User Manual
for
Krypton v1.1

Saurabh Agrawal
Abhishek Kamath
Kaushlesh Sharma
Debapratim Ghosh

Reconfigurable Hardware Design Team
Virtual Labs (e-Prayog) Wadhwani Electronics Lab
Indian Institute of Technology Bombay

September 2012

1 Board Synopsis

Krypton v1.1 (hereafter referred to as Krypton) is a 5M1270ZT144C5N CPLD-based board developed at e-Prayog, WEL Lab, IIT Bombay under the Virtual Labs Project by the Ministry of Human Resource and Development (MHRD), Government of India. 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.

2 Prerequisites for using Krypton

Before using Krypton, you are expected to have a basic familiarity/ working knowledge of:

- Basic concepts of digital logic and digital systems- logic circuits, combinational and sequential logic, and their applications in systems design
- Familiarity with any Hardware Description Language (HDL), such as VHDL or Verilog
- Not mandatory, but familiarity with various kinds of Programmable Logic Devices (PLDs)

On the e-Prayog webpage and in the support disk, there are introductory study material available on programmable logic devices, digital electronics, as well as introduction to hardware description languages.

3 Precautions for using Krypton

- Do not touch the pins of any onboard IC directly, to avert the risk of damage by electrostatic discharge.
- When interfacing with large number of external peripherals, power the board through a single $+5V$ DC supply (tested).
- The voltage at any I/O pin should not exceed $3.3V$.

4 Contents in the Support Disk/Webpage

- Krypton user manual (obviously!)
- Example applications with manual (labsheets)
- Driver files for the host PC (CMD20817)
- UrJTAG software for programming the CPLD
- Installer executable for Altera Quartus II v12.0
- Video tutorials for Krypton and using Quartus II and UrJTAG

5 Features and Specifications for Krypton

Figure 1 shows a top view of Krypton with detailed labels.

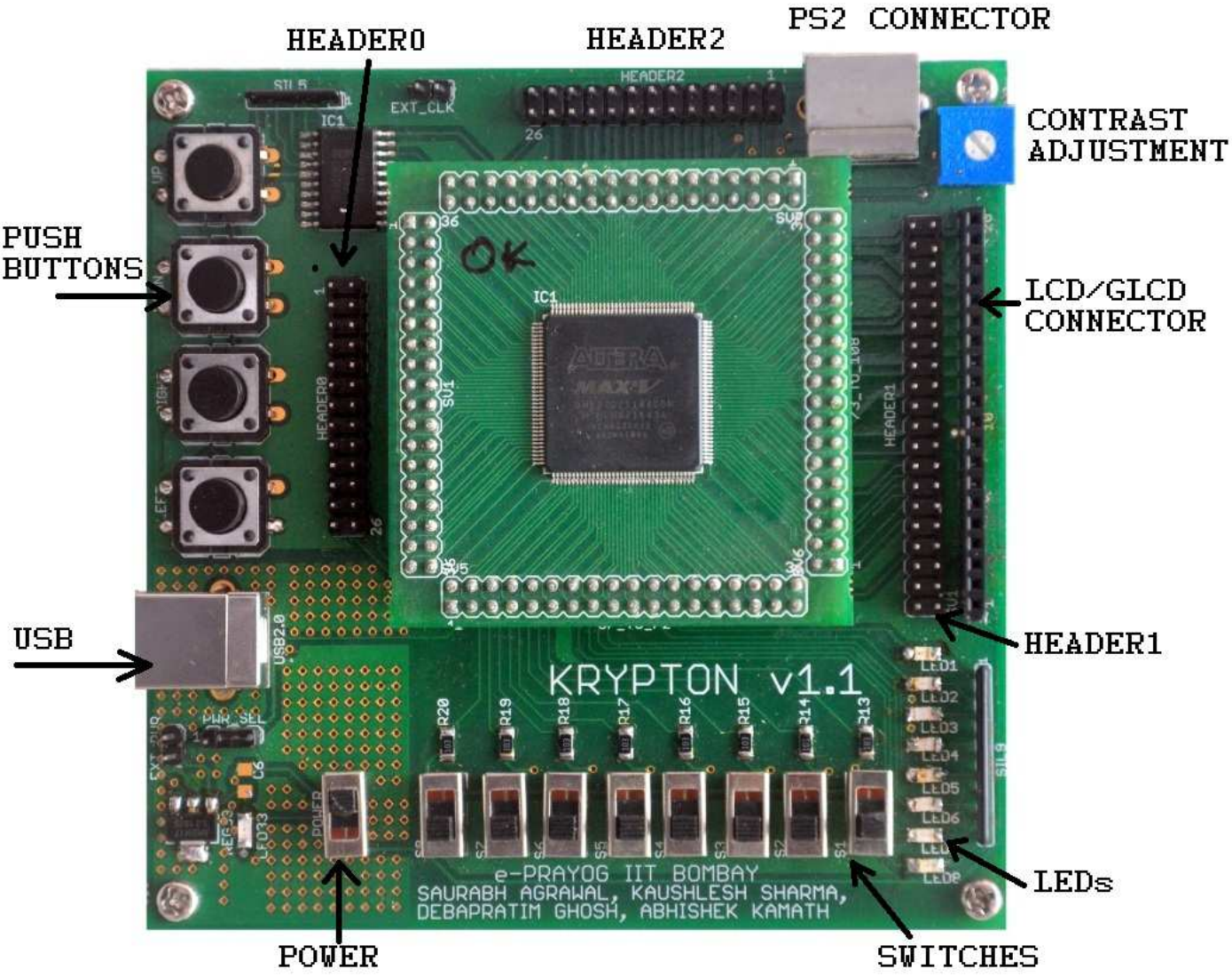


Figure 1: Krypton top-view with labeled features

Following are the features on Krypton.

- A board centered around 5M1270ZT144C5N, a CPLD of Altera's MAX V family (980 macrocells/1270 logic elements)
- USB programmable and powered, with provision for external DC supply
- Preconfigured on-board I/Os- 8 switches, 8 LEDs and 4 push-buttons with hardware debounce
- On-board clock of $1Hz$ and $50MHz$, and provision for external clock source connection
- Connectors provided on-board to interface standard peripherals directly (LCD/GLCD/PS2)
- Large number of on-board I/Os (86) provided for various applications
- Hot-socketing provided by detachability of the CPLD daughter card for compatibility with —— of the MAX II family.

6 Configuring Krypton Before Programming

You will need to ensure that an appropriate power source is selected for the CPLD while programming or running an application. To do this, an on-board jumper PWRSEL needs to be connected suitably.

`PWRSEL` Jumper Setting	Power Source for Krypton
1. Between pin 1 and center	Board powered through USB (Bus-powered via host PC)
2. Between pin 2 and center	Board powered through external power supply header `EXTPWR`

Table 1: Configuration for `PWRSEL` jumper

7 Getting Krypton working with your PC

1. Plug in the USB cable to the USB connector on the 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 should glow.
2. You will need to install the drivers for Krypton 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 XP:

3. Once you connect Krypton to your PC and power it on, a `Found New Hardware` window will open up. Select `No, not this time` and click `Next`. Now, select `Install from a specific list or location (Advanced)`, and click `Next`.
4. Now, click on the `Browse` button and provide the path to the folder CMD20817 (which is available in the support disk). Click `Next`.
5. This installation should now begin. The `Found New Hardware` may open up more than once (usually, not more than four times). In such cases, repeat steps 3-4.

6. At the end of the installation, you should see a desktop pop-up notification `Your device is now installed and ready to use`.

For Windows 7:

3. Once you connect Krypton to your PC and power it on, Windows 7 automatically searches for available drivers. If no drivers are available, a desktop notification showing `Device driver was NOT successfully installed` will pop up. In such case, right click `Computer` → `Properties` → `Device Manager`.

4. Inside `Device Manager`, under `Other devices` you will see 2 uninstalled devices named `RS232-HS`. 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 CMD20817 (which is available in the support disk). Click `Next`.

5. The installation should now begin. After this is complete, repeat step 4 for the second uninstalled device `RS232-HS`. Krypton is now ready to be used.

8 Using Quartus II 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. The IDE allows for code compilation, followed by generating a programming file for a specific target PLD (a CPLD or and FPGA). Quartus II has a free subscription/web edition that can be used. Suppose we wish to design and implement on Krypton, the simple logic circuit shown in Figure 2. This section provides a step-by-step method for doing the same.

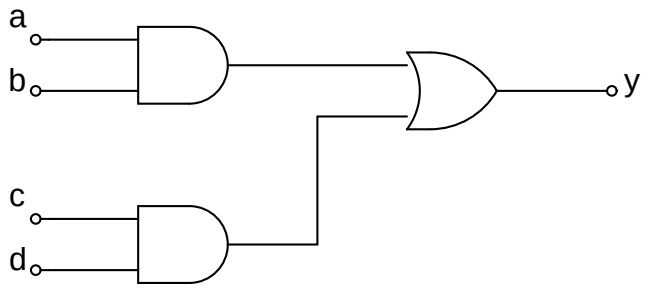


Figure 2: A simple logic circuit to be implemented on Krypton

1. Open Quartus II IDE. A pop-up will appear asking you to either create a new project or open an existing one. Click on `Create New Project`. 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. 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. 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`.

4. This opens up 'Page 2'. 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`.

5. On 'Page 3', 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 Krypton.

6. Click on the `Family` drop-down list and select `MAX V`. The device window shows a long list of devices available in the MAX V family.

7. On the right side of the window, you can filter out the device list by selecting 'Package' as `TQFP`, 'Pin Count' as `144` and 'Speed Grade' as `5`. You will see that the list is now quite short! Select the device `5M1270ZT144C5N` and click `Next`.

8. In the 'Page 4' window, you will be asked to select a simulation tool. Again, this is not mandatory and can be skipped. Click `Next`, and on 'Page 5' 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`. A text editor window will open, where you can write your HDL code. For our circuit, enter the following code.

```
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
end architecture struct;
```

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

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.

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.

12. You need to now assign the port pins of your design (`a, b, c, d, y`) to I/O pins on the CPLD 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.

13. Go to `Assignments` → `Pin Planner`. A new window opens up, showing you the schematic of the device selected (in this case, MAX V), and below, the signal lines that need to be pin-assigned. Follow the information in Table 2 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.

14. Once the pin assignment is complete, compile the design again (i.e. repeat Step 11).

Switch	CPLD Pin No.	LED	CPLD Pin No.
S1	48	LED1	58
S2	45	LED2	57
S3	44	LED3	55
S4	43	LED4	53
S5	42	LED5	52
S6	41	LED6	51
S7	40	LED7	50
S8	39	LED8	49

Table 2: Pin mapping for on-board Switches and LEDs

15. Now go to `Tools` → `Programmer`. A programmer window will open. You should see the project output file `myLogicCkt.pof` in this window. If not, you may need to create a fresh project from the start.

16. In the programmer window, go to `File` → `Generate (JAM, SVF...)`. A new window will open. Select the programming file type as `Serial Vector Format (SVF)`. Browse to the directory where you wish to store the `.svf` file. Preferably, save it directly in `C:\` drive. The filename need not be the same as the project name. Let us give the name `myLogic.svf`. Click on `Generate`. The programming file is now ready.

9 Using UrJTAG to program Krypton

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 Krypton. Assume that you have your programming file created from Quartus II, and saved in `C:\` as `myLogic.svf`. Follow the given instructions to program Krypton with this `.svf` file.

1. Copy the UrJTAG folder from the support disk to your PC desktop. Open the `jtag.exe` executable in the folder. A terminal window with a `jtag>` prompt opens up.

2. Give the command `cable ft2232` and hit enter. If the drivers have been installed, you should see a message `Connected to libftd2xx driver`.

3. Now, give the command `detect`, an hit enter. It should now show you the CPLD (target device) details as follows.

```
IR Length:  10
Chain Length:  1
Device ID: 0000001.............
Manufacturer:  Altera
.
.
.
.
.
.
jtag>
```

4. The steps 2-3 need to be followed every time (a) Krypton is connected to your PC and (b) a new `jtag.exe` window is opened.

5. The device can now be programmed. Give the command `svf c:\myLogic.svf`. The `jtag>` prompt will disappear for around 40 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.).

10 Using Krypton 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 ground (*GND*), which can be read by the CPLD. The pin mapping for these switches are in Table 3.

Push-button	CPLD Pin No.
UP	141
DOWN	142
LEFT	143
RIGHT	144

Table 3: Pin mapping for push-buttons

2. Clock sources for Krypton: Krypton has on-board $1Hz$ and $50MHz$ clock sources which can be used independently. Also, you may connect an external clock source to the `EXT_CLK` header. Refer Table 4 for the pin mapping.

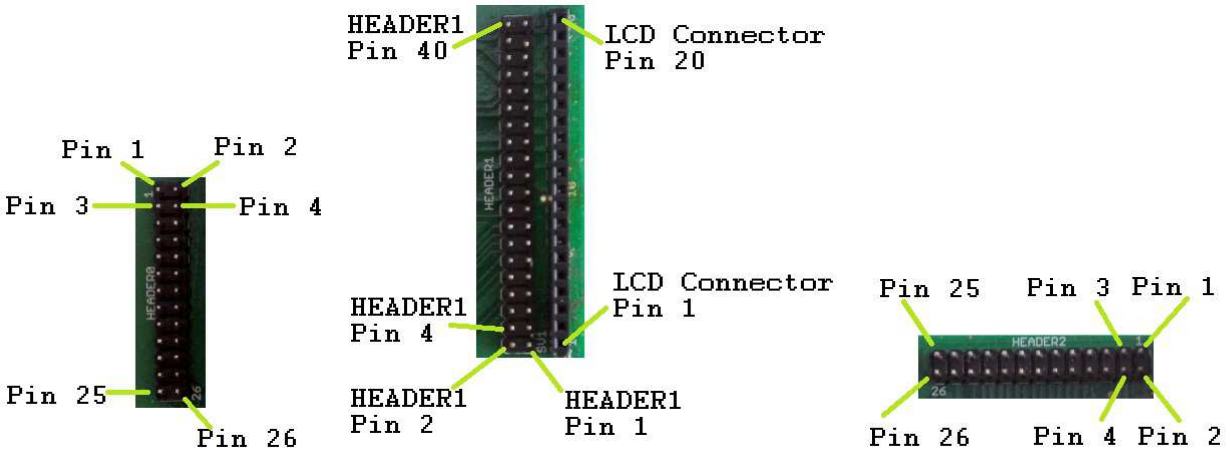
Clock Source	CPLD Pin No.
On-board $1Hz$	18
On-board $50MHz$	20
External clock	91

Table 4: Pin mapping for clock sources

3. Using the 26-pin `HEADER0`, 40-pin `HEADER1` and 26-pin `HEADER2`: Figure 3 shows the numbering convention for the general purpose I/O headers `HEADER0`, `HEADER1`, `HEADER2` and the LCD Connector. Tables 5-7 show the pin mapping for these using the same numbering convention and pattern as in Figure 3.

4. Using the LCD Connector: Krypton has a 20 pin on-board LCD connector that can be used to interface either a 16×2 character LCD module (most commonly, the JHD162A) or a Graphics LCD (GLCD) module. Table 8 shows the pin mapping for this connector.
Important: **1.** The pin 1 of the LCD/GLCD module should align with the pin 1 in the LCD connector. Wrong connections may damage the LCD module! **2.** Backlight facility is available only for GLCD and not for character LCD.

5. Using the PS2 Connector: The on-board PS2 connector may be used for direct interfacing to a PS2 device such as a keyboard or a mouse. PS2 has two communication lines- CLOCK and DATA. Table 9 shows the pin mapping for the PS2 connector.



(a) HEADER0 (b) HEADER1 and LCD Connector (c) HEADER2

Figure 3: Pin mapping for general purpose I/O headers

Odd No. Pin	Even No. Pin
(Pin 1) 1	2
3	4
5	6
7	12
13	14
15	16
21	22
23	24
27	28
29	30
31	32
37	38
GND	VDD ($5V$) (Pin 26)

Table 5: Pin mapping for HEADER0

11 Frequently Asked Questions (FAQs)

1. From where can I download Quartus II?

Ans: You can visit the following URL to download a free web edition:
`https://www.altera.com/download/software/quartus-ii-we`

2. Can I use UrJTAG directly from `www.urjtag.org`?

Ans: Yes you can, however, the distribution does not have support for MAX V devices. We at e-Prayog have created the necessary libraries. Simply use the UrJTAG provided on the e-Prayog webpage `http://59.181.142.81/`. 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 Krypton to function as a standalone unit. Is this possible?

Ans: Yes it is! You may use a standard power source such as a battery or a regulated DC power supply, after the programming is complete. Follow the power settings explained in Section 6. Take care not to use a supply of below $5V$.

Even No. Pin	Odd No. Pin
(Pin 40) VDD ($5V$)	GND
108	105
107	103
106	101
104	97
102	95
98	93
96	89
94	87
88	85
86	81
84	79
80	77
76	75
74	73
72	71
70	69
68	67
66	63
62	59 (Pin 1)

Table 6: Pin mapping for `HEADER1`

Odd No. Pin	GND	133	131	129	127	125	123	121	119	117	113	111	109 (Pin 1)
Even No. Pin	$5V$(Pin 26)	139	138	134	132	130	124	122	120	118	114	112	110

Table 7: Pin mapping for `HEADER2`

4. Suppose Krypton is currently configured to synthesize a particular logic function. Can this setting be erased somehow?

Ans: No. There is no direct concept of "erasing" a CPLD. You will notice that even if Krypton is powered off, it will run the last programmed configuration the next time it is powered on. This continues for the rest of its life, unless you program it for some other logic function.

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, in 'Page 4' as explained in Section 8. 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/software/modelsim-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 Krypton?

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 e-Prayog webpage. This problem will not occur.

Pin No.	CPLD Pin No.	GLCD Functional Pin
1	GND	GND
2	$+5V$	VDD
3	—	V0
4	70	RS
5	72	$R/\bar{W}$
6	74	EN
7	76	D0
8	80	D1
9	84	D2
10	86	D3
11	88	D4
12	94	D5
13	96	D6
14	98	D7
15	102	CS1
16	104	CS2
17	106	RESET
18	—	Vout
19	$+5V$	LED+
20	GND	LED-

Table 8: Pin mapping for LCD Connector

PS2 Pin Name	CPLD Pin No.
CLOCK	139
DATA	140

Table 9: Pin mapping for PS2 Connector

12 Help and Support

You are welcome to contact the Reconfigurable Hardware Design Team at e-Prayog, IIT Bombay to report software bugs, hardware failures/ other issues, and help and information in application developments. Suggestions regarding mistakes in the manual and other documentation/ distributables are welcome. Various queries and hate mail are also welcome.

Send an e-mail to `cpld.wel@gmail.com`

For administrative queries, e-mail to `vlabs_wel@ee.iitb.ac.in`