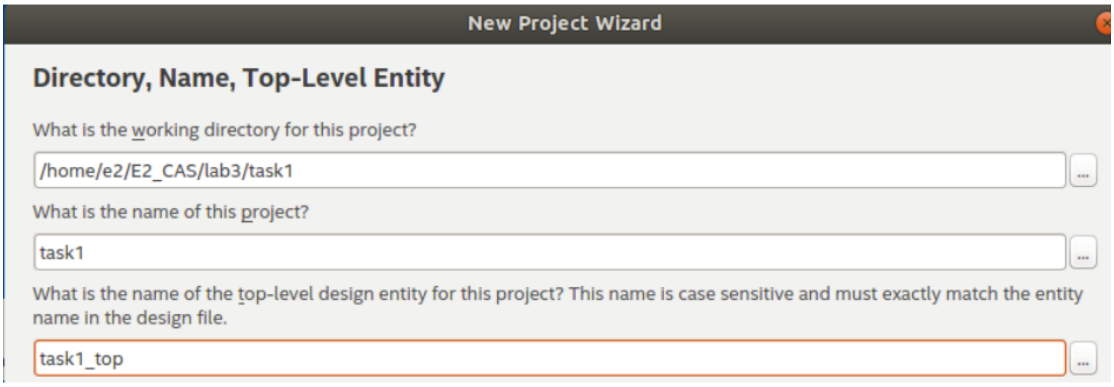
Note that the device indicated here is the one on the DE10 board: 10M50DAF484. Click the **Start** button.

The **lab1task1_sol.sof** file contains the solution to Task 1. It has the bit-stream to configure (or programme/blast) the FPGA Max10 chip. Once the bit-stream is successfully sent to the FPGA chip, the task1 design will take over the function of the chip. You should be able to change the least significant four switches and see a hexadecimal number displayed on rightmost 7-segment display. Now you are ready to create this design from scratch.

Task 1: The Design Flow – 7 Segment LED Display

Step 5: Create the project "task1"

- Create in your home directory the folder **../lab1/task1**.
- Click **file>New Project Wizard**, complete the form. Use **task1** as the project name and **task1_top** as top-level design name.
- Click **Finish**.



Directory, Name, Top-Level Entity

What is the working directory for this project?

/home/e2/E2_CAS/lab3/task1

What is the name of this project?

task1

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 file.

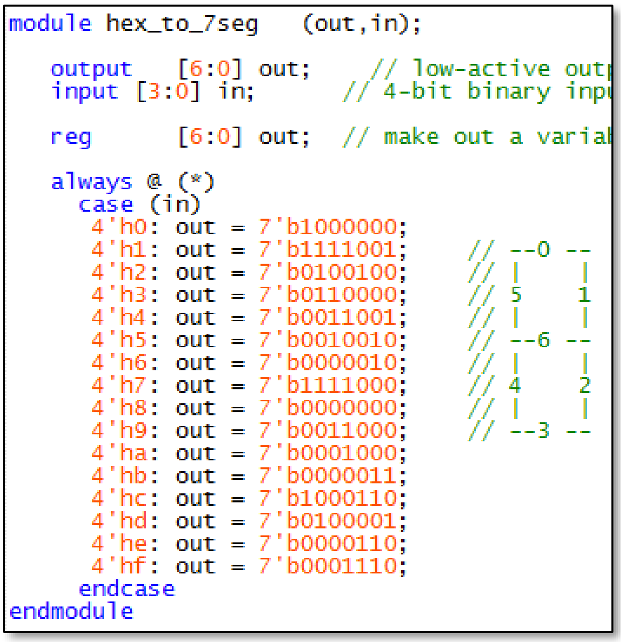
task1_top

Step 6: Device Assignment

Click **Assignments -> Device**, and select the Max 10 chip used in DE10, which is: **10M50DAF484C7G**.

(Interpretation of the device code: 10M is Max10 family. 50 is the size of the device of around 50,000 logic elements). 484 is the number of pins. C7 is the speed grade.)

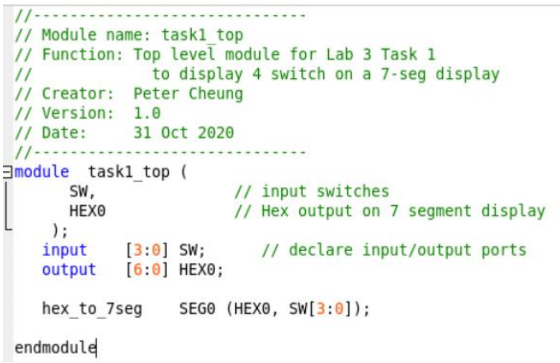
Step 7: Creating the Verilog specification



```
module hex_to_7seg   (out,in);

   output   [6:0] out;    // low-active outp
   input [3:0] in;       // 4-bit binary inpu

   reg      [6:0] out;   // make out a varial

   always @ (*)
     case (in)
      4'h0: out = 7'b1000000;
      4'h1: out = 7'b1111001;     // --0 --
      4'h2: out = 7'b0100100;     // |     |
      4'h3: out = 7'b0110000;     // 5     1
      4'h4: out = 7'b0011001;     // |     |
      4'h5: out = 7'b0010010;     // --6 --
      4'h6: out = 7'b0000010;     // |     |
      4'h7: out = 7'b1111000;     // 4     2
      4'h8: out = 7'b0000000;     // |     |
      4'h9: out = 7'b0011000;     // --3 --
      4'ha: out = 7'b0001000;
      4'hb: out = 7'b0000011;
      4'hc: out = 7'b1000110;
      4'hd: out = 7'b0100001;
      4'he: out = 7'b0000110;
      4'hf: out = 7'b0001110;
     endcase
endmodule
```

- In Quartus, create a design file for the decoder module in Verilog HDL as **hex_to_7seg.v** using: **File > New ….** and select Verilog HDL from the list.
- Type the Verilog source file as shown here. Make sure that you pay attention to the syntax of Verilog. Save your file.
- At this stage, you check the syntax of your code by clicking: **Process > Analyze current file**. You should get into a habit of ALWAYS perform this step to make sure that the new Verilog module you have created is error free. It will save you a lot of time later.
- Click: **Project > Add Current file to Project** to include this module in your design.

Step 8: Create Top-Level Specification in Verilog



```
//-----------------------------
// Module name: task1_top
// Function: Top level module for Lab 3 Task 1
//              to display 4 switch on a 7-seg display
// Creator:  Peter Cheung
// Version:  1.0
// Date:     31 Oct 2020
//-----------------------------
module  task1_top (
      SW,                  // input switches
      HEX0                 // Hex output on 7 segment display
   );
   input    [3:0] SW;      // declare input/output ports
   output   [6:0] HEX0;

   hex_to_7seg    SEG0 (HEX0, SW[3:0]);

endmodule
```

- We need to create a top-level (at chip level) design that make sure of the decoder module. Create the file "**task1_top.v**" as shown here. Specify that this is our top-level design with:
  Project > Set as Top-level Entity ….
- Verify that everything works properly with: **Process > Start > Start analysis & elaboration**. Make sure that there is **no error**. (Warnings often capture potential errors. However, the Quartus system generates many warnings, and nearly all of which are not important. Once you have gain confidence in the system, you may start ignoring the warning, but never ignore any error.)

> You will save a lot of time if you ALWAYS use these two steps: analyze, and analysis & elaboration, and ensure that ALL errors are dealt with (and warning understood).

Note: Every time you create a new entity or module as part of your design, you must include the file in the project.

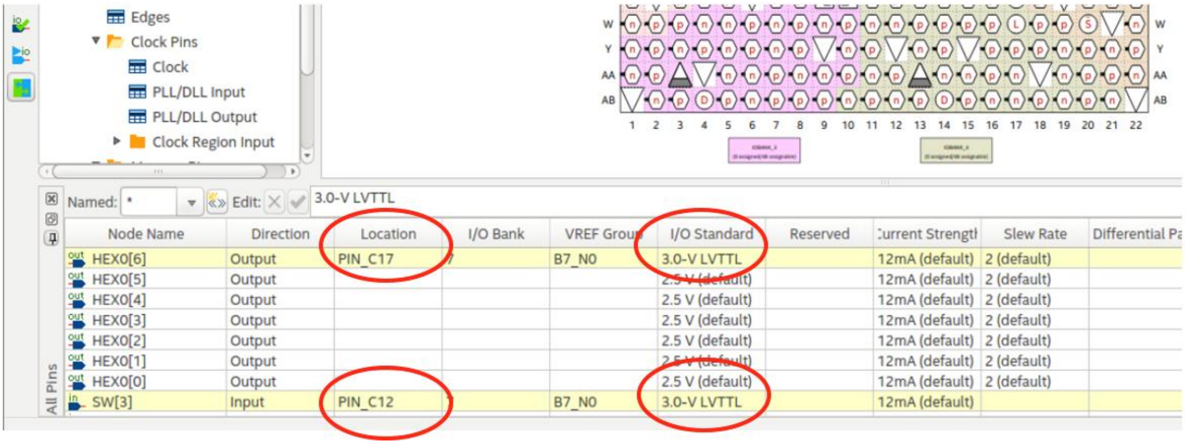
- Click: **Project > Add Current Files to Project ….**,

Signal Name	Pin Location
HEX0[6]	PIN_C17
HEX0[5]	PIN_D17
HEX0[4]	PIN_E16
HEX0[3]	PIN_C16
HEX0[2]	PIN_C15
HEX0[1]	PIN_E15
HEX0[0]	PIN_C14
SW[3]	PIN_C12
SW[2]	PIN_D12
SW[1]	PIN_C11
SW[0]	PIN_C10

Step 9: Pin assignment – the hard way

You need to associate your design with the **physical pins** of the Max 10 FPGA on the DE10 board. We will now assign two of 11 pins used in our design.

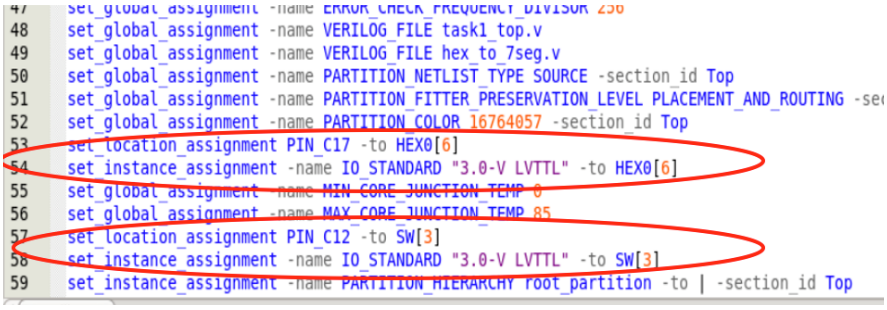
- Click **Assignment > Pin Planner** and a new window with the chip package diagram. You should also see the top-level input/output ports shown as a list.



Named: * Edit: 3.0-V LVTTL

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pa
HEX0[6]	Output	PIN_C17	7	B7_N0	3.0-V LVTTL		12mA (default)	2 (default)	
HEX0[5]	Output				2.5 V (default)		12mA (default)	2 (default)	
HEX0[4]	Output				2.5 V (default)		12mA (default)	2 (default)	
HEX0[3]	Output				2.5 V (default)		12mA (default)	2 (default)	
HEX0[2]	Output				2.5 V (default)		12mA (default)	2 (default)	
HEX0[1]	Output				2.5 V (default)		12mA (default)	2 (default)	
HEX0[0]	Output				2.5 V (default)		12mA (default)	2 (default)	
SW[3]	Input	PIN_C12	7	B7_N0	3.0-V LVTTL		12mA (default)		

- Click on the field shown and select the appropriate values for Location and I/O standard.
- Close the pin assignment window and click: **File > open…** Enter *.* in the file name field and select: **task1.qsf** (qsf = **Q**uartus **S**etting **F**ile). Examine its contents. You should see the effect of the manual pin assignment step as highlighted in RED.
- The first line defines the physical pin location of HEX0[6] is PIN_C17.
- The second line defines the voltage standard used is 3.0V LVTTL.



```
set_global_assignment -name ERROR_CHECK_FREQUENCY_DIVISOR 256
set_global_assignment -name VERILOG_FILE task1_top.v
set_global_assignment -name VERILOG_FILE hex_to_7seg.v
set_global_assignment -name PARTITION_NETLIST_TYPE SOURCE -section_id Top
set_global_assignment -name PARTITION_FITTER_PRESERVATION_LEVEL PLACEMENT_AND_ROUTING -se
set_global_assignment -name PARTITION_COLOR 16764057 -section_id Top
set_location_assignment PIN_C17 -to HEX0[6]
set_instance_assignment -name IO_STANDARD "3.0-V LVTTL" -to HEX0[6]
set_global_assignment -name MIN_CORE_JUNCTION_TEMP 0
set_global_assignment -name MAX_CORE_JUNCTION_TEMP 85
set_location_assignment PIN_C12 -to SW[3]
set_instance_assignment -name IO_STANDARD "3.0-V LVTTL" -to SW[3]
set_instance_assignment -name PARTITION_HIERARCHY root_partition -to | -section_id Top
```

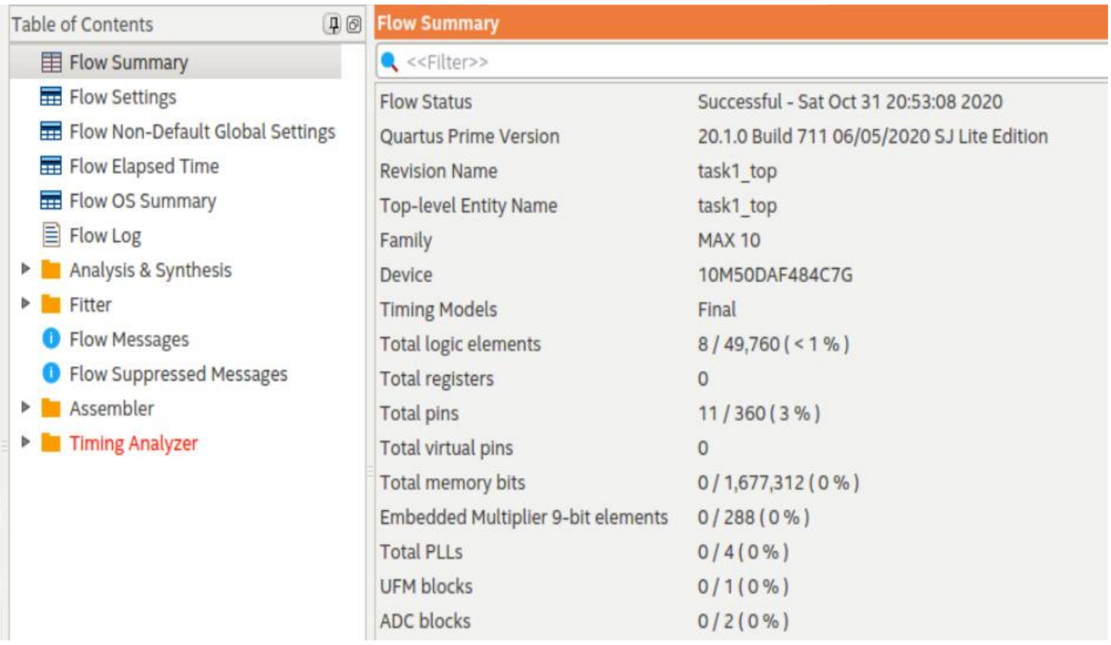
Step 10: Pin Assignment – the easy way

- Manual pin assignment is tedious and prone to errors. A much better way to perform pin assignment is to insert a text file with the necessary information directly into the **.qsf** file.
- Delete the four lines highlight above which was created through the manual pin assignment in Step 9.
- Download from the course webpage: **pin_assignment.txt** to the task1 folder.
- Click: Edit > Insert File … and insert pin_assignment.txt at the end of the file.

ALL the pins used on the DE10 are assigned here. However, unused pins are ignored.

Step 11: Compile the design & Programming the FPGA

- Click Process > Start Compilation. This will perform all the steps of compilation, placement, routing, fitting etc. and produce a bit-stream file (.sof) ready to blast onto the FPGA.
- Examine the Compilation Report and you should see a Flow Summary similar to the one shown here.



Flow Summary

Flow Status	Successful - Sat Oct 31 20:53:08 2020
Quartus Prime Version	20.1.0 Build 711 06/05/2020 SJ Lite Edition
Revision Name	task1_top
Top-level Entity Name	task1_top
Family	MAX 10
Device	10M50DAF484C7G
Timing Models	Final
Total logic elements	8 / 49,760 (< 1 %)
Total registers	0
Total pins	11 / 360 (3 %)
Total virtual pins	0
Total memory bits	0 / 1,677,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 288 (0 %)
Total PLLs	0 / 4 (0 %)
UFM blocks	0 / 1 (0 %)
ADC blocks	0 / 2 (0 %)

- This correctly shows that the design used 8 logic elements (out of nearly 50,000) and 11 pins.
- Programme the DE10 with YOUR design with the file: task1_top.sof. (See Task 0 if you have forgotten how to do this.) Test you design.

Congratulations! You have now completed your design from beginning to the end.

Put verified modules in mylib

For the rest of this module, you will design and verify various Verilog modules which you will reuse. You should copy **hex_to_7seg.v** (and others in the future) to the "**mylib**" folder and include them in your new design as necessary.

Note: When you perform a compilation, there may be a popup window informing you that some "Chain_x.cdf" file has been modified, and ask if you wish to save it. Just click NO.

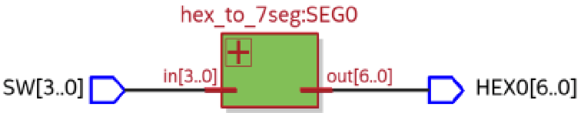
Task 2: Explore Netlist Viewer and Timing Analyzer

Step 1: Viewing the design

Quartus Prime provides a graphical view of the synthesized design. Exploring this provides you with some insight into how the Verilog HDL code is turned into actual FPGA hardware.

- Click **Tools > Netlist Views > RTL Viewer**

This should appear on your screen:



- Push down into this block and investigate what is being displayed and how it relates to the decode logic.

RTL Viewer only shows the abstract Boolean description of the design, not the physical implementation on the FPGA.

- Click **Tools > Netlist Views > Technology Map Viewer (post mapping)**

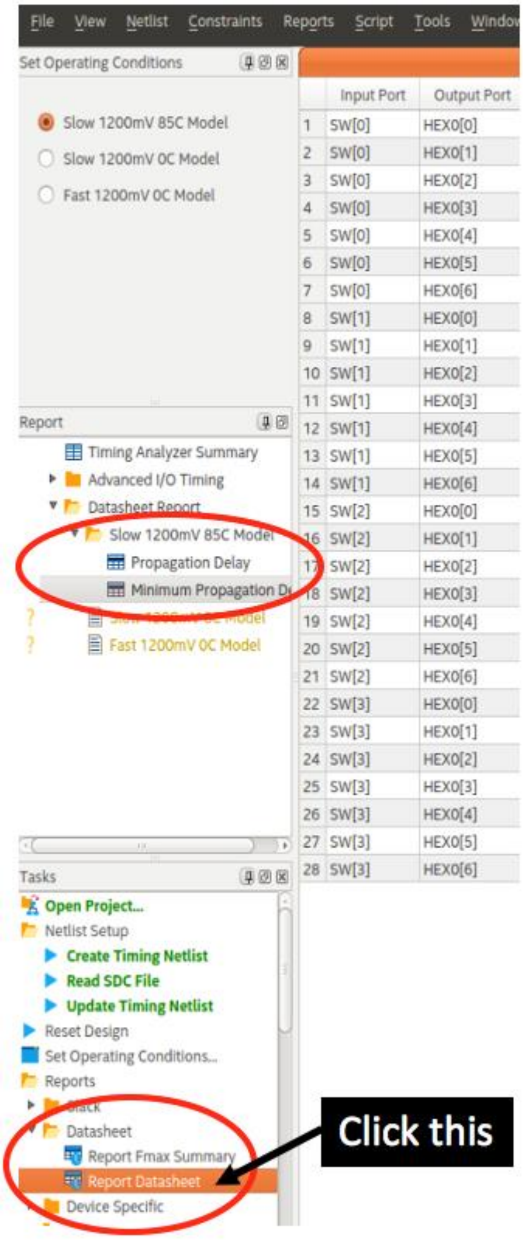
Explain what you find and link this back to the Compilation Report.

Step 2: Timing Analyzer

Click **Tools > Timing Analyzer**

A Timing Analyzer window will appear.
Now click Report datasheet. The Timing Analyzer tool will provide datasheet type table showing the propagation delays of all the paths in your design

Study the results for worst-case delay at different temperature and explain what you found.



● Slow 1200mV 85C Model
○ Slow 1200mV 0C Model
○ Fast 1200mV 0C Model

	Input Port	Output Port
1	SW[0]	HEX0[0]
2	SW[0]	HEX0[1]
3	SW[0]	HEX0[2]
4	SW[0]	HEX0[3]
5	SW[0]	HEX0[4]
6	SW[0]	HEX0[5]
7	SW[0]	HEX0[6]
8	SW[1]	HEX0[0]
9	SW[1]	HEX0[1]
10	SW[1]	HEX0[2]
11	SW[1]	HEX0[3]
12	SW[1]	HEX0[4]
13	SW[1]	HEX0[5]
14	SW[1]	HEX0[6]
15	SW[2]	HEX0[0]
16	SW[2]	HEX0[1]
17	SW[2]	HEX0[2]
18	SW[2]	HEX0[3]
19	SW[2]	HEX0[4]
20	SW[2]	HEX0[5]
21	SW[2]	HEX0[6]
22	SW[3]	HEX0[0]
23	SW[3]	HEX0[1]
24	SW[3]	HEX0[2]
25	SW[3]	HEX0[3]
26	SW[3]	HEX0[4]
27	SW[3]	HEX0[5]
28	SW[3]	HEX0[6]

Step 3: Test yourself

Create your own design in task2 folder (top-level file is **task2_top.v**) to display all 10-bit sliding switches as hexadecimal on three of the 7-segment LED displays.

Appendix - Installing USB-Blaster Device Drive under Windows 10

If you are using Quartus under the Virtual Machine, the driver should already been installed, and you should not need to do anything.

If you are installing Quartus directly to your PC running Windows 10, you may need to install the driver manually according to the following steps:

1. Plug the USB-Blaster into your PC.
2. Open the **Device and Printers** (Control Panel **|** Devices and Printers).
3. Under Unspecified, USB Blaster should be listed. Right mouse click on this and then select Properties.
4. Select the **Hardware** tab and select **Properties**.
5. A new window should pop up with the General tab already selected. Select **Change Setting**s.
6. Again a new window should pop up with the General tab already selected. Select **Update**.
7. Select **Browse my computer** f**or driver software**.
8. Find **intelFPGA_lite\20.1\quartus\drivers\**

- (Note 1: Your altera file is located at the location you selected when you first installed quartus. The location listed in this document is the default location)
- (Note 2: Stop at the drivers folder, i.e., do NOT go deeper by opening a folder within the drivers folder)

1. Select **OK**. Make sure the proper path was selected then select **Next**.
2. If the Windows security window pops up, check the Always trust software from “Altera Corporation” box and select **Install**.