# DE2-115

## User Manual

terasIC
www.terasic.com

ALTERA

Copyright © 2003-2010 Terasic Technologies Inc. All Rights Reserved.

# 目录


**第 1 章. 关于 DE2-115 工具包 ..... 4**

1.1 包装内容 ..... 4

1.2 DE2-115 的组装 ..... 5

1.3 获得帮助 ..... 5

**第 2 章. DE2-115 开发板简介 ..... 7**

2.1 开发板布局和组件 ..... 7

2.2 DE2-115 系统框图 ..... 9

2.3 DE2-115 上电 ..... 13

**第 3 章. DE2-115 控制面板 ..... 15**

3.1 控制面板初始化 ..... 15

3.2 控制LED灯, 7段数码显示管 和 LCD显示器 ..... 17

3.3 开关与按钮 ..... 19

3.4 SDRAM/SRAM/EEPROM/Flash 控制器和编程器 ..... 20

3.5 USB 监测 ..... 22

3.6 PS/2 设备 ..... 22

3.7 SD 卡 ..... 23

3.8 RS-232 通信 ..... 24

3.9 VGA ..... 25

3.10 HSMC ..... 26

3.11 红外接收器 ..... 27

3.12 DE2-115 控制面板的整体结构 ..... 28

**第 4 章. 使用 DE2-115 ..... 30**

4.1 配置 Cyclone IV E FPGA 芯片 ..... 30

4.2 使用按钮开关和拨动开关 ..... 33

4.3 使用 LED ..... 34

4.4 使用七段数码管 ..... 36
4.5 时钟电路 ..... 38
4.6 使用 LCD 模块 ..... 39
4.7 HSMC 接口 ..... 41
4.8 使用通用扩展接头 ..... 45
4.9 使用 14 脚扩展口 ..... 49
4.10 使用 VGA ..... 50
4.11 使用 24 比特音频编解码芯片 ..... 52
4.12 RS-232 串口 ..... 53
4.13 PS/2 接口 ..... 54
4.14 千兆以太网接口 ..... 55
4.15 TV 解码器 ..... 58
4.16 TV 编码器实现 ..... 59
4.17 使用 USB 界面 ..... 60
4.18 使用 IR 模块 ..... 61
4.19 使用 SRAM/SDRAM/Flash/EEPROM/SD 卡 ..... 62

**第 5 章. DE2-115 系统生成器 ..... 69**

5.1 简介 ..... 69
5.2 一般设计流程 ..... 69

**第 6 章. 高阶设计范例 ..... 76**

6.1 DE2-115 默认配置 ..... 76
6.2 TV 电视盒设计范例 ..... 77
6.3 USB 画笔 ..... 79
6.4 USB 设备 ..... 81
6.5 卡拉 OK 机 ..... 83
6.6 SD 卡设计范例 ..... 85
6.7 SD 卡音乐播放器 ..... 88
6.8 PS/2 鼠标控制器设计范例 ..... 92
6.9 IR 接收器设计范例 ..... 95
6.10 音乐合成器设计范例 ..... 99

6.11 音乐录制和回放设计范例 ..... 101
6.12 网页服务器设计范例 ..... 104

**第 7 章. 附录 ..... 114**

7.1 修改历史 ..... 114
7.2 版权声明 ..... 114

# 第1章.

# 关于 DE2-115 工具包

DE2-115 套装包含了所有使用开发板会用到的器件资源，您额外需要的仅是一台装有微软视窗操作系统的个人电脑。

## 1.1 包装内容

图 1-1 给出了 DE2-115 的图片。

图 1-1 DE2-115 包装内容

DE2-115 工具包包括以下内容：

- DE2-115 主板
- 用于 FPGA 编程和控制的 USB 电缆
- 一片 DE2-115 系统 CD，包括 DE2-115 相关文档和辅助材料。具体来说有使用手册、DE2-115 控制面板、系统生成器、Altera 的监控程序实用工具、参考设计、演示实例、元器件数据手册、指导手册和一系列的实验课素材。

- CD-ROMs 包括 Altera’s Quartus® II 网络版 和 Nios® II Embedded Design 评估版
- 附件包一个，内含 6 个橡胶套，若干扩展销（便于探测与测试开发板上的 I/O 扩展口）
- 装配在主板上的树脂玻璃片
- 12 V 直流电源
- 遥控器

## 1.2 DE2-115 的组装

组装橡皮脚

- 如图 1-2 所示，给板子上的 6 个铜脚各套上一个橡皮脚
- 通过板子上额外的支脚和螺丝装上树脂玻璃片，可以为板子提供额外的保护

图 1-2 DE2-115 开发板的基脚

## 1.3 获得帮助

遇到问题可以从以下几处获得帮助：

- Altera Corporation
- 101 Innovation Drive San Jose, California, 95134 USA
- Email: university@altera.com

- Terasic Technologies（友晶科技台湾总部）
- No. 356, Sec. 1, Fusing E. Rd. Jhubei City, HsinChu County, Taiwan, 302
- Email: support@terasic.com
- Tel.: +886-3-550-8800
- Web: DE2-115.terasic.com

# 第2章.

# DE2-115 开发板简介

这一个章节将带您熟悉 DE2-115 板上的每个部分及其特性。

## 2.1 开发板布局和组件

图 2-1 和图 2-2 给出了 DE2-115 开发板的全貌。它描述了开发板的布局，并标注出连接器和关键部件的位置。

图 2-1 DE2-115 开发板 (背面)

图 2-2 DE2-115 开发板 (反面)

DE2-115 开发板拥有能够为用户实现广泛设计的特征，包括从简单的设计到各种多媒体电路的设计。

DE2-115 开发板包括以下硬件资源：

- Altera Cyclone® IV 4CE115 FPGA 芯片
- Altera 串行配置芯片－ EPCS64
- 板上 USB Blaster 下载电路 ，同时支持 JTAG 模式和 AS 模式
- 2MB SRAM
- 2 片 64MB SDRAM
- 8MB 闪存
- SD 卡插槽
- 4 个按钮开关
- 18 个滑动开关
- 18 个红色 LEDs
- 9 个绿色 LEDs

- 50MHz 晶振提供给时钟源
- 24-bit CD 品质 CODEC 芯片，具备线路输入、输出、麦克风输入接口
- VGA DAC (8-比特高速三通道 DACs) 带有 VGA 输出接口
- TV 解码器 (NTSC/PAL/SECAM) 和 TV 输入接口
- 2 千兆以太网 PHY 带 RJ45 连接器
- 带有 A 类和 B 类 USB 接口的 USB 主从控制器
- RS-232 收发器和 9 针连接器
- PS/2 鼠标/键盘接口
- IR 收发器
- 2 个 SMA 接头，用于外部时钟输入/输出
- 1 个 40 引脚扩展接口，带二极管保护电路
- 1 个 HSMC 连接器
- 16x2 LCD 模组

除了这些硬件功能外，DE2-115 开发板还支持标准 I/O 接口和用于评估各项组件的控制面板等软件工具。该软件也提供用于验证 DE2-115 开发板高级功能的大量实例演示。

为确保正常使用 DE2-115 开发板，用户必须先熟悉 Quartus II 软件。一些基础知识您可以通过 *"Getting Started with Altera's DE2-115 Board"* (tut_initialDE2-115.pdf) 和 *"Quartus II Introduction"*（根据使用者键入设计的方法有三个版本：Verilog, VHDL 或者 schematic）来获取。这两个教程在 DE2-115 随附的系统盘中" DE2_115_tutorials"有提供，您也可以在友晶科技 DE2-115 网页上得到.

## 2.2 DE2-115 系统框图

图 2-3 展示了 DE2-115 的系统框图 。为了给用户提供最大的便利性，所有的连接器都是通过 Cyclone IV E FPGA 器件来完成的。因此，使用者可以通过调试 FPGA 来实现任何系统设计。

图 2-3 DE2-115 系统框图

接下来是关于图 2-3 中各功能块更详细的信息：

## FPGA 器件

- Cyclone IV EP4CE115F29 器件
- 114,480 个逻辑单元
- 432 M9K 内存模块
- 3,888 Kbits 嵌入式存储器位
- 4 个锁相环

## FPGA 配置

- 同时支持 JTAG 模式和 AS 模式
- 提供系列配置器件——EPCS64
- 内建 USB Blaster 电路

## 存储器配置

- 128MB (32Mx32bit) SDRAM
- 2MB (1Mx16) SRAM
- 8 位 8MB (4Mx16) Flash 存储器，配置为 8-bit 工作模式
- 32Kb EEPROM

## SD 卡接口

- 提供 SPI 模式 和 4 位 SD 模式用于 SD 卡接入

## 连接器

- 2 个 10/100/1000 以太网接口
- 1 个 HSMC
- 可配置的 I/O 标准 (电平：3.3/2.5/1.8/1.5V)
- A 型和 B 型 USB 接口
  - 完全兼容 USB 2.0 的主从控制器
  - 支持全速和低速数据传输
  - 可用于 PC 驱动
- 40 针扩展口
- 可配置的 I/O 标准 (电平：3.3/2.5/1.8/1.5V)
- VGA 输出接口

- VGA DAC (三通道高速视频 DACs)
- 带有流控制的 RS-232 界面，提供 DB9 连接接口
- 提供 PS/2 鼠标/键盘 连接器

## 时钟输入

- 3 个 50MHz 晶振
- 1 个 SMA 外部时钟输入

## 音频 CODEC

- 24 位编码器/解码器 CODEC
- 包括线路输入，线路输出和麦克风输入

## 显示输出

- 16x2 LCD 模组

## 开关和七段数码管

- 18 个滑动开关和 4 个按钮开关
- 18 个红色和 9 个绿色 LEDs
- 8 个七段数码管

## 其他特征

- 红外遥控接收模块

- TV 解码器 (NTSC/PAL/SECAM 制式) 和 TV-in 连接器

## 电源

- 直流电源输入
- 开关和降压调节器 LM3150MH
- 
- 

## 2.3 DE2-115 上电

DE2-115 开发板带有一个出厂默认配置数据来演示板子的某些功能。此配置数据还可以让用户快速验证板子是否正常工作。请执行以下步骤开启 DE2-115：

1. 用包装盒里的 USB 电缆将 PC 的 USB 端口和 DE2-115 开发板的 USB Blaster 连接器连接起来。 为了实现主机和开发板之间的通讯，必须安装 USB Blaster 驱动软件。如该驱动还没有在主机上安装，您可以参考教材 “Getting Started with Altera's DE2-115 Board ” (tut_initialDE2-115.pdf).。该教材可以在 DE2-115 系统 CD 目录 DE2_115_tutorials 中得到。
2. 在将 12V 的适配器连接到 DE2-115 主板前，按下红色的 ON/OFF 开关确保电源断开。
3. 将一个 VGA 显示器连接到 DE2-115 开发板的 VGA 接口上。
4. 将耳机插到 DE2-115 主板上的音频输出端口。
5. 将 DE2-115 主板左侧的 RUN/PROG 开关(SW19)调到 RUN 的状态； PROG 状态仅仅适用于 AS 模式的编程。
6. 按下 DE2-115 主板上红色的电源开关 OFF /ON 重新上电。

此时，您应该观察到以下现象：

- 所有的 LEDs 都在闪烁
- 7 段显像管都是循环显示 0 到 F
- LCD 显示“Welcome to the Altera DE2-115”
- VGA 显示器显示图 2-4 中的图像

- 将滑动开关 SW17 拨到 DOWN 位置; 你会听到一种 1-kHz 的声音。请注意，默认输出的音量比较大，为了避免任何损伤，请将耳机音量调至最低
- 将滑动开关 SW17 拨到 UP 位置 ，并将音频播放器输出端口插到 DE2-115 主板的线路输入连接器，通过话筒和耳机，您会听到从该音频播放器（MP3, PC, iPod, 或者类似的设备）中播放出来的音乐
- 您可以将麦克风接入 DE2-115 开发板的麦克风输入插口， 您的声音就会和音频播放器的音乐混合到一起了

**图 2-4 默认的 VGA 输出模式**

# 第3章.

# DE2-115 控制面板

DE2-115 开发板附带的控制面板软件允许用户经由 PC 主机访问板上的各种组件。主机通过 USB 与开发板通信。该软件可以被用来测试开发板上的各组件功能或作为开发 RTL 代码时的调试工具。

本章首先介绍了控制面板的一些基本功能，然后以框图的形式，描述了其结构组成。

## 3.1 控制面板初始化

控制面板应用软件位于 DE2-115 系统 CD 中的 DE2_115_tools\DE2_115_control_panel 目录下，它无需安装，只要将整个目录复制到您的电脑上并运行 DE2_115_ControlPanel.exe 即可。（Windows 7 64 位的用户：如果在启动控制面板软件时弹出 a missing jtag_client.dll file (cannot find jtag_client.dll)的报错，请您从以下目录/DE2_115_tools\DE2_115_control_panel\win7_64bits 重新运行 DE2-115_ControlPanel.exe）

在控制面板软件要求 FPGA 开发板运行指定的任务之前，特定的控制电路将会下载至开发板中。该应用程序将自动启用 Quartus II 工具，并经由 USB-Blaster[USB-0] 连接下载控制电路至 FPGA 开发板。

请运行如下步骤激活控制面板应用程序:

1. 确认 Quartus II 10.0 或者更高的版本已成功安装至您的电脑。
2. 将 RUN/PROG 开关置于 RUN 档。
3. 将提供的 USB 电缆线接到 USB Blaster 端口，连接 12V 电源，然后开启 DE2-115 电源开关
4. 启动主机上的 DE2_115_ControlPanel.exe 程序，控制面板的用户界面就会出现，如图 3-1 所示
5. 一旦启动 DE2_115_ControlPanel.exe 程序， DE2_115_ControlPanel.sof 比特流(bit stream)将自动加载至开发板
6. 如果界面上显示未连接的状态，点击 CONNECT，.sof 文件将重新加载至开发板

7. 需要注意的是，控制面板将会一直占用 USB 端口直至你关闭这些端口；在未关闭控制面板的 USB 端口之前，你将不能使用 Quartus II 下载配置文件至 FPGA

8. 控制面板程序已经完成了安装，用户可以设置一些 LED 灯的 ON/OFF 状态，并观察 DE2-115 开发板上的显示结果

**图 3-1 DE2-115 控制面板**

DE2-115 控制面板示意图如图 3-2 所示。实现控制功能的’控制电路”将在 FPGA 开发板内运行。它将通过 USB Blaster 连接来和主机上的 Control Panel 窗口进行通信。GUI 界面发出命令到控制电路。它将会处理所有的应用需求及实现电脑与 DE2-115 开发板之间的数据传输。

图 3-2 DE2-115 控制面板示意图

DE2-115 控制面板可以用来点亮 LED 灯，改变显示在 7 段数码管和 LCD 显示器上的数值、监测按钮/开关的状态、读/写 SDRAM、SRAM、EEPROM 和 Flash 内存、监测 USB 设备的状态、与 PS/2 鼠标通信、 输出 VGA color pattern 至 VGA 显示器、验证 HSMC I/O 的功能,经由 RS-232 接口与 PC 通信和读取 SD 卡规格信息。读取/写入一个字或整个文件的功能允许用户进行多媒体应用的开发(Flash 音频播放器，(Flash 图片查看器) 而不用去担忧如何创建一个内存编程器。

## 3.2 控制 LED 灯，七段数码显示管 和 LCD 显示器

控制面板的一个简单功能就是允许设置显示在 LED 灯、7 段数码显示管和 LCD 字符显示器的上的数值。

单击 LED 标签，如图 3-3 所示窗口即会出现。你可以直接分别点亮或熄灭所有的 LED 灯或者点击“Light All” 或者“Unlight All”。

图 3-3 控制 LED 灯

单击 7-SEG 标签，如图 3-4 所示窗口即会出现。从该窗口我们可以观察到，直接使用左-右箭头即可控制七段数码管的显示数值，并且这些更新会立即在 DE2-115 开发板的相应位置显示出来。注意：七段数码管的点在 DE2-115 开发板上是不可用的。

图 3-4 控制 7 段数码显示管

单击 LCD 标签，如图 3-5 所示窗口即会出现。直接输入文本至 LCD 文本框并按下 Set 按钮，可将文本写入 LCD 显示器。

图 3-5 控制液晶显示器

将任意值输出显示简单显示设备上的功能并不是典型的设计应用所必需的。但是，它却为用户提供了一个简单的机制去验证这些设备的功能是否正常，特别是怀疑某个设备出现故障的时候。因此，它可以用于故障排除。

## 3.3 开关与按钮

单击开关标签，图 3-6 的窗口就会出现。该功能可以实时侦测到滑动开关和按钮的状态并将这些状态显示在 GUI 界面上。它也可以用来验证滑动开关与按钮的功能。

图 3-6 侦测开关和按钮

检查按钮和滑动开关的状态并不是典型设计应用中所必需的。 但是，它却为用户提供了一个简单的方法去验证按钮和开关的功能是否正常。因此，它可以用于故障排除。

## 3.4 SDRAM/SRAM/EEPROM/Flash 控制器和编程器

控制面板可以用于向/从 DE2-115 board 上的 SDRAM, SRAM, EEPROM 和 Flash 芯片中写/读数据。我们将以 SDRAM 为例，说明如何访问 SDRAM ；同样的方法也适用于 SRAM, EEPROM 和 Flash。单击 Memory 标签并选择 “SDRAM”，如图 3-7 所示的窗口即会出现。

图 3-7 访问 SDRAM

通过输入所需写入位置的地址，指定要写入的数据，并按下 Write 按钮，一个 16 位字就可以被写入 SDRAM 中。 按下 Read 按钮可读取该位置写入的内容。图 3-7 显示了写入 16 进制 06CA 数值至偏移地址 200 和读取该地址内容的结果。

控制面板的连续写功能可将一个文件的内容写入 SDRAM，具体步骤如下:

1. 在 Address 栏指定其起始地址。
2. 在 Length 栏指定写入的字节数。 如果是加载整个文件，须勾选文件长度而不是给出字节数。
3. 单击 Write a File to Memory 按钮，启动写入操作。
4. 当控制面板以标准的 Windows 对话框响应并要求指定源文件，请选择常见格式的文件作为输入。

控制面板也支持加载.hex 扩展名的文件。.hex 扩展名的文件是使用 ASCII 字符来表示十六进制值以指定内存值的 ASCII 文本文件。例如，一个包含 0123456789ABCDEF 的文件定义了 8 个 8 位值: 01, 23, 45, 67, 89, AB, CD, EF。这些值将会被连续加载至存储器中。

控制面板的连续读功能可读取 SDRAM 的内容并将它们填写至文件中，具体步骤如下：

1. 在 Address 栏指定其起始地址。
2. 在 Length 栏指定复制到文件的字节数。如果将要复制整个 SDRAM 中的内容(共计 128

Mbytes)，勾选 Entire Memory 框。

3. 单击 Load Memory Content to a File 按钮。
4. 当控制面板以标准的 Windows 对话框响应并要求指定目的文件时，请选择常见格式的文件。

用户可以用类似的方法去访问 SRAM, EEPROM 和 Flash。 需要注意的是，在向 Flash 写入数据之前，用户必须先做 erase。

## 3.5 USB 监测

控制面板给用户提供了一个 USB 监测工具，用来监测连接到 DE2-115 开发板的 USB 端口的设备类型。通过向开发板的 USB host 端口插入 USB 设备，设备类型就会显示在控制面板的窗口上。图 3-8 显示的是侦测到一个 USB 鼠标插入至 USB host 端口。

图 3-8 USB 鼠标监测工具

## 3.6 PS/2 设备

控制面板给用户提供了一个 PS/2 监测工具以监测连接到 DE2-115 开发板上一个 PS/2 鼠标的实时状态。鼠标的移动和三个按钮的状态将显示在图形和文字界面。 鼠标的运动状态被转化为在(0,0)~(1023,767)范围内的一个坐标(x,y)。该功能还可用于验证的 PS / 2 连接功能是否正常。

请遵照如下步骤运行 PS/2 鼠标监测工具:

1. 单击 PS/2 标签，如图 3-9 所示窗口即会出现。
2. 连接一个 PS/2 鼠标至 DE2-115 开发板的 PS/2 端口。
3. 单击 Start 按钮以启动 PS/2 鼠标监测，之后该按钮选项就由 Start 变为 Stop。在监测过程中，PS/2 鼠标的状态不断被更新并实时显示在 控制面板的 GUI 窗口。单击 Stop 按钮即可终止监测。

图 3-9 PS/2 鼠标监测工具

## 3.7 SD 卡

该功能可用于读取 SD 卡的标识符与规格信息。采用 4 位的 SD 模式来访问 SD Card。该功能还可用于验证 SD 卡接口的功能。请遵照如下步骤运行 SD Card 功能:

1. 单击 SD Card 标签，如图 3-10 所示窗口即会出现
2. 插入一张 SD 卡至 DE2-115 开发板，然后按下 Read 按钮以读取 SD 卡的内容，SD 卡的标识符、规格及文件格式信息将会显示在控制窗口中。

图 **3-10** 读取 **SD** 卡标识符与规格

## 3.8 RS-232 通信

控制面板允许用户验证 DE2-115 上的 RS-232 串行通信接口是否运行正常。从 PC 连接一根 RS-232 9-pin 公转母的电缆到 RS-232 端口即完成程序安装。控制面板可通过这根电缆与 PC 上的终端仿真软件进行双向通信。另外，如果你不想用 PC 去完成验证测试，也可使用一根 RS-232 loopback 电缆。控制面板上的接收终端窗口将会监测串行通信的状态。请遵照以下步骤启用 RS-232 通信功能：

1. 选择 RS-232 标签，如图 **3-11** 所示窗口即会出现。
2. 从 PC 连接一根 RS-232 9-pin 公转母的电缆，或者直接插入一根 RS - 232 loopback 电缆到 RS - 232 端口。
3. RS-232 设置如下所示（假设连接到了 PC:

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

4. 输入具体的字母内容并点击发送以启动通信操作。在通信过程中，观察终端接收窗口的状态以确认其运行的正常性。

图 **3-11 RS-232** 串行通信

## 3.9 VGA

DE2-115 控制面板提供 VGA pattern 功能，允许用户使用 DE2-115 开发板输出 color pattern 到 LCD/CRT 显示器上 。请遵照如下步骤实现 VGA pattern 功能：

1. 单击 VGA 标签，出现如图 **3-12** 所示窗口。
2. 插入 D-sub 电缆以连接 DE2-115 开发板的 VGA 接口与 LCD/CRT 显示器。
3. LCD/CRT 显示器上将会显示与控制面板窗口上相同的 color pattern。
4. 点击如图 **3-12** 所示的下拉菜单，您可以分别输出不同的已选颜色。

图 3-12 控制 VGA 显示

## 3.10 HSMC

单击 HSMC 标签，图 3-13 所示窗口就会出现。此功能的目的是验证位于 HSMC 接口信号的功能。在运行 HSMC loopback 验证测试之前，遵照 Loopback 安装章节的指示进行安装并点击 Verify。需要注意的是，在安装 HSMC loopback 适配器之前，请关闭 DE2-115 开发板电源，以避免给开发板造成任何损害。

HSMC loopback 适配器并未随开发套件包附赠，您可以经由如下网站购买：(http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=78&No=495)

图 **3-13** 控制面板下的 **HSMC loopback** 验证测试

## 3.11 红外接收器

我们可以通过遥控器发送扫描码，从而在控制面板上测试 DE2-115 开发板上的红外接收器功能。一旦按下 IR 标签，如图 3-14 所示的窗口就会出现。当接收到扫描码的时候,相应的信息会以 16 进制的形式显示在红外接收器的窗口上。同时，遥控器上按下的按钮也会在 IR 接收器窗口的遥控器图形上标示出来。注意：不同品牌的遥控器可能会有不同的编码形式。只有随该套件附赠的遥控器才能与该软件兼容！

图 3-14 用遥控器测试红外接收器

## 3.12 DE2-115 控制面板的整体结构

DE2-115 控制面板是能够在 Cyclone IV E FPGA 芯片上实例化的一个基于 Nios II 的 SOPC 系统，并有相应的软件运行于片上存储器上。软件部分是用 C 语言实现的；硬件部分是用 Verilog HDL 代码与 SOPC builder 共同实现的。在 DE2_115 的系统 CD 上并未提供源代码。

用户须依照本章 3.1 节进行配置，方可运行控制面板程序。图 3-15 描述了控制面板的结构。每一个输入/输出设备均由 FPGA 芯片内实例化的 Nios II 处理器所控制。与电脑之间的数据通信经由 USB Blaster 链接实现。Nios II 将会处理来自 PC 的具体需求并执行相应的操作。

图 **3-15 DE2-115** 控制面板的框图

# 第4章.

# 使用 DE2-115

这一章主要介绍如何使用 DE2-115 开发板以及各种外围设备的功能、和 FPGA 的连接信息等。

## 4.1 配置 Cyclone IV E FPGA 芯片

在 Quartus II 软件使用简介中，我们已经介绍了如何从 PC 下载一个硬件电路到 DE2-115 开发板。这份使用说明书可以在 DE2-115 系统光盘里面的 DE2_115_tutorials 目录下面找到。用户应该首先仔细阅读这份说明书，而把以下的介绍作为简短的参考。

DE2-115 开发板包含一个存储有 Cyclone IV E FPGA 芯片配置数据的串行配置芯片。每次开发板上电的时候，芯片里面的配置数据会自动从配置芯片加载到 FPGA 芯片。使用 Quartus II 软件，用户可以随时重新配置 FPGA，并可以改变存储在非易失性串行存储器芯片（EPCS）里面的数据。下面分述两种不同的配置方式：

1. JTAG 编程：这种下载方式-名字起源于 IEEE 标准，联合测试行动组-会把配置数据直接加载到 Cyclone IV E FPGA 芯片。FPGA 芯片会保持这些配置信息直到芯片掉电。
2. AS 编程：这种下载方式被叫称作串行主动编程，它会下载配置数据到 Altera EPCS64 芯片。它将配置数据保存在非易失性器件中，即使 DE2-115 开发板掉电，数据也不会丢失。在每次开发板上电的时候，EPCS64 芯片里面的数据会自动加载到 Cyclone IV E FPGA 芯片。

### ■ DE2-115 开发板上的 JTAG 链

当使用 JTAG 接口配置 FPGA 芯片的时候，DE2-115 上的 JTAG 链必须形成一个回路，这样 Quartus II 软件才可以正确检测到 JTAG 链上的 FPGA/CPLD 器件。**错误！未找到引用源。**给出了 DE2-115 开发板上的 JTAG 链。短接 JP3 上的第一、二引脚会旁路 HSMC 接头的 JTAG 信号，直接在 DE2-115 上形成 JTAG 回路（参考**错误！未找到引用源。**），这样只有 DE2-115 上的 FPGA 器件（Cyclone IV E）才可以被 Quartus II 软件检测到。如果用户想通过 HSMC 接头接入其它的 FPGA 器件或者包含 FPGA 器件的界面，短路 JP3 的第二、三脚，从而使能 HSMC 接头上的 JTAG 链。

图 4-1 DE2-115 开发板 JTAG 链

图 4-2 JTAG 链配置跳线

以下是关于 JTAG 和 AS 编程的操作步骤。对于这两种方式，DE2-115 开发板均通过 USB 电缆连接到 PC 主机。通过这种连接，主机将开发板当做一个 Altera USB-Blaster 设备。在主机上安装 USB-Blaster 驱动的步骤在说明书“*Getting Started with Altera's DE2-115 Board*”(tut_initialDE2-115.pdf)中有详细描述。这份说明书可以在 DE2-115 的系统光盘中找到。

## ■ 使用 JTAG 模式配置 FPGA

**错误！未找到引用源。**给出了 JTAG 编程模式的设定信息。执行以下步骤，以将配置数据下载到 Cyclone IV E FPGA：

- 确保 DE2-115 已经正确连接好电源
- 将 RUN/PROG 拨动开关（SW19）放置在 RUN 位置（参考**错误！未找到引用源。**）
- 将附带的 USB 电缆连接到 DE2-115 开发板的 USB-Blaster 电路（参考**错误！未找到引用**

源。）

- 现在可以通过 Quartus II 编程器选择合适的以.sof 为扩展名的配置数据来配置 DE2-115 的 FPGA 芯片了

图 4-3 JTAG 配置电路

图 4-4 设置 RUN/PROG 开关（SW19）在 JTAG 模式

## ■ 使用 AS 模式配置 EPCS64 芯片

图 4-5 给出了 AS 配置模式的设定。执行以下步骤，以将配置数据下载到 EPCS64 芯片：

- 确保 DE2-115 已经连接好电源
- 连接附带的 USB 电缆到 DE2-115 的 USB-Blaster 接口 （参考 **错误！未找到引用源。**）
- 将 RUN/PROG 拨动开关（SW19）放置在 PROG 位置

- 现在可以通过 Quartus II 编程器选择以.pof 位扩展名的配置文件来编程 EPCS64 器件了
- 编程结束后，将 RUN/PROG 开关拨回 RUN 位置，关闭 DE2-115 电源，然后再开启。通过这次重启，FPGA 将从 EPCS64 器件读取新的配置数据

图 4-5　AS 配置模式

## 4.2 使用按钮开关和拨动开关

DE2-115 提供了四个按钮开关，如**错误！未找到引用源。**所示。每个按钮开关都通过一个施密特触发器进行了去抖动处理，如**错误！未找到引用源。**所示。四个施密特触发器的输出信号，分别为 KEY0、KEY1、KEY2、KEY3，直接连接到了 Cyclone IV E FPGA。当按钮没有被按下的时候，它的输出是高电平，按下去则给出一个低电平。得益于去抖动电路，这些按钮开关适合用来给内部电路提供（模拟的）时钟或者复位信号。

图 4-6 按钮开关和 FPGA 的连接示意图

图 4-7 开关去抖动

DE2-115 开发板上还有 18 个拨动开关（参考**错误！未找到引用源。**）。这些开关没有去抖动电路，它们可以作为对电平敏感的电路的输入数据。每个开关都直接连接到 Cyclone IV E FPGA。当拨动开关在 DOWN 位置（靠近开发板边缘）的时候输出为低电平，当在 UP 位置时输出为高电平。

图 4-8 拨动开关和 Cyclone IV E FPGA 间的连接示意图

## 4.3 使用 LED

DE2-115 开发板共有 27 个直接由 FPGA 控制的 LED。18 个红色的 LED 位于 18 个拨动开关的正上方，8 个绿色 LED 可以在按钮开关的上方找到（第九个 LED 位于七段数码管的中间）。每一个 LED 都由 Cyclone IV E FPGA 的一个引脚直接驱动，其输出高电平则点亮 LED，输出低电平 LED 熄灭。图 4-9 给出了 LED 和 Cyclone IV E FPGA 之间的连接示意图。

图 4-9 Cyclone IV E FPGA 和 LED 间连接示意图

Cyclone IV E FPGA 到拨动开关间的详细引脚连接信息请参考表 4-1。同时，表 4-2 和表 4-3 分别给出了按钮开关和 LED 到 FPGA 间的各个引脚连接信息。

表 4-1 拨动开关引脚配置

| *信号名* | *FPGA 引脚号* | *说明* | *I/O 标准* |
|---|---|---|---|
| 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 |

表 4-2 按钮开关引脚配置

| *信号名* | *FPGA 引脚号* | *说明* | *I/O 标准* |
|---|---|---|---|
| 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 |

表 4-3 LED 引脚配置

| 信号名 | FPGA 引脚号 | 说明 | I/O 标准 |
|---|---|---|---|
| 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 使用七段数码管

DE2-115 配有八个七段数码管。它们被分成两组，每组四个，用来作为数字显示用。正如图 图 4-10 所示，七段数码管的每个引脚（共阳模式）均连接到 Cyclone IV E FPGA。FPGA 输出低电压的时候，对应的字码段点亮，反之则熄灭。

每个数码管的字段都从 0 到 6 依次编号，图 4-10 给出了它们的编号次序。表 4-4 给出了所有数码管和 FPGA 芯片的引脚连接信息。

图 4-10 Cyclone IV E FPGA 和七段数码管间连接示意图

表 4-4 七段数码管引脚配置

| *信号名* | *FPGA 引脚号* | *说明* | *I/O 标准* |
|---|---|---|---|
| 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] | 取决于 JP7 |
| HEX0[4] | PIN_L25 | Seven Segment Digit 0[4] | 取决于 JP7 |
| HEX0[5] | PIN_J22 | Seven Segment Digit 0[5] | 取决于 JP7 |
| HEX0[6] | PIN_H22 | Seven Segment Digit 0[6] | 取决于 JP7 |
| HEX1[0] | PIN_M24 | Seven Segment Digit 1[0] | 取决于 JP7 |
| HEX1[1] | PIN_Y22 | Seven Segment Digit 1[1] | 取决于 JP7 |
| HEX1[2] | PIN_W21 | Seven Segment Digit 1[2] | 取决于 JP7 |
| HEX1[3] | PIN_W22 | Seven Segment Digit 1[3] | 取决于 JP7 |
| HEX1[4] | PIN_W25 | Seven Segment Digit 1[4] | 取决于 JP7 |
| HEX1[5] | PIN_U23 | Seven Segment Digit 1[5] | 取决于 JP7 |
| HEX1[6] | PIN_U24 | Seven Segment Digit 1[6] | 取决于 JP7 |
| HEX2[0] | PIN_AA25 | Seven Segment Digit 2[0] | 取决于 JP7 |
| HEX2[1] | PIN_AA26 | Seven Segment Digit 2[1] | 取决于 JP7 |
| HEX2[2] | PIN_Y25 | Seven Segment Digit 2[2] | 取决于 JP7 |
| HEX2[3] | PIN_W26 | Seven Segment Digit 2[3] | 取决于 JP7 |
| HEX2[4] | PIN_Y26 | Seven Segment Digit 2[4] | 取决于 JP7 |
| HEX2[5] | PIN_W27 | Seven Segment Digit 2[5] | 取决于 JP7 |
| HEX2[6] | PIN_W28 | Seven Segment Digit 2[6] | 取决于 JP7 |
| HEX3[0] | PIN_V21 | Seven Segment Digit 3[0] | 取决于 JP7 |
| HEX3[1] | PIN_U21 | Seven Segment Digit 3[1] | 取决于 JP7 |
| HEX3[2] | PIN_AB20 | Seven Segment Digit 3[2] | 取决于 JP6 |
| HEX3[3] | PIN_AA21 | Seven Segment Digit 3[3] | 取决于 JP6 |
| HEX3[4] | PIN_AD24 | Seven Segment Digit 3[4] | 取决于 JP6 |
| HEX3[5] | PIN_AF23 | Seven Segment Digit 3[5] | 取决于 JP6 |
| HEX3[6] | PIN_Y19 | Seven Segment Digit 3[6] | 取决于 JP6 |
| HEX4[0] | PIN_AB19 | Seven Segment Digit 4[0] | 取决于 JP6 |
| HEX4[1] | PIN_AA19 | Seven Segment Digit 4[1] | 取决于 JP6 |
| HEX4[2] | PIN_AG21 | Seven Segment Digit 4[2] | 取决于 JP6 |

| | | | |
|---|---|---|---|
| HEX4[3] | PIN_AH21 | Seven Segment Digit 4[3] | 取决于 JP6 |
| HEX4[4] | PIN_AE19 | Seven Segment Digit 4[4] | 取决于 JP6 |
| HEX4[5] | PIN_AF19 | Seven Segment Digit 4[5] | 取决于 JP6 |
| HEX4[6] | PIN_AE18 | Seven Segment Digit 4[6] | 取决于 JP6 |
| HEX5[0] | PIN_AD18 | Seven Segment Digit 5[0] | 取决于 JP6 |
| HEX5[1] | PIN_AC18 | Seven Segment Digit 5[1] | 取决于 JP6 |
| HEX5[2] | PIN_AB18 | Seven Segment Digit 5[2] | 取决于 JP6 |
| HEX5[3] | PIN_AH19 | Seven Segment Digit 5[3] | 取决于 JP6 |
| HEX5[4] | PIN_AG19 | Seven Segment Digit 5[4] | 取决于 JP6 |
| HEX5[5] | PIN_AF18 | Seven Segment Digit 5[5] | 取决于 JP6 |
| HEX5[6] | PIN_AH18 | Seven Segment Digit 5[6] | 取决于 JP6 |
| HEX6[0] | PIN_AA17 | Seven Segment Digit 6[0] | 取决于 JP6 |
| HEX6[1] | PIN_AB16 | Seven Segment Digit 6[1] | 取决于 JP6 |
| HEX6[2] | PIN_AA16 | Seven Segment Digit 6[2] | 取决于 JP6 |
| HEX6[3] | PIN_AB17 | Seven Segment Digit 6[3] | 取决于 JP6 |
| HEX6[4] | PIN_AB15 | Seven Segment Digit 6[4] | 取决于 JP6 |
| HEX6[5] | PIN_AA15 | Seven Segment Digit 6[5] | 取决于 JP6 |
| HEX6[6] | PIN_AC17 | Seven Segment Digit 6[6] | 取决于 JP6 |
| HEX7[0] | PIN_AD17 | Seven Segment Digit 7[0] | 取决于 JP6 |
| HEX7[1] | PIN_AE17 | Seven Segment Digit 7[1] | 取决于 JP6 |
| HEX7[2] | PIN_AG17 | Seven Segment Digit 7[2] | 取决于 JP6 |
| HEX7[3] | PIN_AH17 | Seven Segment Digit 7[3] | 取决于 JP6 |
| HEX7[4] | PIN_AF17 | Seven Segment Digit 7[4] | 取决于 JP6 |
| HEX7[5] | PIN_AG18 | Seven Segment Digit 7[5] | 取决于 JP6 |
| HEX7[6] | PIN_AA14 | Seven Segment Digit 7[6] | 3.3V |

## 4.5 时钟电路

DE2-115 包含一个生成 50MHz 频率时钟信号的有源晶体振荡器，另有一个时钟缓冲器用来将缓冲后的低抖动 50MHz 时钟信号分配给 FPGA。这些时钟信号用来驱动 FPGA 内的用户逻辑电路。开发板还包含两个 SMA 连接头，用来接收外部时钟输入信号到 FPGA 或者将 FPGA 的时钟信号输出到外部。另外，所有这些时钟输入都连接到 FPGA 内部的 PLL 模块上，用户可以将这些时钟信号作为 PLL 电路的时钟输入。

图 4-11 给出了 DE2-115 板子上的时钟分配信息。和 FPGA 芯片相关的引脚配置信息可以在表 4-5 中找到。

图 **4-11** 时钟分配电路方块图

表 **4-5** 时钟信号引脚配置信息

| *信号名* | *FPGA 引脚号* | *说明* | *I/O 标准* |
|---|---|---|---|
| **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** | **取决于 JP6** |
| **SMA_CLKOUT** | **PIN_AE23** | **External (SMA) clock output** | **取决于 JP6** |
| **SMA_CLKIN** | **PIN_AH14** | **External (SMA) clock input** | **3.3V** |

## 4.6 使用 LCD 模块

DE2-115 开发板上的 LCD 模块配有内置英文字库，发送合适的命令控制字到显示控制器 HD44780 便可以在 LCD 显示文字信息。如何使用 LCD 模块的详细资料可以在 DE2-115 系统光盘下面的 *DE2_115_datasheet\LCD* 文件夹下找到。Cyclone IV E FPGA 和 LCD 模块间的连接信息原理框图如图 **4-12** 所示。相关的引脚连接信息可以在表 **4-6** 中找到。

图 4-12 Cyclone IV E FPGA 芯片和 LCD 模块间连接示意图

*请注意，在 DE2-115 中使用的 LCD 模块并不含背光单元，故而 LCD_BLON 信号在用户工程中的设定是无效的。*

表 4-6 LCD 模块引脚配置

| *信号名* | *FPGA 引脚号* | *说明* | *I/O 标准* |
|---|---|---|---|
| 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 | 启用 LCD | 3.3V |
| LCD_RW | PIN_M1 | LCD 读/写选择, 0 = 写, 1 =读 | 3.3V |
| LCD_RS | PIN_M2 | LCD 命令/数据选择, 0 = 命令 d, 1 =数据 | 3.3V |
| LCD_ON | PIN_L5 | LCD 电源开/关 | 3.3V |
| LCD_BLON | PIN_L6 | LCD 背光开/关 | 3.3V |

## 4.7 HSMC 接口

DE2-115 提供了一个 HSMC 接口，通过连接 HSMC 子卡来扩展 FPGA 母板的外围设备。HSMC 接口可以容纳当前主流的各种高速信号规范和传统的低速信号。HSMC 接口支持 JTAG，时钟输出/输入，高速 LVDS 信号以及单端信号。HSMC 接口共有 82 个引脚直接连接到 Cyclone IV E FPGA 芯片，而 HSMC_SDA 以及 HSMC_SCLK 则与开发板上的 WM8731 音频芯片以及 ADV7180 视频芯片共用 I2C_SCL 、I2C_SDA 两根 I2C 总线。**错误！未找到引用源。**给出了 HSMC 接口上各路电源所能给子板提供的最大电流数值。

**表 4-7 HSMC 接口上的电源供给**

| *电源供给* | *最大电流* |
|---|---|
| **12V** | **1A** |
| **3.3V** | **1.5A** |

*(1).请注意，表 4-7 所给出之 HSMC 最大电流，是基于 FPGA 资源消耗不超过 50%之基础之上的。如果在使用 HSMC 子板之时，DE2-115 上 FPGA 芯片消耗资源超过 50%，请联系(support@terasic.com)获取更进一步的技术指导。*

*(2).如果在 HSMC 接口上装配有 HSMC 回环测试适配器，则因回环之缘故，HSMC 上的 I2C_SCL 信号将会连接到 I2C_SDA。由于 TV 解码芯片和音频芯片亦共用此 I2C 总线，从而它们也无法得到正确的配置而正常工作。*

通过设定 JP7 上跳线的位置，用户可以自行设定 HSMC 接口上的信号的 I/O 电压标准为 3.3V、2.5V、1.8V 或者 1.5V（默认设定为 2.5V，参考图 4-13）。因为 HSMC 接口的信号连接到 FPGA 芯片的 Bank 5 以及 Bank 6，而这两个 Bank 的 I/O 电压由 JP7 控制，所以用户可以通过为 VCCIO5 以及 VCCIO6 选择不同的输入电压的方法来设定这些 I/O 引脚的电压标准。表 4-8 罗列出 JP7 的各种设定下的 I/O 电压。

图 **4-13** **HSMC VCCIO** 电源电压设置跳线

表 **4-8** 各种跳线设定下的 **I/O** 电压

| *JP7 跳线设置* | *VCCIO5 与 VCCIO6 的电源供给* | *HSMC 接口的 I/O 电压(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** （默认） |
| **Short Pins 7 and 8** | **3.3V** | **3.3V** |

***请注意，用户在使用不同类型的 HSMC 子板时，要特别注意子板的 I/O 电压标准要和 DE2-115 母版的设定相匹配。如果失配，譬如使用 3.3V 电压 I/O 标准的子板连接到 HSMC 电压设定为 1.8V 的 DE2-115，子板很可能无法正常工作。***

另外，如果要在 HSMC 接口上使用 LVDS 信号标准，LVDS 接收器对需要装配 100 欧姆的终端电阻，如图 **4-14** 所示。表 **4-9** 给出了 HSMC 接口的引脚配置信息。

图 4-14 Cyclone IV FPGA 与 HSMC 接口之间的 LVDS 信号之连接

表 4-9 HSMC 接口引脚配置

| *信号名* | *FPGA 引脚号* | *说明* | *I/O 标准* |
|---|---|---|---|
| HSMC_CLKIN0 | PIN_AH15 | Dedicated clock input | 取决于 JP6 |
| HSMC_CLKIN_N1 | PIN_J28 | LVDS RX or CMOS I/O or differential clock input | 取决于 JP7 |
| HSMC_CLKIN_N2 | PIN_Y28 | LVDS RX or CMOS I/O or differential clock input | 取决于 JP7 |
| HSMC_CLKIN_P1 | PIN_J27 | LVDS RX or CMOS I/O or differential clock input | 取决于 JP7 |
| HSMC_CLKIN_P2 | PIN_Y27 | LVDS RX or CMOS I/O or differential clock input | 取决于 JP7 |
| HSMC_CLKOUT0 | PIN_AD28 | Dedicated clock output | 取决于 JP7 |
| HSMC_CLKOUT_N1 | PIN_G24 | LVDS TX or CMOS I/O or differential clock input/output | 取决于 JP7 |
| HSMC_CLKOUT_N2 | PIN_V24 | LVDS TX or CMOS I/O or differential clock input/output | 取决于 JP7 |
| HSMC_CLKOUT_P1 | PIN_G23 | LVDS TX or CMOS I/O or differential clock input/output | 取决于 JP7 |
| HSMC_CLKOUT_P2 | PIN_V23 | LVDS TX or CMOS I/O or differential clock input/output | 取决于 JP7 |
| HSMC_D[0] | PIN_AE26 | LVDS TX or CMOS I/O | 取决于 JP7 |
| HSMC_D[1] | PIN_AE28 | LVDS RX or CMOS I/O | 取决于 JP7 |
| HSMC_D[2] | PIN_AE27 | LVDS TX or CMOS I/O | 取决于 JP7 |
| HSMC_D[3] | PIN_AF27 | LVDS RX or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[0] | PIN_F25 | LVDS RX bit 0n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[1] | PIN_C27 | LVDS RX bit 1n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[2] | PIN_E26 | LVDS RX bit 2n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[3] | PIN_G26 | LVDS RX bit 3n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[4] | PIN_H26 | LVDS RX bit 4n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[5] | PIN_K26 | LVDS RX bit 5n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[6] | PIN_L24 | LVDS RX bit 6n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[7] | PIN_M26 | LVDS RX bit 7n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[8] | PIN_R26 | LVDS RX bit 8n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[9] | PIN_T26 | LVDS RX bit 9n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[10] | PIN_U26 | LVDS RX bit 10n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[11] | PIN_L22 | LVDS RX bit 11n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[12] | PIN_N26 | LVDS RX bit 12n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[13] | PIN_P26 | LVDS RX bit 13n or CMOS I/O | 取决于 JP7 |

| | | | |
|---|---|---|---|
| HSMC_RX_D_N[14] | PIN_R21 | LVDS RX bit 14n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[15] | PIN_R23 | LVDS RX bit 15n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_N[16] | PIN_T22 | LVDS RX bit 16n or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[0] | PIN_F24 | LVDS RX bit 0 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[1] | PIN_D26 | LVDS RX bit 1 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[2] | PIN_F26 | LVDS RX bit 2 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[3] | PIN_G25 | LVDS RX bit 3 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[4] | PIN_H25 | LVDS RX bit 4 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[5] | PIN_K25 | LVDS RX bit 5 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[6] | PIN_L23 | LVDS RX bit 6 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[7] | PIN_M25 | LVDS RX bit 7 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[8] | PIN_R25 | LVDS RX bit 8 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[9] | PIN_T25 | LVDS RX bit 9 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[10] | PIN_U25 | LVDS RX bit 10 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[11] | PIN_L21 | LVDS RX bit 11 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[12] | PIN_N25 | LVDS RX bit 12 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[13] | PIN_P25 | LVDS RX bit 13 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[14] | PIN_P21 | LVDS RX bit 14 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[15] | PIN_R22 | LVDS RX bit 15 or CMOS I/O | 取决于 JP7 |
| HSMC_RX_D_P[16] | PIN_T21 | LVDS RX bit 16 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[0] | PIN_D28 | LVDS TX bit 0n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[1] | PIN_E28 | LVDS TX bit 1n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[2] | PIN_F28 | LVDS TX bit 2n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[3] | PIN_G28 | LVDS TX bit 3n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[4] | PIN_K28 | LVDS TX bit 4n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[5] | PIN_M28 | LVDS TX bit 5n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[6] | PIN_K22 | LVDS TX bit 6n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[7] | PIN_H24 | LVDS TX bit 7n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[8] | PIN_J24 | LVDS TX bit 8n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[9] | PIN_P28 | LVDS TX bit 9n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[10] | PIN_J26 | LVDS TX bit 10n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[11] | PIN_L28 | LVDS TX bit 11n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[12] | PIN_V26 | LVDS TX bit 12n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[13] | PIN_R28 | LVDS TX bit 13n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[14] | PIN_U28 | LVDS TX bit 14n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[15] | PIN_V28 | LVDS TX bit 15n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_N[16] | PIN_V22 | LVDS TX bit 16n or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[0] | PIN_D27 | LVDS TX bit 0 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[1] | PIN_E27 | LVDS TX bit 1 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[2] | PIN_F27 | LVDS TX bit 2 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[3] | PIN_G27 | LVDS TX bit 3 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[4] | PIN_K27 | LVDS TX bit 4 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[5] | PIN_M27 | LVDS TX bit 5 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[6] | PIN_K21 | LVDS TX bit 6 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[7] | PIN_H23 | LVDS TX bit 7 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[8] | PIN_J23 | LVDS TX bit 8 or CMOS I/O | 取决于 JP7 |

| | | | |
|---|---|---|---|
| HSMC_TX_D_P[9] | PIN_P27 | LVDS TX bit 9 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[10] | PIN_J25 | LVDS TX bit 10 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[11] | PIN_L27 | LVDS TX bit 11 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[12] | PIN_V25 | LVDS TX bit 12 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[13] | PIN_R27 | LVDS TX bit 13 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[14] | PIN_U27 | LVDS TX bit 14 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[15] | PIN_V27 | LVDS TX bit 15 or CMOS I/O | 取决于 JP7 |
| HSMC_TX_D_P[16] | PIN_U22 | LVDS TX bit 16 or CMOS I/O | 取决于 JP7 |

## 4.8 使用通用扩展接头

DE2-115 提供了一个 40 引脚的扩展头(GPIO),它有 36 个引脚直接连接到 Cyclone IV E FPGA 芯片，并提供 5V 和 3.3V 电压引脚和两个接地引脚。图 4-15 给出了 GPIO 上的引脚定义。表 4-10 列出了 GPIO 可以给子板提供的各路电压的最大电流值。

图 4-15 GPIO 引脚定义

表 4-10 通用扩展接口最大电源供给

| *电源供给* | *最大电流* |
|---|---|
| 5V | 1A |
| 3.3V | 1.5A |

通用扩展接口上的每根信号线都提供了额外的两个钳位二极管和一个电阻，用来保护 FPGA 不会因过高或者过低的外部输入电压损坏。图 4-16 给出了通用扩展口上某个引脚的保护电路，但这个电路在每个信号引脚上都存在。

图 4-16　Cyclone IV E FPGA 与 GPIO 间连接示意图

使用 JP6，通用扩展口上的 I/O 电压可以在 3.3V、2.5V、1.8V 或者 1.5V 间选择（默认电压为 3.3V，参考图 4-17）。因为 GPIO 的信号连接到 FPGA 的 Bank 4，而这个 Bank 的 I/O 电压由 JP6 控制，用户可以为 VCCIO4 选择不同的输入电压来达到控制这个 Bank 上的电压标准的目的。表 4-11 给出了 JP6 的具体设定信息。JP6 在开发板上的位置信息请参考图 4-17。

图 4-17　GPIO 电压标准设定跳线

表 4-11　使用 JP6 为 GPIO 设定不同的电压标准

| *JP6 跳线设置* | *VCCIO4 的电源供给* | *扩展接口的 I/O 电压 (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)** |

***请注意，用户在使用不同类型的 GPIO 子板时，要特别注意子板的 I/O 电压标准要和 DE2-115 母版的设定相匹配。如果失配，譬如使用 3.3V 电压 I/O 标准的子板连接到 GPIO 电压设定为 1.8V 的 DE2-115，子板很可能无法正常工作。***

图 4-18 描述了当将 GPIO 上的信号用做 LVDS 发送器的时候，GPIO 的引脚功能定义。GPIO 所连接的 FPGA 引脚属于列 I/O 系列，它们在作为 LVDS 使用时，仅支持模拟模式的 LVDS 发送器，在使用中需要额外的外接电阻网络，如图 4-19 所示。在 Quartus II 软件中，相关的差分对 I/O 标准必须设定为 LVDS_E_3R。

图 4-18　GPIO 用做 LVDS 发送器使用时的 I/O 功能定义

图 4-19 中，电阻 Rs 的出厂默认值为 47 欧姆，而 Rp 默认没有安装。当这些 I/O 用做 LVDS 发送器的时候，请在 Rs 的位置上安装 170 欧姆的电阻，在 Rp 的位置上安装 120 欧姆的电阻。

最后，表 4-12 给出了 GPIO 和 FPGA 间的引脚连接信息。

图 4-19 在 GPIO 接口上使用模拟 LVDS 信号

表 4-12 GPIO 引脚配置信息

| *信号名* | *FPGA 引脚号* | *说明* | *I/O 标准* |
|---|---|---|---|
| GPIO[0] | PIN_AB22 | GPIO Connection DATA[0] | 取决于 JP6 |
| GPIO[1] | PIN_AC15 | GPIO Connection DATA[1] | 取决于 JP6 |
| GPIO[2] | PIN_AB21 | GPIO Connection DATA[2] | 取决于 JP6 |
| GPIO[3] | PIN_Y17 | GPIO Connection DATA[3] | 取决于 JP6 |
| GPIO[4] | PIN_AC21 | GPIO Connection DATA[4] | 取决于 JP6 |
| GPIO[5] | PIN_Y16 | GPIO Connection DATA[5] | 取决于 JP6 |
| GPIO[6] | PIN_AD21 | GPIO Connection DATA[6] | 取决于 JP6 |
| GPIO[7] | PIN_AE16 | GPIO Connection DATA[7] | 取决于 JP6 |
| GPIO[8] | PIN_AD15 | GPIO Connection DATA[8] | 取决于 JP6 |
| GPIO[9] | PIN_AE15 | GPIO Connection DATA[9] | 取决于 JP6 |
| GPIO[10] | PIN_AC19 | GPIO Connection DATA[10] | 取决于 JP6 |
| GPIO[11] | PIN_AF16 | GPIO Connection DATA[11] | 取决于 JP6 |
| GPIO[12] | PIN_AD19 | GPIO Connection DATA[12] | 取决于 JP6 |
| GPIO[13] | PIN_AF15 | GPIO Connection DATA[13] | 取决于 JP6 |
| GPIO[14] | PIN_AF24 | GPIO Connection DATA[14] | 取决于 JP6 |
| GPIO[15] | PIN_AE21 | GPIO Connection DATA[15] | 取决于 JP6 |
| GPIO[16] | PIN_AF25 | GPIO Connection DATA[16] | 取决于 JP6 |
| GPIO[17] | PIN_AC22 | GPIO Connection DATA[17] | 取决于 JP6 |
| GPIO[18] | PIN_AE22 | GPIO Connection DATA[18] | 取决于 JP6 |
| GPIO[19] | PIN_AF21 | GPIO Connection DATA[19] | 取决于 JP6 |
| GPIO[20] | PIN_AF22 | GPIO Connection DATA[20] | 取决于 JP6 |
| GPIO[21] | PIN_AD22 | GPIO Connection DATA[21] | 取决于 JP6 |
| GPIO[22] | PIN_AG25 | GPIO Connection DATA[22] | 取决于 JP6 |
| GPIO[23] | PIN_AD25 | GPIO Connection DATA[23] | 取决于 JP6 |
| GPIO[24] | PIN_AH25 | GPIO Connection DATA[24] | 取决于 JP6 |
| GPIO[25] | PIN_AE25 | GPIO Connection DATA[25] | 取决于 JP6 |
| GPIO[26] | PIN_AG22 | GPIO Connection DATA[26] | 取决于 JP6 |
| GPIO[27] | PIN_AE24 | GPIO Connection DATA[27] | 取决于 JP6 |
| GPIO[28] | PIN_AH22 | GPIO Connection DATA[28] | 取决于 JP6 |

| | | | |
|---|---|---|---|
| GPIO[29] | PIN_AF26 | GPIO Connection DATA[29] | 取决于 JP6 |
| GPIO[30] | PIN_AE20 | GPIO Connection DATA[30] | 取决于 JP6 |
| GPIO[31] | PIN_AG23 | GPIO Connection DATA[31] | 取决于 JP6 |
| GPIO[32] | PIN_AF20 | GPIO Connection DATA[32] | 取决于 JP6 |
| GPIO[33] | PIN_AH26 | GPIO Connection DATA[33] | 取决于 JP6 |
| GPIO[34] | PIN_AH23 | GPIO Connection DATA[34] | 取决于 JP6 |
| GPIO[35] | PIN_AG26 | GPIO Connection DATA[35] | 取决于 JP6 |

## 4.9 使用 14 脚扩展口

DE2-115 开发板提供一个 14 引脚的扩展接头，其中有 7 根信号线直接连接到 Cyclone IV E FPGA 芯片，并提供 3.3V 的电源引脚和 6 根接地引脚，如图 4-20 所示。扩展接口上的 I/O 电压标准为 3.3V。表 4-13 给出了扩展接头的 I/O 引脚连接信息。

图 4-20 FPGA 与 14 引脚扩展接头间连接示意图

表 4-13 扩展接口引脚配置信息

| *信号名* | *FPGA 引脚号* | *说明* | *I/O 标准* |
|---|---|---|---|
| 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 使用 VGA

DE2-115 开发板包含一个用于 VGA 视频输出的 15 引脚 D-SUB 接头。VGA 同步信号直接由 Cyclone IV E FPGA 所驱动，Analog Device 公司的 ADV7123 三通道 10 位（仅高八位连接到 FPGA）高速视频 DAC 芯片用来将输出的数字信号转换为模拟信号（R,G,B）。芯片可支持的分辨率为 SVGA 标准（1280*1024），带宽达 100MHz。图 4-21 给出了相关的电路图。

图 4-21　VGA 与 FPGA 间连接示意图

VGA 信号相关的时序规范可以在很多教育类网站上找到（譬如，搜索‘VGA 信号时序’）。图 4-22 给出了 VGA 显示器所要求的单一扫描行基本时序。水平同步信号给出的指定宽度的低电平有效信号指示了上一扫描行的结束和新扫描行的开始。随之而来的便是行扫描后沿（Back porch），这期间的 RGB 输入是无效的，紧接着是行显示区间，这期间的 RGB 信号将在显示器上逐点显示出来。最后，是持续特定时间的行显示前沿（Front porch），这期间的 RGB 信号也是无效的。场同步信号的时序完全类似，如图 4-22 所示，只不过场同步脉冲指示某一帧的结束和下一帧的开始，消隐期长度的单位不再是像素，而是行数（参考行时序）。表 4-14 和表 4-15 给出了不同分辨率情况下行和场时序中各区间的持续长度。

如何使用 ADV7123 的详细信息可以在它的数据手册上找到，数据手册可以在芯片制造商的网站上下载，或者在 DE2-115 系统光盘里面的 DE2_115_datasheet\VIDEO-DAC 文件夹下面找到。Cyclone IV E FPGA 芯片和 ADV7123 间的引脚连接信息如表 4-16 所示。如何驱动 VGA 显示器的范例可以在第 6.2 节、6.3 节找到。

*请注意，DE2-115 上的 RGB 信号位宽为 8 比特，与 DE2/DE2-70 开发板上的 10 比特宽度有所区别。*

图 4-22 VGA 行扫描时序

表 4-14 VGA 行时序规范

| *VGA 模式* | | *行时序规范* | | | | |
|---|---|---|---|---|---|---|
| *配置* | *分辨率(HxV)* | *a(us)* | *b(us)* | *c(us)* | *d(us)* | *像素时钟(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 |

表 4-15 VGA 场时序规范

| *VGA 模式* | | *场时序规范* | | | | |
|---|---|---|---|---|---|---|
| *配置* | *分辨率(HxV)* | *a(lines)* | *b(lines)* | *c(lines)* | *d(lines)* | *像素时钟(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 |

表 4-16 ADV7123 引脚配置

| 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 使用 24 比特音频编解码芯片

DE2-115 开发板凭借 Wolfon WM8731 音频编解码（CODEC）芯片为用户提供 24 位的高品质音频界面。芯片支持麦克风输入、线路输入以及线路输出端口，采样率在 8kHz 到 96kHz 间可调。用户通过 I2C 总线可以配置 WM8731，这两根信号线直接连接到 FPGA。图 4-23 给出了音频电路相关的原理框图，表 4-17 列出了音频芯片到 FPGA 的引脚配置。如何使用 WM8731 编解码器的详细资料可以参考其数据手册，这可以在制造商的网站下载，或者在 DE2-115 系统光盘里的 DE2_115_datasheets\Audio CODEC 目录下面找到。

图 4-23 FPGA 与音频 CODEC 芯片间连接示意图

表 4-17 音频编解码芯片引脚配置

| *信号名* | *FPGA 引脚号* | *说明* | *I/O 标准* |
|---|---|---|---|
| **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** |

*请注意，如果在 HSMC 接口上装配有 HSMC 回环测试适配器，则因回环之缘故，HSMC 上的 I2C_SCL 信号将会连接到 I2C_SDA。由于 TV 解码芯片和音频芯片亦共用此 I2C 总线，从而它们也无法得到正确的配置而正常工作。*

## 4.12 RS-232 串口

DE2-115 开发板使用 ZT3232 收发器芯片和 9 引脚 DB9 接口用于 RS-232 通信之用。有关如何使用串口收发器芯片的详细资料，可以参考芯片数据手册，它可以在制造商的网站上下载，也可以在 DE2-115 系统光盘里面的 DE_115_datasheets\RS-232 目录下找到。图 4-24 给出了接口相关的原理图，表 4-18 给出了 RS-232 界面和 Cyclone IV E FPGA 间的引脚连接信息。

图 4-24 FPGA 与 ZT3232（RS-232）间连接示意图

表 4-18 RS-232 引脚配置

| *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 接口

DE2-115 包含一个标准的 PS/2 界面，可以用来外接 PS/2 鼠标或键盘。图 4-25 给出了 PS/2 界面相关的原理图。另外，用户可以通过连接 Y 型电缆到 PS/2 的方式，同时连接鼠标和键盘到 DE2-115 开发板（参考图 4-26）。如何使用 PS/2 鼠标或者键盘的相关资料，用户可自行搜索相关的教育类网站。PS/2 界面相关的引脚配置信息可以在表 4-19 中找到。

*请注意，如果用户仅连接一个 PS/2 类设备到 PS/2 接口，使用到的 PS/2 接口信号线应该为“PS2_CLK”以及“PS2_DAT”。*

图 4-25 Connection between FPGA and PS/2

图 4-26 使用 Y 型电缆同时连接 PS/2 鼠标和键盘

表 4-19 PS/2 引脚配置

| *信号名* | *FPGA 引脚号* | *说明* | *I/O 标准* |
|---|---|---|---|
| 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 千兆以太网接口

DE2-115 通过两个 Marvell 88E1111 以太网 PHY 芯片为用户提供网络界面。88E1111 芯片支持 10/100/1000 Mbps 传输速率，支持的 MAC 层传输界面有 GMII/MII/RGMII/TBI 等。表 4-20 描述了两个芯片上电之后的默认设定信息。图 4-27 给出了 FPGA 和以太网芯片间的连线信息。

表 4-20 千兆以太网卡默认设置

| *配置* | *说明* | *默认值* |
|---|---|---|
| PHYADDR[4:0] | MDIO/MDC 模式的 PHY 地址 | 10000 for Enet0;10001 for Enet1 |
| ENA_PAUSE | 启用暂停功能 | 1-默认寄存器 4.11:10 to 11 |
| ANEG[3:0] | 自动配置模式 | 1110-Auto-neg, advertise all capabilities, prefer master |
| ENA_XC | 启用交叉 | 0-禁用 |
| DIS_125 | 禁用 125MHz 时钟 | 1-禁用 125CLK |
| HWCFG[3:0] | 硬件配置模式 | 1011/1111 RGMII to copper/GMII to copper |
| DIS_FC | 禁用光纤/铜线界面 | 1-禁用 |
| DIS_SLEEP | 能量检测 | 1-禁用能量检测 |
| SEL_TWSI | 界面选择 | 0-选择 MDC/MDIO 界面 |
| INT_POL | 中断极性 | 1-INTn 信号低电平有效 |
| 75/50OHM | 终端电阻 | 0-50 ohm termination for fiber |

DE2-115 开发板上的网络芯片仅仅支持 RGMII 以及 MII 两种传输模式（默认模式为 RGMII）。每个芯片都有一个跳线开关用来在 RGMII 以及 MII 模式间切换（参考图 4-28）。在利用跳线

开关设置为新的模式后，可能需要对网络芯片执行一次复位动作，用以使能新的工作模式。表 4-21 以及表 4-22 描述了如何通过跳线分别设定 ENET0 芯片（U8）以及 ENET1 芯片（U9）的工作模式。

另外，使用普通的 5 类非屏蔽双绞线，芯片可以在线动态配置为支持 10Mbps、100Mbps 或者 1000Mbps 的传输速率。相关的引脚配置信息可以在表 4-23 中找到。如需详细的关于如何使用 88E1111 芯片的资料，请参考此芯片的数据手册和应用笔记，这些都可以在制造商的官方网站上找到。

图 4-27　FPGA 与以太网芯片间连接示意图

图 4-28　以太网芯片工作模式切换跳线

表 4-21 ENET0 芯片工作模式切换 (U8)

| *JP1 跳线设置* | *ENET0 PHY 工作模式* |
|---|---|
| 短接引脚 1 和 2 | RGMII 模式 |
| 短接引脚 2 和 3 | MII 模式 |

表 4-22 ENET1 芯片工作模式切换(U9)

| *JP2 跳线设置* | *ENET1 PHY 工作模式* |
|---|---|
| 短接引脚 1 和 2 | RGMII 模式 |
| 短接引脚 2 和 3 | MII 模式 |

表 4-23 千兆以太网芯片引脚配置

| *信号名* | *FPGA 引脚号* | *说明* | *I/O 标准* |
|---|---|---|---|
| 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 解码器

DE2-115 开发板配备有 Analog Device 公司的 ADV7180 视频解码芯片。ADV7180 是一个高度集成的视频解码芯片，可以自动检测输入信号的电视标准（NTSC/PAL/SECAM），并将其数字化为兼容 ITU-R BT.656 的 4:2:2 分量视频数据。ADV7180 兼容各种视频设备，包括 DVD 播放器、磁带机、广播级视频源以及安全、监控类摄像头。

TV 解码器芯片的控制寄存器可以通过芯片的 I2C 总线来访问，而 I2C 总线直接连接到 Cyclone IV E FPGA 芯片，如图 4-29 所示。TV 解码芯片的 I2C 总线读、写地址分别为 0x41/0x40。表 4-24 给出了芯片相关的引脚配置信息。详细的使用 ADV7180 芯片的数据手册可以在制造商的官方网站上下载，或者在 DE2_115 系统光盘里面的 DE2_115_datasheet\TV Decoder 目录下找到。

图 4-29 FPGA 与 TV 解码器连接示意图

*请注意，如果在 HSMC 接口上装配有 HSMC 回环测试适配器，则因回环之缘故，HSMC 上的 I2C_SCL 信号将会连接到 I2C_SDA。由于 TV 解码芯片和音频芯片亦共用此 I2C 总线，从而它们也无法得到正确的配置而正常工作。*

表 4-24　TV 解码芯片引脚配置

| *信号名* | *FPGA 引脚号* | *说明* | *I/O 标准* |
|---|---|---|---|
| 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 TV 编码器实现

虽然 DE2-115 没有专用的 TV 编码器芯片，但是使用 ADV7123 芯片，搭配在 Cyclone IV E FPGA 中实现的数字信号处理电路，可以组成一个专业品质的 TV 编码器。图 4-30 给出了一个以这种方式实现的 TV 编码器。

图 4-30　结合 Cyclone IV E FPGA 和 ADV7123 芯片的 TV 编码器实现

## 4.17 使用 USB 界面

DE2-115 通过飞利浦 ISP1362USB 控制器芯片同时提供 USB 主/从界面。主/从设备控制器完全符合通用串行总线协议 2.0 版规范，支持全速（12Mbit/s）以及低速（1.5Mbit/s）模式。图 4-31 给出了 USB 电路相关的原理框图，引脚配置信息可以在表 4-25 中找到。

如何使用芯片的详细资料可以参考芯片数据手册，用户可以到制造商的网站下载，或者在 DE2-115 系统光盘里面的 DE2_115_datasheet\USB 文件夹下面找到。USB 应用中最具挑战性的部分应该是如何为 USB 设备开发合适的驱动了。两个完整的使用 USB 芯片的范例，可以在 6.4 以及 6.5 节中找到。这些范例展示了如何为 Nios II 处理器开发软件驱动。

图 4-31 FPGA 与 USB 芯片（ISP1362）间的连接示意图

表 4-25 USB (ISP1362)引脚配置

| *Signal Name* | *FPGA 引脚号* | *说明* | *I/O 标准* |
|---|---|---|---|
| OTG_ADDR[0] | PIN_H7 | ISP1362 Address[0] | 3.3V |
| OTG_ADDR[1] | PIN_C3 | ISP1362 Address[1] | 3.3V |
| OTG_DATA[0] | PIN_J6 | ISP1362 Data[0] | 3.3V |
| OTG_DATA[1] | PIN_K4 | ISP1362 Data[1] | 3.3V |
| OTG_DATA[2] | PIN_J5 | ISP1362 Data[2] | 3.3V |
| OTG_DATA[3] | PIN_K3 | ISP1362 Data[3] | 3.3V |
| OTG_DATA[4] | PIN_J4 | ISP1362 Data[4] | 3.3V |
| OTG_DATA[5] | PIN_J3 | ISP1362 Data[5] | 3.3V |
| OTG_DATA[6] | PIN_J7 | ISP1362 Data[6] | 3.3V |
| OTG_DATA[7] | PIN_H6 | ISP1362 Data[7] | 3.3V |
| OTG_DATA[8] | PIN_H3 | ISP1362 Data[8] | 3.3V |
| OTG_DATA[9] | PIN_H4 | ISP1362 Data[9] | 3.3V |
| OTG_DATA[10] | PIN_G1 | ISP1362 Data[10] | 3.3V |
| OTG_DATA[11] | PIN_G2 | ISP1362 Data[11] | 3.3V |

| | | | |
|---|---|---|---|
| OTG_DATA[12] | PIN_G3 | ISP1362 Data[12] | 3.3V |
| OTG_DATA[13] | PIN_F1 | ISP1362 Data[13] | 3.3V |
| OTG_DATA[14] | PIN_F3 | ISP1362 Data[14] | 3.3V |
| OTG_DATA[15] | PIN_G4 | ISP1362 Data[15] | 3.3V |
| OTG_CS_N | PIN_A3 | ISP1362 Chip Select | 3.3V |
| OTG_RD_N | PIN_B3 | ISP1362 Read | 3.3V |
| OTG_WR_N | PIN_A4 | ISP1362 Write | 3.3V |
| OTG_RST_N | PIN_C5 | ISP1362 Reset | 3.3V |
| OTG_INT[0] | PIN_A6 | ISP1362 Interrupt 0 | 3.3V |
| OTG_INT[1] | PIN_D5 | ISP1362 Interrupt 1 | 3.3V |
| OTG_DACK_N[0] | PIN_C4 | ISP1362 DMA Acknowledge 0 | 3.3V |
| OTG_DACK_N[1] | PIN_D4 | ISP1362 DMA Acknowledge 1 | 3.3V |
| OTG_DREQ[0] | PIN_J1 | ISP1362 DMA Request 0 | 3.3V |
| OTG_DREQ[1] | PIN_B4 | ISP1362 DMA Request 1 | 3.3V |
| OTG_FSPEED | PIN_C6 | USB Full Speed, 0 = Enable, Z = Disable | 3.3V |
| OTG_LSPEED | PIN_B6 | USB Low Speed, 0 = Enable, Z = Disable | 3.3V |

## 4.18 使用 IR 模块

DE2-115 开发板配备有一个红外接收（IR）模组（型号：IRM-V538N7/TR1），相关的电路图可以在 DE2-115 系统光盘里面的 DE2_115_datasheet\IR_Receiver 目录下找到。需要注意的是，这个一体化接收模组，仅兼容 38kHz 载波脉宽调制模式。使用附带的 uPD6121G 芯片编码的遥控器可以产生与接收器匹配的调制信号。图 4-32 给出了 IR 相关的电路图，相关引脚配置信息可以在表 4-26 中找到。

图 4-32　FPGA 与 IR 间连接示意图

表 4-26 IR 引脚配置

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

## 4.19 使用 SRAM/SDRAM/Flash/EEPROM/SD 卡

### ■ SRAM

DE2-115 开发板配有一片 2MB 容量，16 比特位宽的 SRAM 芯片。它在 3.3V 的标准 I/O 电压标准下，可以在 125MHz 的频率下运行。鉴于其高速特性，在高速多媒体数据处理应用中，可以把它用做数据缓存等。图 4-33 给出了 SRAM 相关的原理框图。

图 4-33 FPGA 与 SRAM 间连接示意图

### ■ SDRAM

开发板配有 32 比特位宽、128MB SDRAM 内存，由两片 16 比特位宽、64MB 容量的 SDRAM 芯片并联运用而成，两个芯片共用地址和控制信号线。这两个芯片使用 3.3V LVCMOS 信号电平标准。SDRAM 相关的原理框图如图 4-34 所示。

图 **4-34** **FPGA 与 SDRAM 间连接示意图**

## ■ Flash

开发板配有一片 8MB 大小，8 比特数据位宽的 Flash 芯片，它使用 3.3V CMOS 电压标准。由于它的非易失特性，在引用中，可以用于存储软件代码、图像、声音或者其它媒体。图 **4-35** 给出了 Flash 相关的原理框图。

图 **4-35** **Flash 与 FPGA 间连接示意图**

## ■ EEPROM

开发板上还有一片 I2C 接口 32Kb 容量 EEPROM 芯片，由于 I2C 接口的简洁通用性，它一般用来存储如版本信息，IP 地址等描述性信息。图 4-36 给出了和它相关的原理图信息。

图 4-36　FPGA 与 EEPROM 间连接示意图

## ■ SD Card

很多应用需要大容量的外部存储器来存储数据，譬如 SD 卡或者 CF 卡。DE2-115 提供了存取 SD 卡所需的硬件。用户可以自行开发控制器以 SPI 或者 SD 卡 4 比特/1 比特模式来读写 SD 卡。图 4-37 给出了相关的原理框图。

最后，表 4-27~表 4-31 列出了存储界面相关的器件引脚配置信息。

图 4-37　FPGA 与 SD 卡间连接示意图

表 4-27 SRAM 引脚配置

| 信号名 | FPGA 引脚号 | 说明 | I/O 标准 |
|---|---|---|---|
| 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 |

表 4-28 SDRAM 引脚配置

| 信号名 | FPGA 引脚号 | 说明 | I/O 标准 |
|---|---|---|---|
| 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 |

表 4-29 Flash 引脚配置

| *信号名* | *FPGA 引脚号* | *说明* | *I/O 标准* |
|---|---|---|---|
| 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 |

表 4-30 EEPROM 引脚配置

| *信号名* | *FPGA 引脚号.* | *说明* | *I/O 标准* |
|---|---|---|---|
| EEP_I2C_SCLK | PIN_D14 | EEPROM clock | 3.3V |
| EEP_I2C_SDAT | PIN_E14 | EEPROM data | 3.3V |

表 4-31 SD 卡插槽引脚配置

| *信号名* | *FPGA 引脚号* | *说明* | *I/O 标准* |
|---|---|---|---|
| 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 |

# 第5章.

# DE2-115 系统生成器

本章介绍用户如何使用 DE2-115 软件工具－ DE2-115 系统生成器来创建用户设计项目文件。

## 5.1 简介

DE2-115 系统生成器是一款基于 Windows 的软件工具，旨在帮助用户快速创建 Quartus II 项目文件。可生成的 Quartus II 项目文件包括：

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

上述文件的自动创建，可以避免用户自己手动编辑顶层设计文件或引脚分配时发生的错误。用户经常会遇到的问题：

1. Board damaged for wrong pin/bank voltage assignments.（开发板因错误的引脚分配/Bank 电压分配受损）
2. Board malfunction caused by wrong device connections or missing pin counts for connected ends.（板子由于设备连接错误或者连接引脚丢失发生故障）
3. Performance degeneration because of improper pin assignments.（不正确的引脚分配造成性能退化）

## 5.2 一般设计流程

这部分将为您介绍用 DE2-115 系统生成器生成一个项目的一般设计流程。图 5-1 阐述了一般设计流程。

使用者应该先启动 DE2-115 系统生成器，并根据自己的设计要求创建一个新的项目。使用者完成该设置后，DE2-115 系统生成器将生成两个主要文件 ：顶层设计文件 (.v) 和 Quartus II 工程设置文件(.qsf)。

顶层设计文件包括系统的 Verilog HDL 描述文件，供用户添加他们自己的设计/逻辑。Quartus II setting file 包括如下信息： FPGA 配置型号， 顶层引脚的分配，以及用户自定义 I/O 引脚的电平标准等。

最后，必须用 Quartus II 编程器通过 JTAG 接口将.SOF 文件下载到 DE2-115 开发板上。

图 **5-1** 生成一项设计的一般流程

## 5.3 使用 DE2-115 系统生成器

这个部分详细介绍了如何使用 DE2-115 系统生成器。

### ■ 安装和启动 DE2-115 系统生成器

您可以在 DE2-115 系统 CD 的以下位置找到 DE2-115 系统生成器："DE2_115_tools\DE2_115_system_builder" 。用户无需额外安装该工具，只须复制整个文件夹到主机上。点击主机上的文件 DE2_115_SystemBuilder.exe 就可以启动 DE2-115 系统生成器，图形用户界面（GUI） 窗口将如图 5-2 所示。

图 **5-2 DE2-115** 系统生成器窗口

## ■ 输入项目名称

输入项目名称，如图 5-3 所示 。

项目名称： 输入一个正确的名称， 它将作为顶层设计实体的名称。

图 **5-3 DE2-115** 开发板的类型和项目名称

## ■ 系统配置

在系统配置栏，用户可以任意选择 DE2-115 是否在自己的设计中包含开发板上的组件（如图 5-4 所示）。DE2-115 的每个组件都一一列出，用户根据自己的设计，可以通过简单的标记"√"或者移除"√"来启用或者禁止某个组件，如果某个组件启用了，该系统生成器将会自动生成相关的引脚分配，包括：引脚名称、引脚地址、引脚方向和 I/O 标准。

图 5-4 系统配置栏

## ■ GPIO 扩展口

如图 5-5 所示。从下拉菜单里可以选择您希望包含在您设计里面的正确的 GPIO 子卡。系统生成器将自动生成相关的引脚分配，包括：引脚名称，引脚地址，引脚方向和 I/O 标准。如果要使用自定义的子卡，用户可以选择“GPIO Default”根据该子板的规格来改变引脚名称，引脚地址，引脚方向和 I/O 标准。

图 5-5 GPIO 扩展卡组

该“前缀名”是一个可选的功能，是指在您的设计里面赋予该子卡的前缀名。用户可以忽略此栏不填。

## ■ HSMC 扩展口

如图 5-6 所示。从可以与 HSMC 连接器相连的子卡里面选择您希望包含在您设计里面的子卡。系统生成器将自动生成相关的引脚分配，包括：引脚名称，引脚地址，引脚方向和 I/O 标准。

图 5-6 HSMC 扩展卡组

该“前缀名”是一个可选的功能，是指在您的设计里面赋予该子卡的前缀名。用户可以忽略此栏不填。

## ■ 项目功能管理

DE2-115 系统生成器还提供如下功能：恢复默认设置、加载设置和保存用户的板卡配置文件，如图 5-7 所示。用户可以将当前的板卡配置信息保存为.cfg 文件，用其配置 DE2-115 系统生成器。

图 **5-7** 项目设置

## ■ 项目生成

当用户按下“Generate”按钮后， DE2-115 系统生成器将生成相应的 Quartus II 文件和说明文档，如表 5-1 所列。

表 **5-1 DE2-115** 系统生成器生成的文件

| *No.* | *文件名* | *说明* |
|---|---|---|
| **1** | **<Project name>.v** | **Quartus II** 顶层 **verilog HDL** 文件 |
| **2** | **<Project name>.qpf** | **Quartus II** 工程文件 |
| **3** | **<Project name>.qsf** | **Quartus II** 设置文件 |
| **4** | **<Project name>.sdc** | **Quartus II** 概要设计约束文件 |
| **5** | **<Project name>.htm** | 引脚分配文档 |

用户可以使用 Quartus II 软件将定制逻辑添加到项目中，并编译该项目使生成 SRAM 目标文件(.sof)。

# 第6章.

# 高阶设计范例

这一章给出了在 DE2-115 上实现的多个高阶应用范例。这些设计例程展示了开发板多方面的特性，譬如它在音频、视频信号处理方面的能力，在 USB、网络方面的通信能力。给出的每个应用范例，均提供有 Cyclone IV E FPGA（或者是 EPCS64、EEPROM）配置文件，以及完整的 Verilog HDL 源代码。

### ■ 安装设计范例

安装过程：

复制 *DE2_115_demonstrations* 目录及目录下的内容到电脑上指定位置。请注意这个文件夹所在的路径不能包含空格或者中文，否则工程里面的 Nios II 软件工程可能无法再次正确编译。另外，确保用户的电脑上已经安装好了 Quartus II 9.1SP2 或者更高的版本，以正确识别 DE2-115 上的 Cyclone IV E 系列 FPGA 芯片。用户也可以选择安装 Altera 设计工具包 DVD 光盘中的 Quartus II 10.0 版本。

## 6.1 DE2-115 默认配置

DE2-115 出厂时写有默认的配置数据，用于演示开发板的一些基本特征。下面给出了默认配置的存放位置，和执行默认配置需要准备的工作。

### ■ 准备工作、文件位置及操作步骤

- 工程目录：DE2_115_Default
- 使用到的配置文件：DE2_115_Default.sof or DE2_115_Default.pof
- 开启 DE2-115 电源，连接 USB-Blaster 电缆到 DE2-115 的 USB-Blaster 端口，如果有必要（当 EPCS 器件里面没有出厂默认代码时），使用 JTAG 或者 AS 模式下载默认工程编程文件到开发板。

- 现在，用户可以观察到以下现象：七段数码管重复显示数字序列，红色和绿色 LED 灯以约 1Hz 的频率闪烁，LCD 上会显示字符串“Welcome to the DE2-115”。
- 如果额外的把 VGA 输出端口连接到 VGA 兼容型显示器，显示器会展示默认的彩色图片。
- 连接有源音箱或者耳机到开发板线路输出端口
- 将拨动开关 SW17 放置在 DOWN 位置，你可以听到频率为 1kHz 的单音信号（请注意，在默认配置下，线路输出音量很大，请将耳机或者扬声器音量调到较小的值，以免造成任何损伤）。如果额外的连接一个麦克风到开发板的麦克风输入端口或者连接音频设备的输出到线路输入端口，并且把 SW17 切换到 UP 位置，你可以听到来自两种音源的混合信号。

这个默认工程的源代码可以在文件夹 *DE2_115_Default folder* 下面找到。这个工程的顶层 Verilog HDL 文件（DE2_115_Default.v），还可以用作其它工程的设计模板，因为在这个顶层模块里面定义了 DE2-115 开发板中所有用户可访问的与 Cyclone IV E FPGA 芯片连接的外围设备引脚连接信息。

## 6.2 TV 电视盒设计范例

这个设计范例使用 DE2-115 上的 VGA 输出、音频编解码芯片以及 TV 解码芯片（U6）播放来自 DVD 播放器输出的视频和音频信号。图 6-1 给出了设计的系统框图。系统主要由两个模块组成，它们是 I2C_AV_Config 以及 TV_to_VGA 模块。TV_to_VGA 模块由 ITU-R 656 解码器，SDRAM 帧缓冲器，YUV422 转 YUV444，YcrCb 转 RGB 以及 VGA 控制器组成。从图中还可以看出，设计使用了 TV 解码芯片 ADV7180 和 VGA DAC 芯片 ADV7123。

当 FPGA 配置比特流(bit stream)下载完成后，硬件寄存器中的值会通过 I2C 总线去配置 TV 解码芯片，它们之间通过 I2C 协议通信。在上电过程中，TV 解码芯片会在一段时间内不稳定，锁定检测器负责检测这个过程。

ITU-R 解码器模块从 TV 解码器芯片送来的 ITU-R656 数据流里面抽取出 YcrCb4:2:2（YUV4:2:2）。它同时生成一个指示数据输出是否有效的控制信号。因为 TV 解码芯片输出的视频信号是交织排列的，我们必须对输出的数据进行解交织操作。这里，我们使用一个 SDRAM 帧缓冲器和多路选择器（MUX）进行解交织操作。YUV422_to_YUV444 模块完成从 YcrCb4:2:2 到 YcrCb4:4:4 格式的转换。

最后，YcrCb_to_RGB 模块将数据从 YcrCb 格式转成 RGB 格式。VGA 控制器模块生成 VGA 所需的时序信号 VGA_HS 以及 VGA_VS 信号。

图 6-1 TV 电视和设计范例系统框图

## ■ 准备工作、文件位置及操作步骤

- 工程目录： DE2_115_TV
- 使用到的配置数据：DE2_115_TV.sof or DE2_115_TV.pof
- 连接 DVD 的复合信号输出信号（黄色插头）到开发板的视频输入 RCA 接口（J12）。DVD 播放器须配置为输出以下格式的信号：
  - NTSC 电视格式
  - 60Hz 刷新率
  - 4:3 荧幕比
  - 隔行扫描
- 将 DE2-115 开发板的 VGA 输出接到 VGA 显示器（CRT 或者 LCD 均可）
- 将 DVD 播放机的音频输出接到 DE2-115 的线路输入接口，连接一个有源音箱到线路输出端口。如果 DVD 播放器的输出端口为 RCA 型，则需要一个额外的转接线。
- 执行目录 DE2_115_TV\demo_batch\ 下面的 de2_115_tv.bat 批处理命令下载设计到 FPGA
- 按下 DE2-115 开发板上的 KEY0 可以复位硬件电路

*请注意，如果在 HSMC 接口上装配有 HSMC 回环测试适配器，则因回环之缘故，HSMC 上的 I2C_SCL 信号将会连接到 I2C_SDA。由于 TV 解码芯片和音频芯片亦共用此 I2C 总线，从而它们也无法得到正确的配置而正常工作。*

图 6-2 展示了设计范例运行环境的搭建

图 6-2 TV 电视盒设计范例运行环境构建

## 6.3 USB 画笔

USB 接口在多媒体设备里面广泛应用。DE2-115 为 USB 主/从设备提供了完整的解决方案。在这个设计范例中，我们给出了一个以 USB 鼠标为输入设备的 USB ‘画笔’。

这个设计范例使用配有飞利浦 ISP1362 芯片的主设备端口，结合 Nios II 处理器，实现了 USB 鼠标的移动检测。系统还实现了一个视频帧缓冲器，结合 VGA 控制器完成实时图像存储和显示。图 6-3 给出了系统电路框图。VGA 控制器作为一个 Avalon 从设备，可以使用 Nios II 处理器直接控制。

程序运行后，Nios II 处理器开始检测 DE2-115 上的 USB 鼠标是否存在。当鼠标有移动时，Nios II 处理器可以跟踪这些移动并将其记录在帧缓冲存储器。VGA 控制器会在 VGA 缓冲器中存储的默认图像基础上叠加移动坐标数据点，并将其显示在 VGA 显示器上。

图 6-3 USB 画笔设计范例系统框图

## ■ 准备工作、文件位置及操作步骤

工程目录：DE2_115_NIOS_HOST_MOUSE_VGA

使用到的配置数据： DE2_115_NIOS_HOST_MOUSE_VGA.sof

Nios II 工程目录：DE2_115_NIOS_HOST_MOUSE_VGA\Software

- 连接一个 USB 鼠标到 DE2-115 开发板的 USB 主机端口（A 型接口）
- 连接 DE2-115 的 VGA 输出到 VGA 显示器（LCD 或者 CRT）
- 下载 FPGA 配置数据（*）
- 运行 Nios II IDE 并切换工作目录到 DE2_115_NIOS_HOST_MOUSE_VGA\Software
- 现在，你可以观察到 VGA 显示器给出的一幅带 Altera 标识蓝色背景图片
- 移动鼠标会观察到屏幕上相应的光标动作
- 点击鼠标左键会在屏幕上光标位置绘制白色的点，右键抹除白色的点

*请注意，执行 DE2_115_NIOS_HOST_MOUSE_VGA\demo_batch\nios_host_mouse_vga.bat 会自动下载软硬件到 DE2-115 开发板。*

图 6-4 给出了设计范例运行环境构架图

图 6-4 USB 画笔设计范例运行环境构架

## 6.4 USB 设备

很多 USB 类的应用和产品是作为一个 USB 设备来使用，而不是 USB 主机。在这个设计范例中，我们演示了如何将 DE2-115 开发板作为一个 USB 设备连接到 PC。正如图 6-5 所示，通过 DE2-115 开发板上的飞利浦 ISP1362 芯片及 Nios II 处理器用来和主机通信。

当连接 DE2-115 开发板的 USB 端口到主机后，Nios II 处理器必须执行一段软件代码，用以初始化 ISP1362 芯片。当初始化完成之后，主机为识别出 DE2-115 上的 USB 设备并询问相关的驱动程序；设备会被认为是飞利浦 PDIUSBD12 SMART Evaluation Board。在主机上完成相关驱动的安装后，下一步是在主机上执行名为 ISP1362DcUsb.exe 的程序，通过它来和 DE2-115 上的开发板通信。

在 ISP1362DcUsb 程序界面中，点击 Add 按钮，主机会发送一个特定的 USB 数据包到 DE2-115 开发板；Nios II 处理器会接收到这个数据包，并将硬件计数器的值加一。计数器的值会在七段数码管上显示出来。如果用户点击界面上的 Clear 按钮，会清零硬件计数器值。

图 6-5 USB 设备设计范例系统框图

## ■ 准备工作、文件位置及操作步骤

- 工程目录：DE2_115_NIOS_DEVICE_LED
- 使用到的配置数据：DE2_115_NIOS_DEVICE_LED.sof
- Nios II 工程目录： DE2_115_NIOS_DEVICE_LED\Software
- Borland C++软件驱动：DE2_115_NIOS_DEVICE_LED\SW
- 通过 USB 电缆连接 DE2-115 开发板从设备接头到 PC 主机
- 下载配置数据到 FPGA（*）
- 运行 Nios II IDE，切换工作目录到 DE2_115_NIOS_DEVICE_LED\Software，点击 Run（*）
- 电脑会检测到新的 USB 设备，选择驱动为 DE2_115_NIOS_DEVICE_LED\D12test.inf (Philips PDIUSBD12 SMART Evaluation Board)，忽略驱动安装过程中的警告信息
- 主机会报告说新的飞利浦 PDIUSBD12 SMART Evaluation Board 已经安装完毕
- 执行主机端程序 DE2_115_NIOS_DEVICE_LED\SW\ISP1362DcUsb.exe，点击 ADD 或者 Clear 按钮并观察 DE2-115 开发板的反应。
-

*请注意，执行 DE2_115_NIOS_DEVICE_LED\demo_batch\nios_device_led.bat 会自动下载.sof 和.elf 到 DE2-115 开发板。*

图 6-6 给出了设计范例运行环境的构架

图 6-6 USB 设备设计范例运行环境构架

## 6.5 卡拉 OK 机

这个设计范例使用 DE2-115 开发板上的麦克风输入、线路输入以及线路输出端口组成一个卡拉 OK 机。Wolfson WM78731 音频编解码芯片配置为主机模式，在这种模式下，编解码芯片自己生成 AD/DA 串行位时钟（BCK）以及左/右声道时钟信号（LRCK）。正如图 6-7 所示，芯片的 I2C 界面用于配置音频 CODEC。CODEC 芯片的采样率就是通过这个接口来设定的，从线路输入的音频信号会和从麦克风输入的信号混合，然后送到线路输出端口。

在这个设计范例中，采样率设置为 48kHz。按下 KEY0 会通过 I2C 总线调整音频编解码芯片的增益，增益值在 10 个预定义的值间循环。

图 6-7　卡拉 OK 机设计范例系统框图

## ■ 准备工作、文件位置及操作步骤

- 工程目录：DE2_115_i2sound
- 使用到的配置数据：DE2_115_i2sound.sof or DE2_115_i2sound.pof
- 连接麦克风到 DE2-115 开发板麦克风输入端口（粉红色插孔）
- 连接音乐播放器，譬如 MP3 播放器或者电脑的音频输出到 DE2-115 的线路输入端口（蓝色插孔）
- 连接耳机或者有源音箱到 DE2-115 开发板的线路输出端口（绿色插孔）
- 执行 DE2_115_i2sound\demo_batch folder 目录下的‘DE2_115_i2sound’，下载配置数据
- 现在可以听到耳机/音箱播放的来自麦克风和音乐播放器的混合声音了
- 按下 KEY0 调整音量，音量大小会在 0-9 级间循环

*请注意，如果在 HSMC 接口上装配有 HSMC 回环测试适配器，则因回环之缘故，HSMC 上的 I2C_SCL 信号将会连接到 I2C_SDA。由于 TV 解码芯片和音频芯片亦共用此 I2C 总线，从而它们也无法得到正确的配置而正常工作。*

图 **6-8** 给出了演示范例运行环境的构架。

图 6-8 卡拉 OK 机运行环境构架

## 6.6 SD 卡设计范例

很多应用场合使用大容量的外部存储设备来存储数据，譬如 SD 卡或者 CF 卡。DE2-115 开发板提供了存取 SD 卡所需的硬件和软件环境。在这个设计范例中，我们展示了如何浏览 SD 卡根目录中的文件并读取指定文件的内容。SD 卡需要预先格式化为 FAT 文件系统。设计范例支持长文件名。

图 6-9 给出了设计的硬件系统框图。FPGA 系统所需求的 50MHz 时钟由外部晶振产生。内部 PLL 模块生成 100MHz 频率给 Nios II 处理器及其它控制器使用。四根 PIO 引脚连接到 SD 卡插槽，并使用 SD 卡 4 比特模式存取 SD 卡。SD 卡 4 比特模式协议以及 FAT 文件系统均由 Nios II 软件实现。软件代码存放在在 SOPC 片上存储器中。

图 6-9　SD 卡设计范例系统框图

图 6-10 给出了设计范例的软件结构。Nios II PIO 部分提供基本的 IO 函数，用来直接访问硬件。这些函数由 Nios II 系统提供，函数原型可以在头文件 io.h 中找到。SD 卡部分实现了 SD 卡 4 比特模式协议，负责和 SD 卡进行通信。FAT 文件系统提供读取 FAT16 以及 FAT32 文件系统的接口函数。调用 FAT 公共函数，用户可以浏览 SD 卡根目录下的文件，并且，用户可以打开指定的文件并读取里面的内容。

Main 模块实现了设计范例的控制协调工作。当程序执行后，它首先检测系统是否存在 SD 卡。当找到 SD 卡后，它判断 SD 卡是否被格式化为 FAT 文件系统。如果条件满足，它会搜索根目录下的文件，然后把它们的名字打印到终端。如果名为“test.txt”的文件存在，程序也会读取此文件里面的内容，并打印到终端。当程序正确识别到 FAT 文件系统后，它还会点亮绿色的 LED 灯；否则，如果初始化 SD 卡文件系统失败或者找不到 SD 卡，它会熄灭绿色 LED 灯。用户按下 KEY3，可以复位系统 CPU，重新执行软件。

图 6-10　SD 卡设计范例软件框图

## 设计范例源代码

- 工程目录：DE2_115_SD_CARD
- 使用到的配置数据： DE2_115_SD_CARD.sof
- Nios II 工程目录：DE2_115_SD_CARD\Software

## 设计范例批处理文件

批处理文件目录： DE2_115_SD_CARD \demo_batch

批处理命令调用的文件：

- 脚本文件：DE2_115_SD_Card.bat, DE2_115_SD_CARD_bashrc
- 硬件配置文件：DE2_115_SD_CARD.sof
- 软件代码：DE2_115_SD_CARD.elf

## 设计范例设置

- 确保 Quartus II 以及 Nios II EDS（9.1sp2 或以上版本）已经在 PC 上安装好
- 开启 DE2-115 电源
- 连接 USB 电缆到 DE2-115 开发板，如有必要，安装 USB-Blaster 驱动
- 执行 DE2_115_SD_CARD\demo_batch 目录下的批处理文件“DE2_115_SD_Card.bat”
- 当 Nios II 代码下载好并成功执行后，Nios II 终端窗口会打印执行情况
- 拷贝测试文件到 SD 卡根目录
- 将 SD 卡插到 DE2-115 开发板 SD 卡插槽，如图 **6-11** 所示
- 按下 KEY3，开始读取 SD 卡的内容
- 程序会将 SD 卡测试文件中内容打印到终端，如图 **6-12** 所示

图 6-11　装载 SD 卡到 DE2-115

图 6-12　SD 卡设计范例运行结果

## 6.7 SD 卡音乐播放器

许多商品多媒体/音频播放器采用大容量的外接媒介，如 SD 卡或者 CF 卡，来存储音乐或者视频文件。这些播放器也通常包含有高品质 DAC 芯片，提供高品质的音频信号。DE2-115 开发板提供有存取 SD 卡所需的软件和硬件环境以及专业品质的音频，用户可以使用这些资源去设计高级的多媒体产品。

在这个设计范例中，我们在 DE2-115 上实现了一个 SD 卡音乐播放器，它读取存储在 SD 卡里面的音频文件，并通过 CD 品质的音频 DAC 芯片播放出来。设计使用 Nios II 处理器读取音乐数据并用 Wolfson WM8731 音频 CODEC 芯片完成播放工作。

图 6-13 给出了设计的系统框图。系统所需 50MHz 时钟信号由开发板上的晶振提供，内部锁相环模块用来生成 Nios II 处理器及其它外设所需的 100MHz 时钟。音频芯片通过自定义的

SOPC 外设—音频控制器控制。音频控制器需要一个 18.432MHz 频率的时钟信号，它由 PLL 模块生成。音频控制器需要音频 CODEC 芯片工作在主设备模式，这样它可以自己生成串行位时钟（BCK）以及左/右声道时钟信号。七段数码管由 SOPC 组件 SEG7 Controller 控制，它也是一个自定义的 SOPC 外设。两个 PIO 引脚连接到系统 I2C 总线，协议由软件模拟实现。四根 PIO 引脚连接到 SD 卡插槽。IR 接收器由 IR Controller 控制。Nios II 软件同时实现了 SD 卡 4 比特模式访问协议。

图 6-13　SD 卡音乐播放器设计范例系统框图

图 6-14 给出了设计的软件结构。SD 4-bit Mode 模块用来实现 SD 卡 4 比特模式协议，从 SD 卡中读取原始数据。FAT 模块实现了 FAT16/32 文件系统，让上层应用程序可以读取存储在 SD 卡里面的文件（仅实现了部分 FAT 文件系统函数）。WAVE Lib 模块用来解码 WAVE 文件，从中提取音频数据。I2C 模块实现了配置音频芯片所用的 I2C 协议。SEG7 模块用来控制七段数码管显示音乐持续播放时间信息。音频模块负责检查音频数据 FIFO 和完成收发函数。IR 模块实现音乐播放过程的控制。

图 6-14　SD 卡音乐播放器设计范例软件结构

音频芯片在接收音频信号之前需要进行一些参数的设置。主函数用 I2C 协议以使音频芯片工作在主设备模式，音频输出接口工作在 I2S 16 比特/声道模式；采样率则与原始 WAVE 文件的采样率保持一致。在音频播放过程中，主函数每次从 SD 卡读取 512 字节的音频数据，然后写入到 Audio Controller 中的 DAC FIFO。写入之前，程序会检测 FIFO 是否为满。如果使能 BYPASS 以及 SITETONE 功能的话，设计会将麦克风输入和线路输入的信号混合在一起，产生卡拉 OK 的效果。

最后，用户可以在 16x2 LCD 模块、七段数码管上获取到所播放的 SD 卡音乐长度。LCD 的第一行和第二行分别显示正在播放的文件名和音量。七段数码管显示已播放的时间。LED 指示播放的音频信号的强度。

## ■ 准备工作、文件位置及操作步骤

- 工程路径：DE2_115_SD_Card_Audio_Player
- 使用到的配置文件：DE2_115_SD_Card_Audio_Player.sof
- Nios II 工程路径：DE2_115_SD_Card_Audio_Player\Software
- 格式化 SD 卡位 FAT16 或者 FAT32
- 将音频 WAVE 文件存放到 SD 卡的根目录里面。所提供的波形文件采样率必须是 96K, 48K, 44.1K 32K 或者 8K，采样深度必须为 16 比特
- 下载配置数据到 FPGA（*）
- 运行 NiosII IDE 并且切换工作目录到 DE2_115_SD_Card_Audio_Player\Software（*）
- 连接耳机或者有源音箱到 DE2-115 的线路输出端口，你应该可以听到从 SD 卡播放的音乐
- 按下按钮 KEY3 会播放下一个音乐文件
- 按下 KEY2 可以增大音量，KEY1 减小音量
- 用户也可以使用遥控器用于播放控制，具体的按键功能如表 6-1 所示

**表 6-1 遥控器按键功能详细定义**

| *按键名* | *功能描述* |
|---|---|
| **PLAY** | 播放或暂停音乐 |
| **CHANNEL** | 选择欲播放的上一首/下一首音乐文件 |
| **VOLUME** | 增加/降低音量 |
| **MUTE** | 静音/取消静音 |

请注意：

1. 执行批处理文件 DE2_115_SD_Card_Audio_Player\demo_batch

\ DE2_115_SD_Card_Audio_Player.bat 自动下载硬件和软件数据到 DE2-115

2. 如果 SD 卡容量在 8G 或以上，请注意选择速度等级为 Class 4 或以上以保证音乐的读取速度

3. 如果在 HSMC 接口上装配有 HSMC 回环测试适配器，则因回环之缘故，HSMC 上的 I2C_SCL 信号將会连接到 I2C_SDA。由于 TV 解码芯片和音频芯片亦共用此 I2C 总线，从而它们也无法得到正确的配置而正常工作

图 6-15 给出了设计范例的运行环境构架。

图 6-15 SD 卡设计范例运行环境构架

## 6.8 PS/2 鼠标控制器设计范例

这个设计范例给出了一个 PS/2 鼠标控制器实例，展示如何在 PS/2 鼠标控制器和设备-PS/2 鼠标-间进行双向通信。用户可以把这个设计当作进一步实践的基础，完成更复杂的通信操作，譬如实现双字节指令的发送、四按键鼠标数据的收集。

如要了解详细的 PS/2 协议相关信息，可以自行搜索各种教育类网站。下面是 PS/2 协议的简述：

### ■ 概述

PS/2 协议使用两根信号线进行双向的通信，一根时钟线，一根数据线。时钟线上的时钟信号由 PS/2 设备产生，PS/2 控制器则对信号线拥有全部的控制权。

## ■ 设备到控制器的数据传输

PS/2 鼠标处于流工作模式时，如果接收到一条数据发送使能指令，PS/2 设备便开始将采集到的位移/按键数据连续不断的报告给控制器。每个数据包由 33 个比特（不使能滚轮时）组成，它可以被分成三个相似的片段，每个片段均由一个起始位（总是零）、8 个数据位（LSB 在前）、一个奇偶校验位（奇校验）、一个停止位（总是 1）组成。

PS/2 控制器在时钟信号的下降沿采样输入信号。在硬件上，这可以用一个 33 比特的移位寄存器来实现，但是需要注意跨时钟域的信号处理。

## ■ 控制器到设备的数据传输

任何时候，如果控制器想要发送数据到设备，它会先将时钟线拉低约超过一个时钟周期时间，用以抑制现有的通信过程或者发起一次新的传输，这通常被叫做抑制态。随后，控制器拉低数据线，然后释放时钟线，这个过程被称为请求态。释放时钟信号所形成之上升沿，即为数据起始位（因为此时数据线为低，故其值为零）。设备会检测到这一系列动作，然后在不超过 10ms 的时间内给出时钟信号。每次数据传输过程传送共 12 比特，包含一个起始位（如前所述，其值为零），8 个数据位，一个奇偶校验位（奇校验），一个停止位（其值为一），一个握手位（其值为零）。在发送完奇偶校验位之后，控制器应该释放数据线，设备会在接下来的时钟周期内检测数据线上的变化，并校验接收到的数据，如果数据信号没有变化（为固定高电平），则在接下来的一个时钟周期拉低数据线，作为握手信号。

PS/2 鼠标复位后，会进入流工作模式，并处在禁止数据发送的状态，除非它接收到一条数据发送使能指令。图 **6-16** 给出了信号线上发送或接收数据时的波形。

图 6-16 PS/2 发送/接收波形示意图

## ■ 准备工作、文件位置及操作步骤

- 工程目录： DE2_115_PS2_DEMO
- 使用到的配置数据：DE2_115_PS2_DEMO.sof
- 执行批处理文件 DE2_115_PS2_DEMO\demo_batch\DE2_115_PS2_DEMO.bat 以下载配置信息
- 接入 PS/2 鼠标
- 按下 KEY0 以使能鼠标数据发送
- 按下 KEY1 清零七段数码管显示的累积位移数据

当鼠标有移动时，你应该可以观察到七段数码管显示的数字跟随变动。当按下左中右键时，相应的绿色 LED 灯 LED[2:0]会点亮。表 6-2 给出了详细的信息。

表 6-2 按键/位移指示信息

| *指示器名* | *说明* |
|---|---|
| **LEDG[0]** | 指示左键是否按下情况 |

| | |
|---|---|
| LEDG[1] | 指示右键是否按下情况 |
| LEDG[2] | 指示中间键是否按下情况 |
| HEX0 | 低字节的 X 位移 |
| HEX1 | 高字节的 X 位移 |
| HEX2 | 低字节的 Y 位移 |
| HEX3 | 高字节的 Y 位移 |

图 6-17 给出了设计范例的运行环境构架

图 6-17 PS/2 鼠标设计范例运行环境构架

## 6.9 IR 接收器设计范例

在这个设计范例中，遥控器发送的按键编码信息会在 DE2-115 上显示出来。用户只需将遥控器对准 DE2-115，然后按下某个按键。DE2-115 在解码接收到的数据后，相关的信息会在七段数码管上显示出来，它们是用户码、设备码及设备码的反码。用户码用来标识所按按键的编码，设备码用来标识特定遥控器。

下面介绍 IR 接收器如何解调接收到的信号。

红外遥控器有按键按下时，会发送一个数据帧，波形如图 6-19 所示。帧数据的起始是一串引导码，然后是按键相关编码信息，最后是一个结束比特。

图 6-18　遥控器外形

表 6-3　遥控器按键编码信息

| 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 |
| ▶II | 0x16 | ◀ | 0x14 | ▶ | 0x18 | 🔇 | 0x0C |

图 6-19　IR 遥控器发射波形（基带信号）

当 DE2-115 接收到数据帧后，一体化红外接收头的数据输出会送给 FPGA，FPGA 内部实现了帧数据解码所需电路。电路主要包括三个部分，编码检测器、状态机及移位寄存器，如图 6-20 所示。编码检测器检测引导码并将信息回馈给状态机。

一旦检测到引导码，状态机即转移到 GUIDANCE 状态。当编码检测器完成用户码的检测后，状态机转入 DATAREAD 状态。在这个状态，用户码、反码及设备码信息会移入移位寄存器，并译码输出给七段数码管。图 6-21 给出了状态机状态转移过程。此设计须使用 50MHz 时钟输入。

图 6-20　IR 接收控制器框图

图 6-21　状态机状态转移图

用户可以在将此设计集成到很多应用中，譬如用来控制 SD 卡音乐播放器。

## ■ 准备工作、文件位置及操作步骤

- 工程目录：DE2_115_IR
- 使用到的配置文件： DE2_115_IR.sof
- 执行 DE2_115_IR\demo_batch\DE2_115_IR.bat，下载配置数据到 FPGA
- 将遥控器指向 DE2-115 开发板，按下某一按键。

表 6-4 给出了数码管所显示信息对应的编码段。

表 6-4 指示器详细信息

| *指示器名* | *说明* |
|---|---|
| **HEX0** | 关键码低字节反码 |
| **HEX1** | 关键码高字节反码 |
| **HEX2** | 关键码低字节 |
| **HEX3** | 关键码高字节 |
| **HEX4** | 自定义码低字节 |
| **HEX5** | 自定义码高字节 |
| **HEX6** | 重复自定义代码低字节 |
| **HEX7** | 重复自定义代码高字节 |

图 6-22 给出了设计范例运行环境构架。

图 6-22 IR 接收器设计范例运行环境构架

## 6.10 音乐合成器设计范例

这个设计范例展示了如何使用 DE2-115 开发板以及一个 PS/2 键盘和扬声器，来实现多音电子键盘。

PS/2 键盘模拟钢琴键盘用于输入按键信息。DE2-115 上的 FPGA 芯片用来实现音乐合成器，生成不同音调的音频信号。连接到 DE2-115 开发板上的 VGA 显示器用来显示按键动作。

图 6-23 给出了音乐合成器的系统框图。它主要由四个模块构成，DEMO_SOUND,，PS2_KEYBOARD，STAFF 以及 TONE_GENERATOR。DEMO_SOUND 模块预存有音乐资料。PS2_KEYBOARD 处理来自 PS/2 键盘的用户输入。STAFF 模块画出显示在 VGA 上的键盘图像，并在用户按下某个按键时，更新相应的图像。TONE_GENERATOR 是音乐合成器的核心部分。

使用 SW9，用户可以切换输入来源位置，可以是 PS2_KEYBOARD 或者 DEMO_SOUND。如要重复播放预存音乐，可以按下 KEY1。

TONE_GENERATOR 有两种音调：String 和 Brass，它由 SW0 控制。DE2-115 上的音频编解码器芯片有两个声道。它们通过 SW1 和 SW2 进行开关控制。

图 6-24 给出了设计范例运行环境的构建。

图 6-23 音乐合成器设计系统框图

## ■ 准备工作、文件位置及操作步骤

- 工程路径：DE2_115_Synthesizer
- 使用到的配置文件：DE2_115_Synthesizer.sof or DE2-115_Synthesizer.pof
- 连接 PS/2 鼠标到 DE2-115 开发板
- 将 DE2-115 开发板的 VGA 输出连接到 VGA 显示器（LCD 或者 CRT 均可）
- 将 DE2-115 的线路输出连接到扬声器
- 执行 DE2_115_Synthesizer\demo_batch\DE2_115_Synthesizer.bat 下载配置文件
- 确保 SW[9:0]在 DOWN 的位置
- 按下 KEY1，开始运行设计范例
- 按下 KEY0 复位 DE2-115 电路

*请注意，如果在 HSMC 接口上装配有 HSMC 回环测试适配器，则因回环之缘故，HSMC 上的 I2C_SCL 信号将会连接到 I2C_SDA。由于 TV 解码芯片和音频芯片亦共用此 I2C 总线，从而它们也无法得到正确的配置而正常工作。*

表 6-5 及表 6-6 给出了开关、按钮开关及 PS/2 键盘各按键对应的功能

拨动开关和按钮开关

**表 6-5 拨动开关以及按钮开关功能**

| *信号名* | *说明* |
|---|---|
| **KEY[0]** | 重置电路 |
| **KEY[1]** | 重复播放范例中的音乐 |
| **SW[0]** | 关:铜管乐，开：弦乐 |
| **SW[9]** | 关：范例，开: **PS/2** 键盘 |
| **SW[1]** | 通道 **1 ON / OFF** |
| **SW[2]** | 通道 **2 ON / OFF** |

PS/2 键盘

**表 6-6 PS/2 键盘按键功能**

| *信号名* | *说明* |
|---|---|
| **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 |

图 6-24　音乐合成器设计范例运行环境构架

## 6.11 音乐录制和回放设计范例

这个设计范例展示如何使用 DE2-115 上的音频 CODEC 芯片和 FPGA 来实现音乐播放和录制功能。这个设计范例基于 SOPC 构架。图 6-25 给出了设计范例的人机界面。两个按钮开关以及两个拨动开关用作系统的用户控制：SW0 用于指定录音的音源是 MIC 输入还是线路输入。

SW1 用来设定是否使能过零点检测。SW3，SW4 及 SW5 用来指定录音采样率。16x2 LCD 模块指示录音/回放状态信息。七段数码管用来显示录制/回放的持续时间（单位为毫秒）。LED 用来指示音频信号的强度。表 6-7 及表 6-8 列出了在控制音乐录制和播放中各个拨动开关的作用。

图 6-25　音频录制/回放设计人机界面

图 6-26 给出了设计的系统框图。设计分为软硬件两个部分，软件部分作为代码存储在 SRAM 中，在 Nios II IDE 开发环境中用 C 写成。硬件部分用 SOPC 生成器在 Quartus II 下开发而成。AUDIO Controller 是一个自定义的 SOPC 模块。它被用于发送音频数据到音频芯片，或者从音频芯片接收数据。

音频芯片通过由 C 写成的 I2C 协议来得到配置。音频芯片的两个 I2C 引脚通过 PIO 控制器直接连接到 SOPC 系统互连总线。在这个范例中，音频 CODEC 工作主设备模式。外部音频界面为 I2S 16 比特模式。由 PLL 模块生成的 18.432MHz 时钟信号连接到芯片的 XTI/MCLK 引脚。

图 6-26　音乐录制/回放设计系统框图

## ■ 准备工作、文件位置及操作步骤

- 硬件工程目录：DE2_115_AUDIO
- 用到的配置文件：DE2_115_AUDIO.sof
- 软件工程目录：DE2_115_AUDIO\software\
- 将音频信号源连接到 DE2-115 开发板的线路输入接口
- 连接麦克风到 DE2-115 麦克风输入接口
- 连接耳机或者有源音箱到 DE2-115 线路输出端口
- 下载 FPGA 配置数据（*）
- 下载软件到 DE2-115 开发板
- 使用拨动开关控制播放/录制系统
- 按下 DE2-115 开发板上的 KEY3 开启/停止录音
- 按下 DE2-115 开发板上的 KEY2 开启/停止音频回放

*请注意：*

*(1). 执行 DE2_115_AUDIO\demo_batch\audio.bat 自动下载..sof and .elf 文件到 DE2-115.*

*(2). 在音频缓冲写满后，录制过程会自动结束*

*(3). 播放过程在播放所录制之音乐结束后自动停止*

*(4). 如果在 HSMC 接口上装配有 HSMC 回环测试适配器，则因回环之缘故，HSMC 上的 I2C_SCL 信号将会连接到 I2C_SDA。由于 TV 解码芯片和音频芯片亦共用此 I2C 总线，从而它们也无法得到正确的配置而正常工作。*

表 6-7 拨动开关用于选择音源

| *拨动开关* | *0 – 开关关* | *1 – 开关开* |
|---|---|---|
| **SW0** | 音频源于麦克风 | 音频源于 **LINE-IN** |
| **SW1** | 禁用麦克风 | 启用麦克风 |
| **SW2** | 禁用过零点侦测 | 启用过零点侦测 |

表 6-8 拨动开关用于控制选择采样率

| *SW5 (0 –关;1- 开)* | *SW4 (0 –关;1-开)* | *SW3 (0 –关;1-开)* | *采样率* |
|---|---|---|---|
| **0** | **0** | **0** | **96K** |
| **0** | **0** | **1** | **48K** |
| **0** | **1** | **0** | **44.1K** |
| **0** | **1** | **1** | **32K** |
| **1** | **0** | **0** | **8K** |
| 未列出的组合 | | | **96K** |

## 6.12 网页服务器设计范例

这个设计范例给出了一个基于 Nios II 处理器的 HTTP 网页服务器，它藉由 MicroC/OS-II 操作系统上的 NicheStack™ TCP/IP 协议栈来实现，网页内容存放在 DE2-115 开发板的 Flash 芯片中。网页服务器可以处理基本的一些网络请求，包括 HTML，JPEG，GIF，PNG，JS，CSS，SWF，ICO 等文件，这些内容包含在一个 Aletera 只读的.zip 文件系统中。另外，它允许用户从网页上控制 DE2-115 开发板上不同的外设。

作为 Nios II EDS 的一部分，NicheStack™ TCP/IP 网络协议栈是一个为 Nios II 在网络方面应用所提供的最优解决方案。

使用这个 demo，我们假设用户具备 TCP/IP 方面的基础知识。

下面介绍系统硬件的主要部分 SOPC 系统，它主要由 Nios II 处理器、片上存储单元、JTAG

UART、定时器、三倍速以太网控制器、SGDMA 控制器等组成。在 Altera 三倍速以太网控制器的设计界面中，用户可以选择其 MAC 界面为 MII 或者 RGMII，如图 6-27、图 6-28 所示。

图 6-27　MII 界面 MAC 配置

图 6-28　RGMII 界面 MAC 配置

在 MAC 选项标签页面（参考图 6-29），用户应该给以太网控制器设置合适的值。控制器须包含 MDIO 模块，它用于给 PHY 芯片生成 2.5MHz 的 MDC 时钟，这个频率从控制器主时钟分频而得，故其分频系数应该为主时钟频率和 MDC 频率的比值。

图 6-29 MAC Options Configuration

当参数都设置好后，点击 Finish，完成以太网控制器设定过程并返回 SOPC 页面如图 6-30 所示。

| Use | Connections | Module Name | Description | Clock | Base | End | Tags | IRQ |
|---|---|---|---|---|---|---|---|---|
| ☑ | | usb | ISP1362_IF | | | | | |
| | | hc | Avalon Memory Mapped Slave | altpll_sys | 0x134424d8 | 0x134424df | | 5 |
| | | dc | Avalon Memory Mapped Slave | altpll_sys | 0x134424e0 | 0x134424e7 | | 6 |
| ☑ | | sdram | SDRAM Controller | | | | | |
| | | s1 | Avalon Memory Mapped Slave | altpll_sys | 0x08000000 | 0x0fffffff | | |
| ☑ | | tse_mac | Triple-Speed Ethernet | | | | | |
| | | transmit | Avalon Streaming Sink | altpll_sys | | | | |
| | | receive | Avalon Streaming Source | altpll_sys | | | | |
| | | control_port | Avalon Memory Mapped Slave | altpll_sys | 0x13442000 | 0x134423ff | | |
| ☑ | | sgdma_tx | Scatter-Gather DMA Controller | | | | | |
| | | csr | Avalon Memory Mapped Slave | altpll_sys | 0x13442400 | 0x1344243f | | 7 |
| | | descriptor_read | Avalon Memory Mapped Master | | | | | |
| | | descriptor_write | Avalon Memory Mapped Master | | | | | |
| | | m_read | Avalon Memory Mapped Master | | | | | |
| | | out | Avalon Streaming Source | | | | | |
| ☑ | | sgdma_rx | Scatter-Gather DMA Controller | | | | | |
| | | csr | Avalon Memory Mapped Slave | altpll_sys | 0x13442440 | 0x1344247f | | 8 |
| | | descriptor_read | Avalon Memory Mapped Master | | | | | |
| | | descriptor_write | Avalon Memory Mapped Master | | | | | |
| | | m_write | Avalon Memory Mapped Master | | | | | |
| | | in | Avalon Streaming Sink | | | | | |
| ☑ | | descriptor_me... | On-Chip Memory (RAM or ROM) | | | | | |
| | | s1 | Avalon Memory Mapped Slave | altpll_sys | 0x13440000 | 0x13440fff | | |

图 6-30 SOPC 生成器

图 6-31 给出了 MII 模式下 88E1111 PHY 芯片和 MAC 控制器间的连接示意图。

图 6-31 MII 模式 PHY 芯片与 MAC 控制器间连接示意图

图 6-32 给出了 RGMII 模式下的连接示意图。

图 6-32　RGMII 模式下 PHY 芯片和 MAC 控制器间连接示意图

当硬件一切设置完毕后，开始构建此硬件上的运行软件，其基础结构如图 6-33 所示。最底层是硬件相关信息，然后是硬件抽象层，运行于其上的是 MicroC/OS-II 操作系统，最后是 NicheStack TCP/IP 协议栈和其上的网页服务器应用程序。

图 6-33 Nios II 软件结构

最后，是关于网页服务器软件流程的描述。

首先，网页服务器初始化 MAC 和网络设备，然后调用 get_mac_addr()函数为 PHY 芯片设置合适的 MAC 地址。第二步，它初始化自动协商过程，检查 PHY 和网关设备间的连接。如果连接存在，PHY 和网关会分别广播它们的传输参数，传输速度，双工工作模式等信息。第三步，网页服务器会为建立好的链路准备好发送和接收通道，准备好后，随即调用 get_ip_addr()函数为 DE2-115 设定 IP。最后，IP 分配成功后，NicheStack™ TCP/IP 协议栈开始执行网页服务器应用程序。

图 6-34 给出了网页服务器演示范例运行环境的构建。

*请注意，演示程序所需的网关应该支持 DHCP 协议，因为程序使用 DHCP 服务来获取合法的 IP 地址，否则，你可能需要重新设定系统库属性，为程序指定静态 IP 地址。另外，网页服务器设计范例 MAC 层使用 RGMII 或者 MII 界面，用户可通过设定 JP1 和 JP2 跳线开关的位置来为 Ethernet0 或 Ethernet1 设定不同的 MAC 层接口。表 6-9 给出了不同的工程所分属的 Ethernet 芯片和 MAC 接口类型。*

表 6-9 演示范例工程目录

| PHY<br>工程<br>目录<br>界面 | ENET0 | ENET1 |
|---|---|---|
| RGMII 界面 | DE2_115_Web_Server\<br>DE2_115_WEB_SERVER_RGMII_ENET0 | DE2_115_Web_Server\<br>DE2_115_WEB_SERVER_RGMII_ENET1 |
| MII 界面 | DE2_115_Web_Server\<br>DE2_115_WEB_SERVER_MII_ENET0 | DE2_115_Web_Server\<br>DE2_115_WEB_SERVER_MII_ENET1 |

## ■ 准备工作、文件位置及操作步骤

下面以工作在 RGMII 模式下 ENET0 芯片的网页服务器设计为例，讲述如何架设 Demo

- 工程目录： DE2_115_Web_Server\ DE2_115_WEB_SERVER_RGMII_ENET0
- Nios II 工作目录： Project directory\software
- 使用到的硬件配置文件： DE2_115_WEB_SERVER.sof
- 网页内容 zip 文件：ro_zipfs.zip
- 确保 Ethernet0 芯片工作在 RGMII 模式（短接 JP1 的第一、二引脚）
- 运行 Qauartus II，下载硬件配置到 FPGA
- 运行 Nios II IDE，切换工作目录到前述路径
- 使用 Nios II IDE Flash Programmer 下载.zip 文件到 Flash
- 将 DE2-115 Ethernet0RJ-45 网络口连接到网关
- 在 Nios II IDE 中，选择 Run->Run as Nios II Hardware 运行软件工程
- 当 DE2-115 上的 LCD 显示出正确的 IP 地址后，在 PC 机上开启浏览器
- 在浏览器地址栏输入 LCD 上所显示的 IP 地址
- 现在，你应该可以看到全新的 DE2-115 网页
- 在这个页面上，你可以通过左边框栏来访问 DE2-115 上不同的外设，你可以设定 LED 亮

灭状态，可以向 LCD 输出信息或者设定七段数码管显示的数值。图 6-35 给出了网页服务器设计页面截图

*请注意，执行 DE2_115_Web_Server\<Web Server Mode-Port Specific>\demo_batch\web_server.bat 批处理文件自动下载软硬件配置到 DE2-115 开发板*

图 6-34 系统运行环境构建框图

图 6-35 DE2—115 网页服务器网页截图

# 第7章.

# 附录

## 7.1 修改历史

| *Version* | *Change Log* |
|---|---|
| **V1.0** | **Initial Version (Preliminary)** |
| **V1.1** | 修改 **6.2** 章节 |

## 7.2 版权声明

版权所有 © 2011 友晶科技. 保留所有权利.