# Getting Started with Intel® Quartus® Prime v21.0 and the Intel® MAX® 10-10M08 Evaluation Kit

By Sean D. Liming and John R. Malin Annabooks – <u>www.annabooks.com</u>

October 2022

The Intel Max 10 is the lower cost and lower end of the Intel FPGA family. The Max 10 FPGAs are a popular solution for many applications. The Max 10 can be found on development boards featuring the higher-end Intel FPGAs. The MAX 10 -10M08 Evaluation Kit developed by Terasic Inc. is a very simple development board to help one get started developing with the Quartus Prime software. As a getting started to our hands-on articles on Intel FPGAs with the latest software, this article walks through a couple of basic designs to demonstrate the latest Quartus Prime V21.0 software targeting the MAX 10 -10M08 Evaluation Kit.

The Project Requirements:

- Intel Quartus Prime Lite Edition V21.0 7zip is required to extract the tar file.
- Intel® MAX® 10 10M08 Evaluation Kit and the schematic for the evaluation board are required. The schematic PDF file can be downloaded from the Intel FPGA website.
- Intel FPGA Programming cable USB Blaster II or EthernetBlaster II. Unlike other FPGA boards, the MAX 10 -10M08 Evaluation Kit doesn't have a built-in USB Blaster solution, so a separate programming cable is required.

**Note**: There are equivalent MAX 10 development and evaluation boards available. These boards can also be used as the target, but you will have to adjust to the available features on the board. Please make sure that you have the board's schematic files as these will be needed to identify pins.

### 1.1 Download and Install the Quartus Prime Lite Edition Software

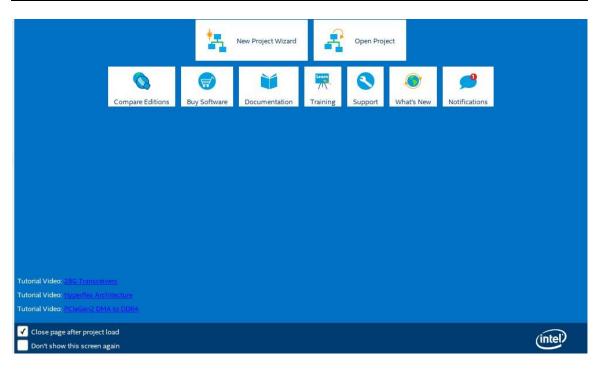
The Quartus can run on Windows and different Linux distributions. The installation of the software has many steps. Please see the article <u>Intel® Quartus® Prime Lite and NIOS® II SBT for Eclipse</u> <u>Installation Instructions</u> on Annabooks.com to install the software needed for this hands-on exercise. All you need for this hands-on paper is the installation of the Quartus Prime Lite software with Max 10 support.

### 1.2 LED Toggle Project

As hello world is the first programming project, toggling an LED is the first electronic circuit project. The MAX 10 -10M08 Evaluation Kit has a set of 5 LEDs on board available for user control. One of the LEDs will be toggled on and off using the 50Mhz clock on the board.

### 1.2.1 Create the Project

- 1. Open Quartus.
- 2. Click on the New Project Wizard.



- 3. Click Next to the Introduction dialog.
- 4. Select or create a new project directory (other than the Quartus installation directory) and name the project "LedToggle". Click Next.

**Note**: By default, the root directory is the Quartus installation directory. Make sure the root project directory is a separate path from the Quartus installation files. i.e. c:\MAX10DesignExamples

- 5. Project Type: Empty project, click Next.
- 6. Add File: no files to add, click Next.
- 7. Family, Device & Board Settings: click the Board tab and select: MAX 10 FPGA 10M08 Evaluation Kit and click Next.

elect	the board/development kit you want to target	for compila	ation.					
amily	<i>r.</i> MAX 10	•	Develo	pment Kit:	Any			¥
<u>v</u> ailał	ble boards:							
	Name	Versi	ion	Fami	ly	Device	Vendor	
	Arrow MAX 10 DECA	0.9		MAX 10		10M50DAF484C6GES	Arrow	4976
	BeMicro MAX 10 FPGA Evaluation Kit	1.0		MAX 10		10M08DAF484C8GES	Arrow	8064
	MAX 10 DE10 - Lite	1.0		MAX 10		10M50DAF484C6GES	Altera	4976
<b>.</b>	MAX 10 FPGA 10M08 Evaluation Kit	1.0		MAX 10		10M08SAE144C8GES	Altera	8064
	MAX 10 FPGA Development Kit	1.0		MAX 10		10M50DAF256C7G	Altera	4976
	MAX 10 NEEK	1.0		MAX 10		10M50DAF484I7G	Terasic	4976
	Odyssey MAX 10 FPGA Kit	1.0		MAX 10		10M08SAU169C8GES	Macnica	8064
Cre	cate top-level design file.							4

- 8. EDA Tools: click Next.
- 9. Summary: click Finish.

**Note**: The actual MAX 10 on our board is the 10M08SAE144C8G, thus it is not an Engineering Sample (ES). The next two steps change the device to the production device. Your experience with your hardware might be different.

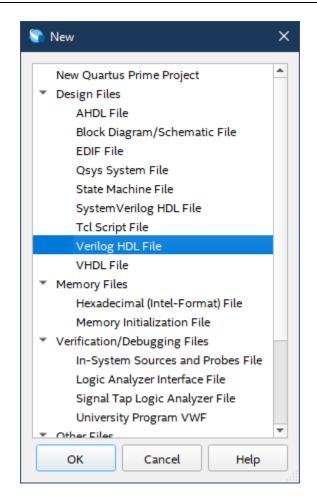
- 10. In the project navigation pane on the left, right-click on 10: 10M08SAE144C8GE, and select Device from the context menu.
- 11. In the Available devices, scroll down and select the 10M08SAE144C8G, click OK.

Device Board										
Select the family and de You can install addition To determine the versio	al device support v	vith the In	istall Devices co				refer to the	e <u>Device Su</u>	pport List	webpa
Device family				Show in 'Av	ailable de	vices' li	st			
Eamily: MAX 10 (DA	/DF/DC/SA/SC/SI	.)	•	Pac <u>k</u> age:		Any				•
Dev <u>i</u> ce: All			•	Pin <u>c</u> ount		Any				¥
Target device				Core sp <u>e</u> e	d grade:	Any				•
<ul> <li><u>A</u>uto device selection</li> <li><u>S</u>pecific device selection</li> </ul>		devices'	list	Name filte		device	25			
O Other: n/a				Device and	Pin Optic	ons				
A <u>v</u> ailable <mark>d</mark> evices:				-1 eV.						
Name	Core Voltage	LEs	Total I/Os	GPIOs	Memor	y Bits	Embe	dded multi	iplier 9-b	it elem
10M08SAE144C8G	3.3V	8064	101	101 3	87072		48			
10M08SAE144C8GES	3.3V	8064	101	101 3	87072		48			
10M08SAE144I7G	3.3V	8064	101	101 3	87072		48			
	3 3V	8064	101	101 -	87072		48			Þ
10M0854F144I7P										

#### 1.2.2 Create the Verilog file

Quartus supports many design types to create an FPGA design. Verilog is a popular Hardware Definition Language (HDL) that is like the C programming language. We will create a Verilog file to have the 50Mhz clock toggle the LED.

1. From the menu, select New, and from the New dialog, select Design Files--> Verilog HDL File.



2. In the text area, write the following code:

//LED Toggle
//New Module
module LedToggle (
input wire clk_50Mhz,
output wire LEDD1
);
//Create a register
reg [31:0] counter;
//Assign the LED to one of the bits in the counter
assign LEDD1 = counter[25];
initial begin
counter <= 32'h0;
end
always @(posedge clk_50Mhz) begin

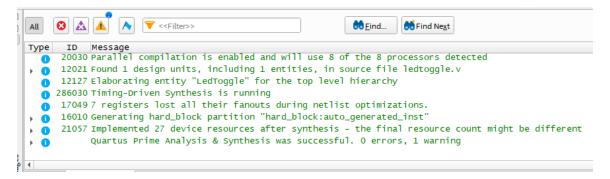
counter <= counter + 1;

Copyright © 2022 Annabooks, LLC. All rights reserved

end
endmodule

The Verilog file is considered a module. Within the module creation, the input and output pins are defined for the clock and the LED. A 32-bit register is then created, and the LED is assigned to bit 26 of the register. The register is set to 0h on startup. The always statement says that on every positive edge of the clock, the register will be incremented by one. The counter will roll over once the maximum count is reached. The LED will turn on and off as the 26<sup>th</sup> bit is set to 1 and 0.

- 3. Save the file as LedToggle.v to the project folder.
- 4. Now let's test compile the code to make sure there is nothing mistyped. Under Task, rightclick on Analysis & Synthesis, and select Start from the context menu. The output should show that the compilation was successful. If there are any errors, fix them and recompile the project.



 The pin assignments need to be set. From the menu, select Assignments->Pin Planner. The wire keyword populated the input and output in the Node Name list at the bottom of the Pin Planner dialog.



■ <b>► 0 ≯</b> 1 ∩ с	iport Report not available	I Ø B			144 145	142 141 140 139 138 13		man j Project Projectel			18.10		Pin	Search Intel	<b>P</b> Ø Ø
1 ■ ► ■ ■ ■ ■ ■		100			• AA			Concept d' Provigentine							
N ≤					• AA				77 126 126 126 126 128 1	22 121 120 110 118 1	T 116 115 114 115 11	2 111 110 109	Su	mbol Pin Type	
ر پ پ پ					1		/•p 0 •n 🗸	p •n •p C 0 •						User I/O	
ر الج												108		User assigned I/	
د   <b>۲</b> ♦					z 🔺							107		Fitter assigned I	
د   <b>۲</b> ♦					3 💿							E 106			
2 Z					4 0							6 105		Reserved pin	
2					5 O							0 103	c		
2					7			-				n 102			
								гор	View			101			
					9 0							*p 100			
C				A Area (10	10 -0		Mire I	and wi	th Eve		-1	• p 99			
6					11 -0		wire	3ond, wi	in Expo	seu Pa	L L	• 98	۲		
					12 - p							97			
_00_					13 - 6							P 96 March and and a			
≣₩≣					14 <b>*</b> p							95	6		
≡ <mark>n</mark> ≣					15 D 16 H							0 94	٥		
- M-					17							- 92			
E	Groups Report				10 💽							1 91	8		
	isks	708			19 1							_ ∞ _		TMS	
<b>#1</b> **	* 芦 Early Pin Plannir			100.00.00	20 0							1 89		TDO	
	Early Pin Pla				21 • n							J 88		VREF	
2	Run I/O Assi				22 - p							· 0 87		VCCP/VCCR/	
	Export Pin A				23 0							• 0 00		VCCA	
Ρ	Pin Finder_	signments			25 * 0				× 40			• 0 84	٨	VCCID	
	<ul> <li>Pin Finder</li> <li>Highlight Pins</li> </ul>				26 1			MA	X 10			63			
-	<ul> <li>Highlight Pins</li> <li>I/O Banks</li> </ul>				27 5							0 82			
0	VREF Group				20 1							- n a1			
0					29 5		-	014000		200		an and a company of the second			
88	Edges			(8.00.) (10.00.01.01.01)			1	0M08S/	4E1440	.8G		• p 79			
00	* 芦 Clock Pins				31 0							0 78			
<b>6</b>	TTT Clock				52 L							• 77 • 76			
<b>V</b>	III PLL/DLL				34 (A)							+p 75			
10/	PLL/DLL				35 A							*p 74			
	Clock Re				36							73			
io	Differential F	ins					) +p +n +p 🛆		9 n p n p		n • p o <u>o </u> v • n				
					37 38	39 40 41 42 43 4			4 55 56 57 58	19 60 61 62 63 (		70 71 72			
								rano, a agent? a agente			10000.4 Biological Prospected				
-	8 Named: • • • 8	> Edit × ✓												Filter P	
8	3													Filter: P	ins: all
4	Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate	Differential Pair	Strict Preservation				
	LEDD1	Output				2.5 V (default)			2 (default)						
	<cnew node="">&gt;</cnew>	Input				2.5 V (default)		12mA (default)							
Dine	50														
														c	00:00:00

 Using the board schematic, locate the pins for the LEDD1 and the 50Mhz clock. Set the Location values for both node names. For the MAX 10 – 10M08 Evaluation Board, these values are as follows:

LEDD1: PIN\_132 Clk\_50MHz: PIN\_27

7. Set the I/O Standard to 3.3V-LVTTL. You can see from the schematic that the I/O are all tied to 3.3V.

	Named: *	Edit: 🔀 🗹							
0 I	Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate
	LEDD1	Output	PIN_132	8	B8_N0	3.3-V LVTTL		8mA (default)	2 (default)
	in_ clk_50Mhz	Input	PIN_27	2	B2_N0	3.3-V LVTTL		8mA (default)	
	< <new node="">&gt;</new>								

The chip pin diagram will show that both pins are assigned. You can double-click on a pin and the properties pane will appear with the pin information.

Pin	Properties		Ţ	0 X
Pin	number	PIN_13	2	
No	de name:	LEDD1		•
I/O	Standard:	3.3-V I	LVTTL	*
Res	erved:			-
Pro	perties:			
Na	me		Value	-
	I/O bank		8	
	VREF grou	цр	B8_N0	
	Edge		ТОР	
	General fu	nction	Column I/O	
Ŧ	Special fu	nction		
			DIFFIO_RX_T22p	
			DIFFOUT_T22p	
			Low_Speed	
	Pad ID		240	
	VREF pad	ID	231	
	Pad group	)	12	
				Ŧ

- 8. Close the Pin Planner when finished.
- 9. Finally, create a timing constraints file. From the menu, select New->Text File.
- 10. Enter the following:

#### **#Set timing constraint of clk\_50Mhz is 50Mhz**

create\_clock -name clk\_50Mhz -period "50Mhz" [get\_ports clk\_50Mhz]

#### #LEDD1 has no timing constraints

#### set\_false\_path -from \* -to [get\_ports LEDD1]

- 11. Save the file as LedToggle.sdc to the project folder.
- 12. Finally, compile the design. In the Task pane, right-click on Compile and Design and select Start from the context menu, or you can click on the Play symbol in the toolbar. Make sure the project compiles successfully.

Flow Summary	
< <filter>&gt;</filter>	
Flow Status	Successful - Sat Jun 25 17:28:59 2022
Quartus Prime Version	21.1.0 Build 842 10/21/2021 SJ Lite Edition
Revision Name	LedToggle
Top-level Entity Name	LedToggle
Family	MAX 10
Device	10M08SAE144C8G
Timing Models	Final
Total logic elements	26 / 8,064 ( < 1 % )
Total registers	25
Total pins	2 / 101 ( 2 % )
Total virtual pins	0
Total memory bits	0 / 387,072 ( 0 % )
Embedded Multiplier 9-bit elements	0 / 48 (0 %)
Total PLLs	0/1(0%)
UFM blocks	0/1(0%)
ADC blocks	0/1(0%)

#### 1.2.3 Program the Board

With the design compiled, we can now test the design on the board.

1. Connect the board and the programming cable together per the cable instructions.

**Note**: The MAX 10 – 10M08 Evaluation Kit doesn't come with a programming cable or built-in JTAG USB Blaster II. You will have to use either the USB Blaster II or EthernetBlaster II external cables. The EthernetBlaster II was used in this example. DHCP setup was not working, so a direct Ethernet cable connection between a PC and the EthernetBlaster II had to be made. Set the static IP for the PC network card to 198.162.0.1. Access the EthernetBlaster II via a browser and then change the IP to a static IP that matches the network. The new IP address was used as the Server name in this example. Your experience might be different.

- 2. Power on the board and the programming cable box.
- 3. In Quartus Prime, from the Task pane, right-click on Program Device (Open Programmer) and select Open from the context menu.
- 4. The Programmer dialog appears. Click on the "Hardware Setup" button.
- 5. Click the Add hardware button, select the Hardware type and file and any remaining information, and click OK.

Hardware Settings	JTAG S	Settings		
Select a programmin hardware setup appli				ces. This programming
Currently selected ha	rdware:	No Hardware		
Hardwore from on or Add Hardward	2		×	F
Hardware type:	Ethern	etBlaster	•	Add Hardware
Port:			Ŧ	Remove Hardware
Baud rate:			•	
Server name:			•	
Server port:	1309			
Server password:				
Auto Detect		ок	Cancel	

6. The tool allows you to connect to a number of programming cables. We need to select the one for our board. In the "Currently selected hardware", click the drop-down, select the hardware cable for the board, and click Close when finished.

Hardware Settings JTAG S	ettings		
Select a programming hardwar hardware setup applies only to			ces. This programming
Currently selected hardware:	EthernetBlasterII	on 192.168.1.198 [Et	hernetBlasterII] 🔹
Hardware frequency:			н
Available hardware items			
Hardware	Server	Port	Add Hardware
EthernetBlasterII	192.16	8.1 EthernetBl	Remove Hardware

7. A LedToggle.sof file gets created during the Compile Design flow. The file is automatically filled in. There is only one FPGA on the board and in the JTAG chain, so the file already has the Program/Configure checkbox checked. Click the Start button.

Hardware Setup	. on 192.168.1.198 [EthernetE	lasterii] Mode:	JTAG	•	Progress:			
Enable real-time I	SP to allow background program	ming when available						
▶ <sup>™</sup> Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
Stop	output_files/LedToggle.sof	10M08SAE144	00088D26	00088D26	✓			
Auto Detect								
X Delete								
Add File								
Change File	4							
Save File								
Add Device	TDI							
1 <sup>™</sup> ∪р	│ <b>──→</b>							
J <sup>™</sup> Down	:							
	10M08SAE144 TDO							
	100							

The process takes a few seconds and shows that the task was completed successfully.

Hardware Setup Enable real-time	ISP to allow background program		JTAG	•	Progress:		100% (Suc	cessful)
▶ <sup>™</sup> Start	File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank- Check	Examine
Auto Detect X Delete	4							
Change File Save File Add Device The Up	TDI 10M085AE144 TDO							

If you look at the board, you will see that the LED D1 is blinking.

### 1.2.4 Program the Internal Flash with POF file.

The downloading of the .SOF file was to the SRAM of the MAX 10. Once the system has been power cycled the design is lost. In this section, we will convert the LedToggle.sof to LedToggle.pof that will be programmed into the internal flash and stay resident.

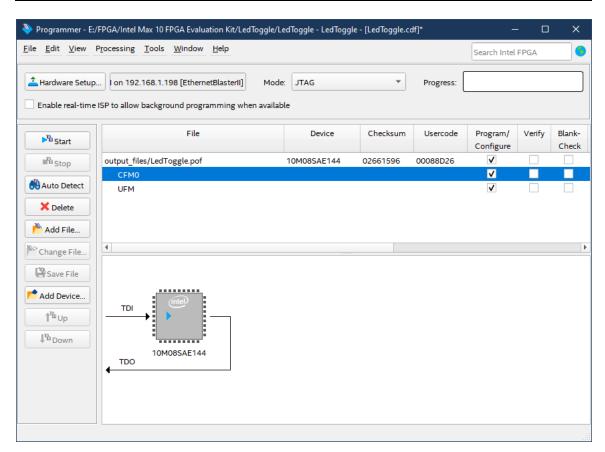
- 1. Unplug the power for the Evaluation Board, wait a few seconds, and then plug the power back in. You will notice that the design doesn't come back up and flash the LED. The .SOF file is downloaded to the internal SRAM and lost on a power cycle.
- The MAX 10 has internal flash that provides non-volatile storage for the program allowing it to persist through a power-down and launch on power on. We need to convert the .SOF file to a .POF file to program the internal flash. In Quartus, from the menu, select File->Convert Programming Files...
- 3. In the Convert Programming Files dialog set the following:

Programming file time: Programmer Object File (.pof). Mode: Internal Configuration. File name: output\_files/LedToggle.pof.

4. Under the Input files to convert, click Add File and browse to the output folder location to open the LedToggle.sof file.

le <u>T</u> ools <u>W</u> indow					Search Int	tel FPGA
pecify the input files to o ou can also import inpu ture use.				o information crea	ated here for	
onversion setup files						
Ор	en Conversion Set	up Data		Save Cor	version Setup	
utput programming file	•					
Programming file type:	Programmer Ob	ject File (.pof)				Ŧ
Options/Boot info	Configuration de	vice: EPCE16	•	<u>M</u> ode:	Internal Configu	ration -
File <u>n</u> ame:	output_files/Led	Toggle.pof				
Advanced	Remote/ <u>L</u> ocal up	date difference file:	NONE			~
		data RPD (Generate Le	dToggle_auto.rpd)			
File/Data :	area	Properties	Start Addre	55		Add He <u>x</u> Dat
	area			55		
File/Data : * SOF Data	area	Properties Page_0	Start Addre	55		
File/Data : * SOF Data	area	Properties Page_0	Start Addre	55		Add <u>S</u> of Pag
File/Data : * SOF Data	area	Properties Page_0	Start Addre	55		Add <u>S</u> of Pag Add <u>E</u> ile
File/Data : * SOF Data	area	Properties Page_0	Start Addre	55		Add <u>S</u> of Pag Add <u>E</u> ile Remove
File/Data : * SOF Data	area	Properties Page_0	Start Addre	55		Add <u>S</u> of Pag Add <u>E</u> ile Remove Up
SOF Data	area	Properties Page_0	Start Addre	55		Remove Up Down

- 5. Click the Generate button. A dialog will appear stating generation was success full. Click OK.
- 6. Click Close to close the Convert Program File dialog.
- 7. In Quartus Prime, from the Task pane, right-click on Program Device (Open Programmer) and select Open from the context menu.
- 8. The programming cable should already be selected. If not, select your programming cable.
- 9. In the file list, right-click on the LEDToggle.sof file name and select Change File.
- 10. Open the LedToggle.pof file.
- 11. Check the Program/Configure checkbox. This will automatically check the CFMo and UFM.



- 12. Click the Start button. The process will take a little longer since the internal flash is being programmed.
- 13. LED D1 should be blinking. Power cycle the board and the program should persist through the power cycling and start up and run this time.

### 1.3 Logic Switches Project

The previous project used Verilog to create the design. In the project, we will create a design using a schematic. The MAX® 10 - 10M08 Evaluation Kit has a 5-switch pack on board. Three switches will be used to light up LED D2 and LED D3 as follows:

	Input		Ou	tput
Switch1	Switch2	Switch3	LEDD2	LEDD3
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	1
1	0	0	1	0
1	0	1	1	1
1	1	0	1	0
1	1	1	1	1

Looking at the schematic, the LEDs are tied to 3.3 V. The LEDs turn on when the output pin is low. When the switches slide to the ON position, they are in the logic 0 state, and when the switches slide to the OFF position, they are in the logic 1 state.

#### 1.3.1 Create the Project

- 1. Open Quartus.
- 2. Click on the New Project Wizard.

		ね	New Project Wizard	2	Open Proj	ect	
	Compare Editions	Buy Software	Documentation	Training	Support	What's New	Notifications
Tutorial Video: <u>286 Transcer</u> Tutorial Video: <u>Hyperflex Arc</u> Tutorial Video: <u>PCleGen2 DM</u>							
✓ Close page after project Don't show this screen a							

- 3. Click Next to the Introduction dialog.
- 4. Select the directory (other than the Quartus directory) and name the project: SwitchLogic. Click Next.

**Note**: By default, the root directory is the Quartus installation directory. Make sure the root project directory is a separate path from the Quartus installation files. i.e. c:\MAX10DesignExamples.

- 5. Project Type: Empty project, click Next.
- 6. Add File: There are no files to add, click Next.
- 7. Family, Device & Board Settings: click the Board tab and select: MAX 10 FPGA 10M08 Evaluation Kit. Click Next.

	te board/development kit you want to target	for compila	ation.					
amily:	MAX 10	•	Develo	pment Kit:	Any			¥
<u>v</u> ailable	e boards:							
	Name	Versi	ion	Fami	ly	Device	Vendor	
Ar	rrow MAX 10 DECA	0.9		MAX 10		10M50DAF484C6GES	Arrow	4976
Ве	eMicro MAX 10 FPGA Evaluation Kit	1.0		MAX 10		10M08DAF484C8GES	Arrow	8064
м	AX 10 DE10 - Lite	1.0		MAX 10		10M50DAF484C6GES	Altera	4976
≡ м	AX 10 FPGA 10M08 Evaluation Kit	1.0		MAX 10		10M08SAE144C8GES	Altera	8064
в	AX 10 FPGA Development Kit	1.0		MAX 10		10M50DAF256C7G	Altera	4976
м	AX 10 NEEK	1.0		MAX 10		10M50DAF484I7G	Terasic	4976
	dyssey MAX 10 FPGA Kit	1.0		MAX 10		10M08SAU169C8GES	Macnica	8064
Creat	te top-level design file.							•

- 8. EDA Tools: click Next.
- 9. Summary: click Finish.

**Note**: The actual MAX 10 on our board is the 10M08SAE144C8G, thus it is not an Engineering Sample (ES). The next two steps change the device to the production device. Your experience might be different depending on your hardware.

- 10. In the project navigation pane on the left, right-click on 10: 10M08SAE144C8GE and select Device from the context menu.
- 11. In the Available devices, scroll down and select the 10M08SAE144C8G. Click OK.

Device Board							
Select the family and de You can install addition To determine the versio	al device support v	vith the Ir	istall Devices co				efer to the <u>Device Support List</u> webpa
Device family				Show in 'Av	/ailable de	vices' li	st
Eamily: MAX 10 (DA	/DF/DC/SA/SC/SL	.)	*	Pac <u>k</u> age:		Any	*
Dev <u>i</u> ce: All			•	Pin <u>c</u> oun	t	Any	×
Target device				Core sp <u>e</u>	ed grade:	Any	•
<ul> <li><u>A</u>uto device select</li> <li><u>S</u>pecific device se</li> </ul>		devices'	list	Name filt ✓ S <u>h</u> ow	er: advanced	device	5
O <u>O</u> ther: n/a				Device and	Pin Optic	ons	
A <u>v</u> ailable devices:							
Name	Core Voltage	LEs	Total I/Os	GPIOs	Memor	y Bits	Embedded multiplier 9-bit elem
10M08SAE144C8G	3.3V	8064	101	101	387072		48
10M08SAE144C8GES	3.3V	8064	101	101	387072		48
10M085AE144I7G	3.3V	8064	101	101	387072		48
	3 3V	8064	101	101	387072		48
10M0854F144I7P							

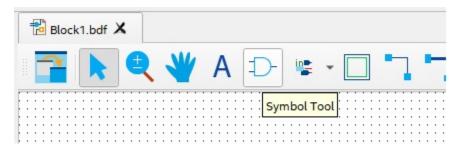
### 1.3.2 Create the Schematic File

Now we will create the schematic file.

- 1. Open Quartus.
- 2. From the menu, select New; and from the New dialog, select Design Files--> Block Diagram/Schematic File.

🕤 New	×
New Quartus Prime Project Classing Files AHDL File	•
Block Diagram/Schematic File	
EDIF File	
Qsys System File	
State Machine File	
SystemVerilog HDL File	
Tcl Script File	
Verilog HDL File	
VHDL File	
<ul> <li>Memory Files</li> </ul>	
Hexadecimal (Intel-Format) File	
Memory Initialization File	
<ul> <li>Verification/Debugging Files</li> </ul>	
In-System Sources and Probes File	
Logic Analyzer Interface File	
Signal Tap Logic Analyzer File	
University Program VWF	
Other Eiler	×
OK Cancel Help	

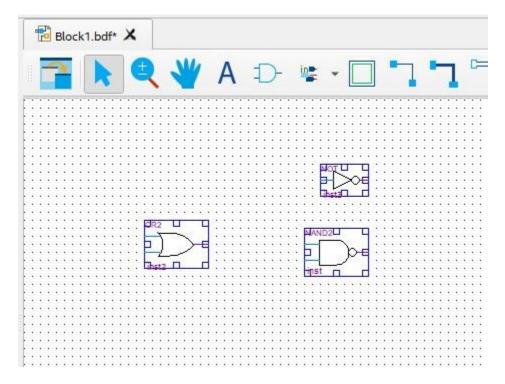
3. A blank schematic editor appears. In the tools bar, click on the AND symbol to open the Symbol Tool, or double-click anywhere in the empty space of the design.



4. The Symbol dialog appears, which contains the libraries of available symbols from which to choose. Expand the hierarchy to get to the logic symbols: primitives->logic.

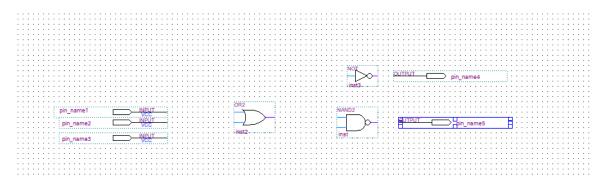
Symbol		
Libraries:		
▼ 🖻 c:/intelfpga_lite/21.1/quartus/li		
▶ 🛱 megafunctions		
In the others		
👻 🖻 primitives		
▶ 🛱 buffer		
🔻 🗁 logic	NAND2	
다 and12	inst	
▼ + 12 and 2		
Name:		
nand2		
✓ <u>R</u> epeat-insert mode		· · · · · · · · · · · · · · · · · · ·
Insert symbol as block		
	ОК	Cancel

- 5. Scroll down the list and click on nand2. This is a 2-input NAND gate.
- 6. Click Ok. The dialog will disappear and the mouse will have the symbol attached.
- 7. Drag and place (right-click) the NAND gate on the schematic diagram.
- 8. The tool allows you to add more NAND gates, and we only need one. Hit the ESC key when finished.
- 9. Repeat the steps to add an OR gate (or2) and a NOT gate (inverter).



- 10. Open the Symbol Tool, and expand the hierarchy to primitives->pin.
- 11. Select input and click OK.
- 12. Drop the input symbol three times and then hit ESC.
- 13. Repeat steps 10-12 to add 2 output pins.

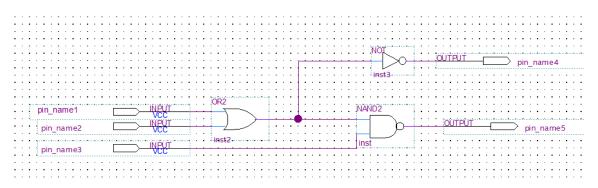
Copyright © 2022 Annabooks, LLC. All rights reserved



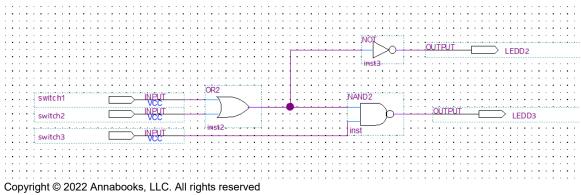
14. From the toolbar, click on the Orthogonal Node Tool. The tool is used to connect the wires between the gates and pins.

	[	•	R	3	в	0	ck	1	.b	d	f*		×																																								
				1	7					8			(				-	1	1	1	ł	4			-		)	-	in			•					•	-			•	-	1		C	7	2	ļ	•	~	-		•
1	ł	:	•	:	:		:	:	:	:	:	:	:	:	:	 :	•	• • •	:	:	 		:		:	:	:		:	:	:			:	:	•	• •		•	(	Dr	th	oį	go	na	al	N	bd	e	То	ol		:
	-	:	:	:	:		:	•	:	:	:	•	:	:	:	 :	:	:	:	:	 		:	:	:	:	:		 :	:	:		:	•	:	:		:	:	•		•	•			•	:		•			•	:

15. Using the Orthogonal Node tool, connect the gates and pins as follows:



- 16. The final step is to assign the proper names to the pins. Double-click on the name pin\_name1 to edit the name. Rename to switch1.
- 17. Rename pin\_name2 to switch2.
- 18. Rename pin\_name3 to switch3.
- 19. Rename pin\_name4 to LEDD2.
- 20. Rename pin\_name5 to LEDD3.

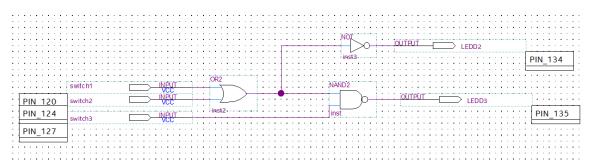


- 21. Save the file as SwitchLogic.bdf.
- 22. Now, we need to assign the pins to the input and output pins in the schematic. In the Task pane, right-click on Analysis & Synthesis. This will check the design and generate the pin hierarchy.
- 23. Correct any errors if they appear, but the process should return successfully.
- 24. From the menu, select Assignments->Pin Planner.
- 25. All the pins will be listed. Using the board schematic, set all the node names with the proper pin locations and change I/O Standard for all pins to 3.3V-LVTTL.

Node Name	Location
LEDD2	PIN_134
LEDD3	PIN_135
Switch1	PIN_120
Switch2	PIN_124
Switch3	PIN_127

Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	Current Strength	Slew Rate
LEDD2	Output	PIN_134	8	B8_N0	3.3-V LVTTL		8mA (default)	2 (default)
LEDD3	Output	PIN_135	8	B8_N0	3.3-V LVTTL		8mA (default)	2 (default)
switch1	Input	PIN_120	8	B8_N0	3.3-V LVTTL		8mA (default)	
switch2	Input	PIN_124	8	B8_N0	3.3-V LVTTL		8mA (default)	
in switch3	Input	PIN_127	8	B8_N0	3.3-V LVTTL		8mA (default)	
< <new node="">&gt;</new>								

26. Close the Pin Planner when finished. The schematic will now show the pin assignment names.



- 27. Save the project.
- 28. There is no clock, so a timing constraint file is not required. In the Task pane, right-click on Compile Design and select start from the context menu; or you can click on the Play symbol in the toolbar. The process should be successful.

#### 1.3.3 Program the Board

With the design compiled, we can now test the design on the board.

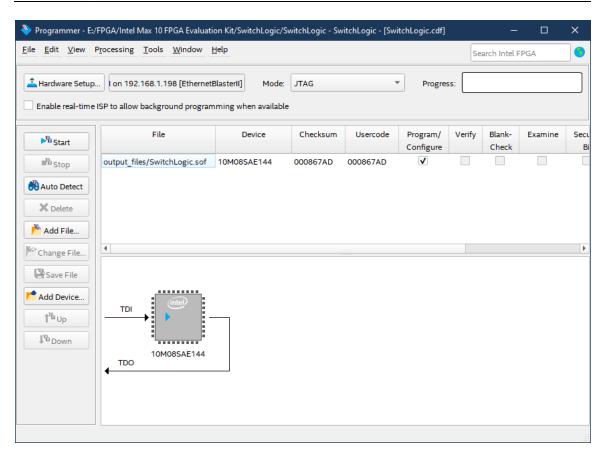
- 1. Connect the board to the programming cable per the cable instructions.
- 2. Power on the board and the programming cable box.
- 3. In Quartus Prime, from the Task pane, right-click on Program Device (Open Programmer) and select Open from the context menu.
- 4. The Programmer dialog appears. Click on the "Hardware Setup" button.
- 5. Click the Add hardware button, select the Hardware type and file in any remaining information, and click OK.

Hardware Settings	JTAG S	ettings		
Select a programmin hardware setup appli	-			es. This programming
	-		annier window.	
Currently selected ha	rdware:	No Hardware		
Hardware from and the second s	e		×	
Hardware type:	Ethern	etBlaster	•	Add Hardware
Port:			*	Remove Hardware
Baud rate:			Ŧ	
Server name:			•	
Server port:	1309			
Server password	:			
Auto Detect		ок	Cancel	

6. The tool allows you to connect to a number of programming cables. We need to select the one for our board. In the "Currently selected hardware", click the drop-down and select the hardware cable for the board, and click Close when finished.

Hardware Settings JTAG S	ettings			
Select a programming hardwa hardware setup applies only to				es. This programming
Currently selected hardware:	Etherne	tBlasterII on 19	92.168.1.198 [Eth	ernetBlasterII] 🔹
Hardware frequency:				н
Available hardware items				
Hardware		Server	Port	Add Hardware
EthernetBlasterII		192.168.1	EthernetBl	Remove Hardware

7. A SwitchLogic.sof file gets created during the Compile Design flow. The file is automatically filled in. There is only one FPGA on the board and in the JTAG chain so the file already has the Program/Configure checkbox checked. Click the Start button.



The process takes a few seconds and shows that the task completed successfully. The board will be running the design. Flip the switches on and off to make sure the LEDs light up per the truth table.

### 1.4 Summary: Getting Started

Quartus Prime is a powerful FPGA development tool. The deeper details of the tools' features and compilation steps can be found online at Intel.com The examples in this article provide an update to the Intel Quartus Prime Pro Edition User Guide: Getting Started for the latest Quartus Prime release.

### 1.5 References

The following references were used for this article:

- Electronics Verilog Blinking a LED <u>https://www.badprog.com/electronics-verilog-blinking-a-led</u>
- How to Begin a Simple FPGA Design intel.com Intel FPGA training site Intel® FPGA
   Technical Training / https://www.youtube.com/watch?v=bwoyQ RnaiA
- MAX 10 FPGA (10M08S, 144-EQFP) Evaluation Kit User Guide, Intel (Altera), 2015

Intel, Quartus, and MAX 10 are registered trademarks of Intel Corporation

All other copyrighted, registered, and trademarked material remains the property of the respective owners.