User Manual for Xen10

Pranav Pawar, 213079020
Rahul Verma, 213079030

August 17, 2023

Contents

1 Board Synopsis

Xen10 is a 10M25SAE144C8G / 10M08SAE144C8G MAX-10 FPGA-based board developed by Wadhwani Electronics Lab, IIT Bombay, NPTEL, Intel. I/O supply voltage required is 3.3 V. Power supply either through USB or EXT_PWR header must not exceed 5 V (refer figure 1). This development board is equipped with on chip 12-bit ADC, TSD for temperature sensing, PLL, on-board 12-bit DAC and peripherals consisting of 8-switches, 8-leds, 4-push buttons, 10,50 MHz and 1 Hz clock generators, header connectors for GPIO's, LCD, OLED, I2C, etc. This is a low-cost solution aimed to cater to the needs of undergraduate and graduate electrical/electronics engineering students in a course in digital design and thereafter, design of fairly complex digital systems.

Note:

1. Xen10 board is equipped either with 10M25SAE144C8G or 10M08SAE144C8G. This manual is based on 10M25SAE144C8G. All the procedures are same for 10M08SAE144C8G unless mentioned.
2. Web-address of web-links provided in this document are subject to change

2 Features and specifications for Xen10

Figure 1 shows a top view of Xen10 with detailed labels. Functionality of labelled devices will be discussed in appropriate sections.

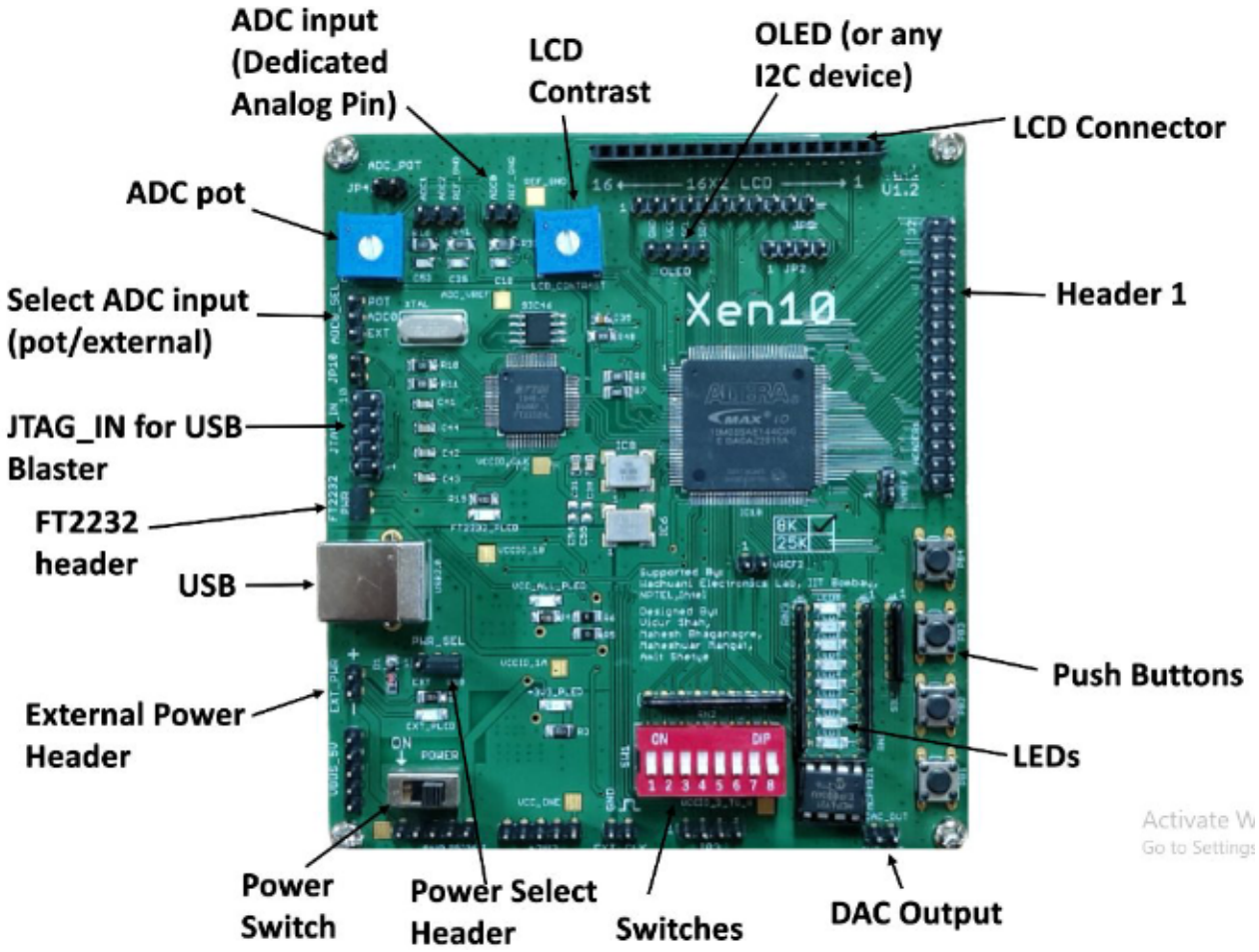

Figure 1: Xen10 topview with labelled features

Following are the features on Xen10.

1. A board centered around 10M25SAE144C8G / 10M08SAE144C8G, a FPGA of Altera's MAX 10 family.

2. Programmable by USB or USB blaster JTAG and powered with the same, with provision for external DC supply.
3. Preconfigured on-board I/O- 8 switches, 8 LEDs and 4 push-buttons with hardware debounce
4. On-board clock generator of 1Hz, 10MHz and 50MHz frequencies, and provision for external clock source connection through EXT_CLK header.
5. Connectors provided on-board to interface with standard peripherals directly like LCD with contrast control through on-board potentiometer, OLED or any other device with I2c interface.
6. On-board 12-bit single channel MCP4921 DAC with SPI interface
7. Large number of on-board I/Os (32) provided for various applications
8. Specifications specific to the FPGA are tabulated below

Specifications	10M25SAE144C8G	10M08SAE144C8G
I/O pins supply voltage	3.3v	3.3v
Logic elements	24960	8064
GPIOs	101	101
Memory bits	691200	387072
Maximum user flash memory bits	6291456	2555904
On-chip 12-bit ADC	1	1
PLL	1	1
TSD (Temperature Sensing Diode)	1	1

3 Precautions for using Xen10

1. Do not touch the pins of any onboard IC directly, to avert the risk of damage by electrostatic discharge.
2. When interfacing with large number of external peripherals, power the board through a single +5V DC supply either through USB or EXT_PWR header (refer figure 1) and ensure enough current can be provided by supply.
3. The voltage at any I/O pin should not exceed 3.3V.
4. If programming board using USB Blaster through JTAG_IN header (refer figure 1) then ensure the FT2232_PWR jumper is disconnected.

4 Pin Mapping of On-Board Peripherals

On-board peripherals are connected to one of the FPGA pin as given in below tabular form.

Switch	FPGA Pin no.	LED	FPGA Pin no.
SW 8	47	LED 8	60
SW 7	46	LED 7	59
SW 6	45	LED 6	58
SW 5	44	LED 5	57
SW 4	43	LED 4	56
SW 3	41	LED 3	54
SW 2	39	LED 2	52
SW 1	38	LED 1	50

Table 1: Pin-mapping for on-board Switches and LED's

Push Button	FPGA Pin no.
PB 1	66
PB 2	70
PB 3	69
PB 4	74

Table 2: Pin-mapping for on-board Push Buttons

Clock Source Frequency	FPGA Pin no.
1 Hz CLK	55
50 MHz CLK	26
Ext CLK	27
10 MHz CLK	29

Table 3: Pin-mapping for on-board Clock Sources

ADC	
Analog input	FPGA Pin no.
ADC1IN1	6
ADC1IN2	7
ANAIN1	3

DAC	
DAC pinout	FPGA Pin no.
DAC_CS	65
DAC_SDI	62
DAC_SCLK	64

Table 4: Pin-mapping for ADC and DAC

OLED (or any I2C device)	
I2C Pin	FPGA Pin no.
SCL	126
SDA	136

Table 5: Pin-mapping for OLED

Header-1				LCD	
Header pin no.	FPGA pin no.	Header pin no.	FPGA pin no.	LCD pin no.	FPGA pin no.
1	75	2	76	RS	122
3	77	4	78	RW	124
5	79	6	81	EN	127
7	84	8	85	D0	130
9	86	10	87	D1	131
11	88	12	89	D2	132
13	90	14	91	D3	134
15	92	16	93	D4	135
17	96	18	98	D5	121
19	99	20	100	D6	140
21	101	22	102	D7	141
23	105	24	106		
25	110	26	111		
27	113	28	114		
29	118	30	119		
31	GND	32	3.3V		

Table 6: Pin-mapping for Header-1 and LCD

JTAG_IN pin no.	Pin function	FPGA pin no.
1	TCK	18
2	GND	-
3	TDO	20
4	VCC	-
5	TMS	16
6	NC	-
7	NC	-
8	NC	-
9	TDI	19
10	GND	-

Table 7: Pin mapping of JTAG_IN header

Jumper-3 (JP3)		Jumper-2 (JP2)		Reference-2/3 (REF2/REF3)	
JP3-1	28	JP2-1	17	REF2-1	48
JP3-2	32	JP2-2	123	REF2-2	61
JP3-3	33	JP2-3	112	REF3-1	97
JP3-4	30			REF3-2	80

Table 8: Pin mapping of JP2, JP3, REF2 and REF3 header

5 Configuring Xen10 Before Programming

You will need to ensure that an appropriate power source and interface for programming is selected for the FPGA while programming or running an application. To do this, an on-board jumper PWR_SEL and FT2232_PWR respectively needs to be connected suitably. Below table 9 and 10 describes the connections and their functionality

PWR_SEL Jumper Setting	Power Source for Xen10
Between pin 2 and pin 3	Board powered through USB (Bus-powered via host PC)
Between pin 1 and pin 2	Board powered through external power supply header EXT_PWR

Table 9: Configuration for PWR_SEL jumper

FT2232_PWR Jumper Setting	Programming Interface for Xen10
Jumper connected	USB interface is selected to program board
Jumper disconnected	Programming through JTAG_IN header is selected (suitable USB Blaster can be used)

Table 10: Configuration for FT2232_PWR jumper

Note: If Programming through JTAG_IN header is desired then along with the configuration of FT2232_PWR Jumper as shown in table 10, JP10 jumper should also be shorted.

6 Getting Xen10 working with your PC (via USB interface)

1. Plug in the USB cable to the USB connector on the Xen10 board, and the other end of the cable to a USB port on your PC. Turn on the power switch. There are three power LEDs on the board which are FT2232_PLED, VCC_ALL_PLED and +3V3_PLED, which should glow.

2. You will need to install the drivers for Xen10 the first time it is connected to your PC. The procedure to do so varies slightly, depending on the operating system in use.

For Windows 7 and above:

3. Once you connect Xen10 to your PC and power it on, Windows 7 and higher versions automatically search for available drivers. If no drivers are available, a desktop notification showing Device driver was NOT successfully installed will pop up. In such case, follow the below steps.

4. Download the CMD_MAX10.zip file and extract it to the desired directory. Zip file is available on resource drive

5. Open the device manager by right clicking on This PC / Computer - Properties - Device Manager.

6. Inside Device Manager, under other devices you will see 2 uninstalled devices named Dual RS232-HS with yellow warning (Ensure the Xen10 board is connected to the host as given in step 1). Right click on the first one and select Update driver. Now select Browse the computer for driver software, and click Next. Now, provide the path to the folder CMD_MAX10 (which you extracted in step 4). Click Next.

7. The installation should now begin. After this is complete a dialogue box will appear displaying successfully installed USB serial converter A, repeat step 6 for the second uninstalled device Dual RS232-HS. Similar dialogue box displaying successfully installed USB serial converter B will appear. Xen10 is now ready to be used.

.
For Linux:
All FTDI device drivers are supported in Ubuntu 11.10 and above versions. Just follow the step-1.
For MAC OS:
Visit the FTDI drivers official website FTDI drivers and download the latest version of driver shown on webpage which is compatible to your OS version. For more information you can refer to the installation guidelines provided by FTDI on the webpage , Select the appropriate OS version to view installation guideline document.

7 Testing on-board peripherals of Xen10

It is a good practice to test Xen10 board before using it. Below are the steps to test on-board LEDs, Switches, Push Buttons and Bidirectional Pin Header:

1. Ensure Xen10 board is powered as shown in section 6 step 1

2. Connect LCD on LCD header named as 16x2 LCD. Ensure pin-1 of LCD is connected to the corresponding pin-1 of the 16x2 LCD header.

Even if LCD is not desired to be tested, it is recommended to connect LCD, as run-time instructions will be displayed on LCD. In case if LCD is not available then one can refer instructions in this manual carefully.

3. Connect header pins of Header-1 using jumper wires. Do not connect pin-31 to pin-32. Pin-1 should be connected to adjacent pin i.e., pin-2, similarly pin-3 to pin-4, pin-5 to pin-6 and so on until pin-29 to pin-30. Pin-31 and pin-32 corresponds to the VCC and GND and thus should not be connected.

If Header-1 is not desired to be tested, then you can skip step-3. However, LCD will display that Header-1 is not working since connections are not made, ignore this message.

4. Program the Xen10 board with test_MAX10_10M25SAE144C8G.svf file or test_MAX10_10M08SAE144C8G.svf file (Depending on which device you are using) using UrJTAG as shown in section-9. UrJTAG software and svf file are available on Xen10 resource drive

5. After successful file upload, first ensure all switches are off. Switch-8 (SW8) is reserved as reset, make SW8 on and within one second make it off. This will reset the system. On successful reset LCD will display "Welcome: Xen10 Design by WEL" this also implies SW8 is working properly. Adjust LCD_CONTRAST pot if message is not clearly visible.

6. After few seconds "Push PB1 for starting" will be displayed on LCD. Press and hold PB1 at least for a second to start Xen10 board testing.

7. Clock Frequency Testing: Testing procedure starts from on-board clock generators. 50 MHz, 10 MHz and 1 Hz clock generators are tested sequentially. Clock pulses are counted and displayed on LED's in binary encoded format, once a threshold count is reached, LCD will display "Done" message. In about 5 seconds all clock sources will be tested.

8. Switches and LEDs Testing: After testing of clock generators "Test SW and LED" will appear on LCD. Turn on SW1. Notice one led will blink at a time and shift through all the LEDs. Similarly when SW2 is turned on keeping the SW1 on, two LEDs will blink and shift and so on until SW7 is turned on where seven LEDs will blink and shift. Remember not to turn on SW8 as this is reserved for reset. Once all switches till SW7 are on, LCD will display done. User has to manually check the correctness of switches and LEDs by observing the desired LED pattern output. If desired output occurs then corresponding switches and LEDs are working. When Switch and LED test is done then LCD will display "Switch Off ALL Switches". So we have switch off all switches from SW1 to SW7.

9. Push Button Testing: After completing switch and LED test, LCD will display "Test: Push Button". Press each push button at least once to pass the test. If any of the LEDs are glowing then push button is working properly.

10. Bidirectional Pin Header Testing: Switch on SW4 and press PB2 after the LCD displays "SW on SW4, Press PB2: test Header". After pressing PB2,if LCD displays all zeros in second row then all pin headers are working properly. But if any index display 1 in the second row then we can identify either (2*index - 1) or (2*index) pin header is faulty and index number is displayed on LCD first row. After that LCD will display "All tests Done." which means all test are completed.

8 Using Quartus II v18.1 IDE to make Applications

Quartus II is the IDE provided by Altera for the user to write their HDL code (VHDL or Verilog) to target a digital system designed to be implemented on a target PLD (a CPLD or and FPGA). The IDE allows for code compilation, followed by generating a programming file for a specific target PLD. Quartus II has a free subscription/web edition that can be used. Suppose we wish to design and implement on Xen10 the simple logic circuit shown in Figure 2. This section provides a step-by-step method for doing the same.
Note: Below steps are specifically for Quartus II 18.1 version, In other versions, although the steps are quite similar there may be some differences.

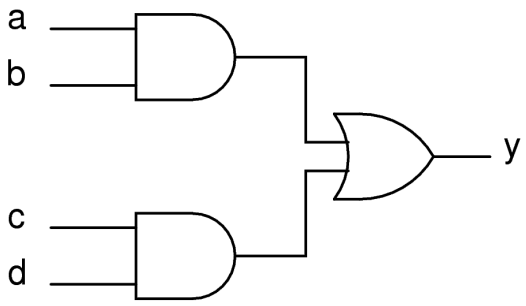

Figure 2: Example circuit to be implemented on Xen10

1. Open Quartus II IDE. A pop-up will appear as shown in figure 3 asking you to either create a new project or open an existing one. Click on New Project Wizard. Alternatively, you may click on File → New Project Wizard.

2. An 'Introduction' page opens up. Click on Next.

3. This opens up 'Page 1'. Here, you need to specify a working directory for your project. Refer figure 4. Click Browse (...) to create a new folder for this project. Next specify the project name and top-level design entity. Important: This is a very critical step in your design. Top level design entity refers to the name of the entity in your HDL code (if you are using VHDL) or name of the module (if you are using Verilog) which you wish to implement. By default, the entity name follows the same name as the project. It is preferred not to change this. However, project name can be other than top-level entity name. Since, we wish to implement the circuit in Figure 2, name the project as myLogicCkt. The top-level entity is also named myLogicCkt by default. Click Next.

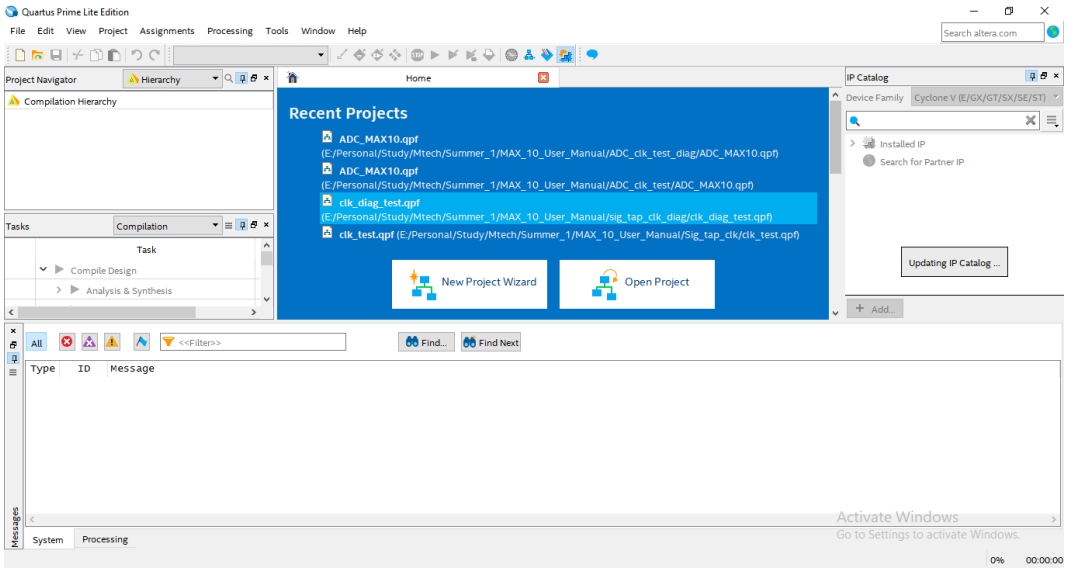

Figure 3: Quartus home page

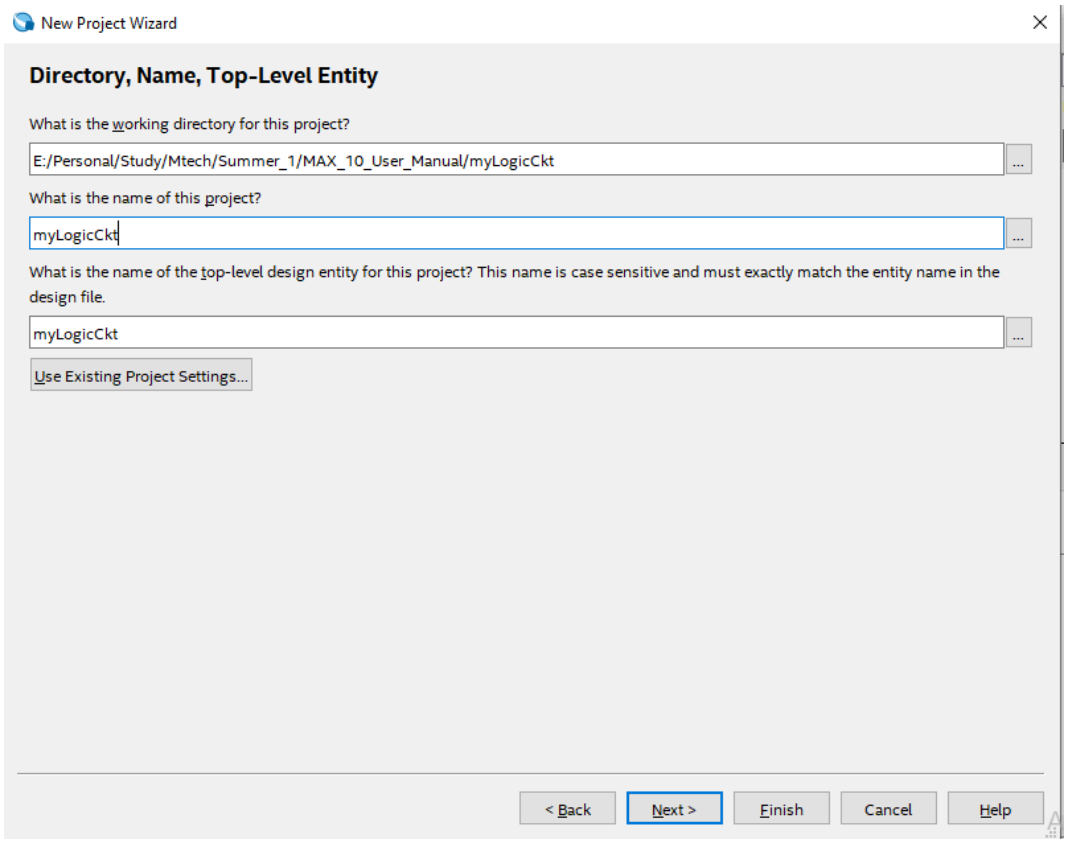

Figure 4: Quartus Project Directory info

Note: By default path for the working directory for new project will be "C:\intelFPGA_lite\18.1". Don't create project in this default directory. Create a new project directory as explained above. Creating project in default directory may cause Quartus software to stop responding.

4. This opens up 'Page 2'. This page allows you to select project type. You can create a project from an existing design template but this is not needed in this example. Select Empty project radio button. Click next.

5. This opens up 'Page 3'. This page allows you to include any existing VHDL/Verilog program files as part of your project. You may skip this step as this is not mandatory. Click Next.

6. On 'Page 4', you are asked for the family and device settings, i.e. the target PLD on which

you wish to implement your design. Important: This is a very crucial step to select the correct device, which is on Xen10 as shown in figure 5. Click on the Family drop-down list and select MAX 10. The Available devices section shows a long list of devices available in the MAX 10 family. You can filter out the device list by typing the device name in Name filter, device name is 10M25SAE144C8G. Select the device 10M25SAE144C8G (or 10M08SAE144C8G if you are using this device) from the Available device section and click Next.

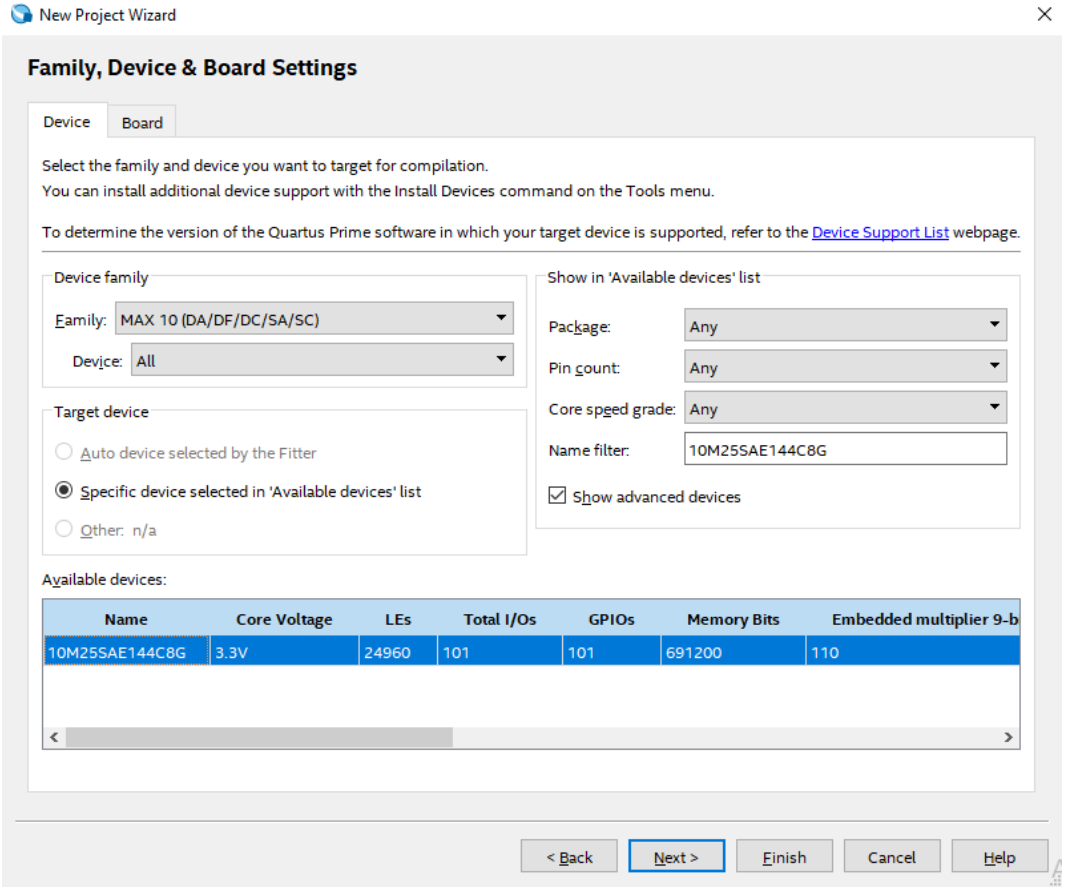

Figure 5: Quartus device page

7. In the 'Page 5' window, you will be asked to select a simulation tool.From drop down menu in Tool Name column, corresponding to the Tool Type simulation select Modelsim-Altera, refer figure 6. From formats drop down menu select the language, in our case VHDL. Click Next

8. This opens up 'Page 6' you will be shown a project summary. You may use this to review your settings, and can go back to rectify any mistake. Once confirmed, click Finish. The project is now created.

9. Now, go to File → New. A window will open, asking you to select the HDL you will use. In this case, select VHDL File and click on OK. A text editor window will open, where you can write your HDL code. For our circuit, copy and paste the following code in the editor.

```
library ieee;
use ieee.std_logic_1164.all;
entity myLogicCkt is port (a, b, c, d: in std_logic; y : out std_logic);
end entity myLogicCkt;
architecture struct of myLogicCkt is
begin
y <= (a and b) or (c and d);
end architecture struct;
```

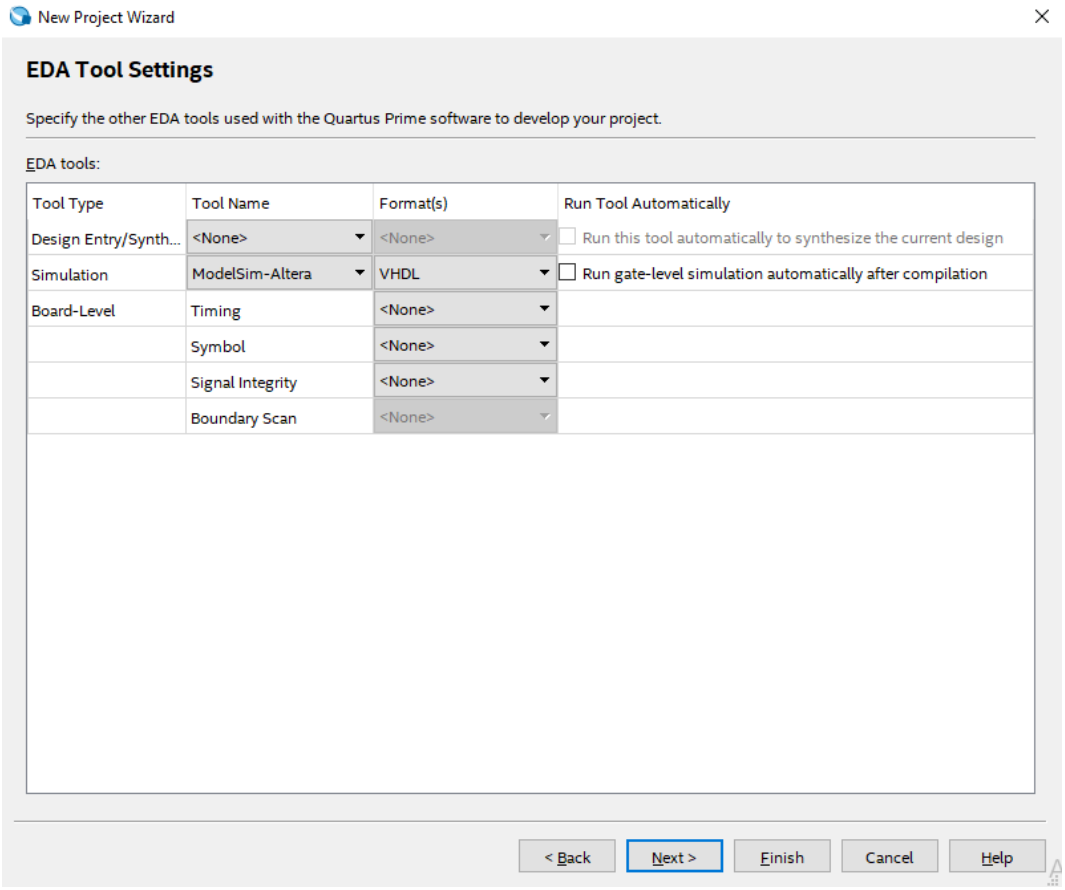

Figure 6: Quartus EDA tool settings page

Notice that the entity name myLogicCkt is kept the same as we had specified while starting the new project wizard.

Note: While copy pasting there might be a change in text formatting, ensure once that proper code is copied.Generally underscores are converted to blank spaces.

10. Save this file (preferably in the same project directory) with any name e.g. circuit1.vhd. The file name should have a .vhd extension. By default file name will be same as entity name.

11. Go to Processing → Start Compilation. This starts the compilation process, and errors in the code, if any, are shown on the post-compilation report. Wait till the compilation is completed.

12. In this manual we have skipped simulation part. But, it is a good practice to do simulation for fairly complex designs before flashing program on target device. For more info about simulation using modelsim, refer tutorials here.

    Note: For the first time, while opening simulation window, Quartus will show error saying "Path to modelsim not found...". To solve this issue go to Tools -> Options -> EDA Tool Options and provide path to win32aloem in section Modelsim-Altera. The path will look typically like this "C:\intelFPGA_lite\18.1\modelsim_ase\win32aloem"

13. You need to now assign the port pins of your design (a, b, c, d, y) to I/O pins on the FPGA to verify the logic function working. For simplicity, we will assign the input lines to the on-board switches and outputs to on-board LEDs.

14. Go to Assignments → Pin Planner. A new window opens up, showing you the schematic of the device selected (in this case, MAX 10), and below, the signal lines that need to be pin assigned under All Pins section. Referring to the figure 7, In-front of a Node Name 'a' click in the blank space in column Location. Type 38 and automatically PIN_38 will appear.

Press Enter. Similarly, pin map 'b', 'c', 'd' and 'y' to pin 39, 41, 43 and 50 respectively. Note that pin 50 is associated with on-board LED 1 as 'y' is an output. You can use any other appropriate pins by referring to the information in Table 1 to do the pin assignment. Warning: Sometimes the pin planner window may show the signals TDI, TDO, TMS, TCK. DO NOT assign these to any pins.

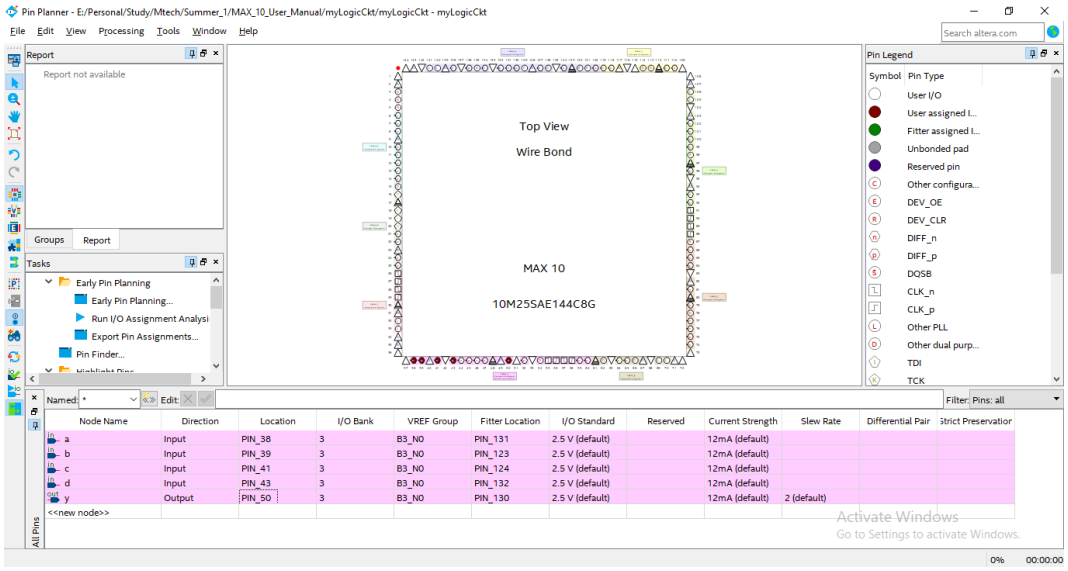

Figure 7: Quartus pin planner tool window

15. Once the pin assignment is complete close the pin planner window, compile the design again (i.e., repeat Step 12).

16. Now go to Tools → Programmer. A programmer window will open. You should see the project output file myLogicCkt.sof in this window as shown in figure 8. If not then click on Add File button present on left panel, your project directory will open up. Navigate to the output_files folder and select .sof file. If this file is not present then you may need to create a fresh project from the start.

17. In the programmer window, go to File → Create (JAM, SVF...). A new window will open. Select the file format as Serial Vector Format (SVF) and click on ok. SVF file with the name same as top-level entity I.e., myLogicCkt.svf is created in working project directory. The programming file is now ready.

9 Using UrJTAG to program Xen10

UrJTAG is a free software to program devices using the JTAG protocol. UrJTAG uses a virtual JTAG port on the PC which is usually accessed through the USB port, as in Xen10. Assume that you have your programming file created from Quartus II, and saved somewhere in directory as myLogicCkt.svf (as done in previous section). Follow the given instructions to program Xen10 with this .svf file.

1. Connect the Xen10 board with the host using USB cable. Turn on the power switch on Xen10 board. 3 Leds will turn on indicating that Xen10 is powered on.

2. Download the UrJTAG folder from here to your PC desktop. Open the jtag.exe executable in the folder. A terminal window with a jtag> prompt opens up.

3. Give the command "cable ft2232" without double inverted commas and hit enter. If the drivers have been installed, you should see a message Connected to libftd2xx driver.

4. Now, give the command "detect" and hit enter. It should now show you the FPGA (target device) details as follows.
IR Length: 10
Chain Length: 1

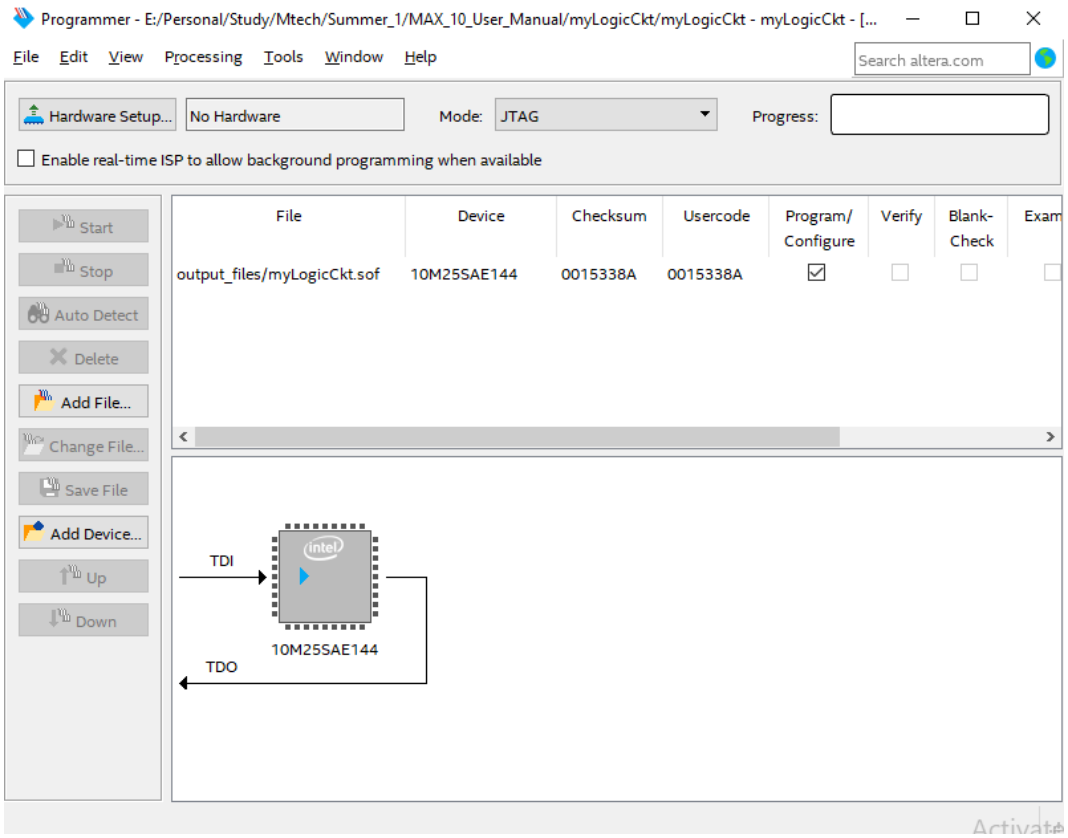

Figure 8: Quartus programmer tool window

Device ID: 0000001.....
Manufacturer: Altera
.
.
.
.
.
.
jtag>

5. The steps 3-4 need to be followed every time when (a) Xen10 is connected to your PC and (b) a new jtag.exe window is opened.

6. The device can now be programmed. Give the command svf "path to the file myLogic.svf" without the double inverted commas for example if path to the file is E:\max10_code\project\ myLogic.svf then the command will be svf E:\max10_code\project\myLogic.svf ,the jtag prompt will disappear for around 10-20 seconds while the device is being programmed. Once the prompt returns, it indicates that programming is complete. You may now verify the working of your logic design on the board (using the switches/LEDs etc.).

   Note:

   (a) In case you get an error "TDO is stuck at 1 ..." then change the USB port of your host PC and repeat from step-3.

   (b) If you get "TDO mismatch..." error then probably you have selected wrong device name while creating project. You can change the settings from Assignments -> Device, select correct device name and click OK, again repeat from step-12 of section-8.

10 Using Xen10 for More Applications

Apart from the on-board switches and LEDs, there are push-buttons and connectors/headers are provided for connecting commonly used peripherals such as character LCD/graphics LCD/PS2

devices and also for general-purpose I/O. This section details more about using these features.

1. Using the push-buttons: There are four on-board push-buttons, which are hardware debounced. When pressed, they are connected to high, which can be read by the FPGA. The pin mapping push-buttons are in table 2.

2. Using the switches: There are eight on-board switches. Looking at Xen10 board from front if switch is turned up then it is connected to 1 i.e., high or else it is connected to 0 i.e., low voltage. The pin mapping for these switches are in table 1.

3. Using the clock sources: Xen10 has on-board 1Hz, 10MHz and 50MHz clock sources which can be used independently. Also, you may connect an external clock source to the EXT_CLK header. Refer table 3 for the pin mapping.

4. Using the header pins: Figure 9 shows the numbering convention for the general purpose I/O headers HEADER1, JTAG_IN and LCD Connector. Tables 6 shows the pin mapping for these using the same numbering convention and pattern as in Figure 9.

5. Using the LCD Connector: Xen10 has a 16-pin on-board LCD connector that can be used to interface a 16 × 2-character LCD module (most commonly, the JHD162A). Table 6 shows the pin mapping for this connector. Important:

   (a) The pin 1 of the LCD module should align with the pin 1 in the LCD connector. Wrong connections may damage the LCD module!

   (b) Adjust the LCD_CONTRAST pot to view characters on LCD properly.

6. Using the OLED (or any I2C device) Connector: Xen10 has a 4-pin on-board I2C (named as OLED) connector that can be used to interface a 4-pin OLED module supporting I2c protocol or any other I2C device. Figure 3 shows the numbering convention. Table 5 shows the pin mapping for OLED.

7. Using the 12-bit ADC: Will be updated soon

8. Using the on-board DAC: Will be updated soon

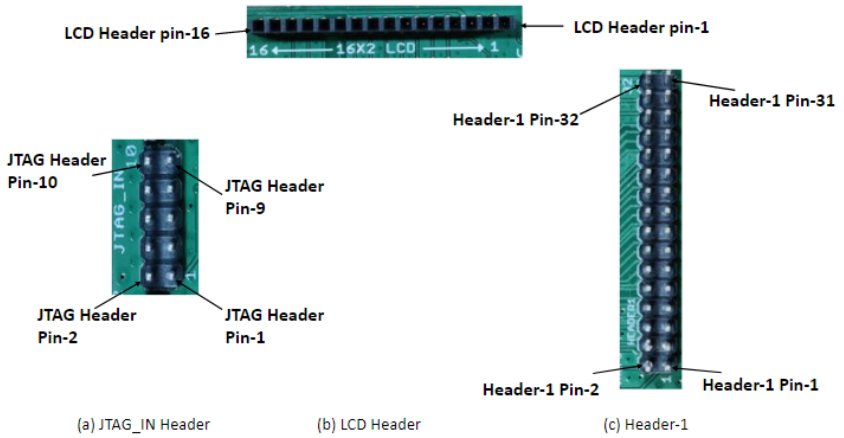

Figure 9: Pin mapping of headers

11 Resources

Resources are available on Xen10 resource drive. List of contents available on website are given below:

1. Xen10 user manual

2. Example applications with manual (labsheets)

3. Driver files for the host PC (CMD_Max10)
4. UrJTAG software for programming the FPGA
5. SVF file for testing Xen10
6. Installer executable for Altera Quartus II v20.1
7. Video tutorials for Xen10 and using Quartus II and UrJTAG
8. Max10 Datasheet and specifications
9. Schematics and layout of Xen10 board
10. Basic concepts of digital logic and digital systems

If you don't find desired content in this manual then please visit here :)

12 Frequently Asked Questions (FAQs)

1. From where can I download Quartus II?
   Ans: You can visit this URL to download a free web edition:

2. Can I use UrJTAG directly from www.urjtag.org?
   Ans: Yes you can, however, the distribution does not have support for MAX 10 devices. Simply use the UrJTAG provided on the resource drive. UrJTAG allows you to create your own libraries for any other PLD you wish to use.

3. I wish to use the USB only for programming, after which I want Xen10 to function as a standalone unit. Is this possible?
   Ans: Yes you can use .pof format file to program Xen10; Max 10 provides both Pof(Programmable Object file) as well as Sof(SRAM Object file). Sof format is used to store the programme in the SRAM and after power off the content is gone thus if programmed with sof file then we can't use Xen10 as standalone (Assuming that once programmed you will be disconnecting USB and putting Xen10 on external supply). However, Pof file is stored in the flash memory and flash memory does maintain its content once power is disconnected.

4. Suppose Max 10 is currently configured to synthesize a particular logic function. Can this setting be erased somehow?
   Ans: There is no concept of erasing programmed content. If sof file is programmed anyways after power off content will be lost whereas in pof file, program is stored in user flash memory which is non-volatile. In this case program cannot be erased but can be over-written.

5. can I simulate my design before implementing it?
   Ans: Yes. Quartus II supports the use of third-party simulators such as ModelSIM. If you wish to simulate, specify the simulator while creating the project.Once you compile your design, you can invoke the simulator by going to Tools → Run Simulation Tool → Gate Level Simulation. Altera's ModelSIM web edition may be downloaded from https://www.altera.com/download/softwar starter Use of a simulation tool is recommended for large and complex designs.

6. What system requirements do I need to get my PC working with Xen10?
   Ans: You need a minimum of Windows XP (Service Pack 2 or higher) with enough disk space to install Quartus II.

7. In the UrJTAG folder, the jtag.exe shows an error while opening/fails to open. What could be the problem?
   Ans: Use the UrJTAG provided on the resource drive. This problem will not occur.