

Xi'An Intelligent Silicon Technology Co.

(XiST)

XSSL12-F256-V1

用户手册

Version 1.0 2019

西安智多晶微电子有限公司
XIAN Intelligent Silicon Technology
西安市高新区科技二路 72 号， 邮编 710075
西安软件园西岳阁 102 室
+86 029 88860-013
网址: <http://www.isilicontek.com>

版权声明

本手册版权归属西安智多晶微电子有限公司所有，并保留一切权力。非经本公司同意，任何单位及个人不得擅自摘录或修改本手册内容，违者我们将追究其法律责任。

在使用产品之前，请仔细地阅读该手册并且确保知道如何正确使用该产品，不合理的操作可能会损坏开发板，使用过程中随时参考该手册以确保正确使用。

版本信息

版本号	版权	发布日期	备注
Version 1.0	西安智多晶微电子有限公司	2019.3.18	

目 录

目 录	4
1. 关于本手册	7
1.1 适用范围	7
1.2 手册内容	7
1.3 相关文档	7
2. 开发板简介	8
2.1 概述	8
2.2 开发板套件	9
2.3 硬件框图	10
2.4 功能及特点	11
3. 开发板电路	13
3.1 FPGA	13
3.1.1 FPGA 资源	13
3.1.2 FPGA 核心架构	14
3.1.3 I/O 特性	14
3.1.4 时钟管理	14
3.1.5 外部存储器接口	15
3.2 电源	15
3.2.1 概述	15
3.2.2 电源电路	15
3.3 时钟资源	17
3.3.1 概述	17
3.3.2 时钟电路	17
3.3.3 引脚分配	17
3.4 按键	17
3.4.1 概述	17
3.4.2 按键电路	17
3.4.3 引脚分配	19

3.5 LED	19
3.5.1 概述	19
3.5.2 LED 电路	19
3.5.3 引脚分配	20
3.6 扩展口	20
3.6.1 概述	20
3.6.2 扩展口电路	20
3.6.3 引脚分配	21
3.7 单片机	26
3.7.1 概述	26
3.7.2 单片机电路	26
3.7.3 引脚分配	27
3.8 千兆以太网接口	27
3.8.1 概述	27
3.8.2 千兆以太网接口电路	28
3.8.3 引脚分配	29
3.9 SDRAM	30
3.9.1 概述	30
3.9.2 SDRAM 电路	30
3.9.3 引脚分配	31
3.10 USB-UART	33
3.10.1 概述	33
3.10.2 USB-UART 电路	33
3.10.3 引脚分配	35
3.11 FLASH	35
3.11.1 概述	35
3.11.2 FLASH 电路	35
3.11.3 引脚分配	37
3.12 EEPROM	37
3.12.1 概述	37
3.12.2 EEPROM 电路	37

3.12.3 引脚分配	38
3.13 配置模式	38
3.13.1 概述	38
3.13.2 JTAG 模式	38
3.13.3 AS 模式	39
3.13.4 PS 模式	错误！未定义书签。
3.13.5 跳线设置	40
4. 软件系统	40
4.1 关于本章节	40
4.2 HqFpga 简介	40
4.3 软件安装	41
4.3.1 HqFpga 安装	41
4.3.2 下载器驱动安装	42
4.4 例程说明	44
4.4.1 代码说明	44
4.4.2 创建工程	50
4.4.3 下载及固化	52
4.5 其他注意事项	54

1. 关于本手册

适用范围

本用户手册适用对象为西安智多晶微电子公司的 FPGA 开发板，型号：XSSL12-F256-V1。

手册内容

本手册包含如下内容：

- XSSL12-F256-V1 开发板的功能及特点
- 各部分硬件电路及引脚分配
- 软件使用说明及工程案例
- 其他注意事项

相关文档

XSSL12-F256-V1 开发板相关文档如下：

- Sealion 2000 FPGA Family Data Sheet
- Sealion (海狮) 2000/2000s 系列 FPGA 产品数据手册
- XSSL12-F256-V1 开发板原理图
- HqFpga 软件开发系统用户手册

2. 开发板简介

概述

XSSL12-F256-V1 开发板的核心器件为西安智多晶微电子有限公司的 SL2-12K FPGA 器件。该 FPGA 基于低功耗工艺，通过最低的成本实现较高的功能性，可用于无线、有线、广播、工业用户以及通信等行业中的低成本应用领域。

XIST 提供全正向设计的软硬件工具链，编译软件采用全自主开发的 HQFPGA 软件开发系统，在同一可执行环境下集成了从综合、布局、布线等所有 FPGA 实现功能。HQFPGA 还提供一键式 (Push-Button) 的运行模式，并提供多种界面语言，方便用户完成简单、快速、高效及优化的 FPGA 开发。

XSSL12-F256-V1 开发板具有丰富的硬件资源和外围接口，秉承简洁、实用、可靠的设计原则，旨在引导客户快速熟悉智多晶的 FPGA 产品。适用于软件无线电、工业控制、多媒体应用等项目开发初期验证，同时也可用于教学及培训。



图 2-1 开发板实物图

开发板套件

XSSL12-F256-V1 开发板套件包括：

- XSSL12-F256-V1 开发板
- Xist CPLD/FPGA 通用 USB 下载线：XSKIT-CABLE-USB
- 5V 电源供电线（MINI USB 口）
- XSSL12-F256-V1 用户手册

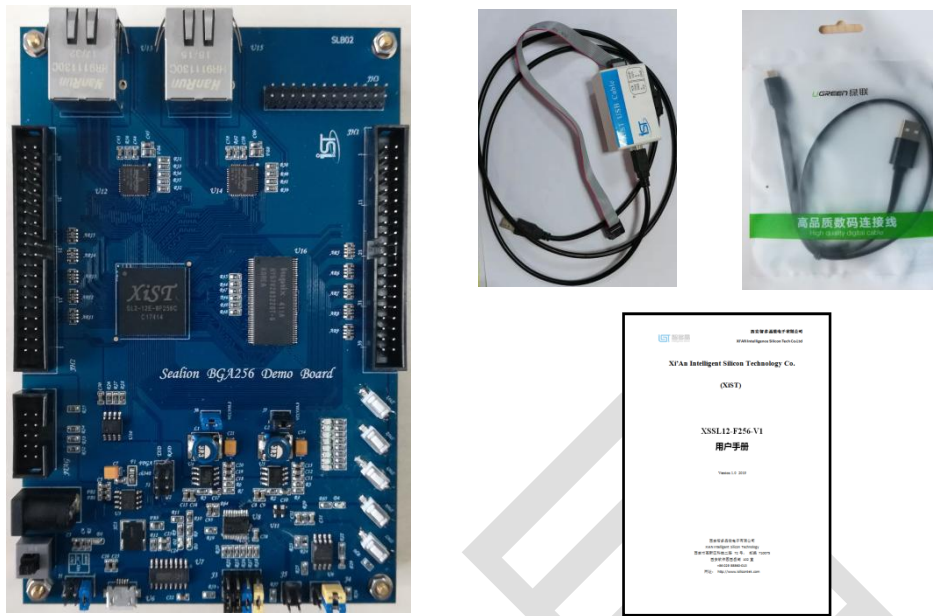


图 2-2 开发板套件

硬件框图

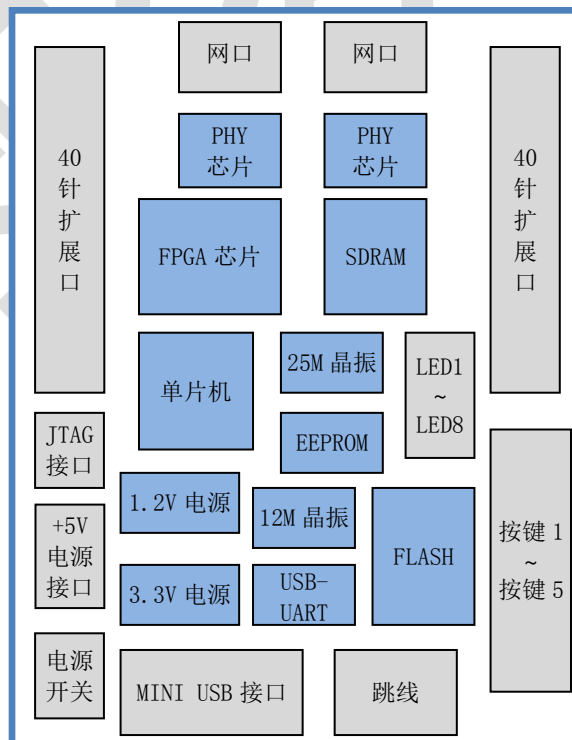


图 2-3 开发板硬件框图

功能及特点

- FPGA
 - 12K LE
 - 432Kb Block RAM
 - BGA256 封装，引脚个数 256
- 电源
 - 5V 电源输入，具有过流保护功能
 - 板上电压：5V、3.3V、1.2V
- 时钟资源
 - 25MHz 有源晶振
- 按键
 - 1 个单片机复位按键
 - 4 个用户按键
- LED
 - 1 个电源指示灯
 - 1 个配置完成指示灯
 - 8 个用户 LED
- 扩展口
 - 2 个双排 40 PIN 插针
 - 1 个双排 24 PIN 插针
 - 提供 5V、3.3V 和数字地
- 单片机
 - STC12LE5604AD，增强型 8051 内核
 - 加密性强，有全球唯一 ID 号
 - 8 通道，10 位 ADC

- 以太网
 - 2 个博通公司 B50610M PHY 芯片
 - 1000BASE-T/100BASE-TX/10BASE-T Gigabit
 - 支持 RGMII MAC interface
- SDRAM
 - Hynix 公司 HY57V283220-6
 - 存储容量：4 Banks x 1M x 32Bit
 - 最高时钟频率：166MHz
- USB-UART
 - 支持 5V 或 3.3V 电源
 - 全速 USB 设备接口
 - 提供 12MHz 时钟源输入
 - 硬件全双工串口，支持通讯波特率 50bps ~ 2Mbps
- FLASH
 - Winbond 公司 W25Q16
 - 存储容量：16Mbit
 - 接口类型：Standard SPI、Dual SPI、Quad SPI
- EEPROM
 - Microchip 公司 24LC04
 - 存储容量：4Kbit
 - IIC 接口
- 配置模式
 - JTAG 模式
 - AS 模式
 - PS 模式

3. 开发板电路

FPGA

FPGA 资源

SL2-12K 器件资源如下表所示：

表 3-1 SL2-12K 器件系列资源

器件		SL2-12K
逻辑单元		12032
嵌入式存储器 (Kbits)		432
嵌入式存储器单元数 (9kbits/单元)		48
嵌入式 18 × 18 乘法器(注释 2)		20
通用 PLL+DLL 数量		2+2
全局时钟网络(注释 3)		8+8
用户 I/O 模块		8
最大用户 I/O(注释 1)		186
核心工作电压		1.2V
芯片等级(注释 4)		C/I
封装规格		I/Os
F256	256 fBGA (17 x 17mm, 1.0mm)	186

注释：

- 1、管脚列表文件中的用户 I/O 管脚包括所有的通用 I/O 管脚、专用时钟管脚以及两用配置管脚。收发器管脚和专用配置管脚不包括在这一管脚列表中。
- 2、嵌入式 18 × 18 乘法器均为 DSP 硬核，可做乘法器兼累加器。
- 3、16 个全局时钟均包含 8 个主时钟及 8 个次时钟。
- 4、C：商业级，工作温度：0°C - 85°C
I：工业级，工作温度：-40°C - 100°C

FPGA 核心架构

SL2-12K 的核心构架由四输入查找表(LUTs)、逻辑单元存储器模块以及乘法器构成。

嵌入式存储器模块具有 9Kbit 的 SRAM 存储器。可以把嵌入式存储器模块配置成单端口、伪双端口、真双端口 RAM 以及 FIFO 缓冲器或者 ROM，通过配置也可以实现表 1-2 中的数据宽度。

表 3-2 SL2-12K 的嵌入式存储模块数据宽度

模式	数据宽度配置
单端口或伪双端口	×1, ×2, ×4, ×8/9, ×16/18, 和 ×32/36
真双端口	×1, ×2, ×4, ×8/9, 和 ×16/18

嵌入式乘法器模块，可以在单一模块中实现一个 18×18 或两个 9×9 乘法器。

I/O 特性

SL2-12K 的 I/O 支持可编程总线保持、可编程上拉电阻、可编程下拉电阻、可编程延迟、可编程驱动能力以及可编程 slew-rate 控制，从而实现了信号完整性以及热插拔的优化。

表 3-3 SL2-12K 所支持的 I/O 标准

类型	I/O 标准
单端 I/O	LVTTTL、LVCMOS、SSTL、HSTL 和 PCI
差分 I/O	LVDS、SSTL、HSTL、LVPECL、BLVDS、mini-LVDS 和

时钟管理

SL2-12K 器件包含了 8 个全局时钟(GCLK)网络，和 8 个高扇出网路。该器件有 2 个 PLL（每个 PLL 上均有五个输出端）和 2 个 DLL，以提供可靠的时钟管理与综合。

外部存储器接口

SL2-12K 器件支持 SDR、DDR、DDR2 等 SDRAM 接口。接口可能位于器件的一个或多个 I/O 模块（I/O bank），以实现更灵活的电路板设计。

电源

概述

开发板供电电压为+5V，具有过流保护功能。电源芯片采用 AX3514A，将输入+5V 转化成+3.3V 和+1.2V，最大输出电流 2A。5V 电源有两种输入接口，分别为 DC 插座和 MINI USB 接口，可由跳线进行设置，默认为 MINI USB 接口供电。

电源电路

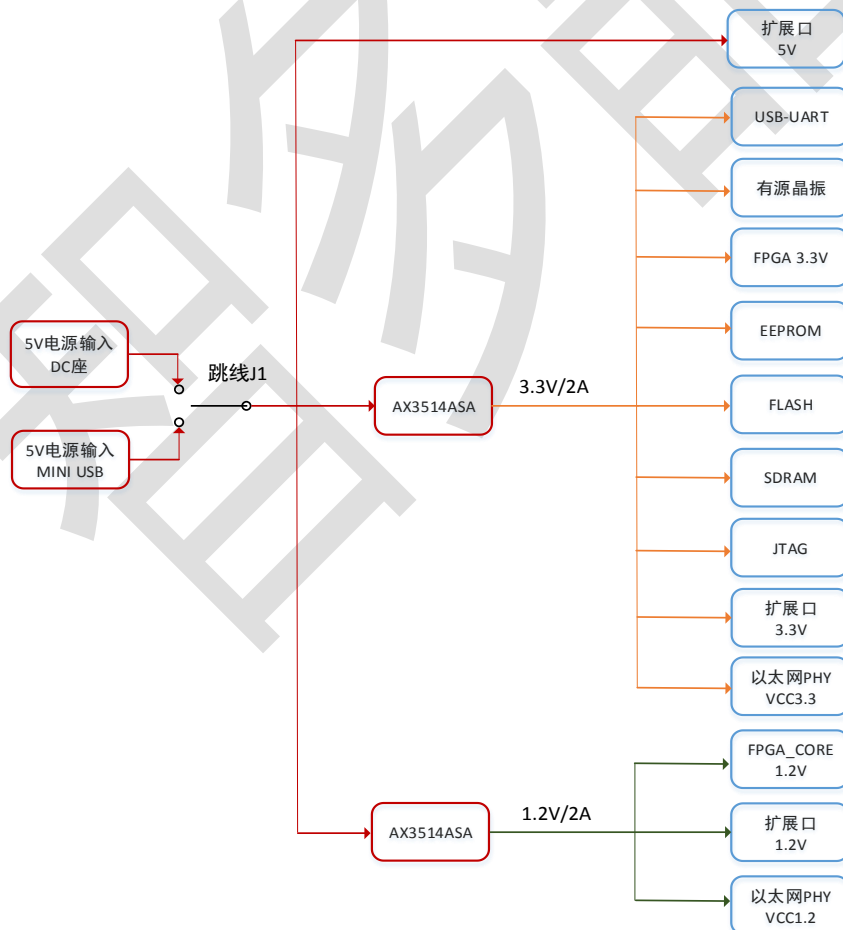


图 3-1 电源分配框图

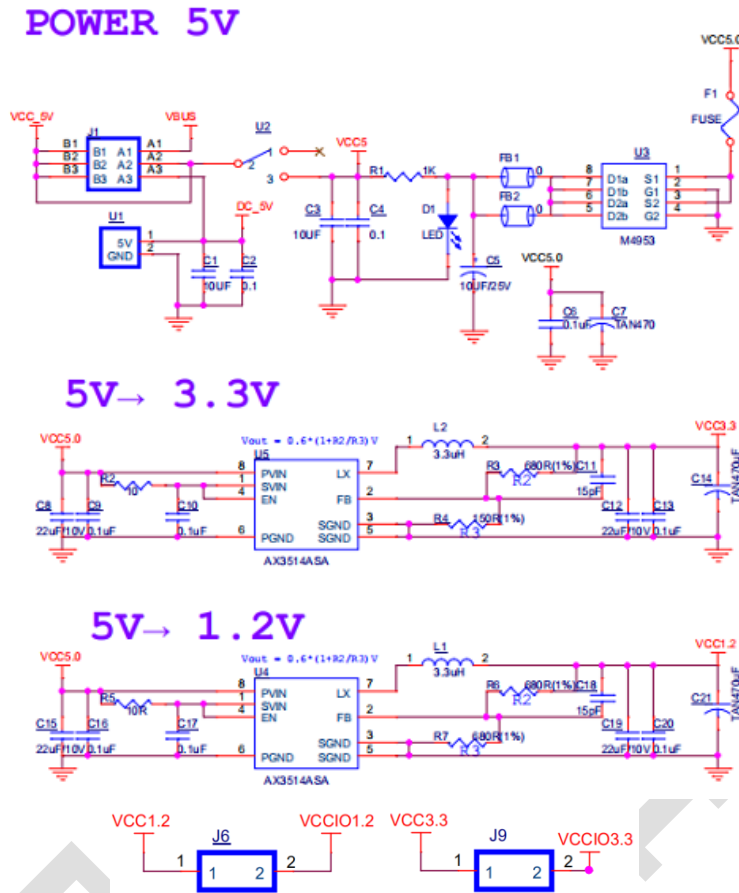


图 3-2 电源电路

跳线 J6 为 FPGA 核电压供电跳线，J9 为 FPGA IO 供电跳线，一般情况下应保持该跳线处于设置状态，否则会造成 FPGA 供电异常。+5V 电源的输入接口可通过跳线 J1 进行选择，如图 3-3 所示，1 处接上跳线帽选择 DC 座电源供电，2 处接上跳线帽选择 MINI USB 电源供电。

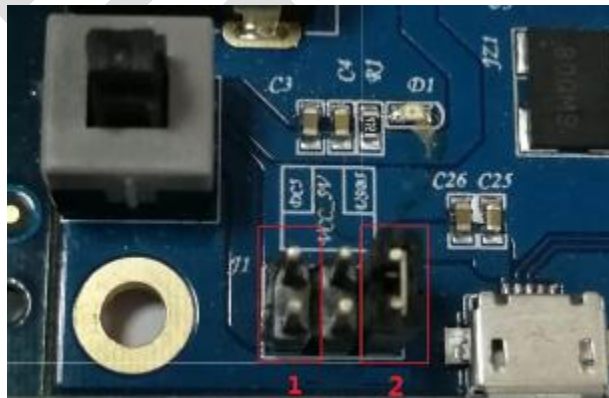


图 3-3 电源跳线设置方式

时钟资源

概述

开发板主时钟为 25MHz 有源晶振，为其 FPGA 和单片机提供时钟源。

时钟电路

Mainclk 25MHZ

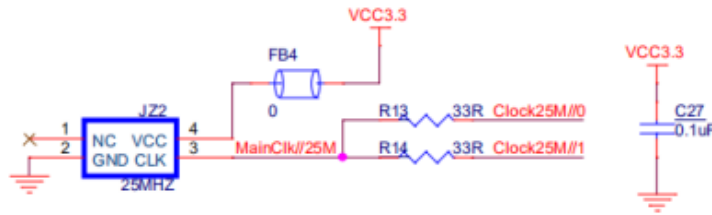


图 3-4 时钟电路

其中，Clock25M//0 为 FPGA 全局时钟输入，Clock25M//1 为单片机时钟源。

引脚分配

表 3-4 时钟引脚分配

信号名称	FPGA 引脚	描述
Clock25M//0	E1	FPGA 全局时钟输入
信号名称	单片机引脚	描述
Clock25M//1	5	单片机时钟

按键

概述

开发板具有 5 个按键 SW1~SW5，其中 SW5 为单片机复位按键，SW1~SW4 为 FPGA 用户按键。所有按键默认为 3.3V 上拉，按键按下去之后为低电平。

按键电路

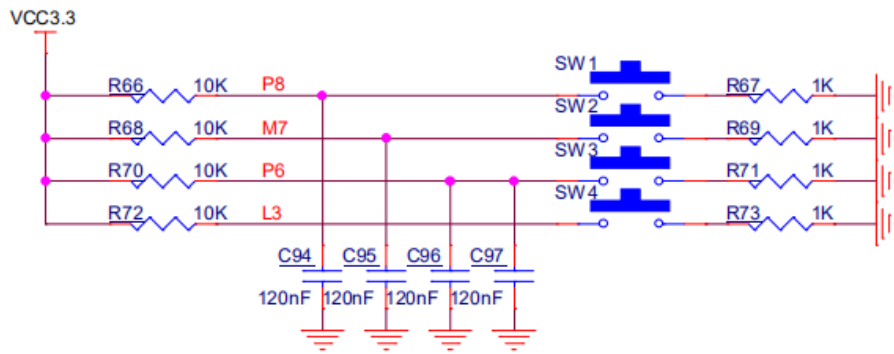


图 3-5 用户按键电路

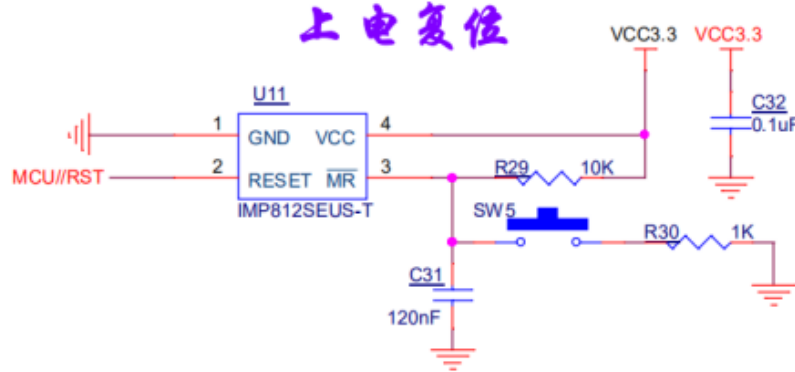


图 3-6 单片机复位电路

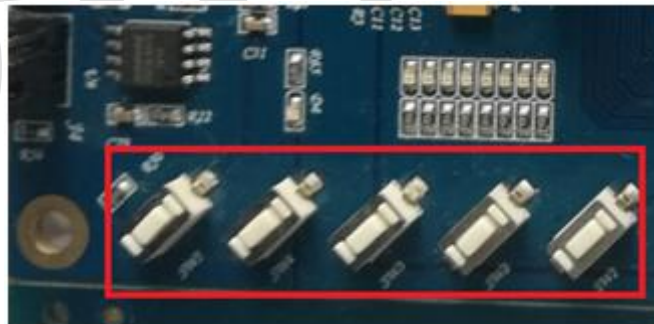


图 3-7 按键实物图

引脚分配

表 3-5 按键引脚分配

信号名称	FPGA 引脚	描述
SW1	P8	FPGA 用户按键
SW2	M7	FPGA 用户按键
SW3	P6	FPGA 用户按键
SW4	L3	FPGA 用户按键
信号名称	单片机引脚	描述
SW5	1	单片机复位按键

LED

概述

开发板具有 8 个用户 LED 与 FPGA 相连，分别为 D5~D12。LED 负极接地，正极与 FPGA 相应引脚连接。当 FPGA 相关引脚输出高电平时 LED 点亮，反之 LED 熄灭。

LED 电路

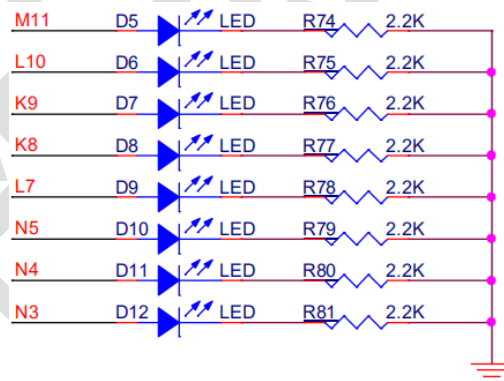


图 3-8 用户 LED 电路



图 3-9 用户 LED 实物图

引脚分配

表 3-6 LED 引脚分配

信号名称	FPGA 引脚	描述
D5	M11	FPGA 用户 LED
D6	L10	FPGA 用户 LED
D7	K9	FPGA 用户 LED
D8	K8	FPGA 用户 LED
D9	L7	FPGA 用户 LED
D10	N5	FPGA 用户 LED
D11	N4	FPGA 用户 LED
D12	N3	FPGA 用户 LED

扩展口

概述

开发板具有 2 个 40 PIN 的双排插针 JH1 和 JH2，以及 1 个 24 PIN 的双排插针 JH3，用于将 FPGA 的 GPIO 引出，且 JH1 和 JH2 扩展口均带有 +5V、+3.3V 和地，有利于用户灵活便捷的使用 GPIO 引脚。

扩展口电路

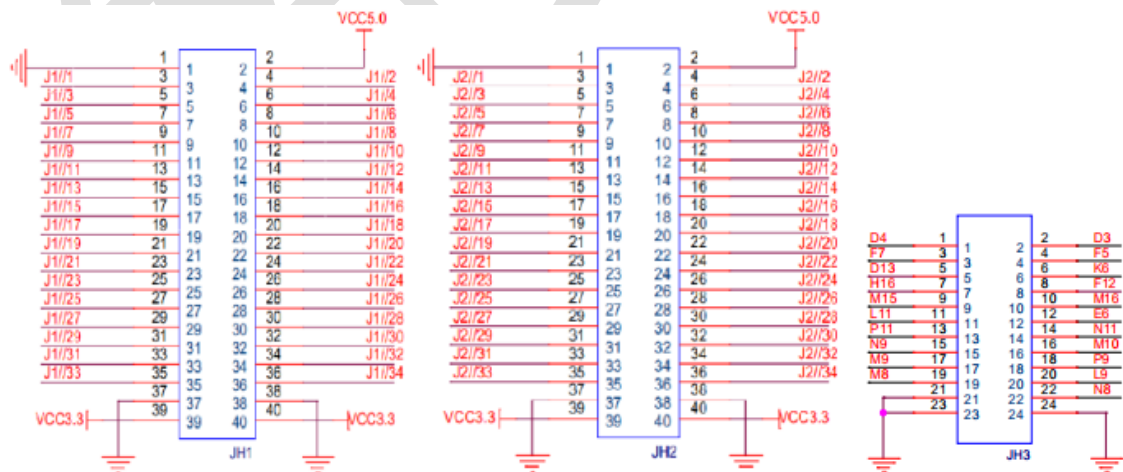


图 3-10 扩展口电路

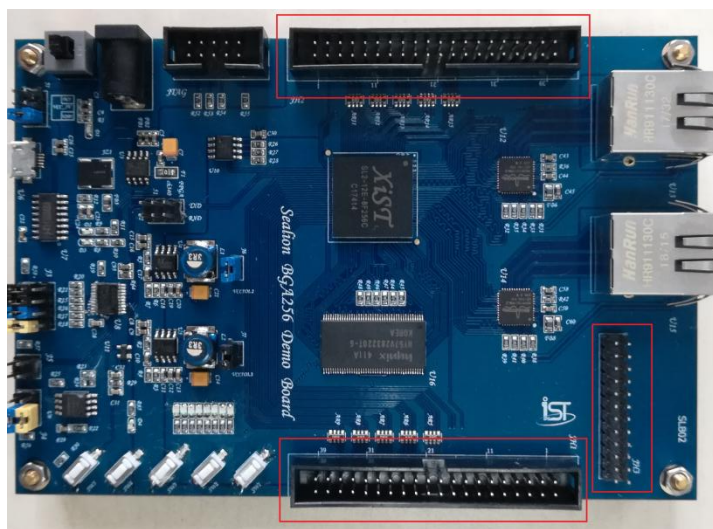


图 3-11 扩展口实物图

引脚分配

表 3-7 扩展口 JH1 引脚分配

引脚编号	FPGA 引脚	描述
1	GND	数字地
2	VCC5.0	+5V 电源
3	D6	GPIO
4	C6	GPIO
5	A5	GPIO
6	F6	GPIO
7	C8	GPIO
8	D8	GPIO
9	E7	GPIO
10	F8	GPIO
11	C9	GPIO
12	E8	GPIO

13	D9	GPIO
14	C11	GPIO
15	B14	GPIO
16	E9	GPIO
17	T14	GPIO
18	R13	GPIO
19	T13	GPIO
20	R12	GPIO
21	T12	GPIO
22	R11	GPIO
23	T11	GPIO
24	R10	GPIO
25	T10	GPIO
26	R9	GPIO
27	T9	GPIO
28	L8	GPIO
29	T8	GPIO
30	R8	GPIO
31	T7	GPIO
32	R7	GPIO
33	T6	GPIO
34	R6	GPIO
35	T5	GPIO
36	R5	GPIO
37	GND	数字地
38	GND	数字地
39	VCC3.0	+3V 电源
40	VCC3.0	+3V 电源

表 3-8 扩展口 JH2 引脚分配

引脚编号	FPGA 引脚	描述
1	GND	数字地
2	VCC5.0	+5V 电源
3	T2	GPIO
4	P3	GPIO
5	R1	GPIO
6	P2	GPIO
7	P1	GPIO
8	N2	GPIO
9	N1	GPIO
10	M2	GPIO
11	M1	GPIO
12	L2	GPIO
13	L1	GPIO
14	K2	GPIO
15	K1	GPIO
16	J2	GPIO
17	J1	GPIO
18	G1	GPIO
19	G2	GPIO
20	F1	GPIO
21	F2	GPIO
22	E2	GPIO
23	D1	GPIO
24	M6	GPIO
25	C2	GPIO
26	B1	GPIO

27	N6	GPIO
28	K5	GPIO
29	F3	GPIO
30	G5	GPIO
31	E5	GPIO
32	D5	GPIO
33	L4	GPIO
34	L6	GPIO
35	L5	GPIO
36	C3	GPIO
37	GND	数字地
38	GND	数字地
39	VCC3.0	+3V 电源
40	VCC3.0	+3V 电源

表 3-9 扩展口 JH3 引脚分配

引脚编号	FPGA 引脚	描述
1	D4	GPIO
2	D3	GPIO
3	F7	GPIO
4	F5	GPIO
5	D13	GPIO
6	K6	GPIO
7	H16	GPIO
8	F12	GPIO
9	M15	GPIO
10	M16	GPIO
11	L11	GPIO
12	E6	GPIO
13	P11	GPIO
14	N11	GPIO
15	N9	GPIO
16	M10	GPIO
17	M9	GPIO
18	P9	GPIO
19	M8	GPIO
20	L9	GPIO
21	GND	数字地
22	N8	GPIO
23	GND	数字地
24	GND	数字地

单片机

概述

开发板具有 1 个增强型 8051 单片机，型号为 STC12LE5604AD。该单片机主要实现 4 个功能。

- 与 USB-UART 电路通信，实现串口数据的收发
- 实现 FPGA 的 PS 配置功能
- 实现以太网 PHY 芯片的配置功能
- 实现 FLASH 的读写功能
-

单片机电路

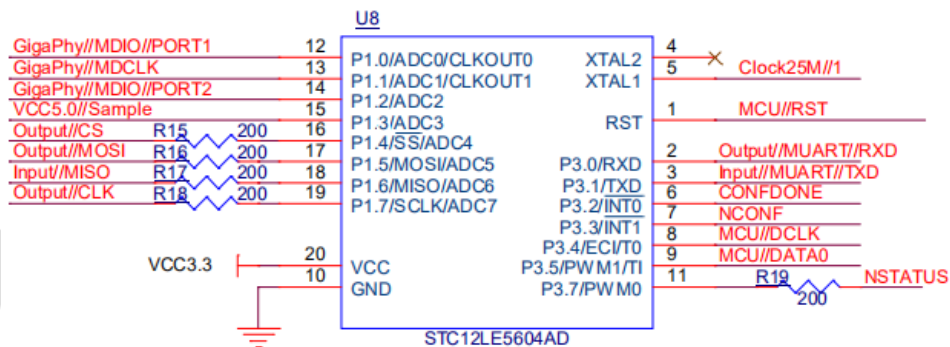


图 3-12 单片机电路

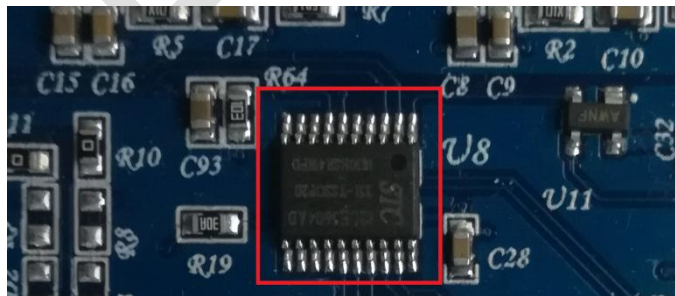


图 3-13 单片机实物图

引脚分配

表 3-10 单片机引脚分配

信号名称	单片机引脚	描述
MCU/RST	1	复位信号，低有效
Output//MUART//RXD	2	UART 接收端口
Input//MUART//TXD	3	UART 发送端口
Clock25M//1	5	单片机时钟输入
CONFDONE	6	FPGA PS 模式配置完成信号
NCONF	7	FPGA PS 模式配置启动信号
MCU//DCLK	8	FPGA PS 模式配置时钟信号
MCU//DATA0	9	FPGA PS 模式配置数据信号
NSTATUS	11	FPGA PS 模式配置标志信号
GigaPhy//MDIO//PORT1	12	以太网口 1 MDIO 配置数据信号
GigaPhy//MDCLK	13	以太网口 MDCLK 配置时钟信号
GigaPhy//MDIO//PORT2	14	以太网口 2 MDIO 配置数据信号
VCC5.0//Sample	15	5V 电源采样信号
Output//CS	16	FLASH 片选信号
Output//MOSI	17	FLASH MOSI 信号
Input//MISO	18	FLASH MISO 信号
Output//CLK	19	FLASH 时钟信号

千兆以太网接口

概述

开发板具有 2 个以太网 PHY 芯片，型号为 Broadcom 公司的 B50610M。该 PHY 芯片可实现 1000BASE-T/100BASE-TX/10BASE-T 三速收发器功能。千兆以太网接口的配置由单片机实现，数据收发与 FPGA 实现通信。

千兆以太网接口电路

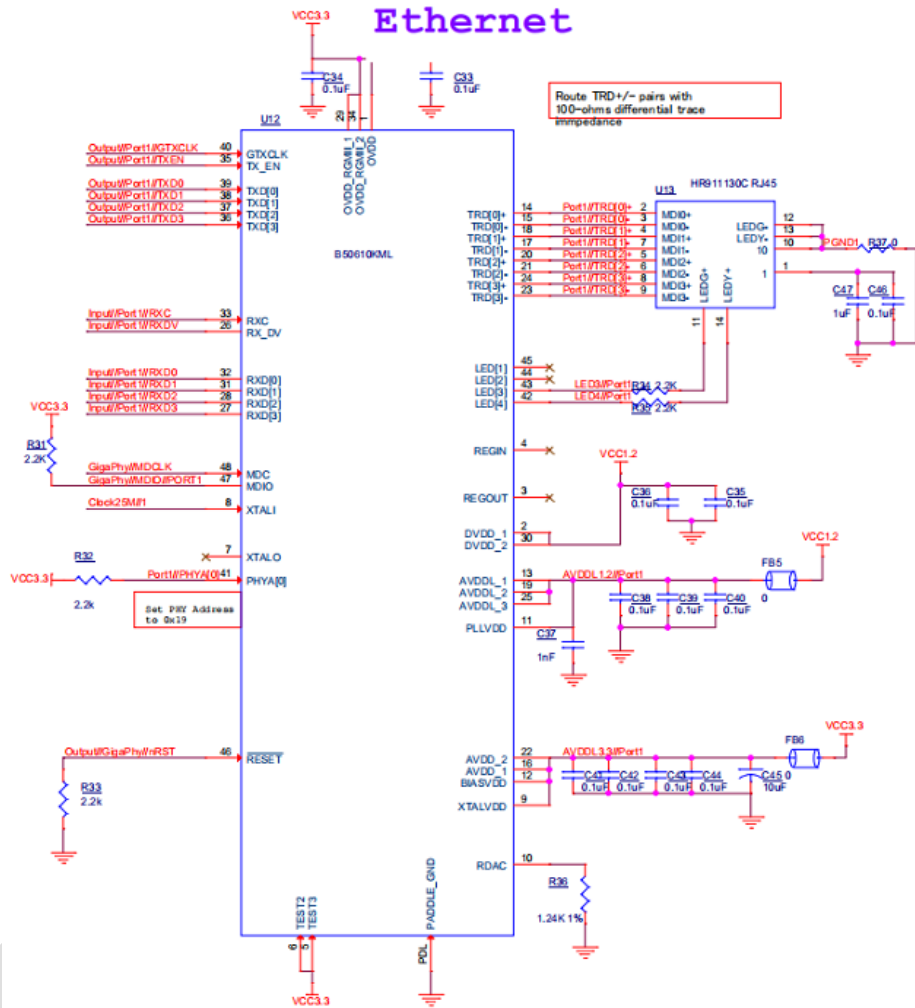


图 3-14 千兆以太网接口电路



图 3-15 千兆以太网接口实物图

引脚分配

表 3-11 千兆以太网接口引脚分配

信号名称	FPGA 引脚	描述
Output//Port1//GTCLK	B8	PORT1 RGMII 发送时钟
Output//Port1//TXEN	B4	PORT1 发送使能
Output//Port1//TXD0	A7	PORT1 发送数据
Output//Port1//TXD1	B7	PORT1 发送数据
Output//Port1//TXD2	A6	PORT1 发送数据
Output//Port1//TXD3	B6	PORT1 发送数据
Input//Port1//RXC	E15	PORT1 接收时钟
Input//Port1//RXDV	A2	PORT1 接收数据有效
Input//Port1//RXD0	A4	PORT1 接收数据
Input//Port1//RXD1	B3	PORT1 接收数据
Input//Port1//RXD2	A3	PORT1 接收数据
Input//Port1//RXD3	B5	PORT1 接收数据
Output//Port2//GTCLK	A14	PORT2 RGMII 发送时钟
Output//Port2//TXEN	A11	PORT2 发送使能
Output//Port2//TXD0	A13	PORT2 发送数据
Output//Port2//TXD1	B13	PORT2 发送数据
Output//Port2//TXD2	A12	PORT2 发送数据
Output//Port2//TXD3	B12	PORT2 发送数据
Input//Port2//RXC	E16	PORT2 接收时钟
Input//Port2//RXDV	B9	PORT2 接收数据有效
Input//Port2//RXD0	B11	PORT2 接收数据
Input//Port2//RXD1	A10	PORT2 接收数据
Input//Port2//RXD2	B10	PORT2 接收数据
Input//Port2//RXD3	A9	PORT2 接收数据
Output//GigaPhy//nRST	A8	以太网接口复位信号

SDRAM

概述

开发板具有 1 个 SDRAM 存储器件，型号为 hynix 公司的 HY57V283220T-6。该 SDRAM 存储容量为 4 Banks x 1M x 32Bit，最高时钟频率为 166MHz，支持 1,2,4,8 以及全页突发操作。

SDRAM 电路

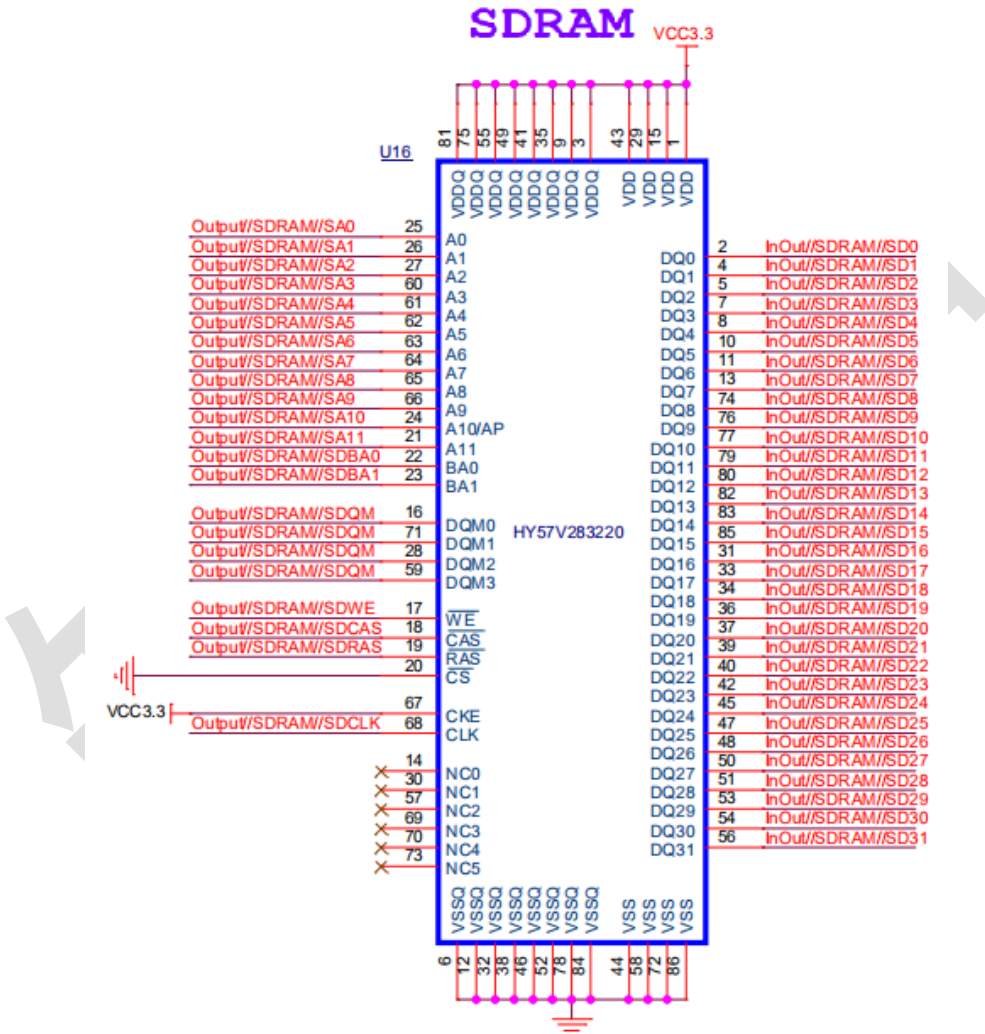


图 3-16 SDRAM 电路



图 3-17 SDRAM 实物图

引脚分配

表 3-12 SDRAM 引脚分配

信号名称	FPGA 引脚	描述
Output//SDRAM//SDCLK	F16	SDRAM 时钟
Output//SDRAM//SDWE	J14	SDRAM 写使能
Output//SDRAM//SDCAS	J16	SDRAM 列选通
Output//SDRAM//SDRAS	J15	SDRAM 行选通
Output//SDRAM//SA0	F9	SDRAM 地址
Output//SDRAM//SA1	K12	SDRAM 地址
Output//SDRAM//SA2	G11	SDRAM 地址
Output//SDRAM//SA3	L15	SDRAM 地址
Output//SDRAM//SA4	L16	SDRAM 地址
Output//SDRAM//SA5	K15	SDRAM 地址
Output//SDRAM//SA6	K16	SDRAM 地址
Output//SDRAM//SA7	J13	SDRAM 地址
Output//SDRAM//SA8	G16	SDRAM 地址
Output//SDRAM//SA9	G15	SDRAM 地址
Output//SDRAM//SA10	K11	SDRAM 地址
Output//SDRAM//SA11	J6	SDRAM 地址

Output//SDRAM//SDBA0	J12	SDRAM BANK
Output//SDRAM//SDBA1	J11	SDRAM BANK
Output//SDRAM//SDQM	F10	SDRAM DQM
InOut//SDRAM//SD0	D11	SDRAM 数据
InOut//SDRAM//SD1	D12	SDRAM 数据
InOut//SDRAM//SD2	D14	SDRAM 数据
InOut//SDRAM//SD3	E10	SDRAM 数据
InOut//SDRAM//SD4	E11	SDRAM 数据
InOut//SDRAM//SD5	F13	SDRAM 数据
InOut//SDRAM//SD6	F14	SDRAM 数据
InOut//SDRAM//SD7	F11	SDRAM 数据
InOut//SDRAM//SD8	F15	SDRAM 数据
InOut//SDRAM//SD9	D16	SDRAM 数据
InOut//SDRAM//SD10	D15	SDRAM 数据
InOut//SDRAM//SD11	C16	SDRAM 数据
InOut//SDRAM//SD12	C15	SDRAM 数据
InOut//SDRAM//SD13	B16	SDRAM 数据
InOut//SDRAM//SD14	C14	SDRAM 数据
InOut//SDRAM//SD15	A15	SDRAM 数据
InOut//SDRAM//SD16	K10	SDRAM 数据
InOut//SDRAM//SD17	L13	SDRAM 数据
InOut//SDRAM//SD18	L14	SDRAM 数据
InOut//SDRAM//SD19	L12	SDRAM 数据
InOut//SDRAM//SD20	M12	SDRAM 数据
InOut//SDRAM//SD21	N14	SDRAM 数据
InOut//SDRAM//SD22	N13	SDRAM 数据
InOut//SDRAM//SD23	N12	SDRAM 数据
InOut//SDRAM//SD24	R14	SDRAM 数据

InOut//SDRAM//SD25	T15	SDRAM 数据
InOut//SDRAM//SD26	P14	SDRAM 数据
InOut//SDRAM//SD27	R16	SDRAM 数据
InOut//SDRAM//SD28	P15	SDRAM 数据
InOut//SDRAM//SD29	P16	SDRAM 数据
InOut//SDRAM//SD30	N15	SDRAM 数据
InOut//SDRAM//SD31	N16	SDRAM 数据

USB-UART

概述

开发板具有型号为 CH340G 的 USB-UART 芯片，可通过板上的跳线 J2 进行设置，选择该芯片与 FPGA 连接还是与单片机连接。上位机可通过 MINI USB 接口实现与该串口的通信。同时对串口信号设置了两个 led 指示灯 D2 和 D3，用来对串口数据收发进行指示。

USB-UART 电路

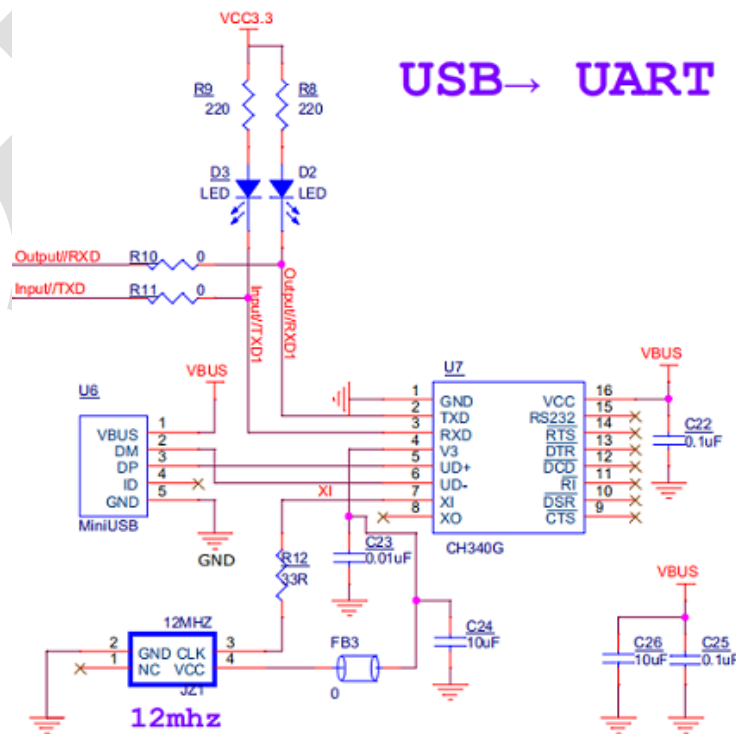


图 3-18 USB-UART 电路

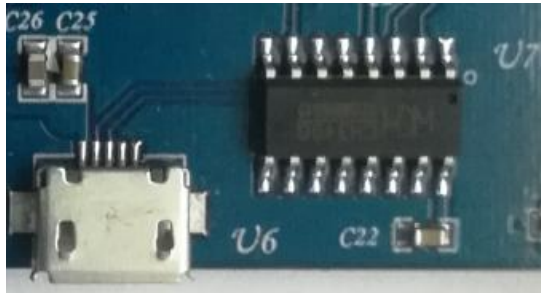


图 3-19 USB-UART 实物图

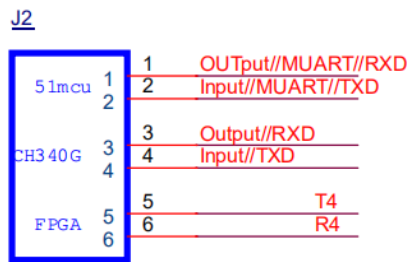


图 3-20 USB-UART 跳线电路

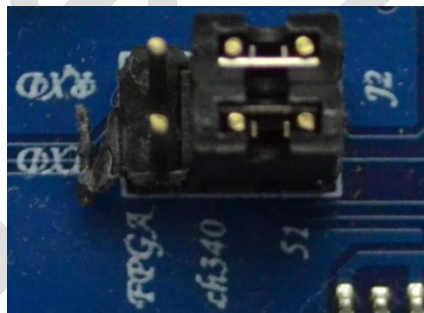


图 3-21 USB-UART 跳线实物图

USB-UART 跳线设置可参考图 3-20 及图 3-21。若要实现串口与 FPGA 通信，需要用跳线帽将 J2 的 PIN3 和 PIN5 连接，PIN4 和 PIN6 连接即可；若要实现串口与单片机通信，则需要用跳线帽将 J2 的 PIN3 和 PIN1 连接，PIN4 和 PIN2 连接即可。

引脚分配

表 3-13 USB-UART 引脚分配

信号名称	FPGA 引脚	描述
RXD	T4	FPGA 串口接收端
TXD	R4	FPGA 串口发送端
信号名称	单片机引脚	描述
Output//MUART//RXD	2	单片机串口接收端
Input//MUART//TXD	3	单片机串口发送端

FLASH

概述

开发板具有 1 个 FLASH 芯片，型号为 Winbond 公司的 W25Q16。该 FLASH 芯片存储容量为 16M-bit，具有 SPI 接口。板上 FPGA 或单片机均可对该 FLASH 进行读写操作，可通过板上的跳线 J3 进行设置，选择该芯片与 FPGA 连接还是与单片机连接。该 FLASH 在板上主要作为 FPGA 的配置器件，用于存储 FPGA 的配置比特流文件，结合 FPGA 的配置端口，可实现 FPGA 的 AS 及 PS 配置模式。

FLASH 电路

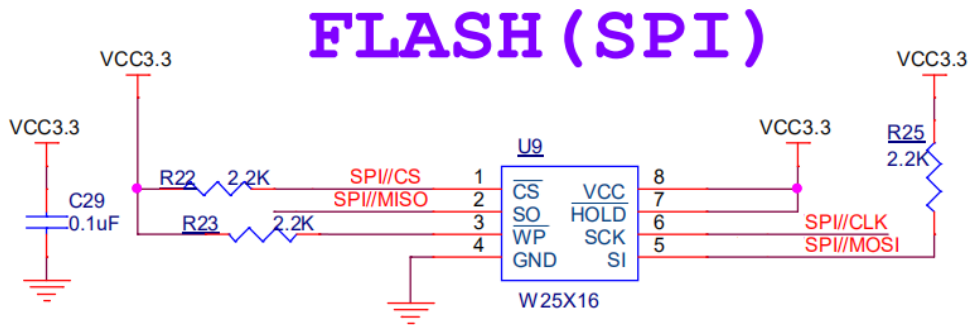


图 3-22 FLASH 电路

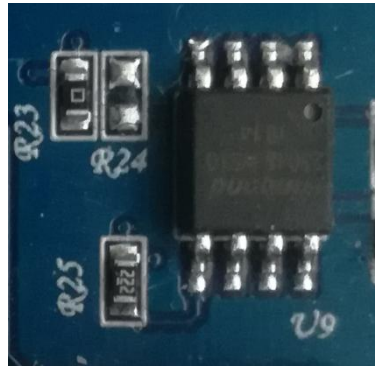


图 3-23 FLASH 实物图

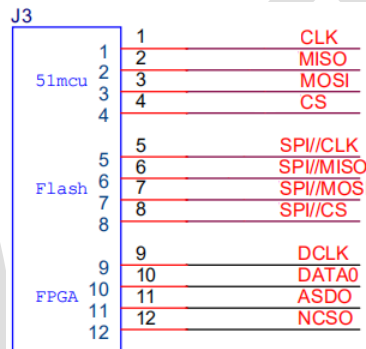


图 3-24 FLASH 跳线电路



图 3-25 FLASH 跳线实物图

FLASH 跳线设置可参考图 3-24 及图 3-25。若要实现 FPGA 读写 FLASH，需要用跳线帽将 J3 的 PIN5 和 PIN9 连接，PIN6 和 PIN10 连接，PIN7 和 PIN11 连接，PIN8 和 PIN12 连接即可；若要实现单片机读写 FLASH，则需要用跳线帽将 J3 的 PIN5 和 PIN1 连接，PIN6 和 PIN2 连接，PIN7 和 PIN3 连接，PIN8 和 PIN4 连接即可。

引脚分配

表 3-14 FLASH 引脚分配

信号名称	FPGA 引脚	描述
NCSO	D2	FPAG AS 模式片选信号
ASDO	C1	FPAG AS 模式 ASDO 信号
DATA0	H2	FPAG AS 模式 DATA0 信号
DCLK	H1	FPAG AS 模式时钟信号
信号名称	单片机引脚	描述
Output//CS	16	FLASH 片选信号
Output//MOSI	17	FLASH MOSI 信号
Input//MISO	18	FLASH MISO 信号
Output//CLK	19	FLASH 时钟信号

EEPROM

概述

开发板具有一片 EEPROM，型号为 24LC04，容量为：4Kbit（2*256*8bit），由 2 个 256byte 的 block 组成，通过 IIC 总线与 FPGA 进行通信。

EEPROM 电路

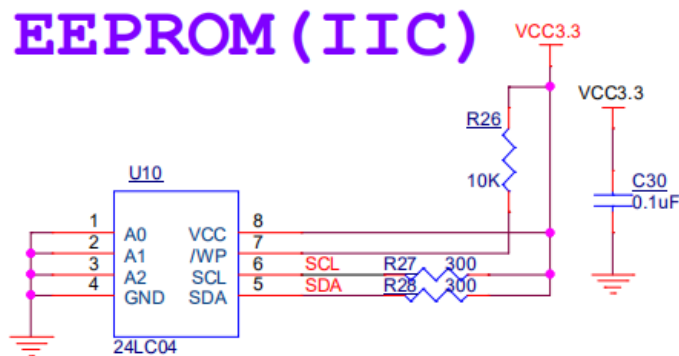
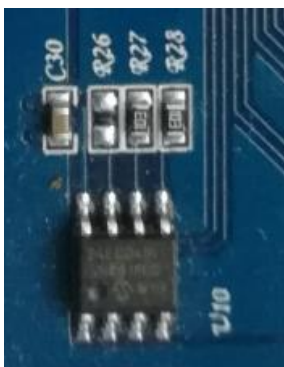


图 3-26 EEPROM 电路


图 3-27 EEPROM 实物图

引脚分配

表 3-15 EEPROM 引脚分配

信号名称	FPGA 引脚	描述
SCL	T3	IIC 总线时钟信号
SDA	R3	IIC 总线数据信号

配置模式

概述

SL2-12K FPGA 支持 JTAG、AS 以及 PS 多种配置模式，不同模式可通过与配置相关的跳线 J3、J4、J5 进行设置。XIST 提供自主研发的下载器及下载软件，将下载器连接至开发板的 JTAG 端口，通过下载软件即可实现 JTAG 模式和 AS 模式下载。

JTAG 模式

JTAG 是最常使用的下载模式，可通过 JTAG 口将 .bit 文件下载到 FPGA 中，断电后信息会丢失需要重新下载，每次下载前需要对 FPGA 重新上电。

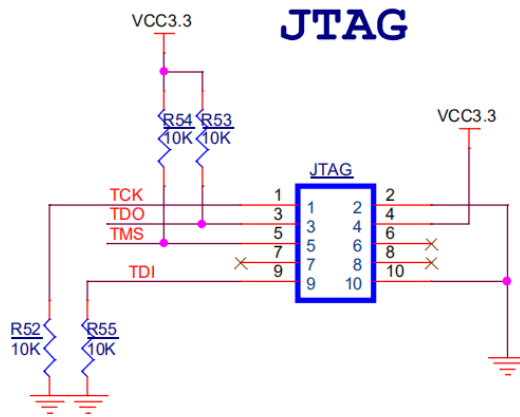


图 3-28 JTAG 接口电路



图 3-29 JTAG 插座实物图

AS 模式

主动模式时，FPGA 上电会主动读取外置串行 FLASH。配置完成后会释放出 CONF_DONE 引脚，它由一个上拉电阻拉高。

使用 AS 模式时，可通过 JTAG 口将.bin 文件固化到 FLASH 中，重新上电后 AS 模式可实现从 FLASH 到 FPGA 比特流加载。

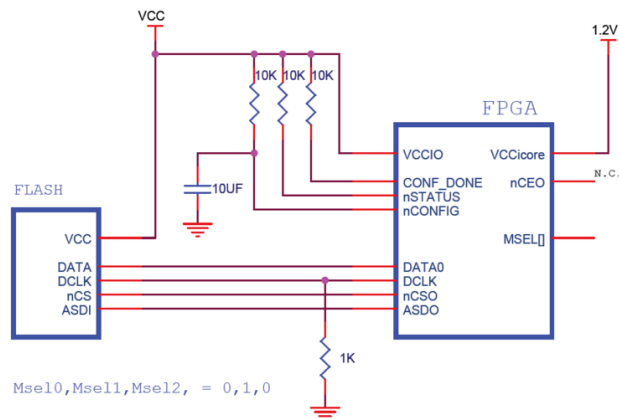


图 3-30 AS 配置模式电路

跳线设置

AS 模式或 PS 模式的选择，可通过跳线 J3、J4、J5 进行设置。

- AS 模式的设置：

- 1) 将 J4 的 MSEL1 引脚连接至 VCC3.3；
- 2) 将 J3 设置为 FLASH 与 FPGA 相连；
- 3) 将 J5 跳线悬空。

- PS 模式的设置：

- 1) 将 J4 的 MSEL1 引脚接地；
- 2) 将 J3 设置为 FLASH 与单片机相连；
- 3) 将 J5 跳线 MCU//DCLK 连接至 DCLK，MCU//DATA0 连接至 DATA0。

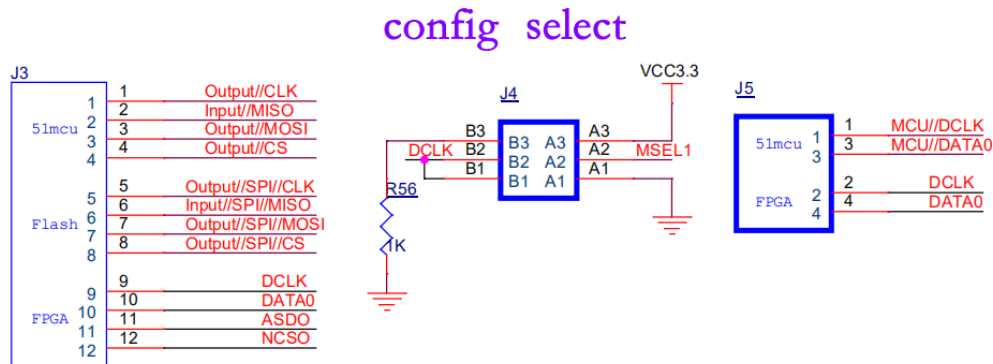


图 3-31 配置模式跳线电路

4. 软件系统

关于本章节

本章节内容包括 HqFpga 常用操作和注意事项，并通过新建工程说明软件的操作步骤，旨在引导客户快速熟悉软件环境并避免常见的错误。关于对本软件更为详尽的说明，请参见我公司《HqFpga 软件开发系统用户手册》。

HqFpga 简介

HqFpga 是西安智多晶微电子有限公司（XIST）开发的 FPGA 开发系统，在单一的可执行文件中集成了从综合、布局、布线等所有 FPGA 实现功能，在同一可执行环境下所集成各项功能可以共享所有的信息，

例如统一的数据模型、时序分析结果等，HqFpga 还提供一键式(Push-Button)的运行模式，并提供多种界面语言，方便用户完成简单、快速、高效及优化的 FPGA 开发。

HqFpga 软件图形化界面环境如图 4-1 所示，左侧列的工具栏包含了软件常用的操作和设置，如新建工程、设计管理、IP 管理、物理和时序约束以及设置等。界面中间 3 个图标对应了设计实现的 3 个阶段：综合、布局布线和比特流生成。界面风格简洁友好。

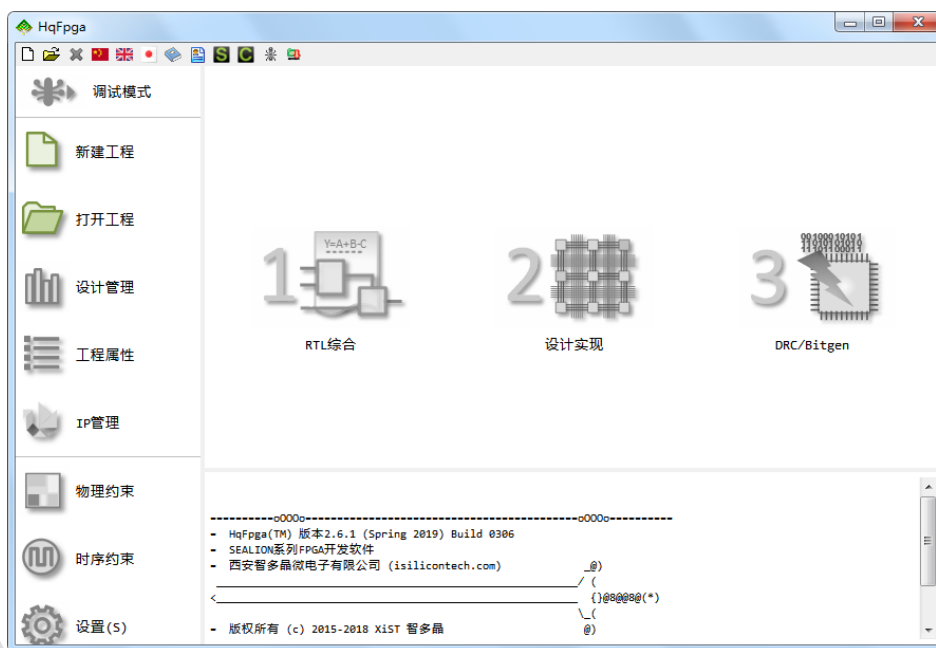


图 4-1 HqFpga 图形化界面

软件安装

HqFpga 安装

HqFpga 软件为纯绿色免安装版本，通常以压缩文件夹的方式提供，包括 WINDOWS 和 LINUX 平台下的 32 位及 64 位版本。软件安装简单便捷，解压后即可使用。请注意解压后的文件夹应放在非中文目录下，否则在软件使用过程中会出现各种异常错误，用户新建工程也应为非中文目录。

解压后的安装包包含 3 个一级子目录：**build**、**doc**、**samples**，其中 **build** 目录为软件系统的主体，**doc** 目录下存放了《HqFpga 软件开发系统用户手册》的中英文文档，**samples** 目录中提供了 2 个工程例程。以 WINDOWS 32 位平台版本为例，在 <HqFpga 安装目

录>\build\win_x86\hqui 目录下可以找到 hqui.exe 可执行文件，该文件为 HqFpga 的启动文件。为了使用方便，建议创建该文件的快捷方式，后面通过双击快捷方式即可进入 HqFpga 软件图形化界面环境。

下载器驱动安装

HqFpga 软件图形化界面环境已集成了下载器软件工具界面，但是该下载器在使用之前还需要进行驱动。下载器的驱动程序我们会以压缩文件的方式提供给客户，解压后参照以下步骤进行安装。

- 1) 将 XSKIT-CABLE-USB 下载器连接到计算机；
- 2) 打开计算机设备管理器，找到带 ? 的其他设备中的“Xist USB Cable”；

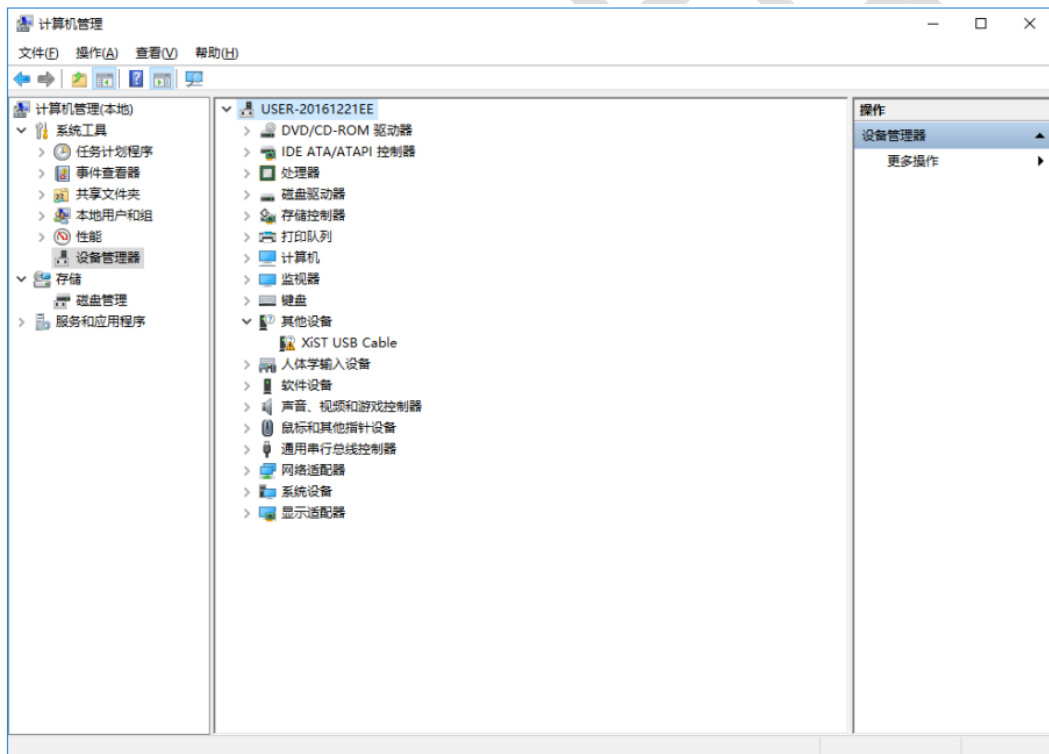


图 4-2 下载器驱动安装界面 1

- 3) 单击右键，选择更新驱动程序（手动查找并安装），选择安装路径下 XIST_USB_Cable 目录下的 Driver 文件夹，点击下一步；

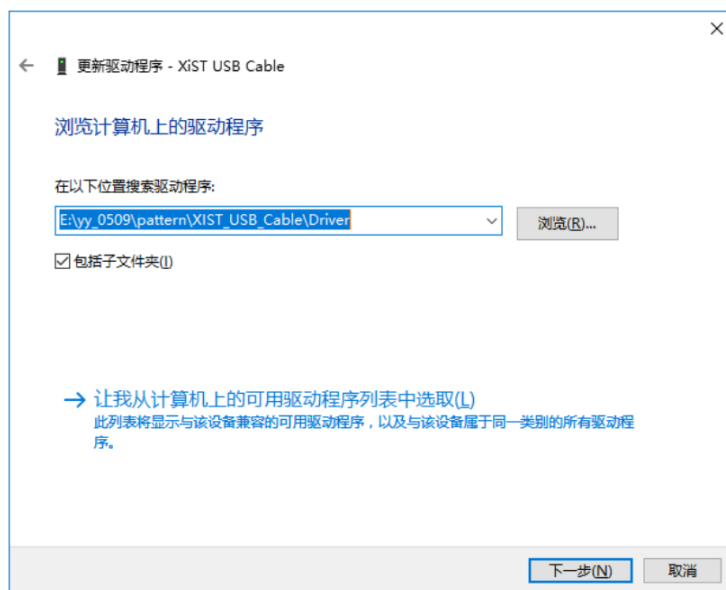


图 4-3 下载器驱动安装界面 2

- 4) 点击“始终安装此驱动程序软件”；
注：如果安装失败，关闭计算机驱动程序签名后重新
- 5) 驱动更新完成，点击“关闭”。

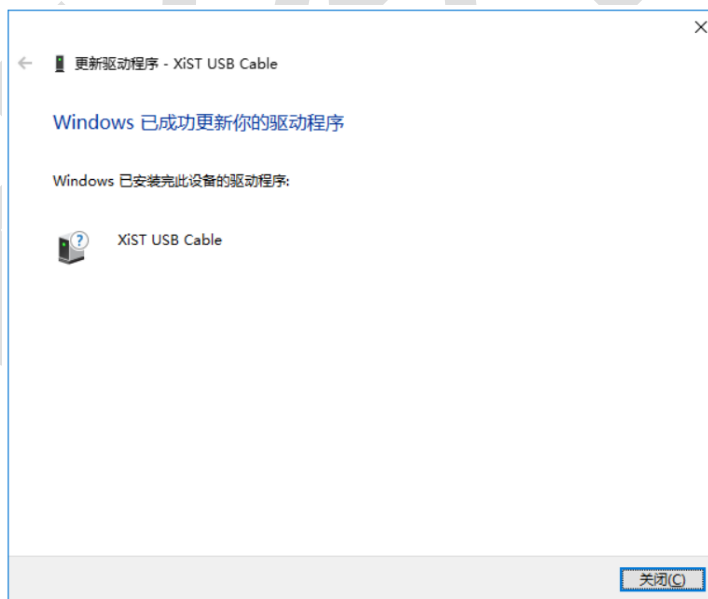


图 4-4 下载器驱动安装完成界面

例程说明

本节提供一个简单的使用例程，通过该例程引导用户熟悉软件界面的使用，达到快速上手的目的。同时，该例程也对开发板上的 4 个用户按键及 8 个 LED 进行了测试，下载运行后效果直观。

代码说明

该例程代码功能为通过 4 个用户按键控制 8 个 LED 以不同方式点亮，SW1 按下后为正向流水灯，SW2 按下后为逆向流水灯，SW3 按下后为流水灯从中间开始依次向两侧点亮，之后又由两侧向中间依次亮起，SW4 按下后 8 个 LED 分两组变为呼吸灯式逐渐点亮，之后逐渐熄灭。具体代码如下：

```
module led_4key_test(
    key_in,
    clk,
    led_out
);
input[3:0] key_in;
input clk;
output[7:0] led_out;

reg[7:0] led_out = 8'd0;
reg[26:0] cnt = 'd0;
reg[2:0] cnt2=0;
reg[2:0] cnt3=0;
//key_in fliter
reg[17:0] cnt_fliter0 = 'd0;
reg[17:0] cnt_fliter1 = 'd0;
reg[17:0] cnt_fliter2 = 'd0;
reg[17:0] cnt_fliter3 = 'd0;
reg[1:0] key_sel = 'd0;
reg [26:0] counter = 'd0;
reg [5:0] PWM_adj = 'd0;
```

```
reg [6:0] PWM_width = 'd0;
reg [24:0] cnt_shift = 'd0;

//key_in filter
//*****//
always@(posedge clk)
begin
    if(key_in[0]==1'b0)
        begin
            if(cnt_fliter0>='d250000)
                cnt_fliter0 <= 'd0;
            else
                cnt_fliter0 <= cnt_fliter0 + 1'd1;
            end
        else
            cnt_fliter0 <= 'd0;
        end
    always@(posedge clk)
    begin
        if(key_in[1]==1'b0)
            begin
                if(cnt_fliter1>='d250000)
                    cnt_fliter1 <= 'd0;
                else
                    cnt_fliter1 <= cnt_fliter1 + 1'd1;
                end
            else
                cnt_fliter1 <= 'd0;
            end
        end
    always@(posedge clk)
```

```
begin
  if(key_in[2]==1'b0)
    begin
      if(cnt_fliter2>='d250000)
        cnt_fliter2 <= 'd0;
      else
        cnt_fliter2 <= cnt_fliter2 + 1'd1;
      end
    else
      cnt_fliter2 <= 'd0;
    end

always@(posedge clk)
begin
  if(key_in[3]==1'b0)
    begin
      if(cnt_fliter3>='d250000)
        cnt_fliter3 <= 'd0;
      else
        cnt_fliter3 <= cnt_fliter3 + 1'd1;
      end
    else
      cnt_fliter3 <= 'd0;
    end

end

always@(posedge clk)
begin
  if(cnt_fliter0 == 'd249999)
    key_sel <= 'd0;
  else if(cnt_fliter1 == 'd249999)
    key_sel <= 'd1;
  else if(cnt_fliter2 == 'd249999)
```

```
        key_sel <= 'd2;
    else if(cnt_fliter3 == 'd249999)
        key_sel <= 'd3;
    else
        key_sel <= key_sel;
end
//*****//

//led loop
//*****//
always@(posedge clk)
begin
    if(cnt >= 'd12500000) //0.5s
        cnt <= 'd0;
    else
        cnt <= cnt + 1'd1;
end

always@(posedge clk)
begin
    if(cnt == 'd12499997)
        cnt2 <= cnt2 + 'd1;
    else
        cnt2 <= cnt2 ;
end

always@(posedge clk)
begin
    if(cnt == 'd12499997)
        cnt3 <= cnt3 - 'd1;
    else
        cnt3 <= cnt3 ;
end
```

```
end
//*****//

//led_pwm
//*****//

always @(posedge clk)
begin
    counter <= counter+1;
    PWM_width <= PWM_width[5:0]+ PWM_adj;
    if(counter[26])
        PWM_adj <= counter[25:20];
    else
        PWM_adj <= ~ counter[25:20];
end
//*****//

//led_out
//*****//

always@(posedge clk)
begin
    case(key_sel)
    'd0:begin
        if(cnt == 'd1249998)
            led_out <= 8'b0;
        else if(cnt == 'd1249999)
            led_out[cnt2] <= 1'b1;
        else
            led_out <= led_out ;
        end
    'd1:begin
        if(cnt == 'd1249998)
            led_out <= 8'b0;
```



```
        else if(cnt == 'd12499999)
            led_out[cnt3] <= 1'b1;
        else
            led_out <= led_out ;
    end
'd2:begin
    if(cnt == 'd12499998)
        led_out <= 8'b0;
    else if(cnt == 'd12499999)
        begin
            led_out[cnt2] <= 1'b1;
            led_out[7-cnt2] <= 1'b1;
        end
    else
        led_out <= led_out ;
    end
'd3:
    begin
        led_out[0] <= PWM_width[6];
        led_out[1] <= PWM_width[6];
        led_out[2] <= PWM_width[6];
        led_out[3] <= PWM_width[6];
        led_out[4] <=~PWM_width[6];
        led_out[5] <=~PWM_width[6];
        led_out[6] <=~PWM_width[6];
        led_out[7] <=~PWM_width[6];
    end
default:
    led_out <= 8'h0;
endcase
end
//*****//
```

endmodule

输入引脚为 4 个按键输入 key_in[3:0]，时钟输入 clk 为板上 25MHz 晶振；输出为 8 个 LED led_out[7:0]。

首先对 4 个按键输入进行消抖处理，滤波的脉宽为 10ms，即按键按下超过 10ms 后检测到有效。之后用锁存器的方式对 4 个按键状态进行锁存，在后面用于控制 LED 已哪种模式输出。之后是 4 种 LED 点亮方式的电路，其中，前 3 种方式均属于流水灯，只需要用计数器对 clk 先进行分频（代码中为 0.5s 点亮一次），之后控制 LED 的下标已不同方式动作即可。第 4 种为呼吸灯方式，是采用脉宽计数器控制输出脉宽的占空比，以达到对 LED 等亮度的控制目的。

创建工程

- 1) 双击 HqFpga 快捷启动图标，进入软件界面，并点击左侧“新建工程按钮”，出现图 4-5 界面，在界面中选择如图所示目标器件，并创建工程名称，点击“下一步”；

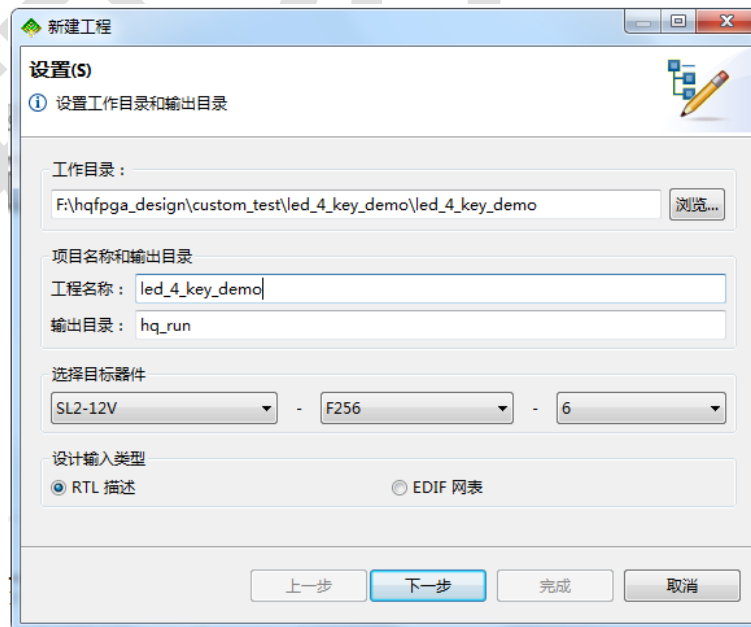


图 4-5 工程创建界面 1

2) 在图 4-6 界面中添加我们提供的 verilog 源文件，点击“完成”；

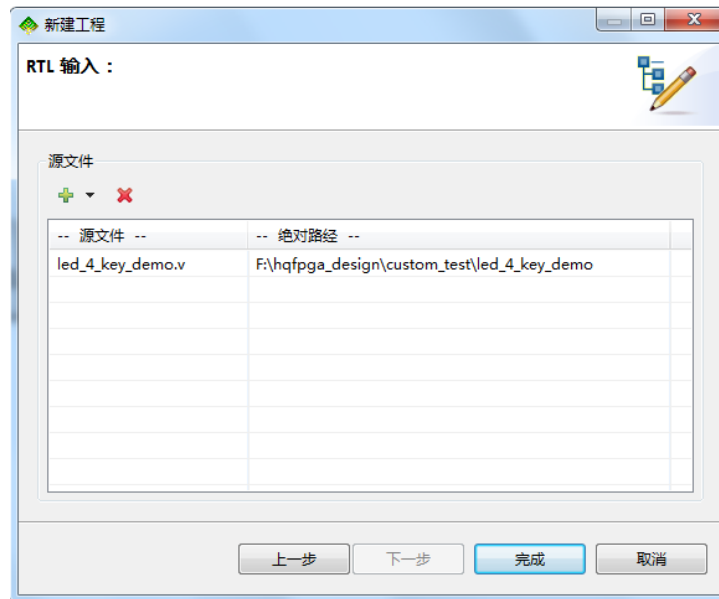


图 4-6 工程创建界面 2

- 3) 点击“RTL 综合”，完成电路综合步骤；
- 4) 点击左侧“物理约束”，在出现的 4-7 界面中选择“导入约束文件”，添加我们提供的 io.upc 引脚约束文件，点击“确定”；

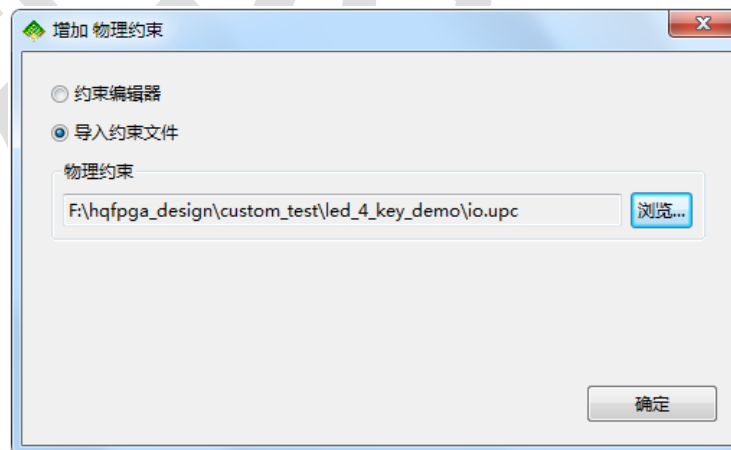


图 4-7 物理约束导入界面

- 5) 点击“设计实现”，完成布局布线；
- 6) 点击“DRC/Bitgen”，完成比特流生成。

下载及固化

通过下载器可实现.bit 文件下载（掉电丢失）或者将程序固化进板上的 FLASH，实现程序的非易失性加载。.bit 文件的下载可参考以下步骤。

- 1) 点击图 4-8 红色框部分“下载/编程”启动图标，软件会调用集成的下载器软件界面；



图 4-8 下载启动图标

- 2) 下载器软件界面如图 4-9 所示，在红色框部分已自动加载了刚编译生成的.bit 文件，用户可直接点击“下载”，在下载前需要对开发板重新上电；

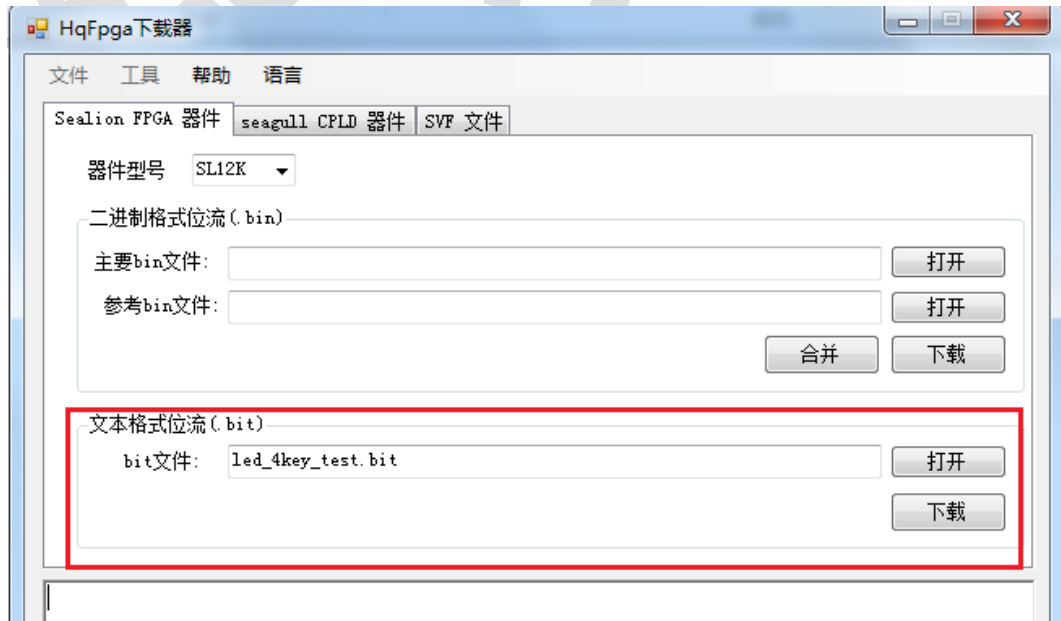


图 4-9 下载器软件界面

- 3) 在下载过程中，下载器的提示界面中会出现 MCU 版本号、Device ID、Verify 等信息，当出现 Verify successfully 提示后，说明下载完成。此时，我们观察开发板，8 个 LED 已开始正向流水电亮。点击 4 个按键，均为我们设计的功能。

用户还可通过下载器实现程序的固化。固化程序需要将配置模式设置为 AS 模式，之后还需要正确生成.bin 文件。程序固化可参考以下步骤。

- 1) 在 HqFpga 界面左侧点击“设置”，在出现的界面中选择“位流生成”子菜单，并勾选“生成二进制文件格式”及“压缩比特流”选项；

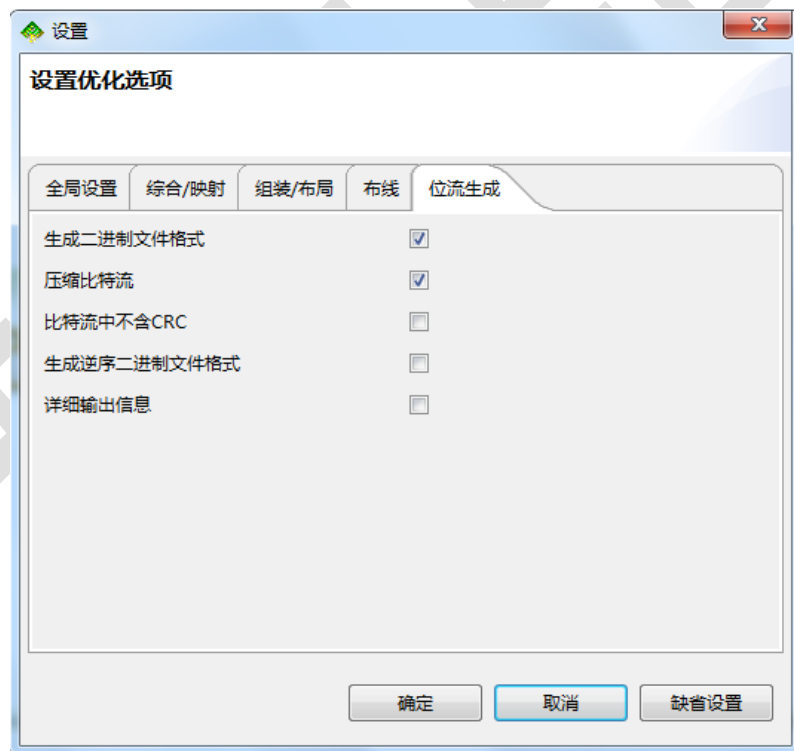


图 4-10 位流生成设置界面

- 2) 点击“DRC/Bitgen”，重新生成比特流，并点击下载软件启动图标，启动下载软件界面；
- 3) 在红色框部分已自动加载了刚编译生成的.bin 文件，用户可直接点击“下载”将 bin 文件烧录进 FLASH。

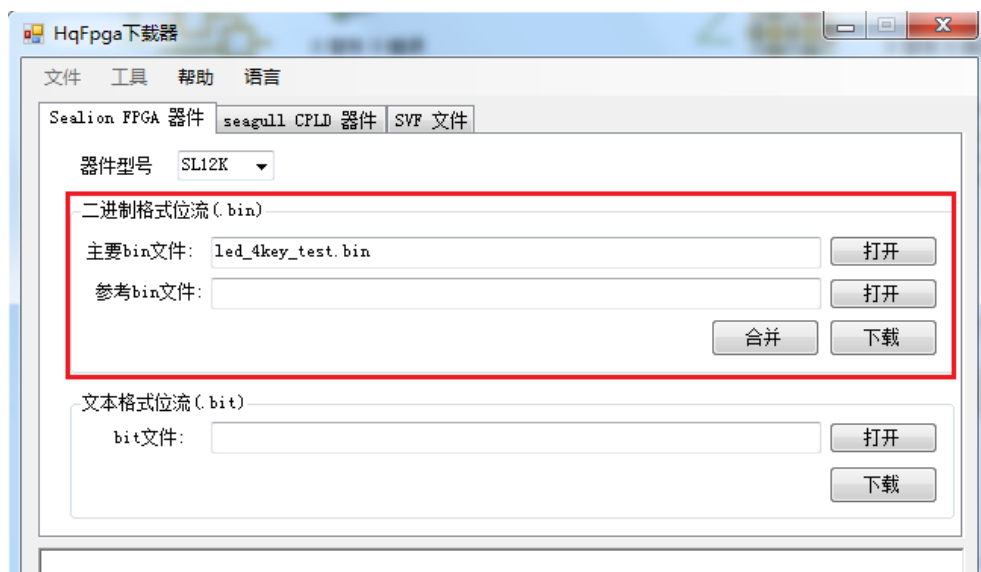


图 4-11 bin 文件下载界面

其他注意事项

- 软件安装目录及工程创建目录不应有中文目录
- 程序下载之前需要对开发板重新上电
- 不要重复多次打开软件下载界面，当下载完成后需要先关闭下载界面才能进行其他软件操作
- 在 HqFpga 软件环境中左侧的“设计管理”中显示了源代码的树状层次化结构，并可编辑源代码
- 引脚约束.upc 可参考如下语法

```

phycst.pin.set {clk} E1
phycst.pin.set {key_in[0]} P8

```
- 时序约束.sdc 可参考如下语法，该语法约束了 PLL 的时钟输入端及所有时钟输出端频率

```

create_clock -period 40 [get_ports ex_clk]
derive_generated_clocks

```