OTH REGENSBURG

Getting Started With *DE1-SoC* Board

Using VHDL

Prof. Dr. Martin J. W. Schubert, Electronics Laboratory,

OTH Regensburg, Regensburg, Germany

Getting Started With *DE1-SoC* Board

Using VHDL

Abstract. This communication teaches first step to operate Terasic's *DE1-SoC* board using an Intel *Cyclone V* FPGA.

1 Introduction

1.1 Objectives and Organization of this Document

This document is intended as introduction for students at OTH Regensburg that use the *DE1-SoC* board in some way.

The organization of this document is as follows:

Section **1** introduction,
Section **2** makes the user familiar with starting the *DE1-SoC* board at OTH Regensburg,
Section **3** gives some more detailed insight into the *DE1-SoC* board hardware,
Section **4** provides a first understanding of a VHDL code,
Section **5** demonstrates how to save a result as new project,
Section **6** summarizes the document and
Section **7** lists references.

1.2 Acknowledgements

The author would like to thank *Terasic Technologies* [1] for admission to use screen copies of *Terasic's* documentation for teaching purposes in this lectures.

```
At 19.09.2014 08:49, Terasic - Dong Liu wrote:
  Dear Martin,
  Thank you for using DE boards to teach VHDL. Yes, you can open all
  DE design resources for teaching purpose. Thank you!
  Best Regards,
  Doreen Liu
```

1.3 Tools

1.3.1 *DE1-SoC* Hardware

The course makes you familiar with the *Terasic's* [2] *DE1-SoC* board using an *Intel Cyclone V FPGA* [2]. Identify the version of your board [3].

What is the revision letter of your hardware board?

. . . . . .

Hint: Revisions *F* and *G* differ in a printed company label only. To get it from the internet, go to [4] to find and download *DE1-SoC_v.5.1.2_HWrevF_SystemCD.zip* [5] or a later version and download it. It contains amongst other things important documents such as DE1-SoC User Manual [6] and Schematic [7]. The latest available Version is *Rev .F*. On the computer system of OTH Regensburg you will also find the CD on drive K:\Sb\ [8]. Do not use any other manual revision to follow this documentation. The differences are sometimes considerable.

1.3.2 Quartus [9] and ModelSim [10] Software Tools

It is assumed that you have *Intel's ModelSim and Quartus* software available. To download this freeware for your private PC you have to sign in at *Intel* [11]. At *OTH Regensburg's* PC pools of faculty *EI* this software is installed.

1.3.3 Use of VHDL

The *IEEE standard VHDL Language reference manual* [12] is difficult to read. Qualis VHDL Quick reference Card [13] and 1164 Packages Quick reference Card [14] are compact but difficult to understand. Feel free to find your own sources.

VHDL is not case sensitive. In the following, **`KEYWORDS`** will be written in **`ALL CAPITAL LETTERS`** and user defined names in **`lowercase letters`**. Exception: Capitalized initials are used for composed self-made names, e.g. *AddressBus* or *DataBus*. Self-made data types begin with *t_*, e.g. *t_StateVector*.

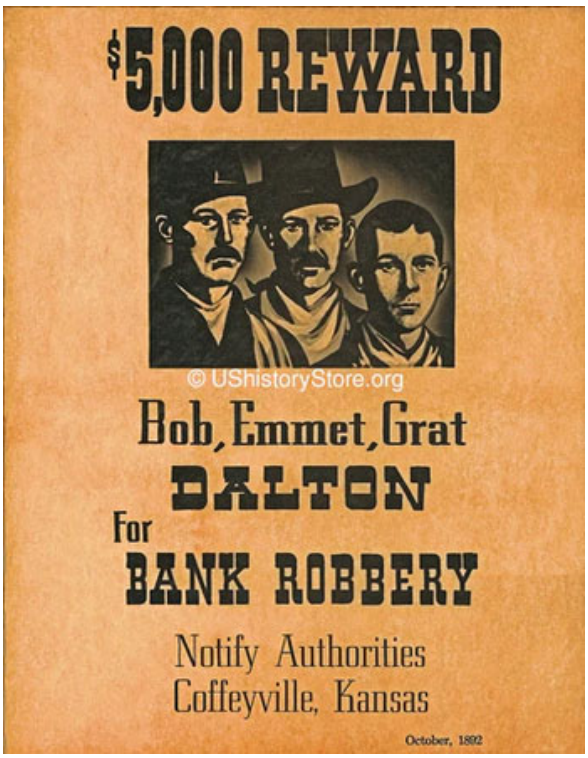
1.4 Wanted: *Bounty* Hunters!

Bounty hunters normally hunt people. Here, you can hunt a *Bounty*:

1 *Bounty* for reporting an error in the author's documentation that he is not yet aware of.

The bounty will be paid regardless of whether the incorrect script is presented torn or intact.

Notify Authorities
Regensburg, Bavaria



2 Getting Started

Goal of this subchapter is to get started with the *DE1-SoC* board and *Quartus* [6] software on the computer system of *OTH Regensburg*.

2.1 Getting Started with the *DE1-SoC* Hardware

Note for all the text below: It is *DE(one)-SoC* hardware, not *DEL-SoC*!

Requirements: have available

- Hardware: *DE1-SoC* board,
- Software: *Quartus* 18 or later installed on your computer,
- Documents from *DE1-SoC_v.5.1.2_HW_RevF_SystemCD.zip*: *UserManual\DE1-SoC_User_manual.pdf, Schematic\DE1-SoC.pdf.*

Take your *DE1-SoC* board, plug in the 12 V_{DC} power supply and connect the USB port with your computer. Press *Power ON/OFF* button. You should see some red light emitting diodes (LEDs) blinking and the 7-segment display counting hexadecimal numbers from 0 … F.

Note how we display hex-numbers 6: , 8: , B:
with 7 bars like on the 7-seg display . . . .

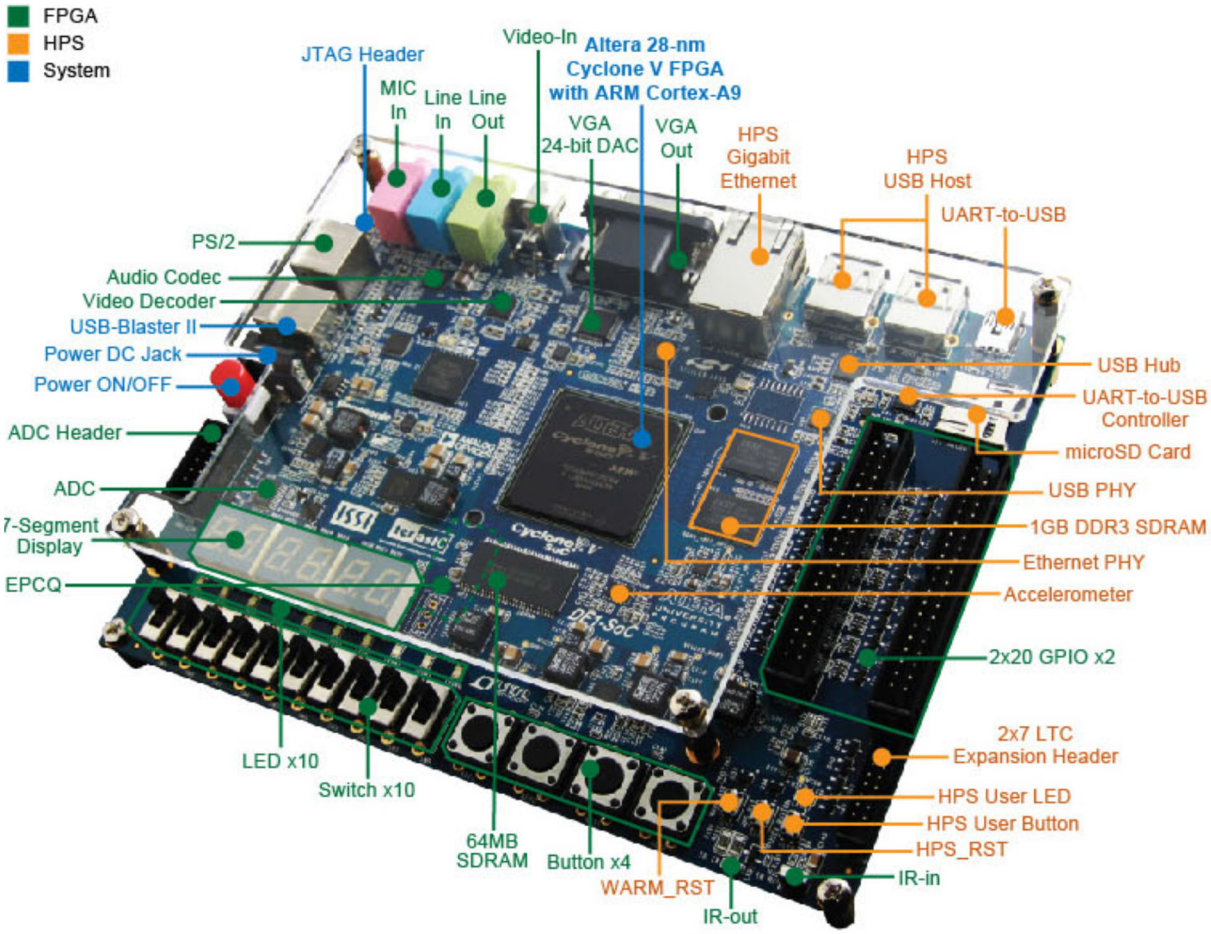


Fig. 2.1: *DE1-SoC* board (copied from *DE1-SoC Quick Start Guide*, document *DE1-Soc_QSG.pdf* in *DE1-SoC_v.5.1.2_HW_RevF_SystemCD.zip* [4]).

2.2 Getting Starting with the *Quartus* Software

Goal of this subchapter is to get started with *Quartus* software and create some VHDL code. In Electronics Lab of OTH Regensburg *Quartus* is available by desktop icon.

2.2.1 Create a *Quartus* Project

In the following board name "*de1soc*" is to be read as *de(one)soc* rather than *de(L)soc*.

- Create an empty directory on your *Windows* operating system, name it *de1soc_start*. Hereinafter it will be referred to as *...\de1soc_start*.
- Start *Quartus* revision 18 or later on your computer.
- Select menu point *Create a New Project (New PROJECT Wizard)* > Next >
- "*What is th*e w*orking directory of the project*?": Navigate to *...\de1soc_start*.
- "*What is th*e name *of the project*?": → *de1soc_start*.
- Top-level entity name: → *de1soc_start* (must match the project name).
- Click on → *Finish*.
- Exit *Quartus*.
- Look into your directory *de2_start*. There will be at least the two files
  + *de1soc_start.qpf*: Quartus Project File
  + *de1soc_start.qsf*: Quartus Specification File
  Use a text editor (e.g. *Notepad++*) to look into *de1soc_start.qpf* and *de1soc_start.qsf*. Note that there is no pin and device assignment in this file at this point.
- Delete all other files in directory *...\de1soc_start*. Then double-click left on file *de1soc_start.qpf*. *Quartus* should start with the respective working directory.

2.2.2 Create and Compile a VHDL Code within *Quartus*

Let us now create a first VHDL code. Close *Quartus*.

- Within working directory *...\de1soc_start* remove any other files than *de1soc_start.qpf* and *de1soc_start.qsf*.
- Double click left on *de1soc_start.qpf*. *Quartus* opens within the working directory *...\de1soc_start*.
- Select from *Quartus* menu > *File* > *New...* > *Design Files* > *VHDL File* > *OK*.
- Copy the VHDL code of listing 2.2.2 into the editor.
- *Quartus* menu > *File* > *Save As...* > *de1soc_start.vhd* > *OK*.
- *Quartus* menu > *Processing* > *Start Compilation*. (Note blue arrow short cut symbol.)

The VHDL code should compile without errors. Otherwise, click on the first red (error) line in the *Quartus* message window and fix the problem. Everything ok? → Close *Quartus*.

- Delete any other files than *de1soc_start.qpf*. ,*de1soc_start.qsf*, *de1soc_start.vhd* in directory *...\de1soc_start*.
- Double click left on *de1soc_start.qpf*.
- Within *Quartus Project Navigator* window double-click left on *de1soc_start*. You should see the VHDL code again.
- Compile it. Anything should work as before.

Question: Save the necessary files to rebuild the project. Which 3 file (extensions?) are indispensable?

. . . . . .

Listing 2.2.2: First VHDL code.

```
-- For Board: Altera Board DE1-SoC with FPGA Cyclone V 5CSEMA5F31C6N
LIBRARY ieee; USE ieee.std_logic_1164.ALL,ieee.std_logic_signed.ALL;
ENTITY de1soc_start IS
    PORT(CLOCK_50:IN std_logic;
      key:IN std_logic_vector(3 DOWNTO 0);      -- low when pressed
      sw:IN std_logic_vector(9 DOWNTO 0);       -- low when pulled down
      ledr:BUFFER std_logic_vector(9 DOWNTO 0); -- high active
      hex0,hex1,hex2,hex3,hex4,hex5:OUT std_logic_vector(0 TO 6);
      gpio_0:BUFFER std_logic_vector(35 DOWNTO 0);
      gpio_1:INOUT  std_logic_vector(35 DOWNTO 0)
    );
END ENTITY de1soc_start;

ARCHITECTURE rtl_de1soc_start OF de1soc_start IS
  TYPE t_seg7 IS ARRAY(0 TO 15) OF std_logic_vector(0 TO 6);
  CONSTANT c_seg7:t_seg7:=("1111110", "0110000", "1101101", "1111001",
    "0110011", "1011011", "1011111", "1110000", "1111111", "1110011",
    "1110111", "0011111", "1001110", "0111101", "1001111", "1000111");
BEGIN
  --
  -- read switches
  gpio_0(0) <= clock_50;
  gpio_1(9 DOWNTO 0) <= sw(9 DOWNTO 0);
  --
  -- control LEDs
  ledr <= sw;  -- same length -> == "ledr(9 DOWNTO 0) <= sw(9 DOWNTO 0);"
  --
  -- Control HEX-displays
  p_check_hex:PROCESS(sw(0))
  BEGIN
    IF sw(0)='0' THEN
      hex0<= c_seg7(0); hex1<= c_seg7(1); hex2<= c_seg7(2);
      hex3<= c_seg7(3); hex4<= c_seg7(4); hex5<= c_seg7(5);
    ELSE
      hex0<= c_seg7(6); hex1<= c_seg7(7); hex2<= c_seg7(8);
      hex3<= c_seg7(9); hex4<= c_seg7(10); hex5<= c_seg7(11);
    END IF;
  END PROCESS p_check_hex;
END ARCHITECTURE rtl_de1soc_start;
```

2.2.3 Download compiled code into the FPGA

After compilation

- Make sure DE1-SoC board is ON and connected to your PC via USB.
- Select from *Quartus* menu > *Tools* > *Programmer*. (Remember the short-cut symbol.)
- The Programmer window opens. In the upper left corner check for *Hardware Setup….* Most probably you see "*No Hardware*".
- Double-click left on *Hardware Setup….* The *Hardware Settings* window opens and you should be offered Hardware *DE-SoC*. Click on *Hardware: DE-SoC* until you see string *DE-SoC [USB-1]* in box *Currently Selected Hardware*.
- Then click *Close* (don't click on *Add Hardware*).
- In the programmer you should now see in box *Hardware Setup* string *DE-SoC [USB-1]*.
- Congratulations, you have successfully connected the hardware!
- Double-click left on *Start* and observe your FPGA. What happens? . . . . . .

Configuring the Software

Most probably you will see that nothing happens in the programming trial above, because a some important data is still missing. Check in the *Quartus Project Navigator* window for which *Cyclone V* FPGA your compilation and synthesis was done. Most probably not the *5CSEMA5F31C6N* device. We will correct and observe this now.

- Open file *de1soc_start.qsf* with an ASCII editor. Observe which device is specified here, e.g. *Family "Cyclone V", Device 5CGXFC7C7F23C8*. To correct it select:
- Quartus*: Assignments > Family: Cyclone V (E/GX/GT/SX/SE/ST)*, *Device: 5CSEMA5F31C6 > OK.*
- Compile the code with the new settings and program the FPGA again.
- Look into *de1soc_start.qsf* with an ASCII editor. You should now see the correct settings.
- Open the *Programmer* tool and press *Start* button. Result? Still no reaction of the FPGA!

Within the programmer window

- Press button *Auto Detect* and select device *5CSEMA5 > OK > Yes*. You will see graphically the devices *SOCVHPS* and *5CSEMA5*. The latter is specified too inaccurate.
- Click on the *5CSEMA5* symbol and delete it with the *Delete* button of your key board.
- Select *Add File… > output_files > de1soc_start.sof >OPEN*. You should now see graphically the devices *SOCVHPS* and *5CSEMA5F31*. This is correct.
- Click *Start* button while observing the *DE1-SoC* hardware. You should see some programming activity now at the red LEDs.
- After programming, the displays show meaningless information and the red LEDs do not react on push-button movements. This is because VHDL signals were not yet mapped correctly to physical pins.

Pin assignment. Get file *DE1_SoC_pin_assignments.csv* from the author's homepage [15].

- Look into file *DE1_SoC_pin_assignments.csv*: pin mappings.
- After programming the displays showed meaningless information and the red LEDs did not react on push-button movements. This is because VHDL signals were not mapped correctly to physical pins. Look on the footprint of the FPGA by selecting:
- *Quartus > Assignments > Pin Planner*. You will see the "*Top View – Wire Bond of the Cyclone V - 5CSEMA5F32C6* device. It is a 30x30 ball-grid array with columns numbered 1-30 and rows *A-AK*. To get the correct pin assignments:
- *Quartus > Assignments > Import Assignments… > DE1_SoC_pin_assignments.csv > Open > OK*. (You might need to navigate to that file.) To get these pin assignments active the code must be compiled and downloaded again:
- *Quartus > Processing > Start Compilation*, or click on compilation short cut.
- *Quartus > Tools > Programmer > Start*, or click on programmer short cut.

The 7-segment display seems to show meaningless information. Ignore it at this time. You should be able to switch the red LEDs on and off with the switches below them. Yes? Congratulations! You have correctly compiled and downloaded your VHDL code into the FPGA. Quit *Quartus* and look into the **.qpf* and **.qsf* files. You should see correct device and pin assignments now. Also look into file *DE1-SoC_pin_assignments.csv* to understand the source of the pin assignments. To which pin is switch *SW(8)* assigned?

. . . . . .

Repair the 7-segment displays:

- Look at the following code line in listing 2.2.2: **`IF sw(0) ='0' THEN`**…, According to this statement, the 7-segment digits should show their indes when *sw(0) = '0'*, i.e. digit 0 should show a 0, digit 1 a 1, …, digit 5 a 5. If *sw(0) = '1'*, then digit X should show X + 6, X=0…5.
- On the first glance, the numbers look different, but still meaningless. Look into the schematic [7] and search for string "7Segment" (p. 19). You will see the signal *HEXxy* driving the segments. Do they need a '0' oor a '1' potential to drive the respective segment on?

  . . . . . .
- What has to be obeayed to read the numbers X or X+6 now?

  . . . . . .
- Any of the statements (which end at a semi-colon ';' each), have to be changed according to `hex#<= c_seg7(#);` → `hex#<=`**`NOT`** `c_seg7(#);` whith # = 0…11.

To clean up the working directory …*de1soc_start*\. Delete all its files with exception to

1. *de1soc_start.vhd* // *VHDL* code, indispensable to rebuild the project
2. *de1soc_start.qpf* // *Quartus* project file , indispensable
3. *de1soc_start.qsf* // *Quartus* specification file , indispensable
4. *output_files\de1soc_start.cdf* // programmer configuration file (device etc.), useful
5. *output_files\de1soc_start.sof* // compiled binary code ready for download, useful

Open the *de1soc_start.cdf* file with an ASCII editor. It configures the programmer. To use it on other file systems replace the path specification *PATH("<absolute_pathname>")* with *PATH("./")*. Save it and preserve the **.cdf* file for correct device configuration of the programmer. (This notation stems from *Linux* and *Unix* operating systems, where path name "`./`" stands for "this directory" and "`../`" stands for "parent directory of this directory")

Open the *de1soc_start.sof* file with an ASCII editor. It is the binary information programming the device. You cannot read it but it can reprogram your device without new compilation. Consequently, preserve good **.sof* files.

With the first 3 files of the list above you can rebuild the whole design. With the last two file you can program the FPGA device without new compilation.

3 Understanding the User Headers

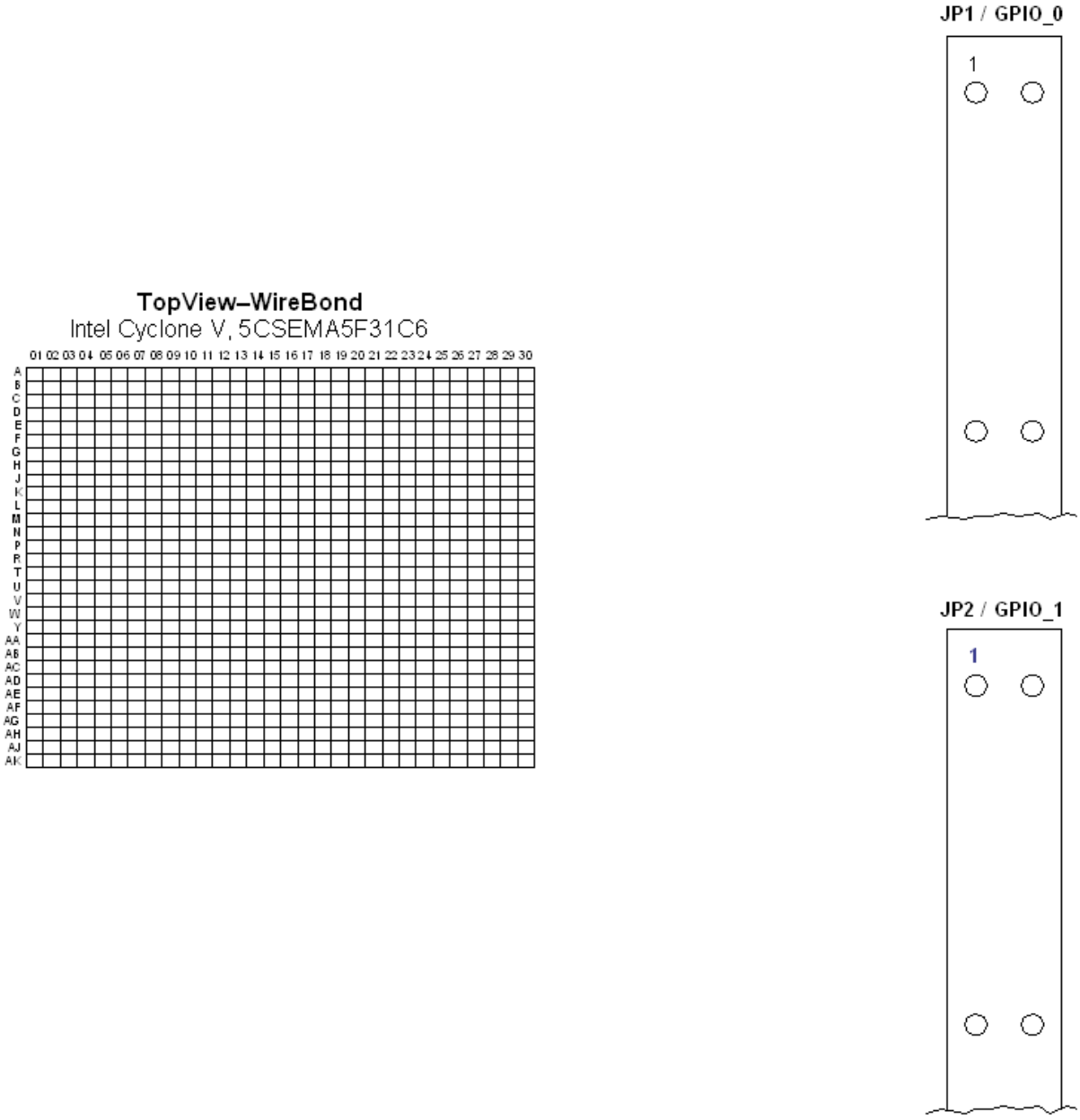


Figure 3: protection circuitry connecting FPGA and user header pins.

Fig. 3 illustrates the ball grid array (BGA) connecting the Cyclone V FPGA electrically to the printed circuit board (PCB). It is a top view of a pad array, which is under the FPGA. When solder balls between FPGA and printed circuit board (PCB) are heated during the production process, the FPGA is connected electrically and thermally to the PCB. The BGA consists of 30 columns numbered 1…30 and 30 rows labeled A…AK. You get the BGA top view after compilation with *Quartus* (e.g. of projects *de1soc_start*) selecting

Quartus > Assignments > Pin Planner (Ctrl + Shift + N) .

Which letters were omitted in the row labeling, and why?

. . . . . .

. . . . . .

Schematics: Note that wires with same labels are electrically connected!

Pin 1 of user header *JP1* is labeled in Fig. 3. Use *De1-SoC* schematic [7] to label also the other 3 pins of *JP1* in Fig. 3.

Pin 1 of user header *JP2* is labeled in Fig. 3. Use *De1-SoC* schematic [7] to label also the other 3 pins of *JP2* in Fig. 3.

Use the schematic to plot the circuit between *JP1's* pin 1 and *JP2's* pin 3 and the FPGA's ball grid array. Label any device, any pin, any wire according to the schematics. Resistors have a value. The diodes come as pairs with a single label.

Note at the wires also the pad IDs, e.g. *AG26*.

Note at the wired also the VHDL top-level entity signal name driving the cited pad, e.g. *GPIO_1(10)*. The mapping of VHDL top-level entity signal names are mapped to BGA pins
(i) in file DE1_SoC_pin_assigments.csv file, and
(ii) after assignment and compilation in the project's *Quartus* specification file (*.*qsf*).

BGA pad with which label is driven by VHDL signal *ledr(4)* ?

. . . . . .

BGA pad with which label is driven by toggle switch *sw(5)* ?

. . . . . .

BGA pad with which label is driven by push button *key(0)* ?

. . . . . .

Search all strings "LEDR3" in the Schematic [7]. Which Diode is driven by this wire and why is ON = high potential?

. . . . . .

Get a feeling for a 50 MHz rectangular clock signal on a pin

In the VHDL source code you will find the VHDL command line "*gpio_0(0) <= clock50;*". Which pin of the user headers is driven by the 50MHz clock signal?

. . . . . .

Use an oscilloscope to measure the 50MHz signal. Connect and disconnect the oscilloscope's ground to/from the ground pin of the respective user header. Move the wires and observe the modification on the oscilloscope. Conclusion: Measurement technique is insufficient.

4 First Understanding of the VHDL Code

You might have noticed that the 7-segment displays seem to show meaningless information.

Search all strings "*HEX50*" in the Schematic [7]. Which Diode element is driven by this wire and why is ON = low potential?

. . . . . .

. . . . . .

The “strange” 7-seg display looks a little bit strange. Why? Correct it! (Hint: The inversion of signal *sigx* is performed with *NOT sigx*, regardless if *sigx* is a scalar or an array.)

. . . . . .
.

. . . . . .
.

. . . . . .
.

Compile and download the design again. Do you see correct seg7-display now? You should! With *sw(0)* you can also change the number set. Check for the difference between **b** and **6**.

5 Copy Project *de1soc_start* to *de1soc_seg7*

Congratulations, you got it! We now want to preserve the results of this state as start point for further lessons with project name *de1soc_seg7.sof*. (The difference between projects *de1soc_start* and *de1soc_seg7* is the corrected 7-segment display driver by insertion of *NOT*.)

- Finish with the getting started session: Quit *Quartus*.
- Delete all files with exception to those with extension **.vhd, *.qpf, *.qsf, output/*.cdf.*
- Copy directory *de1soc_start* to directory *de1soc_seg7*.
- Within filename and file *de1soc_start.vhd* replace all: "*de1soc_start*" by "*de1soc_seg7*".

The official way:

Create a new *Quartus* project entitled *de1soc_seg7* as described in chapter 2 within the new *de1soc_seg7* director.

The quick & dirty way:

Within all filenames and files **.vhd, *.qpf, *.qsf, *.soc* replace all strings "*de1soc_start*" by "*de1soc_seg7*".
Compile and download the new project entitled "*de1soc_seg7*". Should work. Clean up your directories "*de1soc_start*" and "*de1soc_seg7*" to preserve necessary files only.

6 Conclusions

First steps with *DE1-SoC* boards enable the student to start the board. Chapter 3 deepens the understanding of the hardware, chapter 4 provides a first understanding of the VHDL code and chapter 5 demonstrates how to translate a result to a new project.

7 References

[1] Available: https://www.terasic.com.tw.
[2] Available: https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=836
[3] Available: http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=886
[4] Available: https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=165&No=836&PartNo=4
[5] CD ROM *DE1-SoC CD-ROM (rev. F Board) Version 5.1.2 of 2910-01-28* from [4
[6] *DE1*-SoC User Manual, Ref. F, taken from [5], available: https://hps.hs-regensburg.de/~scm39115/homepage/education/labs/Lab_ElectronicBoards/DE1-SoC_UserManual.pdf
[7] *DE1*-SoC Schematic, Ref. F, taken from [5], available: https://hps.hs-regensburg.de/~scm39115/homepage/education/labs/Lab_ElectronicBoards/DE1-SoC_Schematic_revF.pdf
[8] K:\SB\Sources\EDA\Terasic\Hardware\
[9] Available: https://en.wikipedia.org/wiki/Intel_Quartus_Prime
[10] Available: https://en.wikipedia.org/wiki/ModelSim
[11] Available: https://www.intel.com/content/www/us/en/programmable/downloads/download-center.html
[12] Available: https://edg.uchicago.edu/~tang/VHDLref.pdf
[13] Available: https://www.mimuw.edu.pl/~marpe/pul/card_vhdl.pdf
[14] Available: https://www.mimuw.edu.pl/~marpe/pul/card_1164.pdf
[15] Available: https://hps.hs-regensburg.de/scm39115/
[16] MAX4230 - MAX4324 data sheet, Maxm Integerated, Available: https://datasheets.maximintegrated.com/en/ds/MAX4230-MAX4234.pdf
[17] LT1711 / 1712 Rail-to-Rail Comparators, Analog Devices (Linear Technlogy) Available: https://www.analog.com/media/en/technical-documentation/data-sheets/171112f.pdf