



## Introduction:

The DE10-Lite presents a robust hardware design platform built around the Altera MAX 10 FPGA. The MAX 10 FPGA is well equipped to provide cost effective, single-chip solutions in control plane or data path applications and industry-leading programmable logic for ultimate design flexibility. With MAX 10 FPGA, you can get lower power consumption / cost and higher performance. When you need high-volume applications, including protocol bridging, motor control drive, analog to digital conversion, image processing, and handheld devices, the MAX 10 Lite FPGA is your best choice.

The DE10-Lite development board includes hardware such as on-board USB Blaster, 3-axis accelerometer, video capabilities and much more. By leveraging all of these capabilities, the DE10-Lite is the perfect solution for showcasing, evaluating, and prototyping the true potential of the Altera MAX 10 FPGA.

The DE10-Lite contains all components needed to use the board in conjunction with a computer that runs the Win 7/Win 10 64-bit version or later.

## Package Contents:



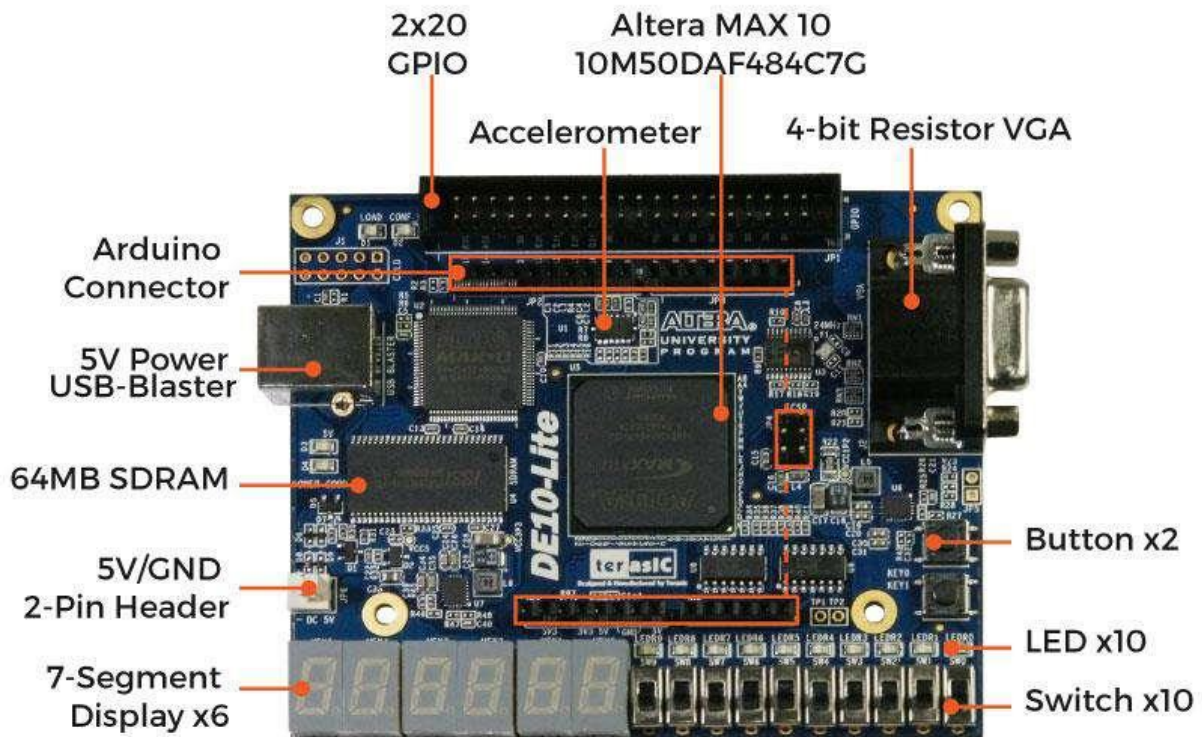
- 1 The DE10-Lite board
- 2 Type A Male to Type B Male USB Cable

The DE10-Lite package includes:

- The DE10-Lite board
- Type A Male to Type B Male USB Cable

## Layout and Components:

A photograph of the board is shown, It depicts the layout of the board and indicates the location of the connectors and key components.



This board has many features that allow users to implement a wide range of designed circuits, from simple circuits to various multimedia projects.

The following hardware are provided on the board:

- MAX 10 10M50DAF484C7G Device
- Integrated dual ADCs, each ADC supports 1 dedicated analog input and 8 dual function pin
- 50K programmable logic elements
- 1,638 Kbits M9K Memory
- 5,888 Kbits user flash memory
- 144  $18 \times 18$  Multiplier
- 4 PLLs

## Programming and Configuration

- On-Board USB Blaster (Normal type B USB connector)

## Memory Device

- 64MB SDRAM, x16 bits data bus

## Connectors

- 2x20 GPIO Header
- Arduino Uno R3 Connector, including six ADC channels.

## Display

- 4-bit resistor-network DAC for VGA (With 15-pin high-density D-sub connector)

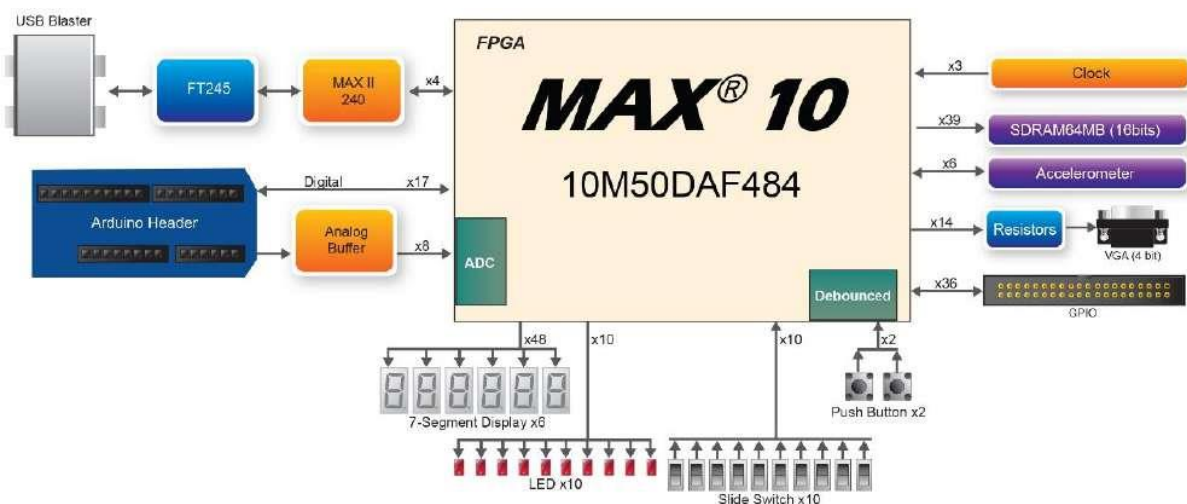
## Switches, Buttons and LEDs

- 10 LEDs
- 10 Slide Switches
- 2 Push Buttons with Debounced.
- Six 7-Segments

## Power

- 5V DC input from USB or external power connector.

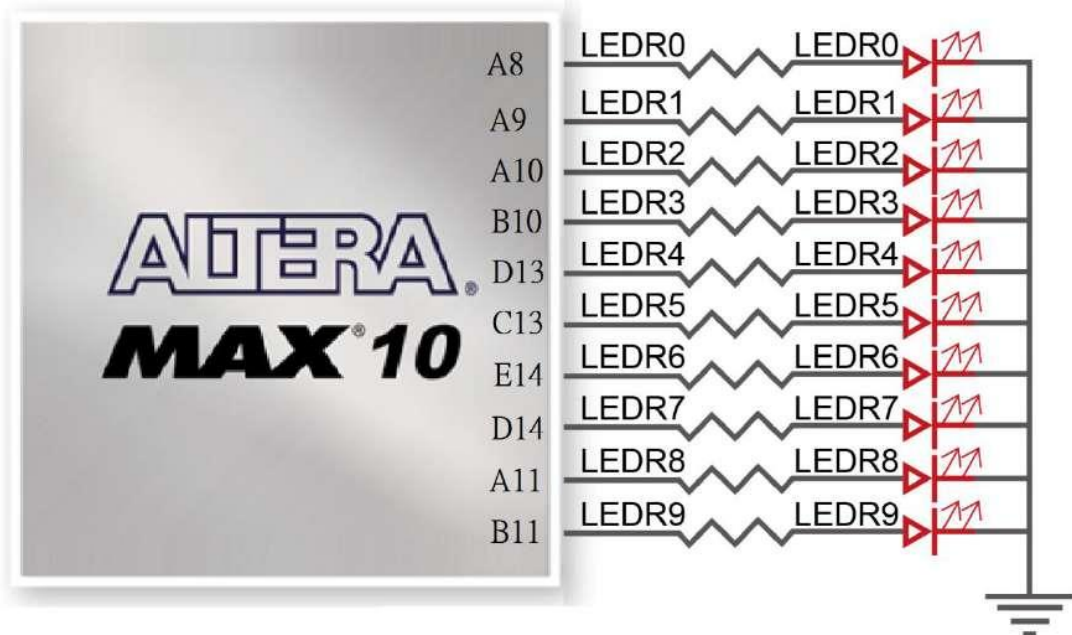
The block diagram of the board. To provide maximum flexibility for the user, all connections are made through the MAX 10 FPGA device. Thus, the user can configure the FPGA to implement any system design.



## I/O pin details of LED, Seven segment and Switch

### User-Defined LEDs

There are also ten user-controllable LEDs connected to FPGA on the board. Each LED is driven directly and individually by a pin on the MAX 10 FPGA; driving its associated pin to a high logic level turns the LED on, and driving the pin low turns it off. The below figure shows the connections between LEDs and MAX 10 FPGA.

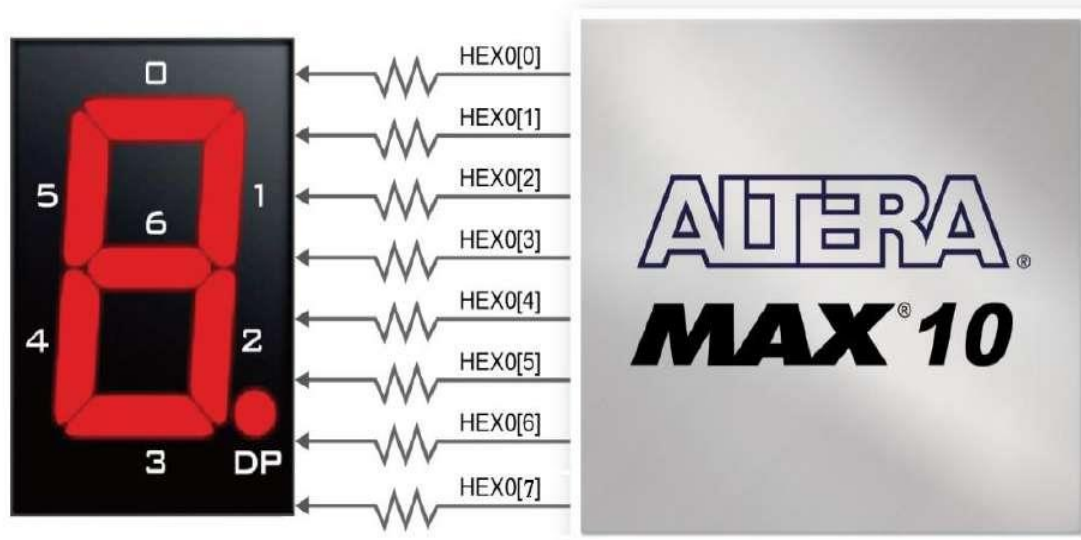


Signal Name	FPGA Pin No.	Description	I/O Standard
LEDR0	PIN_A8	LED [0]	3.3-V LVTTTL
LEDR1	PIN_A9	LED [1]	3.3-V LVTTTL
LEDR2	PIN_A10	LED [2]	3.3-V LVTTTL
LEDR3	PIN_B10	LED [3]	3.3-V LVTTTL
LEDR4	PIN_D13	LED [4]	3.3-V LVTTTL
LEDR5	PIN_C13	LED [5]	3.3-V LVTTTL
LEDR6	PIN_E14	LED [6]	3.3-V LVTTTL
LEDR7	PIN_D14	LED [7]	3.3-V LVTTTL
LEDR8	PIN_A11	LED [8]	3.3-V LVTTTL
LEDR9	PIN_B11	LED [9]	3.3-V LVTTTL

### Using the 7-segment Displays:

The DE10-Lite board has six 7-segment displays to display numbers. The below figure shows the connection of seven segments (common anode) to pins on MAX 10 FPGA. The segment can be turned on or off by applying a low logic level or high logic level from the FPGA, respectively.

Each segment in a display is indexed from 0 to 6 and DP (decimal point), with corresponding positions given in figure below. The table shows the pin assignment of FPGA to the 7-segment displays.



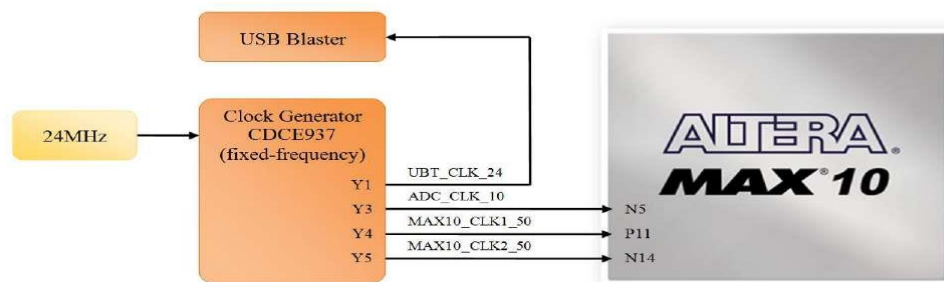
Connections between the 7-segment display HEX0 and the MAX 10 FPGA

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

Signal Name	FPGA Pin No.	Description	I/O Standard
HEX14	PIN_A17	Seven Segment Digit 1[4]	3.3-V LVTTTL
HEX15	PIN_A18	Seven Segment Digit 1[5]	3.3-V LVTTTL
HEX16	PIN_B17	Seven Segment Digit 1[6]	3.3-V LVTTTL
HEX17	PIN_A16	Seven Segment Digit 1[7] , DP	3.3-V LVTTTL
HEX20	PIN_B20	Seven Segment Digit 2[0]	3.3-V LVTTTL
HEX21	PIN_A20	Seven Segment Digit 2[1]	3.3-V LVTTTL
HEX22	PIN_B19	Seven Segment Digit 2[2]	3.3-V LVTTTL
HEX23	PIN_A21	Seven Segment Digit 2[3]	3.3-V LVTTTL
HEX24	PIN_B21	Seven Segment Digit 2[4]	3.3-V LVTTTL
HEX25	PIN_C22	Seven Segment Digit 2[5]	3.3-V LVTTTL
HEX26	PIN_B22	Seven Segment Digit 2[6]	3.3-V LVTTTL
HEX27	PIN_A19	Seven Segment Digit 2[7] , DP	3.3-V LVTTTL
HEX30	PIN_F21	Seven Segment Digit 3[0]	3.3-V LVTTTL
HEX31	PIN_E22	Seven Segment Digit 3[1]	3.3-V LVTTTL
HEX32	PIN_E21	Seven Segment Digit 3[2]	3.3-V LVTTTL
HEX33	PIN_C19	Seven Segment Digit 3[3]	3.3-V LVTTTL
HEX34	PIN_C20	Seven Segment Digit 3[4]	3.3-V LVTTTL
HEX35	PIN_D19	Seven Segment Digit 3[5]	3.3-V LVTTTL
HEX36	PIN_E17	Seven Segment Digit 3[6]	3.3-V LVTTTL
HEX37	PIN_D22	Seven Segment Digit 3[7] , DP	3.3-V LVTTTL
HEX40	PIN_F18	Seven Segment Digit 4[0]	3.3-V LVTTTL
HEX41	PIN_E20	Seven Segment Digit 4[1]	3.3-V LVTTTL
HEX42	PIN_E19	Seven Segment Digit 4[2]	3.3-V LVTTTL
HEX43	PIN_J18	Seven Segment Digit 4[3]	3.3-V LVTTTL
HEX44	PIN_H19	Seven Segment Digit 4[4]	3.3-V LVTTTL
HEX45	PIN_F19	Seven Segment Digit 4[5]	3.3-V LVTTTL
HEX46	PIN_F20	Seven Segment Digit 4[6]	3.3-V LVTTTL
HEX47	PIN_F17	Seven Segment Digit 4[7] , DP	3.3-V LVTTTL
HEX50	PIN_J20	Seven Segment Digit 5[0]	3.3-V LVTTTL
HEX51	PIN_K20	Seven Segment Digit 5[1]	3.3-V LVTTTL
HEX52	PIN_L18	Seven Segment Digit 5[2]	3.3-V LVTTTL
HEX53	PIN_N18	Seven Segment Digit 5[3]	3.3-V LVTTTL
HEX54	PIN_M20	Seven Segment Digit 5[4]	3.3-V LVTTTL
HEX55	PIN_N19	Seven Segment Digit 5[5]	3.3-V LVTTTL
HEX56	PIN_N20	Seven Segment Digit 5[6]	3.3-V LVTTTL
HEX57	PIN_L19	Seven Segment Digit 5[7] , DP	3.3-V LVTTTL

### Clock Circuitry:

Shows the default frequency of all external clocks to the MAX 10 FPGA. A clock generator is used to distribute clock signals with low jitter. The two 50MHz clock signals connected to the FPGA are used as clock sources for user logic. The associated pin assignment for clock inputs to FPGA I/O pins is listed below.



### User-Defined Slide Switch:

There are ten slide switches connected to FPGA on the board (See figure below). These switches are used as level-sensitive data inputs to a circuit. Each switch is connected directly and individually to a pin on the MAX 10 FPGA. When the switch is in the DOWN position (closest to the edge of the board), it provides a low logic level to the FPGA, and when the switch is in the UP position it provides a high logic level. The table list the pin assignments of the user switches is given below.



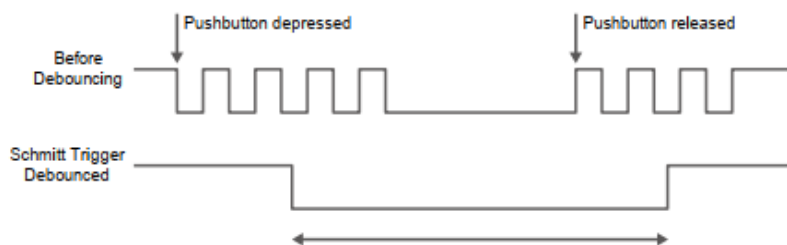
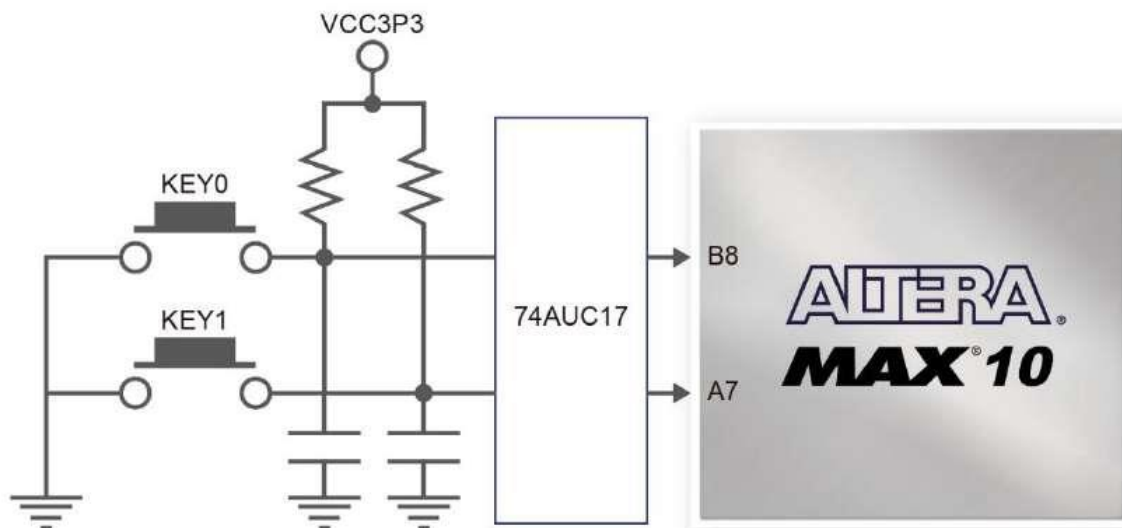
Connections between the slide switches and MAX 10 FPGA

### Pin Assignment of Slide Switches

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

## User-Defined Push-buttons

The board includes two user defined push-buttons that allow users to interact with the MAX 10 FPGA device. Each of these switches is debounced using a Schmitt Trigger circuit as indicated. A Schmitt trigger feature introduces hysteresis to the input signal for improved noise immunity, especially for signal with slow edge rate and act as switch debounce in diagram for the push-buttons connected. Table list the pin assignment of user push-buttons.



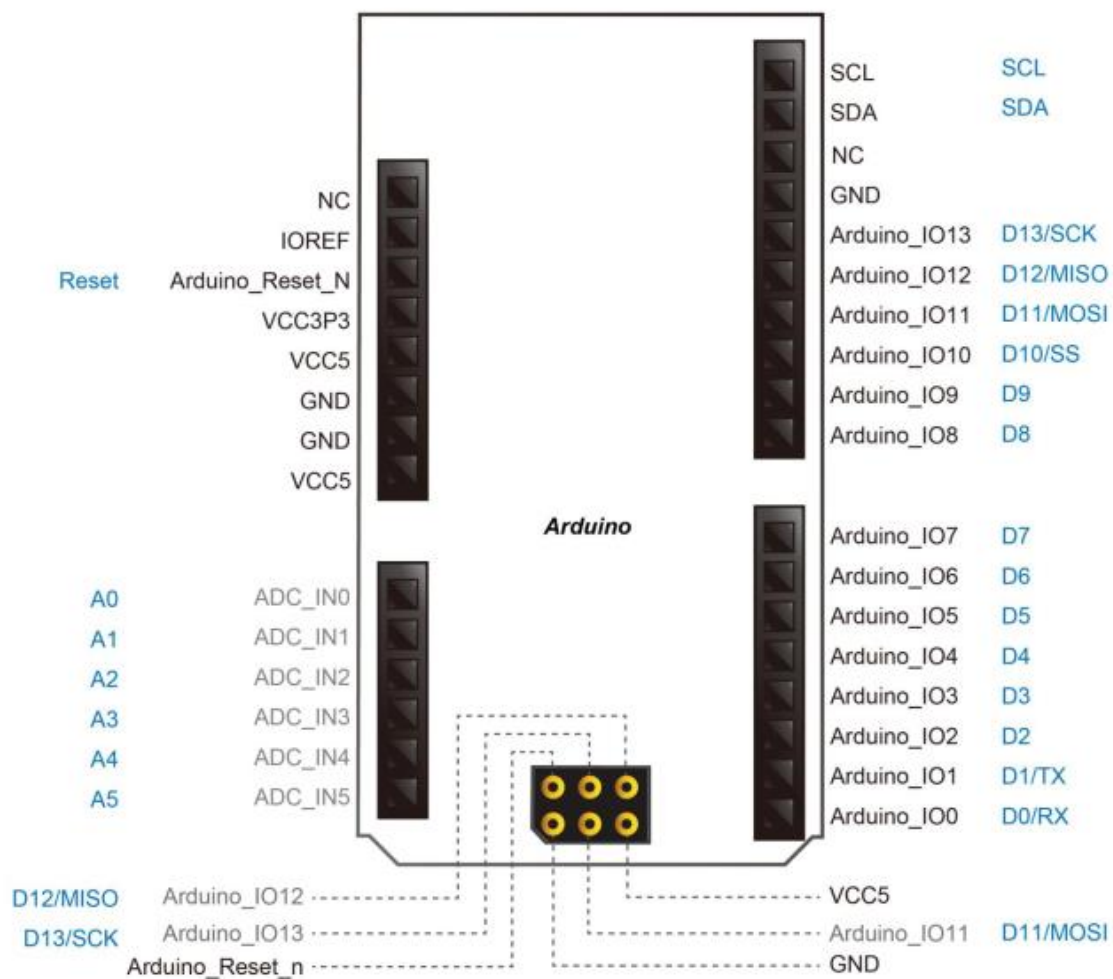
Signal Name	FPGA Pin No.	Description	I/O Standard
KEY0	PIN_B8	Push-button[0]	3.3 V SCHMITT TRIGGER"
KEY1	PIN_A7	Push-button[1]	3.3 V SCHMITT TRIGGER"

## Using Arduino Uno R3 Expansion Header

The board provides Arduino Uno revision 3 compatibility expansion header which comes with four independent headers. The expansion header has 17 user pins (16pins GPIO and 1pin Reset) connected directly to the MAX 10 FPGA. 6-pins Analog input connects to ADC, and also provides DC +5V (VCC5), DC +3.3V (VCC3P3 and IOREF), and three GND pins.

Please refer to below image for detailed pin-out information. The blue font represents the Arduino

Uno R3 board pin-out definition.



The 16 GPIO pins are provided to the Arduino Header for digital I/O. Table below lists all the pin assignments of the Arduino Uno connector (digital), signal names relative to the MAX 10 FPGA.

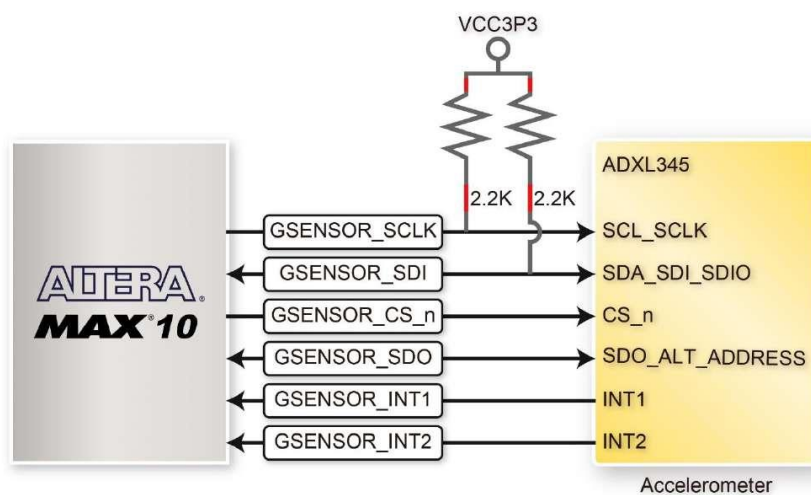
Schematic Signal Name	FPGA Pin No.	Description	Specific features For Arduino	I/O Standard
Arduino_IO0	PIN_AB5	Arduino IO0	RXD	3.3-V LVTTTL
Arduino_IO1	PIN_AB6	Arduino IO1	TXD	3.3-V LVTTTL
Arduino_IO2	PIN_AB7	Arduino IO2		3.3-V LVTTTL
Arduino_IO3	PIN_AB8	Arduino IO3		3.3-V LVTTTL
Arduino_IO4	PIN_AB9	Arduino IO4		3.3-V LVTTTL
Arduino_IO5	PIN_Y10	Arduino IO5		3.3-V LVTTTL
Arduino_IO6	PIN_AA11	Arduino IO6		3.3-V LVTTTL
Arduino_IO7	PIN_AA12	Arduino IO7		3.3-V LVTTTL
Arduino_IO8	PIN_AB17	Arduino IO8		3.3-V LVTTTL
Arduino_IO9	PIN_AA17	Arduino IO9		3.3-V LVTTTL
Arduino_IO10	PIN_AB19	Arduino IO10	SS	3.3-V LVTTTL
Arduino_IO11	PIN_AA19	Arduino IO11	MOSI	3.3-V LVTTTL
Arduino_IO12	PIN_Y19	Arduino IO12	MISO	3.3-V LVTTTL
Arduino_IO13	PIN_AB20	Arduino IO13	SCK	3.3-V LVTTTL
Arduino_IO14	PIN_AB21	Arduino IO14	SDA	3.3-V LVTTTL
Arduino_IO15	PIN_AA20	Arduino IO15	SCL	3.3-V LVTTTL
ARDUINO_RESET_N	PIN_F16	Reset signal, low active.		3.3 V SCHMITT TRIGGER"

## Using Accelerometer Sensor

The board comes with a digital accelerometer sensor module (ADXL345), commonly known as G-sensor. This G-sensor is a small, thin, ultralow power assumption 3-axis accelerometer with high-resolution measurement. Digitalized output is formatted as 16-bit in two's complement and can be accessed through SPI (3- and 4-wire) and I2C digital interfaces.

With GSENSOR\_CS\_N signal to high, the ADXL345 is in I2C mode. With the GSENSOR\_SDO signal to high, the 7-bit I2C address for the device is 0x1D, followed by the R/W bit. This translates to 0x3A for a write and 0x3B for a read. An alternate I2C address of 0x53 (followed by the R/W bit) can be chosen by low the GSENSOR\_SDO signal. This translates to 0xA6 for a write and 0xA7 for a read.

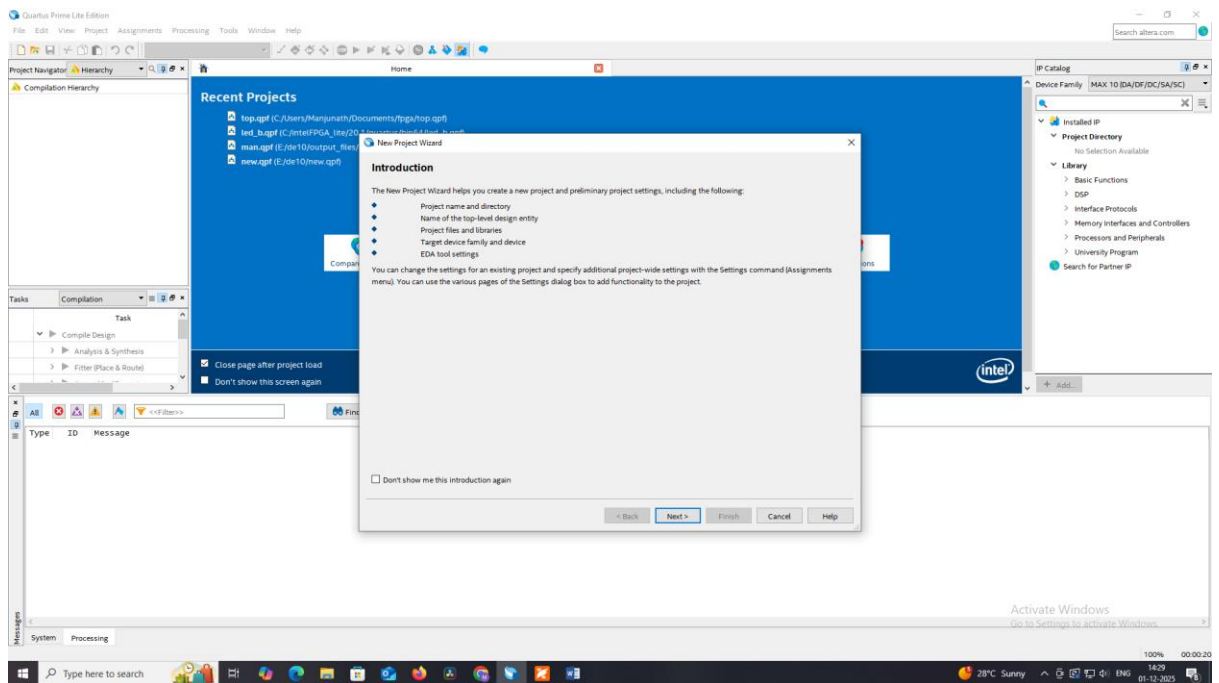
Below diagram shows the connections between the accelerometer sensor and MAX 10 FPGA. Table lists the pin assignment of accelerometer to the MAX 10 FPGA.



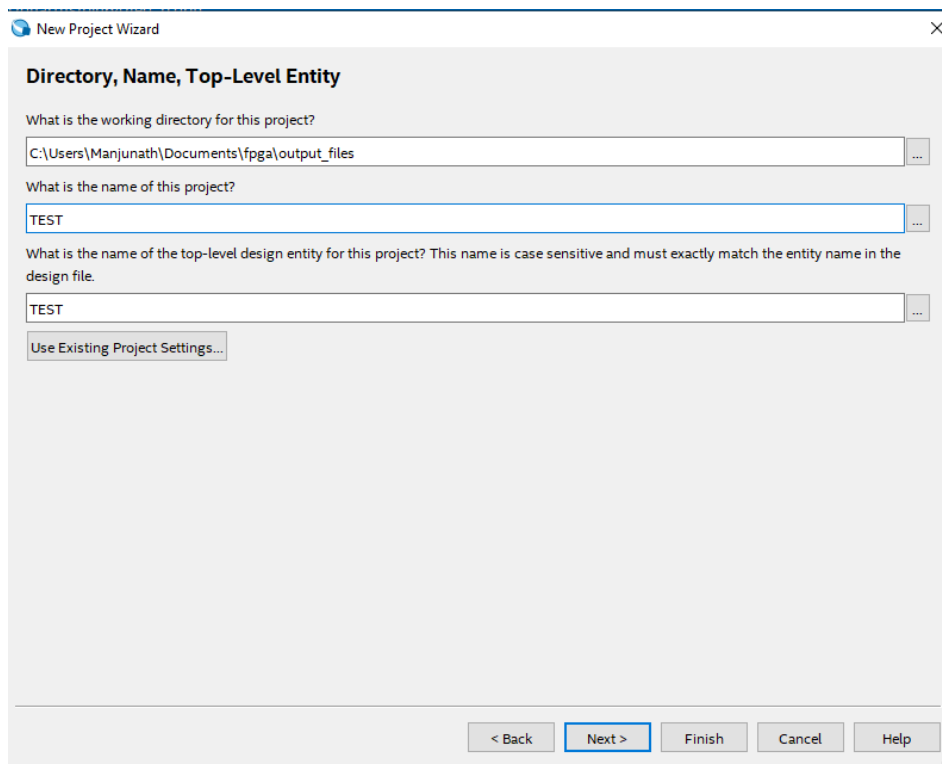
Signal Name	FPGA Pin No.	Description	I/O Standard
GSENSOR_SDI	PIN_V11	I2C serial data SPI serial data input (SPI 4-wire) SPI serial data input and output (SPI 3-wire)	3.3-V LVTTTL
GSENSOR_SDO	PIN_V12	SPI serial data output (SPI 4-wire) Alternate I2C address select	3.3-V LVTTTL
GSENSOR_CS_n	PIN_AB16	I2C/SPI mode selection: 1: SPI idle mode / I2C communication enabled 0: SPI communication mode / I2C disabled SPI Chip Select	3.3-V LVTTTL
GSENSOR_SCLK	PIN_AB15	I2C serial clock SPI serial clock (3- and 4-wire)	3.3-V LVTTTL
GSENSOR_INT1	PIN_Y14	Interrupt pin 1	3.3-V LVTTTL
GSENSOR_INT2	PIN_Y13	Interrupt pin 2	3.3-V LVTTTL

## Programming the ALTERA DE 10 Board

1. Open Intel Quartus software, after it opens go to file and click new project wizard.



2. Click next give the path of your working directory and assign the project name (Note: do not give space or special characters in your project name).



3. Click next and choose “Empty Project” and continue.
4. In the Add files section do not add any files, click next.
5. Family device and Board settings : (VERY IMPORTANT)

Family : MAX 10 (DA/DF/DC/SA/SC)  
 Device : MAX 10 DA  
 Package : FBGA  
 Pin Count : 484  
 Core Speed Grade : 7

In the Available device list select **10M50DAF484C7G**

**Family, Device & Board Settings**

Device | Board

Select the family and device you want to target for compilation.  
 You can install additional device support with the Install Devices command on the Tools menu.

To determine the version of the Quartus Prime software in which your target device is supported, refer to the [Device Support List](#) webpage.

Device family

Family: MAX 10 (DA/DF/DC/SA/SC)  
 Device: MAX 10 DA

Target device

Auto device selected by the Fitter  
 Specific device selected in 'Available devices' list  
 Other: n/a

Show in 'Available devices' list

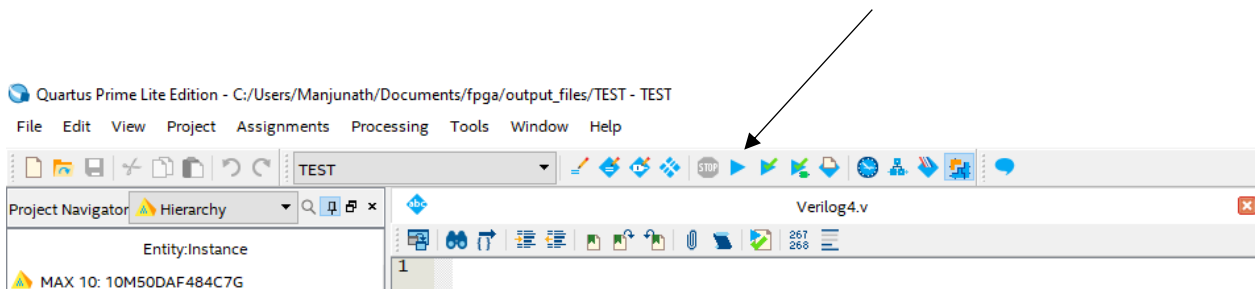
Package: FBGA  
 Pin count: 484  
 Core speed grade: 7  
 Name filter:   
 Show advanced devices

Available devices:

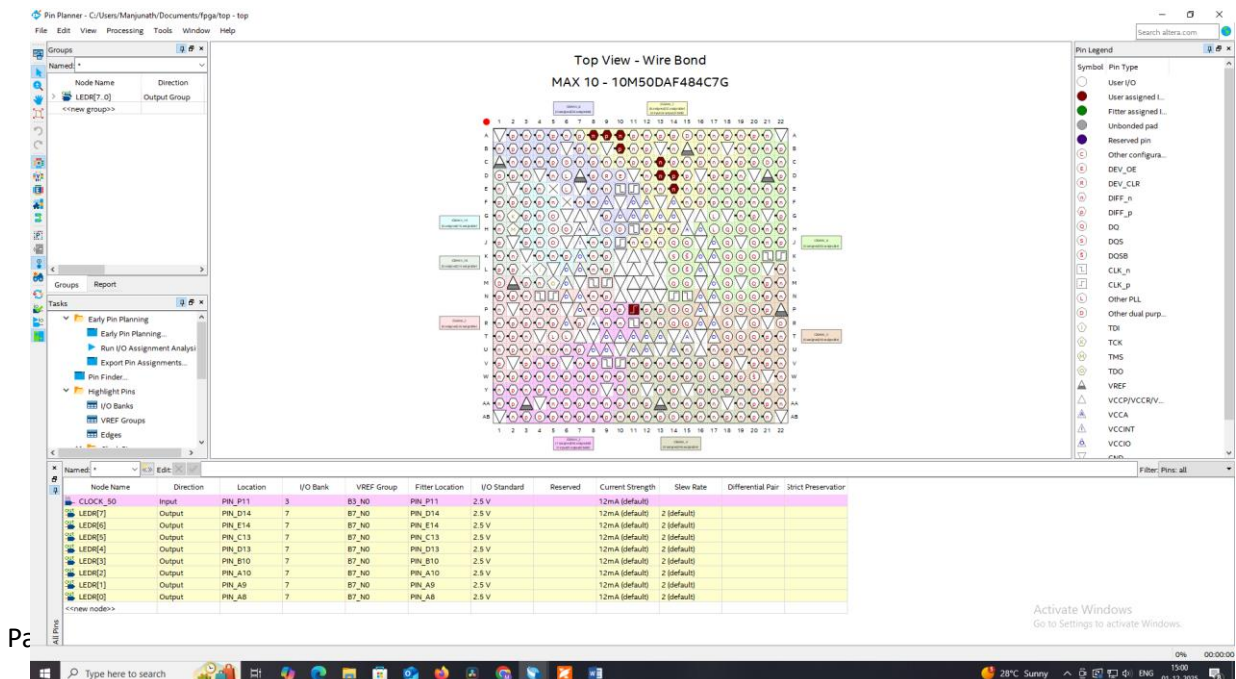
Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits
10M16DAF484I/P	1.2V	15840	320	320	562176
10M25DAF484A7G	1.2V	24960	360	360	691200
10M25DAF484C7G	1.2V	24960	360	360	691200
10M25DAF484I7G	1.2V	24960	360	360	691200
10M40DAF484C7G	1.2V	40368	360	360	1290240
10M40DAF484I7G	1.2V	40368	360	360	1290240
10M50DAF484C7G	1.2V	49760	360	360	1677312
10M50DAF484I7G	1.2V	49760	360	360	1677312
10M50DAF484I7P	1.2V	49760	360	360	1677312

< Back | Next > | Finish | Cancel | Help

6. In the next section “EDA Tools settings” do not edit anything just press next.
7. Once you see the summary page press “finish”.
8. Now our project is ready, we need to create Verilog file.
9. To create file press File menu and select new file. In the file type window select Verilog HDL file.
10. Now write your Verilog code and save it.
11. To check for errors press play button, this will compile your program and check for any errors.



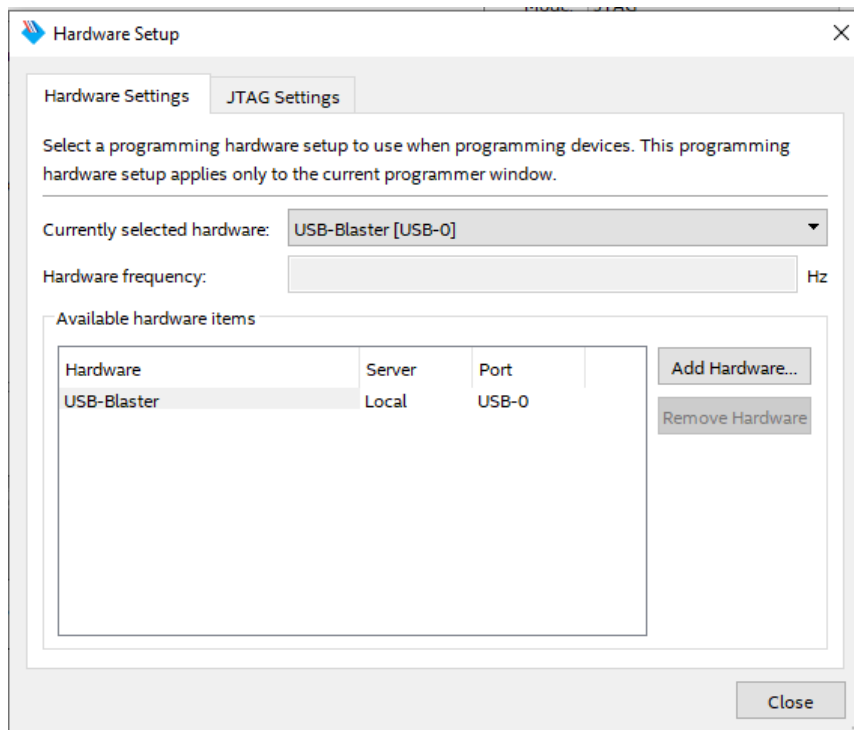
12. Once your program is compiled go to assignments and select pin planner.
13. Assign the pins to respective device which you want to work.  
(Example: Input to switch and output to LED)
14. The hardware pin details of switches, LEDs and seven segments are discussed earlier in this document. Kindly refer the same. (Please note that if you assign any wrong pin then you won't get the required output).



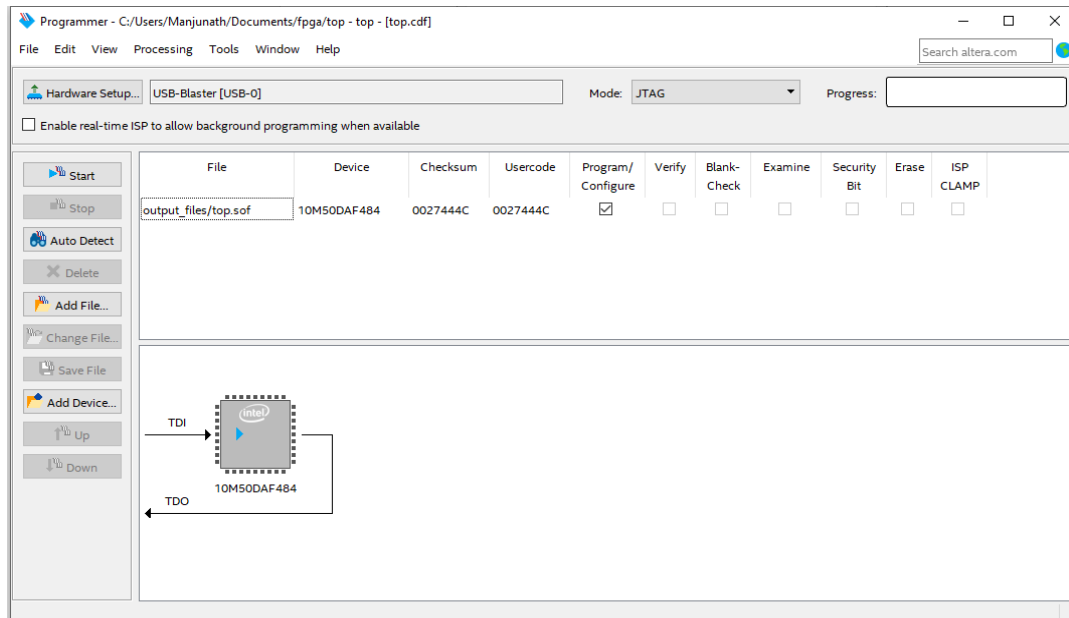
Location where you have to provide the pin details

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Strict Preservation
in CLOCK_50	Input	PIN_P11	3	B3_NO	PIN_P11	2.5 V		12mA (default)			
OUT LEDR[7]	Output	PIN_D14	7	B7_NO	PIN_D14	2.5 V		12mA (default)	2 (default)		
OUT LEDR[6]	Output	PIN_E14	7	B7_NO	PIN_E14	2.5 V		12mA (default)	2 (default)		
OUT LEDR[5]	Output	PIN_C13	7	B7_NO	PIN_C13	2.5 V		12mA (default)	2 (default)		
OUT LEDR[4]	Output	PIN_D13	7	B7_NO	PIN_D13	2.5 V		12mA (default)	2 (default)		
OUT LEDR[3]	Output	PIN_B10	7	B7_NO	PIN_B10	2.5 V		12mA (default)	2 (default)		
OUT LEDR[2]	Output	PIN_A10	7	B7_NO	PIN_A10	2.5 V		12mA (default)	2 (default)		
OUT LEDR[1]	Output	PIN_A9	7	B7_NO	PIN_A9	2.5 V		12mA (default)	2 (default)		
OUT LEDR[0]	Output	PIN_A8	7	B7_NO	PIN_A8	2.5 V		12mA (default)	2 (default)		
<<new node>>											

- Once you have done the pin assignment, re-compile the entire program this will ensure that pin assignments are fetched into the program.
- Now open the tools menu and select programmer.
- Now connect your board to the computer and make sure the board is detected in the programmer.
- Open the **Hardware Setup** utility. Successful detection of your DE10-Lite board is confirmed when you see **USB-BLASTER [USB-0]** listed. If the board is not detected, cycle the connection by trying an alternative USB port on your computer.



19. Once the hardware setup is successfully completed and the board is detected, the Quartus Prime Programmer window should automatically populate with your compiled configuration file. The programmer is designed to auto-detect the **.sof** (SRAM Object File) that corresponds to the project currently open or recently compiled. You can verify the file by checking its name, which will typically match the name of your Quartus project.



20. If the **.sof** file is missing, click **Add File** and browse to your project directory; the configuration file will be located inside the **output\_files** folder. Should the file still be unavailable after this manual search, you must **recompile** your program within Quartus Prime to ensure a successful build.
21. Once **.sof** file is available click Program/Configure and proceed with Start button to program your FPGA board. Once it is programmed you can verify the logic.

----- Thank you -----