

# Introduction to the EPM240 Board

Muhammad Mun'im Ahmad Zabidi, Musa Mohd Mokji, Izam Kamisian, Norhafizah Ramli

#### Abstract

This article introduces the EPM240 board for low-cost digital experimentation.

<sup>1</sup> Department of Electronic and Computer Engineering, School of Electrical Engineering, Faculty of Engineering \*Corresponding author: munim@utm.my

# 1. FPGA vs CPLD

**Field-Programmable Devices** (FPDs) are devices that are programmed by electronically connecting and disconnecting internal logic elements. This differs from microcontroller and microprocessor which are programmed by changing the list of instructions. This enable FPDs perform tasks much faster while consuming about 10x less power.

There are several types of PLDs, those can be classified under SPLD, CPLD and FPGA. The SPLD and CPLD have the same basic design but differ due the keyword 'simple' and 'complex'. SPLD has only one logic element (LE) consisting of an array of AND, OR and NOT logic gates. The simplest CPLD from Altera has 240 LEs, and the next device has 570 LEs. That makes CPLD much more powerful than SPLD.

FPGAs have different internal architecture and working mechanism compared to CPLD. Whereas CPLD a relatively small number of complex LEs, the FPGA has a huge number of simple Configurable Logic Blocks (CLBs). This allows high end FPGAs to contain millions of equivalent gates compared to just thousands for the most complex CPLD.

Despite their differences, CPLDs and FPGAs are programmed using the same design tools. For devices made by Altera, the Quartus software is used for both low-end MAX CPLD, the high end Stratix FPGA, and all devices in between. Therefore, all knowledge you gain while using the MAX will be transferable when you upgrade to more sophisticated FPGA later.

## 2. The EPM240 aka MAX II

The EPM240 mini board is ideal for low budget digital hardware experimentation. In the middle sits the Altera MAX II EPM240T100C5N CPLD. The board has all the necessary auxiliary circuits allow the CPLD to be programmed through Quartus II. The schematics diagram of this board is available in open sources.

The board has 100 general purpose input/output (GPIO) pins accessible through the two (2) units of 22  $\times$  2 headers



Figure 2. EPM240 board bottom view.

as shown Fig. 1 and Fig. 2. For experiments, a breadboard is connected to these pins.

Features:

- JTAG connector
- Can be easily programmed using the USB Blaster
- 3.3 V operating voltage by using the AMS1117 3.3V voltage regulator circuit
- Flexible I/O interface supporting 3.3 V, 2.5 V, 1.8 V, and 1.5 V logic levels
- · High speed operation, up to 150 MHz
- InstantOn, non-volatile storage
- 240 logic elements
- On board power regulator (works with 5 V power adapters)
- On-board 50 MHz oscillator

## 3. Installing the Driver

The EPM240 board requires a USB Blaster for programming. The computer will recognize the new hardware connected to its USB port, but it will be unable to proceed if it does not have the required driver already installed.



Figure 3. USB blaster.

If the USB Blaster driver is not already installed, launch Device Manager.

4	۵	*	Filters $\lor$
Best n	natch		
1	Devic Contro	e Manager Il panel	×

Figure 4. Launch Device Manager.

In Device Manager, find the new hardware under Other devices.

~	Other devices
	📓 USB-Blaster(Altera)

Figure 5. Device Manager -> Other devices.

The driver available within the Quartus II software. Select *Browse my computer for driver software.* 



Figure 6. The driver is not online.

Select [Update Driver] from the dialog box that pops up.

USB-Blas	ter(Alter	ra) Prope	erties	$\times$			
General	Driver	Details	Events				
2	USB-B	laster(Alte	ara)				
	Driver	Provider:	Unknown				
	Driver I	Date:	Not available				
	Driver	Version:	Not available				
	Digital	Signer:	Not digitally signed				
Driv	ver Detai	ls	View details about the installed driver files.				
Up	date Driv	er	Update the driver for this device.				
Roll	Roll Back Driver		If the device fails after updating the driver, roll back to the previously installed driver.				
Disa	able Devi	œ	Disable the device.				
Unin	stall Dev	ice	Uninstall the device from the system (Advanced	).			
			OK Cance	I			

Figure 7. Update the driver.

Specify the driver location in the following dialog box:

C:\altera\ version \quartus \

Replace altera with IntelFPGA for some of the newer Quartus versions.

Browse for drivers on your compu	Browse For Folder X
Search for drivers in this location:	Select the folder that contains drivers for your hardware.
C:\altera\13.0\quartus	a li Mindaug (2)
☑ Include subfolders	Vindows (C)
→ Let me pick from a list of avait This list will show available drivers com	<ul> <li>ip</li> <li>logs</li> <li>modelsim_ase</li> <li>nios2eds</li> <li>quartus</li> </ul>
category as the device.	Folder: quartus

Figure 8. Specify the driver location.

# The driver will now be installed as indicated in Figure 9. You can now use the EPM240 board.



Figure 9. The driver is installed.

Next, we check out the software and hardware installation whether they are working properly. To do this, we will create a project, enter a short Verilog code, compile it, then download the code to the CPLD.

# 4. Project Settings

Run Quartus-II Web Edition and select the **File → New Project Wizard** menu.

#### Page 1

This page sets the location of your project and the name of your project. It is highly recommended to change the directory outside the Quartus software directory. This will cause less problems later if you have to reinstall Quartus. For example, you can create a folder in your home directory called *Documents/QuartusProjects*. Inside this folder, create your project folder for example *Documents/QuartusProjects/blinker*.

C: (quartusprojects)blinker What is the name of this project? blinker 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. Use Existing Project Settings	what is the working directory for this project?					
What is the name of this project?         blinker       -         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.         blinker       -         Use Existing Project Settings	C:\quartusprojects\blinker					
blinker   . 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. blinker   . Use Existing Project Settings	What is the name of this project?					
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.         blinker       .         Use Existing Project Settings	blinker					
Use Existing Project Settings	What is the name of the top-level design entity name in the design file.	for this project	? This name is ca	ise sensitive and	must exactly match	the entity
Use Existing Project Settings	blinker					

After setting the directory, project name and top-level entity, click Next >.

#### Page 2

This page lets you add files from another project.

ile name:						Add
File Name	Type Libr	ary Design Entry/Sy	nthesis Tool	HDL Version	[	Add All
					1	Remove
						Up
						Down
						Properties

Since this is your first project, there is nothing to add. Just click  $\boxed{\text{Next} >}$ .

### Page 3

This page selects the device we are using.

Family: MAX II Devices: All	•		Show in 'Available devices' list				
Devices: All		Package: Any					
Devices. All	_	rackaye.	Any				
		Pin count: Any					
Target device		Speed grade: Any Name filter:					
Auto device celected by the	Eittor						
vailable devices:	Core Voltage	LEs	UFM blocks	^			
vailable devices: Name EPM240T100C5	Core Voltage 3.3V	LEs 240 1	UFM blocks	^			
vailable devices: Name PM240T100C5 PM240T100I5	Core Voltage 3.3V 3.3V	LEs 240 1 240 1	UFM blocks	^			

#### Page 5

This page summarizes all settings.

,,		
Project directory:	C:\quartusprojects\blinker	
Project name:	blinker	
Top-level design entity:	blinker	
Number of files added:	0	
Number of user libraries added:	0	
Device assignments:		
Family name:	MAX II	
Device:	EPM240T100C5	
EDA tools:		
Design entry/synthesis:	<none> (<none>)</none></none>	
Simulation:	ModelSim-Altera (Verilog HDL)	
Timing analysis:	0	
Operating conditions:		
VCCINT voltage:	3.3V	
Junction temperature range:	0-85 °C	

Verify the device assignment, then click [Finish].

- For device family, select MAX II.
- To narrow down the device list, select Pin count **100**. This will shorten the list so that you can find the exact device faster. Or you can simply scroll through the long list.
- In the device list, select **EPM240T100C5**.

When you have selected the correct device, click Finish. Optionally, you can click Next > to go to Page 4.

### Page 4 (optional)

This page sets the EDA Settings.

Tool Type	Tool Name	Format(s)	Run Tool Automatically
Design Entry/Synt	<none> •</none>	<none> ~</none>	Run this tool automatically to synthesize the current design
Simulation	ModelSim-Altera 🔹	Verilog HDL 🔹	Run gate-level simulation automatically after compilation
ormal Verification	<none> -</none>	VHDL	
Board-Level	Timing	SystemVerilog HDL	
	Symbol	<none> -</none>	
	Signal Integrity	<none> -</none>	
	Boundary Scan	<none> -</none>	

Simply make sure the check box is OFF then click Next >.

## 5. Blinker Source Code

From the top menu, choose File  $\rightarrowtail$  New. In the pop-up dialog box, select Verilog HDL File:



Then key in this code as is (case sensitive):



Save it as **blinker.v** then at the Project Navigator set the code as Top level entity.

Project Naviga	ator		₽ <i>₽</i> ×	د 🔶
Files	v			📑 🕅
Dinike		<u>O</u> pen <u>R</u> emove File from Project		
	R	Set as Top-Level <u>E</u> ntity	2	Ctrl+Shift+J

Click on the **Processing** Start Compilation menu. If you found errors, correct the mistake and recompile. You can ignore warnings.



### 6. Pin Assignment

From the top menu, choose **Assignments**  $\Rightarrow$  **Pin Planner** menu:

Ass	ignments	Processing	Tools	Window	Help	P
¥ ∡	Device Settings.				Ctrl+Shi	ft+E
	TimeQue	est Timing An	alyzer <u>V</u>	izard		
ø	Assignm	ent Editor	Ctrl+Shift+A			

In the dialog box, assign led to Pin 74 and clkin to Pin 64.

Eile	<u>E</u> dit ⊻iew	Processing	Tools Window	Help 🛛	P		Search altera	a.com 🔇
9	Report Report not	₽ ₽ × available		100 99 98			™ 78 77 ™ V0000	
-			1	8			75	
			4	000	Тор \	liew	0 72	
0			1	00	Wire E	Bond	0 **	
X				¥.				
5	<	>	5					
0	Gro	Rep	1	000	MAX	C II	× ***	
=\;;=	Tasks Rui	n Ana 🔨	1	000	EDM240	T100CE	000	
1 <u>0</u> 21	✓ 🦢 Ear	ly Pir Early	2 2 2		EPM240	110005		
6.0	-	Run : Expo	2	:8	000000000000000000000000000000000000000	00000000	0000	
P	<ul> <li>Chi</li> <li>Chi</li> </ul>	ange Y		26 27 28	29 30 31 32 33 34 35 36 37 38	39 40 41 42 43 44 45 46	47 48 49 50	
2	× Named:	· ~ *	» Edit: 📈 🛷	PIN_74			Filter: Pins: a	
¥6	4 No	de Name	Direction	Locat	tion I/O Bank	Fitter Location	I/O Standard	Reserver ^
~	e ed		Input	PIN_64 PIN_74	2	PIN_64 PIN_74	3.3-V Lefault)	
<b>~</b>		1.1.1	ouput	1.4.0	3	120271	SIS V Eliteradicy	×
*							0%	00:00:00

After setting the pins, you must recompile to generate the FPGA programming file. The design is ready to be down-loaded into the CPLD.

# 7. Wiring

Connect pin 74 on EPM240 board to a resistor. Use a resistor with a value anywhere from 100  $\Omega$  to 470  $\Omega$ . Connect the resistor to the positive terminal of an LED. Connect ground pin to the ground of the LED board.



# Connect USB Blaster to PC and EPM240 board. Connect power.



## 8. Programming

From the top menu, select **Tools → Programmer**.

Tools	<u>W</u> indow <u>H</u> elp								
	Run Simulation Tool								
<u>2</u> .	Launch Simulation Library Compiler								
15.	Launch Design Space Explorer								
Ŏ	TimeQuest Timing Analyzer								
	<u>A</u> dvisors								
4	Chip Planner								
٠	Design Partition Planner								
	Netlist Viewers								
n,	SignalTap II Logic Analyzer								
	In-System Memory Content Editor								
<b>5</b>	Logic Analyzer Interface Editor								
01	In-System Sources and Probes Editor								
	SignalProbe Pins								
۵	Programmer								

#### Hardware Setup

Do this step once to setup USB Blaster for programming.

Ardware Setup.... No Hardware

Select Hardware Setup...

In the dialog box that follows, select USB Blaster.

Ľ	Hardware Setup							×
	Hardware Settings Select a programmin hardware setup appli	JTAG Se g hardware es only to f	ettings e setup t the curre	to use whe	n programmi mmer windov	ing devices. 1	This programming	
Currently selected hardware: USB-Blaster [USB-0] Available hardware items							•	
	Hardware USB-Blaster		Lo	erver ical	Port USB-0		Add Hardware	
							Close	

#### **Proceed to Programming**

Click Add File...

Add File...

Navigate to **output\_files** directory under your project. Select the POF file.

Select Pro	ogramming File					×
Look in:	C:\Users\mun3im\Documents\Quartus\knight\output_files	•	00	0		≣
My Cor	nputer			5		
File name:	knight.pof				Open	
Files of type:	Programming Files (*.sof *.pof *.jam *.jbc *.ekp *.jic)			•	Cance	4

#### Check the Program/Configure buttons.

Hardware Setup USB-Blaster [USB-0] Mode: JTAG Progress:     Enable real-time ISP to allow background programming (for MAX II and MAX V devices)      File Device Checksum Usercode Program/ Verify Configure Verify Configu	<u>File E</u> dit <u>V</u> iew Pro	ocessing <u>T</u> ools <u>W</u> indow	Help 🛡		Se	arch altera.com	3
Image: File     Device     Checksum     Usercode     Program/     Verify       Image: File     CFM     OU18811D     OU18811D     OU188105     Image: File       Image: File     Image: File     Image: File     Image: File     Image: File       Image: File     Image: File     Image: File     Image: File       Image: File     Image: File     Image: File     Image: File       Image: File     Image: File     Image: File     Image: File       Image: File     Image: File     Image: File     Image: File       Image: File     Image: File     Image: File     Image: File       Image: File     Image: File     Image: File     Image: File       Image: File     Image: File     Image: File     Image: File       Image: File     Image: File     Image: File     Image: File       Image: File     Image: File     Image: File     Image: File       Image: File     Image: File     Image: File     Image: File       Image: File     Image: File     Image: File     Image: File       Image: File     Image: File     Image: File     Image: File       Image: File     Image: File     Image: File     Image: File       Image: File     Image: File     Image: File       Image: Fil	Lardware Setup	USB-Blaster [USB-0] to allow background progra	Mode: JTAG	d MAX V devices)	▼ Progre	255:	
	🔊 Start	File	Device	Checksum	Usercode	Program/ Configure	Verify
∑ <sup>2</sup> Change Fie ∑ <sup>2</sup> Add Device <sup>2</sup> <sup>1</sup> <sup>1</sup> <sup>1</sup> <sup>1</sup> <sup>1</sup> <sup>1</sup> <sup>1</sup>	Auto Delete	output_files/knight.pof CFM UFM	EPM240T100	0018811D	0018ADA5	R	
	Change Hie	<					>
	Add Device						^
	גוי∯ Down		100				v

Then click Start to program the chip.

# Start

If everything goes well, the LED should blink once about every second.

## **Acknowledgments**

Thanks to Siti Nursyuhada binti Mahsahirun and Zulkifli Md. Yusof, both of Faculty of Manufacturing, Universiti Malaysia Pahang for the original idea.

### References

- [1] Munim Zabidi, Izam Kamisian, and Ismahani Ismail. *The Art of Digital Design.* 2019.
- [2] MAX II Device Handbook. Altera, Oct. 2008. URL: https: //www.intel.com/content/dam/www/programmable/ us/en/pdfs/literature/hb/max2/max2\_mii5v1.pdf.
- [3] Introduction to the Quartus® II Software. Version 10. Altera. 2010. URL: https://www.intel.com/content/ dam/www/programmable/us/en/pdfs/literature/ manual/intro\_to\_quartus2.pdf.
- [4] Intel FPGA USB Download Cable User Guide. Intel. Oct. 2016. URL: https://www.intel.com/content/dam/ www/programmable/us/en/pdfs/literature/ug/ug\_ usb\_blstr.pdf.
- [5] My First FPGA Design Tutorial. Altera. July 2008. URL: https://www.intel.com/content/dam/www/programmable/ us/en/pdfs/literature/tt/tt\_my\_first\_fpga.pdf.