

ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

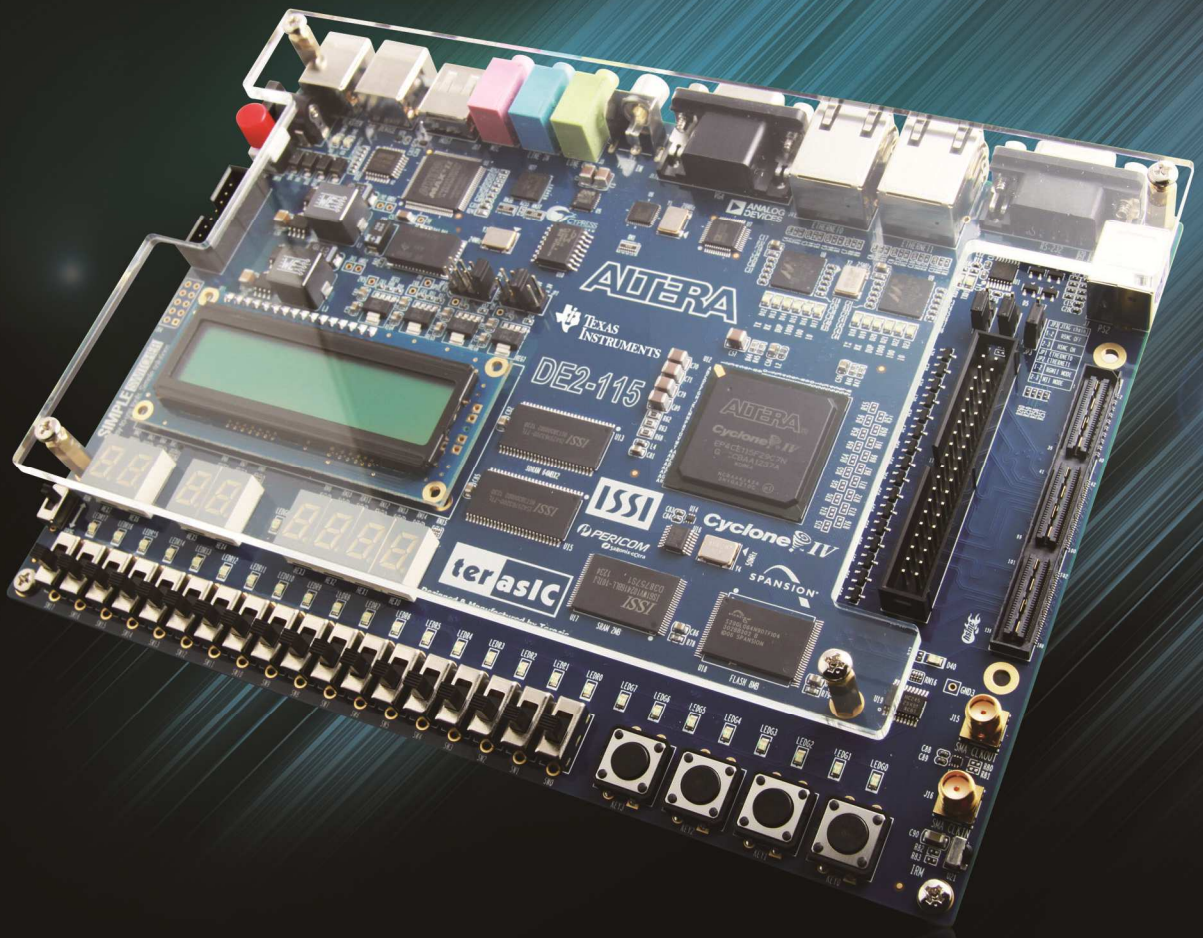
All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Visit our website - *Click HERE*

DE2-115

User Manual

World Leading FPGA Based Products and Design Services





terasIC
www.terasic.com

ALTERA

Copyright © 2003-2013 Terasic Technologies Inc. All Rights Reserved.

CONTENTS

CHAPTER 1 DE2-115 PACKAGE 4

1.1 Package Contents 4

1.2 The DE2-115 Board Assembly 5

1.3 Getting Help 6

CHAPTER 2 INTRODUCTION OF THE ALTERA DE2-115 BOARD 7

2.1 Layout and Components 7

2.2 Block Diagram of the DE2-115 Board 9

2.3 Power-up the DE2-115 Board 12

CHAPTER 3 DE2-115 CONTROL PANEL 14

3.1 Control Panel Setup 14

3.2 Controlling the LEDs, 7-segment Displays and LCD Display 16

3.3 Switches and Push-buttons 18

3.4 SDRAM/SRAM/EEPROM/Flash Controller and Program-mer 19

3.5 USB Monitoring 21

3.6 PS/2 Device 22

3.7 SD Card 23

3.8 RS-232 Communication 23

3.9 VGA 24

3.10 HSMC 25

3.11 IR Receiver 26

3.12 Overall Structure of the DE2-115 Control Panel 27

CHAPTER 4 USING THE DE2-115 BOARD 29

4.1 Configuring the Cyclone IV E FPGA 29

4.2 Using Push-buttons and Switches 32

4.3 Using LEDs 34

4.4 Using the 7-segment Displays 36

4.5 Clock Circuitry 38
4.6 Using the LCD Module 39
4.7 High Speed Mezzanine Card 40
4.8 Using the Expansion Header 46
4.9 Using 14-pin General Purpose I/O Connector 50
4.10 Using VGA 51
4.11 Using the 24-bit Audio CODEC 53
4.12 RS-232 Serial Port 54
4.13 PS/2 Serial Port 55
4.14 Gigabit Ethernet Transceiver 56
4.15 TV Decoder 59
4.16 Implementing a TV Encoder 61
4.17 Using the USB Interface 61
4.18 Using IR 62
4.19 Using SRAM/SDRAM/FLASH/EEPROM/SD Card 63
CHAPTER 5 DE2-115 SYSTEM BUILDER 71
5.1 Introduction 71
5.2 General Design Flow 71
5.3 Using DE2-115 System Builder 72
CHAPTER 6 EXAMPLES OF ADVANCED DEMONSTRATIONS 79
6.1 DE2-115 Factory Configuration 79
6.2 TV Box Demonstration 80
6.3 USB Paintbrush 82
6.4 USB Device 84
6.5 A Karaoke Machine 84
6.6 SD Card Demonstration 88
6.7 SD Card Music Player 91
6.8 PS/2 Mouse Demonstration 95
6.9 IR Receiver Demonstration 98
6.10 Music Synthesizer Demonstration 102
6.11 Audio Recording and Playing 105

6.12 Web Server Demonstration 108
CHAPTER 7 APPENDIX 119
7.1 EPCS Programming via nios-2-flash-programmer 119
7.2 Revision History 119
7.3 Copyright Statement 120

Chapter 1

DE2-115 Package

The DE2-115 package contains all components needed to use the DE2-115 board in conjunction with a computer that runs the Microsoft Windows OS.

1.1 Package Contents

Figure 1-1 shows a photograph of the DE2-115 package.

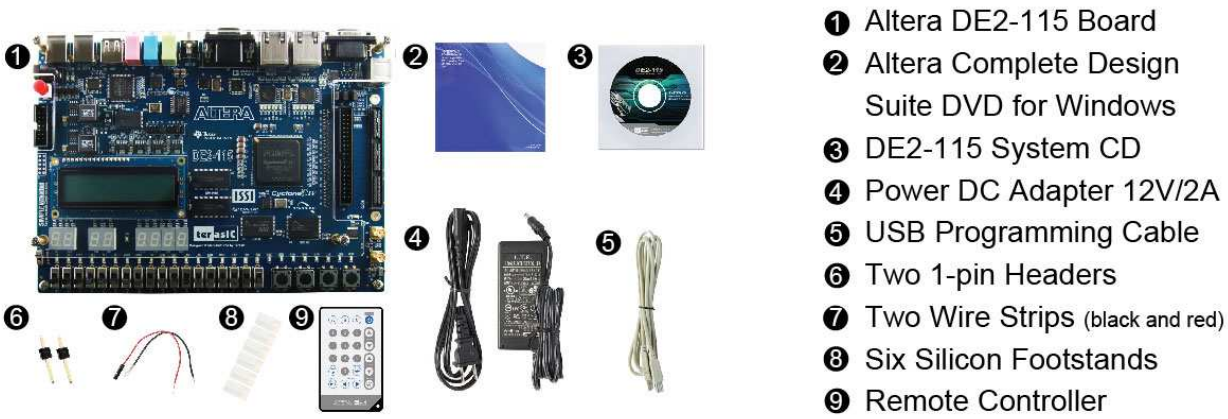


Figure 1-1 The DE2-115 package contents

The DE2-115 package includes:

- **The DE2-115 board.**
- **USB Cable for FPGA programming and control.**
- **DE2-115 System CD containing the DE2-115 documentation and supporting materials, including the User Manual, the Control Panel, System Builder and Altera Monitor Program utility, reference designs and demonstrations, device datasheets, tutorials, and a set of laboratory exercises.**
- **CD-ROMs containing Altera's Quartus® II Web Edition and the Nios® II Embedded Design Suit Evaluation Edition software.**

-
- **Bag of six rubber (silicon) covers for the DE2-115 board stands. The bag also contains some extender pins, which can be used to facilitate easier probing with testing equipment of the board's I/O expansion headers.**
- **Clear plastic cover for the board.**
- **12V DC desktop power supply.**
- **Remote controller**

1.2 The DE2-115 Board Assembly

To assemble the included stands for the DE2-115 board:

- **Assemble a rubber (silicon) cover, as shown in Figure 1-2, for each of the six copper stands on the DE2-115 board**
- **The clear plastic cover provides extra protection, and is mounted over the top of the board by using additional stands and screws**

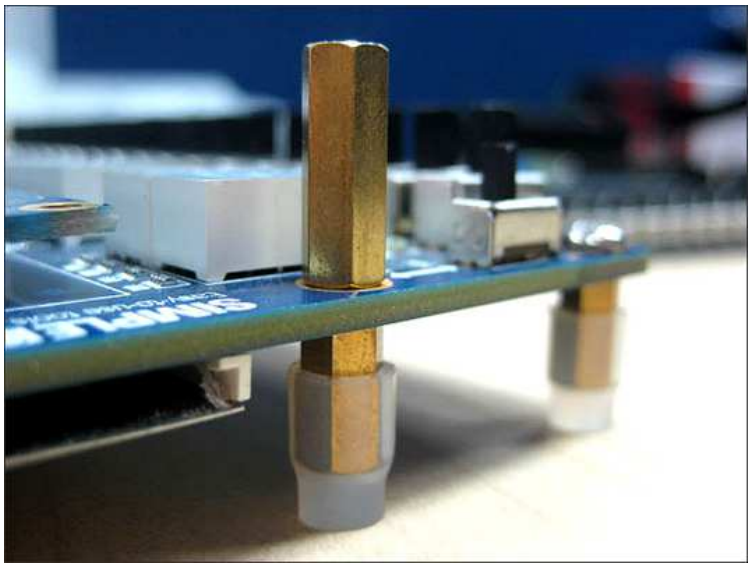


Figure 1-2 The feet for the DE2-115 board

1.3 Getting Help

Here is information of how to get help if you encounter any problem:

- **Terasic Technologies**
- **Tel: +886-3-575-0880**
- **Email: support@terasic.com**

- **Altera Corporation**
- **Email: university@altera.com**

Chapter 2

Introduction of the Altera DE2-115 Board

This chapter presents the features and design characteristics of the DE2-115 board.

2.1 Layout and Components

A photograph of the DE2-115 board is shown in Figure 2-1 and Figure 2-2. It depicts the layout of the board and indicates the location of the connectors and key components.

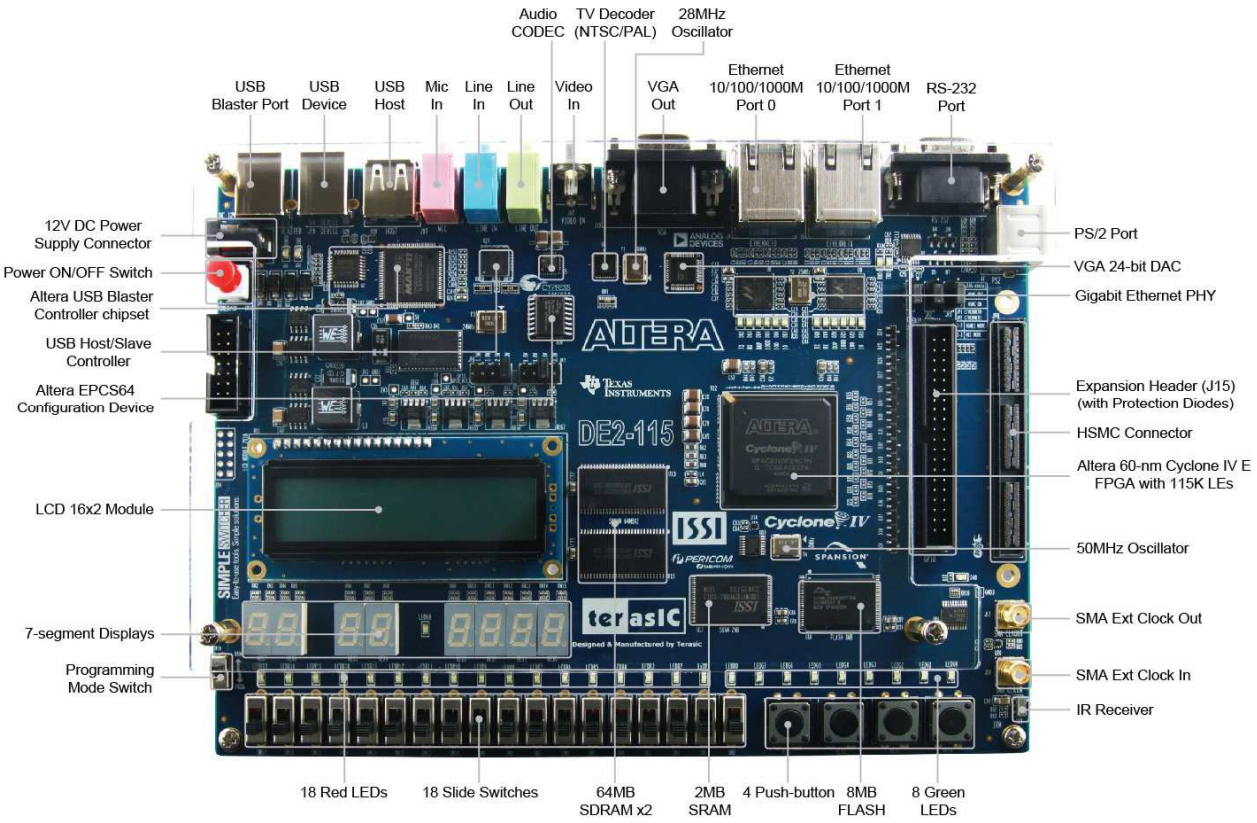


Figure 2-1 The DE2-115 board (top view)

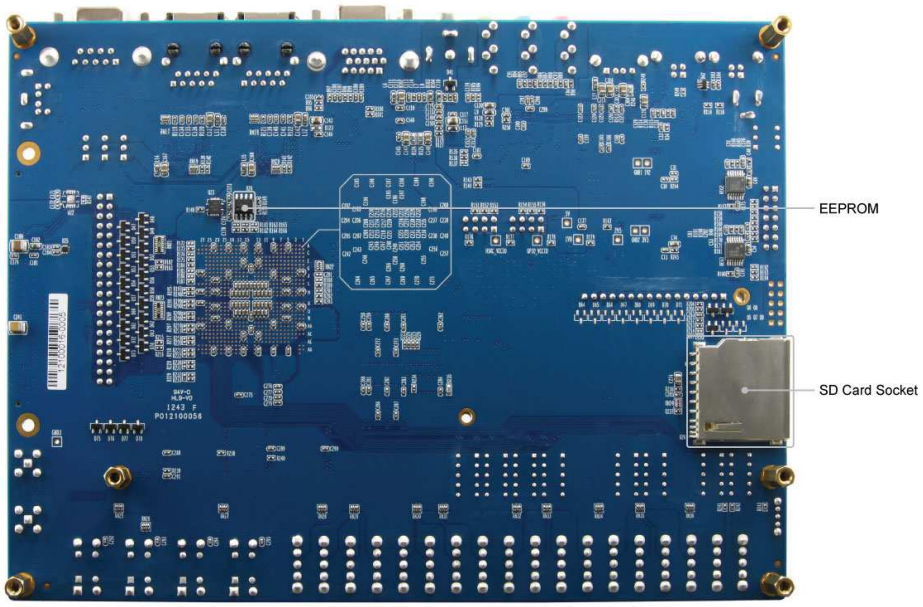


Figure 2-2 The DE2-115 board (bottom view)

The DE2-115 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 is provided on the DE2-115 board:

- **Altera Cyclone® IV 4CE115 FPGA device**
- **Altera Serial Configuration device – EPCS64**
- **USB Blaster (on board) for programming; both JTAG and Active Serial (AS) programming modes are supported**
- **2MB SRAM**
- **Two 64MB SDRAM**
- **8MB Flash memory**
- **SD Card socket**
- **4 Push-buttons**
- **18 Slide switches**
- **18 Red user LEDs**
- **9 Green user LEDs**
- **50MHz oscillator for clock sources**
- **24-bit CD-quality audio CODEC with line-in, line-out, and microphone-in jacks**
- **VGA DAC (8-bit high-speed triple DACs) with VGA-out connector**
- **TV Decoder (NTSC/PAL/SECAM) and TV-in connector**
- **2 Gigabit Ethernet PHY with RJ45 connectors**
- **USB Host/Slave Controller with USB type A and type B connectors**
- **RS-232 transceiver and 9-pin connector**

- **PS/2 mouse/keyboard connector**
- **IR Receiver**
- **2 SMA connectors for external clock input/output**
- **One 40-pin Expansion Header with diode protection**
- **One High Speed Mezzanine Card (HSMC) connector**
- **16x2 LCD module**

In addition to these hardware features, the DE2-115 board has software support for standard I/O interfaces and a control panel facility for accessing various components. Also, the software is provided for supporting a number of demonstrations that illustrate the advanced capabilities of the DE2-115 board.

In order to use the DE2-115 board, the user has to be familiar with the Quartus II software. The necessary knowledge can be acquired by reading the tutorials "*Getting Started with Altera's DE2-115 Board*" (tut_initialDE2-115.pdf) and "*Quartus II Introduction*" (which exists in three versions based on the design entry method used, namely Verilog, VHDL or schematic entry). These tutorials are provided in the directory DE2_115_tutorials on the **DE2-115 System CD** that accompanies the DE2-115 kit and can also be found on Terasic's DE2-115 web pages**.**

2.2 Block Diagram of the DE2-115 Board

Figure 2-3 gives the block diagram of the DE2-115 board. To provide maximum flexibility for the user, all connections are made through the Cyclone IV E FPGA device. Thus, the user can configure the FPGA to implement any system design.

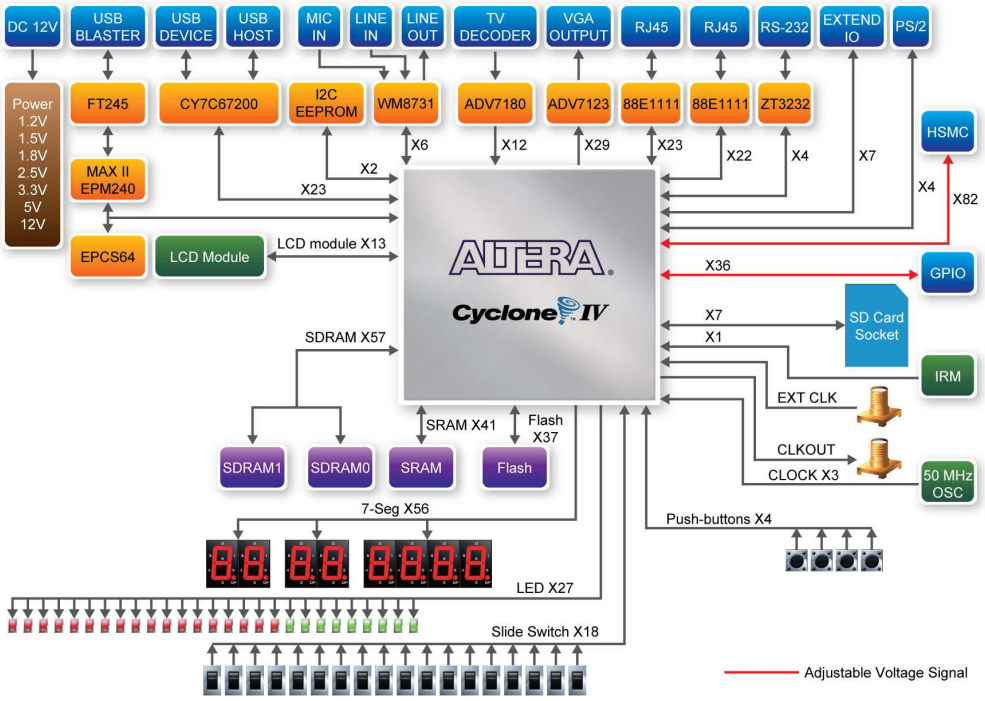


Figure 2-3 Block Diagram of DE2-115

Following is more detailed information about the blocks in **Figure 2-3**:

FPGA device

- **Cyclone IV EP4CE115F29 device**
- **114,480 LEs**
- **432 M9K memory blocks**
- **3,888 Kbits embedded memory**
- **4 PLLs**

FPGA configuration

- **JTAG and AS mode configuration**
- **EPCS64 serial configuration device**
- **On-board USB Blaster circuitry**

Memory devices

- **128MB (32Mx32bit) SDRAM**
- **2MB (1Mx16) SRAM**

- **8MB (4Mx16) Flash with 8-bit mode**
- **32Kb EEPROM**

SD Card socket

- **Provides SPI and 4-bit SD mode for SD Card access**

Connectors

- **Two Ethernet 10/100/1000 Mbps ports**
- **High Speed Mezzanine Card (HSMC)**
- **Configurable I/O standards (voltage levels:3.3/2.5/1.8/1.5V)**
- **USB type A and B**
    - Provide host and device controllers compliant with USB 2.0
    - Support data transfer at full-speed and low-speed
    - PC driver available

- **40-pin expansion port**
    - Configurable I/O standards (voltage levels:3.3/2.5/1.8/1.5V)

- **VGA-out connector**
    - VGA DAC (high speed triple DACs)

- **DB9 serial connector for RS-232 port with flow control**
- **PS/2 mouse/keyboard**
-

Clock

- **Three 50MHz oscillator clock inputs**
- **SMA connectors (external clock input/output)**

Audio

- **24-bit encoder/decoder (CODEC)**
- **Line-in, line-out, and microphone-in jacks**

Display

- **16x2 LCD module**

Switches and indicators

- **18 slide switches and 4 push-buttons switches**
- **18 red and 9 green LEDs**
- **Eight 7-segment displays**

Other features

- **Infrared remote-control receiver module**
- **TV decoder (NTSC/PAL/SECAM) and TV-in connector**

Power

- **Desktop DC input**
- **Switching and step-down regulators LM3150MH**

2.3 Power-up the DE2-115 Board

The DE2-115 board comes with a preloaded configuration bit stream to demonstrate some features of the board. This bit stream also allows users to see quickly if the board is working properly. To power-up the board perform the following steps:

1. Connect the provided USB cable from the host computer to the USB Blaster connector on the DE2-115 board. For communication between the host and the DE2-115 board, it is necessary to install the Altera USB Blaster driver software. If this driver is not already installed on the host computer, it can be installed as explained in the tutorial *"Getting Started with Altera's DE2-115 Board"* (tut_initialDE2-115.pdf). This tutorial is available in the directory DE2_115_tutorials on the **DE2-115 System CD.**
2. Turn off the power by pressing the red ON/OFF switch before connecting the 12V adapter to the DE2-115 board.
3. Connect a VGA monitor to the VGA port on the DE2-115 board.
4. Connect your headset to the line-out audio port on the DE2-115 board.

5. Turn the RUN/PROG switch (SW19) on the left edge of the DE2-115 board to RUN position; the PROG position is used only for the AS Mode programming.

6. Recycle the power by turning the red power switch on the DE2-115 board OFF and ON again .

At this point you should observe the following:

- **All user LEDs are flashing**
- **All 7-segment displays are cycling through the numbers 0 to F**
- **The LCD display shows "Welcome to the Altera DE2-115"**
- **The VGA monitor displays the image shown in Figure 2-4**
- **Set the slide switch SW17 to the DOWN position; you should hear a 1-kHz sound. Be careful of the very loud volume for avoiding any discomfort**
- **Set the slide switch SW17 to the UP position and connect the output of an audio player to the line-in connector on the DE2-115 board; on your speaker or headset you should hear the music played from the audio player (MP3, PC, iPod, or the like)**
-
-
- **You can also connect a microphone to the microphone-in connector on the DE2-115 board; your voice will be mixed with the music playing on the audio player**



Figure 2-4 The default VGA output pattern

Chapter 3

DE2-115 Control Panel

The DE2-115 board comes with a Control Panel facility that allows users to access various components on the board from a host computer. The host computer communicates with the board through a USB connection. The facility can be used to verify the functionality of components on the board or be used as a debug tool while developing RTL code.

This chapter first presents some basic functions of the Control Panel, then describes its structure in block diagram form, and finally describes its capabilities.

3.1 Control Panel Setup

The Control Panel Software Utility is located in the directory *"DE2_115_tools/DE2_115_control_panel"* in the **DE2-115 System CD**. It's free of installation, just copy the whole folder to your host computer and launch the control panel by executing the "DE2_115_ControlPanel.exe". (Windows 7 64-bit Users: If an error message that shows a missing jtag_client.dll file (cannot find jtag_client.dll) while the Control Panel is commencing, users should re-launch the DE4_ControlPanel.exe from the following directory (/DE2_115_tools/DE2_115_control_panel/win7_64bits))

Specific control circuit should be downloaded to your FPGA board before the control panel can request it to perform required tasks. The program will call Quartus II tools to download the control circuit to the FPGA board through USB-Blaster[USB-0] connection.

To activate the Control Panel, perform the following steps:

1. Make sure Quartus II 10.0 or later version is installed successfully on your PC.
2. Set the RUN/PROG switch to the RUN position.
3. Connect the supplied USB cable to the USB Blaster port, connect the 12V power supply, and turn the power switch ON.
4. Start the executable DE2_115_ControlPanel.exe on the host computer. The Control Panel user interface shown in **Figure 3-1** will appear.

5. The DE2_115_ControlPanel.sof bit stream is loaded automatically as soon as the DE2_115_control_panel.exe is launched.

6. In case the connection is disconnected, click on CONNECT where the .sof will be re-loaded onto the board.

7. *Note, the Control Panel will occupy the USB port until you close that port; you cannot use Quartus II to download a configuration file into the FPGA until the USB port is closed.*

8. The Control Panel is now ready for use; experience it by setting the ON/OFF status for some LEDs and observing the result on the DE2-115 board.



Figure 3-1 The DE2-115 Control Panel

The concept of the DE2-115 Control Panel is illustrated in **Figure 3-2**. The "Control Circuit" that performs the control functions is implemented in the FPGA board. It communicates with the Control Panel window, which is active on the host computer, via the USB Blaster link. The graphical interface is used to issue commands to the control circuit. It handles all requests and performs data transfers between the computer and the DE2-115 board.

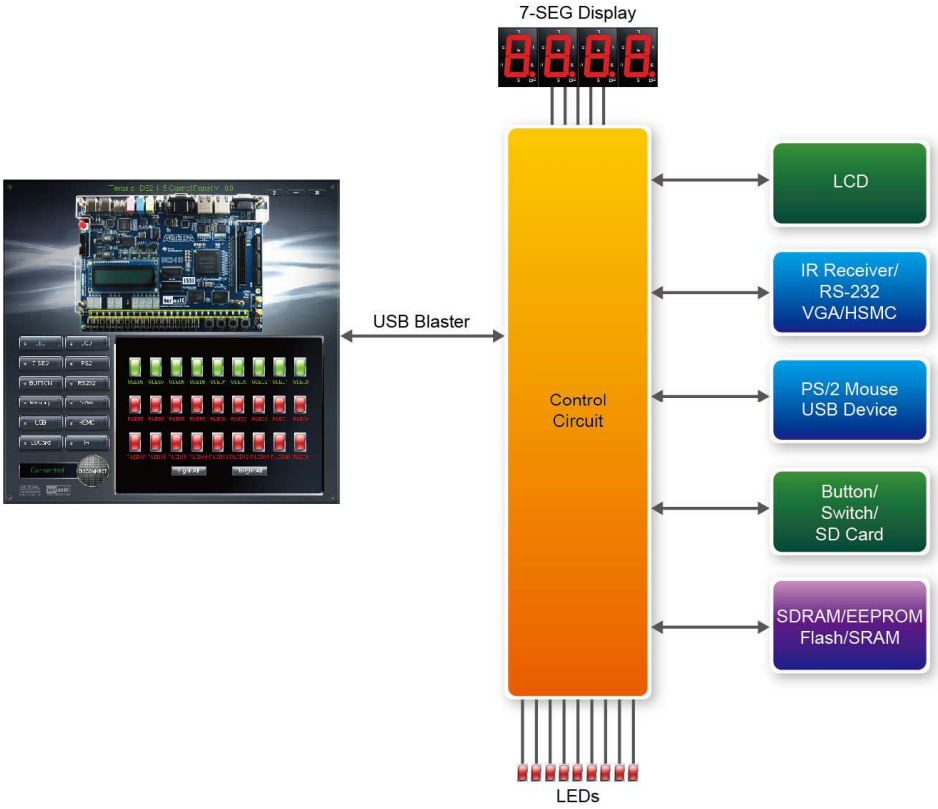


Figure 3-2 The DE2-115 Control Panel concept

The DE2-115 Control Panel can be used to light up LEDs, change the values displayed on 7-segment and LCD displays, monitor buttons/switches status, read/write the SDRAM, SRAM, EEPROM and Flash Memory, monitor the status of an USB device, communicate with the PS/2 mouse, output VGA color pattern to VGA monitor, verify functionality of HSMC connector I/Os, communicate with PC via RS-232 interface and read SD Card specification information. The feature of reading/writing a word or an entire file from/to the Flash Memory allows the user to develop multimedia applications (Flash Audio Player, Flash Picture Viewer) without worrying about how to build a Memory Programmer.

3.2 Controlling the LEDs, 7-segment Displays and LCD Display

A simple function of the Control Panel is to allow setting the values displayed on LEDs, 7-segment displays, and the LCD character display.

Choosing the **LED** tab leads to the window in Figure 3-3. Here, you can directly turn the LEDs on or off individually or by clicking “Light All” or “Unlight All”.



Figure 3-3 Controlling LEDs

Choosing the 7-SEG tab leads to the window shown in **Figure 3-4**. From the window, directly use the left-right arrows to control the 7-SEG patterns on the DE2-115 board which are updated immediately. Note that the dots of the 7-SEGs are not enabled on DE2-115 board.

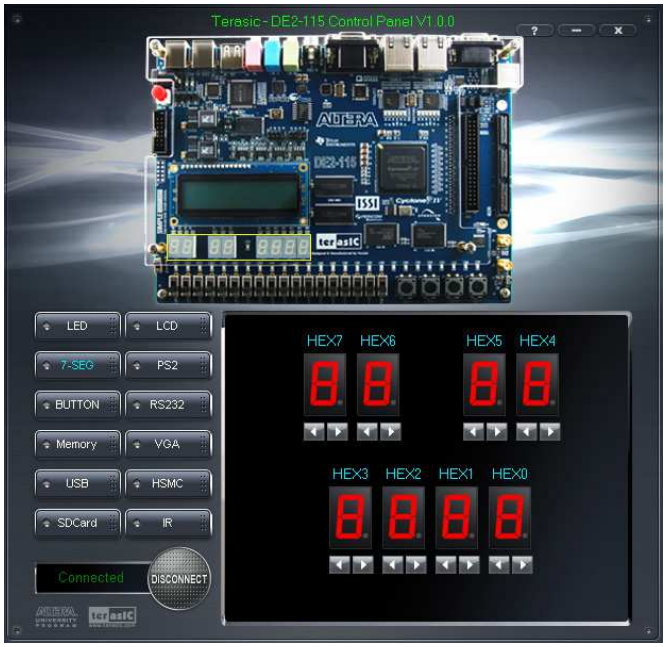


Figure 3-4 Controlling 7-SEG display

Choosing the LCD tab leads to the window in **Figure 3-5**. Text can be written to the LCD display by typing it in the LCD box then pressing the Set button.

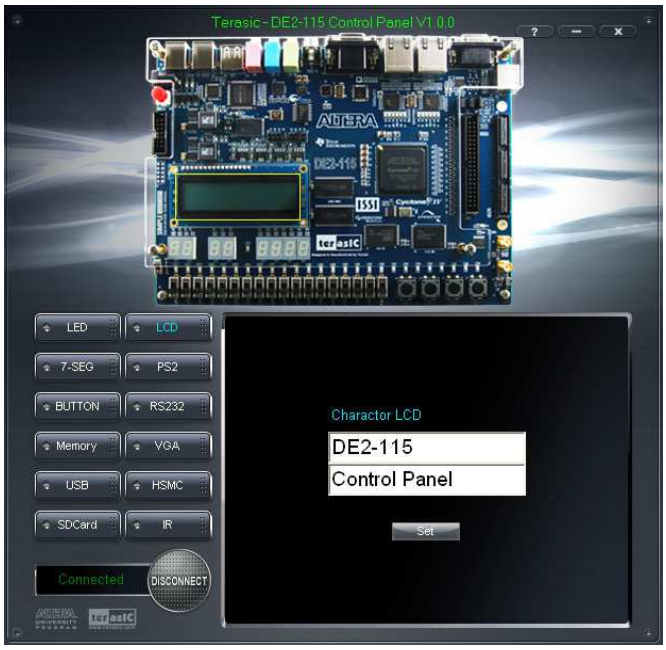


Figure 3-5 Controlling the LCD display

The ability to set arbitrary values into simple display devices is not needed in typical design activities. However, it gives the user a simple mechanism for verifying that these devices are functioning correctly in case a malfunction is suspected. Thus, it can be used for troubleshooting purposes.

3.3 Switches and Push-buttons

Choosing the Switches tab leads to the window in **Figure 3-6**. The function is designed to monitor the status of slide switches and push-buttons in real time and show the status in a graphical user interface. It can be used to verify the functionality of the slide switches and push-buttons.



Figure 3-6 Monitoring switches and buttons

The ability to check the status of push-button and slide switch is not needed in typical design activities. However, it provides users a simple mechanism for verifying if the buttons and switches are functioning correctly. Thus, it can be used for troubleshooting purposes.

3.4 SDRAM/SRAM/EEPROM/Flash Controller and Programmer

The Control Panel can be used to write/read data to/from the SDRAM, SRAM, EEPROM, and Flash chips on the DE2-115 board. As an example, we will describe how the SDRAM may be accessed; the same approach is used to access the SRAM, EEPROM, and Flash. Click on the Memory tab and select “SDRAM” to reach the window in **Figure 3-7**.



Figure 3-7 Accessing the SDRAM

A 16-bit word can be written into the SDRAM by entering the address of the desired location, specifying the data to be written, and pressing the Write button. Contents of the location can be read by pressing the Read button. **Figure 3-7** depicts the result of writing the hexadecimal value 06CA into offset address 200, followed by reading the same location.

The Sequential Write function of the Control Panel is used to write the contents of a file into the SDRAM as follows:

1. Specify the starting address in the Address box.
2. Specify the number of bytes to be written in the Length box. If the entire file is to be loaded, then a checkmark may be placed in the File Length box instead of giving the number of bytes.
3. To initiate the writing process, click on the Write a File to Memory button.
4. When the Control Panel responds with the standard Windows dialog box asking for the source file, specify the desired file in the usual manner.

The Control Panel also supports loading files with a .hex extension. Files with a .hex extension are ASCII text files that specify memory values using ASCII characters to represent hexadecimal values. For example, a file containing the line

0123456789ABCDEF

Defines eight 8-bit values: 01, 23, 45, 67, 89, AB, CD, EF. These values will be loaded consecutively into the memory.

The Sequential Read function is used to read the contents of the SDRAM and fill them into a file as follows:

1. Specify the starting address in the Address box.
2. Specify the number of bytes to be copied into the file in the Length box. If the entire contents of the SDRAM are to be copied (which involves all 128 Mbytes), then place a checkmark in the Entire Memory box.
3. Press Load Memory Content to a File button.
4. When the Control Panel responds with the standard Windows dialog box asking for the destination file, specify the desired file in the usual manner.

Users can use the similar way to access the SRAM, EEPROM and Flash. Please note that users need to erase the Flash before writing data to it.

3.5 USB Monitoring

The Control Panel provides users a USB monitoring tool which monitors the status of the USB devices connected to the USB port on the DE2-115 board. By plugging in a USB device to the USB host port of the board, the device type is displayed on the control window. **Figure 3-8** shows a USB mouse plugged into the host USB port.



Figure 3-8 USB Mouse Monitoring Tool

3.6 PS/2 Device

The Control Panel provides users a PS/2 monitoring tool which monitors the real-time status of a PS/2 mouse connected to the DE2-115 board. The movement of the mouse and the status of the three buttons will be shown in the graphical and text interface. The mouse movement is translated as a position (x,y) with range from (0,0)~(1023,767). This function can be used to verify the functionality of the PS/2 connection.

Follow the steps below to exercise the PS/2 Mouse Monitoring tool:

1. Choosing the PS/2 tab leads to the window in **Figure 3-9**.
2. Plug a PS/2 mouse to the PS/2 port on the DE2-115 board.
3. Press the Start button to start the PS/2 mouse monitoring process, and the button caption is changed from Start to Stop. In the monitoring process, the status of the PS/2 mouse is updated and shown in the Control Panel's GUI window in real-time. Press Stop to terminate the monitoring process.



Figure 3-9 PS/2 Mouse Monitoring Tool

3.7 SD Card

The function is designed to read the identification and specification information of the SD Card. The 4-bit SD MODE is used to access the SD Card. This function can be used to verify the functionality of the SD Card Interface. Follow the steps below to exercise the SD Card:

1. Choosing the SD Card tab leads to the window in **Figure 3-10.**
2. Insert an SD Card to the DE2-115 board, and then press the Read button to read the SD Card. The SD Card's identification, specification, and file format information will be displayed in the control window.



Figure 3-10 Reading the SD Card Identification and Specification

3.8 RS-232 Communication

The Control Panel allows users to verify the operation of the RS-232 serial communication interface on the DE2-115. The setup is established by connecting a RS-232 9-pin male to female cable from the PC to the RS-232 port where the Control Panel communicates to the terminal emulator software on the PC, or vice versa. Alternatively, a RS-232 loopback cable can also be used if you do not wish to use the PC to verify the test. The Receive terminal window on the Control Panel monitors the serial communication status. Follow the steps below to initiate the RS-232 communication:

1. Choosing the RS-232 tab leads to the window in **Figure 3-11.**

2. Plug in a RS-232 9-pin male to female cable from PC to RS-232 port or a RS-232 loopback cable directly to RS-232 port.

3. The RS-232 settings are provided below in case a connection from the PC is used:

- **Baud Rate: 115200**
- **Parity Check Bit: None**
- **Data Bits: 8**
- **Stop Bits: 1**
- **Flow Control (CTS/RTS): ON**

4. To begin the communication, enter specific letters followed by clicking Send. During the communication process, observe the status of the Receive terminal window to verify its operation.



Figure 3-11 RS-232 Serial Communication

3.9 VGA

DE2-115 Control Panel provides VGA pattern function that allows users to output color pattern to LCD/CRT monitor using the DE2-115 board. Follow the steps below to generate the VGA pattern function:

1. Choosing the VGA tab leads to the window in **Figure 3-12.**

2. Plug a D-sub cable to VGA connector of the DE2-115 board and LCD/CRT monitor.

3. The LCD/CRT monitor will display the same color pattern on the control panel window.

4. Click the drop down menu shown in **Figure 3-12** where you can output the selected color individually.

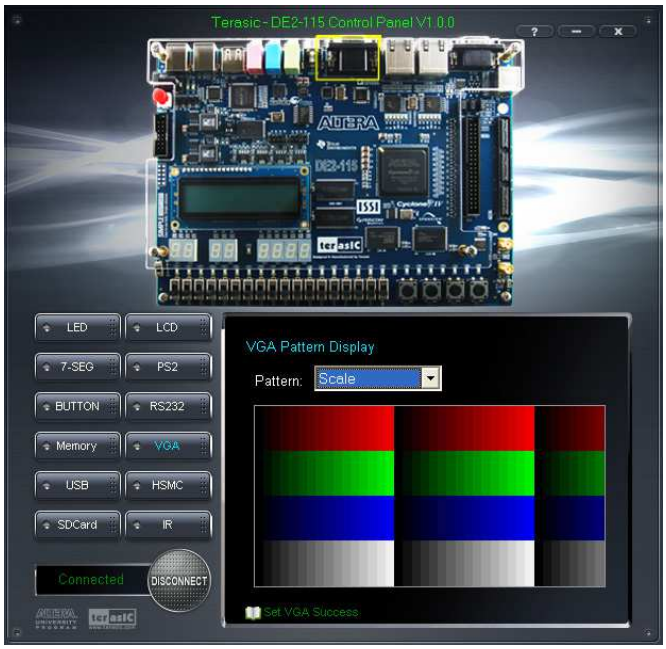


Figure 3-12 Controlling VGA display

3.10 HSMC

Select the HSMC tab to reach the window shown in **Figure 3-13**. This function is designed to verify the functionality of the signals located on the HSMC connector. Before running the HSMC loopback verification test, follow the instruction noted under the Loopback Installation section and click on Verify. Please note to turn off the DE2-115 board before the HSMC loopback adapter is installed to prevent any damage to the board.

The HSMC loopback adapter is not provided in the kit package but can be purchased through the website below: (http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=78&No=495)

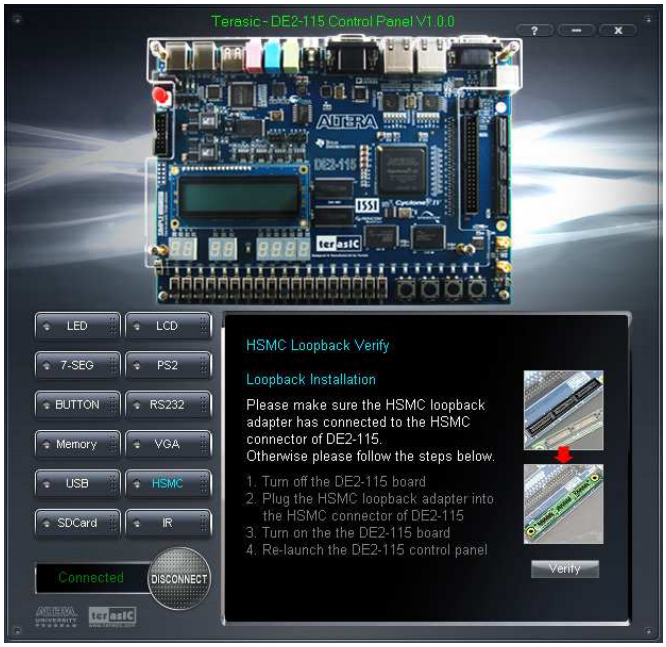


Figure 3-13 HSMC loopback verification test performed under Control Panel

3.11 IR Receiver

From the control panel, we can test the IR receiver on the DE2-115 by sending scan code from a remote controller. **Figure 3-14** depicts the IR receiver window when the IR tab is pressed. When the scan code is received, the information will be displayed on the IR Receiver window represented in hexadecimal. Also, the pressed button on the remote controller will be indicated on the graphic of remote controller on the IR receiver window. Note that there exists several encoding form among different brands of remote controllers. Only the remote controller comes with the kit is confirmed to be compatible with this software.



Figure 3-14 Testing the IR receiver using remote controller

3.12 Overall Structure of the DE2-115 Control Panel

The DE2-115 Control Panel is based on a Nios II SOPC system instantiated in the Cyclone IV E FPGA with software running on the on-chip memory. The software part is implemented in C code; the hardware part is implemented in Verilog HDL code with SOPC builder. The source code is not available on the DE2_115 System CD.

To run the Control Panel, users should make the configuration according to Section 3.1. **Figure 3-15** depicts the structure of the Control Panel. Each input/output device is controlled by the Nios II Processor instantiated in the FPGA chip. The communication with the PC is done via the USB Blaster link. The Nios II interprets the commands sent from the PC and performs the corresponding actions.

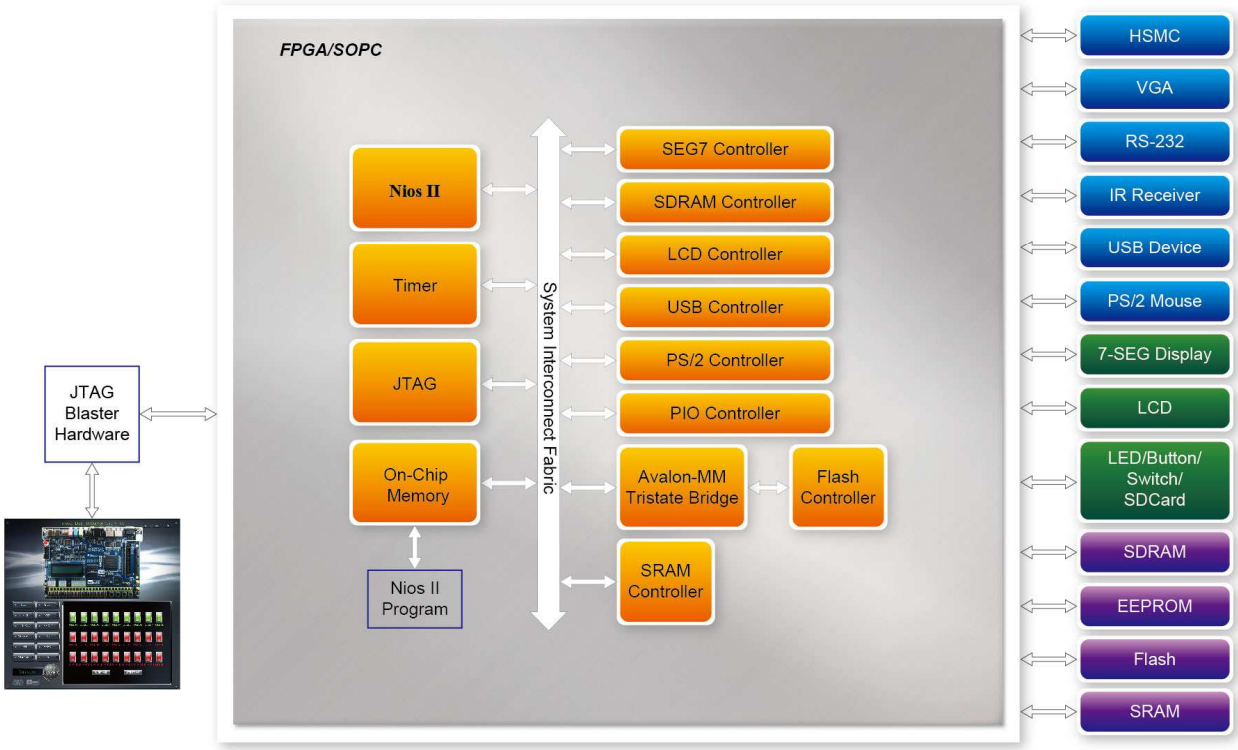


Figure 3-15 The block diagram of the DE2-115 control panel

Chapter 4

Using the DE2-115 Board

This chapter gives instructions for using the DE2-115 board and describes each of its peripherals.

4.1 Configuring the Cyclone IV E FPGA

The procedure for downloading a circuit from a host computer to the DE2-115 board is described in the tutorial Quartus II Introduction. This tutorial is found in the DE2_115_tutorials folder on the DE2-115 System CD. The user is encouraged to read the tutorial first, and treat the information below as a short reference.

The DE2-115 board contains a serial configuration device that stores configuration data for the Cyclone IV E FPGA. This configuration data is automatically loaded from the configuration device into the FPGA every time while power is applied to the board. Using the Quartus II software, it is possible to reconfigure the FPGA at any time, and it is also possible to change the non-volatile data that is stored in the serial configuration device. Both types of programming methods are described below.

1. JTAG programming: In this method of programming, named after the IEEE standards Joint Test Action Group, the configuration bit stream is downloaded directly into the Cyclone IV E FPGA. The FPGA will retain this configuration as long as power is applied to the board; the configuration information will be lost when the power is turned off.
2. AS programming: In this method, called Active Serial programming, the configuration bit stream is downloaded into the Altera EPCS64 serial configuration device. It provides non-volatile storage of the bit stream, so that the information is retained even when the power supply to the DE2-115 board is turned off. When the board's power is turned on, the configuration data in the EPCS64 device is automatically loaded into the Cyclone IV E FPGA.

■ JTAG Chain on DE2-115 Board

To use JTAG interface for configuring FPGA device, the JTAG chain on DE2-115 must form a close loop that allows Quartus II programmer to detect FPGA device. **Figure 4-1** illustrates the

JTAG chain on DE2-115 board. Shorting pin1 and pin2 on JP3 can disable the JTAG signals on HSMC connector that will form a close JTAG loop chain on DE2-115 board (See **Figure 4-2**). Thus, only the on board FPGA device (Cyclone IV E) will be detected by Quartus II programmer. If users want to include another FPGA device or interface containing FPGA device in the chain via HSMC connector, short pin2 and pin3 on JP3 to enable the JTAG signal ports on the HSMC connector.

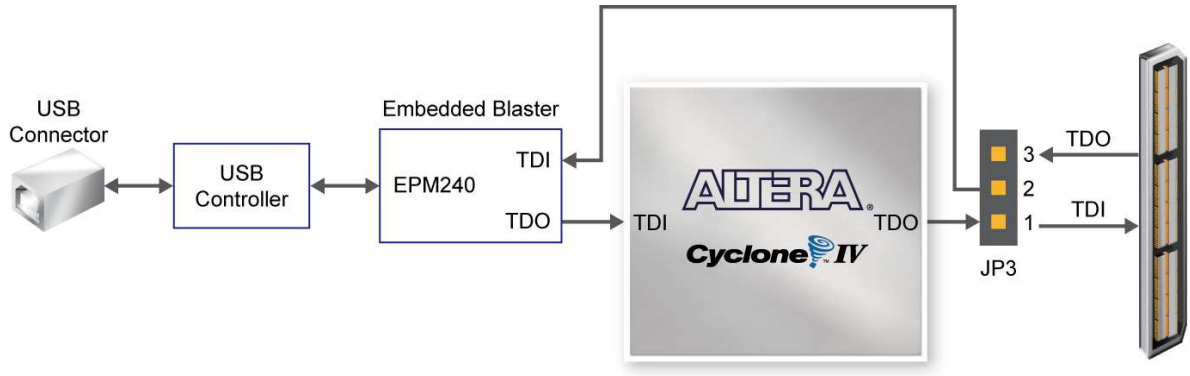


Figure 4-1 The JTAG chain on DE2-115 board

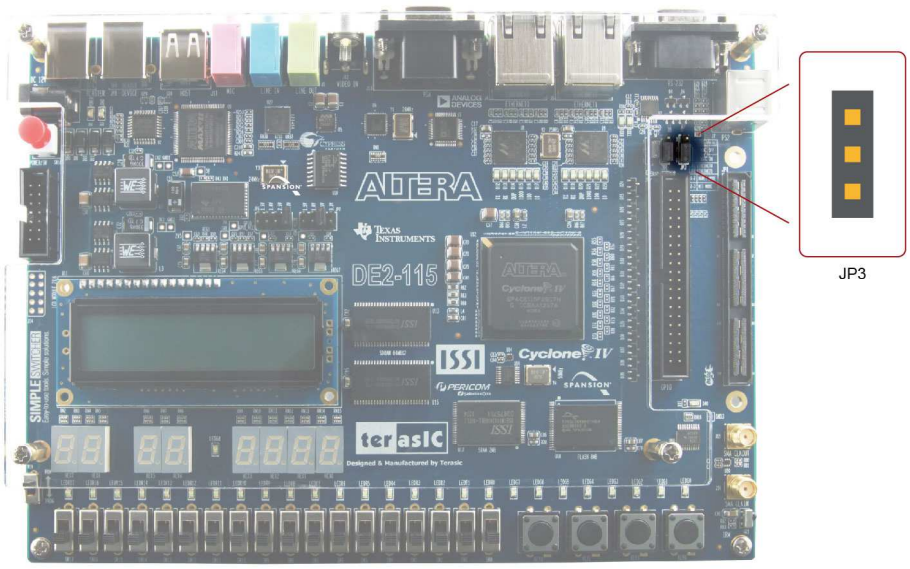


Figure 4-2 The JTAG chain configuration header

The sections below describe the steps used to perform both JTAG and AS programming. For both methods the DE2-115 board is connected to a host computer via a USB cable. Using this connection, the board will be identified by the host computer as an Altera USB Blaster device. The process for installing on the host computer the necessary software device driver that communicates with the USB Blaster is described in the tutorial *"Getting Started with Altera's DE2-115 Board"* (tut_initialDE2-115.pdf). This tutorial is available on the DE2-115 System CD.

■ Configuring the FPGA in JTAG Mode

Figure 4-3 illustrates the JTAG configuration setup. To download a configuration bit stream into the Cyclone IV E FPGA, perform the following steps:

- **Ensure that power is applied to the DE2-115 board**
- **Configure the JTAG programming circuit by setting the RUN/PROG slide switch (SW19) to the RUN position (See Figure 4-4)**
- **Connect the supplied USB cable to the USB Blaster port on the DE2-115 board (See Figure 2-1)**
- **The FPGA can now be programmed by using the Quartus II Programmer to select a configuration bit stream file with the .sof filename extension**

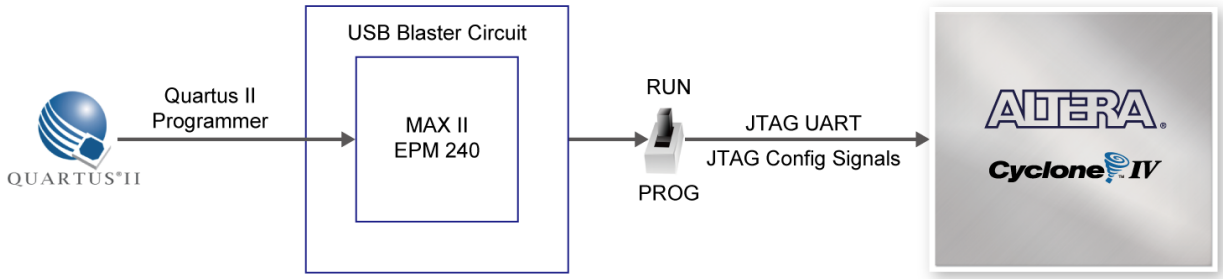


Figure 4-3 The JTAG configuration scheme

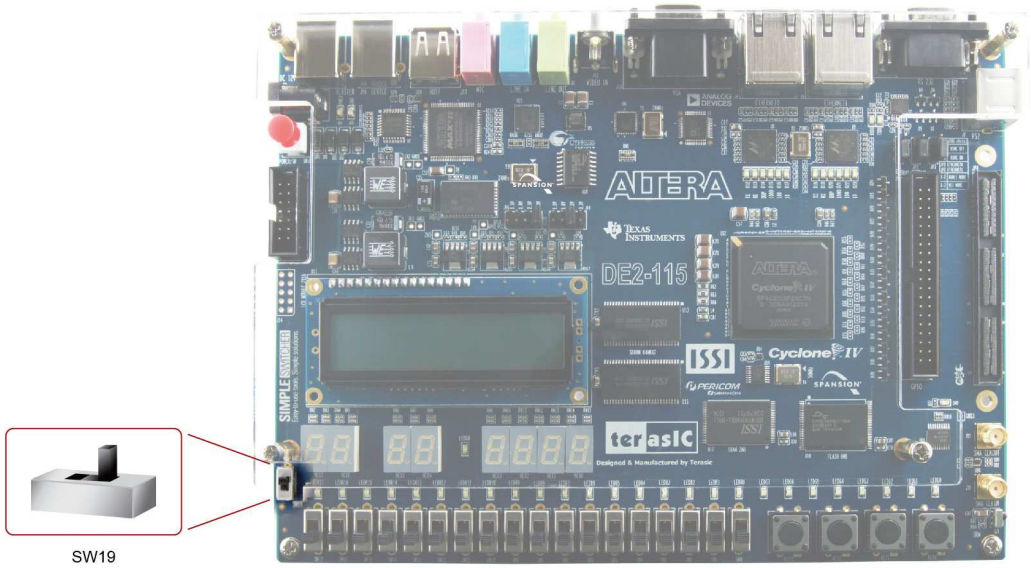


Figure 4-4 The RUN/PROG switch (SW19) is set in JTAG mode

■ Configuring the EPCS64 in AS Mode

Figure 4-5 illustrates the AS configuration setup. To download a configuration bit stream into the EPCS64 serial configuration device, perform the following steps:

- **Ensure that power is applied to the DE2-115 board.**
- **Connect the supplied USB cable to the USB Blaster port on the DE2-115 board (See Figure 4-5)**
- **Configure the JTAG programming circuit by setting the RUN/PROG slide switch (SW19) to the PROG position.**
- **The EPCS64 chip can now be programmed by using the Quartus II Programmer to select a configuration bit stream file with the .pof filename extension.**
- **Once the programming operation is finished, set the RUN/PROG slide switch back to the RUN position and then reset the board by turning the power switch off and back on; this action causes the new configuration data in the EPCS64 device to be loaded into the FPGA chip.**
-

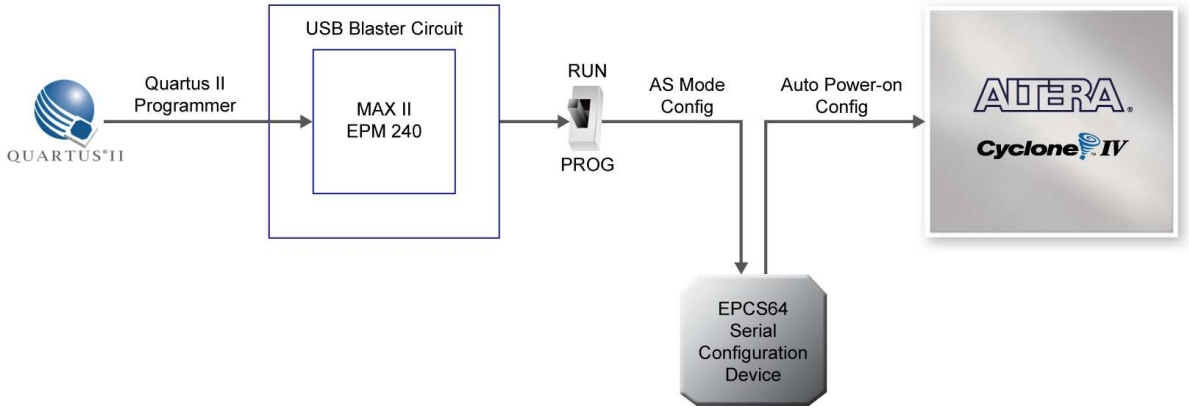


Figure 4-5 The AS configuration scheme

4.2 Using Push-buttons and Switches

The DE2-115 board provides four push-button switches as shown in Figure 4-6. Each of these switches is debounced using a Schmitt Trigger circuit, as indicated in Figure 4-7. The four outputs called KEY0, KEY1, KEY2, and KEY3 of the Schmitt Trigger devices are connected directly to the Cyclone IV E FPGA. Each push-button switch provides a high logic level when it is not pressed, and provides a low logic level when depressed. Since the push-button switches are debounced, they are appropriate for using as clock or reset inputs in a circuit.

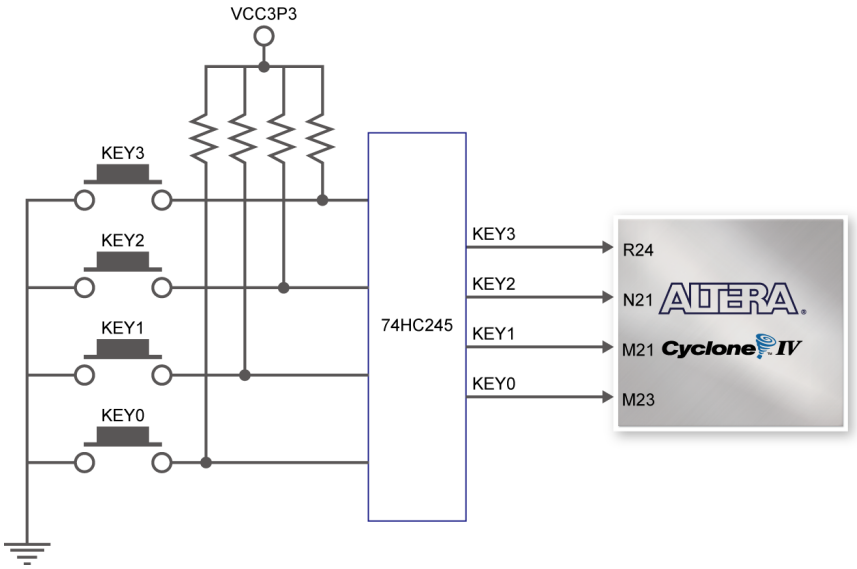


Figure 4-6 Connections between the push-button and Cyclone IV E FPGA

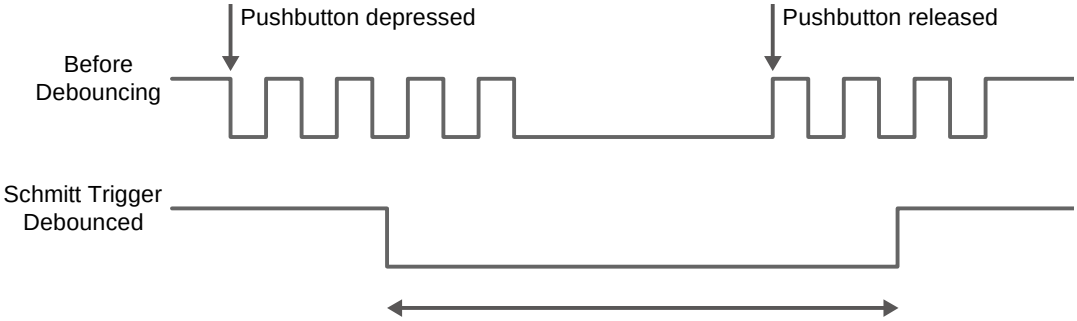


Figure 4-7 Switch debouncing

There are also 18 slide switches on the DE2-115 board (See **Figure 4-8**). These switches are not debounced, and are assumed for use as level-sensitive data inputs to a circuit. Each switch is connected directly to a pin on the Cyclone IV E 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.

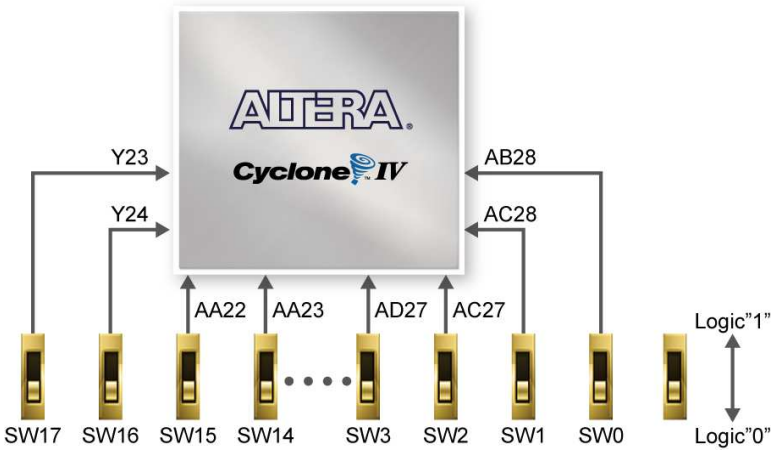


Figure 4-8 Connections between the slide switches and Cyclone IV E FPGA

4.3 Using LEDs

There are 27 user-controllable LEDs on the DE2-115 board. Eighteen red LEDs are situated above the 18 Slide switches, and eight green LEDs are found above the push-button switches (the 9th green LED is in the middle of the 7-segment displays). Each LED is driven directly by a pin on the Cyclone IV E FPGA; driving its associated pin to a high logic level turns the LED on, and driving the pin low turns it off. Figure 4-9 shows the connections between LEDs and Cyclone IV E FPGA.



Figure 4-9 Connections between the LEDs and Cyclone IV E FPGA

A list of the pin names on the Cyclone IV E FPGA that are connected to the slide switches is given in Table 4-1. Similarly, the pins used to connect to the push-button switches and LEDs are displayed in Table 4-2 and Table 4-3, respectively.

Table 4-1 Pin Assignments for Slide Switches

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
SW[0]	PIN_AB28	Slide Switch[0]	Depending on JP7
SW[1]	PIN_AC28	Slide Switch[1]	Depending on JP7
SW[2]	PIN_AC27	Slide Switch[2]	Depending on JP7
SW[3]	PIN_AD27	Slide Switch[3]	Depending on JP7
SW[4]	PIN_AB27	Slide Switch[4]	Depending on JP7
SW[5]	PIN_AC26	Slide Switch[5]	Depending on JP7
SW[6]	PIN_AD26	Slide Switch[6]	Depending on JP7
SW[7]	PIN_AB26	Slide Switch[7]	Depending on JP7
SW[8]	PIN_AC25	Slide Switch[8]	Depending on JP7
SW[9]	PIN_AB25	Slide Switch[9]	Depending on JP7
SW[10]	PIN_AC24	Slide Switch[10]	Depending on JP7
SW[11]	PIN_AB24	Slide Switch[11]	Depending on JP7
SW[12]	PIN_AB23	Slide Switch[12]	Depending on JP7
SW[13]	PIN_AA24	Slide Switch[13]	Depending on JP7
SW[14]	PIN_AA23	Slide Switch[14]	Depending on JP7
SW[15]	PIN_AA22	Slide Switch[15]	Depending on JP7
SW[16]	PIN_Y24	Slide Switch[16]	Depending on JP7
SW[17]	PIN_Y23	Slide Switch[17]	Depending on JP7

Table 4-2 Pin Assignments for Push-buttons

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
KEY[0]	PIN_M23	Push-button[0]	Depending on JP7
KEY[1]	PIN_M21	Push-button[1]	Depending on JP7
KEY[2]	PIN_N21	Push-button[2]	Depending on JP7
KEY[3]	PIN_R24	Push-button[3]	Depending on JP7

Table 4-3 Pin Assignments for LEDs

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
LEDR[0]	PIN_G19	LED Red[0]	2.5V
LEDR[1]	PIN_F19	LED Red[1]	2.5V
LEDR[2]	PIN_E19	LED Red[2]	2.5V
LEDR[3]	PIN_F21	LED Red[3]	2.5V
LEDR[4]	PIN_F18	LED Red[4]	2.5V
LEDR[5]	PIN_E18	LED Red[5]	2.5V
LEDR[6]	PIN_J19	LED Red[6]	2.5V
LEDR[7]	PIN_H19	LED Red[7]	2.5V
LEDR[8]	PIN_J17	LED Red[8]	2.5V
LEDR[9]	PIN_G17	LED Red[9]	2.5V
LEDR[10]	PIN_J15	LED Red[10]	2.5V
LEDR[11]	PIN_H16	LED Red[11]	2.5V
LEDR[12]	PIN_J16	LED Red[12]	2.5V
LEDR[13]	PIN_H17	LED Red[13]	2.5V

LEDR[14]	PIN_F15	LED Red[14]	2.5V
LEDR[15]	PIN_G15	LED Red[15]	2.5V
LEDR[16]	PIN_G16	LED Red[16]	2.5V
LEDR[17]	PIN_H15	LED Red[17]	2.5V
LEDG[0]	PIN_E21	LED Green[0]	2.5V
LEDG[1]	PIN_E22	LED Green[1]	2.5V
LEDG[2]	PIN_E25	LED Green[2]	2.5V
LEDG[3]	PIN_E24	LED Green[3]	2.5V
LEDG[4]	PIN_H21	LED Green[4]	2.5V
LEDG[5]	PIN_G20	LED Green[5]	2.5V
LEDG[6]	PIN_G22	LED Green[6]	2.5V
LEDG[7]	PIN_G21	LED Green[7]	2.5V
LEDG[8]	PIN_F17	LED Green[8]	2.5V

4.4 Using the 7-segment Displays

The DE2-115 Board has eight 7-segment displays. These displays are arranged into two pairs and a group of four, behaving the intent of displaying numbers of various sizes. As indicated in the schematic in **Figure 4-10**, the seven segments (common anode) are connected to pins on Cyclone IV E FPGA. Applying a low logic level to a segment will light it up and applying a high logic level turns it off.

Each segment in a display is identified by an index from 0 to 6, with the positions given in **Figure 4-10**. **Table 4-4** shows the assignments of FPGA pins to the 7-segment displays.

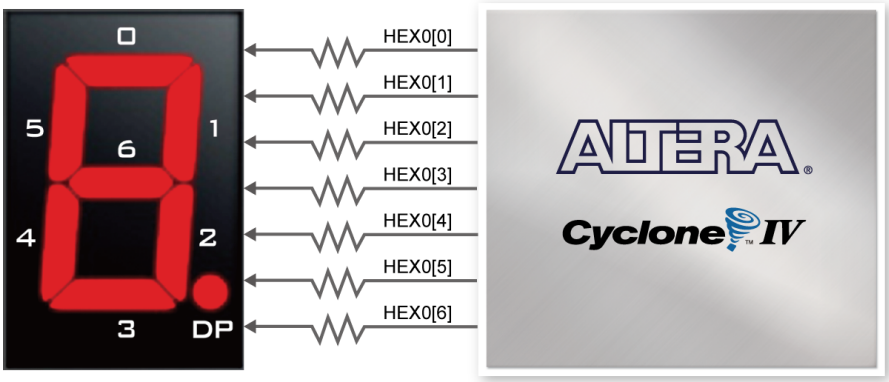


Figure 4-10 Connections between the 7-segment display HEX0 and Cyclone IV E FPGA

Table 4-4 Pin Assignments for 7-segment Displays

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
HEX0[0]	PIN_G18	Seven Segment Digit 0[0]	2.5V
HEX0[1]	PIN_F22	Seven Segment Digit 0[1]	2.5V
HEX0[2]	PIN_E17	Seven Segment Digit 0[2]	2.5V

HEX0[3]	PIN_L26	Seven Segment Digit 0[3]	Depending on JP7
HEX0[4]	PIN_L25	Seven Segment Digit 0[4]	Depending on JP7
HEX0[5]	PIN_J22	Seven Segment Digit 0[5]	Depending on JP7
HEX0[6]	PIN_H22	Seven Segment Digit 0[6]	Depending on JP7
HEX1[0]	PIN_M24	Seven Segment Digit 1[0]	Depending on JP7
HEX1[1]	PIN_Y22	Seven Segment Digit 1[1]	Depending on JP7
HEX1[2]	PIN_W21	Seven Segment Digit 1[2]	Depending on JP7
HEX1[3]	PIN_W22	Seven Segment Digit 1[3]	Depending on JP7
HEX1[4]	PIN_W25	Seven Segment Digit 1[4]	Depending on JP7
HEX1[5]	PIN_U23	Seven Segment Digit 1[5]	Depending on JP7
HEX1[6]	PIN_U24	Seven Segment Digit 1[6]	Depending on JP7
HEX2[0]	PIN_AA25	Seven Segment Digit 2[0]	Depending on JP7
HEX2[1]	PIN_AA26	Seven Segment Digit 2[1]	Depending on JP7
HEX2[2]	PIN_Y25	Seven Segment Digit 2[2]	Depending on JP7
HEX2[3]	PIN_W26	Seven Segment Digit 2[3]	Depending on JP7
HEX2[4]	PIN_Y26	Seven Segment Digit 2[4]	Depending on JP7
HEX2[5]	PIN_W27	Seven Segment Digit 2[5]	Depending on JP7
HEX2[6]	PIN_W28	Seven Segment Digit 2[6]	Depending on JP7
HEX3[0]	PIN_V21	Seven Segment Digit 3[0]	Depending on JP7
HEX3[1]	PIN_U21	Seven Segment Digit 3[1]	Depending on JP7
HEX3[2]	PIN_AB20	Seven Segment Digit 3[2]	Depending on JP6
HEX3[3]	PIN_AA21	Seven Segment Digit 3[3]	Depending on JP6
HEX3[4]	PIN_AD24	Seven Segment Digit 3[4]	Depending on JP6
HEX3[5]	PIN_AF23	Seven Segment Digit 3[5]	Depending on JP6
HEX3[6]	PIN_Y19	Seven Segment Digit 3[6]	Depending on JP6
HEX4[0]	PIN_AB19	Seven Segment Digit 4[0]	Depending on JP6
HEX4[1]	PIN_AA19	Seven Segment Digit 4[1]	Depending on JP6
HEX4[2]	PIN_AG21	Seven Segment Digit 4[2]	Depending on JP6
HEX4[3]	PIN_AH21	Seven Segment Digit 4[3]	Depending on JP6
HEX4[4]	PIN_AE19	Seven Segment Digit 4[4]	Depending on JP6
HEX4[5]	PIN_AF19	Seven Segment Digit 4[5]	Depending on JP6
HEX4[6]	PIN_AE18	Seven Segment Digit 4[6]	Depending on JP6
HEX5[0]	PIN_AD18	Seven Segment Digit 5[0]	Depending on JP6
HEX5[1]	PIN_AC18	Seven Segment Digit 5[1]	Depending on JP6
HEX5[2]	PIN_AB18	Seven Segment Digit 5[2]	Depending on JP6
HEX5[3]	PIN_AH19	Seven Segment Digit 5[3]	Depending on JP6
HEX5[4]	PIN_AG19	Seven Segment Digit 5[4]	Depending on JP6
HEX5[5]	PIN_AF18	Seven Segment Digit 5[5]	Depending on JP6
HEX5[6]	PIN_AH18	Seven Segment Digit 5[6]	Depending on JP6
HEX6[0]	PIN_AA17	Seven Segment Digit 6[0]	Depending on JP6
HEX6[1]	PIN_AB16	Seven Segment Digit 6[1]	Depending on JP6
HEX6[2]	PIN_AA16	Seven Segment Digit 6[2]	Depending on JP6
HEX6[3]	PIN_AB17	Seven Segment Digit 6[3]	Depending on JP6
HEX6[4]	PIN_AB15	Seven Segment Digit 6[4]	Depending on JP6
HEX6[5]	PIN_AA15	Seven Segment Digit 6[5]	Depending on JP6

HEX6[6]	PIN_AC17	Seven Segment Digit 6[6]	Depending on JP6
HEX7[0]	PIN_AD17	Seven Segment Digit 7[0]	Depending on JP6
HEX7[1]	PIN_AE17	Seven Segment Digit 7[1]	Depending on JP6
HEX7[2]	PIN_AG17	Seven Segment Digit 7[2]	Depending on JP6
HEX7[3]	PIN_AH17	Seven Segment Digit 7[3]	Depending on JP6
HEX7[4]	PIN_AF17	Seven Segment Digit 7[4]	Depending on JP6
HEX7[5]	PIN_AG18	Seven Segment Digit 7[5]	Depending on JP6
HEX7[6]	PIN_AA14	Seven Segment Digit 7[6]	3.3V

4.5 Clock Circuitry

The DE2-115 board includes one oscillator that produces 50 MHz clock signal. A clock buffer is used to distribute 50 MHz clock signal with low jitter to FPGA. The distributing clock signals are connected to the FPGA that are used for clocking the user logic. The board also includes two SMA connectors which can be used to connect an external clock source to the board or to drive a clock signal out through the SMA connector. In addition, all these clock inputs are connected to the phase locked loops (PLL) clock input pins of the FPGA to allow users to use these clocks as a source clock for the PLL circuit.

The clock distribution on the DE2-115 board is shown in Figure 4-11. The associated pin assignments for clock inputs to FPGA I/O pins are listed in Table 4-5.



Figure 4-11 Block diagram of the clock distribution

Table 4-5 Pin Assignments for Clock Inputs

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
CLOCK_50	PIN_Y2	50 MHz clock input	3.3V
CLOCK2_50	PIN_AG14	50 MHz clock input	3.3V
CLOCK3_50	PIN_AG15	50 MHz clock input	Depending on JP6
SMA_CLKOUT	PIN_AE23	External (SMA) clock output	Depending on JP6
SMA_CLKIN	PIN_AH14	External (SMA) clock input	3.3V

4.6 Using the LCD Module

The LCD module has built-in fonts and can be used to display text by sending appropriate commands to the display controller called HD44780. Detailed information for using the display is available in its datasheet, which can be found on the manufacturer's website, and from the *DE2_115_datasheets\LCD* folder on the DE2-115 System CD. A schematic diagram of the LCD module showing connections to the Cyclone IV E FPGA is given in **Figure 4-12**. The associated pin assignments appear in **Table 4-6**.

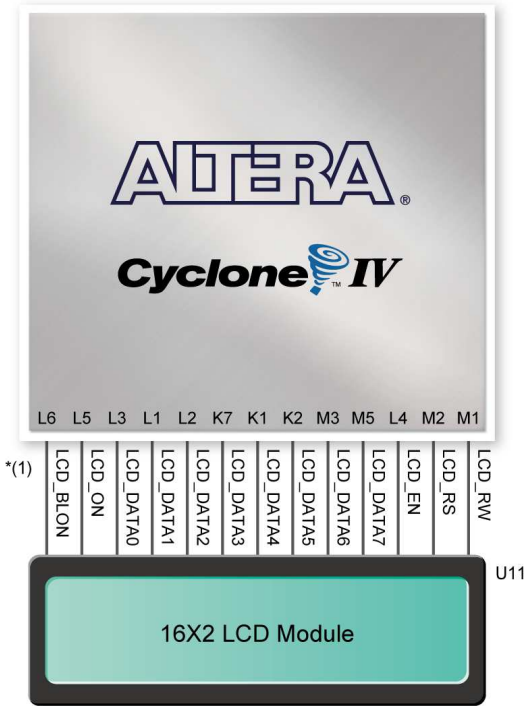


Figure 4-12 Connections between the LCD module and Cyclone IV E FPGA

*(1): Note the current LCD modules used on DE2-115 boards do not have backlight. Therefore the LCD_BLON signal should not be used in users' design projects.

Table 4-6 Pin Assignments for LCD Module

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
LCD_DATA[7]	PIN_M5	LCD Data[7]	3.3V
LCD_DATA[6]	PIN_M3	LCD Data[6]	3.3V
LCD_DATA[5]	PIN_K2	LCD Data[5]	3.3V
LCD_DATA[4]	PIN_K1	LCD Data[4]	3.3V
LCD_DATA[3]	PIN_K7	LCD Data[3]	3.3V

LCD_DATA[2]	PIN_L2	LCD Data[2]	3.3V
LCD_DATA[1]	PIN_L1	LCD Data[1]	3.3V
LCD_DATA[0]	PIN_L3	LCD Data[0]	3.3V
LCD_EN	PIN_L4	LCD Enable	3.3V
LCD_RW	PIN_M1	LCD Read/Write Select, 0 = Write, 1 = Read	3.3V
LCD_RS	PIN_M2	LCD Command/Data Select, 0 = Command, 1 = Data	3.3V
LCD_ON	PIN_L5	LCD Power ON/OFF	3.3V
LCD_BLON	PIN_L6	LCD Back Light ON/OFF	3.3V

4.7 High Speed Mezzanine Card

The DE2-115 development board contains a HSMC interface to provide a mechanism for extending the peripheral-set of a FPGA host board by means of add-on cards. This can address today's high speed signaling requirement as well as low-speed device interface support. The HSMC interface support JTAG, clock outputs and inputs, high speed LVDS and single-ended signaling. The HSMC connector connects directly to the Cyclone IV E FPGA with 82 pins. Signals HSMC_SDA and HSMC_SCLK share the same bus with the respected signals I2C_SDA and I2C_SCL of the WM8731 audio ship and ADV7180 TV decoder chip. Table 4-7 shows the maximum power consumption of the daughter card that connects to HSMC port.

Table 4-7 Power Supply of the HSMC

Supplied Voltage	*Max. Current Limit*
12V	1A
3.3V	1.5A

(1).Note the current levels indicated in Table 4-7 are based on 50% resource consumption. If the HSMC interface is utilized with design resources exceeding 50%, please notify our support (support@terasic.com).

(2).If the HSMC loopback adapter is mounted, the I2C_SCL will be directly routed back to I2C_SDA. Because audio chip, TV decoder chip and HSMC share one I2C bus, therefore audio and video chip won't function correctly.

The voltage level of the I/O pins on the HSMC connector can be adjusted to 3.3V, 2.5V, 1.8V, or 1.5V using JP7 (The default setting is 2.5V, see Figure 4-13). Because the HSMC I/Os are connected to Bank 5 & 6 of the FPGA and the VCCIO voltage (VCCIO5 & VCCIO6) of these banks are controlled by the header JP7, users can use a jumper to select the input voltage of VCCIO5 & VCCIO6 to 3.3V, 2.5V, 1.8V, and 1.5V to control the voltage level of the I/O pins. Table 4-8 lists the jumper settings of the JP7.

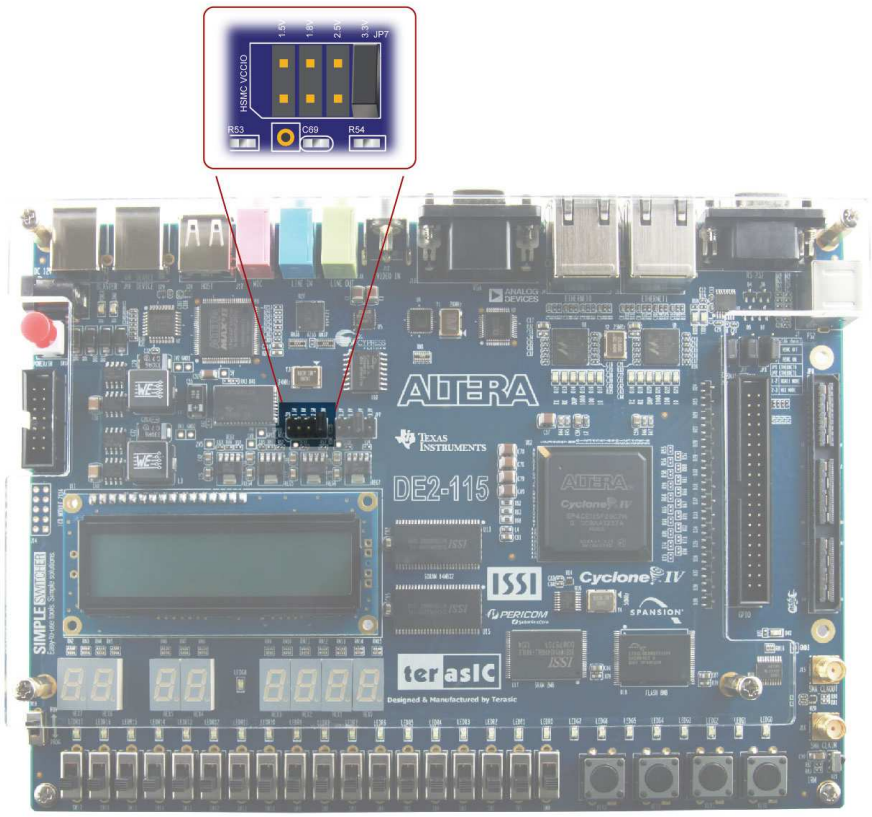


Figure 4-13 HSMC VCCIO supply voltage setting header

Table 4-8 Jumper Settings for different I/O Standards

JP7 Jumper Settings	*Supplied Voltage to VCCIO5 & VCCIO6*	*IO Voltage of HSMC Connector (JP8)*
Short Pins 1 and 2	**1.5V**	**1.5V**
Short Pins 3 and 4	**1.8V**	**1.8V**
Short Pins 5 and 6	**2.5V**	**2.5V (Default)**
Short Pins 7 and 8	**3.3V**	**3.3V**

Note: Users that connect a daughter card on the HSMC connector need to pay close attention on the I/O standard between DE2-115 HSMC connector pins and daughter card system. For example, if the I/O standard of HSMC pins on DE2-115 board is set to 1.8V, a daughter card with 3.3V I/O standard may not work properly on DE2-115 board due to I/O standard mismatch.

Additionally, when LVDS is used as the I/O standard of the HSMC connector, the LVDS receivers need to assemble a 100 Ohm resistor between two input signals for each pairs as shown in **Figure 4-14. Table 4-9** shows all the pin assignments of the HSMC connector.

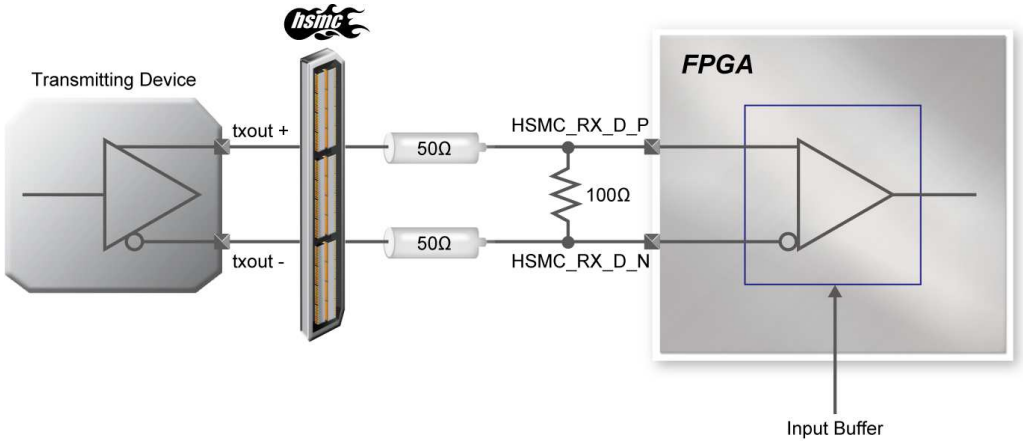


Figure 4-14 LVDS interface on HSMC connector and Cyclone IV E FPGA

Table 4-9 Pin Assignments for HSMC connector

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
HSMC_CLKIN0	PIN_AH15	Dedicated clock input	Depending on JP6
HSMC_CLKIN_N1	PIN_J28	LVDS RX or CMOS input or differential clock input	Depending on JP7
HSMC_CLKIN_N2	PIN_Y28	LVDS RX or CMOS input or differential clock input	Depending on JP7
HSMC_CLKIN_P1	PIN_J27	LVDS RX or CMOS input or differential clock input	Depending on JP7
HSMC_CLKIN_P2	PIN_Y27	LVDS RX or CMOS input or differential clock input	Depending on JP7
HSMC_CLKOUT0	PIN_AD28	Dedicated clock output	Depending on JP7
HSMC_CLKOUT_N1	PIN_G24	LVDS TX or CMOS I/O or differential clock output	Depending on JP7
HSMC_CLKOUT_N2	PIN_V24	LVDS TX or CMOS I/O or differential clock output	Depending on JP7
HSMC_CLKOUT_P1	PIN_G23	LVDS TX or CMOS I/O or differential clock output	Depending on JP7
HSMC_CLKOUT_P2	PIN_V23	LVDS TX or CMOS I/O or differential clock output	Depending on JP7
HSMC_D[0]	PIN_AE26	LVDS TX or CMOS I/O	Depending on JP7
HSMC_D[1]	PIN_AE28	LVDS RX or CMOS I/O	Depending on JP7
HSMC_D[2]	PIN_AE27	LVDS TX or CMOS I/O	Depending on JP7
HSMC_D[3]	PIN_AF27	LVDS RX or CMOS I/O	Depending on JP7

HSMC_RX_D_N[0]	PIN_F25	LVDS RX bit 0n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[1]	PIN_C27	LVDS RX bit 1n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[2]	PIN_E26	LVDS RX bit 2n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[3]	PIN_G26	LVDS RX bit 3n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[4]	PIN_H26	LVDS RX bit 4n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[5]	PIN_K26	LVDS RX bit 5n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[6]	PIN_L24	LVDS RX bit 6n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[7]	PIN_M26	LVDS RX bit 7n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[8]	PIN_R26	LVDS RX bit 8n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[9]	PIN_T26	LVDS RX bit 9n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[10]	PIN_U26	LVDS RX bit 10n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[11]	PIN_L22	LVDS RX bit 11n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[12]	PIN_N26	LVDS RX bit 12n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[13]	PIN_P26	LVDS RX bit 13n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[14]	PIN_R21	LVDS RX bit 14n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[15]	PIN_R23	LVDS RX bit 15n or CMOS I/O	Depending on JP7
HSMC_RX_D_N[16]	PIN_T22	LVDS RX bit 16n or CMOS I/O	Depending on JP7
HSMC_RX_D_P[0]	PIN_F24	LVDS RX bit 0 or CMOS I/O	Depending on JP7
HSMC_RX_D_P[1]	PIN_D26	LVDS RX bit 1 or CMOS I/O	Depending on JP7
HSMC_RX_D_P[2]	PIN_F26	LVDS RX bit 2 or CMOS I/O	Depending on JP7
HSMC_RX_D_P[3]	PIN_G25	LVDS RX bit 3 or CMOS I/O	Depending on JP7
HSMC_RX_D_P[4]	PIN_H25	LVDS RX bit 4 or CMOS I/O	Depending on JP7
HSMC_RX_D_P[5]	PIN_K25	LVDS RX bit 5 or CMOS I/O	Depending on JP7

HSMC_RX_D_P[6]	PIN_L23	LVDS RX bit 6 or CMOS I/O	Depending on JP7
HSMC_RX_D_P[7]	PIN_M25	LVDS RX bit 7 or CMOS I/O	Depending on JP7
HSMC_RX_D_P[8]	PIN_R25	LVDS RX bit 8 or CMOS I/O	Depending on JP7
HSMC_RX_D_P[9]	PIN_T25	LVDS RX bit 9 or CMOS I/O	Depending on JP7
HSMC_RX_D_P[10]	PIN_U25	LVDS RX bit 10 or CMOS I/O	Depending on JP7
HSMC_RX_D_P[11]	PIN_L21	LVDS RX bit 11 or CMOS I/O	Depending on JP7
HSMC_RX_D_P[12]	PIN_N25	LVDS RX bit 12 or CMOS I/O	Depending on JP7
HSMC_RX_D_P[13]	PIN_P25	LVDS RX bit 13 or CMOS I/O	Depending on JP7
HSMC_RX_D_P[14]	PIN_P21	LVDS RX bit 14 or CMOS I/O	Depending on JP7
HSMC_RX_D_P[15]	PIN_R22	LVDS RX bit 15 or CMOS I/O	Depending on JP7
HSMC_RX_D_P[16]	PIN_T21	LVDS RX bit 16 or CMOS I/O	Depending on JP7
HSMC_TX_D_N[0]	PIN_D28	LVDS TX bit 0n or CMOS I/O	Depending on JP7
HSMC_TX_D_N[1]	PIN_E28	LVDS TX bit 1n or CMOS I/O	Depending on JP7
HSMC_TX_D_N[2]	PIN_F28	LVDS TX bit 2n or CMOS I/O	Depending on JP7
HSMC_TX_D_N[3]	PIN_G28	LVDS TX bit 3n or CMOS I/O	Depending on JP7
HSMC_TX_D_N[4]	PIN_K28	LVDS TX bit 4n or CMOS I/O	Depending on JP7
HSMC_TX_D_N[5]	PIN_M28	LVDS TX bit 5n or CMOS I/O	Depending on JP7
HSMC_TX_D_N[6]	PIN_K22	LVDS TX bit 6n or CMOS I/O	Depending on JP7
HSMC_TX_D_N[7]	PIN_H24	LVDS TX bit 7n or CMOS I/O	Depending on JP7
HSMC_TX_D_N[8]	PIN_J24	LVDS TX bit 8n or CMOS I/O	Depending on JP7
HSMC_TX_D_N[9]	PIN_P28	LVDS TX bit 9n or CMOS I/O	Depending on JP7
HSMC_TX_D_N[10]	PIN_J26	LVDS TX bit 10n or CMOS I/O	Depending on JP7
HSMC_TX_D_N[11]	PIN_L28	LVDS TX bit 11n or CMOS I/O	Depending on JP7

HSMC_TX_D_N[12]	PIN_V26	LVDS TX bit 12n or CMOS I/O	Depending on JP7
HSMC_TX_D_N[13]	PIN_R28	LVDS TX bit 13n or CMOS I/O	Depending on JP7
HSMC_TX_D_N[14]	PIN_U28	LVDS TX bit 14n or CMOS I/O	Depending on JP7
HSMC_TX_D_N[15]	PIN_V28	LVDS TX bit 15n or CMOS I/O	Depending on JP7
HSMC_TX_D_N[16]	PIN_V22	LVDS TX bit 16n or CMOS I/O	Depending on JP7
HSMC_TX_D_P[0]	PIN_D27	LVDS TX bit 0 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[1]	PIN_E27	LVDS TX bit 1 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[2]	PIN_F27	LVDS TX bit 2 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[3]	PIN_G27	LVDS TX bit 3 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[4]	PIN_K27	LVDS TX bit 4 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[5]	PIN_M27	LVDS TX bit 5 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[6]	PIN_K21	LVDS TX bit 6 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[7]	PIN_H23	LVDS TX bit 7 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[8]	PIN_J23	LVDS TX bit 8 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[9]	PIN_P27	LVDS TX bit 9 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[10]	PIN_J25	LVDS TX bit 10 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[11]	PIN_L27	LVDS TX bit 11 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[12]	PIN_V25	LVDS TX bit 12 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[13]	PIN_R27	LVDS TX bit 13 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[14]	PIN_U27	LVDS TX bit 14 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[15]	PIN_V27	LVDS TX bit 15 or CMOS I/O	Depending on JP7
HSMC_TX_D_P[16]	PIN_U22	LVDS TX bit 16 or CMOS I/O	Depending on JP7

4.8 Using the Expansion Header

The DE2-115 Board provides one 40-pin expansion header. The header connects directly to 36 pins of the Cyclone IV E FPGA, and also provides DC +5V (VCC5), DC +3.3V (VCC3P3), and two GND pins. Figure 4-15 shows the I/O distribution of the GPIO connector. The maximum power consumption of the daughter card that connects to GPIO port is shown in Table 4-10.

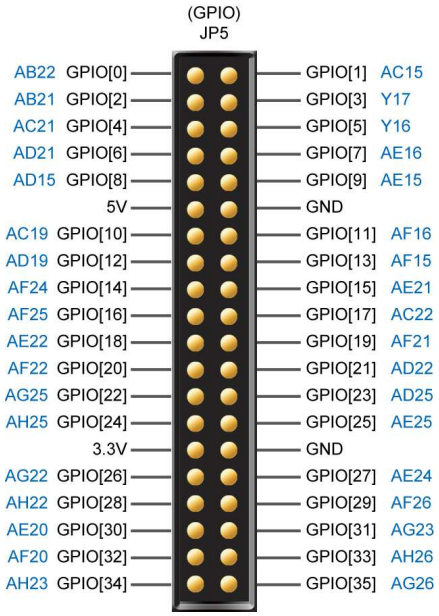


Figure 4-15 GPIO Pin Arrangement

Table 4-10 Power Supply of the Expansion Header

Supplied Voltage	*Max. Current Limit*
5V	1A
3.3V	1.5A

Each pin on the expansion headers is connected to two diodes and a resistor that provides protection against high and low voltages. Figure 4-16 shows the protection circuitry for only one of the pin on the header, but this circuitry is included for all 36 data pins.

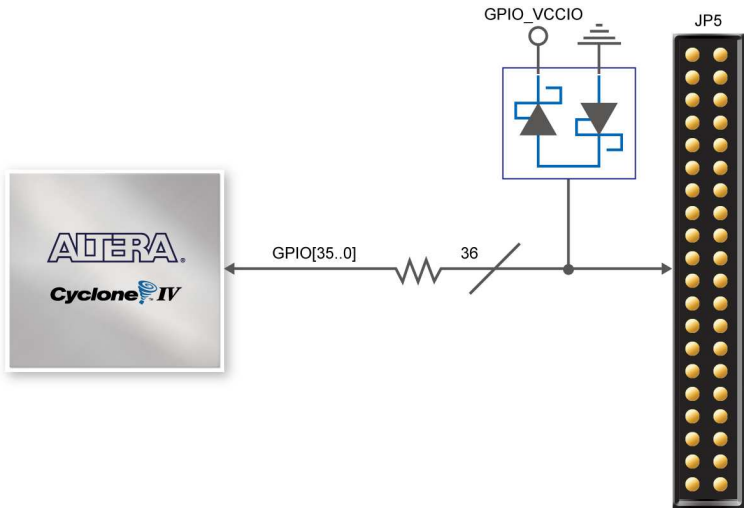


Figure 4-16 Connections between the GPIO connector and Cyclone IV E FPGA

The voltage level of the I/O pins on the expansion headers can be adjusted to 3.3V, 2.5V, 1.8V, or 1.5V using JP6 (The default value is 3.3V, see **Figure 4-17**). Because the expansion I/Os are connected to Bank 4 of the FPGA and the VCCIO voltage (VCCIO4) of this bank is controlled by the header JP6, users can use a jumper to select the input voltage of VCCIO4 to 3.3V, 2.5V, 1.8V, and 1.5V to control the voltage level of the I/O pins. **Table 4-11** lists the jumper settings of the JP6. The pin-outs of the JP6 appear in **Figure 4-17**.

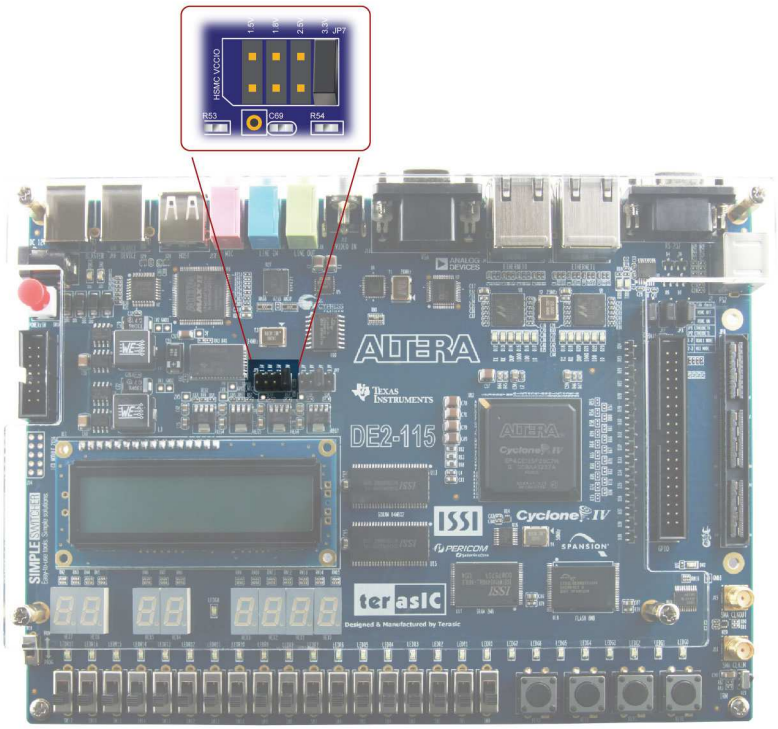


Figure 4-17 GPIO VCCIO supply voltage setting header

Table 4-11 Voltage Level Setting of the Expansion Headers Using JP6

JP6 Jumper Settings	Supplied Voltage to VCCIO4	IO Voltage of Expansion Headers (JP5)
Short Pins 1 and 2	1.5V	1.5V
Short Pins 3 and 4	1.8V	1.8V
Short Pins 5 and 6	2.5V	2.5V
Short Pins 7 and 8	3.3V	3.3V (Default)

Note : Users who want to use daughter card on GPIO connector need to pay close attention to the I/O standard between DE2-115 GPIO connector pins and daughter card system. For example, if the I/O standard of GPIO pins on DE2-115 board is set to 1.8V, a daughter card with 3.3V I/O standard may not work properly on the DE2-115 board due to I/O standard mismatch.

Figure 4-18 depicts the pin definition on the expansion connector for using these I/Os as LVDS transmitters. Due to the reason that the column I/Os of the FPGA the expansion pins connecting with can only support emulated LVDS transmitters, two single-ended output buffers and external resistors must be used as shown in Figure 4-19. The associated I/O standard of these differential FPGA I/O pins on Quartus II project should set to LVDS_E_3R.

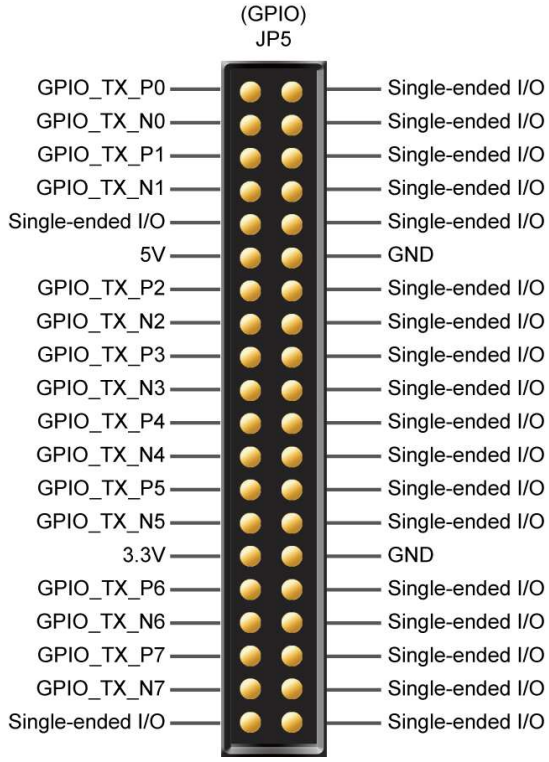


Figure 4-18 Pin defined when using LVDS interface on GPIO FPGA pins

The factory default setting on Rs resistor will be 47 ohm and Rp resistor will not be assembled for single-ended I/O standard application. For LVDS transmitter application, please assemble 120 and 170 ohm resistor on Rs and Rp position, respectively.

Finally, **Table 4-12** shows all the pin assignments of the GPIO connector.

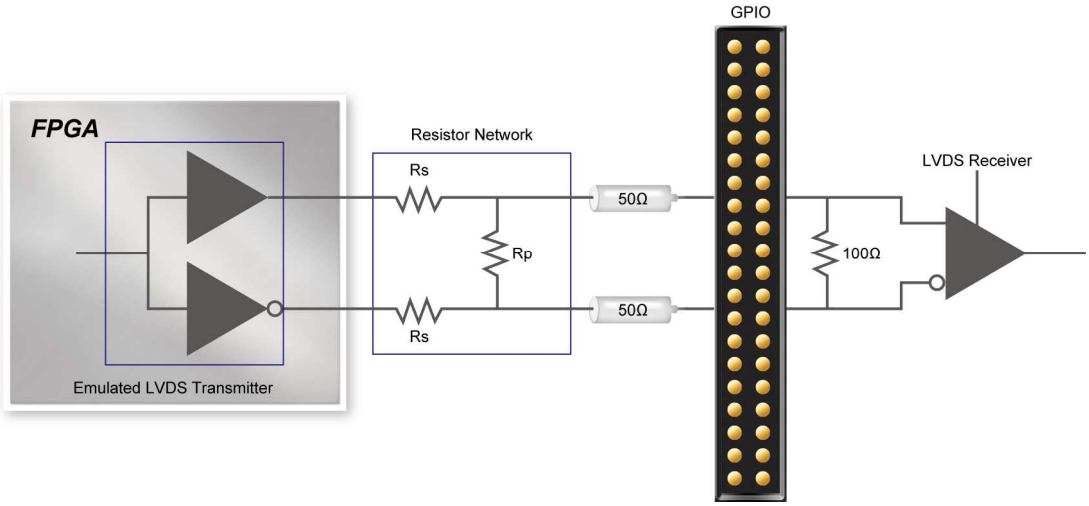


Figure 4-19 Using Emulated LVDS on GPIO

Table 4-12 Pin Assignments for Expansion Headers

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
GPIO[0]	PIN_AB22	GPIO Connection DATA[0]	Depending on JP6
GPIO[1]	PIN_AC15	GPIO Connection DATA[1]	Depending on JP6
GPIO[2]	PIN_AB21	GPIO Connection DATA[2]	Depending on JP6
GPIO[3]	PIN_Y17	GPIO Connection DATA[3]	Depending on JP6
GPIO[4]	PIN_AC21	GPIO Connection DATA[4]	Depending on JP6
GPIO[5]	PIN_Y16	GPIO Connection DATA[5]	Depending on JP6
GPIO[6]	PIN_AD21	GPIO Connection DATA[6]	Depending on JP6
GPIO[7]	PIN_AE16	GPIO Connection DATA[7]	Depending on JP6
GPIO[8]	PIN_AD15	GPIO Connection DATA[8]	Depending on JP6
GPIO[9]	PIN_AE15	GPIO Connection DATA[9]	Depending on JP6
GPIO[10]	PIN_AC19	GPIO Connection DATA[10]	Depending on JP6
GPIO[11]	PIN_AF16	GPIO Connection DATA[11]	Depending on JP6
GPIO[12]	PIN_AD19	GPIO Connection DATA[12]	Depending on JP6
GPIO[13]	PIN_AF15	GPIO Connection DATA[13]	Depending on JP6
GPIO[14]	PIN_AF24	GPIO Connection DATA[14]	Depending on JP6
GPIO[15]	PIN_AE21	GPIO Connection DATA[15]	Depending on JP6
GPIO[16]	PIN_AF25	GPIO Connection DATA[16]	Depending on JP6
GPIO[17]	PIN_AC22	GPIO Connection DATA[17]	Depending on JP6
GPIO[18]	PIN_AE22	GPIO Connection DATA[18]	Depending on JP6
GPIO[19]	PIN_AF21	GPIO Connection DATA[19]	Depending on JP6
GPIO[20]	PIN_AF22	GPIO Connection DATA[20]	Depending on JP6
GPIO[21]	PIN_AD22	GPIO Connection DATA[21]	Depending on JP6

GPIO[22]	PIN_AG25	GPIO Connection DATA[22]	Depending on JP6
GPIO[23]	PIN_AD25	GPIO Connection DATA[23]	Depending on JP6
GPIO[24]	PIN_AH25	GPIO Connection DATA[24]	Depending on JP6
GPIO[25]	PIN_AE25	GPIO Connection DATA[25]	Depending on JP6
GPIO[26]	PIN_AG22	GPIO Connection DATA[26]	Depending on JP6
GPIO[27]	PIN_AE24	GPIO Connection DATA[27]	Depending on JP6
GPIO[28]	PIN_AH22	GPIO Connection DATA[28]	Depending on JP6
GPIO[29]	PIN_AF26	GPIO Connection DATA[29]	Depending on JP6
GPIO[30]	PIN_AE20	GPIO Connection DATA[30]	Depending on JP6
GPIO[31]	PIN_AG23	GPIO Connection DATA[31]	Depending on JP6
GPIO[32]	PIN_AF20	GPIO Connection DATA[32]	Depending on JP6
GPIO[33]	PIN_AH26	GPIO Connection DATA[33]	Depending on JP6
GPIO[34]	PIN_AH23	GPIO Connection DATA[34]	Depending on JP6
GPIO[35]	PIN_AG26	GPIO Connection DATA[35]	Depending on JP6

4.9 Using 14-pin General Purpose I/O Connector

The DE2-115 Board provides 14-pin expansion header. The header connects directly to 7 pins of the Cyclone IV E FPGA, and also provides DC +3.3V (VCC3P3), and six GND pins as shown in Figure 4-20. The voltage level of the I/O pins on the 14-pin expansion header is 3.3V. Finally, Table 4-13 shows the pin assignments for I/O connections.

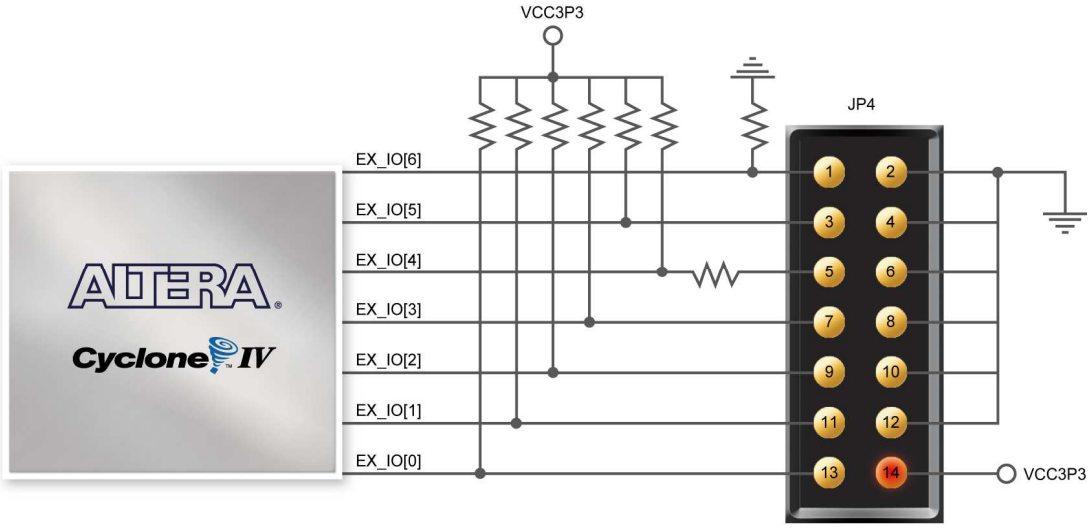


Figure 4-20 Connections between FPGA and 14-pin general purpose I/O

Table 4-13 Pin Assignments for General Purpose I/Os

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
EX_IO[0]	PIN_J10	Extended IO[0]	3.3V
EX_IO[1]	PIN_J14	Extended IO[1]	3.3V
EX_IO[2]	PIN_H13	Extended IO[2]	3.3V

EX_IO[3]	PIN_H14	Extended IO[3]	3.3V
EX_IO[4]	PIN_F14	Extended IO[4]	3.3V
EX_IO[5]	PIN_E10	Extended IO[5]	3.3V
EX_IO[6]	PIN_D9	Extended IO[6]	3.3V

4.10 Using VGA

The DE2-115 board includes a 15-pin D-SUB connector for VGA output. The VGA synchronization signals are provided directly from the Cyclone IV E FPGA, and the Analog Devices ADV7123 triple 10-bit high-speed video DAC (only the higher 8-bits are used) is used to produce the analog data signals (red, green, and blue). It could support the SXGA standard (1280*1024) with a bandwidth of 100MHz. Figure 4-21 gives the associated schematic.

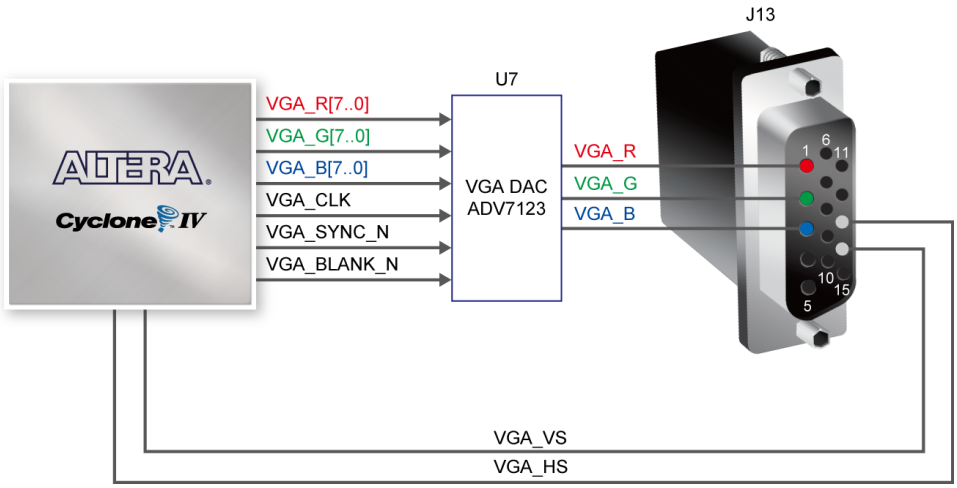


Figure 4-21 Connections between FPGA and VGA

The timing specification for VGA synchronization and RGB (red, green, blue) data can be found on various educational website (for example, search for "VGA signal timing"). Figure 4-22 illustrates the basic timing requirements for each row (horizontal) that is displayed on a VGA monitor. An active-low pulse of specific duration (time (a) in the figure) is applied to the horizontal synchronization (hsync) input of the monitor, which signifies the end of one row of data and the start of the next. The data (RGB) output to the monitor must be off (driven to 0 V) for a time period called the back porch (b) after the hsync pulse occurs, which is followed by the display interval (c). During the data display interval the RGB data drives each pixel in turn across the row being displayed. Finally, there is a time period called the front porch (d) where the RGB signals must again be off before the next hsync pulse can occur. The timing of the vertical synchronization (vsync) is the similar as shown in Figure 4-22, except that a vsync pulse signifies the end of one frame and the start of the next, and the data refers to the set of rows in the frame (horizontal timing). Table 4-14 and Table 4-15 show different resolutions and durations of time periods a, b, c, and d for both horizontal and vertical timing.

Detailed information for using the ADV7123 video DAC is available in its datasheet, which can be found on the manufacturer's website, or in the DE2_115_datasheets\VIDEO-DAC folder on the DE2-115 System CD. The pin assignments between the Cyclone IV E FPGA and the ADV7123 are listed in **Table 4-16**. An example of code that drives a VGA display is described in Sections 6.2 and 6.3.

Note: The RGB data bus on DE2-115 board is 8 bit instead of 10 bit on DE2/DE2-70 board.

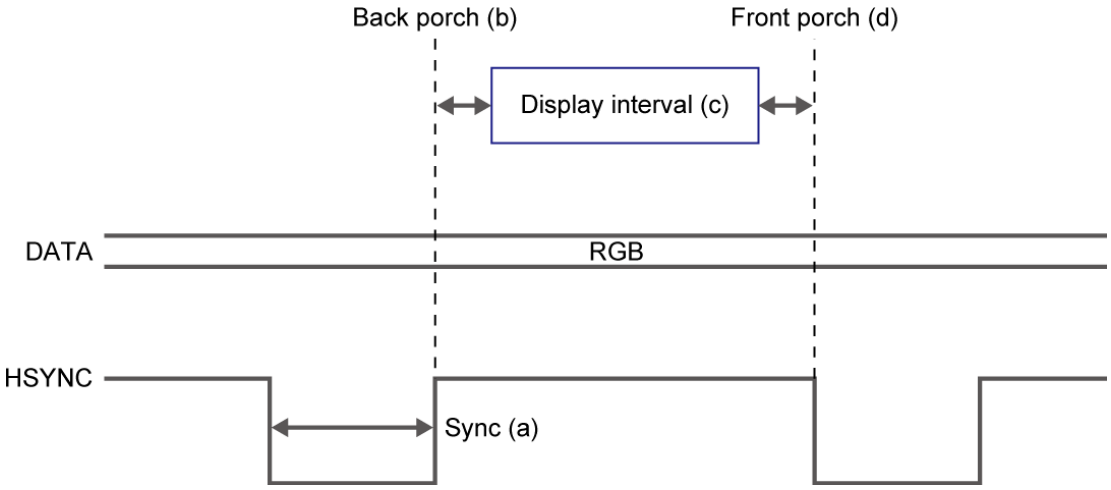


Figure 4-22 VGA horizontal timing specification

Table 4-14 VGA Horizontal Timing Specification

VGA mode		*Horizontal Timing Spec*				
Configuration	*Resolution(HxV)*	*a(us)*	*b(us)*	*c(us)*	*d(us)*	*Pixel clock(MHz)*
VGA(60Hz)	640x480	3.8	1.9	25.4	0.6	25
VGA(85Hz)	640x480	1.6	2.2	17.8	1.6	36
SVGA(60Hz)	800x600	3.2	2.2	20	1	40
SVGA(75Hz)	800x600	1.6	3.2	16.2	0.3	49
SVGA(85Hz)	800x600	1.1	2.7	14.2	0.6	56
XGA(60Hz)	1024x768	2.1	2.5	15.8	0.4	65
XGA(70Hz)	1024x768	1.8	1.9	13.7	0.3	75
XGA(85Hz)	1024x768	1.0	2.2	10.8	0.5	95
1280x1024(60Hz)	1280x1024	1.0	2.3	11.9	0.4	108

Table 4-15 VGA Vertical Timing Specification

VGA mode		*Vertical Timing Spec*				
Configuration	*Resolution(HxV)*	*a(lines)*	*b(lines)*	*c(lines)*	*d(lines)*	*Pixel clock(MHz)*
VGA(60Hz)	640x480	2	33	480	10	25
VGA(85Hz)	640x480	3	25	480	1	36

SVGA(60Hz)	800x600	4	23	600	1	40
SVGA(75Hz)	800x600	3	21	600	1	49
SVGA(85Hz)	800x600	3	27	600	1	56
XGA(60Hz)	1024x768	6	29	768	3	65
XGA(70Hz)	1024x768	6	29	768	3	75
XGA(85Hz)	1024x768	3	36	768	1	95
1280x1024(60Hz)	1280x1024	3	38	1024	1	108

Table 4-16 Pin Assignments for ADV7123

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
VGA_R[0]	PIN_E12	VGA Red[0]	3.3V
VGA_R[1]	PIN_E11	VGA Red[1]	3.3V
VGA_R[2]	PIN_D10	VGA Red[2]	3.3V
VGA_R[3]	PIN_F12	VGA Red[3]	3.3V
VGA_R[4]	PIN_G10	VGA Red[4]	3.3V
VGA_R[5]	PIN_J12	VGA Red[5]	3.3V
VGA_R[6]	PIN_H8	VGA Red[6]	3.3V
VGA_R[7]	PIN_H10	VGA Red[7]	3.3V
VGA_G[0]	PIN_G8	VGA Green[0]	3.3V
VGA_G[1]	PIN_G11	VGA Green[1]	3.3V
VGA_G[2]	PIN_F8	VGA Green[2]	3.3V
VGA_G[3]	PIN_H12	VGA Green[3]	3.3V
VGA_G[4]	PIN_C8	VGA Green[4]	3.3V
VGA_G[5]	PIN_B8	VGA Green[5]	3.3V
VGA_G[6]	PIN_F10	VGA Green[6]	3.3V
VGA_G[7]	PIN_C9	VGA Green[7]	3.3V
VGA_B[0]	PIN_B10	VGA Blue[0]	3.3V
VGA_B[1]	PIN_A10	VGA Blue[1]	3.3V
VGA_B[2]	PIN_C11	VGA Blue[2]	3.3V
VGA_B[3]	PIN_B11	VGA Blue[3]	3.3V
VGA_B[4]	PIN_A11	VGA Blue[4]	3.3V
VGA_B[5]	PIN_C12	VGA Blue[5]	3.3V
VGA_B[6]	PIN_D11	VGA Blue[6]	3.3V
VGA_B[7]	PIN_D12	VGA Blue[7]	3.3V
VGA_CLK	PIN_A12	VGA Clock	3.3V
VGA_BLANK_N	PIN_F11	VGA BLANK	3.3V
VGA_HS	PIN_G13	VGA H_SYNC	3.3V
VGA_VS	PIN_C13	VGA V_SYNC	3.3V
VGA_SYNC_N	PIN_C10	VGA SYNC	3.3V

4.11 Using the 24-bit Audio CODEC

The DE2-115 board provides high-quality 24-bit audio via the Wolfson WM8731 audio CODEC

(Encoder/Decoder). This chip supports microphone-in, line-in, and line-out ports, with a sample rate adjustable from 8 kHz to 96 kHz. The WM8731 is controlled via serial I2C bus interface*, which is connected to pins on the Cyclone IV E FPGA. A schematic diagram of the audio circuitry is shown in Figure 4-23, and the FPGA pin assignments are listed in Table 4-17. Detailed information for using the WM8731 codec is available in its datasheet, which can be found on the manufacturer's website, or in the *DE2_115_datasheets\Audio CODEC* folder on the DE2-115 System CD.

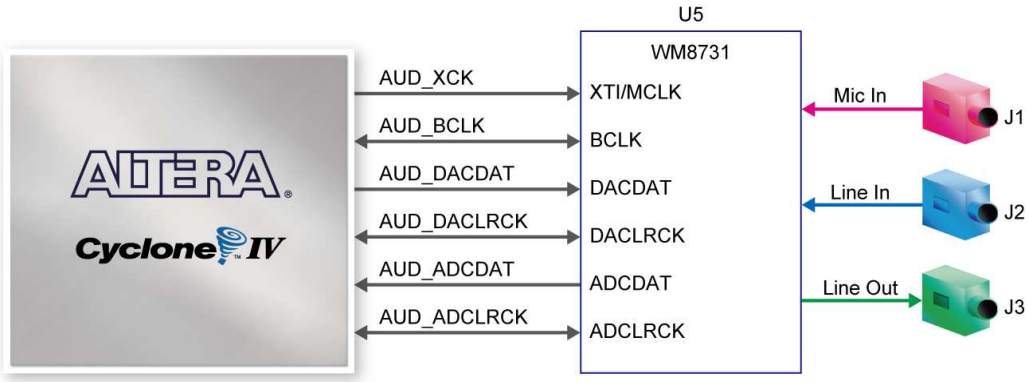


Figure 4-23 Connections between FPGA and Audio CODEC

Table 4-17 Audio CODEC Pin Assignments

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
AUD_ADCLRCK	PIN_C2	Audio CODEC ADC LR Clock	3.3V
AUD_ADCDAT	PIN_D2	Audio CODEC ADC Data	3.3V
AUD_DACLRCK	PIN_E3	Audio CODEC DAC LR Clock	3.3V
AUD_DACDAT	PIN_D1	Audio CODEC DAC Data	3.3V
AUD_XCK	PIN_E1	Audio CODEC Chip Clock	3.3V
AUD_BCLK	PIN_F2	Audio CODEC Bit-Stream Clock	3.3V
I2C_SCLK	PIN_B7	I2C Clock	3.3V
I2C_SDAT	PIN_A8	I2C Data	3.3V

Note: If the HSMC loopback adapter is mounted, the I2C_SCL will be directly routed back to I2C_SDA. Because audio chip, TV decoder chip and HSMC share one I2C bus, therefore audio and video chip won't function correctly.

4.12 RS-232 Serial Port

The DE2-115 board uses the ZT3232 transceiver chip and a 9-pin DB9 connector for RS-232 communications. For detailed information on how to use the transceiver, please refer to the datasheet, which is available on the manufacturer's website, or in the *DE2_115_datasheets\RS-232* folder on the DE2-115 System CD. Figure 4-24 shows the related schematics, and Table 4-18 lists the Cyclone IV E FPGA pin assignments.

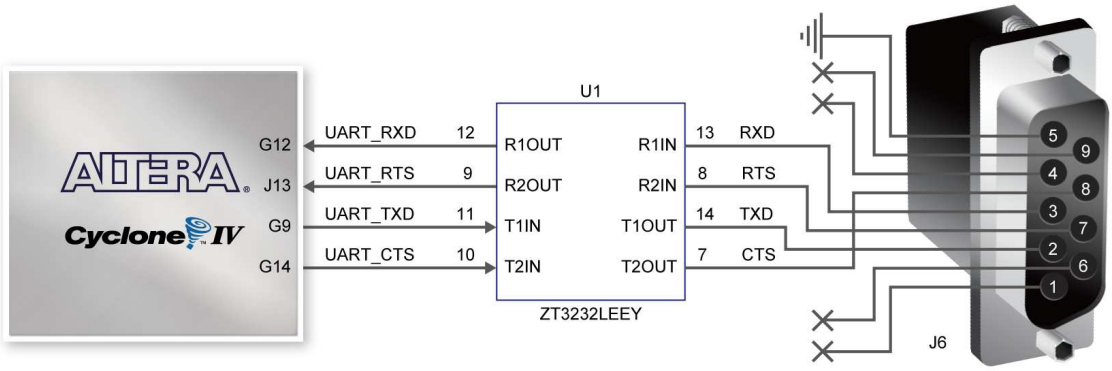


Figure 4-24 Connections between FPGA and ZT3232 (RS-232) chip

Table 4-18 RS-232 Pin Assignments

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
UART_RXD	PIN_G12	UART Receiver	3.3V
UART_TXD	PIN_G9	UART Transmitter	3.3V
UART_CTS	PIN_G14	UART Clear to Send	3.3V
UART_RTS	PIN_J13	UART Request to Send	3.3V

4.13 PS/2 Serial Port

The DE2-115 board includes a standard PS/2 interface and a connector for a PS/2 keyboard or mouse. Figure 4-25 shows the schematic of the PS/2 circuit. In addition, users can use the PS/2 keyboard and mouse on the DE2-115 board simultaneously by plugging an extension PS/2 Y-Cable (See Figure 4-26). Instructions for using a PS/2 mouse or keyboard can be found by performing an appropriate search on various educational websites. The pin assignments for the associated interface are shown in Table 4-19.

Note: If users connect only one PS/2 equipment, the PS/2 interface between FPGA I/O should be "PS2_CLK" and "PS2_DAT".

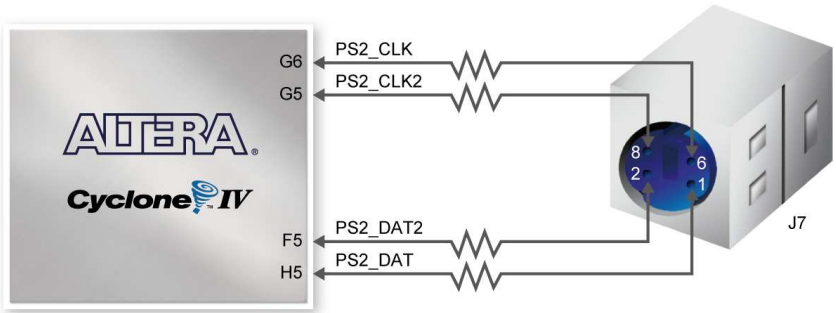


Figure 4-25 Connection between FPGA and PS/2



Figure 4-26 Y-Cable use for both Keyboard and Mouse

Table 4-19 PS/2 Pin Assignments

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
PS2_CLK	PIN_G6	PS/2 Clock	3.3V
PS2_DAT	PIN_H5	PS/2 Data	3.3V
PS2_CLK2	PIN_G5	PS/2 Clock (reserved for second PS/2 device)	3.3V
PS2_DAT2	PIN_F5	PS/2 Data (reserved for second PS/2 device)	3.3V

4.14 Gigabit Ethernet Transceiver

The DE2-115 board provides Ethernet support via two Marvell 88E1111 Ethernet PHY chips. The 88E1111 chip with integrated 10/100/1000 Mbps Gigabit Ethernet transceiver support GMII/MII/RGMII/TBI MAC interfaces. Table 4-20 describes the default settings for both chips. Figure 4-27 shows the connection setup between the Gigabit Ethernet PHY (ENET0) and FPGA.

Table 4-20 Default Configuration for Gigabit Ethernet

Configuration	*Description*	*Default Value*
PHYADDR[4:0]	PHY Address in MDIO/MDC Mode	10000 for Enet0;10001 for Enet1
ENA_PAUSE	Enable Pause	1-Default Register 4.11:10 to 11
ANEG[3:0]	Auto negotiation configuration for copper modes	1110-Auto-neg, advertise all capabilities, prefer master
ENA_XC	Enable Crossover	0-Disable
DIS_125	Disable 125MHz clock	1-Disable 125CLK
HWCFG[3:0]	Hardware Configuration Mode	1011/1111 RGMII to copper/GMII to copper
DIS_FC	Disable fiber/copper interface	1-Disable
DIS_SLEEP	Energy detect	1-Disable energy detect
SEL_TWSI	Interface select	0-Select MDC/MDIO interface
INT_POL	Interrupt polarity	1-INTn signal is active LOW
75/50OHM	Termination resistance	0-50 ohm termination for fiber

Here only RGMII and MII modes are supported on the board (The factory default mode is RGMII).

There is one jumper for each chip for switching work modes from RGMII to MII (See **Figure 4-28**). You will need to perform a hardware reset after any change for enabling new settings. **Table 4-21** and **Table 4-22** describe the working mode settings for ENET0 PHY (U8) and ENET1 PHY (U9) respectively.

In addition, it is dynamically configurable to support 10Mbps, 100Mbps (Fast Ethernet) or 1000Mbps (Gigabit Ethernet) operation using standard Cat 5e UTP cabling. The associated pin assignments are listed in **Table 4-23**. For detailed information on how to use the 88E1111 refers to its datasheet and application notes, which are available on the manufacturer's website.

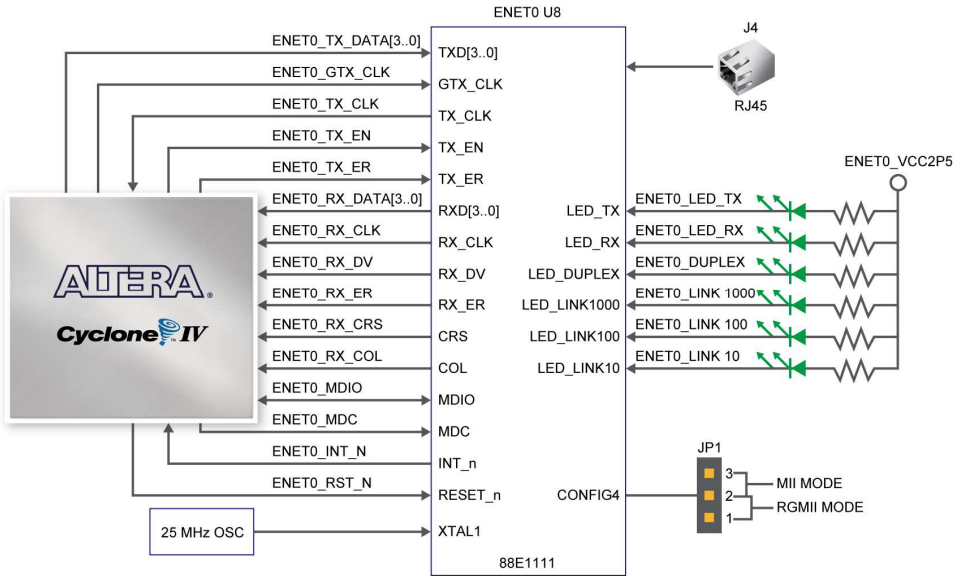


Figure 4-27 Connections between FPGA and Ethernet

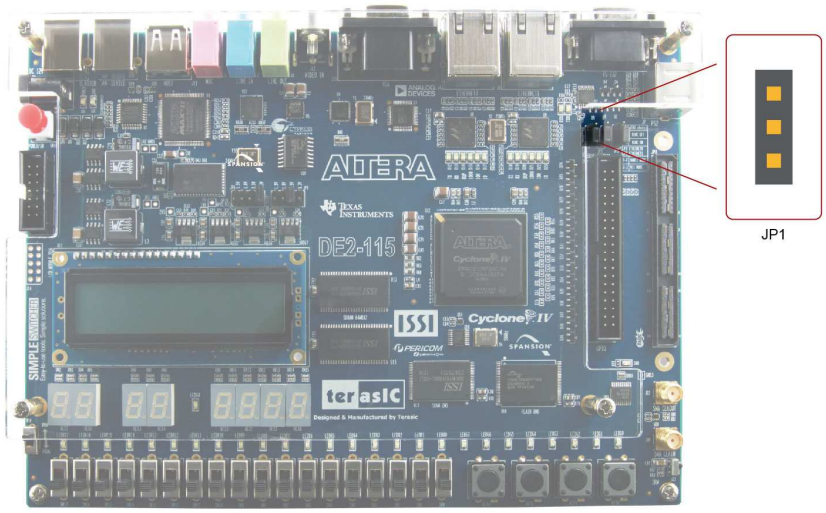


Figure 4-28 Working mode setup header for Ethernet PHY

Table 4-21 Jumper Settings for Working Mode of ENET0 (U8)

JP1 Jumper Settings	*ENET0 PHY Working Mode*
Short Pins 1 and 2	RGMII Mode
Short Pins 2 and 3	MII Mode

Table 4-22 Jumper Settings for Working Mode of ENET1 (U9)

JP2 Jumper Settings	*ENET1 PHY Working Mode*
Short Pins 1 and 2	RGMII Mode
Short Pins 2 and 3	MII Mode

Table 4-23 Pin Assignments for Fast Ethernet

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
ENET0_GTX_CLK	PIN_A17	GMII Transmit Clock 1	2.5V
ENET0_INT_N	PIN_A21	Interrupt open drain output 1	2.5V
ENET0_LINK100	PIN_C14	Parallel LED output of 100BASE-TX link 1	3.3V
ENET0_MDC	PIN_C20	Management data clock reference 1	2.5V
ENET0_MDIO	PIN_B21	Management data 1	2.5V
ENET0_RST_N	PIN_C19	Hardware reset signal 1	2.5V
ENET0_RX_CLK	PIN_A15	GMII and MII receive clock 1	2.5V
ENET0_RX_COL	PIN_E15	GMII and MII collision 1	2.5V
ENET0_RX_CRS	PIN_D15	GMII and MII carrier sense 1	2.5V
ENET0_RX_DATA[0]	PIN_C16	GMII and MII receive data[0] 1	2.5V
ENET0_RX_DATA[1]	PIN_D16	GMII and MII receive data[1] 1	2.5V
ENET0_RX_DATA[2]	PIN_D17	GMII and MII receive data[2] 1	2.5V
ENET0_RX_DATA[3]	PIN_C15	GMII and MII receive data[3] 1	2.5V
ENET0_RX_DV	PIN_C17	GMII and MII receive data valid 1	2.5V
ENET0_RX_ER	PIN_D18	GMII and MII receive error 1	2.5V
ENET0_TX_CLK	PIN_B17	MII transmit clock 1	2.5V
ENET0_TX_DATA[0]	PIN_C18	MII transmit data[0] 1	2.5V
ENET0_TX_DATA[1]	PIN_D19	MII transmit data[1] 1	2.5V
ENET0_TX_DATA[2]	PIN_A19	MII transmit data[2] 1	2.5V
ENET0_TX_DATA[3]	PIN_B19	MII transmit data[3] 1	2.5V
ENET0_TX_EN	PIN_A18	GMII and MII transmit enable 1	2.5V
ENET0_TX_ER	PIN_B18	GMII and MII transmit error 1	2.5V
ENET1_GTX_CLK	PIN_C23	GMII Transmit Clock 2	2.5V
ENET1_INT_N	PIN_D24	Interrupt open drain output 2	2.5V
ENET1_LINK100	PIN_D13	Parallel LED output of 100BASE-TX link 2	3.3V
ENET1_MDC	PIN_D23	Management data clock reference 2	2.5V
ENET1_MDIO	PIN_D25	Management data 2	2.5V
ENET1_RST_N	PIN_D22	Hardware reset signal 2	2.5V
ENET1_RX_CLK	PIN_B15	GMII and MII receive clock 2	2.5V
ENET1_RX_COL	PIN_B22	GMII and MII collision 2	2.5V
ENET1_RX_CRS	PIN_D20	GMII and MII carrier sense 2	2.5V
ENET1_RX_DATA[0]	PIN_B23	GMII and MII receive data[0] 2	2.5V

ENET1_RX_DATA[1]	PIN_C21	GMII and MII receive data[1] 2	2.5V
ENET1_RX_DATA[2]	PIN_A23	GMII and MII receive data[2] 2	2.5V
ENET1_RX_DATA[3]	PIN_D21	GMII and MII receive data[3] 2	2.5V
ENET1_RX_DV	PIN_A22	GMII and MII receive data valid 2	2.5V
ENET1_RX_ER	PIN_C24	GMII and MII receive error 2	2.5V
ENET1_TX_CLK	PIN_C22	MII transmit clock 2	2.5V
ENET1_TX_DATA[0]	PIN_C25	MII transmit data[0] 2	2.5V
ENET1_TX_DATA[1]	PIN_A26	MII transmit data[1] 2	2.5V
ENET1_TX_DATA[2]	PIN_B26	MII transmit data[2] 2	2.5V
ENET1_TX_DATA[3]	PIN_C26	MII transmit data[3] 2	2.5V
ENET1_TX_EN	PIN_B25	GMII and MII transmit enable 2	2.5V
ENET1_TX_ER	PIN_A25	GMII and MII transmit error 2	2.5V
ENETCLK_25	PIN_A14	Ethernet clock source	3.3V

4.15 TV Decoder

The DE2-115 board is equipped with an Analog Device ADV7180 TV decoder chip. The ADV7180 is an integrated video decoder that automatically detects and converts a standard analog baseband television signals (NTSC, PAL, and SECAM) into 4:2:2 component video data compatible with the 8-bit ITU-R BT.656 interface standard. The ADV7180 is compatible with a broad range of video devices, including DVD players, tape-based sources, broadcast sources, and security/surveillance cameras.

The registers in the TV decoder can be programmed by a serial I2C bus, which is connected to the Cyclone IV E FPGA as indicated in **Figure 4-29**. Note that the I2C address W/R of the TV decoder (U6) is 0x40/0x41. The pin assignments are listed in **Table 4-24**. Detailed information of the ADV7180 is available on the manufacturer's website, or in the *DE2_115_datasheets\TV Decoder* folder on the DE2-115 System CD.

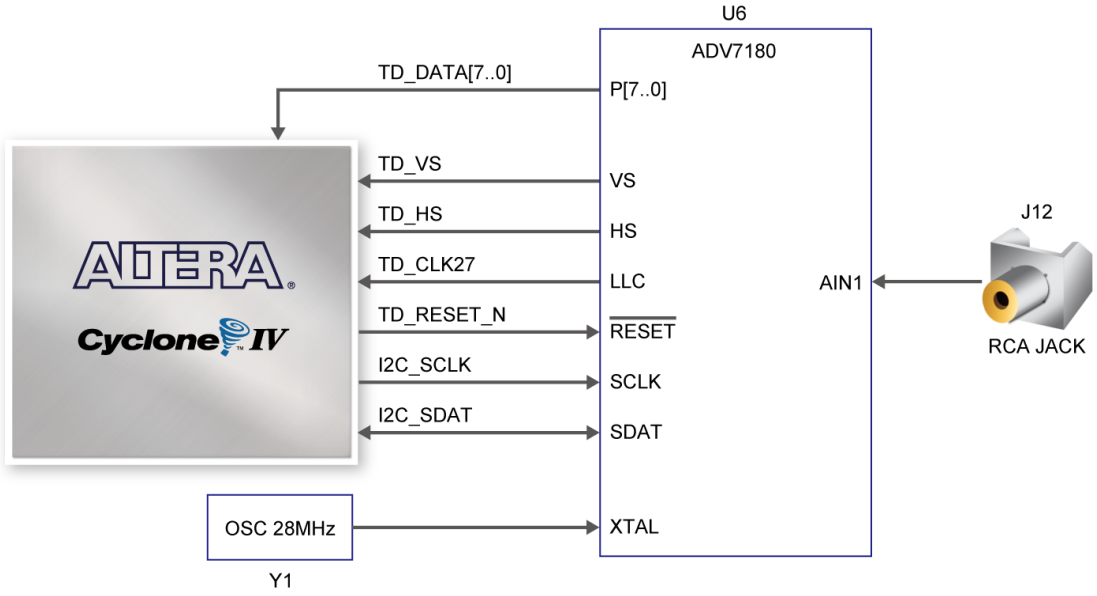


Figure 4-29 Connections between FPGA and TV Decoder

Note: If the HSMC loopback adapter is mounted, the I2C_SCL will be directly routed back to I2C_SDA. Because audio chip, TV decoder chip and HSMC share one I2C bus, therefore audio and video chip won't function correctly.

Table 4-24 TV Decoder Pin Assignments

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
TD_ DATA [0]	PIN_E8	TV Decoder Data[0]	3.3V
TD_ DATA [1]	PIN_A7	TV Decoder Data[1]	3.3V
TD_ DATA [2]	PIN_D8	TV Decoder Data[2]	3.3V
TD_ DATA [3]	PIN_C7	TV Decoder Data[3]	3.3V
TD_ DATA [4]	PIN_D7	TV Decoder Data[4]	3.3V
TD_ DATA [5]	PIN_D6	TV Decoder Data[5]	3.3V
TD_ DATA [6]	PIN_E7	TV Decoder Data[6]	3.3V
TD_ DATA [7]	PIN_F7	TV Decoder Data[7]	3.3V
TD_HS	PIN_E5	TV Decoder H_SYNC	3.3V
TD_VS	PIN_E4	TV Decoder V_SYNC	3.3V
TD_CLK27	PIN_B14	TV Decoder Clock Input.	3.3V
TD_RESET_N	PIN_G7	TV Decoder Reset	3.3V
I2C_SCLK	PIN_B7	I2C Clock	3.3V
I2C_SDAT	PIN_A8	I2C Data	3.3V

4.16 Implementing a TV Encoder

Although the DE2-115 board does not include a TV encoder chip, the ADV7123 (10-bit high-speed triple ADCs) can be used to implement a professional-quality TV encoder with the digital processing part implemented in the Cyclone IV E FPGA. **Figure 4-30** shows a block diagram of a TV encoder implemented in this manner.

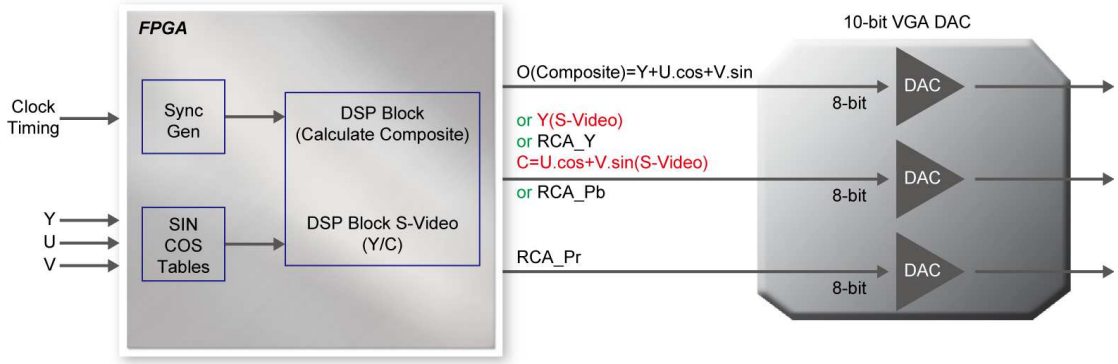


Figure 4-30 A TV Encoder that uses the Cyclone IV E FPGA and the ADV7123

4.17 Using the USB Interface

The DE2-115 board provides both USB host and device interfaces using the Cypress EZ-OTG (CY7C67200) On-The-Go (OTG) host/peripheral controller. The host and device controllers are compliant with the Universal Serial Bus Specification Rev. 2.0, supporting data transfer at full-speed (12 Mbit/s) and low-speed (1.5 Mbit/s). **Figure 4-31** shows the schematic diagram of the USB circuitry. The interface between CY7C67200 and FPGA is set to Host Port Interface (HPI) which can provide DMA access to the CY7C67200 internal memory by FPGA, plus a bidirectional mailbox register for supporting high-level communication protocols. Also the pin assignments for the associated interface are listed in **Table 4-25**.

Detailed information for using the CY7C67200 device is available in its datasheet and programming guide; both documents can be found on the manufacturer's website, or in the *DE2_115_datasheets\USB* folder on the DE2-115 System CD. The most challenging part of a USB application is in the design of the software driver needed. Two complete examples of USB drivers, for both host and device applications, can be found in Sections 6.4 and 6.5. These demonstrations provide examples of software drivers for the Nios II processor.

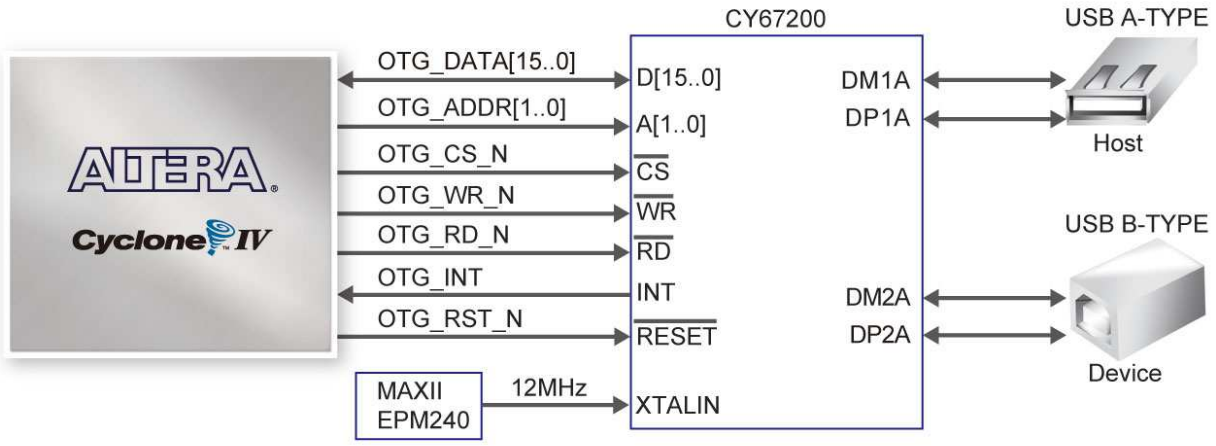


Figure 4-31 Connections between FPGA and USB (CY7C67200)

Table 4-25 USB (CY7C67200) Pin Assignments

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
OTG_ADDR[0]	PIN_H7	CY7C67200 Address[0]	3.3V
OTG_ADDR[1]	PIN_C3	CY7C67200 Address[1]	3.3V
OTG_DATA[0]	PIN_J6	CY7C67200 Data[0]	3.3V
OTG_DATA[1]	PIN_K4	CY7C67200 Data[1]	3.3V
OTG_DATA[2]	PIN_J5	CY7C67200 Data[2]	3.3V
OTG_DATA[3]	PIN_K3	CY7C67200 Data[3]	3.3V
OTG_DATA[4]	PIN_J4	CY7C67200 Data[4]	3.3V
OTG_DATA[5]	PIN_J3	CY7C67200 Data[5]	3.3V
OTG_DATA[6]	PIN_J7	CY7C67200 Data[6]	3.3V
OTG_DATA[7]	PIN_H6	CY7C67200 Data[7]	3.3V
OTG_DATA[8]	PIN_H3	CY7C67200 Data[8]	3.3V
OTG_DATA[9]	PIN_H4	CY7C67200 Data[9]	3.3V
OTG_DATA[10]	PIN_G1	CY7C67200 Data[10]	3.3V
OTG_DATA[11]	PIN_G2	CY7C67200 Data[11]	3.3V
OTG_DATA[12]	PIN_G3	CY7C67200 Data[12]	3.3V
OTG_DATA[13]	PIN_F1	CY7C67200 Data[13]	3.3V
OTG_DATA[14]	PIN_F3	CY7C67200 Data[14]	3.3V
OTG_DATA[15]	PIN_G4	CY7C67200 Data[15]	3.3V
OTG_CS_N	PIN_A3	CY7C67200 Chip Select	3.3V
OTG_RD_N	PIN_B3	CY7C67200 Read	3.3V
OTG_WR_N	PIN_A4	CY7C67200 Write	3.3V
OTG_RST_N	PIN_C5	CY7C67200 Reset	3.3V
OTG_INT	PIN_D5	CY7C67200 Interrupt	3.3V

4.18 Using IR

The DE2-115 provides an infrared remote-control receiver Module (model: IRM-V538N7/TR1),

whose datasheet is offered in the *DE2_115_datasheets\IR_Receiver* folder on DE2-115 system CD. Note that for this all-in-one receiver module, it is only compatible with the 38KHz carrier Standard, with a maximum data rate of about 4kbps for its product information. The accompanied remote controller with an encoding chip of uPD6121G is very suitable of generating expected infrared signals. **Figure 4-32** shows the related schematic of the IR receiver, and the pin assignments of the associated interface are listed in **Table 4-26**.

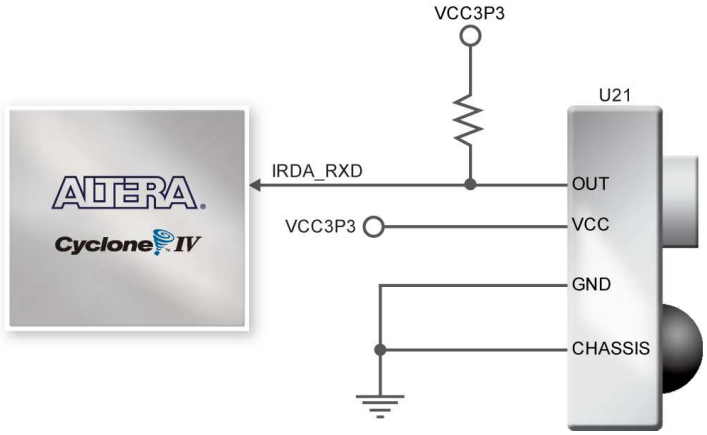


Figure 4-32 Connection between FPGA and IR

Table 4-26 IR Pin Assignments

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
IRDA_RXD	PIN_Y15	IR Receiver	3.3V

4.19 Using SRAM/SDRAM/FLASH/EEPROM/SD Card

■ SRAM

The DE2-115 board has 2MB SRAM memory with 16-bit data width. Being featured with a maximum performance frequency of about 125MHz under the condition of standard 3.3V single power supply makes it suitable of dealing with high-speed media processing applications that need ultra data throughput. The related schematic is shown in **Figure 4-33**.

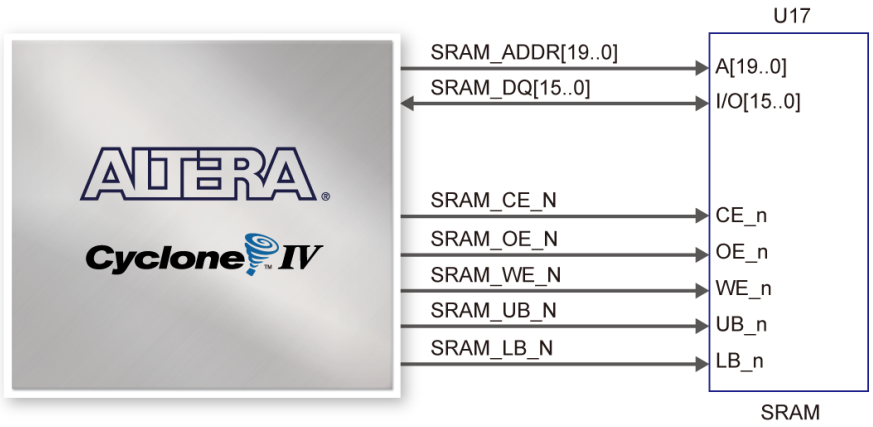


Figure 4-33 Connections between FPGA and SRAM

■ SDRAM

The board features 128MB of SDRAM, implemented using two 64MB SDRAM devices. Each device consists of separate 16-bit data lines connected to the FPGA, and shared control and address lines. These chips use the 3.3V LVCMOS signaling standard. Connections between FPGA and SDRAM are shown in Figure 4-34.

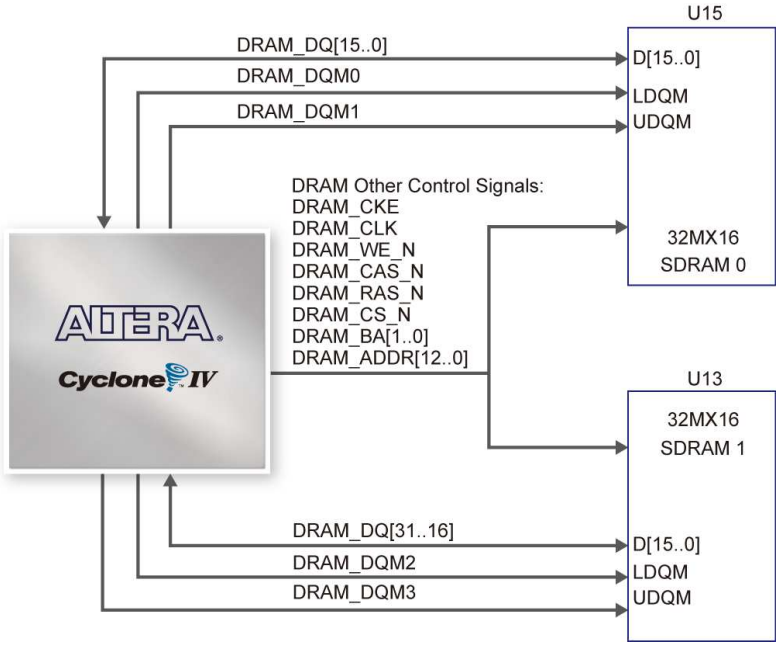


Figure 4-34 Connections between FPGA and SDRAM

■ FLASH

The board is assembled with 8MB of flash memory using an 8-bit data bus. The device uses 3.3V CMOS signaling standard. Because of its non-volatile property, it is usually used for storing software binaries, images, sounds or other media. Connections between FPGA and Flash are shown in Figure 4-35.

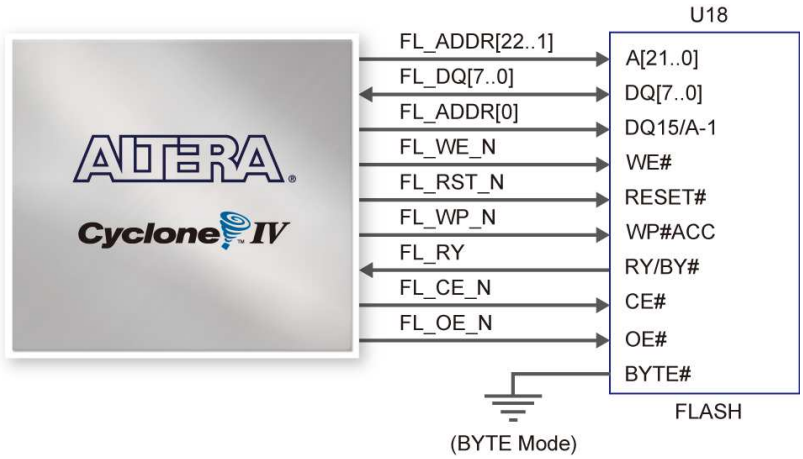


Figure 4-35 Connections between FPGA and Flash

■ EEPROM

The board has 32Kb EEPROM. With the benefit of I2C bus, users could use it as residence of user data like version information, MAC address or other description substance. Figure 4-36 gives the schematic view of the EEPROM. The configured access address of EEPROM is 0xA0 for writing and 0xA1 for reading.

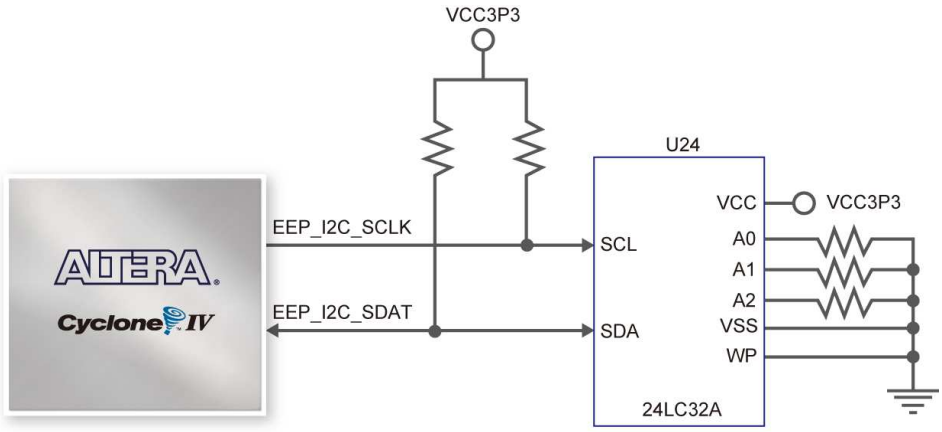


Figure 4-36 Connections between FPGA and EEPROM

■ SD Card

Many applications use a large external storage device, such as SD Card or CF card, for storing data. The DE2-115 board provides the hardware needed for SD Card access. Users can implement custom controllers to access the SD Card in SPI mode and SD Card 4-bit or 1-bit mode. **Figure 4-37** shows the related signals.

Finally, **Table 4-27~Table 4-30** lists all the associated pins for interfacing FPGA respectively.

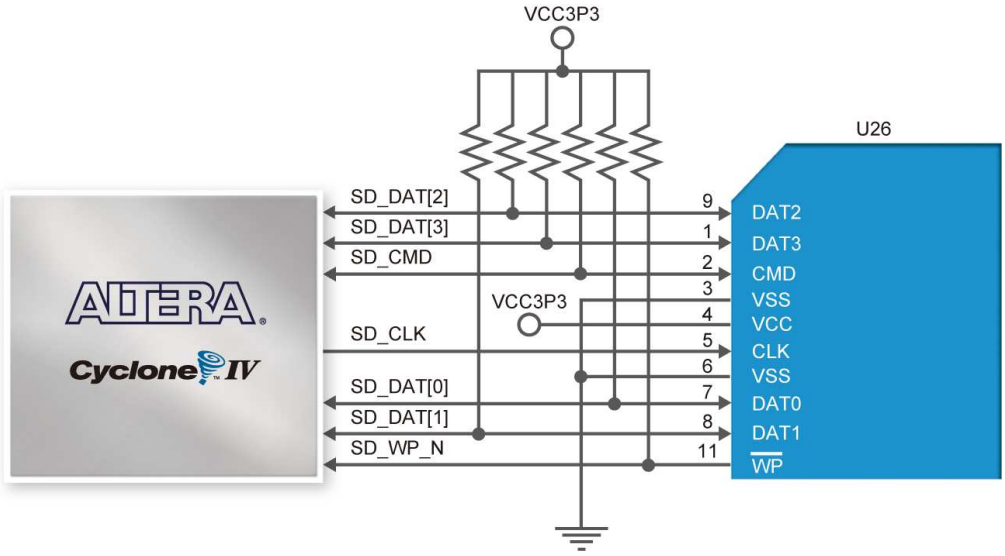


Figure 4-37 Connections between FPGA and SD Card Socket

Table 4-27 SRAM Pin Assignments

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
SRAM_ADDR[0]	PIN_AB7	SRAM Address[0]	3.3V
SRAM_ADDR[1]	PIN_AD7	SRAM Address[1]	3.3V
SRAM_ADDR[2]	PIN_AE7	SRAM Address[2]	3.3V
SRAM_ADDR[3]	PIN_AC7	SRAM Address[3]	3.3V
SRAM_ADDR[4]	PIN_AB6	SRAM Address[4]	3.3V
SRAM_ADDR[5]	PIN_AE6	SRAM Address[5]	3.3V
SRAM_ADDR[6]	PIN_AB5	SRAM Address[6]	3.3V
SRAM_ADDR[7]	PIN_AC5	SRAM Address[7]	3.3V
SRAM_ADDR[8]	PIN_AF5	SRAM Address[8]	3.3V
SRAM_ADDR[9]	PIN_T7	SRAM Address[9]	3.3V
SRAM_ADDR[10]	PIN_AF2	SRAM Address[10]	3.3V
SRAM_ADDR[11]	PIN_AD3	SRAM Address[11]	3.3V
SRAM_ADDR[12]	PIN_AB4	SRAM Address[12]	3.3V
SRAM_ADDR[13]	PIN_AC3	SRAM Address[13]	3.3V
SRAM_ADDR[14]	PIN_AA4	SRAM Address[14]	3.3V

SRAM_ADDR[15]	PIN_AB11	SRAM Address[15]	3.3V
SRAM_ADDR[16]	PIN_AC11	SRAM Address[16]	3.3V
SRAM_ADDR[17]	PIN_AB9	SRAM Address[17]	3.3V
SRAM_ADDR[18]	PIN_AB8	SRAM Address[18]	3.3V
SRAM_ADDR[19]	PIN_T8	SRAM Address[19]	3.3V
SRAM_DQ[0]	PIN_AH3	SRAM Data[0]	3.3V
SRAM_DQ[1]	PIN_AF4	SRAM Data[1]	3.3V
SRAM_DQ[2]	PIN_AG4	SRAM Data[2]	3.3V
SRAM_DQ[3]	PIN_AH4	SRAM Data[3]	3.3V
SRAM_DQ[4]	PIN_AF6	SRAM Data[4]	3.3V
SRAM_DQ[5]	PIN_AG6	SRAM Data[5]	3.3V
SRAM_DQ[6]	PIN_AH6	SRAM Data[6]	3.3V
SRAM_DQ[7]	PIN_AF7	SRAM Data[7]	3.3V
SRAM_DQ[8]	PIN_AD1	SRAM Data[8]	3.3V
SRAM_DQ[9]	PIN_AD2	SRAM Data[9]	3.3V
SRAM_DQ[10]	PIN_AE2	SRAM Data[10]	3.3V
SRAM_DQ[11]	PIN_AE1	SRAM Data[11]	3.3V
SRAM_DQ[12]	PIN_AE3	SRAM Data[12]	3.3V
SRAM_DQ[13]	PIN_AE4	SRAM Data[13]	3.3V
SRAM_DQ[14]	PIN_AF3	SRAM Data[14]	3.3V
SRAM_DQ[15]	PIN_AG3	SRAM Data[15]	3.3V
SRAM_OE_N	PIN_AD5	SRAM Output Enable	3.3V
SRAM_WE_N	PIN_AE8	SRAM Write Enable	3.3V
SRAM_CE_N	PIN_AF8	SRAM Chip Select	3.3V
SRAM_LB_N	PIN_AD4	SRAM Lower Byte Strobe	3.3V
SRAM_UB_N	PIN_AC4	SRAM Higher Byte Strobe	3.3V

Table 4-28 SDRAM Pin Assignments

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
DRAM_ADDR[0]	PIN_R6	SDRAM Address[0]	3.3V
DRAM_ADDR[1]	PIN_V8	SDRAM Address[1]	3.3V
DRAM_ADDR[2]	PIN_U8	SDRAM Address[2]	3.3V
DRAM_ADDR[3]	PIN_P1	SDRAM Address[3]	3.3V
DRAM_ADDR[4]	PIN_V5	SDRAM Address[4]	3.3V
DRAM_ADDR[5]	PIN_W8	SDRAM Address[5]	3.3V
DRAM_ADDR[6]	PIN_W7	SDRAM Address[6]	3.3V
DRAM_ADDR[7]	PIN_AA7	SDRAM Address[7]	3.3V
DRAM_ADDR[8]	PIN_Y5	SDRAM Address[8]	3.3V
DRAM_ADDR[9]	PIN_Y6	SDRAM Address[9]	3.3V
DRAM_ADDR[10]	PIN_R5	SDRAM Address[10]	3.3V
DRAM_ADDR[11]	PIN_AA5	SDRAM Address[11]	3.3V
DRAM_ADDR[12]	PIN_Y7	SDRAM Address[12]	3.3V
DRAM_DQ[0]	PIN_W3	SDRAM Data[0]	3.3V
DRAM_DQ[1]	PIN_W2	SDRAM Data[1]	3.3V
DRAM_DQ[2]	PIN_V4	SDRAM Data[2]	3.3V

DRAM_DQ[3]	PIN_W1	SDRAM Data[3]	3.3V
DRAM_DQ[4]	PIN_V3	SDRAM Data[4]	3.3V
DRAM_DQ[5]	PIN_V2	SDRAM Data[5]	3.3V
DRAM_DQ[6]	PIN_V1	SDRAM Data[6]	3.3V
DRAM_DQ[7]	PIN_U3	SDRAM Data[7]	3.3V
DRAM_DQ[8]	PIN_Y3	SDRAM Data[8]	3.3V
DRAM_DQ[9]	PIN_Y4	SDRAM Data[9]	3.3V
DRAM_DQ[10]	PIN_AB1	SDRAM Data[10]	3.3V
DRAM_DQ[11]	PIN_AA3	SDRAM Data[11]	3.3V
DRAM_DQ[12]	PIN_AB2	SDRAM Data[12]	3.3V
DRAM_DQ[13]	PIN_AC1	SDRAM Data[13]	3.3V
DRAM_DQ[14]	PIN_AB3	SDRAM Data[14]	3.3V
DRAM_DQ[15]	PIN_AC2	SDRAM Data[15]	3.3V
DRAM_DQ[16]	PIN_M8	SDRAM Data[16]	3.3V
DRAM_DQ[17]	PIN_L8	SDRAM Data[17]	3.3V
DRAM_DQ[18]	PIN_P2	SDRAM Data[18]	3.3V
DRAM_DQ[19]	PIN_N3	SDRAM Data[19]	3.3V
DRAM_DQ[20]	PIN_N4	SDRAM Data[20]	3.3V
DRAM_DQ[21]	PIN_M4	SDRAM Data[21]	3.3V
DRAM_DQ[22]	PIN_M7	SDRAM Data[22]	3.3V
DRAM_DQ[23]	PIN_L7	SDRAM Data[23]	3.3V
DRAM_DQ[24]	PIN_U5	SDRAM Data[24]	3.3V
DRAM_DQ[25]	PIN_R7	SDRAM Data[25]	3.3V
DRAM_DQ[26]	PIN_R1	SDRAM Data[26]	3.3V
DRAM_DQ[27]	PIN_R2	SDRAM Data[27]	3.3V
DRAM_DQ[28]	PIN_R3	SDRAM Data[28]	3.3V
DRAM_DQ[29]	PIN_T3	SDRAM Data[29]	3.3V
DRAM_DQ[30]	PIN_U4	SDRAM Data[30]	3.3V
DRAM_DQ[31]	PIN_U1	SDRAM Data[31]	3.3V
DRAM_BA[0]	PIN_U7	SDRAM Bank Address[0]	3.3V
DRAM_BA[1]	PIN_R4	SDRAM Bank Address[1]	3.3V
DRAM_DQM[0]	PIN_U2	SDRAM byte Data Mask[0]	3.3V
DRAM_DQM[1]	PIN_W4	SDRAM byte Data Mask[1]	3.3V
DRAM_DQM[2]	PIN_K8	SDRAM byte Data Mask[2]	3.3V
DRAM_DQM[3]	PIN_N8	SDRAM byte Data Mask[3]	3.3V
DRAM_RAS_N	PIN_U6	SDRAM Row Address Strobe	3.3V
DRAM_CAS_N	PIN_V7	SDRAM Column Address Strobe	3.3V
DRAM_CKE	PIN_AA6	SDRAM Clock Enable	3.3V
DRAM_CLK	PIN_AE5	SDRAM Clock	3.3V
DRAM_WE_N	PIN_V6	SDRAM Write Enable	3.3V
DRAM_CS_N	PIN_T4	SDRAM Chip Select	3.3V

Table 4-29 Flash Pin Assignments

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
FL_ADDR[0]	PIN_AG12	FLASH Address[0]	3.3V

FL_ADDR[1]	PIN_AH7	FLASH Address[1]	3.3V
FL_ADDR[2]	PIN_Y13	FLASH Address[2]	3.3V
FL_ADDR[3]	PIN_Y14	FLASH Address[3]	3.3V
FL_ADDR[4]	PIN_Y12	FLASH Address[4]	3.3V
FL_ADDR[5]	PIN_AA13	FLASH Address[5]	3.3V
FL_ADDR[6]	PIN_AA12	FLASH Address[6]	3.3V
FL_ADDR[7]	PIN_AB13	FLASH Address[7]	3.3V
FL_ADDR[8]	PIN_AB12	FLASH Address[8]	3.3V
FL_ADDR[9]	PIN_AB10	FLASH Address[9]	3.3V
FL_ADDR[10]	PIN_AE9	FLASH Address[10]	3.3V
FL_ADDR[11]	PIN_AF9	FLASH Address[11]	3.3V
FL_ADDR[12]	PIN_AA10	FLASH Address[12]	3.3V
FL_ADDR[13]	PIN_AD8	FLASH Address[13]	3.3V
FL_ADDR[14]	PIN_AC8	FLASH Address[14]	3.3V
FL_ADDR[15]	PIN_Y10	FLASH Address[15]	3.3V
FL_ADDR[16]	PIN_AA8	FLASH Address[16]	3.3V
FL_ADDR[17]	PIN_AH12	FLASH Address[17]	3.3V
FL_ADDR[18]	PIN_AC12	FLASH Address[18]	3.3V
FL_ADDR[19]	PIN_AD12	FLASH Address[19]	3.3V
FL_ADDR[20]	PIN_AE10	FLASH Address[20]	3.3V
FL_ADDR[21]	PIN_AD10	FLASH Address[21]	3.3V
FL_ADDR[22]	PIN_AD11	FLASH Address[22]	3.3V
FL_DQ[0]	PIN_AH8	FLASH Data[0]	3.3V
FL_DQ[1]	PIN_AF10	FLASH Data[1]	3.3V
FL_DQ[2]	PIN_AG10	FLASH Data[2]	3.3V
FL_DQ[3]	PIN_AH10	FLASH Data[3]	3.3V
FL_DQ[4]	PIN_AF11	FLASH Data[4]	3.3V
FL_DQ[5]	PIN_AG11	FLASH Data[5]	3.3V
FL_DQ[6]	PIN_AH11	FLASH Data[6]	3.3V
FL_DQ[7]	PIN_AF12	FLASH Data[7]	3.3V
FL_CE_N	PIN_AG7	FLASH Chip Enable	3.3V
FL_OE_N	PIN_AG8	FLASH Output Enable	3.3V
FL_RST_N	PIN_AE11	FLASH Reset	3.3V
FL_RY	PIN_Y1	FLASH Ready/Busy output	3.3V
FL_WE_N	PIN_AC10	FLASH Write Enable	3.3V
FL_WP_N	PIN_AE12	FLASH Write Protect /Programming Acceleration	3.3V

Table 4-30 EEPROM Pin Assignments

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
EEP_I2C_SCLK	PIN_D14	EEPROM clock	3.3V
EEP_I2C_SDAT	PIN_E14	EEPROM data	3.3V

Table 4-31 SD Card Socket Pin Assignments

Signal Name	*FPGA Pin No.*	*Description*	*I/O Standard*
SD_CLK	PIN_AE13	SD Clock	3.3V

SD_CMD	PIN_AD14	SD Command Line	3.3V
SD_DAT[0]	PIN_AE14	SD Data[0]	3.3V
SD_DAT[1]	PIN_AF13	SD Data[1]	3.3V
SD_DAT[2]	PIN_AB14	SD Data[2]	3.3V
SD_DAT[3]	PIN_AC14	SD Data[3]	3.3V
SD_WP_N	PIN_AF14	SD Write Protect	3.3V

Chapter 5

DE2-115 System Builder

This chapter describes how users can create a custom design project on the DE2-115 board by using DE2-115 Software Tool – DE2-115 System Builder.

5.1 Introduction

The DE2-115 System Builder is a Windows based software utility, designed to assist users to create a Quartus II project for the DE2-115 board within minutes. The generated Quartus II project files include:

- **Quartus II Project File (.qpf)**
- **Quartus II Setting File (.qsf)**
- **Top-Level Design File (.v)**
- **Synopsis Design Constraints file (.sdc)**
- **Pin Assignment Document (.htm)**

By providing the above files, DE2-115 System Builder prevents occurrence of situations that are prone to errors when users manually edit the top-level design file or place pin assignments. The common mistakes that users encounter are the following:

1. Board damaged for wrong pin/bank voltage assignments.
2. Board malfunction caused by wrong device connections or missing pin counts for connected ends.
3. Performance degeneration because of improper pin assignments.

5.2 General Design Flow

This section will introduce the general design flow to build a project for the DE2-115 board via the DE2-115 System Builder. The general design flow is illustrated in **Figure 5-1**.

Users should launch DE2-115 System Builder and create a new project according to their design requirements. When users complete the settings, the DE2-115 System Builder will generate two

major files which include top-level design file (.v) and Quartus II setting file (.qsf).

The top-level design file contains top-level verilog HDL wrapper for users to add their own design/logic. The Quartus II setting file contains information such as FPGA device type, top-level pin assignment, and I/O standard for each user-defined I/O pin.

Finally, Quartus II programmer must be used to download SOF file to DE2-115 board using JTAG interface.

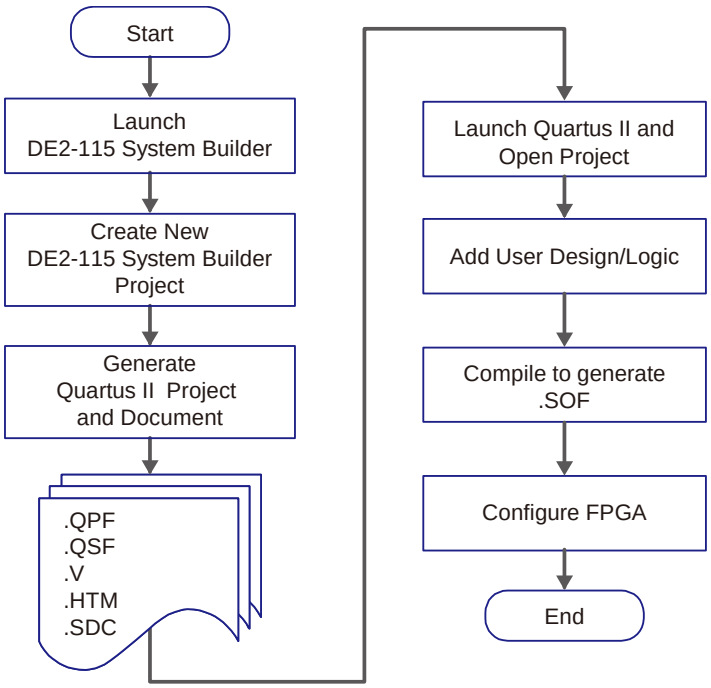


Figure 5-1 The general design flow of building a design

5.3 Using DE2-115 System Builder

This section provides the detailed procedures on how the DE2-115 System Builder is used.

■ Install and launch the DE2-115 System Builder

The DE2-115 System Builder is located in the directory: *"DE2_115_tools\DE2_115_system_builder"* on the DE2-115 System CD. Users can copy the whole folder to a host computer without installing the utility. Launch the DE2-115 System Builder by executing the DE2_115_SystemBuilder.exe on the host computer and the GUI window will appear as shown in **Figure 5-2**.

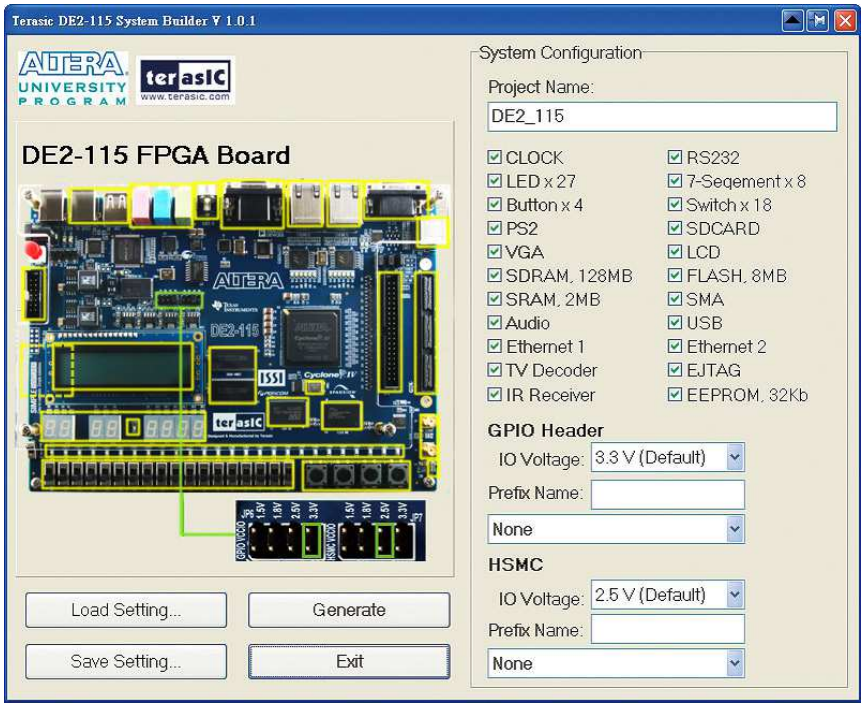


Figure 5-2 The DE2-115 System Builder window

■ Input Project Name

Input project name as show in **Figure 5-3**.

Project Name: Type in an appropriate name here, it will automatically be assigned as the name of your top-level design entity.

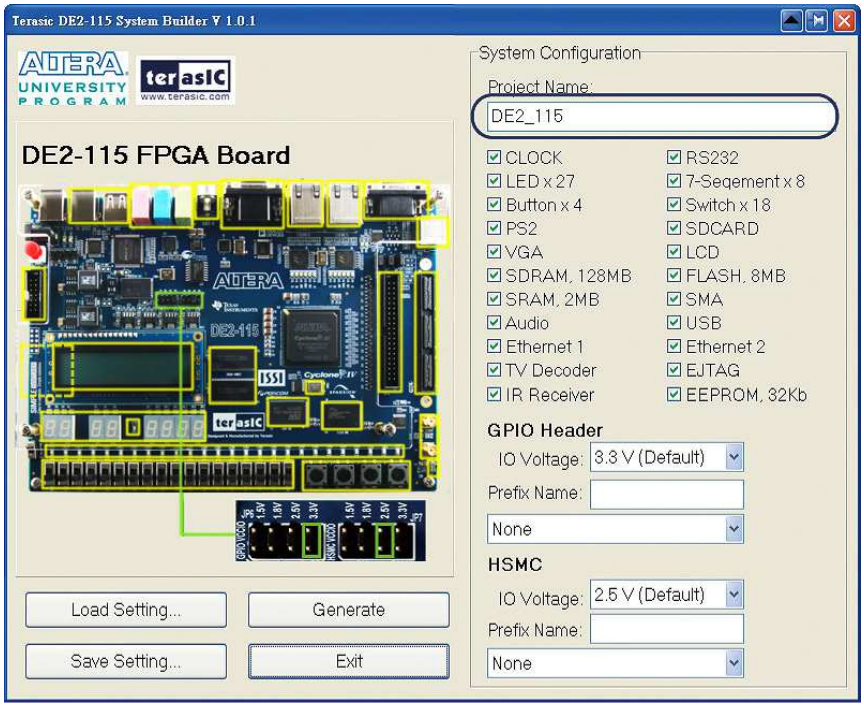


Figure 5-3 The DE2-115 Board Type and Project Name

■ System Configuration

Under System Configuration users are given the flexibility of enabling their choice of included components on the DE2-115 as shown in **Figure 5-4**. Each component of the DE2-115 is listed where users can enable or disable a component according to their design by simply marking a check or removing the check in the field provided. If the component is enabled, the DE2-115 System Builder will automatically generate the associated pin assignments including the pin name, pin location, pin direction, and I/O standard.

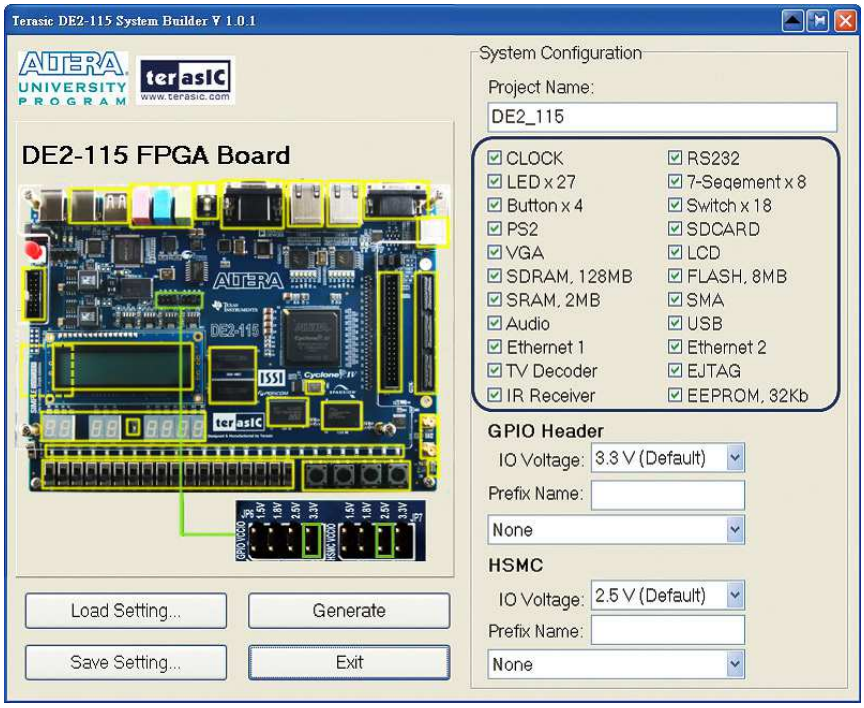


Figure 5-4 System Configuration Group

■ GPIO Expansion

Users can connect GPIO expansion card onto GPIO header located on the DE2-115 board as shown in **Figure 5-5**. Select the appropriate daughter card you wish to include in your design from the drop-down menu. The system builder will automatically generate the associated pin assignments including the pin name, pin location, pin direction, and IO standard.

If a customized daughter board is used, users can select "GPIO Default" followed by changing the pin name, pin direction, and IO standard according to the specification of the customized daughter board.

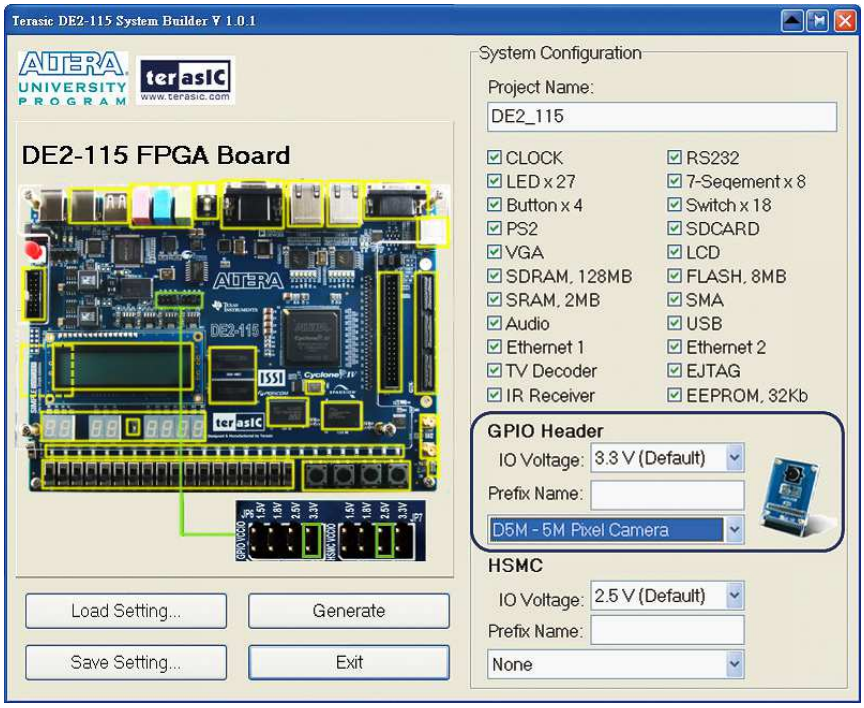


Figure 5-5 GPIO Expansion Group

The "Prefix Name" is an optional feature which denotes the prefix pin name of the daughter card assigned in your design. Users may leave this field empty.

■ HSMC Expansion

Users can connect HSMC-interfaced daughter cards onto HSMC located on the DE2-115 board shown in **Figure 5-6**. Select the daughter card you wish to add to your design under the appropriate HSMC connector where the daughter card is connected to. The System Builder will automatically generate the associated pin assignment including pin name, pin location, pin direction, and IO standard.

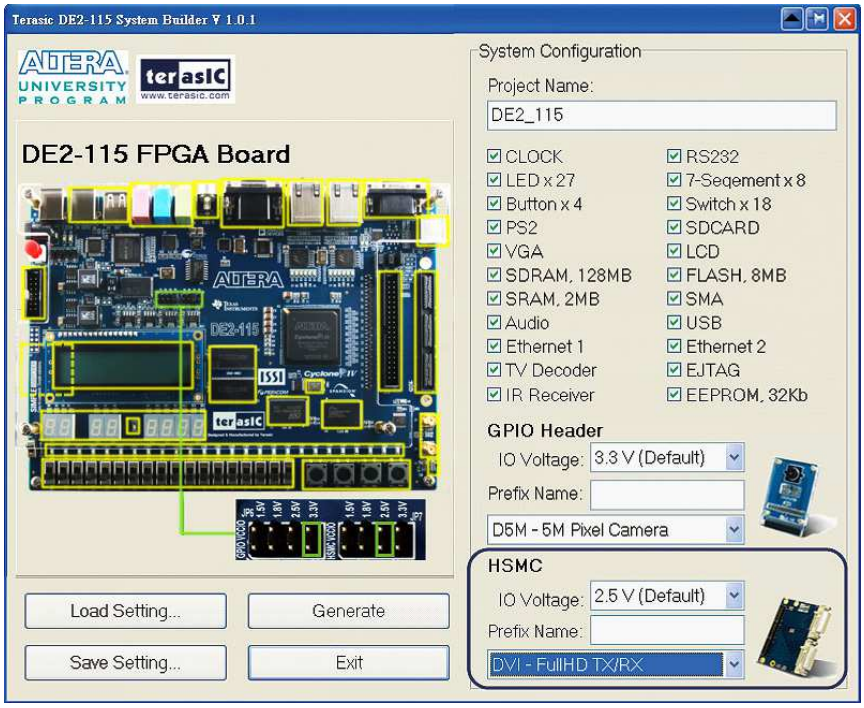


Figure 5-6 HSMC Expansion Group

The “Prefix Name” is an optional feature that denotes the pin name of the daughter card assigned in your design. Users may leave this field empty.

■ Project Setting Management

The DE2-115 System Builder also provides functions to restore default setting, loading a setting, and saving users’ board configuration file shown in **Figure 5-7**. Users can save the current board configuration information into a .cfg file and load it to the DE2-115 System Builder.

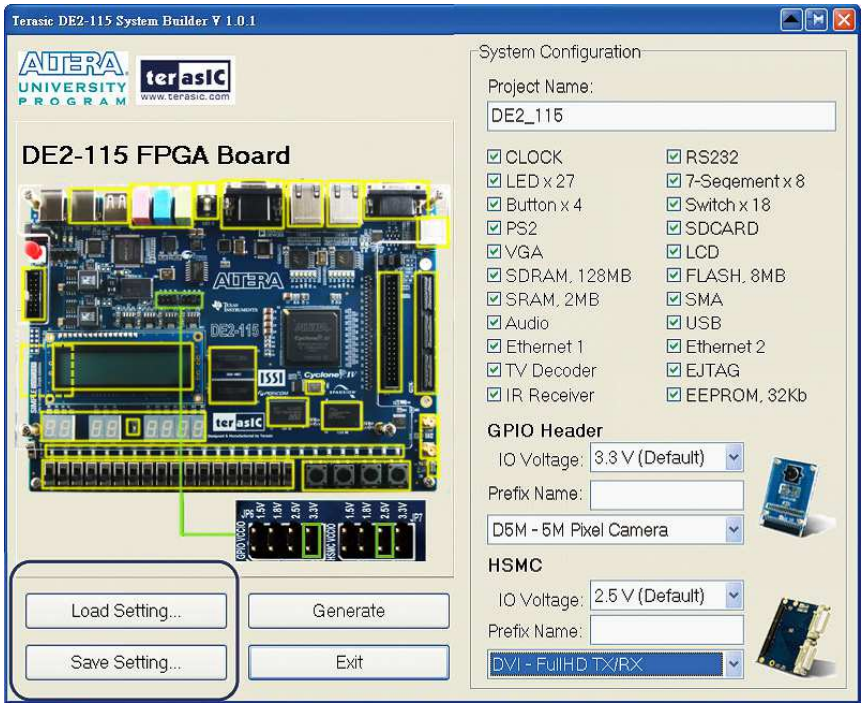


Figure 5-7 Project Settings

■ Project Generation

When users press the Generate button, the DE2-115 System Builder will generate the corresponding Quartus II files and documents as listed in the Table 5-1:

Table 5-1 The files generated by DE2-115 System Builder

No.	*Filename*	*Description*
1	<Project name>.v	Top level verilog HDL file for Quartus II
2	<Project name>.qpf	Quartus II Project File
3	<Project name>.qsf	Quartus II Setting File
4	<Project name>.sdc	Synopsis Design Constraints file for Quartus II
5	<Project name>.htm	Pin Assignment Document

Users can use Quartus II software to add custom logic into the project and compile the project to generate the SRAM Object File (.sof).

Chapter 6

Examples of Advanced Demonstrations

This chapter provides a number of examples of advanced circuits implemented on the DE2-115 board. These circuits provide demonstrations of the major features on the board, such as its audio and video capabilities, USB, and Ethernet connectivity. For each demonstration the Cyclone IV E FPGA (or EPCS64 serial EEPROM) configuration file is provided, as well as the full source code in Verilog HDL. All of the associated files can be found in the *DE2_115_demonstrations* folder on the DE2-115 System CD. For each demonstrations described in the following sections, the name of the project directory for its files is given, which are subdirectories of the *DE2_115_demonstrations* folder.

■ Installing the Demonstrations

To install the demonstrations on your computer:

Copy the directory *DE2_115_demonstrations* into a local directory of your choice. It is important to ensure that the path to your local directory contains no spaces – otherwise, the Nios II software will not work. Note Quartus II v9.1 SP2 is required for all DE2-115 demonstrations to support Cyclone IV E device. Quartus II v10.0 can be installed from the Altera Complete Design Suite DVD provided.

6.1 DE2-115 Factory Configuration

The DE2-115 board is shipped from the factory with a default configuration bit-stream that demonstrates some of the basic features of the board. The setup required for this demonstration, and the locations of its files are shown below.

■ Demonstration Setup, File Locations, and Instructions

- Project directory: DE2_115_Default
- Bit stream used: DE2_115_Default.sof or DE2_115_Default.pof
- Power on the DE2-115 board, with the USB cable connected to the USB Blaster port. If

necessary (that is, if the default factory configuration of the DE2-115 board is not currently stored in EPCS64 device), download the bit stream to the board by using either JTAG or AS programming

- You should now be able to observe that the 7-segment displays are displaying a sequence of characters, and the red and green LEDs are flashing. Also, "Welcome to the Altera DE2-115" is shown on the LCD display
- Optionally connect a VGA display to the VGA D-SUB connector. When connected, the VGA display should show a color picture
- Optionally connect a powered speaker to the stereo audio-out jack
- Place slide switch SW17 in the DOWN position to hear a 1 kHz humming sound from the audio-out port. Alternatively, if slide switch SW17 is in the UP position, and optionally connects the microphone in port with a microphone and/or connects the line-in port with an audio player's output, you will hear the sound from the microphone or audio player or mixed sound from both.

The Verilog HDL source code for this demonstration is provided in the *DE2_115_Default folder*, which also includes the necessary files for the corresponding Quartus II project. The top-level Verilog HDL file, called DE2_115_Default.v, can be used as a template for other projects, because it defines ports that correspond to all of the user-accessible pins on the Cyclone IV E FPGA.

6.2 TV Box Demonstration

This demonstration plays video and audio input from a DVD player using the VGA output, audio CODEC, and one TV decoder (U6) on the DE2-115 board. **Figure 6-1** shows the block diagram of the design. There are two major blocks in the circuit, called I2C_AV_Config and TV_to_VGA. The TV_to_VGA block consists of the ITU-R 656 Decoder, SDRAM Frame Buffer, YUV422 to YUV444, YcrCb to RGB, and VGA Controller. The figure also shows the TV Decoder (ADV7180) and the VGA DAC (ADV7123) chips used.

As soon as the bit stream is downloaded into the FPGA, the register values of the TV Decoder chip are used to configure the TV decoder via the I2C_AV_Config block, which uses the I2C protocol to communicate with the TV Decoder chip. Following the power-on sequence, the TV Decoder chip will be unstable for a time period; the Lock Detector is responsible for detecting this instability.

The ITU-R 656 Decoder block extracts YcrCb 4:2:2 (YUV 4:2:2) video signals from the ITU-R 656 data stream sent from the TV Decoder. It also generates a data valid control signal indicating the valid period of data output. Because the video signal from the TV Decoder is interlaced, we need to perform de-interlacing on the data source. We used the SDRAM Frame Buffer and a field selection multiplexer (MUX) which is controlled by the VGA controller to perform the de-interlacing operation. Internally, the VGA Controller generates data request and odd/even selection signals to the SDRAM Frame Buffer and filed selection multiplexer (MUX). The YUV422 to YUV444 block converts the selected YcrCb 4:2:2 (YUV 4:2:2) video data to the YcrCb 4:4:4 (YUV 4:4:4) video

data format.

Finally, the YcrCb_to_RGB block converts the YcrCb data into RGB data output. The VGA Controller block generates standard VGA synchronous signals VGA_HS and VGA_VS to enable the display on a VGA monitor.

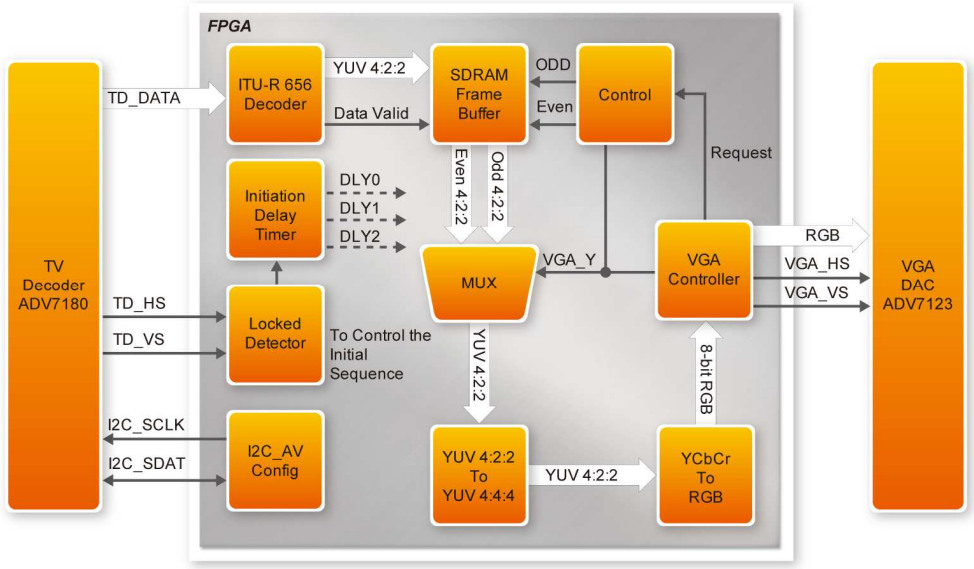


Figure 6-1 Block diagram of the TV box demonstration

■ Demonstration Setup, File Locations, and Instructions

- Project directory: DE2_115_TV
- Bit stream used: DE2_115_TV.sof or DE2_115_TV.pof
- Connect a DVD player's composite video output (yellow plug) to the Video-In RCA jack (J12) of the DE2-115 board. The DVD player has to be configured to provide:
    - NTSC output
    - 60Hz refresh rate
    - 4:3 aspect ratio
    - Non-progressive video

- Connect the VGA output of the DE2-115 board to a VGA monitor (both LCD and CRT type of monitors should work)
- Connect the audio output of the DVD player to the line-in port of the DE2-115 board and connect a speaker to the line-out port. If the audio output jacks from the DVD player are RCA type, then an adaptor will be needed to convert to the mini-stereo plug supported on the DE2-115 board; this is the same type of plug supported on most computers
- Load the bit stream into FPGA by execute the batch file 'de2_115_tv.bat' under DE2_115_TV\demo_batch\ folder

- Press KEY0 on the DE2-115 board to reset the circuit

Note: If the HSMC loopback adapter is mounted, the I2C_SCL will be directly routed back to I2C_SDA. Because audio chip, TV decoder chip and HSMC share one I2C bus, therefore audio and video chip won't function correctly.

Figure 6-2 illustrates the setup for this demonstration.

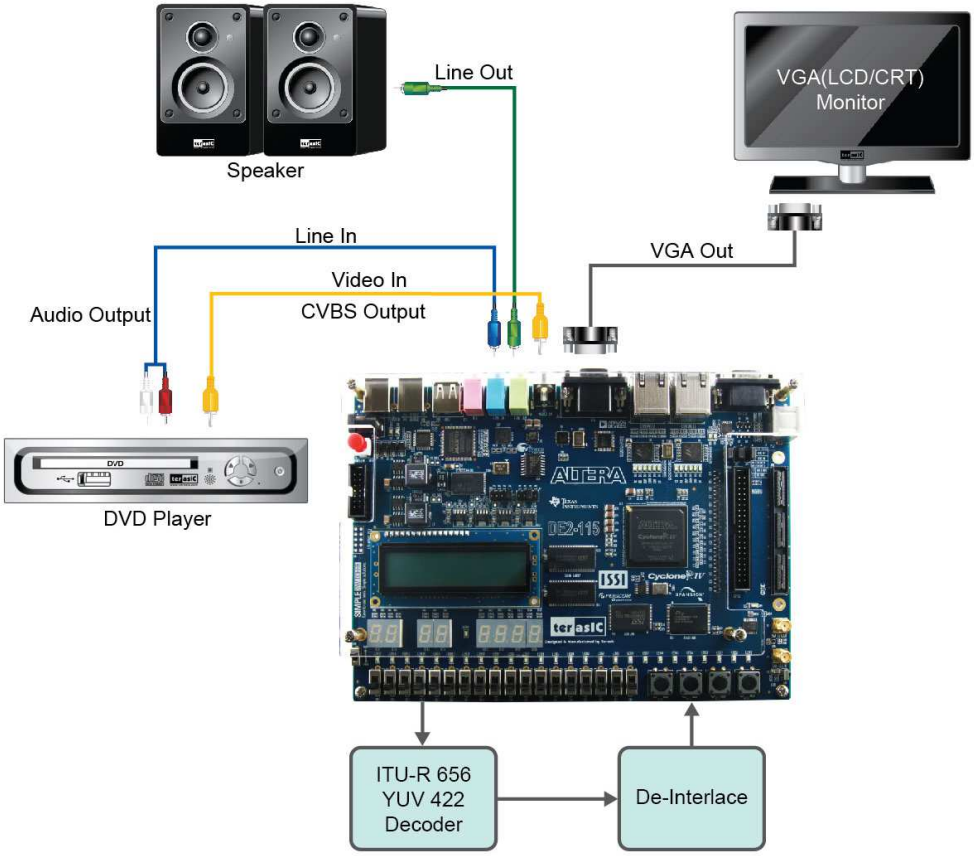


Figure 6-2 The setup for the TV box demonstration

6.3 USB Paintbrush

USB port is widely used in many multimedia products. The DE2-115 board provides a complete USB solution for both host and device applications. In this demonstration, we implement a Paintbrush application by using a USB mouse as the input device.

This demonstration uses the device port of the Cypress CY7C67200 chip and the Nios II processor to implement a USB mouse movement detector. We also implemented a video frame buffer with a

VGA controller to perform the real-time image storage and display. **Figure 6-3** shows the block diagram of the circuit, which allows the user to draw lines on the VGA display screen using the USB mouse. The VGA Controller block is integrated into the Altera Avalon bus so that it can be controlled by the Nios II processor.

Once the program runs, the Nios II processor is started as it will detect the existence of the USB mouse connected to DE2-115 board. When the mouse moves, the Nios II processor is able to keep track of the movement and record it in a frame buffer memory. The VGA Controller will overlap the data stored in the frame buffer with a default image pattern and display the overlapped image on the VGA display.

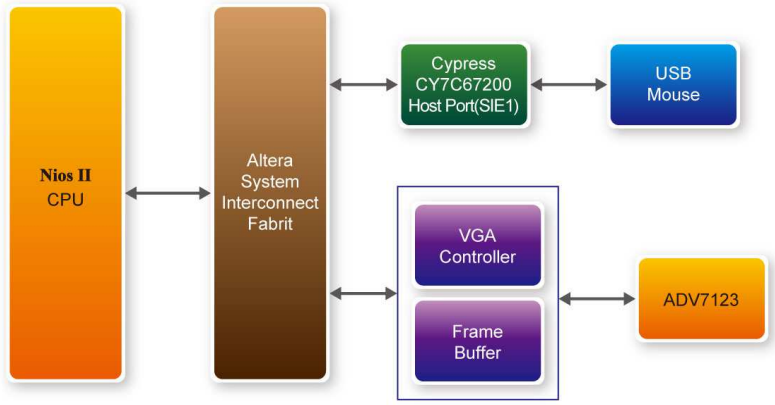


Figure 6-3 Block diagram of the USB paintbrush demonstration

■ Demonstration Setup, File Locations, and Instructions

Project directory: DE2_115_NIOS_HOST_MOUSE_VGA

Bit stream used: DE2_115_NIOS_HOST_MOUSE_VGA.sof

Nios II Workspace: DE2_115_NIOS_HOST_MOUSE_VGA\Software

- Connect a USB Mouse to the USB Host Connector (type A) of the DE2-115 board
- Connect the VGA output of the DE2-115 board to a VGA monitor (both LCD and CRT type of monitors should work)
- Load the bit stream into FPGA(note*)
- Run the Nios II and choose DE2_115_NIOS_HOST_MOUSE_VGA\Software as the workspace. Click on the Run button(note*)
- You should now be able to observe a blue background with an Altera logo on the VGA display
- Move the USB mouse and observe the corresponding movements of the cursor on the screen
- Left-click mouse to draw white dots/lines and right-click the mouse to erase white dots/lines on the screen.

Note: execute DE2_115_NIOS_HOST_MOUSE_VGA\demo_batch\nios_host_mouse_vga.bat will download .sof and .elf files.

Figure 6-4 illustrates the setup for this demonstration.

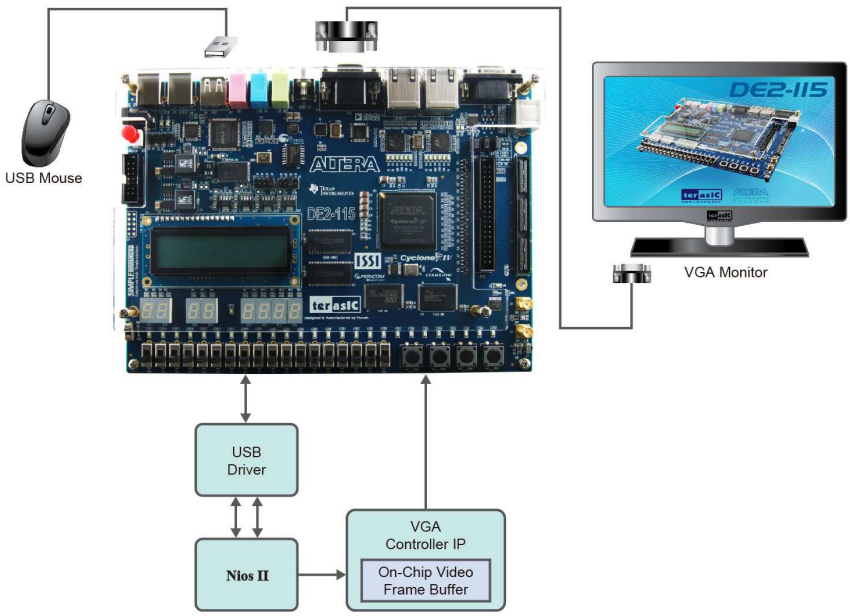


Figure 6-4 The setup for the USB paintbrush demonstration

6.4 USB Device

Most USB applications and products operate as USB devices, rather than USB hosts. In this demonstration, we show how the DE2-115 board can operate as a USB device that can be connected to a host computer. As indicated in the block diagram in Figure 6-5, the Nios II processor is used to communicate with the host computer via host port on the DE2-115 board's Cypress CY7C67200 device.

After connecting the DE2-115 board to a USB port on the host computer, a software program has to be executed on the Nios II processor to initialize the Cypress CY7C67200 chip. Once the software program is successfully executed, the host computer will identify the new device in its USB device list and asks for the associated driver; the device will be identified as a Terasic EZO USB. After completion of the driver installation on the host computer, the next step is to run a software program on the host computer called USB Controller.exe; this program communicates with the DE2-115 board.

Once a USB connection is established between PC and DE2-115 board, it is able to control and read

the status of specific components by using the USB Controller program, such as apply the "LED" Page to illumine the leds or go it out, or select the "Button_Switch" page to monitor the status of the Button & Switch.

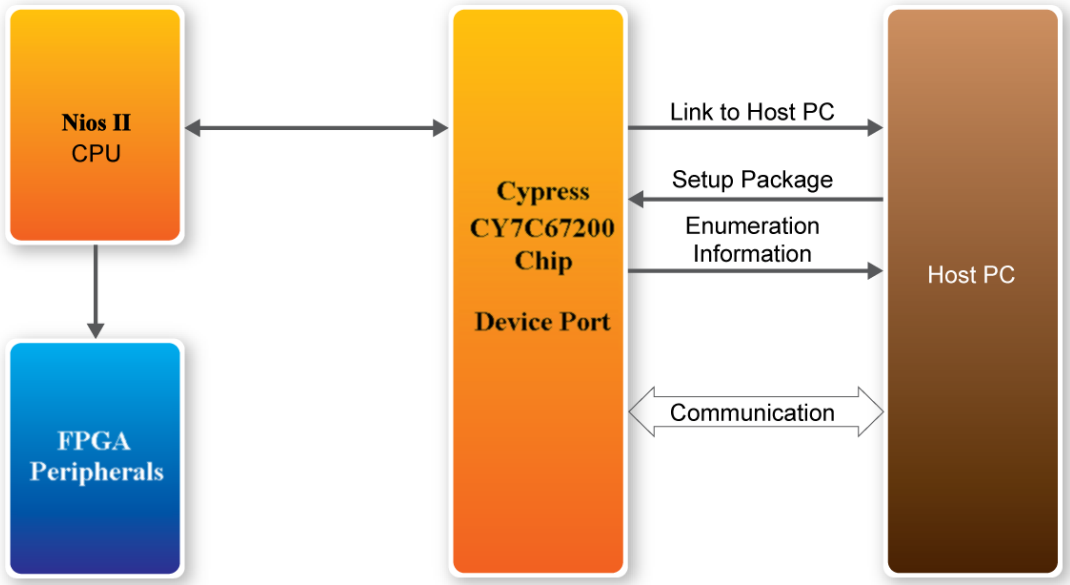


Figure 6-5 Block diagram of the USB device demonstration

■ Demonstration Setup, File Locations, and Instructions

- Project directory: DE2_115_NIOS_DEVICE
- Bit stream used : DE2_115_NIOS_DEVICE_LED.sof
- Quartus project : DE2_115_NIOS_DEVICE\HW
- Nios II Workspace: DE2_115_NIOS_DEVICE\HW\Software
- PC Software : DE2_115_NIOS_DEVICE\SW
- Connect the USB Device connector of the DE2-115 board to the host computer using a USB cable.
- Execute DE2_115_NIOS_DEVICE\demo_batch\ FPGA_bat\test.bat
- A new USB hardware device will be detected. Specify the location of the driver as DE2_115_NIOS_DEVICE\ demo_batch\PC\driver\...\cyusb.inf (Terasic EZO USB). (Note: If failed to install the driver due to digital signature verification. You can reboot PC, press F8 to enter the advanced options, and then disable digital signature verification.)
- The host computer should report that a Terasic EZO USB is now installed
- Execute the software located in DE2_115_NIOS_DEVICE \ demo_batch\PC\ USB_Control.exe on the host computer. Then, experiment with the program:
    - LCD_7SEG Control: Setting LCD and 7SEG
    - LED Control: Light or unlight LEDs

- Button_Switch Control: display the status of button and switches
- Hot-plug: Reconnect

Figure 6-6 illustrates the setup for this demonstration.

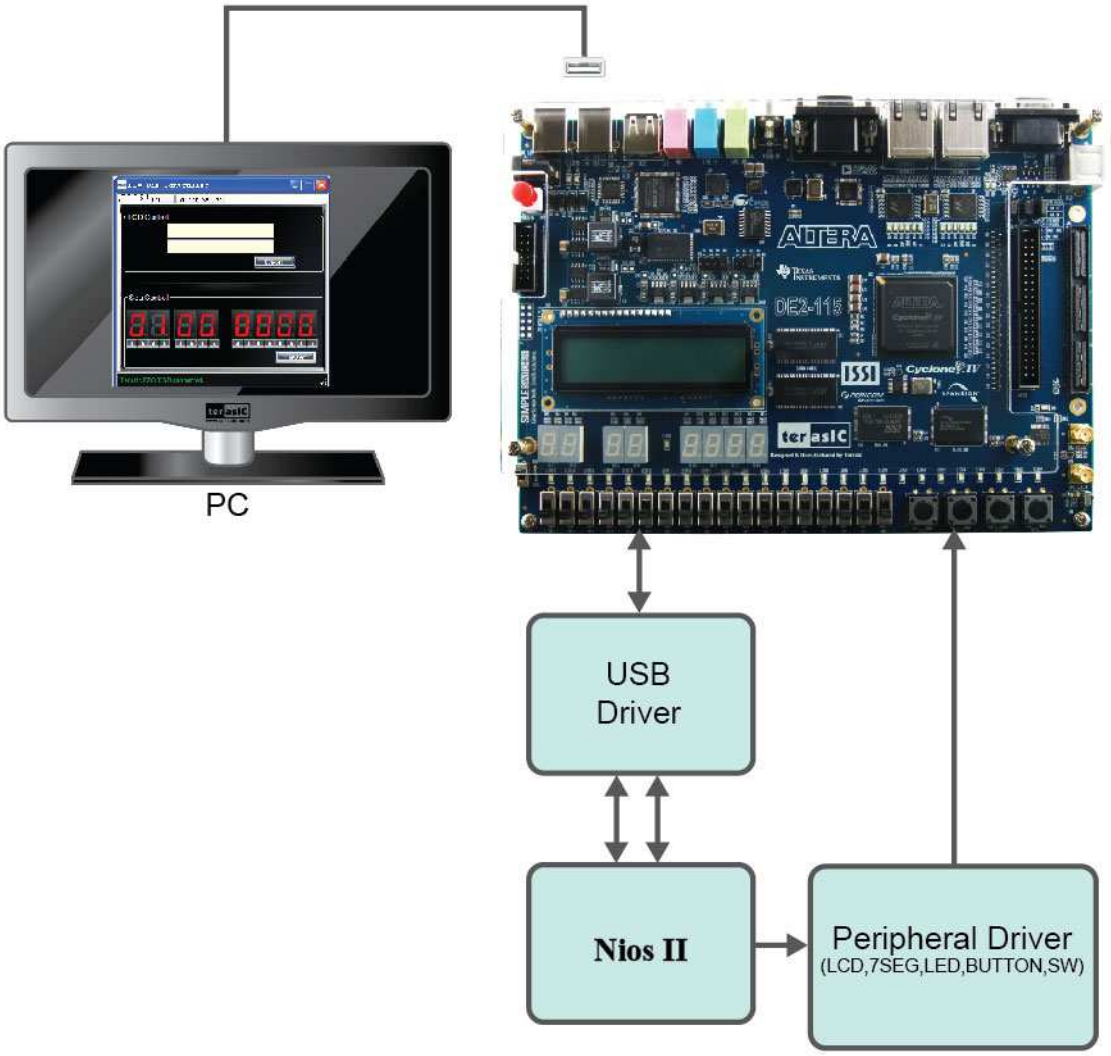


Figure 6-6 Setup for the USB device demonstration

6.5 A Karaoke Machine

This demonstration uses the microphone-in, line-in, and line-out ports on the DE2-115 board to create a Karaoke Machine application. The Wolfson WM8731 audio CODEC is configured in the master mode, with which the audio CODEC generates AD/DA serial bit clock (BCK) and the left/right channel clock (LRCK) automatically. As indicated in Figure 6-7, the I2C interface is used to configure the Audio CODEC. The sample rate and gain of the CODEC are set in this manner, and

the data input from the line-in port is then mixed with the microphone-in port and the result is sent to the line-out port.

For this demonstration the sample rate is set to 48kHz. Pressing the pushbutton KEY0 reconfigures the gain of the audio CODEC via I2C bus, cycling within ten predefined gain values (volume levels) provided by the device.

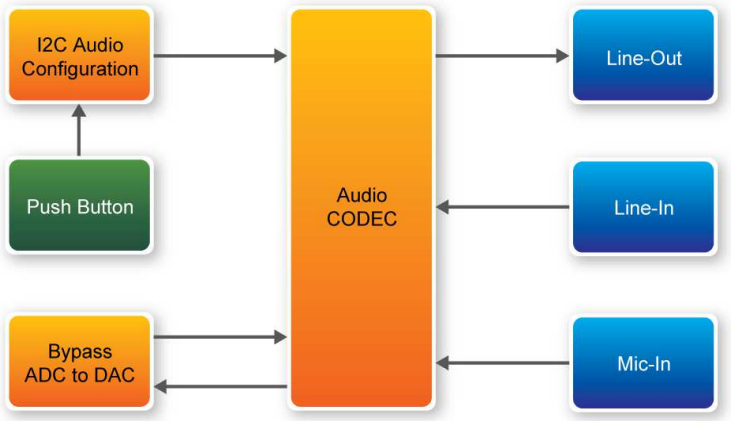


Figure 6-7 Block diagram of the Karaoke Machine demonstration

■ Demonstration Setup, File Locations, and Instructions

- Project directory: DE2_115_i2sound
- Bit stream used: DE2_115_i2sound.sof or DE2_115_i2sound.pof
- Connect a microphone to the microphone-in port (pink color) on the DE2-115 board
- Connect the audio output of a music-player, such as an MP3 player or computer, to the line-in port (blue color) on the DE2-115 board
- Connect a headset/speaker to the line-out port (green color) on the DE2-115 board
- Load the bit stream into the FPGA by execute the batch file 'DE2_115_i2sound' under the DE2_115_i2sound\demo_batch folder
- You should be able to hear a mixture of the microphone sound and the sound from the music player
- Press KEY0 to adjust the volume; it cycles between volume levels 0 to 9

Note: If the HSMC loopback adapter is mounted, the I2C_SCL will be directly routed back to I2C_SDA. Because audio chip, TV decoder chip and HSMC share one I2C bus, therefore audio and video chip won't function correctly.

Figure 6-8 illustrates the setup for this demonstration.

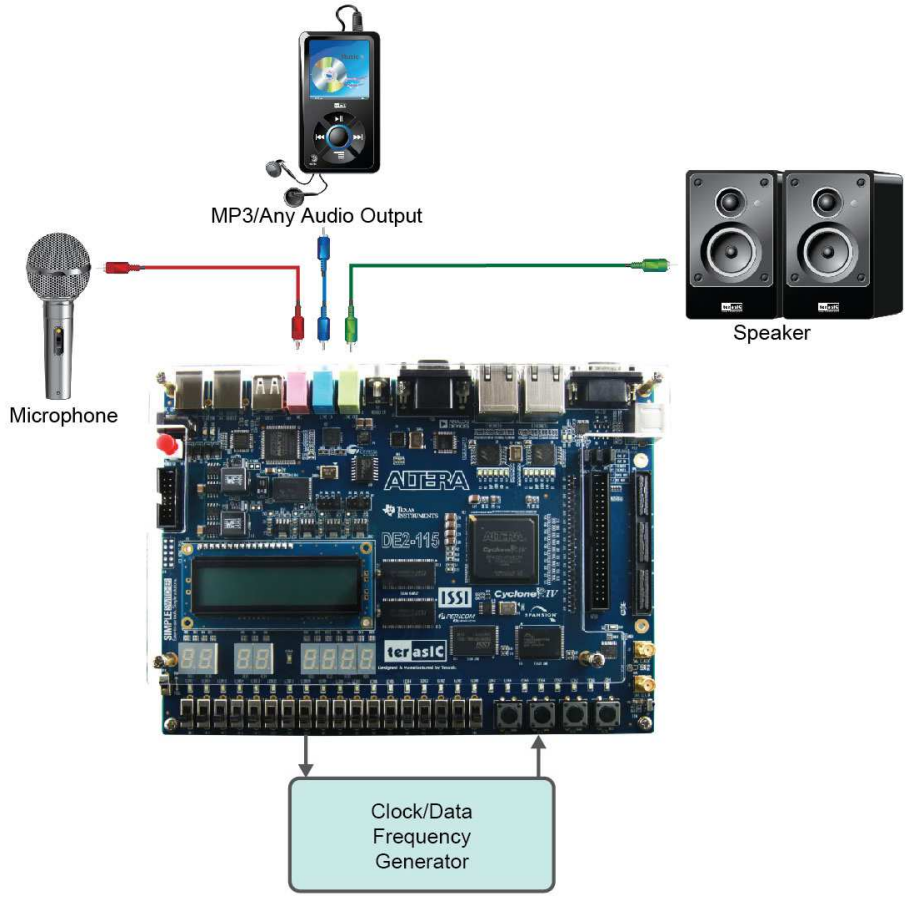


Figure 6-8 The setup for the Karaoke Machine

6.6 SD Card Demonstration

Many applications use a large external storage device, such as an SD Card or CF card to store data. The DE2-115 board provides the hardware and software needed for SD Card access. In this demonstration we will show how to browse files stored in the root directory of an SD Card and how to read the file contents of a specific file. The SD Card is required to be formatted as FAT File System in advance. Long file name is supported in this demonstration.

Figure 6-9 shows the hardware system block diagram of this demonstration. The system requires a 50MHz clock provided by the board. The PLL generates a 100MHz clock for the Nios II processor and other controllers. Four PIO pins are connected to the SD Card socket. SD 4-bit Mode is used to access the SD Card hardware. The SD 4-bit protocol and FAT File System function are all implemented by Nios II software. The software is stored in the on-chip memory.

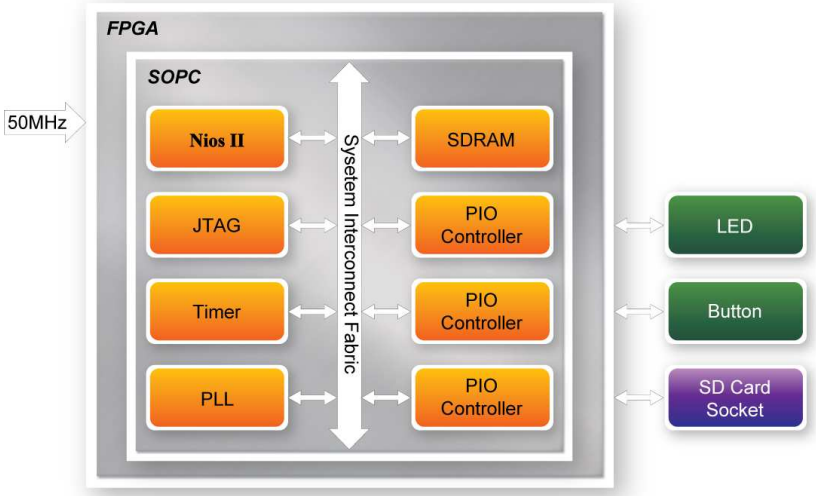


Figure 6-9 Block Diagram of the SD Card Demonstration

Figure 6-10 shows the software stack of this demonstration. The Nios PIO block provides basic IO functions to access hardware directly. The functions are provided from Nios II system and the function prototype is defined in the header file <io.h>. The SD Card block implements SD 4-bit mode protocol for communication with SD Cards. The FAT File System block implements reading function for FAT16 and FAT 32 file system. Long filename is supported. By calling the public FAT functions, users can browse files under the root directory of the SD Card. Furthermore, users can open a specified file and read the contents of the file.

The main block implements main control of this demonstration. When the program is executed, it detects whether an SD Card is inserted. If an SD Card is found, it will check whether the SD Card is formatted as FAT file system. If so, it searches all files in the root directory of the FAT file system and displays their names in the nios2-terminal. If a text file named “test.txt” is found, it will dump the file contents. If it successfully recognizes the FAT file system, it will turn on the green LED. On the other hand, it will turn on the red LED if it fails to parse the FAT file system or if there is no SD Card found in the SD Card socket of the DE2-115 board. If users press KEY3 of the DE2-115 board, the program will perform above process again.

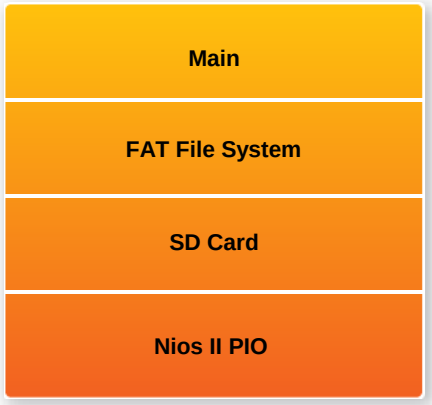


Figure 6-10 Software Stack of the SD Card Demonstration

Demonstration Source Code

- Project directory: DE2_115_SD_CARD
- Bit stream used: DE2_115_SD_CARD.sof
- Nios II Workspace: DE2_115_SD_CARD\Software

Demonstration Batch File

Demo Batch File Folder: DE2_115_SD_CARD \demo_batch

The demo batch file includes the following files:

- Batch File: DE2_115_SD_Card.bat, DE2_115_SD_CARD_bashrc
- FPGA Configure File : DE2_115_SD_CARD.sof
- Nios II Program: DE2_115_SD_CARD.elf

Demonstration Setup

- Make sure Quartus II and Nios II are installed on your PC.
- Power on the DE2-115 board.
- Connect USB Blaster to the DE2-115 board and install USB Blaster driver if necessary.
- Execute the demo batch file “DE2_115_SD_Card.bat” under the batch file folder, DE2_115_SD_CARD\demo_batch.
- After Nios II program is downloaded and executed successfully, a prompt message will be displayed in nios2-terminal.
- Copy test files to the root directory of the SD Card.
- Insert the SD Card into the SD Card socket of DE2-115, as shown in **Figure 6-11**.
- Press KEY3 of the DE2-115 board to start reading SD Card.
- The program will display SD Card information, as shown in **Figure 6-12**.

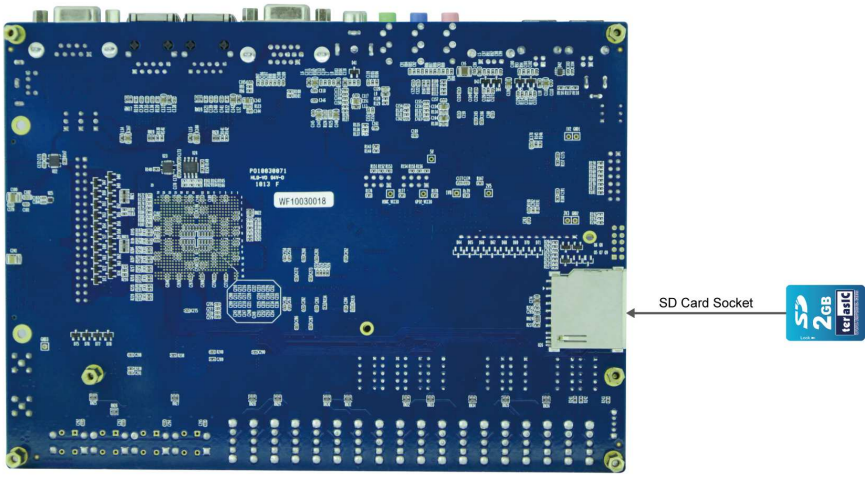


Figure 6-11 Insert SD Card for the SD Card Demonstration

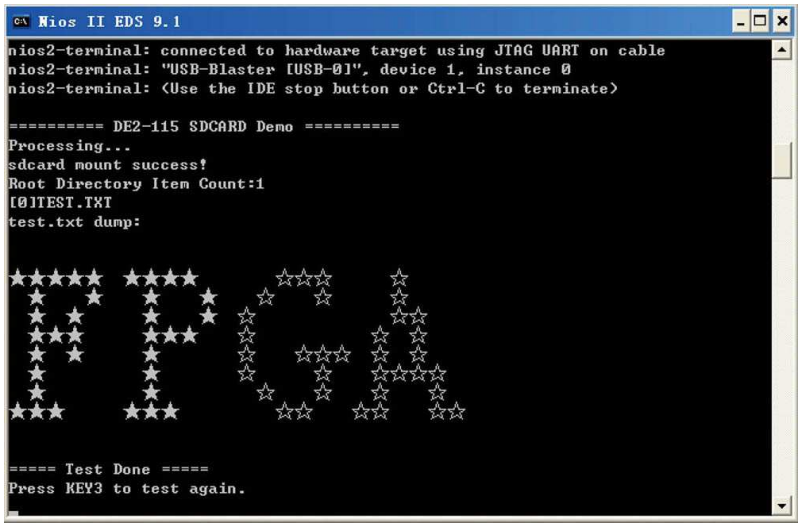


Figure 6-12 Running results of the SD Card demonstration

6.7 SD Card Music Player

Many commercial media/audio players use a large external storage device, such as an SD Card or CF card, to store music or video files. Such players may also include high-quality DAC devices so that good audio quality can be produced. The DE2-115 board provides the hardware and software needed for SD Card access and professional audio performance so that it is possible to design advanced multimedia products using the DE2-115 board.

In this demonstration we show how to implement an SD Card Music Player on the DE2-115 board, in which the music files are stored in an SD Card and the board can play the music files via its CD-quality audio DAC circuits. We use the Nios II processor to read the music data stored in the SD Card and use the Wolfson WM8731 audio CODEC to play the music.

Figure 6-13 shows the hardware block diagram of this demonstration. The system requires a 50 MHz clock provided from the board. The PLL generates a 100MHz clock for Nios II processor and the other controllers except for the audio controller. The audio chip is controlled by the Audio Controller which is a user-defined SOPC component. This audio controller needs an input clock of 18.432 MHz. In this design, the clock is provided by the PLL block. The audio controller requires the audio chip working in master mode, so the serial bit (BCK) and the left/right channel clock (LRCK) are provided by the audio chip. The 7-segment display is controlled by the SEG7 Controller which also is a user-defined SOPC component. Two PIO pins are connected to the I2C bus. The I2C protocol is implemented by software. Four PIO pins are connected to the SD Card socket. The IR receiver is controlled by the IR Controller which also is a user-defined SOPC component. SD 4-Bit Mode is used to access the SD Card and is implemented by software. All of the other SOPC components in the block diagram are SOPC Builder built-in components.

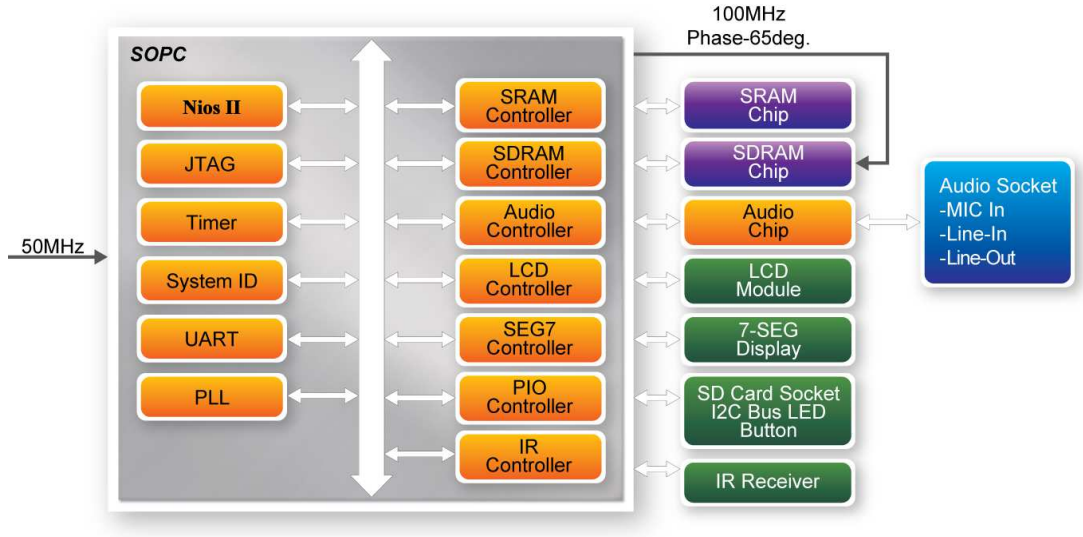


Figure 6-13 Block diagram of the SD music player demonstration

Figure 6-14 shows the software stack of this demonstration. SD 4-Bit Mode block implements the SD 4-Bit mode protocol for reading raw data from the SD Card. The FAT block implements FAT16/FAT32 file system for reading wave files that is stored in the SD Card. In this block, only read function is implemented. The WAVE Lib block implements WAVE file decoding function for extracting audio data from wave files. The I2C block implements I2C protocol for configuring audio chip. The SEG7 block implements displaying function to display elapsed playing time. The Audio block implements audio FIFO checking function and audio signal sending/receiving function. The IR block acts as a control interface of the music player system.

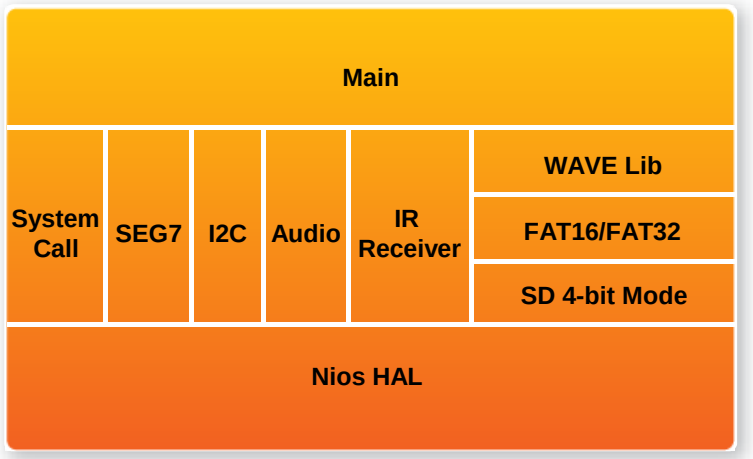


Figure 6-14 Software Stack of the SD music player demonstration

The audio chip should be configured before sending audio signal to the audio chip. The main program uses I2C protocol to configure the audio chip working in master mode; the audio output interface working in I2S 16-bits per channel and with sampling rate according to the wave file contents. In audio playing loop, the main program reads 512-byte audio data from the SD Card, and then writes the data to DAC FIFO in the Audio Controller. Before writing the data to the FIFO, the program will verify if the FIFO is full. The design also mixes the audio signal from the microphone-in and line-in for the Karaoke-style effects by enabling the BYPASS and SITETONE functions in the audio chip.

Finally, users can obtain the status of the SD music player from the 16x2 LCD module, the 7-segment display, and the LEDs. The top and bottom row of the LCD module will display the file name of the music that is played on the DE2-115 board and the value of music volume, respectively. The 7-segment displays will show the elapsed time of the playing music file. The LED will indicate the audio signal strength.

■ Demonstration Setup, File Locations, and Instructions

- **Project directory: DE2_115_SD_Card_Audio_Player**
- **Bit stream used: DE2_115_SD_Card_Audio_Player.sof**
- **Nios II Workspace: DE2_115_SD_Card_Audio_Player\Software**
- **Format your SD Card into FAT16/FAT32 format**
- **Place the wave files to the root directory of the SD Card. The provided wave files must have a sample rate of either 96K, 48K, 44.1K, 32K, or 8K. In addition, the wave files must be stereo and 16 bits per channel.**
- **Load the bitstream into the FPGA on the DE2-115 board(note*1)**
- **Run the Nios II Software under the workspace DE2_115_SD_Card_Audio_Player\Software(note*1)**

-
- **Connect a headset or speaker to the DE2-115 board and you should be able to hear the music played from the SD Card**
- **Press KEY3 on the DE2-115 board to play the next music file stored in the SD Card.**
- **Press KEY2 and KEY1 to increase and decrease the output music volume respectively.**
- **Users can also use the remote control to play/pause the music, select the last/next music file to play and control volume. The detailed information about each function of remote controller is shown in** Table 6-1**.**

Table 6-1 Detailed information of the button on the remote controller

Button Name	*Function Description*
PLAY	Play music or pause music
CHANNEL	Select last/next music file to play
VOLUME	Turn up/down volume
MUTE	Mute/un-mute

Note:

Figure 7- Execute the batch file DE2_115_SD_Card_Audio_Player\demo_batch \ DE2_115_SD_Card_Audio_Player.bat to download both hardware and software bit stream

2. If the capacity of your SD Memory Card is more than or equal 8GB, please make sure it has the performance more than or equal to Class 4

3. If the HSMC loopback adapter is mounted, the I2C_SCL will be directly routed back to I2C_SDA. Because audio chip, TV decoder chip and HSMC share one I2C bus, therefore audio and video chip won't function correctly.

Figure 6-15 illustrates the setup for this demonstration.

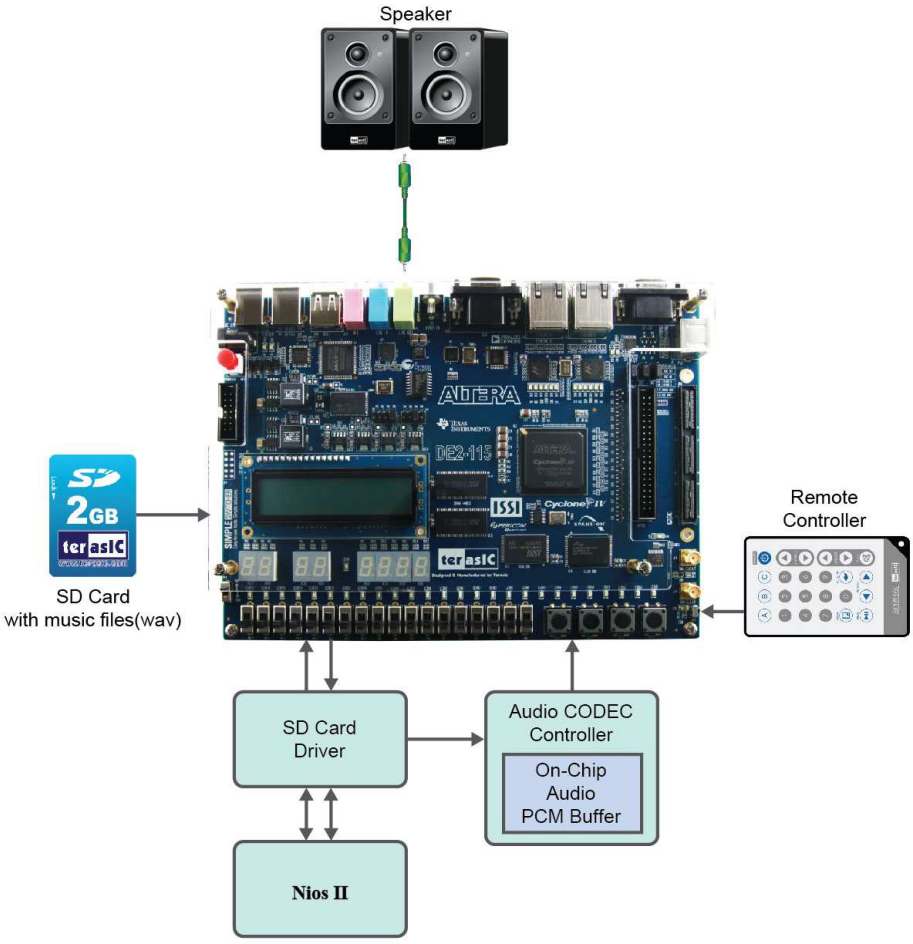


Figure 6-15 The setup for the SD music player demonstration

6.8 PS/2 Mouse Demonstration

We offer this simple PS/2 controller coded in Verilog HDL to demonstrate bidirectional communication between PS/2 controller and the device, the PS/2 mouse. You can treat it as a how-to basis and develop your own controller that could accomplish more sophisticated instructions, like setting the sampling rate or resolution, which need to transfer two data bytes.

For detailed information about the PS/2 protocol, please perform an appropriate search on various educational web sites. Here we give a brief introduction:

■ Outline

PS/2 protocol use two wires for bidirectional communication, one clock line and one data line. The PS/2 controller always has total control over the transmission line, but the PS/2 device generates clock signal during data transmission.

■ Data transmit from the device to controller

After sending an enabling instruction to the PS/2 mouse at stream mode, the device starts to send displacement data out, which consists of 33 bits. The frame data is cut into three similar slices, each of them containing a start bit (always zero) and eight data bits (with LSB first), one parity check bit (odd check), and one stop bit (always one).

PS/2 controller samples the data line at the falling edge of the PS/2 clock signal. This could easily be implemented using a shift register of 33 bits, but be cautious with the clock domain crossing problem.

■ Data transmit from the controller to device

Whenever the controller wants to transmit data to device, it first pulls the clock line low for more than one clock cycle to inhibit the current transmit process or to indicate the start of a new transmit process, which usually be called as inhibit state. After that, it pulls low the data line then release the clock line, and this is called the request state. The rising edge on the clock line formed by the release action can also be used to indicate the sample time point as for a ‘start bit. The device will detect this succession and generates a clock sequence in less than 10ms time. The transmit data consists of 12bits, one start bit (as explained before), eight data bits, one parity check bit (odd check), one stop bit (always one), and one acknowledge bit (always zero). After sending out the parity check bit, the controller should release the data line, and the device will detect any state change on the data line in the next clock cycle. If there’s no change on the data line for one clock cycle, the device will pull low the data line again as an acknowledgement which means that the data is correctly received.

After the power on cycle of the PS/2 mouse, it enters into stream mode automatically and disable data transmit unless an enabling instruction is received. **Figure 6-16** shows the waveform while communication happening on two lines.

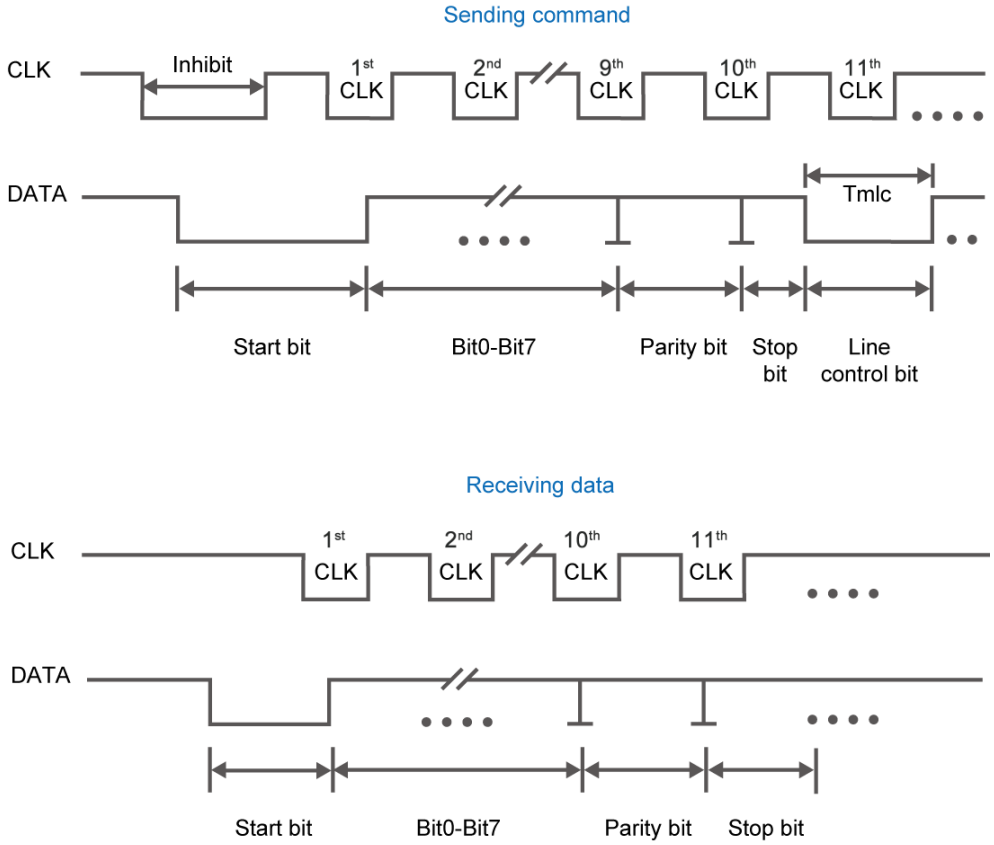


Figure 6-16 Waveforms on two lines while communication taking place

■ Demonstration Setup, File Locations, and Instructions

- **Project directory: DE2_115_PS2_DEMO**
- **Bit stream used : DE2_115_PS2_DEMO.sof**
- **Load the bit stream into FPGA by executing DE2_115_PS2_DEMO\demo_batch\DE2_115_PS2_DEMO.bat**
- **Plug in the PS/2 mouse**
- **Press KEY[0] for enabling data transfer**
- **Press KEY[1] to clear the display data cache**

You should see digital changes on 7-segment display when the PS/2 mouse moves, and the LEDG[2:0] will blink respectively when the left-button, right-button or middle-button is pressed. Table 6-2 gives the detailed information.

Table 6-2 Detailed information of the indicators

Indicator Name	*Description*
LEDG[0]	**Left button press indicator**
LEDG[1]	**Right button press indicator**

LEDG[2]	Middle button press indicator
HEX0	Low byte of X displacement
HEX1	High byte of X displacement
HEX2	Low byte of Y displacement
HEX3	High byte of Y displacement

Figure 6-17 illustrates the setup of this demonstration.

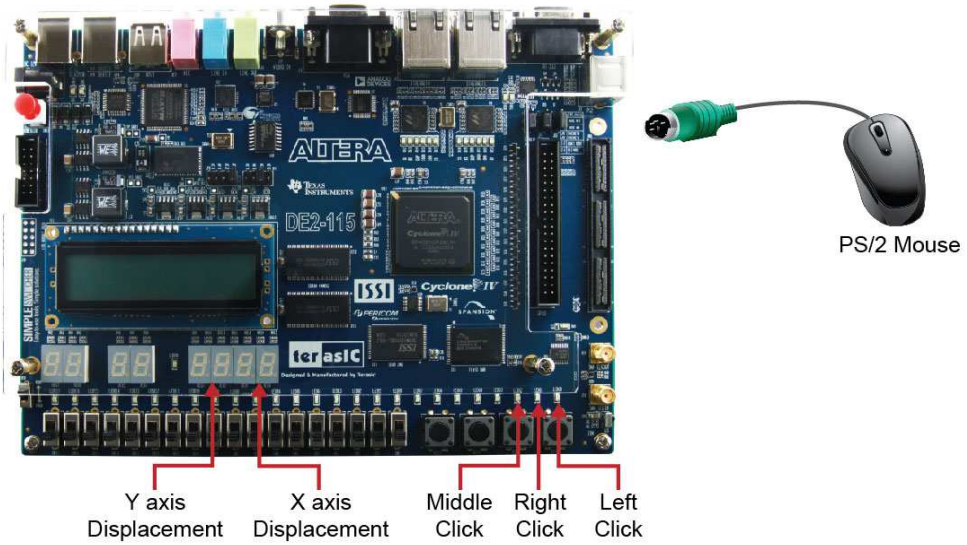


Figure 6-17 The setup of the PS/2 Mouse demonstration

6.9 IR Receiver Demonstration

In this demonstration, the key-related information that the user has pressed on the remote controller(Figure 6-18，Table 6-3) will be displayed on the DE2-115 board. Users only need to point the remote controller to the IR receiver on DE2-115 board and press the key. After the signal being decoded and processed through FPGA, the related information will be displayed on the 7-segment displays in hexadecimal format, which contains Custom Code, Key Code and Inversed Key Code. The Custom Code and Key Code are used to identify a remote controller and key on the remote controller, respectively.

Next we will introduce how this information being decoded and then displayed in this demo.

When a key on the remote controller is pressed, the remote controller will emit a standard frame, shown in Figure 6-19. The beginning of the frame is the lead code represents the start bit, and then is the key-related information, and the last 1 bit end code represents the end of the frame.

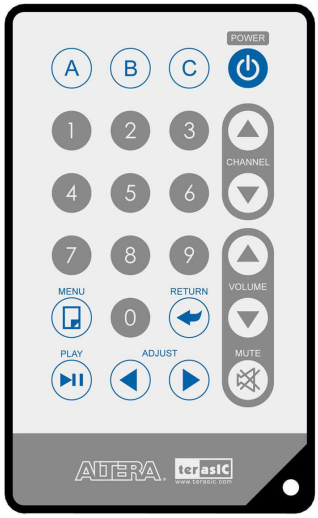


Figure 6-18 Remote controller

Table 6-3 Key code information for each Key on remote controller

Key	Key Code	Key	Key Code	Key	Key Code	Key	Key Code
A	0x0F	B	0x13	C	0x10		0x12
1	0x01	2	0x02	3	0x03		0x1A
4	0x04	5	0x05	6	0x06		0x1E
7	0x07	8	0x08	9	0x09		0x1B
	0x11	0	0x00		0x17		0x1F
	0x16		0x14		0x18		0x0C

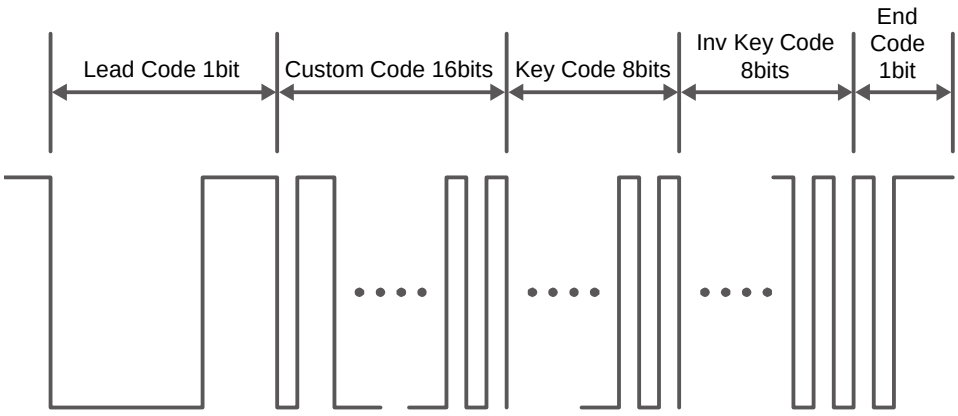


Figure 6-19 The transmitting frame of the IR remote controller

After the IR receiver on DE2-115 board receives this frame, it will directly transmit that to FPGA. In this demo, the IP of IR receiver controller is implemented in the FPGA. As **Figure 6-20** shows, it includes Code Detector, State Machine, and Shift Register. First, the IR receiver demodulates the signal inputs to Code Detector block .The Code Detector block will check the Lead Code and feedback the examination result to State Machine block.

The State Machine block will change the state from IDLE to GUIDANCE once the Lead code is detected. Once the Code Detector has detected the Custom Code status, the current state will change from GUIDANCE to DATAREAD state. At this state, the Code Detector will save the Custom Code and Key/Inv Key Code and output to Shift Register then displays it on 7-segment displays. **Figure 6-21** shows the state shift diagram of State Machine block. Note that the input clock should be 50MHz.

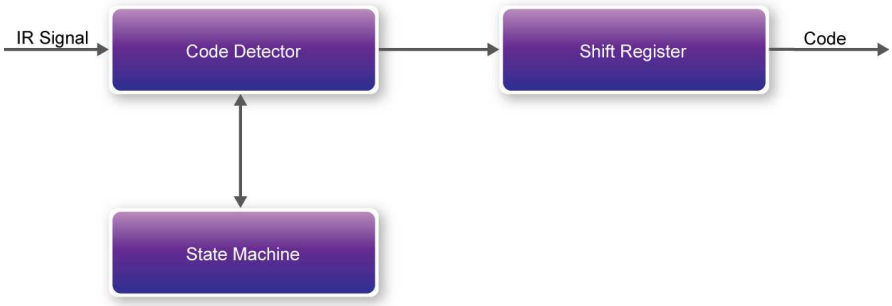


Figure 6-20 The IR Receiver controller

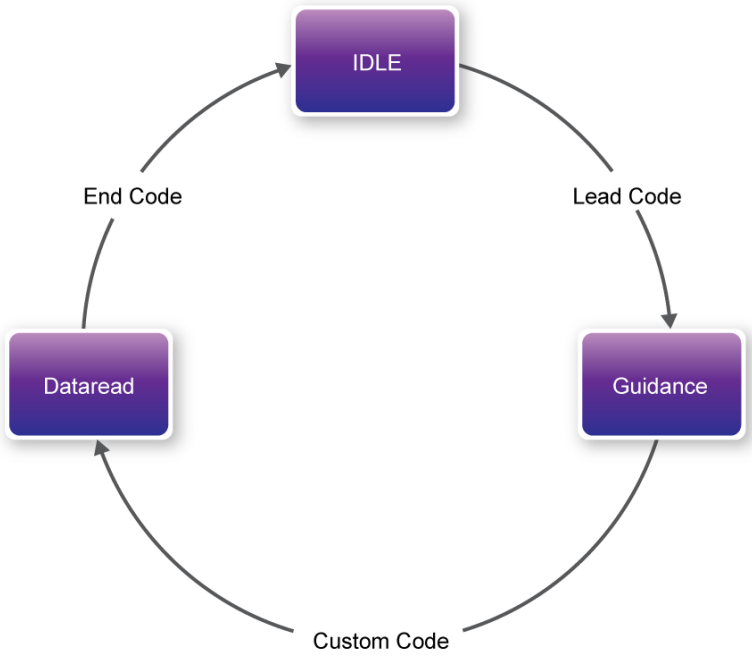


Figure 6-21 State shift diagram of State Machine

We can apply the IR receiver to many applications, such as integrating to the SD Card Demo, and you can also develop other related interesting applications with it.

■ Demonstration Setup, File Locations, and Instructions

- **Project directory: DE2_115_IR**
- **Bit stream used: DE2_115_IR.sof**
- **Load the bit stream into the FPGA by executing DE2_115_IR\demo_batch\DE2_115_IR.bat file**
- **Point the IR receiver with the remote-controller and press any button**

Table 6-4 shows how the received code and key data display on eight 7-segment displays.

Table 6-4 Detailed information of the indicators

Indicator Name	*Description*
HEX0	Inversed low byte of Key Code
HEX1	Inversed high byte of Key Code
HEX2	Low byte of Key Code
HEX3	High byte of Key Code
HEX4	Low byte of Custom Code
HEX5	High byte of Custom Code
HEX6	Repeated low byte of Custom Code
HEX7	Repeated high byte of Custom Code

Figure 6-22 illustrates the setup for this demonstration.

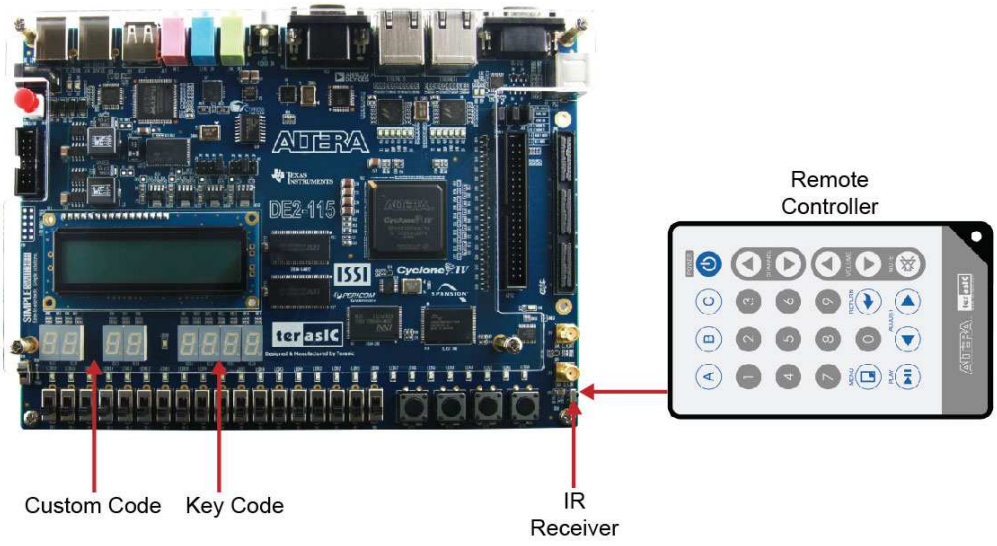


Figure 6-22 The Setup of the IR receiver demonstration

6.10 Music Synthesizer Demonstration

This demonstration shows how to implement a Multi-tone Electronic Keyboard using DE2-115 board with a PS/2 Keyboard and a speaker.

PS/2 Keyboard is used as the piano keyboard for input. The Cyclone IV E FPGA on the DE2-115 board serves as the Music Synthesizer SOC to generate music and tones. The VGA connected to the DE2-115 board is used to show which key is pressed during the playing of the music.

Figure 6-23 shows the block diagram of the design of the Music Synthesizer. There are four major blocks in the circuit: DEMO_SOUND, PS2_KEYBOARD, STAFF, and TONE_GENERATOR. The DEMO_SOUND block stores a demo sound for users to play; PS2_KEYBOARD handles the users' input from PS/2 keyboard; The STAFF block draws the corresponding keyboard diagram on VGA monitor when key(s) are pressed. The TONE_GENERATOR is the core of music synthesizer SOC.

Users can switch the music source either from PS2_KEYBOAD or the DEMO_SOUND block using SW9. To repeat the demo sound, users can press KEY1.

The TONE_GENERATOR has two tones: (1) String. (2) Brass, which is controlled by SW0. The audio codec used on the DE2-115 board has two channels, which can be turned ON/OFF using SW1 and SW2.

Figure 6-24 illustrates the setup for this demonstration.

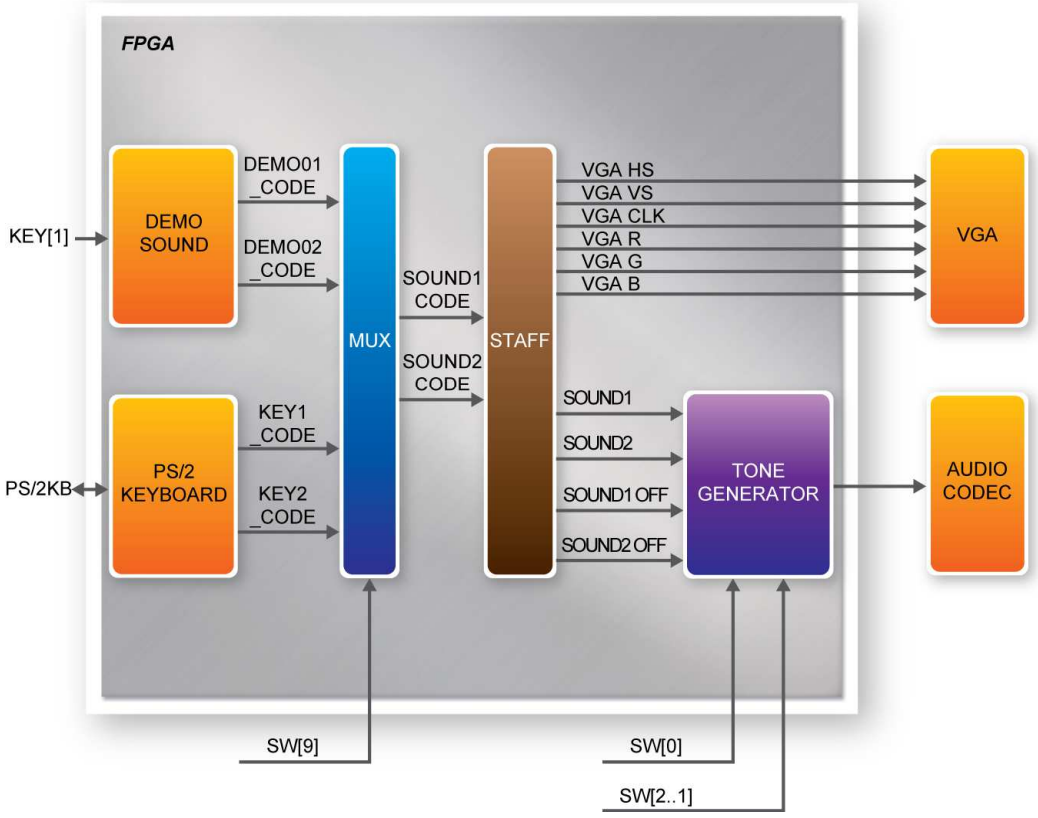


Figure 6-23 Block diagram of the Music Synthesizer design

■ Demonstration Setup, File Locations, and Instructions

- **Project directory: DE2_115_Synthesizer**
- **Bit stream used: DE2_115_Synthesizer.sof or DE2-115_Synthesizer.pof**
- **Connect a PS/2 Keyboard to the DE2-115 board.**
- **Connect the VGA output of the DE2-115 board to a VGA monitor (both LCD and CRT type of monitors should work)**
- **Connect the lineout of the DE2-115 board to a speaker.**
- **Load the bit stream into FPGA by executing DE2_115_Synthesizer\demo_batch\DE2_115_Synthesizer.bat file**
- **Make sure all the switches (SW[9:0]) are set to 0 (Down Position)**
- **Press KEY1 on the DE2-115 board to start the music demo**
- **Press KEY0 on the DE2-115 board to reset the circuit**
- **Note: If the HSMC loopback adapter is mounted, the I2C_SCL will be directly routed back to I2C_SDA. Because audio chip, TV decoder chip and HSMC share one I2C bus, therefore audio and video chip won't function correctly.**

Table 6-5 and Table 6-6 illustrate the usage of the slide switches, push-button switches (KEYs), PS/2 Keyboard.

- **Slide Switches and Push-buttons switches**

Table 6-5 Usage of the slide switches and push-buttons switches (KEYs)

Signal Name	*Description*
KEY[0]	Reset Circuit
KEY[1]	Repeat the Demo Music
SW[0]	OFF: BRASS, ON: STRING
SW[9]	OFF: DEMO, ON: PS/2 KEYBOARD
SW[1]	Channel-1 ON / OFF
SW[2]	Channel-2 ON / OFF

- **PS/2 Keyboard**

Table 6-6 Usage of the PS/2 Keyboard Keys

Signal Name	*Description*
Q	-#4
A	-5
W	-#5
S	-6
E	-#6
D	-7
F	1
T	#1
G	2
Y	#2
H	3
J	4
I	#4
K	5
O	#5
L	6
P	#6
:	7
“	+1

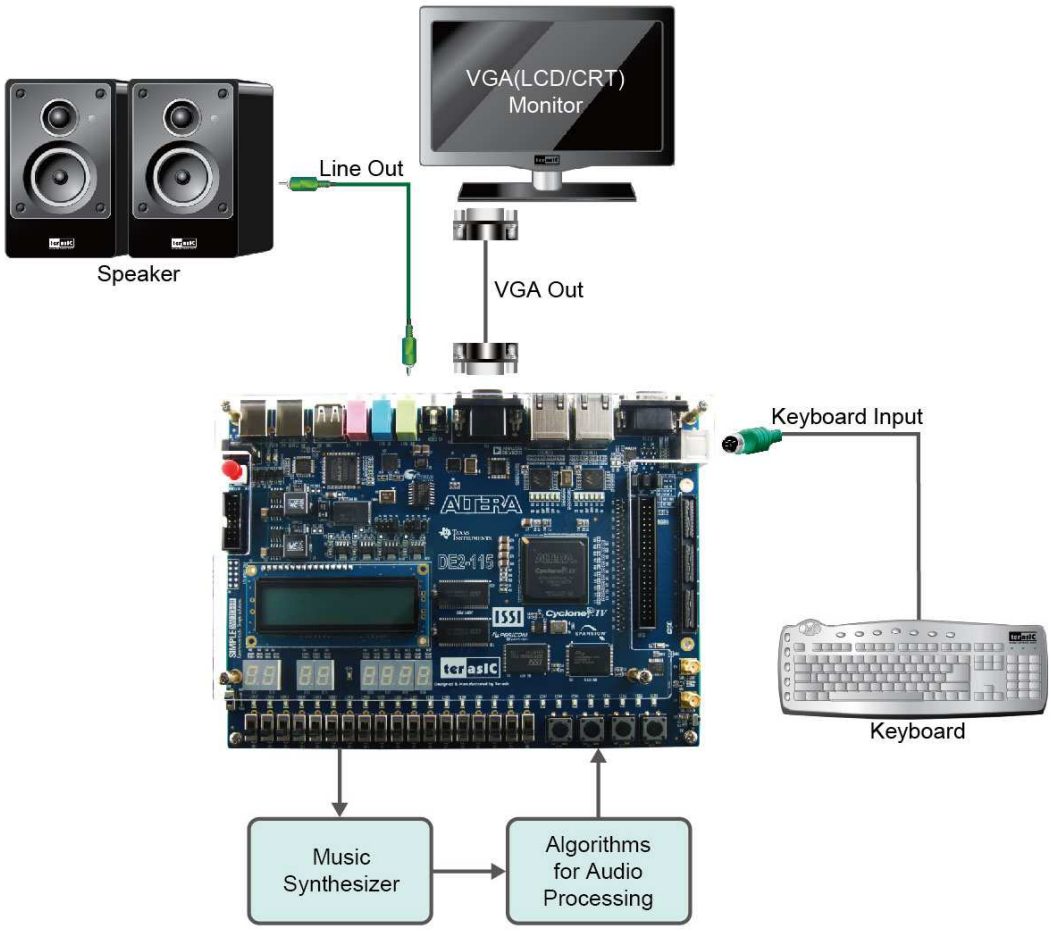


Figure 6-24 The Setup of the Music Synthesizer Demonstration

6.11 Audio Recording and Playing

This demonstration shows how to implement an audio recorder and player using the DE2-115 board with the built-in Audio CODEC chip. This demonstration is developed based on SOPC Builder and Nios II IDE. **Figure 6-25** shows the man-machine interface of this demonstration. Two push-buttons and six slide switches are used for users to configure this audio system: SW0 is used to specify recording source to be Line-in or MIC-In. SW1 is used to enable/disable MIC Boost when the recording source is MIC-In. SW2 is used to enable/disable Zero-Cross Detection for audio playing. SW3, SW4, and SW5 are used to specify recording sample rate as 96K, 48K, 44.1K, 32K, or 8K. The 16x2 LCD is used to indicate the Recording/Playing status. The 7-SEG is used to display Recording/Playing duration with time unit in 1/100 second. The LED is used to indicate the audio signal strength. **Table 6-7** and **Table 6-8** summarize the usage of Slide switches for configuring the audio recorder and player.

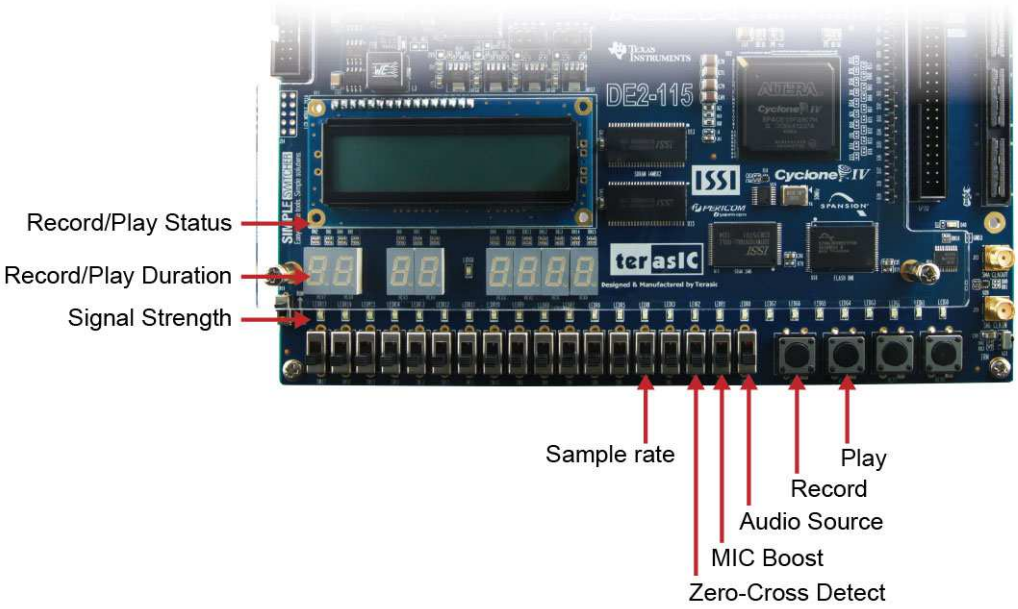


Figure 6-25 Man-Machine Interface of Audio Recorder and Player

Figure 6-26 shows the block diagram of the Audio Recorder and Player design. There are hardware and software parts in the block diagram. The software part stores the Nios II program in SRAM. The software part is built by Nios II IDE in C programming language. The hardware part is built by SOPC Builder under Quartus II. The hardware part includes all the other blocks. The "AUDIO Controller" is a user-defined SOPC component. It is designed to send audio data to the audio chip or receive audio data from the audio chip.

The audio chip is programmed through I2C protocol which is implemented in C code. The I2C pins from audio chip are connected to SOPC System Interconnect Fabric through PIO controllers. In this example, the audio chip is configured in Master Mode. The audio interface is configured as I2S and 16-bit mode. 18.432MHz clock generated by the PLL is connected to the XTI/MCLK pin of the audio chip through the AUDIO Controller.

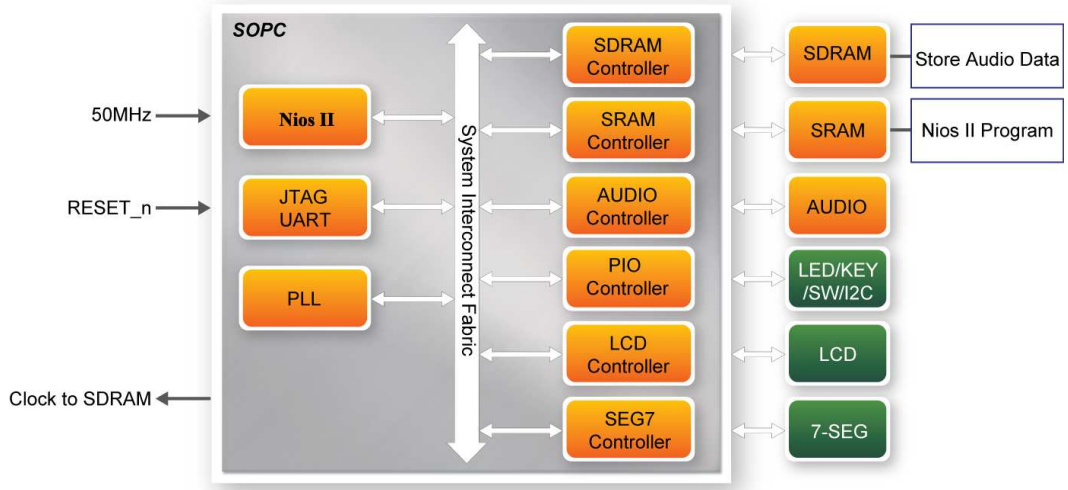


Figure 6-26 Block diagram of the audio recorder and player

■ Demonstration Setup, File Locations, and Instructions

- **Hardware Project directory: DE2_115_AUDIO**
- **Bit stream used: DE2_115_AUDIO.sof**
- **Software Project directory: DE2_115_AUDIO\software**
- **Connect an Audio Source to the LINE-IN port of the DE2-115 board.**
- **Connect a Microphone to MIC-IN port on the DE2-115 board.**
- **Connect a speaker or headset to LINE-OUT port on the DE2-115 board.**
- **Load the bit stream into FPGA. (note *1)**
- **Load the Software Execution File into FPGA. (note *1)**
- **Configure audio with the Slide switches.**
- **Press KEY3 on the DE2-115 board to start/stop audio recoding (note *2)**
- **Press KEY2 on the DE2-115 board to start/stop audio playing (note *3)**
-

Note:

(1). Execute DE2_115_AUDIO\demo_batch\audio.bat will download .sof and .elf files.
(2). Recording process will stop if audio buffer is full.
(3). Playing process will stop if audio data is played completely.
(4).If the HSMC loopback adapter is mounted, the I2C_SCL will be directly routed back to I2C_SDA. Because audio chip, TV decoder chip and HSMC share one I2C bus, therefore audio and video chip won't function correctly.

Table 6-7 Slide switches usage for audio source and signal processing setting

Slide Switches	*0 – DOWN Position*	*1 – UP Position*
SW0	Audio is from MIC	Audio is from LINE-IN
SW1	Disable MIC Boost	Enable MIC Boost
SW2	Disable Zero-cross Detection	Enable Zero-cross Detection

Table 6-8 Slide switch setting for sample rate switching for audio recorder and player

SW5 (0 – DOWN; 1- UP)	*SW4 (0 – DOWN; 1-UP)*	*SW3 (0 – DOWN; 1-UP)*	*Sample Rate*
0	0	0	96K
0	0	1	48K
0	1	0	44.1K
0	1	1	32K
1	0	0	8K
Unlisted combination			96K

6.12 Web Server Demonstration

This design example shows a HTTP server using the sockets interface of the NicheStack™ TCP/IP Stack Nios II Edition on MicroC/OS-II to serve web content from the DE2-115 board. The server can process basic requests to serve HTML, JPEG, GIF, PNG, JS, CSS, SWF, ICO files from the Altera read-only .zip file system. Additionally, it allows users to control various board components from the web page.

As Part of the Nios II EDS, NicheStack™ TCP/IP Network Stack is a complete networking software suite designed to provide an optimal solution for network related applications accompany Nios II.

Using this demo, we assume that you already have a basic knowledge of TCP/IP protocols.

The following describes the related SOPC system. The SOPC system used in this demo contains Nios II processor, On-Chip memory, JTAG UART, timer, Triple-Speed Ethernet, Scatter-Gather DMA controller and other peripherals etc. In the configuration page of the Altera Triple-Speed Ethernet Controller, users can either set the MAC interface as MII or RGMII as shown in **Figure 6-27** and **Figure 6-28** respectively.

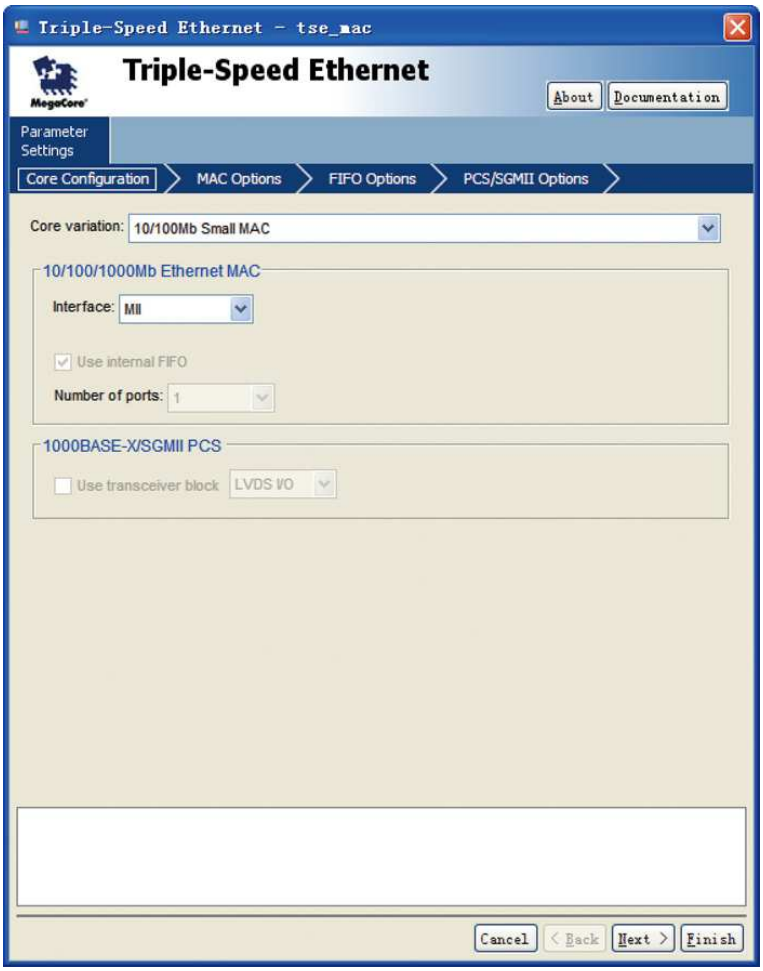


Figure 6-27 MII interface MAC Configuration

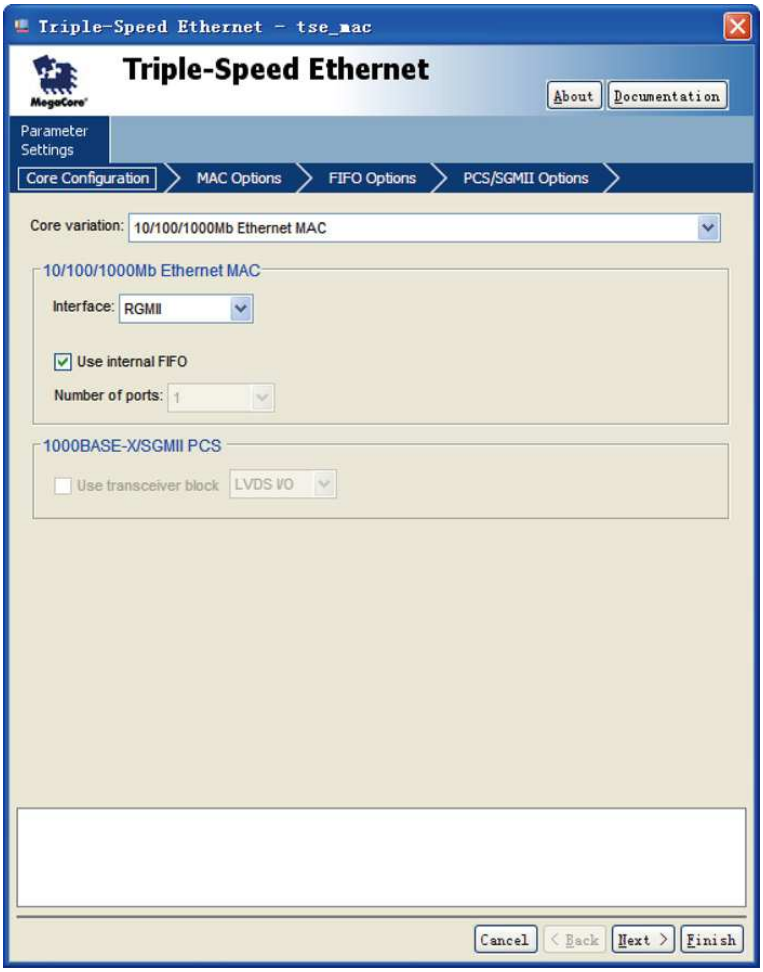


Figure 6-28 RGMII interface MAC Configuration

In the MAC Options tab (See **Figure 6-29**), users should set up proper values for the PHY chip 88E1111. The MDIO Module should be included, as it is used to generate a 2.5MHz MDC clock for the PHY chip from the controller's source clock(here a 100MHz clock source is expected) to divide the MAC control register interface clock to produce the MDC clock output on the MDIO interface. The MAC control register interface clock frequency is 100MHz and the desired MDC clock frequency is 2.5MHz, so a host clock divisor of 40 should be used.

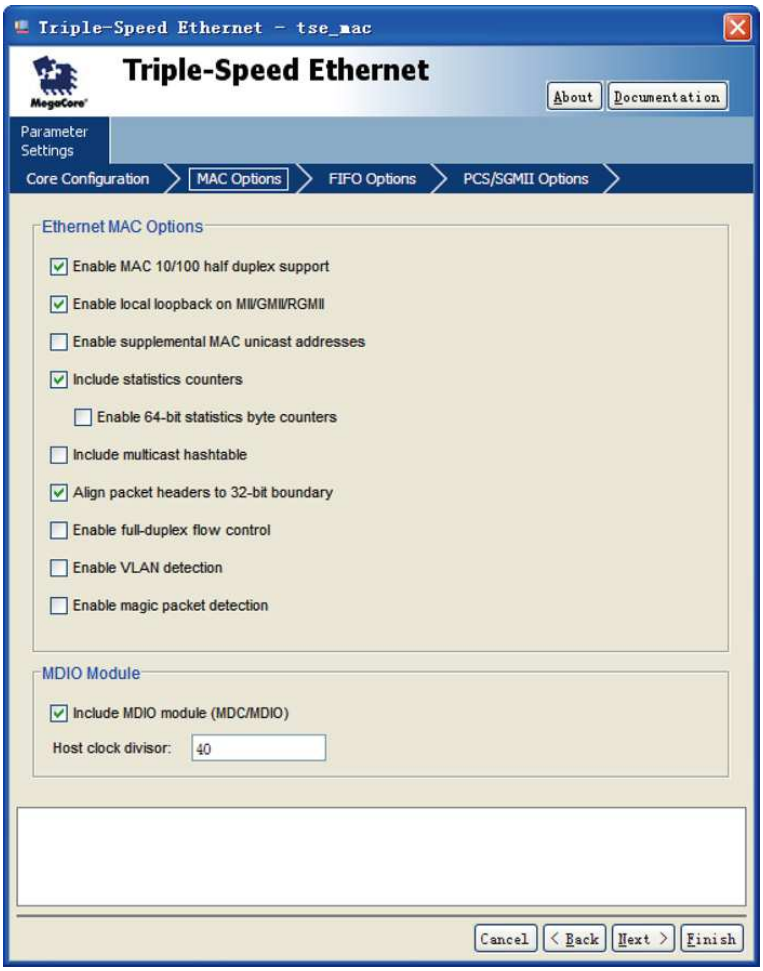


Figure 6-29 MAC Options Configuration

Once the Triple-Speed Ethernet IP configuration has been set and necessary hardware connections have been made as shown in **Figure 6-30**, click on generate.

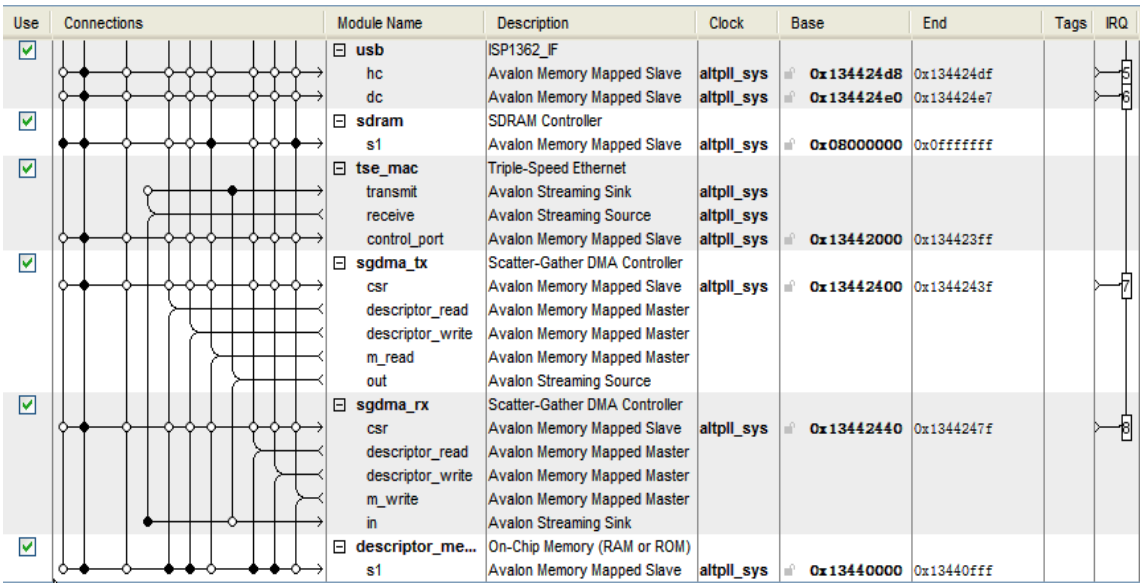


Figure 6-30 SOPC Builder

Figure 6-31 shows the connections for programmable 10/100Mbps Ethernet operation via MII.

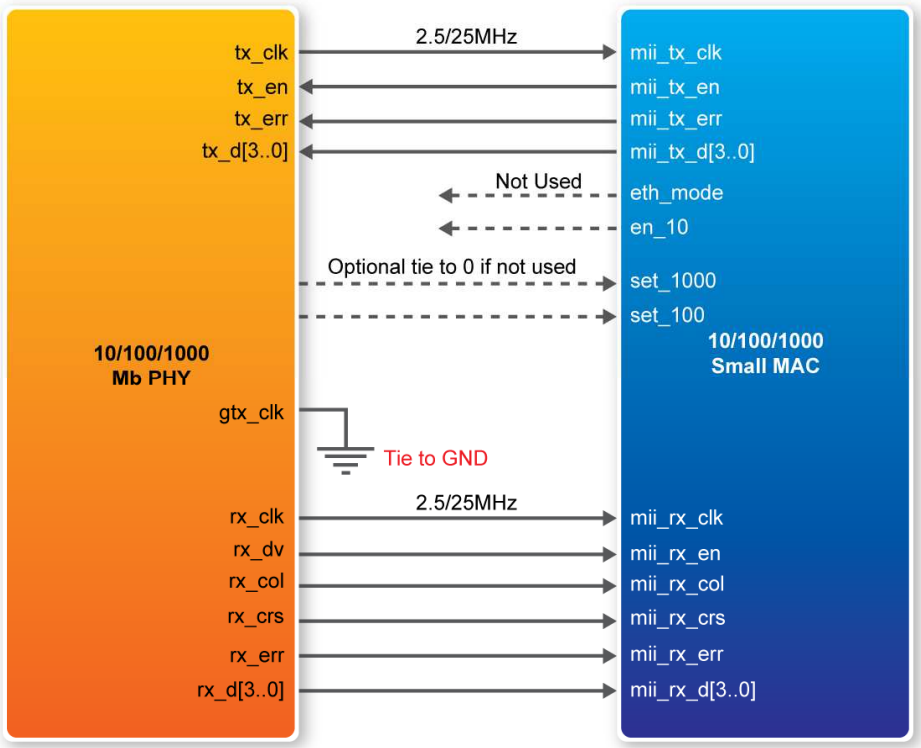


Figure 6-31 PHY connected to the MAC via MII

Figure 6-32 shows the connections for programmable 10/100/1000Mbps Ethernet operation via RGMII.

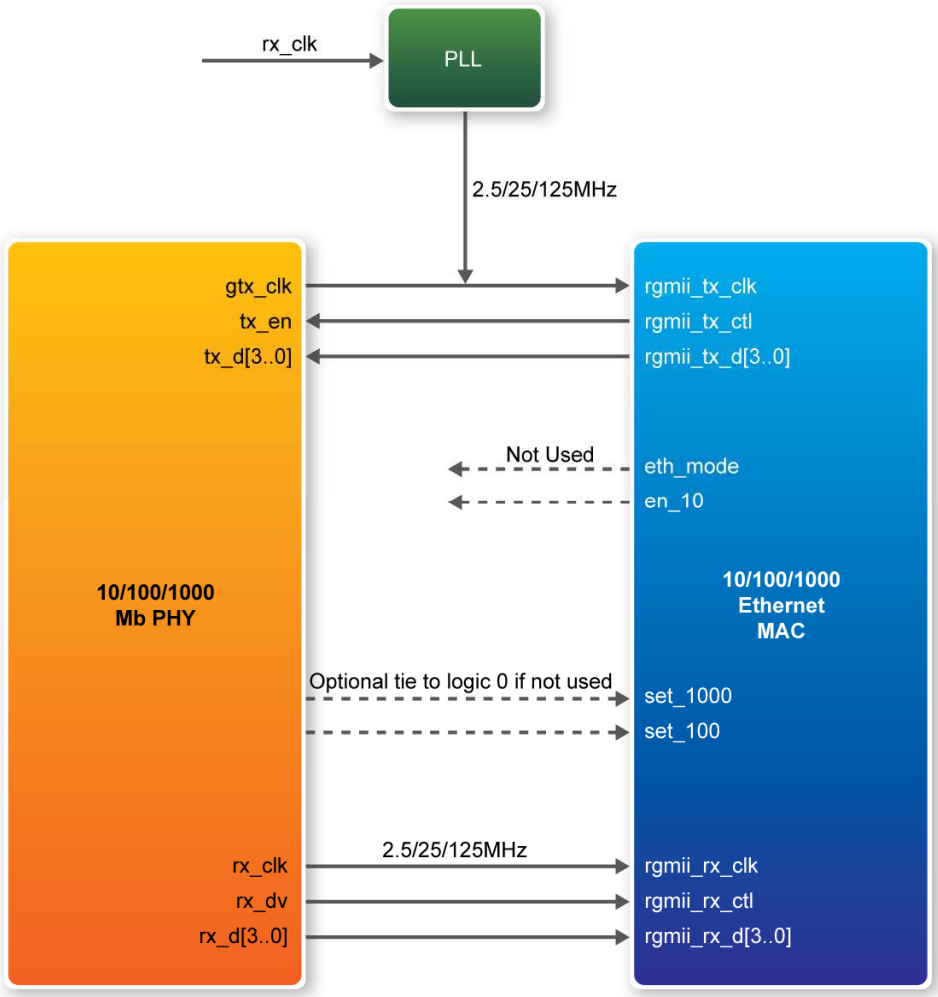


Figure 6-32 PHY connected to the MAC via RGMII

After the SOPC hardware project has been built, develop the SOPC software project, whose basic architecture is shown in **Figure 6-33**. The top block contains the Nios II processor and the necessary hardware to be implemented into the DE2-115 host board. The software device drivers contain the necessary device drivers needed for the Ethernet and other hardware components to work. The HAL API block provides the interface for the software device drivers, while the Micro C/OS-II provides communication services to the NicheStack™ and Web Server. The NicheStack™ TCP/IP Stack software block provides networking services to the application block where it contains the tasks for Web Server.

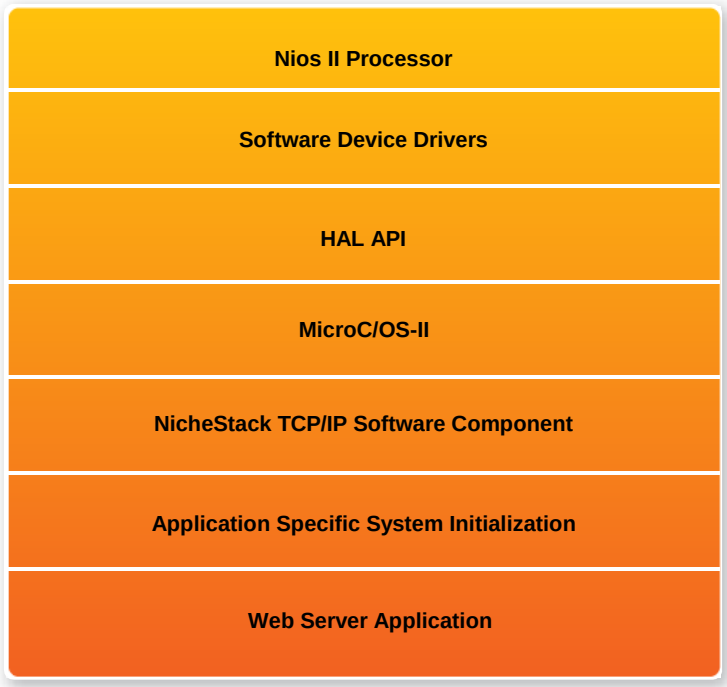


Figure 6-33 Nios II Program Software Architecture

Finally, the detail descriptions for Software flow chart of the Web Server program are listed in below:

Firstly, the Web Server program initiates the MAC and net device then calls the get_mac_addr() function to set the MAC addresses for the PHY. Secondly, it initiates the auto-negotiation process to check the link between PHY and gateway device. If the link exists, the PHY and gateway devices will broadcast their transmission parameters, speed, and duplex mode. After the auto-negotiation process has finished, it will establish the link. Thirdly, the Web Server program will prepare the transmitting and receiving path for the link. If the path is created successfully, it will call the get_ip_addr() function to set up the IP address for the network interface. After the IP address is successfully distributed, the NicheStack™ TCP/IP Stack will start to run for Web Server application.

Figure 6-34 describes this demo setup and connections on DE2-115. The Nios II processor is running NicheStack™ on the MicroC/OS-II RTOS.

Note: your gateway should support DHCP because it uses DHCP protocol to request a valid IP from the Gateway, or else you would need to reconfigure the system library to use static IP assignment. Furthermore, the web server demonstration uses the RGMII or MII interface to access the TCP/IP. You can switch the MAC

Interface via JP1 and JP2 for Ethernet 0 and Ethernet 1 respectively. Table 6-9 shows the project name of web server demonstration for each Ethernet Port and working mode.

Table 6-9 Demo Directory Paths

PHY / *project directory* / *Interface*	*ENET0*	*ENET1*
RGMII interface	**DE2_115_Web_Server\ DE2_115_WEB_SERVER_RGMII_ENET0**	**DE2_115_Web_Server\ DE2_115_WEB_SERVER_RGMII_ENET1**
MII interface	**DE2_115_Web_Server\ DE2_115_WEB_SERVER_MII_ENET0**	**DE2_115_Web_Server\ DE2_115_WEB_SERVER_MII_ENET1**

■ Demonstration Setup, File Locations, and Instructions

The Following steps describe how to setup a Web Server demonstration on the ENET0 in RGMII mode. Before running the demonstration, please copy the demonstration file to your host PC from "DE2_115_demonstrations" folder on the DE2-115 System CD.

Demonstration Source Code

- **Project directory: DE2_115_Web_Server\ DE2_115_WEB_SERVER_RGMII_ENET0**
- **Bit stream used: DE2_115_WEB_SERVER.sof**
- **Nios II Workspace: DE2_115_WEB_SERVER_RGMII_ENET0 \Software**

Demonstration Batch File

Demo batch file Folder for this demonstration:

DE2_115_Web_Server\ DE2_115_WEB_SERVER_RGMII_ENET0 \demo_batch

The demo batch file includes the following files:

- **Batch File: DE2_115_WEB_SERVER.bat, DE2_115_WEB_SERVER_bashrc**
- **FPGA Configure File : DE2_115_WEB_SERVER.sof**

- **Nios II Program: web_server.elf**

Demo batch file folder for writing web site content zip file into flash:

DE2_115_Web_Server\ WEB_SERVER_FLASH

The demo batch file includes the following files:

- **Batch File: WEB_SERVER_FLASH.bat , WEB_SERVER_FLASH_bashrc**
- **FPGA Configure File : WEB_SERVER_FLASH.sof**

Demonstration Setup

- Make sure the PHY device is working on RGMII mode (Short pin 1 and pin 2 of JP1)
- Plug a CAT 5e cable into the Ethernet port (J4) on the DE2-115 board (Make sure the Ethernet cable is connected to router and the DHCP function is supported)
- Power on DE2-115 board
- Execute the demo batch file "WEB_SERVER_FLASH.bat" to write web site content into flash on DE2-115 board
- Execute the demo batch file "DE2_115_WEB_SERVER.bat " for downloading .sof and .elf file for this demonstration.
- See **Figure 6-35**, when the message "Web Server Starting up" is shown on nios2-terminal. The LCD on DE2-115 board will show the valid IP address (It may take a while to get IP address from Gateway)
- Once the LCD on DE2-115 board shows the valid IP address got from Gateway, then launch your web browser
- Input the IP into your browser. (IP is shown on the LCD display)
- You will see the brand new DE2-115 webpage on your computer
- On the web page, you could access the DE2-115 board's peripherals from the left sidebar or link to external pages from the right sidebar. Try check some LEDs on the left sidebar and then press send will light up the specified LEDs on board. You also could send text to the LCD or set the value for 7-segment displays on DE2-115 board. **Figure 6-36** gives a snapshot of the web server page
- To perform the demonstrations of other mode or Ethernet port, please refer to **Table 4-21** and **Table 4-22** for mode setting and running the associate project on the "DE2_115_demonstrations\ DE2_115_Web_Server" folder in the DE2-115 System CD

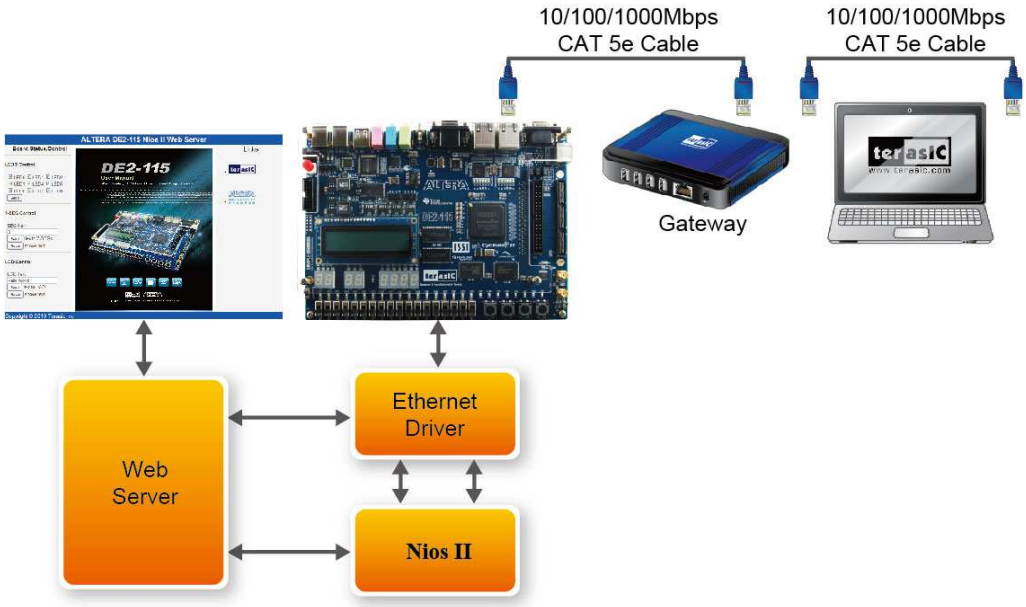


Figure 6-34 System Principle Diagram

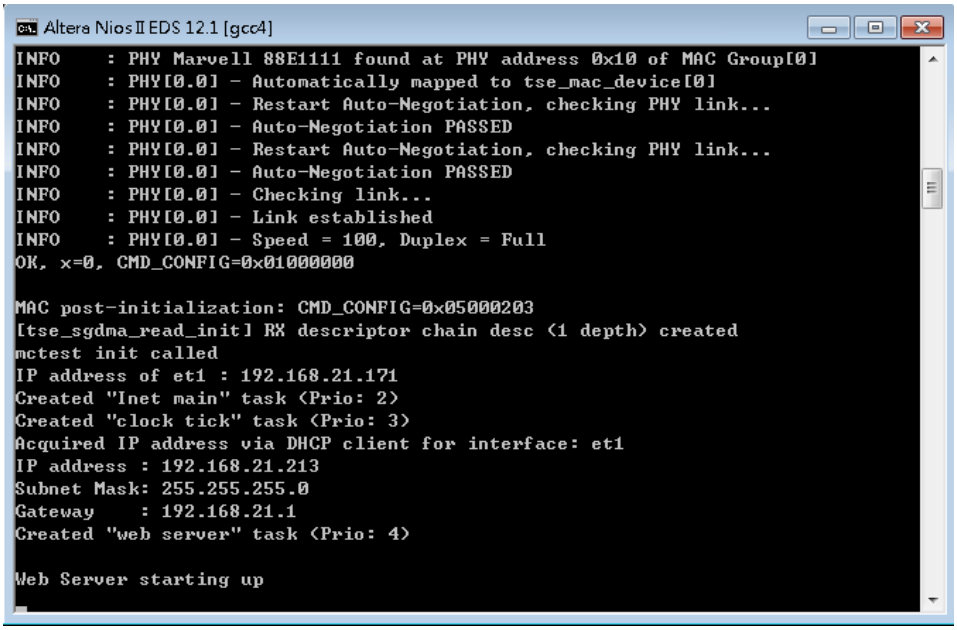


Figure 6-35 Web Server Starting Success



Figure 6-36 Served web page for DE2-115

Chapter 7

Appendix

7.1 EPCS Programming via nios-2-flash-programmer

Before programming the EPCS via nios-2-flash-programmer, users must add an EPCS patch file nios-flash-override.txt into the Nios II EDS folder. The patch file is available in the folder DE2_115_demonstrations\EPCS_Patch of DE2-115 System CD. Please copy this file to the folder [QuartusInstalledFolder]\nios2eds\bin (e.g. C:\altera\11.1\nios2eds\bin)

If the patch file is not included into the Nios II EDS folder, an error will occur as shown in the **Figure 7-1.**

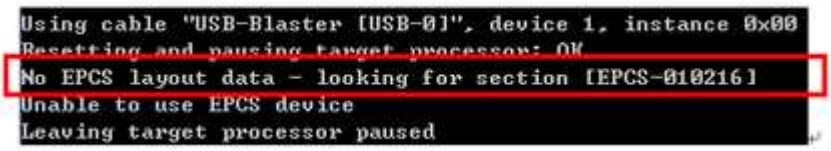


Figure 7-1 EPCS Message

7.2 Revision History

Version	*Change Log*
V1.0	Initial Version (Preliminary)
V1.01	VGA Vertical Timing table correction SDRAM Table reference modification
V1.02	Modify Table4-15 Header info.
V1.03	Modify section 6.3 demo description
V1.04	Added EPCS information
V2.0	Replace USB Chip from ISP1362 to CY67200
V2.1	Modify Table 4-9, change pin description of HSMC_CLKIN_N , HSMC_CLKIN_P , HSMC_CLKOUT_N and HSMC_CLKOUT_P

7.3 Copyright Statement

Copyright © 2012 Terasic Technologies. All rights reserved.

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP