



特权 VIP FPGA 图像视频套件 开发指南

欢迎加入 FPGA/CPLD 助学小组一同学习交流:

EDN:

http://group.ednchina.com/GROUP_GRO_14596_1375.HTM

ChinaAET: <http://group.chinaaet.com/273>

淘宝店链接: <http://myfpga.taobao.com/>

技术咨询: wuhouhang@gmail.com

特权 VIP 最新资料例程下载地址: <http://pan.baidu.com/s/1hqJx3tM>

版本信息		
时间	版本	状态
2014-06-14	V1.00	创建。
2014-06-17	V1.01	新增 2.3、2.4。
2014-06-22	V1.02	新增 2.5、2.6、2.7。
2014-07-22	V1.03	新增 2.8。
2014-07-30	V1.04	修改 2.8, 新增 2.9。
2014-09-16	V1.05	新增 2.10。
2014-10-25	V2.00	新增 2.11、2.12、2.13 节。
2014-11-01	V2.01	新增 2.14 节。





目录

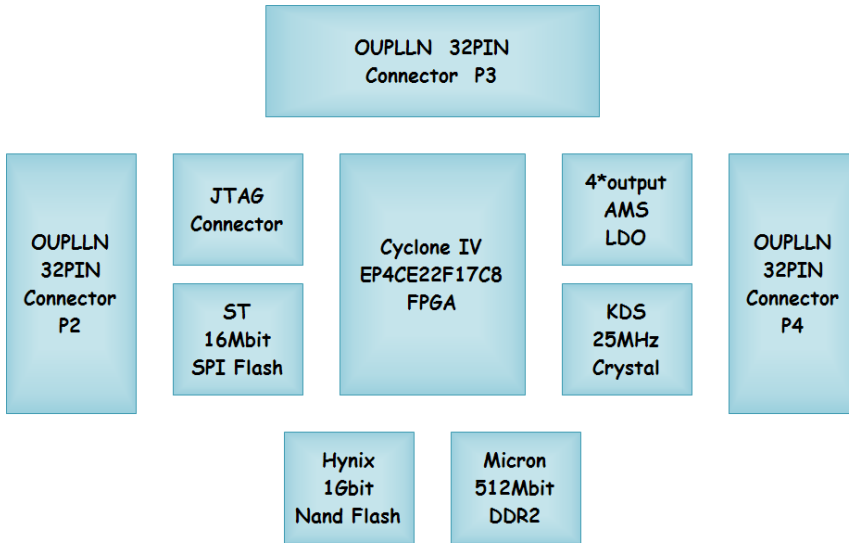
特权 VIP FPGA 图像视频套件 开发指南	1
1 套件使用说明.....	3
1.1 功能框图.....	3
1.2 硬件电路说明	4
1.3 引脚定义.....	11
2 开发例程详解.....	14
2.1 特权 VIP 例程 01--PLL 例化和 LED 闪烁.....	14
2.2 特权 VIP 例程 02--DDR2 控制器读写测试.....	16
2.3 特权 VIP 例程 03--Qsys 系统与软件实例.....	21
2.4 特权 VIP 例程 04--NAND FLASH 读写测试.....	26
2.5 特权 VIP 例程 05--VGA /SVGA /720p /1080p 多分辨率显示驱动.....	31
2.6 特权 VIP 例程 06--7 寸 800*480 LCD 显示驱动.....	35
2.7 特权 VIP 例程 07--触摸屏驱动	39
2.8 特权 VIP 例程 08--电子点菜单	43
2.9 特权 VIP 例程 09--200W 像素 CMOS 摄像头采集显示	56
2.10 特权 VIP 例程 10--AV(PAL/NTSC)视频采集显示	64
2.11 特权 VIP 例程 11--FX2 硬件和驱动安装以及 Bulkloop 实验.....	73
2.12 特权 VIP 例程 12--基于 FX2 和 FPGA 的 Loopback/StreamIN/StreamOUT 实例.....	81
2.13 特权 VIP 例程 13--基于 VGA 驱动的 720p 广告机实例.....	94
2.14 特权 VIP 例程 14--TXT 文本阅读器	104
附录 A 实例与工程映射	111
附录 B 套件淘宝购买链接	113



1 套件使用说明

1.1 功能框图

VIP 核心板的整板硬件功能框图如图所示。



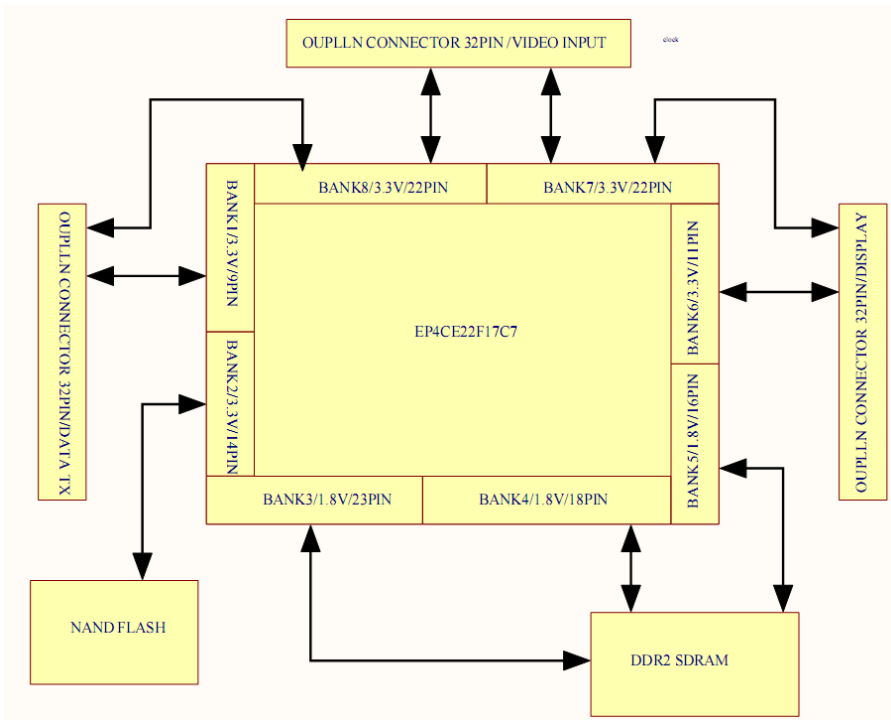
- FPGA 使用 FBGA256 封装的 Altera Cyclone IV EP4CE22F17C8N, 拥有 22320 个逻辑单元 (LEs), 504Kbit 的片内存储器, 66 个嵌入式 8*8 乘法器。
- DDR2 使用 Micron 的 MT47H32M16CC-3, 8M*16bit 合计 512Mbit 的存储量, 可以稳定运行在 200MHz 时钟, 即 400M*16bit 的最大理论数据吞吐量。
- NAND FLASH 使用 Hynix 的 H27U1G8F2B, 128M*8bit 合计 1Gbit 存储量。
- 配置芯片 SPI FLASH 使用 ST 的 M25P16, 16Mbit 存储量。
- 25MHz 的 KDS 原装晶振。



- 4 路最大 800mA 的 LDO, 分别提供系统所需的 3.3V、2.5V、1.8V 和 1.2V。
- 标准 JTAG 接口, 供下载、在线调试使用。
- 3 个 32PIN 的 OUPLLN 插座引出 70 个 FPGA 的 IO 脚。

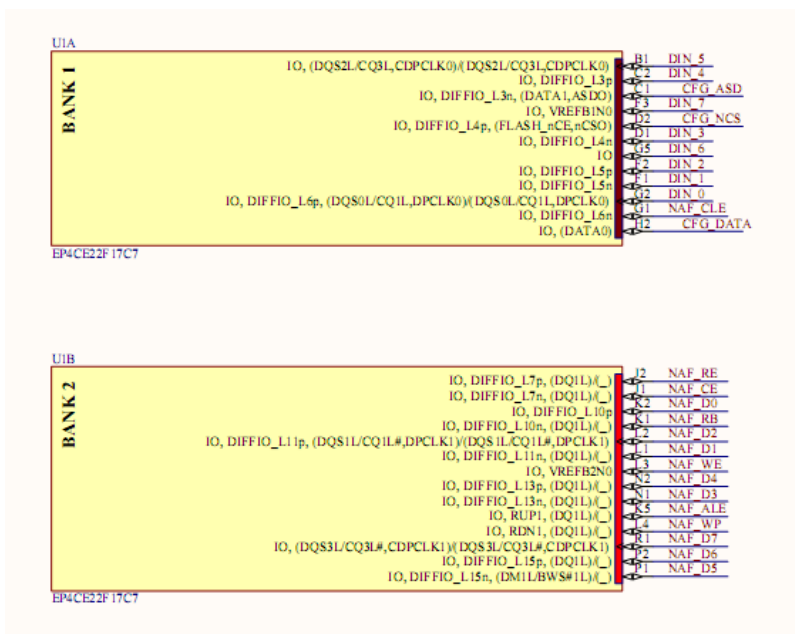
1.2 硬件电路说明

VIP 核心板的整板硬件连接示意如图所示。

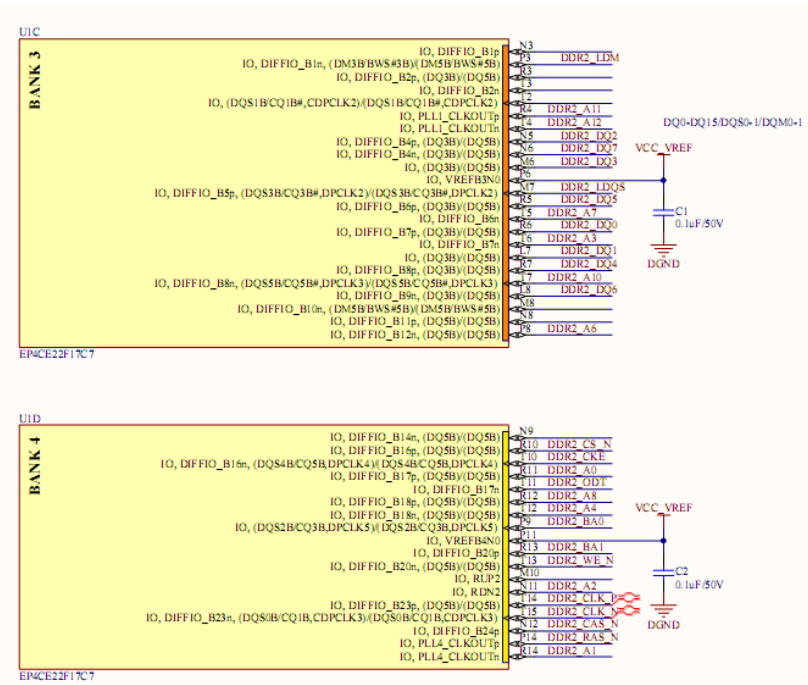


Bank1 的 IO 连接配置芯片的 3 个信号、NAND Flash 的 1 个信号, 其它信号都连接到 OUPLLN 连接器上。

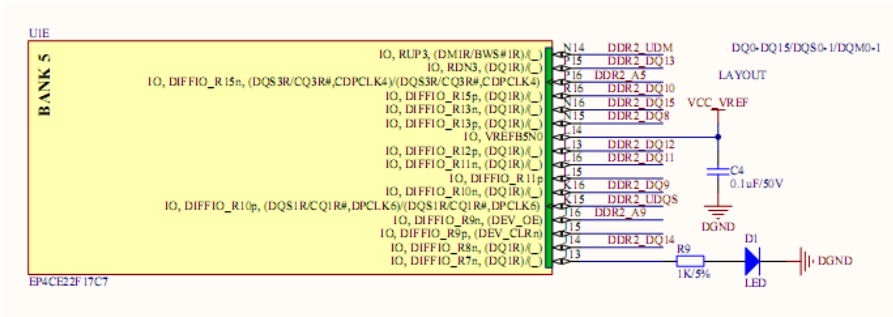
Bank2 的 IO 则都连接到了 NAND Flash 上。



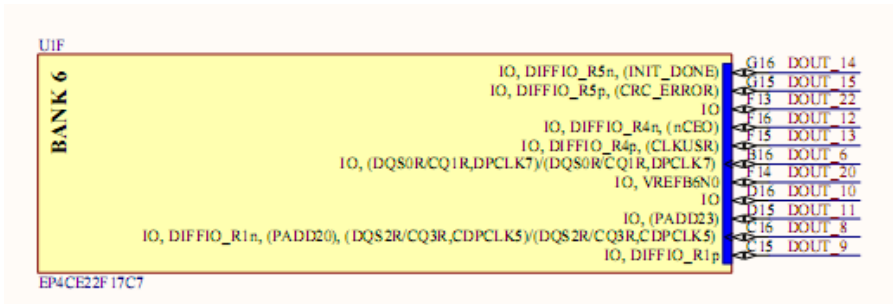
Bank3/4/5 专门用于连接 DDR2，注意这些引脚不是随便定义的，需要参考器件手册和引脚定义来连接。Bank IO 是 1.8V。

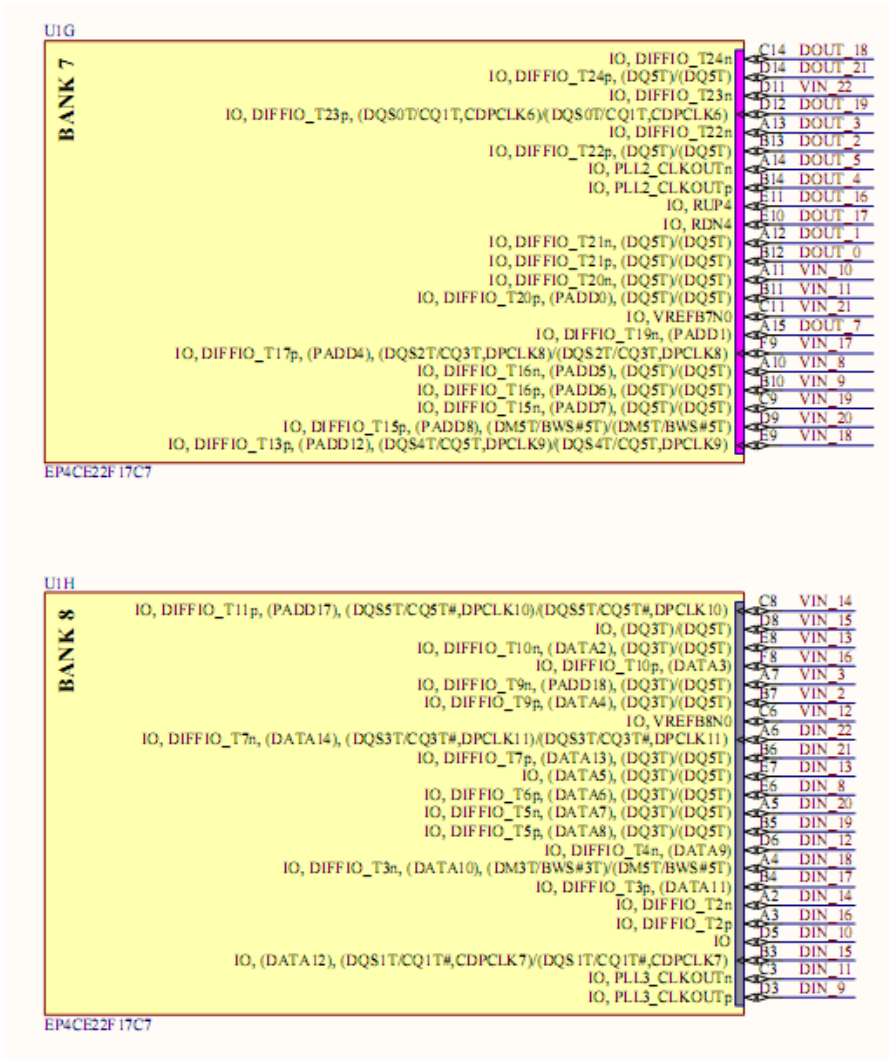


《圣经》箴言九 11“敬畏耶和华是智慧的开端，认识至胜者便是聪明。”

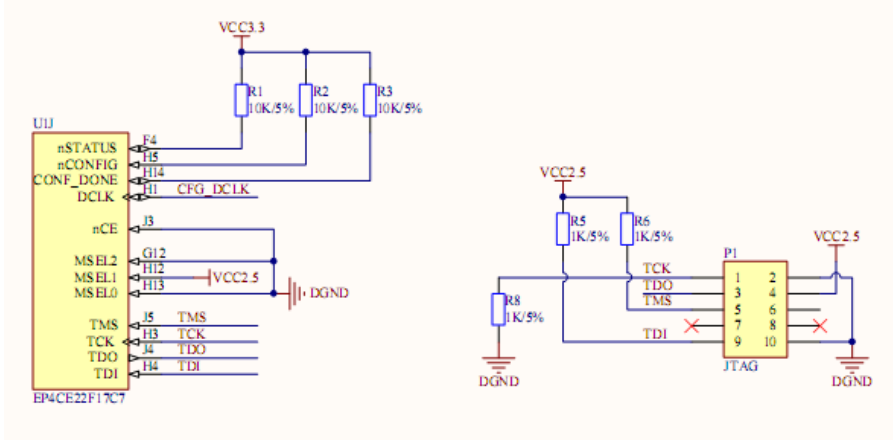


Bank6/7/8 的 IO 则都连接到了 OUPLLN 连接器上引出。

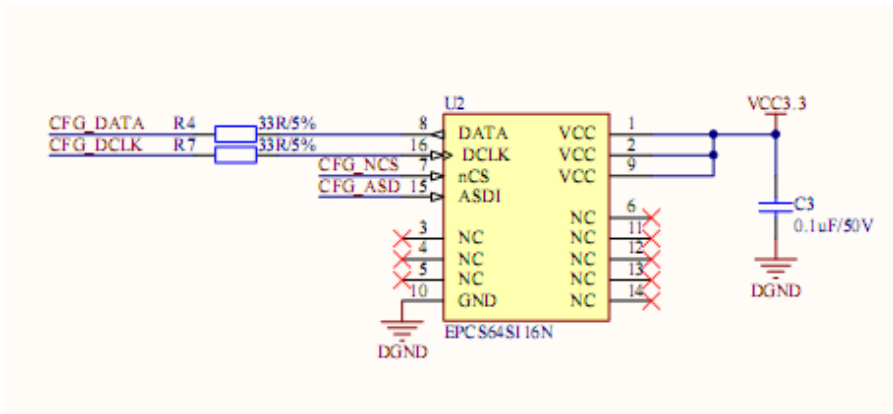




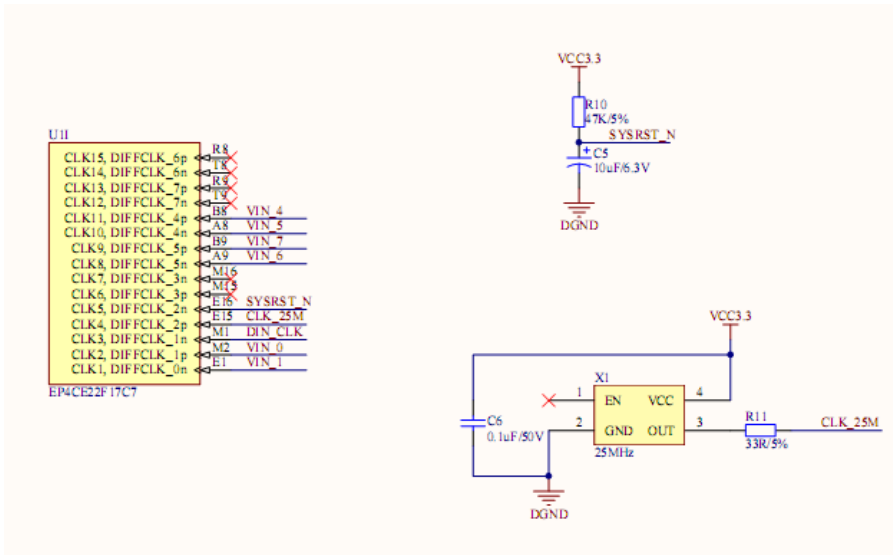
FPGA 配置专用引脚和 JTAG 插座的定义连接如下图所示。



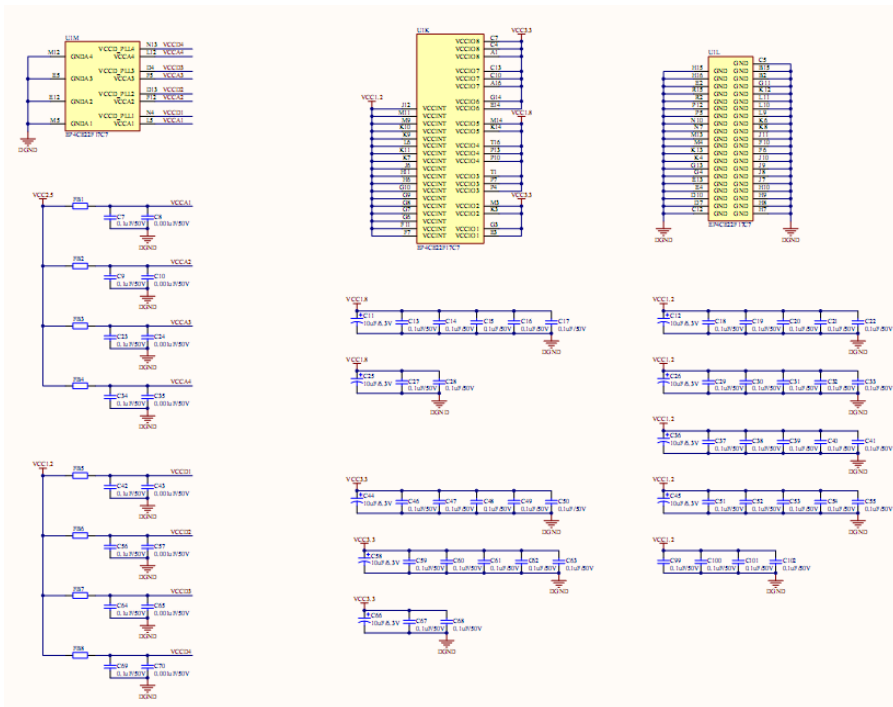
配置芯片的连接如下图所示。



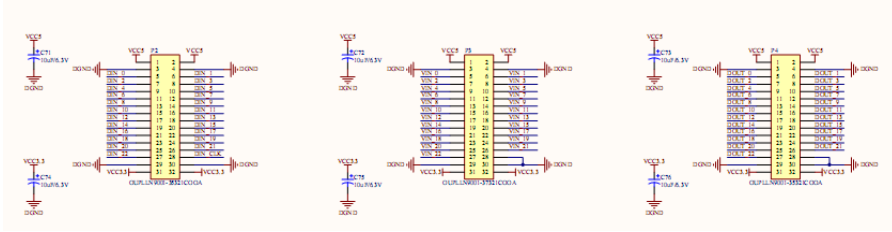
时钟电路和阻容复位电路如下图所示。FPGA 的时钟专用输入引脚还连接了一些 OUTPPN 上的引脚。



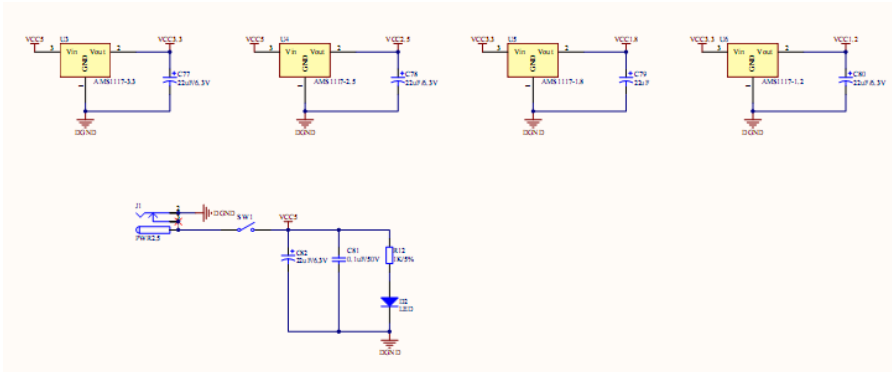
FPGA 的电源设计比较复杂，其电路如下图所示。



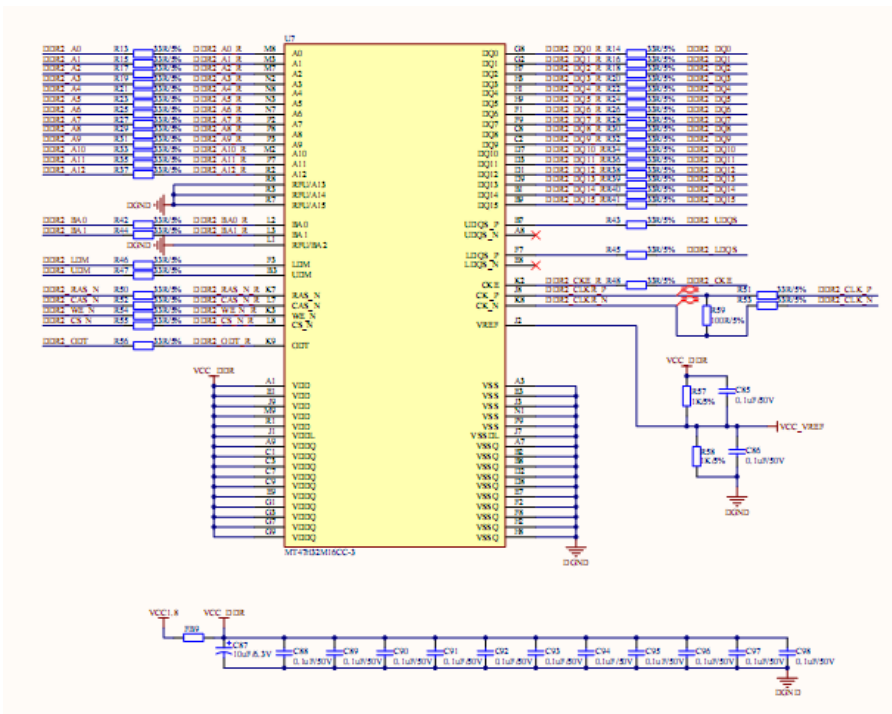
3 个 OUTPPN 连接器的定义如下图所示。



4 个 LDO 供电电路和电源输入插座电路如下图所示。

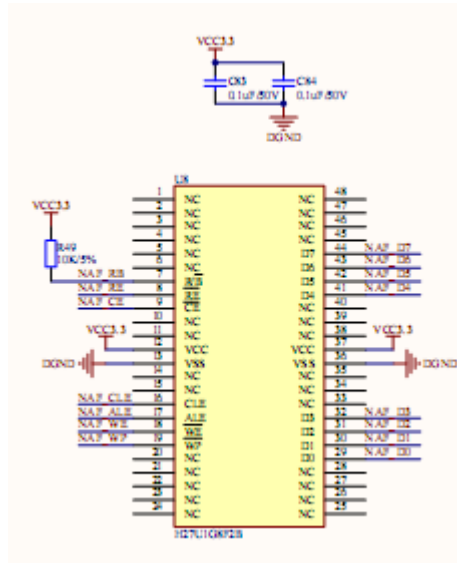


DDR2 的电路如图所示。



NAND Flash 电路如下图所示。

《圣经》箴言九 11“敬畏耶和华是智慧的开端，认识至胜者便是聪明。”



1.3 引脚定义

```
# 时钟和复位引脚
set_location_assignment PIN_E16 -to ext_rst_n
set_location_assignment PIN_E15 -to ext_clk
# LED 指示灯引脚
set_location_assignment PIN_J13 -to led
# DDR2 引脚
set_location_assignment PIN_T14 -to mem_clk[0]
set_location_assignment PIN_T15 -to mem_clk_n[0]
set_location_assignment PIN_N16 -to mem_dq[15]
set_location_assignment PIN_J14 -to mem_dq[14]
set_location_assignment PIN_P15 -to mem_dq[13]
set_location_assignment PIN_L13 -to mem_dq[12]
set_location_assignment PIN_L16 -to mem_dq[11]
set_location_assignment PIN_R16 -to mem_dq[10]
set_location_assignment PIN_K16 -to mem_dq[9]
```



```
set_location_assignment PIN_N15 -to mem_dq[8]
set_location_assignment PIN_N6 -to mem_dq[7]
set_location_assignment PIN_L8 -to mem_dq[6]
set_location_assignment PIN_R5 -to mem_dq[5]
set_location_assignment PIN_R7 -to mem_dq[4]
set_location_assignment PIN_M6 -to mem_dq[3]
set_location_assignment PIN_N5 -to mem_dq[2]
set_location_assignment PIN_L7 -to mem_dq[1]
set_location_assignment PIN_R6 -to mem_dq[0]
set_location_assignment PIN_K15 -to mem_dqs[1]
set_location_assignment PIN_M7 -to mem_dqs[0]
set_location_assignment PIN_N14 -to mem_dm[1]
set_location_assignment PIN_P3 -to mem_dm[0]
set_location_assignment PIN_T4 -to mem_addr[12]
set_location_assignment PIN_R4 -to mem_addr[11]
set_location_assignment PIN_T7 -to mem_addr[10]
set_location_assignment PIN_J16 -to mem_addr[9]
set_location_assignment PIN_R12 -to mem_addr[8]
set_location_assignment PIN_T5 -to mem_addr[7]
set_location_assignment PIN_P8 -to mem_addr[6]
set_location_assignment PIN_P16 -to mem_addr[5]
set_location_assignment PIN_T12 -to mem_addr[4]
set_location_assignment PIN_T6 -to mem_addr[3]
set_location_assignment PIN_N11 -to mem_addr[2]
set_location_assignment PIN_R14 -to mem_addr[1]
set_location_assignment PIN_R11 -to mem_addr[0]
set_location_assignment PIN_R13 -to mem_ba[1]
set_location_assignment PIN_P9 -to mem_ba[0]
set_location_assignment PIN_N12 -to mem_cas_n
set_location_assignment PIN_T10 -to mem_cke[0]
set_location_assignment PIN_R10 -to mem_cs_n[0]
set_location_assignment PIN_T11 -to mem_odt[0]
```



```
set_location_assignment PIN_P14 -to mem_ras_n
set_location_assignment PIN_T13 -to mem_we_n
# NAND FLASH 引脚
set_location_assignment PIN_K2 -to flash_db[0]
set_location_assignment PIN_L1 -to flash_db[1]
set_location_assignment PIN_L2 -to flash_db[2]
set_location_assignment PIN_N1 -to flash_db[3]
set_location_assignment PIN_N2 -to flash_db[4]
set_location_assignment PIN_P1 -to flash_db[5]
set_location_assignment PIN_P2 -to flash_db[6]
set_location_assignment PIN_R1 -to flash_db[7]
set_location_assignment PIN_K1 -to flash_r_bn
set_location_assignment PIN_J2 -to flash_re_n
set_location_assignment PIN_L3 -to flash_we_n
set_location_assignment PIN_J1 -to flash_ce_n
set_location_assignment PIN_G1 -to flash_cle
set_location_assignment PIN_K5 -to flash_ale
set_location_assignment PIN_L4 -to flash_wp_n
# OUTPPN 连接器引脚
# (略)
```

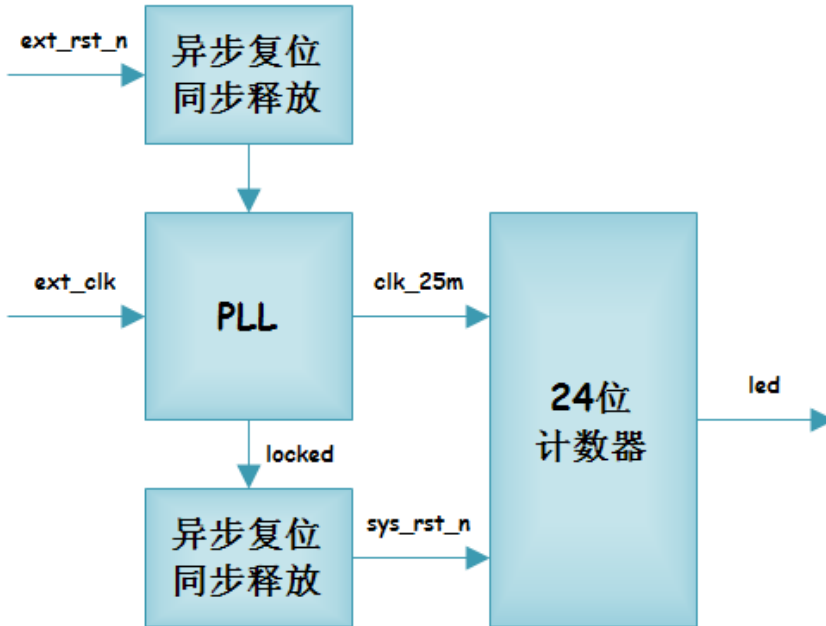


2 开发例程详解

2.1 特权 VIP 例程 01--PLL 例化和 LED 闪烁

功能简介

本实例使用 Megafuction 配置一个 PLL 模块, PLL 模块产生的 25MHz 时钟进行 24 位循环计数, 最终输出计数器最高位到 VIP 板的 LED 进行闪烁显示。功能框图如下。



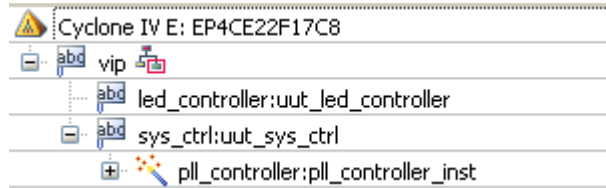
本实例有 3 个模块。

- `Vip.v` 是顶层模块, 其下例化了两个模块。
- `Sys_ctrl.v` 模块中例化了 PLL, 并且对输入 PLL 的复位信号以及 PLL 锁定后的复位信号进行“异步复位, 同步释放”的处理, 确保系统的复位信号稳定可靠。



- Led_controller.v 模块使用一个 24 位计数器产生分频信号，该信号驱动 LED 指示灯闪烁。

3 个模块的层次结构如下图所示。



实验说明

1. 打开光盘目录“...\prj\vip_ex1”下的工程。
2. 使用 Programmer 将“...\prj\vip_ex1\output_files”文件夹下的 vip.sof 文件下载到 VIP 板中，这是 JTAG 在线调试模式，此时可以看到 VIP 板上的指示灯 D1 闪烁。
3. 使用 Programmer 将“...\prj\vip_ex1\output_files”文件夹下的 vip.jic 文件下载到 VIP 板中，此时设计工程文件将固化到配置芯片 M25P16 中。重新给 VIP 板子上电，此时可以看到 VIP 板上的指示灯 D1 仍然闪烁。

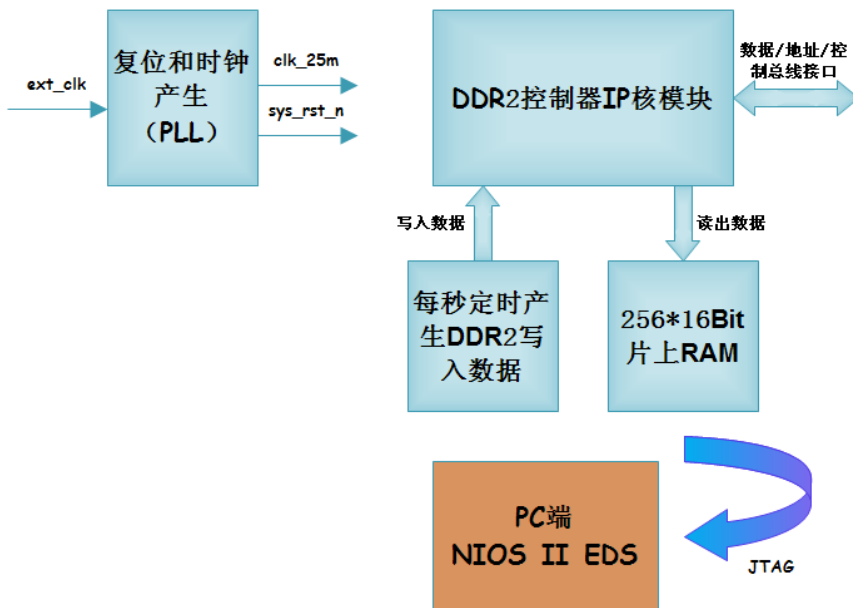


2.2 特权 VIP 例程 02--DDR2 控制器读写测试

功能简介

本实例对 Altera 提供的 DDR2 控制器 IP 核模块进行操作, 每 1.78 秒执行一次 DDR2 的写入和读出操作。先是从 0 地址开始遍历写 256*64bits 数据到 DDR2 的地址 0-1023 中; 在执行完写入后, 执行一次相同地址的读操作, 将读出的 256*64bits 数据写入到片内 RAM 中供查看。在 In-System Memory Content Editor 中可以查看有规律变化的数据。

本实例系统功能框图如下图所示。



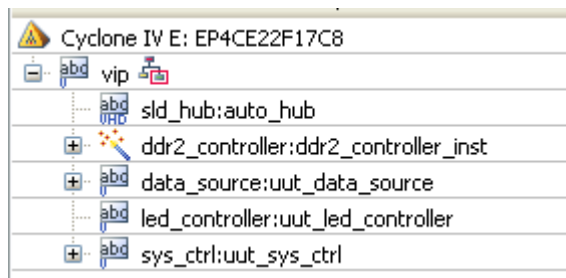
本实例有 4 个模块。

- `Vip.v` 是顶层模块, 其下例化了 3 个模块。
- `Sys_ctrl.v` 模块中例化了 PLL, 并且对输入 PLL 的复位信号以及 PLL 锁定后的复位信号进行“异步复位, 同步释放”的处理, 确保系统的复位信号稳定可靠。



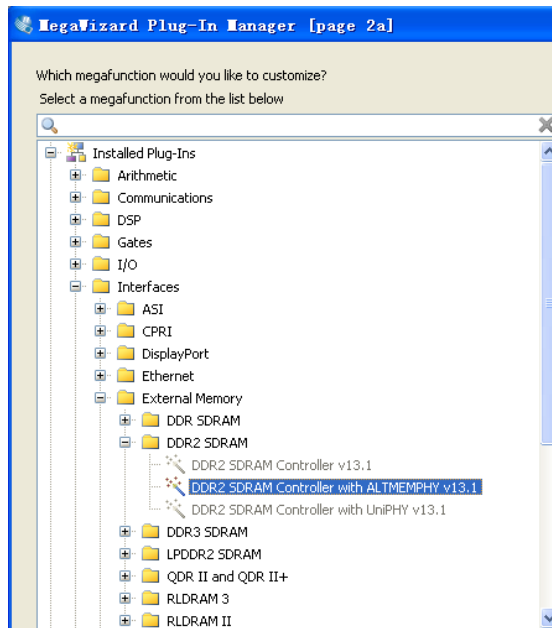
- Ddr2_controller.v 模块是使用 Mefunction 例化的 DDR2 控制器 IP 核模块, 这个模块预留了 DDR2 的读写控制接口供 FPGA 逻辑侧进行操作。
- data_source.v 模块主要用于和 DDR2 控制器进行数据交互, 它既产生写入 DDR2 的数据流, 也读出 DDR2 已经写入地址的数据, 同时将这些数据送往例化的 onchip RAM 中。

4 个模块的层次结构如下图所示。



DDR2 IP 核配置

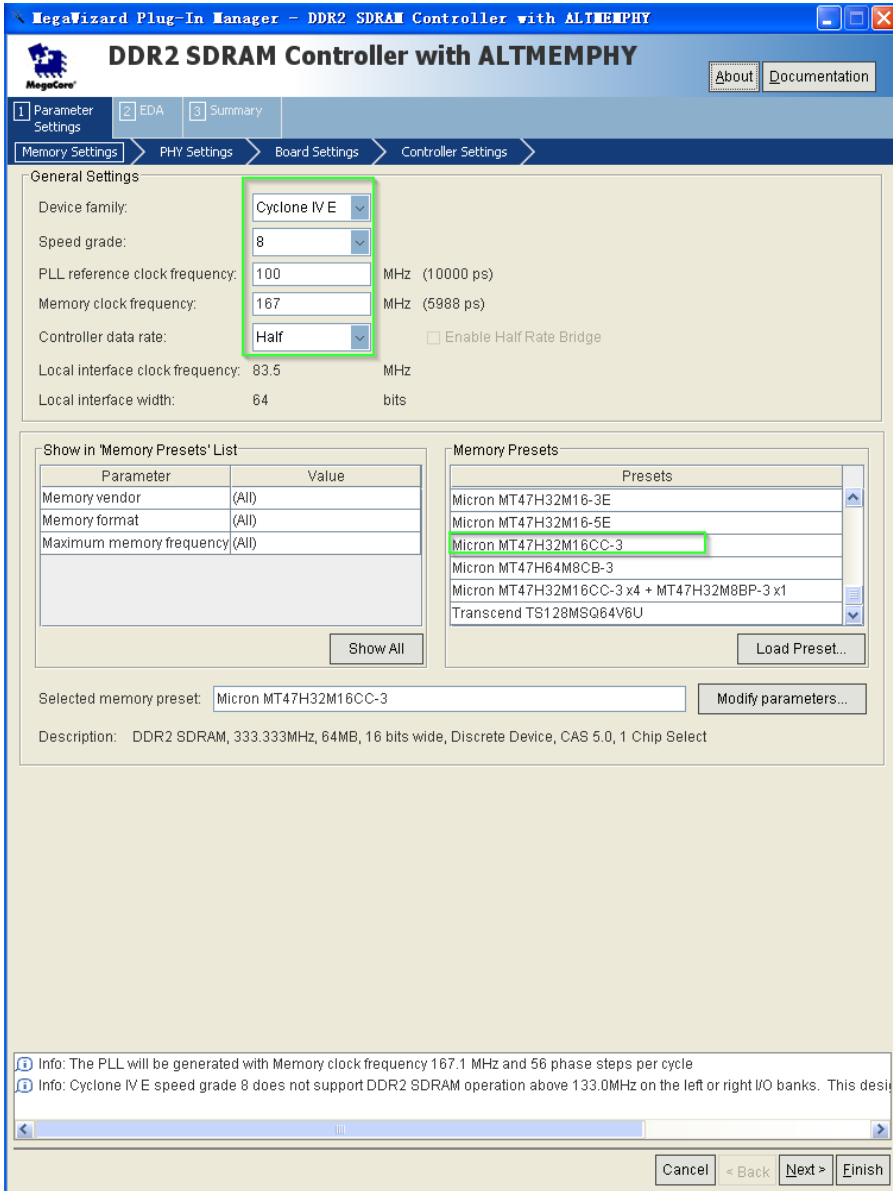
如图所示, 使用 Megafuction 选择 DDR2 的 IP 核。



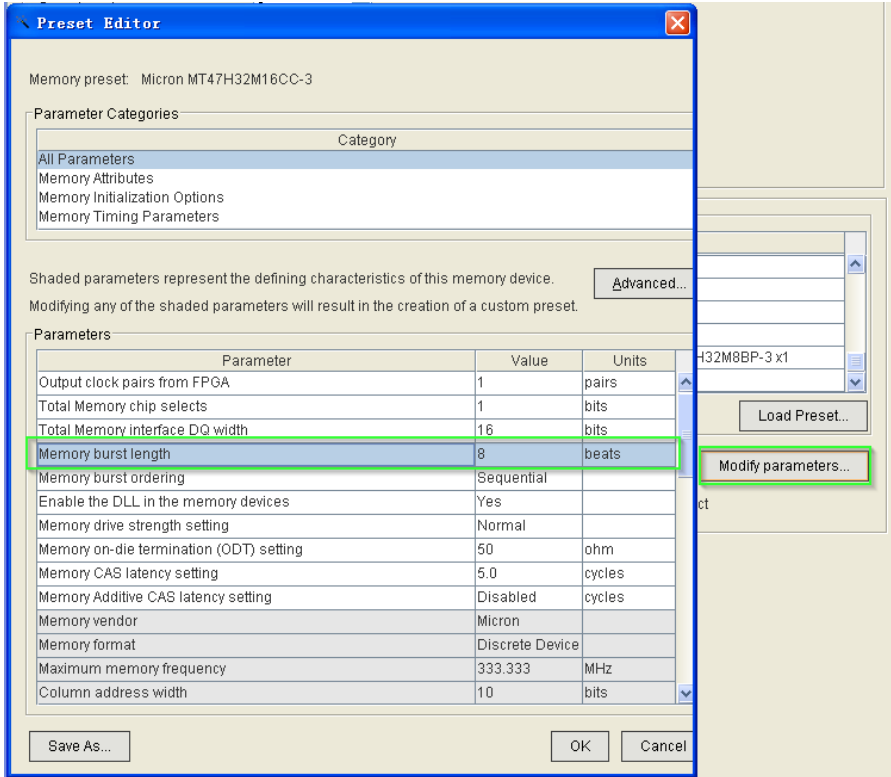
《圣经》箴言九 11“敬畏耶和華是智慧的开端, 认识至胜者便是聪明。”



各个配置页面的设置如下所示。首先预设我们所使用的 DDR2 型号。



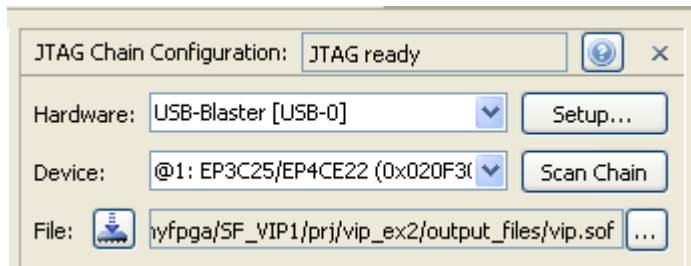
点击“Modify parameters”，如图所示做一些定制化设置，修改 burst length 为 8。



其它选项都使用默认设置。

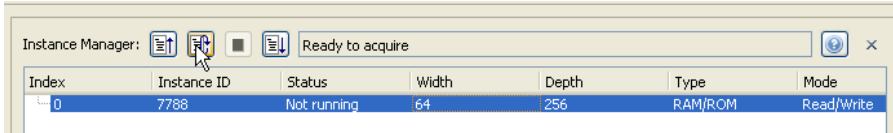
实验说明

1. 打开光盘目录“...\prj\vip_ex2”下的工程。
2. 点击 Quartus II 菜单栏的 Tools→In-System Memory Content Editor, 在界面的右侧, 如下图所示, 选择“...\prj\vip_ex2\output_files”文件夹下的 vip.sof, 执行下载操作, 即点击 File 右侧的小按钮。

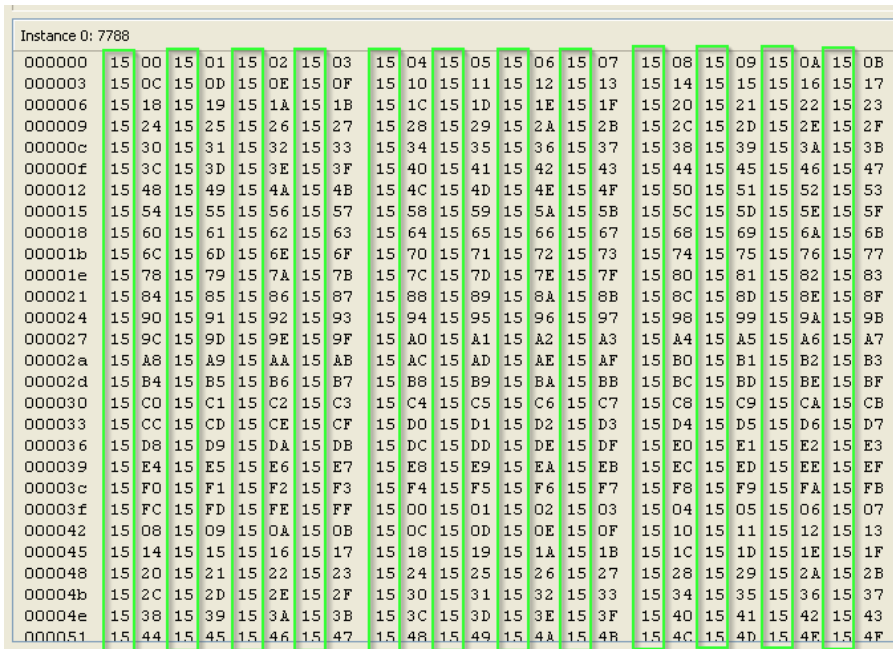




3. 下载完成后可以看到 VIP 板上的指示灯 D1 闪烁。此时我们接着选中 Index 下面的 Memory 项, 然后单击循环读取按钮, 如下图所示。



4. 接着我们可以观察 Memory 当前的数据变化。如下图所示, 绿色框起来的高字节数据, 即我们每隔一秒多统一递增一次的数据, 而其后的低字节数据则始终是从 0 开始递增和地址一一对应的递增数据。因此, 我们看到的实验结果是, 这个 onchip RAM 的所有 16bit 的高字节每隔一秒多递增一, 而其后的数据则一直保持当前状态不变。

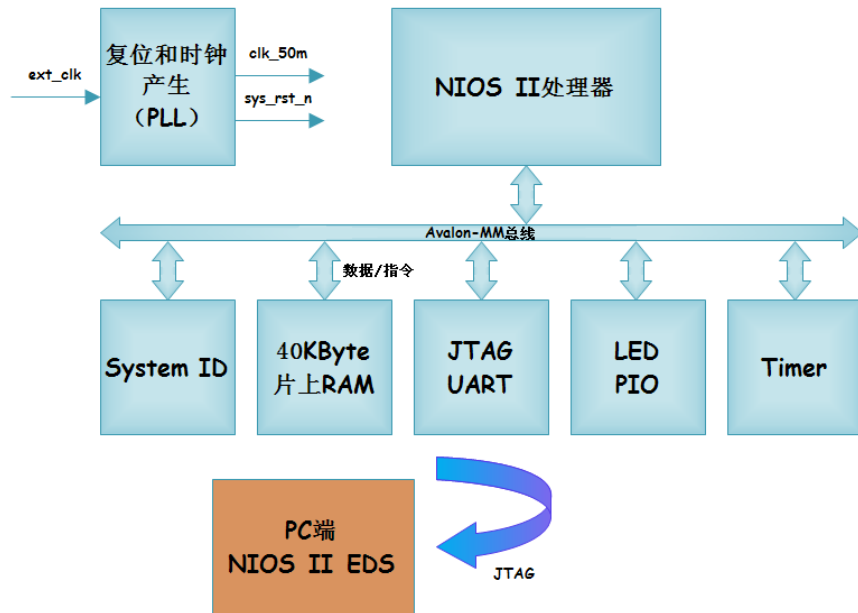




2.3 特权 VIP 例程 03--Qsys 系统与软件实例

功能简介

如图所示, 本实例使用 Quartus II 的 Qsys 评估构建一个片上系统。这个系统有 NIOS II 处理器, 它的代码和数据都是存储在 40Kbyte 的片上 RAM 中运行的。此外, 还有一些常见外设如 System ID、JTAG UART、LED PIO 和 Timer (定时器)。JTAG UART 使用 PC 和 VIP 板之间的 JTAG 连接进行数据传输, 在系统调试中非常实用。

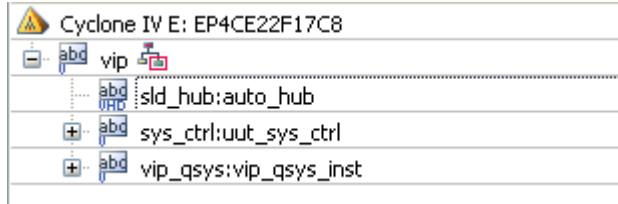


本实例有 3 个模块。

- Vip.v 是顶层模块, 其下例化了两个模块。
- Sys_ctrl.v 模块中例化了 PLL, 并且对输入 PLL 的复位信号以及 PLL 锁定后的复位信号进行“异步复位, 同步释放”的处理, 确保系统的复位信号稳定可靠。
- Vip_qsys.v 模块则是 Qsys 系统的例化。

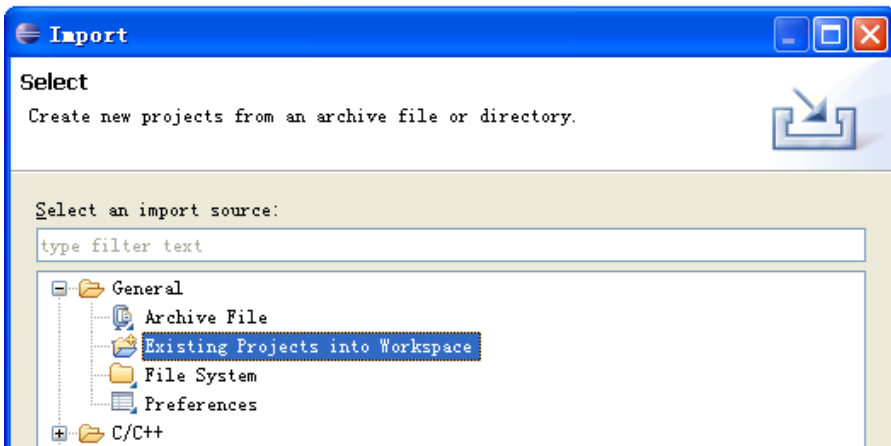


3 个模块的层次结构如下图所示。



实验说明

1. 打开光盘目录“...\prj\vip_ex3”下的工程。
2. 使用 Programmer 将“...\prj\vip_ex3\output_files”文件夹下的 vip.sof 文件下载到 VIP 板中，这是 JTAG 在线调试模式，此时 VIP 板上的指示灯 D1 一直处于熄灭状态。
3. 系统的程序菜单中找到“Nios II 13.1 Software Build Tools for Eclipse”简称 EDS) 并打开。
4. 打开 EDS 后，点击“File→Import”，如下图所示，选择 General→Existing Projects into Workspace，进入下一步。

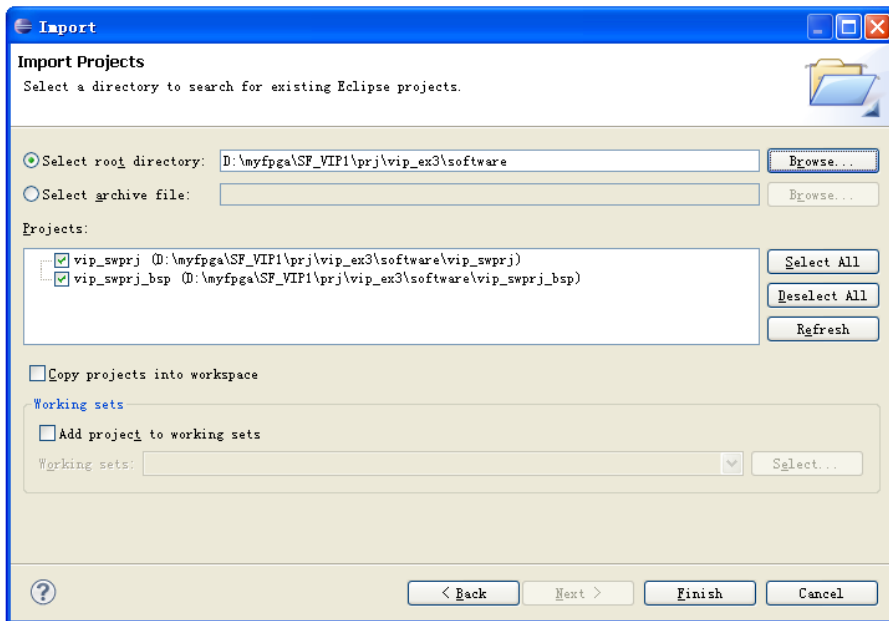


5. 在“Select root directory”后面定位到当前的软件工程目录下，即“...\prj\vip_ex3\software”文件夹下。如图所示，在 Projects 下面将出现 vip_swprj 和 vip_swprj_bsp 两个子文件夹，确认它们都勾选后，点击

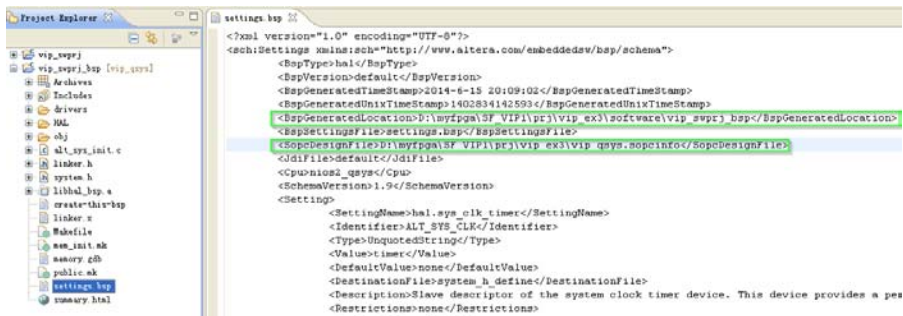
《圣经》箴言九 11“敬畏耶和华是智慧的开端，认识至胜者便是聪明。”



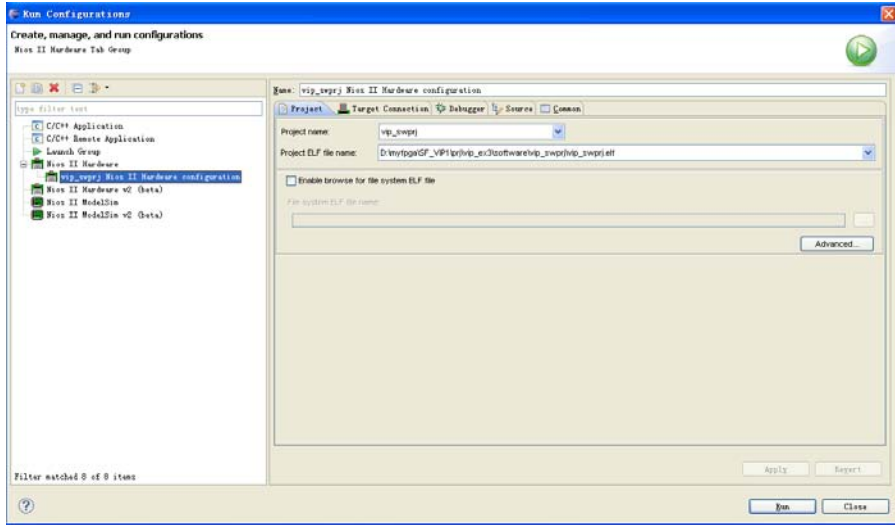
Finish 完成工程导入。



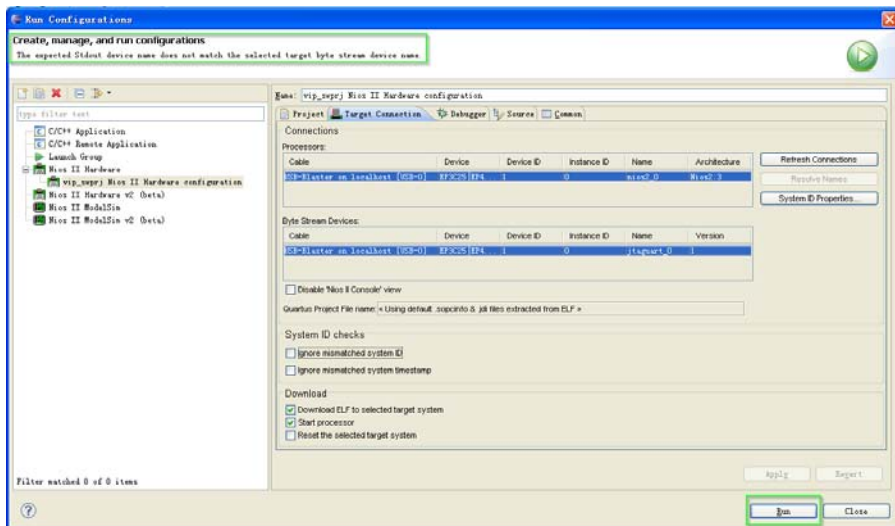
6. 工程打开后, 通常我们还需要在 BSP 工程中更改 setting 文件的某些路径信息。如图所示, 这里的两个路径需要对应为你的工程所在实际路径。



7. 点击菜单栏的“Run→Run Configurations”。如下图所示, 确认选中当前工程的 Nios II Hardware。



8. 单击选中“Target Connection”选项卡，如图所示，可以多次单击“Refresh Connections”直到窗口右上角没有红色的 Error 提醒，并且窗口右下角的 Run 按钮是可点击状态，点击“Run”运行软件。



9. 片刻后，我们可以看到 Nios II Console 窗口中打印出我们在软件上预设的字符如图所示。并且 VIP 板上的指示灯 D1 也开始欢乐的闪烁了。



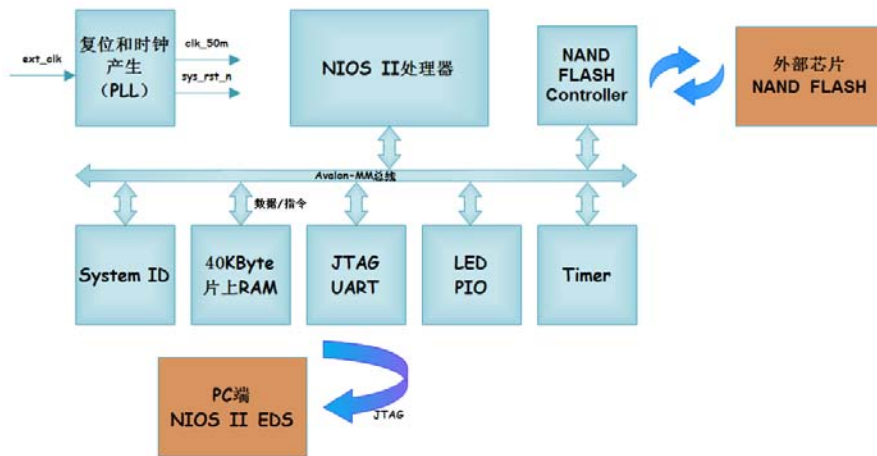
```
Nios II Console
vip_swpri Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0]...
Welcom use VIP board!
Look, LED is running!
```



2.4 特权 VIP 例程 04—NAND FLASH 读写测试

功能简介

本实例在例程 2 的基础上, 添加了一个自定义的 NAND FLASH 控制器组件, 这个组件也是挂在 Qsys 系统的 Avalon-MM 总线上。NAND FLASH 复杂的底层驱动时序都由这个控制器内部产生, 无需 NIOS II 处理器直接参与。NIOS II 处理器只需要通过 Avalon-MM 总线对相关寄存器进行读写就可以实现 NAND FLASH 的读写。



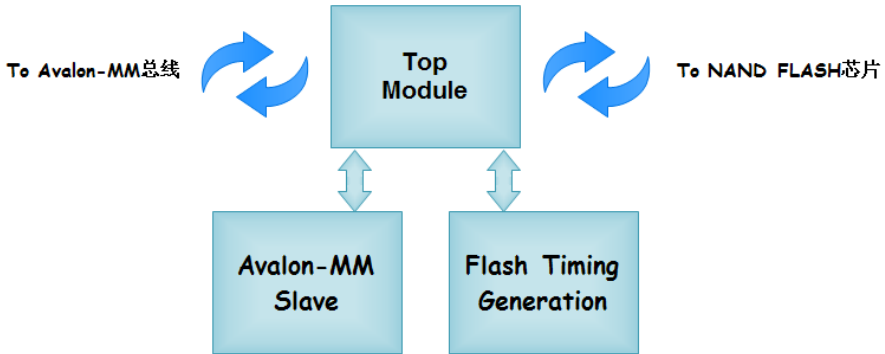
本实例和实例 3 一样, 有 3 个大模块。

- Vip.v 是顶层模块, 其下例化了两个模块。
- Sys_ctrl.v 模块中例化了 PLL, 并且对输入 PLL 的复位信号以及 PLL 锁定后的复位信号进行“异步复位, 同步释放”的处理, 确保系统的复位信号稳定可靠。
- Vip_qsys.v 模块则是 Qsys 系统的例化。

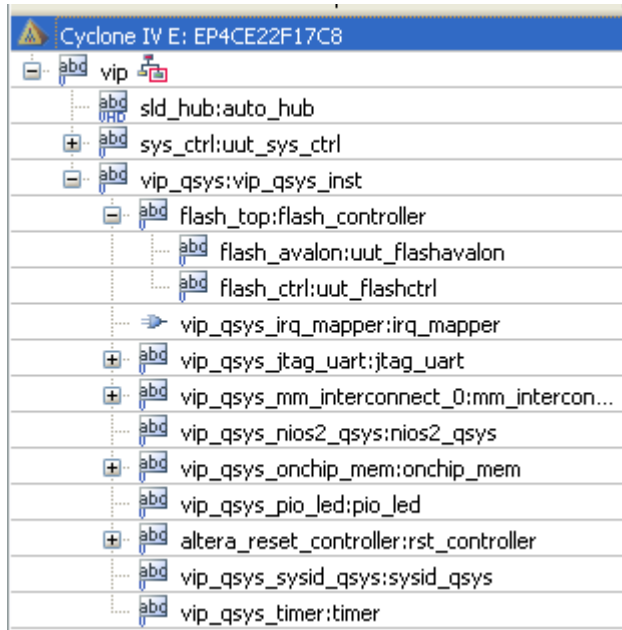
在 Vip_qsys.v 下面, 挂着很多外设组件, 也包括我们自己用逻辑搭建的 NAND FLASH 控制器组件。NAND FLASH 控制器组件有 3 个模块, 一个 top 模块, 其下例化了两个子模块, 分别作为 Avalon-MM 从机逻辑模块和 Flash 《圣经》箴言九 11“敬畏耶和华是智慧的开端, 认识至胜者便是聪明。”



控制时序产生逻辑模块，如下图所示。



整个硬件系统的模块层次如下图所示。



实验说明

1. 打开光盘目录“...\prj\vip_ex4”下的工程。
2. 使用 Programmer 将“...\prj\vip_ex4\output_files”文件夹下的 vip.sof 文件下载到 VIP 板中，这是 JTAG 在线调试模式，此时 VIP 板上的指示



灯 D1 一直处于熄灭状态。

3. 系统的程序菜单中找到“Nios II 13.1 Software Build Tools for Eclipse”简称 EDS) 并打开。导入“...\prj\vip_ex4\software”文件夹下的软件工程(包括应用工程和 BSP 工程)。
4. 运行应用软件, 片刻后, 可以看到 Nios Console 开始打印如下图所示的数据。完成打印后 LED 指示灯 D1 开始闪烁。在打印窗口中, 我们可以看到本实例的软件执行了 4 个主要操作。
 - 擦除 NAND FLASH 的第 1023 Block 的数据。
 - 读出刚刚执行完擦除操作的 NAND FLASH 的第 1023 Block 的第 1 个 Page 数据。刚擦除完, 都是 0xff 的数据。
 - 产生一组 0 到 255 递增的数据写入到 NAND FLASH 的第 1023 Block 的第 1 个 Page 中。
 - 写入完成后, 读出这个 Page 的数据, 以此确认写操作是否正确执行。



```
Nios II Console X
vip_swprj Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instanc
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,

NAND FLASH Block 1023 data write... Successful!

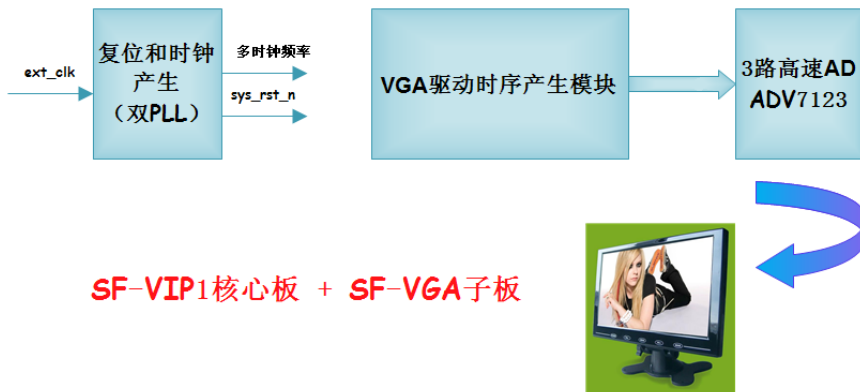
Read NAND FLASH 1023*64 Page data after write... Successful!
0x0, 0x1, 0x2, 0x3, 0x4, 0x5, 0x6, 0x7, 0x8, 0x9, 0xa, 0xb, 0xc, 0xd, 0xe, 0xf,
0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19, 0x1a, 0x1b, 0x1c, 0x1d, 0x1e, 0x1f,
0x20, 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x2a, 0x2b, 0x2c, 0x2d, 0x2e, 0x2f,
0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39, 0x3a, 0x3b, 0x3c, 0x3d, 0x3e, 0x3f,
0x40, 0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47, 0x48, 0x49, 0x4a, 0x4b, 0x4c, 0x4d, 0x4e, 0x4f,
0x50, 0x51, 0x52, 0x53, 0x54, 0x55, 0x56, 0x57, 0x58, 0x59, 0x5a, 0x5b, 0x5c, 0x5d, 0x5e, 0x5f,
0x60, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, 0x67, 0x68, 0x69, 0x6a, 0x6b, 0x6c, 0x6d, 0x6e, 0x6f,
0x70, 0x71, 0x72, 0x73, 0x74, 0x75, 0x76, 0x77, 0x78, 0x79, 0x7a, 0x7b, 0x7c, 0x7d, 0x7e, 0x7f,
0x80, 0x81, 0x82, 0x83, 0x84, 0x85, 0x86, 0x87, 0x88, 0x89, 0x8a, 0x8b, 0x8c, 0x8d, 0x8e, 0x8f,
0x90, 0x91, 0x92, 0x93, 0x94, 0x95, 0x96, 0x97, 0x98, 0x99, 0x9a, 0x9b, 0x9c, 0x9d, 0x9e, 0x9f,
0xa0, 0xa1, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7, 0xa8, 0xa9, 0xaa, 0xab, 0xac, 0xad, 0xae, 0xaf,
0xb0, 0xb1, 0xb2, 0xb3, 0xb4, 0xb5, 0xb6, 0xb7, 0xb8, 0xb9, 0xba, 0xbb, 0xbc, 0xbd, 0xbe, 0xbf,
0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, 0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf,
0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7, 0xd8, 0xd9, 0xda, 0xdb, 0xdc, 0xdd, 0xde, 0xdf,
0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, 0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef,
0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, 0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0xfd, 0xfe, 0xff,
0x0, 0x1, 0x2, 0x3, 0x4, 0x5, 0x6, 0x7, 0x8, 0x9, 0xa, 0xb, 0xc, 0xd, 0xe, 0xf,
0x10, 0x11, 0x12, 0x13, 0x14, 0x15, 0x16, 0x17, 0x18, 0x19, 0x1a, 0x1b, 0x1c, 0x1d, 0x1e, 0x1f,
0x20, 0x21, 0x22, 0x23, 0x24, 0x25, 0x26, 0x27, 0x28, 0x29, 0x2a, 0x2b, 0x2c, 0x2d, 0x2e, 0x2f,
0x30, 0x31, 0x32, 0x33, 0x34, 0x35, 0x36, 0x37, 0x38, 0x39, 0x3a, 0x3b, 0x3c, 0x3d, 0x3e, 0x3f,
0x40, 0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47, 0x48, 0x49, 0x4a, 0x4b, 0x4c, 0x4d, 0x4e, 0x4f,
0x50, 0x51, 0x52, 0x53, 0x54, 0x55, 0x56, 0x57, 0x58, 0x59, 0x5a, 0x5b, 0x5c, 0x5d, 0x5e, 0x5f,
0x60, 0x61, 0x62, 0x63, 0x64, 0x65, 0x66, 0x67, 0x68, 0x69, 0x6a, 0x6b, 0x6c, 0x6d, 0x6e, 0x6f,
0x70, 0x71, 0x72, 0x73, 0x74, 0x75, 0x76, 0x77, 0x78, 0x79, 0x7a, 0x7b, 0x7c, 0x7d, 0x7e, 0x7f,
0x80, 0x81, 0x82, 0x83, 0x84, 0x85, 0x86, 0x87, 0x88, 0x89, 0x8a, 0x8b, 0x8c, 0x8d, 0x8e, 0x8f,
0x90, 0x91, 0x92, 0x93, 0x94, 0x95, 0x96, 0x97, 0x98, 0x99, 0x9a, 0x9b, 0x9c, 0x9d, 0x9e, 0x9f,
0xa0, 0xa1, 0xa2, 0xa3, 0xa4, 0xa5, 0xa6, 0xa7, 0xa8, 0xa9, 0xaa, 0xab, 0xac, 0xad, 0xae, 0xaf,
0xb0, 0xb1, 0xb2, 0xb3, 0xb4, 0xb5, 0xb6, 0xb7, 0xb8, 0xb9, 0xba, 0xbb, 0xbc, 0xbd, 0xbe, 0xbf,
0xc0, 0xc1, 0xc2, 0xc3, 0xc4, 0xc5, 0xc6, 0xc7, 0xc8, 0xc9, 0xca, 0xcb, 0xcc, 0xcd, 0xce, 0xcf,
0xd0, 0xd1, 0xd2, 0xd3, 0xd4, 0xd5, 0xd6, 0xd7, 0xd8, 0xd9, 0xda, 0xdb, 0xdc, 0xdd, 0xde, 0xdf,
0xe0, 0xe1, 0xe2, 0xe3, 0xe4, 0xe5, 0xe6, 0xe7, 0xe8, 0xe9, 0xea, 0xeb, 0xec, 0xed, 0xee, 0xef,
0xf0, 0xf1, 0xf2, 0xf3, 0xf4, 0xf5, 0xf6, 0xf7, 0xf8, 0xf9, 0xfa, 0xfb, 0xfc, 0xfd, 0xfe, 0xff,
```



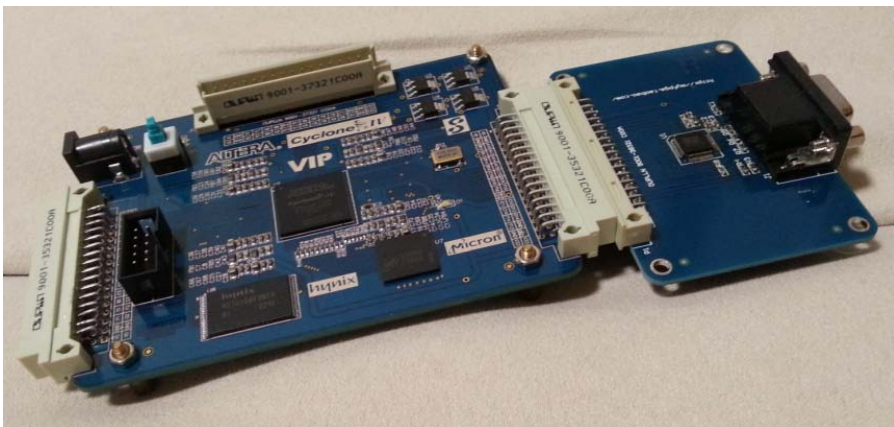
2.5 特权 VIP 例程 05—VGA /SVGA /720p /1080p 多分辨率显示驱动

功能简介

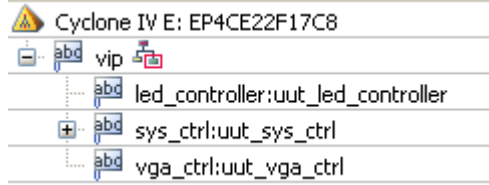
本实例连接 SF-VIP1 核心板和 SF-VGA 子板进行实验。VGA 驱动时序产生模块通过宏定义实现 VGA /SVGA /720p /1080p 多分辨率显示驱动。



SF-VIP1 核心板的 OUTPPN 插座 P4 连接到 SF-VGA 子板的 P1。SF-VIP1 核心板和 SF-VGA 子板的连接示意如下图所示。



整个硬件系统的模块层次如下图所示。



引脚分配

SF-VIP1 核心板的 OUTPPN 插座 P4 连接到 SF-VGA 子板的 P1。其详细引脚定义如下表所示。

信号名	SF-VGA 板定义	SF-VIP1 板定义	FPGA 引脚定义
Vga_r[0]	P1-5	P4-5 / DOUT_0	B12
Vga_r[1]	P1-6	P4-6 / DOUT_1	A12
Vga_r[2]	P1-7	P4-7 / DOUT_2	B13
Vga_r[3]	P1-8	P4-8 / DOUT_3	A13
Vga_r[4]	P1-9	P4-9 / DOUT_4	B14
Vga_g[0]	P1-10	P4-10 / DOUT_5	A14
Vga_g[1]	P1-11	P4-11 / DOUT_6	B16
Vga_g[2]	P1-12	P4-12 / DOUT_7	A15
Vga_g[3]	P1-13	P4-13 / DOUT_8	C16
Vga_g[4]	P1-14	P4-14 / DOUT_9	C15
Vga_g[5]	P1-15	P4-15 / DOUT_10	D16
Vga_b[0]	P1-18	P4-18 / DOUT_13	F15
Vga_b[1]	P1-19	P4-19 / DOUT_14	G16
Vga_b[2]	P1-20	P4-20 / DOUT_15	G15
Vga_b[3]	P1-21	P4-21 / DOUT_16	E11
Vga_b[4]	P1-22	P4-22 / DOUT_17	E10

《圣经》箴言九 11“敬畏耶和华是智慧的开端，认识至胜者便是聪明。”



Vga_clk	P1-23	P4-23 / DOUT_18	C14
Vga_vsy	P1-26	P4-26 / DOUT_21	D14
Vga_hsy	P1-27	P4-27 / DOUT_22	F13
adv7123_sync_n	P1-17	P4-17 / DOUT_12	F16
adv7123_blank_n	P1-16	P4-16 / DOUT_11	D15

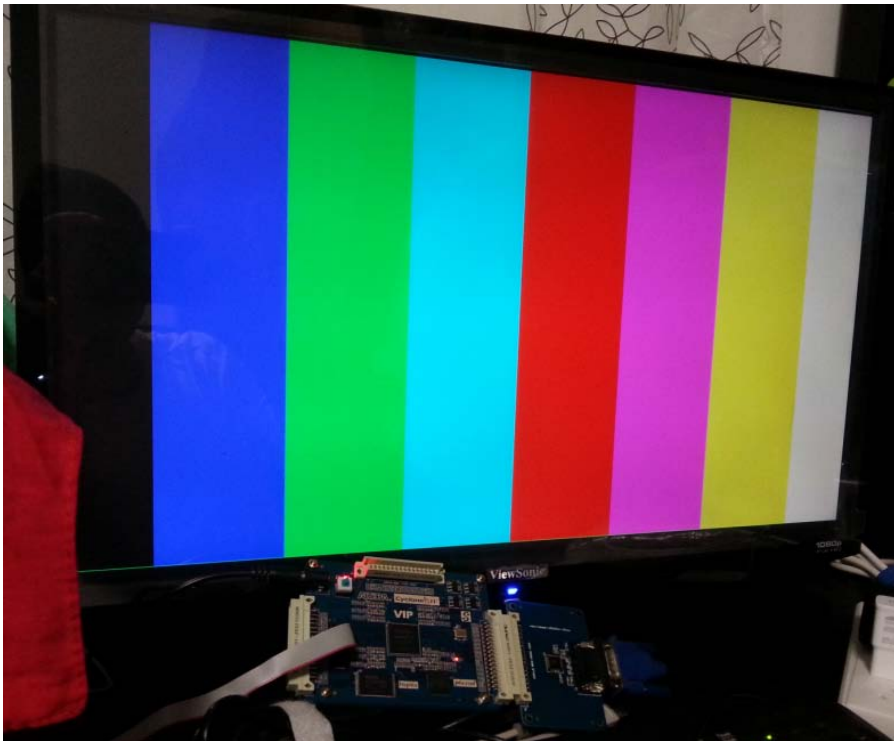
VGA 显示驱动部分的 FPGA 引脚分配 tcl 脚本如下所示。

```
set_location_assignment PIN_A14 -to vga_g[0]
set_location_assignment PIN_A15 -to vga_g[2]
set_location_assignment PIN_C15 -to vga_g[4]
set_location_assignment PIN_D16 -to vga_g[5]
set_location_assignment PIN_B14 -to vga_r[4]
set_location_assignment PIN_G15 -to vga_b[2]
set_location_assignment PIN_C16 -to vga_g[3]
set_location_assignment PIN_B16 -to vga_g[1]
set_location_assignment PIN_A12 -to vga_r[1]
set_location_assignment PIN_A13 -to vga_r[3]
set_location_assignment PIN_B13 -to vga_r[2]
set_location_assignment PIN_B12 -to vga_r[0]
set_location_assignment PIN_F13 -to vga_hsy
set_location_assignment PIN_G16 -to vga_b[1]
set_location_assignment PIN_F15 -to vga_b[0]
set_location_assignment PIN_E11 -to vga_b[3]
set_location_assignment PIN_E10 -to vga_b[4]
set_location_assignment PIN_C14 -to vga_clk
set_location_assignment PIN_D14 -to vga_vsy
set_location_assignment PIN_F16 -to adv7123_sync_n
set_location_assignment PIN_D15 -to adv7123_blank_n
```

实验说明



1. 连接好硬件，SF-VIP1 核心板 + SF-VGA 子板 + VGA 显示器（通用电脑显示器），并且给 VIP1 板上电。
2. 打开光盘目录 “...\prj\vip_ex5” 下的工程。
3. 使用 Programmer 将 “...\prj\vip_ex5\output_files” 文件夹下的 vip.sof 文件下载到 VIP 板中，这是 JTAG 在线调试模式，此时 VIP 板上的指示灯 D1 开始闪烁。
4. 此时电脑显示器上出现了 Color Bar，默认代码的显示驱动分辨率是 1920*1080，大家可以尝试更改 vga_ctrl.v 模块中的宏定义，使用不同分辨率驱动液晶屏显示。1080p 的显示效果如下图所示。





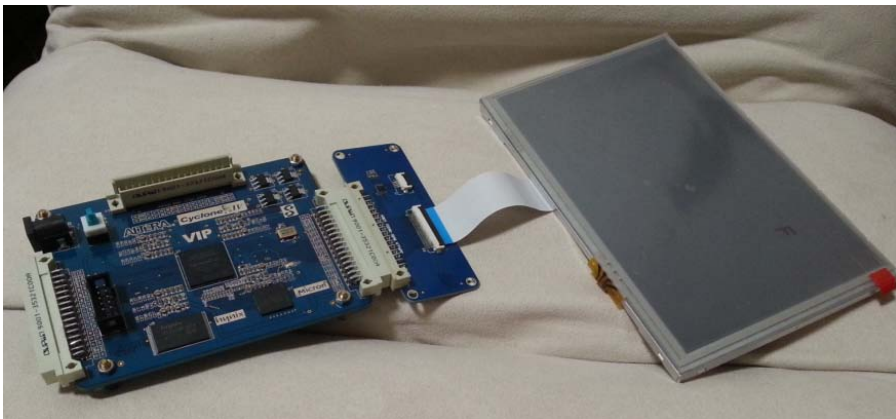
2.6 特权 VIP 例程 06—7 寸 800*480 LCD 显示驱动

功能简介

本实例连接 SF-VIP1 核心板和 SF-L70 子板进行实验。通过抑制 VGA 实例的驱动代码, 可以产生 7 寸 800*480 液晶屏的驱动时序。LCD 的接口是全数字信号驱动, 无需 DAC 芯片转换模拟电平。



整板和液晶屏的装配如下图所示。



引脚分配

SF-VIP1 核心板的 OUTPPN 插座 P4 连接到 SF-LCD 子板的 P2。其详细引

《圣经》箴言九 11“敬畏耶和华是智慧的开端, 认识至胜者便是聪明。”



脚定义如下表所示。

信号名	SF-VGA 板定义	SF-VIP1 板定义	FPGA 引脚定义
Lcd_r[0]	P2-25	P4-25 / DOUT_20	F14
Lcd_r[1]	P2-24	P4-24 / DOUT_19	D12
Lcd_r[2]	P2-23	P4-23 / DOUT_18	C14
Lcd_r[3]	P2-22	P4-22 / DOUT_17	E10
Lcd_r[4]	P2-21	P4-21 / DOUT_16	E11
Lcd_g[0]	P2-20	P4-20 / DOUT_15	G15
Lcd_g[1]	P2-19	P4-19 / DOUT_14	G16
Lcd_g[2]	P2-18	P4-18 / DOUT_13	F15
Lcd_g[3]	P2-17	P4-17 / DOUT_12	F16
Lcd_g[4]	P2-16	P4-16 / DOUT_11	D15
Lcd_g[5]	P2-15	P4-15 / DOUT_10	D16
Lcd_b[0]	P2-14	P4-14 / DOUT_9	C15
Lcd_b[1]	P2-13	P4-13 / DOUT_8	C16
Lcd_b[2]	P2-12	P4-12 / DOUT_7	A15
Lcd_b[3]	P2-11	P4-11 / DOUT_6	B16
Lcd_b[4]	P2-10	P4-10 / DOUT_5	A14
Lcd_de	P2-9	P4-9 / DOUT_4	B14
Lcd_clk	P2-26	P4-26 / DOUT_21	D14
Lcd_en	P2-8	P4-8/DOUT_3	A13

LCD 显示驱动部分的 FPGA 引脚分配 tcl 脚本如下所示。

```
set_location_assignment PIN_A14 -to lcd_b[4]
set_location_assignment PIN_B16 -to lcd_b[3]
set_location_assignment PIN_A15 -to lcd_b[2]
```

《圣经》箴言九 11“敬畏耶和华是智慧的开端，认识至胜者便是聪明。”



```
set_location_assignment PIN_C16 -to lcd_b[1]
set_location_assignment PIN_C15 -to lcd_b[0]
set_location_assignment PIN_B14 -to lcd_de
set_location_assignment PIN_D16 -to lcd_g[5]
set_location_assignment PIN_D15 -to lcd_g[4]
set_location_assignment PIN_F16 -to lcd_g[3]
set_location_assignment PIN_F15 -to lcd_g[2]
set_location_assignment PIN_G16 -to lcd_g[1]
set_location_assignment PIN_G15 -to lcd_g[0]
set_location_assignment PIN_E11 -to lcd_r[4]
set_location_assignment PIN_E10 -to lcd_r[3]
set_location_assignment PIN_C14 -to lcd_r[2]
set_location_assignment PIN_D12 -to lcd_r[1]
set_location_assignment PIN_F14 -to lcd_r[0]
set_location_assignment PIN_A13 -to lcd_en
set_location_assignment PIN_D14 -to lcd_clk
```

实验说明

1. 连接好硬件，SF-VIP1 核心板 + SF-L70 子板 + 7 寸液晶屏，并且给 VIP1 板上电。
2. 打开光盘目录 “...\prj\vip_ex6” 下的工程。
3. 使用 Programmer 将 “...\prj\vip_ex6\output_files” 文件夹下的 vip.sof 文件下载到 VIP 板中，这是 JTAG 在线调试模式，此时 VIP 板上的指示灯 D1 开始闪烁。
4. 此时 LCD 上出现了 Color Bar，显示效果如下图所示。在整个液晶屏显示四个边界则是用绿色线条勾勒出来。

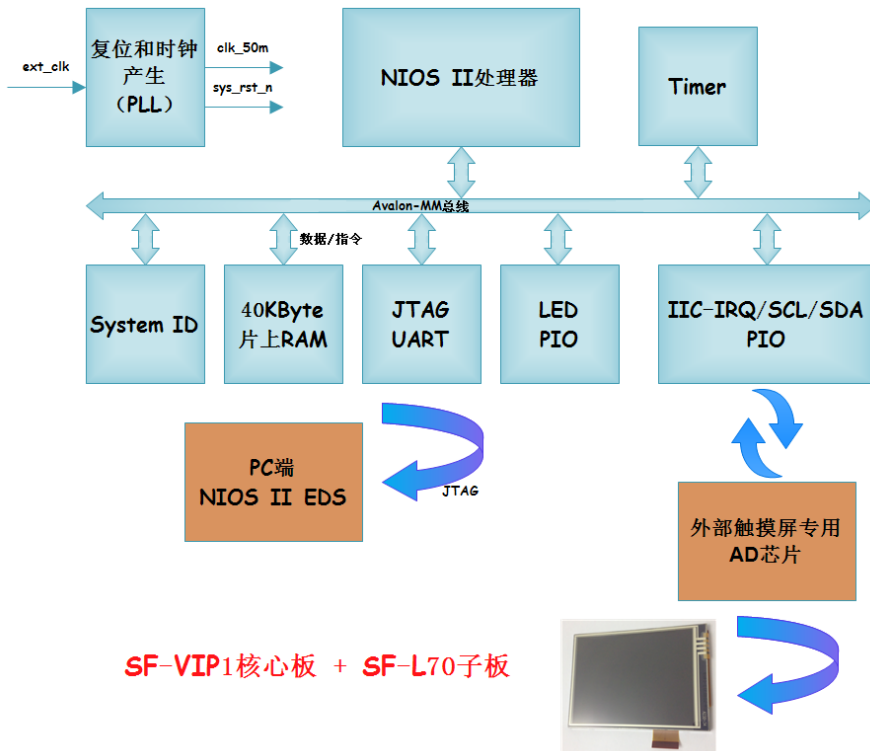




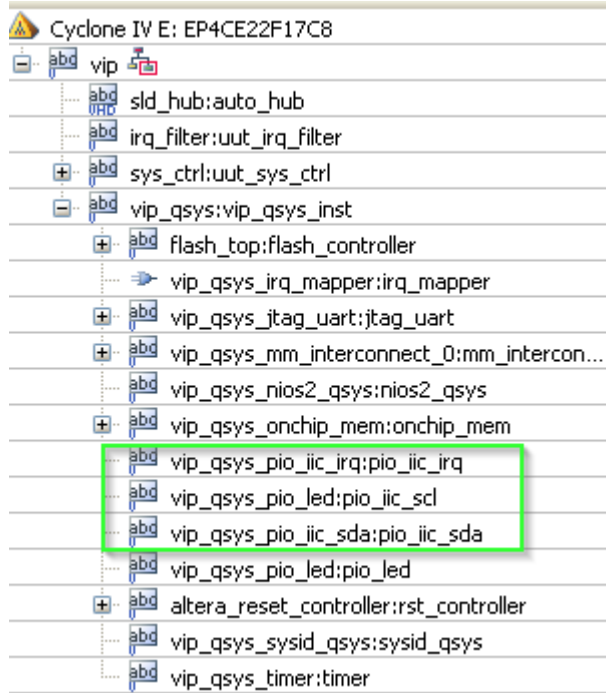
2.7 特权 VIP 例程 07—触摸屏驱动

功能简介

本实例对 SF-L70 子板的 LCD 上的触摸屏进行控制。如图所示, 我们创建 3 个 PIO, 分别作为外部触摸屏芯片的中断输入信号 IRQ、IIC 总线时钟输出信号和 IIC 数据双向信号。在 NIOS II 的软件层, 我们通过模拟 IIC 时序进行通信。触摸屏芯片实际上就是一颗 2 路输入的 ADC 芯片, 我们使用的是电阻式触摸屏, 原理非常简单, 大家可以 baidu 或 google 一下。我们的代码也有很详细的注释, 供大家参考。



该实例工程的模块层次如下图所示, 绿色方框内的 3 个 PIO 组件便是本实例新添加的。



引脚分配

SF-VIP1 核心板的 OUTPPN 插座 P4 连接到 SF-LCD 子板的 P2。其详细引脚定义如下表所示。

信号名	SF-VGA 板定义	SF-VIP1 板定义	FPGA 引脚定义
Pio_iic_irq	P2-7	P4-25 / DOUT_2	B13
Pio_iic_scl	P2-6	P4-24 / DOUT_1	A12
Pio_iic_sda	P2-5	P4-23 / DOUT_0	B12

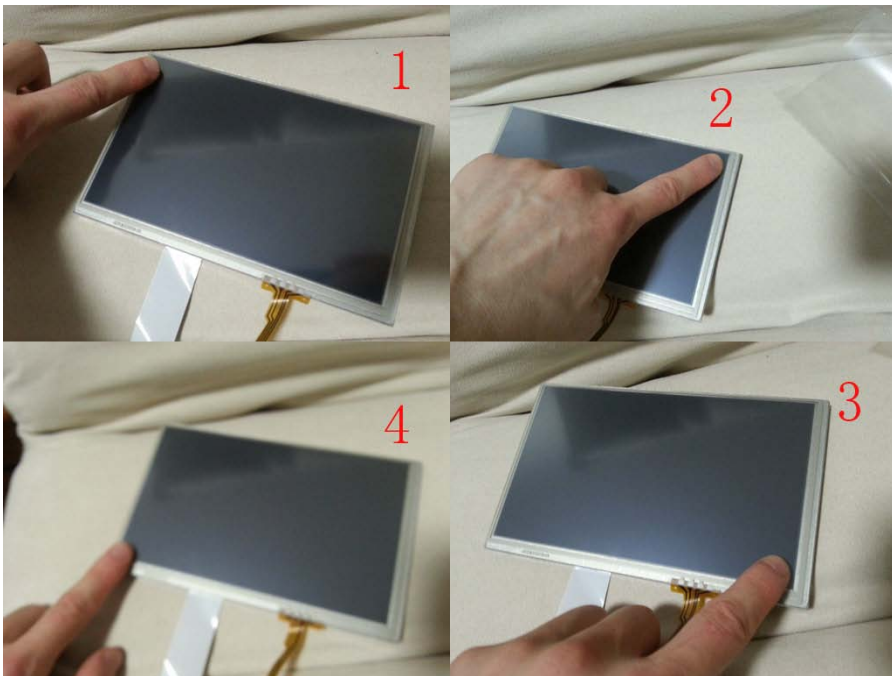
LCD 显示驱动部分的 FPGA 引脚分配 tcl 脚本如下所示。

```
set_location_assignment PIN_B13 -to pio_iic_irq
set_location_assignment PIN_A12 -to pio_iic_scl
set_location_assignment PIN_B12 -to pio_iic_sda
```




实验说明

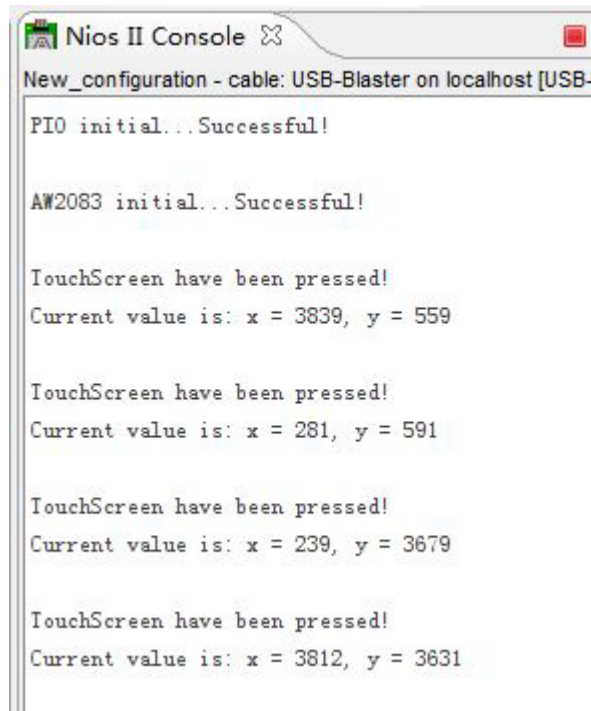
1. 连接好硬件，SF-VIP1 核心板 + SF-L70 子板 + 7 寸液晶屏，并且给 VIP1 板上电。
2. 打开光盘目录 “...\prj\vip_ex7” 下的工程。
3. 使用 Programmer 将 “...\prj\vip_ex7\output_files” 文件夹下的 vip.sof 文件下载到 VIP 板中。
4. 打开 EDS。导入 “...\prj\vip_ex7\software” 文件夹下的软件工程（包括应用工程和 BSP 工程）。
5. 运行应用软件，接着我们可以按照下图所示的顺序分别点击触摸屏的左上角、右上角、右下角和左下角。



6. 点击完毕，我们可以看到如下图所示，Nios II Console 中打印出来刚才触摸屏被按下后 4 组坐标数据。从这四组数据中，我们不难发现，触摸屏被按下后 X 和 Y 值最小的点和我们一般定义的左上角是 (0,0) 坐



标点并不相符（从打印数据看右上角的 X 和 Y 坐标最小）。没关系，我们可以根据实际需要进行处理，对这些数据进行一些运算转换，确保触摸屏被按下后所得到的坐标数据为我们所定义的值。（后续的例程中会对此进行更多的处理。）



```
Nios II Console
New_configuration - cable: USB-Blaster on localhost [USB-...
PIO initial... Successful!

AW2083 initial... Successful!

TouchScreen have been pressed!
Current value is: x = 3839, y = 559

TouchScreen have been pressed!
Current value is: x = 281, y = 591

TouchScreen have been pressed!
Current value is: x = 239, y = 3679

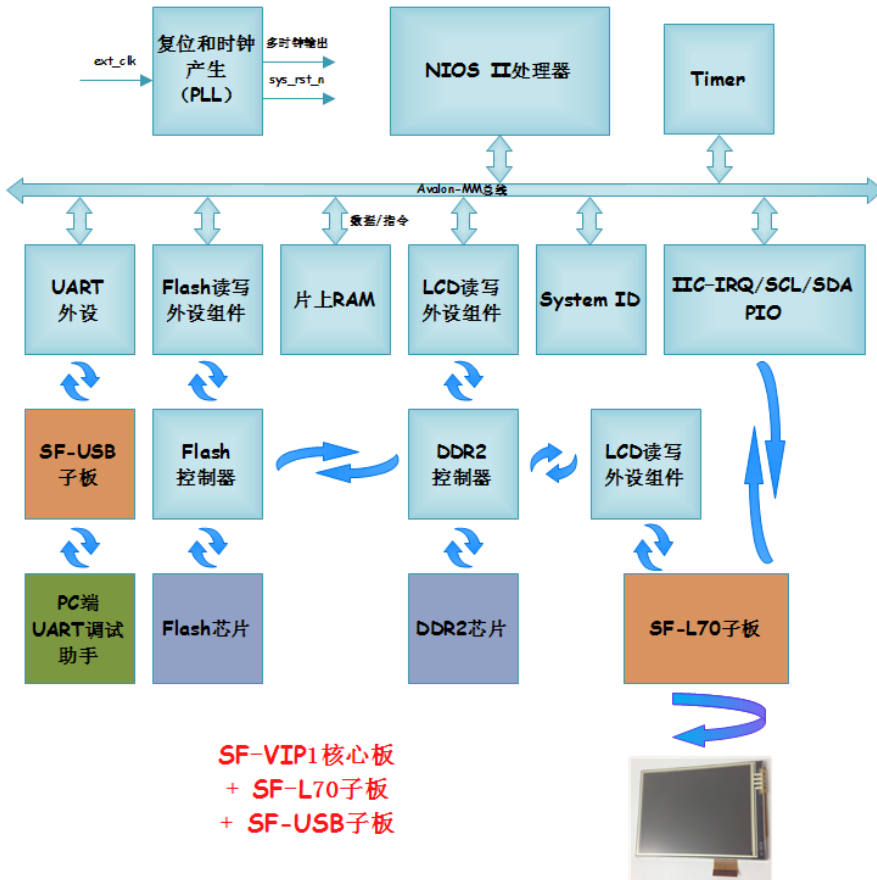
TouchScreen have been pressed!
Current value is: x = 3812, y = 3631
```



2.8 特权 VIP 例程 08—电子点菜单

功能简介

本系统主要实现人机交互界面，主要完成产品的图像预存储、用户指令接收、图像实时显示和界面设计。



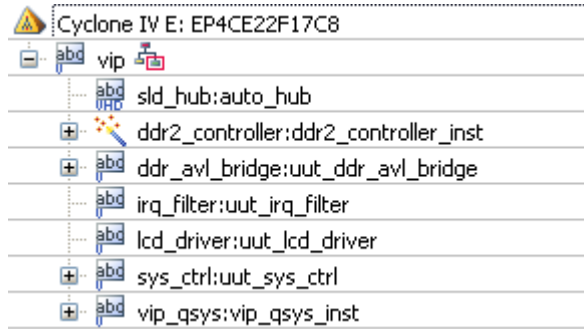
NIOS II 处理器通过 Avalon-MM 总线与系统定制外设进行数据交互，主要的外设包括一个 UART 外设、FLASH 控制器、LCD 控制器以及 GPIO 外设。UART 控制器用于接收上位机软件发送的图片数据，并且接收触摸屏校准指令。FLASH 控制器实现图片数据写入和读出用于显示。LCD 控制器则用于

《圣经》箴言九 11“敬畏耶和华是智慧的开端，认识至胜者便是聪明。”

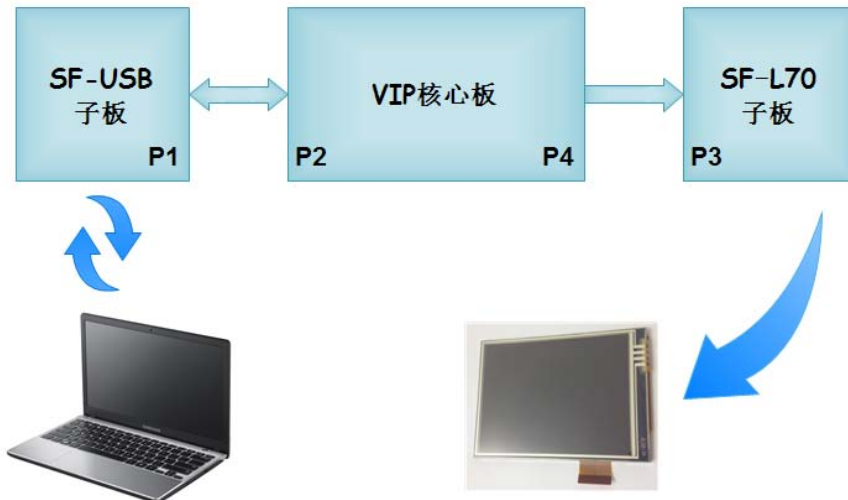


实现 NIOS II 写入当前需要图像数据的显示。GPIO 外设模拟 IIC 时序驱动控制 AD 芯片 (AW2803) 转换的触摸屏坐标参数。

该实例工程的模块层次如下图所示。



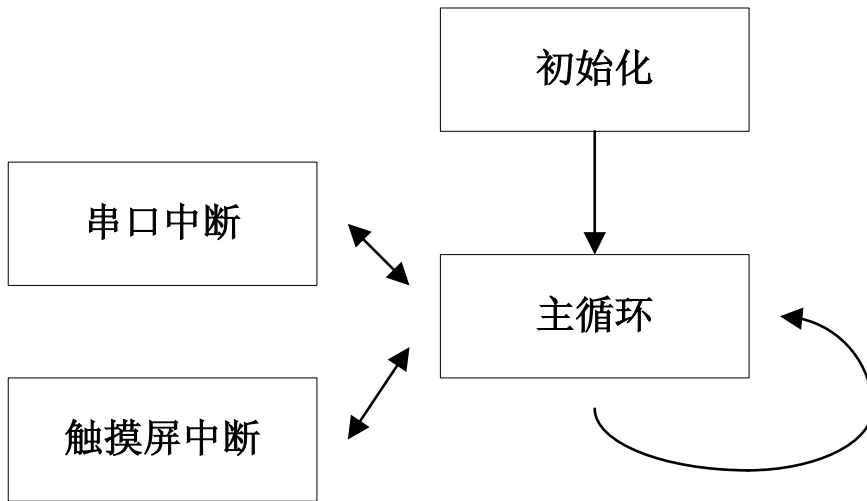
本实例除了主控的 VIP 核心板, 还需要 SF-USB 子板和 SF-L70 子板。它们的连接示意如图所示。VIP 核心板的 P2 插座连接到 SF-USB 子板的 P1 插座, VIP 核心板的 P4 插座连接到 SF-L70 子板的 P3 插座。SF-USB 子板通过 UART-USB 插座 P4 连接到 PC 机, 而 SF-L70 子板的 P1 和 P2 插座则分别连接到 7.0 寸液晶屏的触摸屏和显示控制端口上。



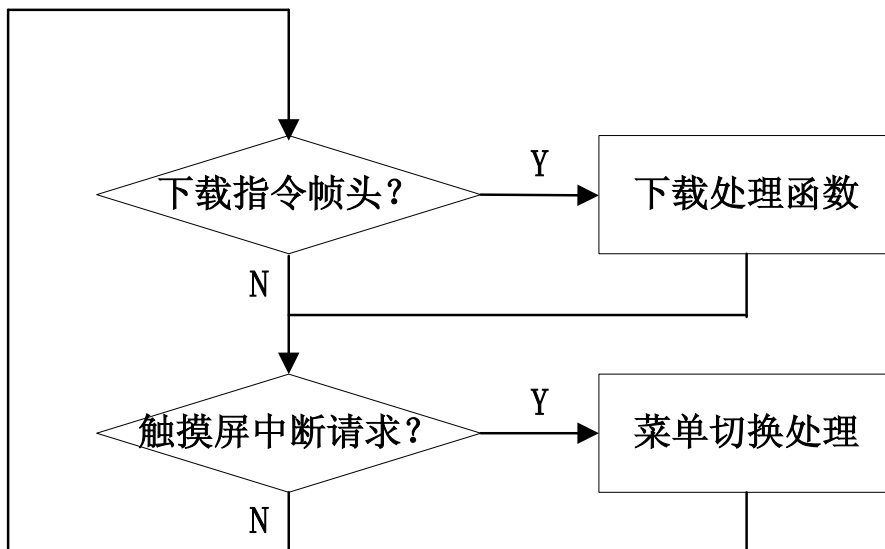
本实例的电子点菜单分 3 级、共 31 张图片。如图所示。



NIOS II 软件的整体流程如图所示。主程序在初始化后，进入主循环中不断检测是否收到串口中断或触摸屏中断。若接收到相应中断，则执行相关操作。



在主循环中,判断串口中断是否接收到下载数据的帧头数据,若是,则进入下载处理函数中执行图片烧录的相关操作。触摸屏中断请求则产生触摸屏坐标,用于菜单切换的判断处理。



引脚分配

《圣经》箴言九 11“敬畏耶和华是智慧的开端,认识至胜者便是聪明。”



前面的实例中都分别给出过 FLASH 接口、DDR2 接口、LCD 接口和触摸屏接口的分配,这里新添加到分配只有通过 SF-USB 子板引出的 UART 接口,它的分配如下所示。

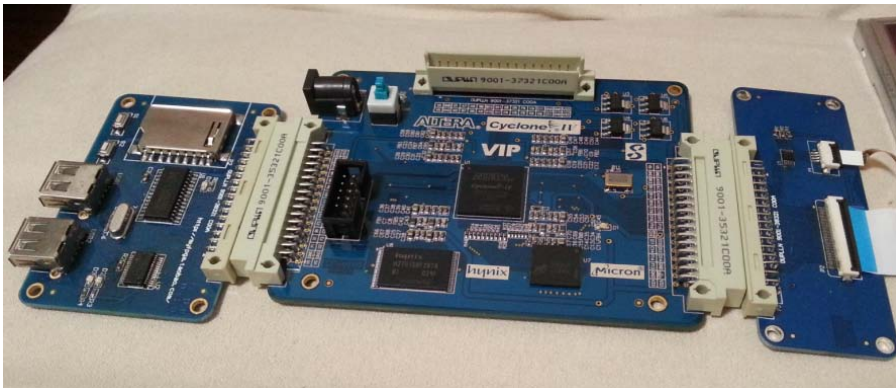
```
set_location_assignment PIN_F1 -to uart_rx  
set_location_assignment PIN_G2 -to uart_tx
```

实验说明

完成这个实验,步骤相对复杂一些,大家需要认真细致的参考这里的说明。

1. 子母板装配。

VIP 核心板的 P2 插座连接到 SF-USB 子板的 P1 插座, VIP 核心板的 P4 插座连接到 SF-L70 子板的 P3 插座。SF-USB 子板通过 UART-USB 插座 P4 连接到 PC 机,而 SF-L70 子板的 P1 和 P2 插座则分别连接到 7.0 寸液晶屏的触摸屏和显示控制端口上。连接示意大体如图所示。



2. 连线。

分别连接好 VIP 核心板的 5V 电源和 USB blaster 下载线。USB 线连接 PC 机和 SF-USB 子板的 P4 插座。

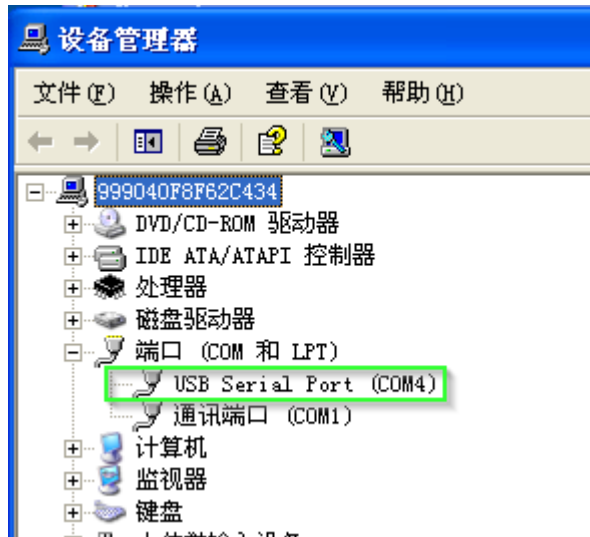
3. USB-UART 驱动安装

《圣经》箴言九 11“敬畏耶和华是智慧的开端,认识至胜者便是聪明。”



给板子上电, 如果是第一次使用 SF-USB 子板, 则会提示安装对应的 FT232 驱动。安装方式可以查看文档《特权 VIP 配套 FT232 驱动安装说明 (by 特权同学) .pdf》。

安装好 FT232 驱动后, 我们可以查看 PC 的设备管理器里面新增加的 COM 口。如图所示, 我们这里新增加了 COM4, 一会在 UART 调试工具里, 我们需要选择 COM4 作为通信端口。



4. 工程下载。

打开 Quartus II 工程, 下载 vip.sof 到 VIP 核心板中。接着打开 EDS, import 当前工程所在文件夹下的 software 文件夹, 打开软件工程, 接着在线运行当前的软件工程。下载完成, 液晶屏显示是花屏 (乱起八糟的), 这是正常情况, 不用理会, 咱图片还没下载进去呢。

5. 开启串口。

vip_ex8 文件夹下的 uart_tools 文件夹中有 UART 调试工具 LCDSOC.exe, 双击打开它。如图所示, 我们首先必须严格按照如下几个步骤进行连接配置。

A. 设置端口号为 COM4。

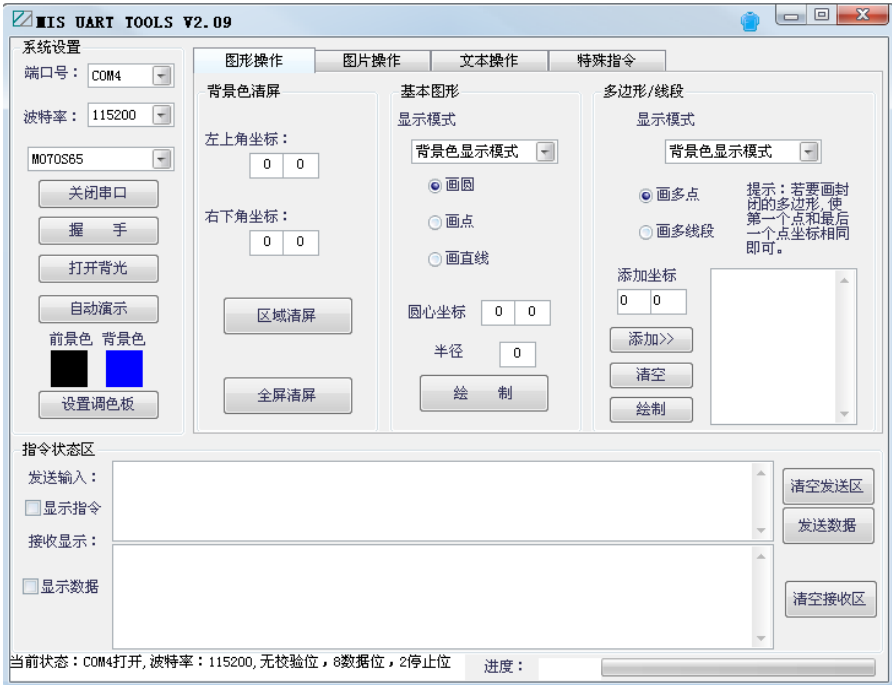
《圣经》箴言九 11“敬畏耶和华是智慧的开端, 认识至胜者便是聪明。”



- B. 点击“请选择模组型号”下拉框, 选择 M070S65。
- C. 点击“打开串口”。
- D. 设置波特率为 115200。



设置完成后如图所示。



我们可以多次点击“握手”命令, 看看 VIP 端是否连接好了。如图所示, 在指令状态区我们看到了“握手成功”的提示。说明我们的整个 VIP

《圣经》箴言九 11“敬畏耶和华是智慧的开端, 认识至胜者便是聪明。”



核心板已经正常工作并连接到 PC 了。

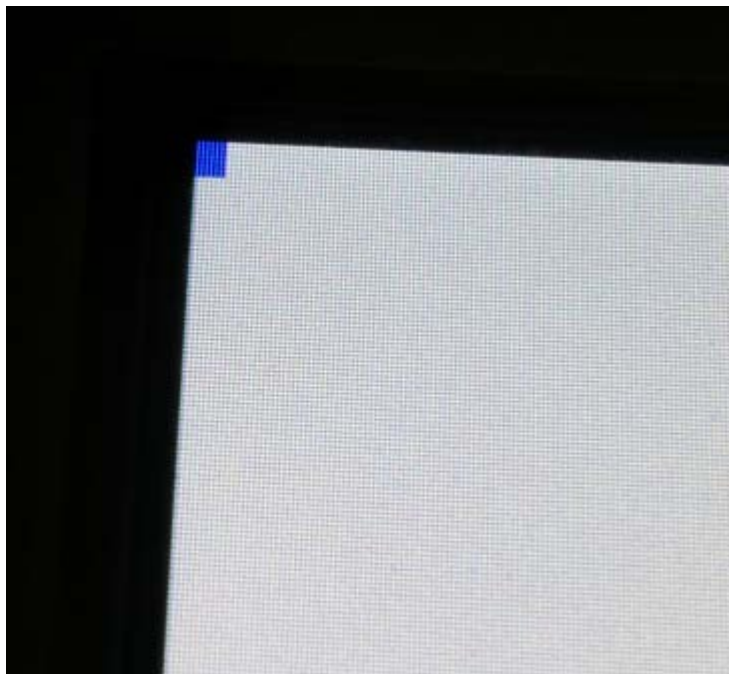


6. 触摸屏校准。

如图所示，我们找到“特殊命令”页面。首先点击“开启触摸屏”按钮。



接着点击“校准触摸屏”按钮。此时我们注意，液晶屏被清屏为白色了，但是在液晶屏的右上角有一个蓝色正方形（如图所示），大家找只触摸笔点击一下那个蓝色正方形，当然是位置越准确越好。点击完后，蓝色正方形就跑到左下角了，接着是右下角，最后是右上角，大家都分别尽可能准确的点击一下。完成好触摸屏校准就 OK 了。

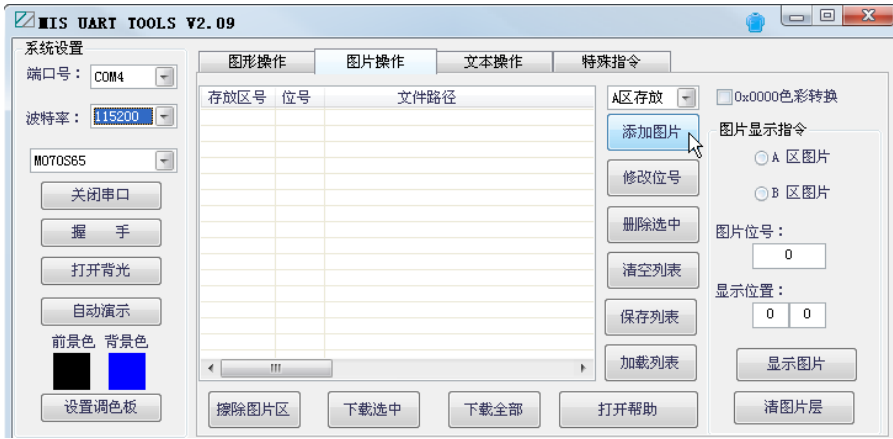


7. 图片下载

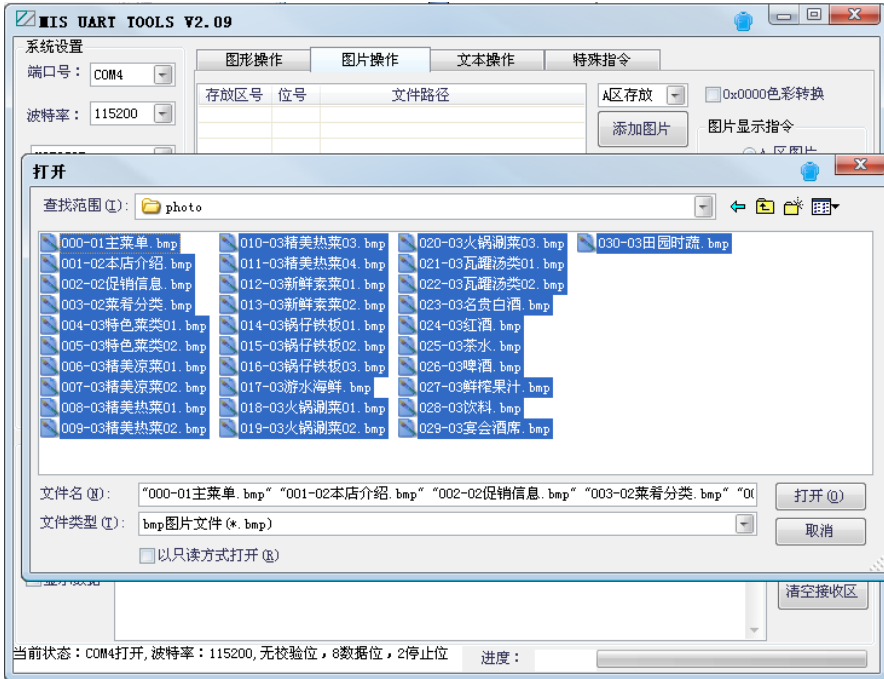
《圣经》箴言九 11“敬畏耶和华是智慧的开端，认识至胜者便是聪明。”



接下来是图片的下载。如图所示,切换到“图片操作”页面,点击“添加图片”按钮。



定位到vip_ex8工程目录下的photo文件夹。首先单击图片文件“030-03 田园时蔬.bmp”,然后按住键盘的 shift 键,再单击图片“000-01 主菜单.bmp”,完成选择后,点击“打开”完成图片加载。



大家可以注意一下,我们刚才的选择图片方式,最终会使图片名称最

《圣经》箴言九 11“敬畏耶和华是智慧的开端,认识至胜者便是聪明。”



前面的三位数字排序和图片的“位号”一一对应。接着我们点击“下载全部”就可以启动图片下载。每张图片的下载大约需要 70 多秒，所有 31 张图片下载完成要 40 来分钟。



8. 效果演示

完成全部图片下载后，我们可以重新运行一次系统的软硬件代码。代码运行起来后，我们就看到液晶屏上显示出了电子点菜单的主页。我们可以点击“本店介绍”后其他两个按钮。



点击“本店介绍”按钮后，图片切换到了下面一张。点击“返回”可以回到主菜单。



若点击“开始点餐”按钮，则显示如图所示。



这里的 12 个分类按钮若被点击, 则进入下一级菜单。例如我们点击了“田园时蔬”按钮, 则显示如图所示的新图片。





2.9 特权 VIP 例程 09—200W 像素 CMOS 摄像头采集显示

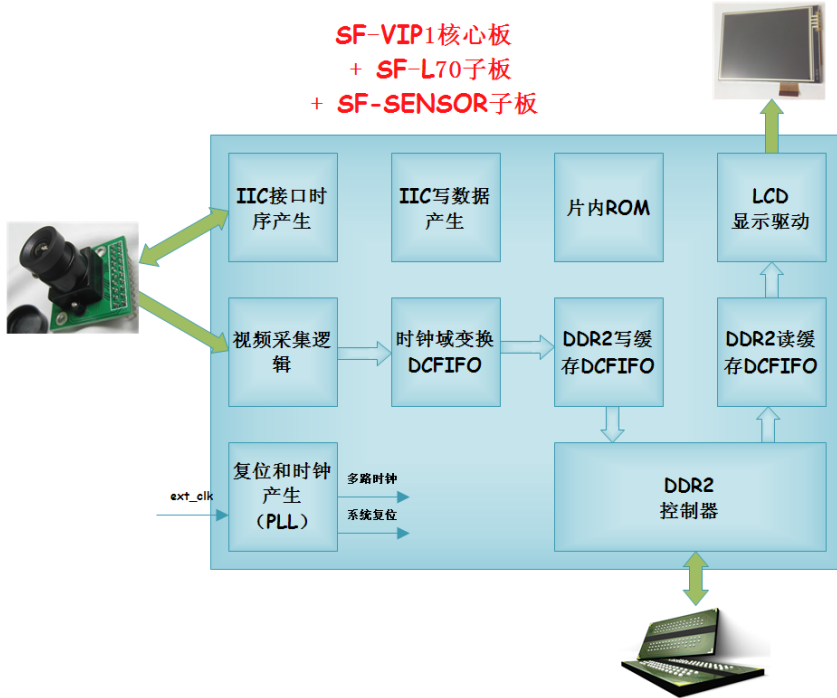
功能简介

本实例使用美光的 CMOS 摄像头 MT9D111, 它是美光的一款在单芯片系统上集成了一个先进的 200 万像素图像传感器和功能强大的图像处理技术芯片。单芯片系统中的自动特性可以调整各种参数, 以便在各种光照条件下拍摄到优质图像。MT9D111 也可以简化设计工程师的工作, 因为所有的处理功能、内存以及与镜头的接口都集成在一个单传感器处理芯片上。

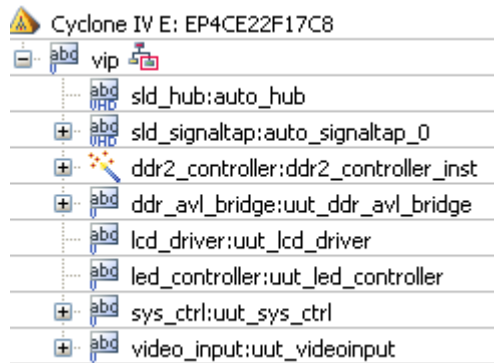
MT9D111 是一颗 1/3 英寸、200 万像素的 CMOS 图像传感器, 它自带集成的先进相机系统。此相机系统包含一个复杂的图像流处理器 (IFP)、一个实时 JPEG 编码器、一个集成的微控制器、闪光控制、自动聚焦、光学缩放以及机械快门。整个系统级芯片 (SoC) 可以在低照度条件下具备卓越的性能, 同时功耗很低。

FPGA 工程的功能框图如图所示。上电初始, FPGA 需要通过 IIC 接口协议对摄像头模块进行寄存器初始化配置。这个初始化的基本参数, 如初始化地址和数据存储在一个预先配置好的 FPGA 内嵌 ROM 中。在初始化配置完成后, 摄像头就能够持续输出 RGB 标准的视频数据流, FPGA 通过对其相应的时钟、行频和场频进行检测, 从而一帧一帧的实时采集图像数据。

采集到的视频数据先通过一个 FIFO, 将原本 25MHz 频率下同步的数据流转换到 50MHz 频率下。接着将这个数据再送入写 DDR2 缓存的 FIFO 中, 最终这个 FIFO 每满 8 个 64bit 数据就会将其写入 DDR2 的相应地址中。在另一侧, 使用另一个异步 FIFO 将 DDR2 缓存的图像数据送给 LCD 驱动模块。LCD 驱动模块不断的读出新的显示图像, 并且驱动 7 寸液晶屏工作。

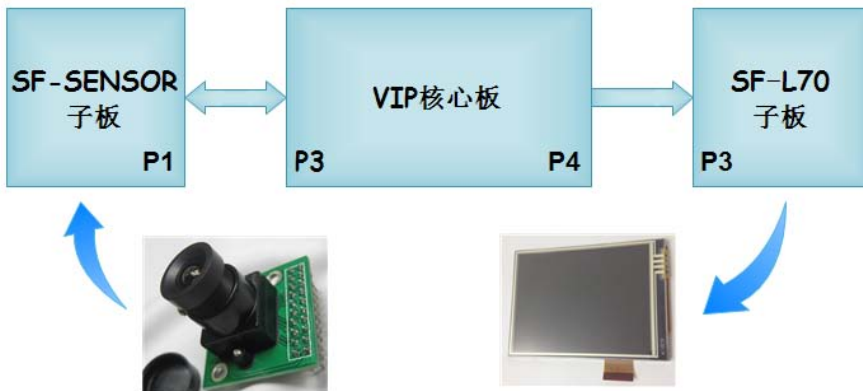


工程的模块层次如图所示。

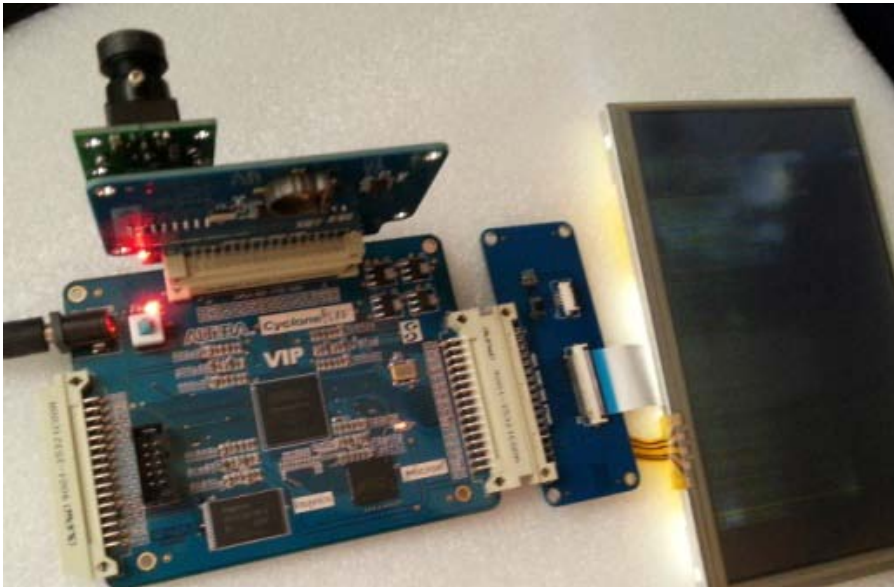


装配说明

MT9D111 摄像头模块是单独的配件，需要配合 SF-VIP 核心板、SF-SENSOR 板（MT9D111）和 SF-L70 板完成实验。3 个板子的连接示意图所示。

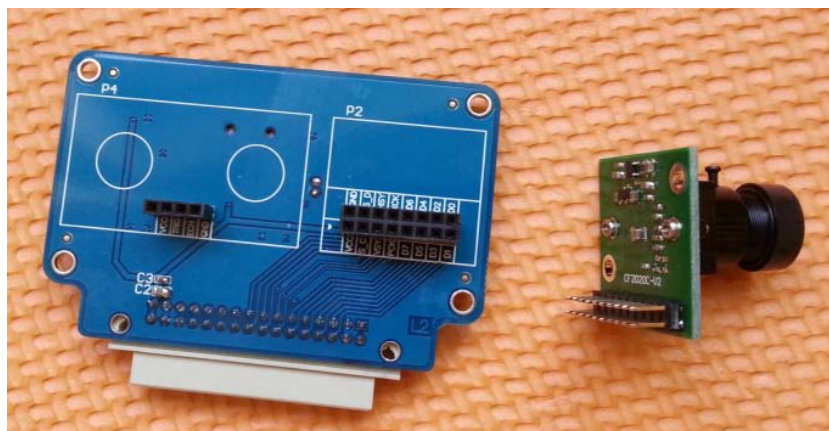
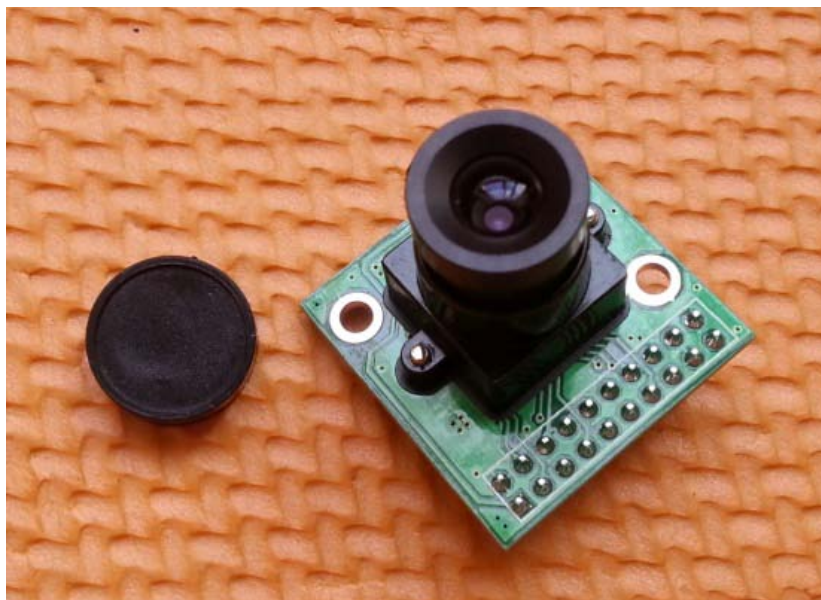


整机安装如图所示。

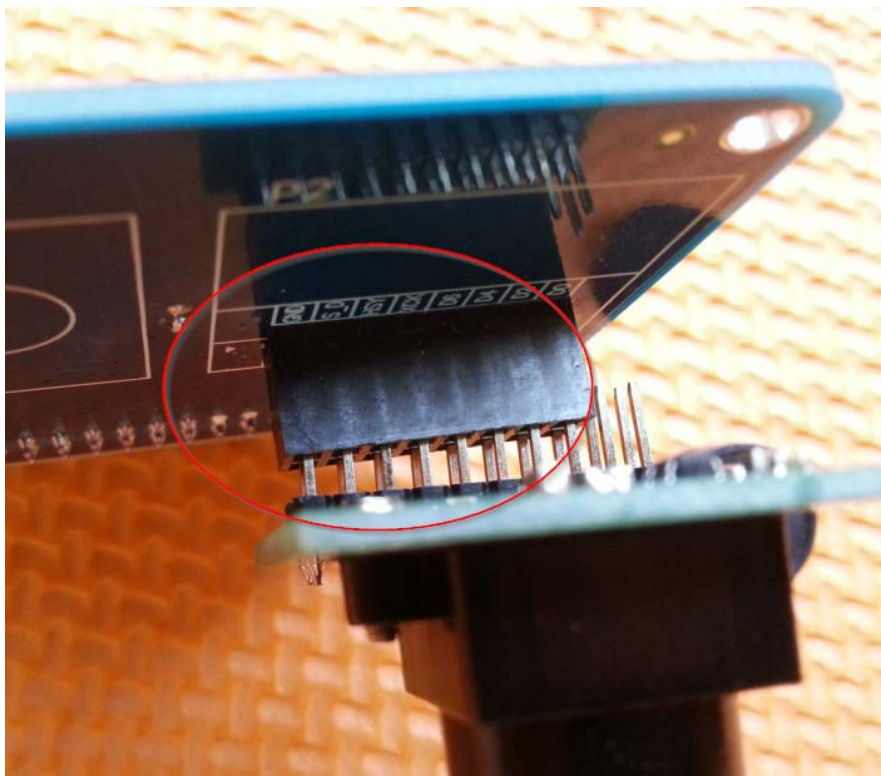


SF-SENSOR 标配的是 OV7670 摄像头模块，我们需要将它用 MT9D111 模块替换。由于 MT9D111 模块的插座是 20PIN 的，而 SF-SENSOR 板的 P2 插座是 16PIN 的，装配时需要注意，不要误差。电气特性上看 20PIN 多出来的 4PIN 完全可以不用接，不影响功能实现。

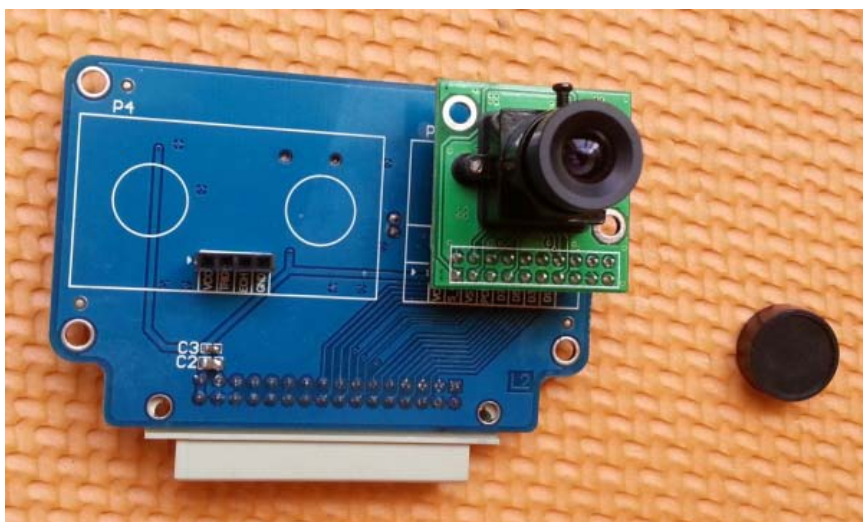
MT9D111 模块如图所示。



如图所示, 装配的时候, 注意摄像头的 PIN1 和 SF-SENSOR 板的 PIN1 对其, 右侧 4 个 PIN 悬空就行。



装配完成, 如图所示。



引脚分配

《圣经》箴言九 11“敬畏耶和华是智慧的开端, 认识至胜者便是聪明。”



SF-VIP1 核心板的 OUTPPN 插座 P3 连接到 SF-SENSOR 子板的 P1。其详细引脚定义如下表所示。

信号名	SF-SENSOR 板定义	SF-VIP1 板定义	FPGA 引脚定义
vpclk	P1-14	P3-14	B10
vvsync	P1-16	P3-16	B11
vhref	P1-15	P3-15	A11
vdb[0]	P1-5	P3-5	M2
vdb[1]	P1-6	P3-6	E1
vdb[2]	P1-7	P3-7	B7
vdb[3]	P1-8	P3-8	A7
vdb[4]	P1-9	P3-9	B8
vdb[5]	P1-10	P3-10	A8
vdb[6]	P1-11	P3-11	A9
vdb[7]	P1-12	P3-12	B9
vxclk	P1-13	P3-13	A10
vscl	P1-18	P3-18	E8
vsda	P1-17	P3-17	C6

LCD 显示驱动部分的 FPGA 引脚分配 tcl 脚本如下所示。

```
set_location_assignment PIN_B9 -to vdb[7]
set_location_assignment PIN_A9 -to vdb[6]
set_location_assignment PIN_A8 -to vdb[5]
set_location_assignment PIN_B8 -to vdb[4]
set_location_assignment PIN_A7 -to vdb[3]
set_location_assignment PIN_B7 -to vdb[2]
set_location_assignment PIN_E1 -to vdb[1]
set_location_assignment PIN_M2 -to vdb[0]
```

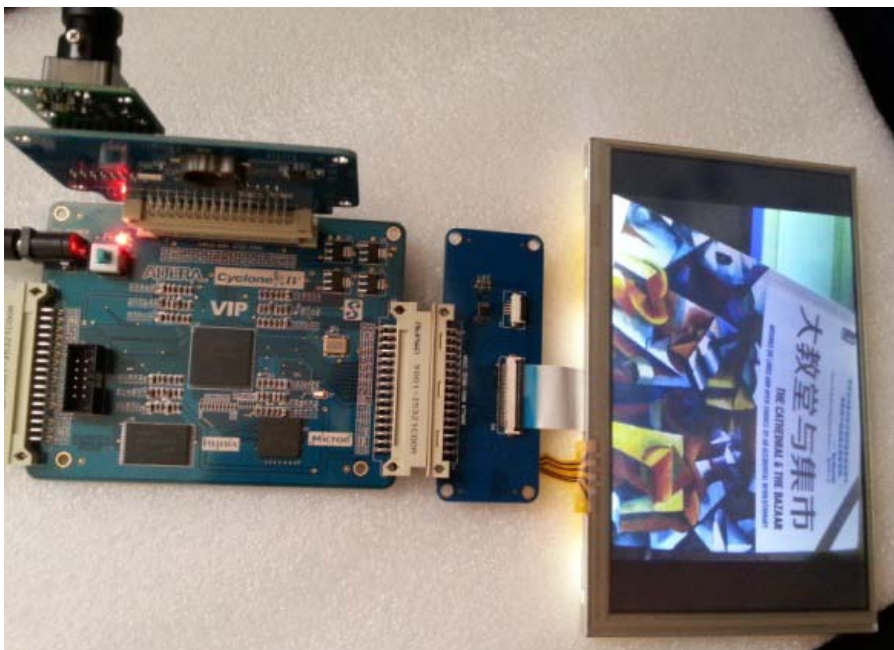
《圣经》箴言九 11“敬畏耶和华是智慧的开端，认识至胜者便是聪明。”

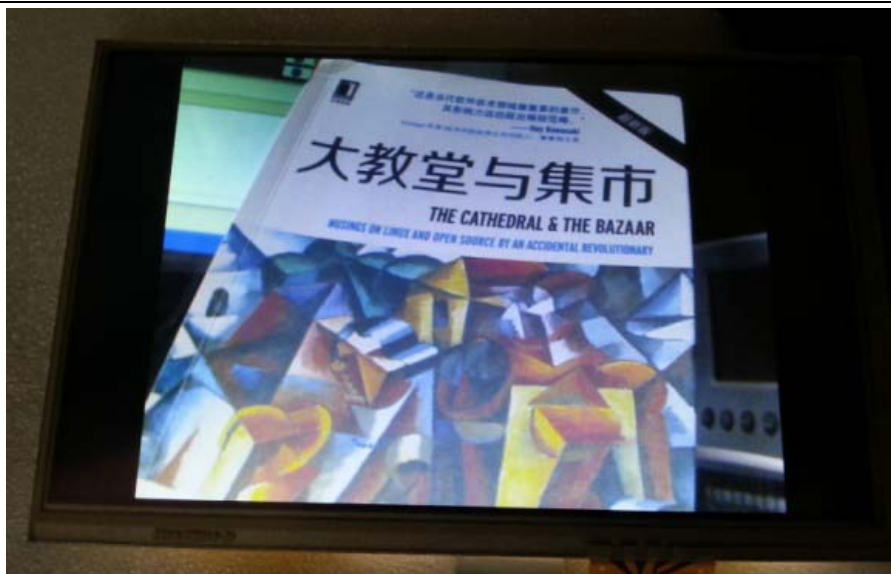


```
set_location_assignment PIN_A11 -to vhref
set_location_assignment PIN_B10 -to vpclk
set_location_assignment PIN_E8 -to vscl
set_location_assignment PIN_C6 -to vsda
set_location_assignment PIN_B11 -to vvsync
set_location_assignment PIN_A10 -to vxclk
```

实验说明

1. 连接好硬件，SF-VIP1 核心板 + SF-L70 子板 + 7 寸液晶屏 + SF-SENSOR 板（MT9D111），并且给 VIP1 板上电。
2. 打开光盘目录“...\prj\vip_ex9”下的工程。
3. 使用 Programmer 将“...\prj\vip_ex9\output_files”文件夹下的 vip.sof 文件下载到 VIP 板中。
4. 此时，我们可以看到液晶屏上出现了摄像头采集的图像。调整摄像头模块前端的镜头可以实现调焦。







2.10 特权 VIP 例程 10—AV(PAL/NTSC)视频采集显示

功能简介

PAL/NTSC 是常用的电视信号制式, 它一模拟信号的方式传输, 因其只需要一条传输线缆, 而且传输距离远, 可谓简单实用, 因此在广播电视、安防监控等各种行业中广泛应用。

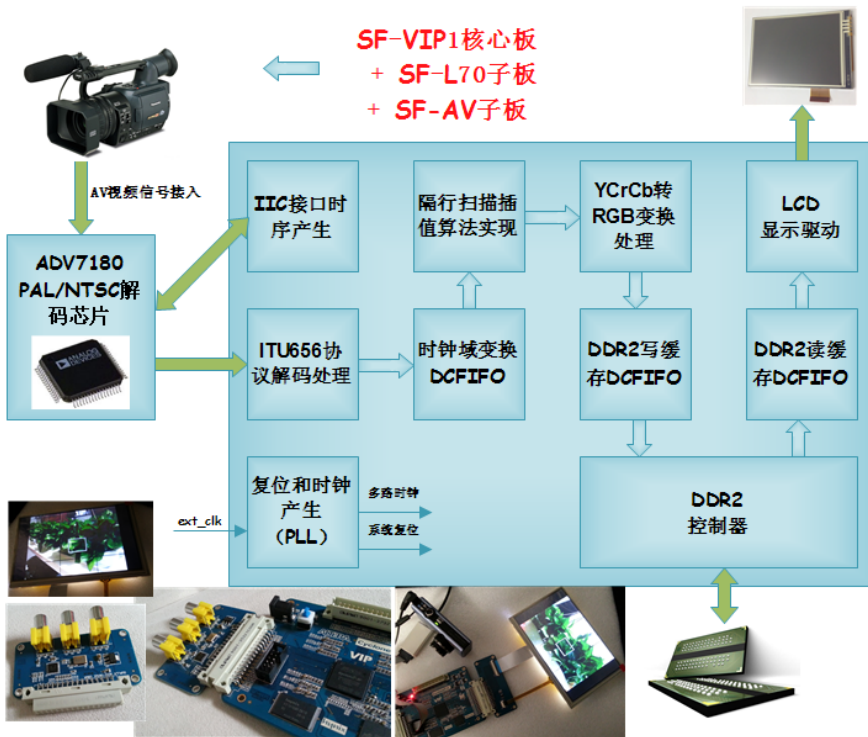
关于 PAL/NTSC 的更多介绍, 大家可以参考百度百科。

<http://baike.baidu.com/subview/66497/7912471.htm?fr=aladdin>

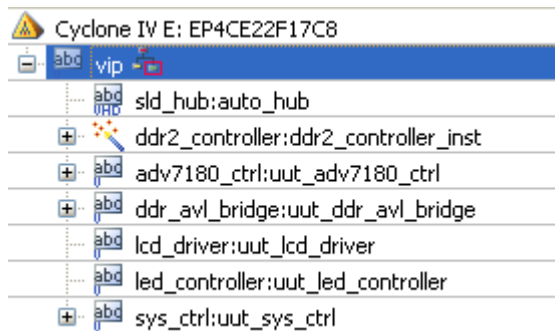
<http://baike.baidu.com/view/66463.htm>

如图所示, 本实验工程非常接近实际应用。一台具有 AV 信号输出的摄像机连接到我们的 SF-AV 子板上的 RCA 端子。SF-AV 子板上的 AV 解码芯片 ADV7180 能够将模拟信号转换为 ITU656 协议标准的数字总线信号输出。FPGA 内部接收到 ITU656 标准的数据后, 首先进行解码处理, 将有效视频数据提取出来, FPGA 内部的主要功能如下:

- IIC 接口实现, 用于初始化或控制 ADV7180
- ITU656 协议解码, 提取有效视频图像数据
- 对隔行的 AV 视频信号进行插值, 实现逐行显示
- 将 YCrCb 格式图像转换为 RGB 图像
- 使用 DCFIFO 对视频流进行缓存和时钟域的变换
- 视频流缓存到 DDR2 中
- 视频流显示在 7 寸 800*480 的液晶屏上



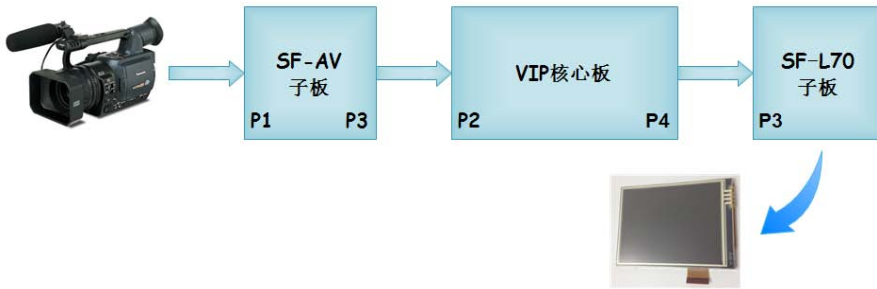
该工程的层次结构如图所示。



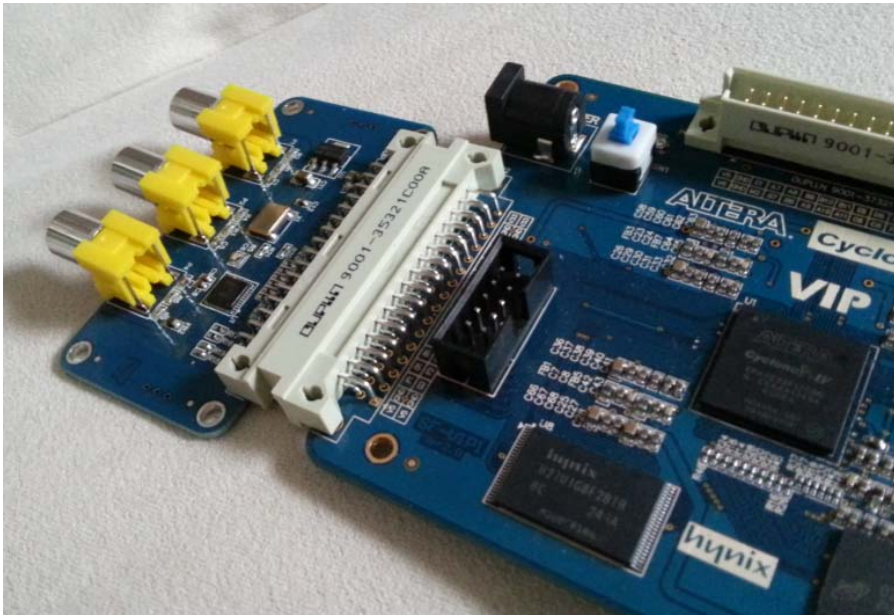
装配说明

本实验用到三个板子，SF-VIP 核心板+SF-L70 液晶屏子板+SF-AV 子板。
 SF-AV 子板的 OUTPPN 插座 P3 连接到 VIP 核心板的 P2，SF-L70 子板的 P3 连接到 VIP 核心板的 P4。AV 摄像头的 RCA 端子连接到 P1 插座。

《圣经》箴言九 11“敬畏耶和华是智慧的开端，认识至胜者便是聪明。”



SF-VIP 核心板和 SF-AV 子板的连接如图所示。



引脚分配

ADV7180 与 FPGA 之间的接口引脚分配如下所示。

```
set_location_assignment PIN_C2 -to adv7180_db[7]
set_location_assignment PIN_F3 -to adv7180_db[6]
set_location_assignment PIN_G5 -to adv7180_db[5]
set_location_assignment PIN_D3 -to adv7180_db[4]
set_location_assignment PIN_E6 -to adv7180_db[3]
```



```
set_location_assignment PIN_C3 -to adv7180_db[2]
set_location_assignment PIN_D6 -to adv7180_db[1]
set_location_assignment PIN_A2 -to adv7180_db[0]
set_location_assignment PIN_G2 -to adv7180_sda
set_location_assignment PIN_D1 -to adv7180_scl
set_location_assignment PIN_F1 -to adv7180_rst_n
set_location_assignment PIN_M1 -to adv7180_vclk
```

实验说明

1. 连接好硬件，SF-VIP1 核心板 + SF-L70 子板 + 7 寸液晶屏 + SF-AV 板，此外，连接好一个能够输出 PAL 或 NTSC 制式的 AV 信号摄像头，并且给 VIP1 板上电。
2. 打开光盘目录 “...\prj\vip_ex10” 下的工程。
3. 使用 Programmer 将 “...\prj\vip_ex10\output_files” 文件夹下的 vip.sof 文件下载到 VIP 板中。
4. 我们首先使用 Quartus II 的在线调试工具 In-System Sources and Probes Editor 读取 ADV7180 的各个状态寄存器。

点击菜单栏的 Tools→In-System Sources and Probes Editor，通过其右侧的 JTAG Chain Configuration 对 VIP 板进行下载烧录配置。

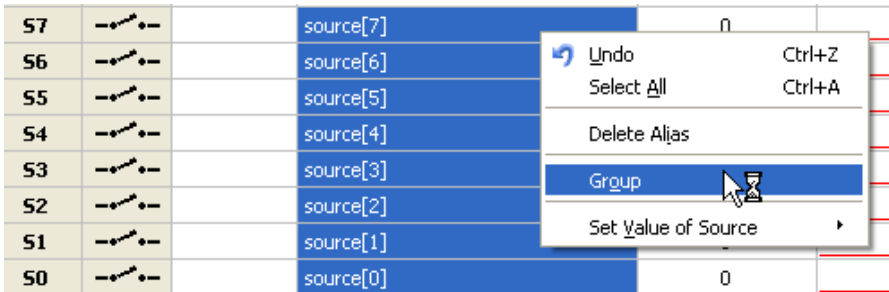
接着来到主窗口中，将显示的 source 和 probe 进行分组。在代码模块 adv7180_ctrl.v 中，我们做了如下的接口映射，在 source 和 probe 的分组中，我们将分别获得相应的接口信号。

```
//需要通过 IIC 接口配置 ADV7180 使能信号，高电平有效
wire tiic_en;
//需要通过 IIC 接口配置 ADV7180 的读或写选择，1—写，0—读
wire tiic_wr_rdn;
//需要通过 IIC 接口配置 ADV7180 地址
```

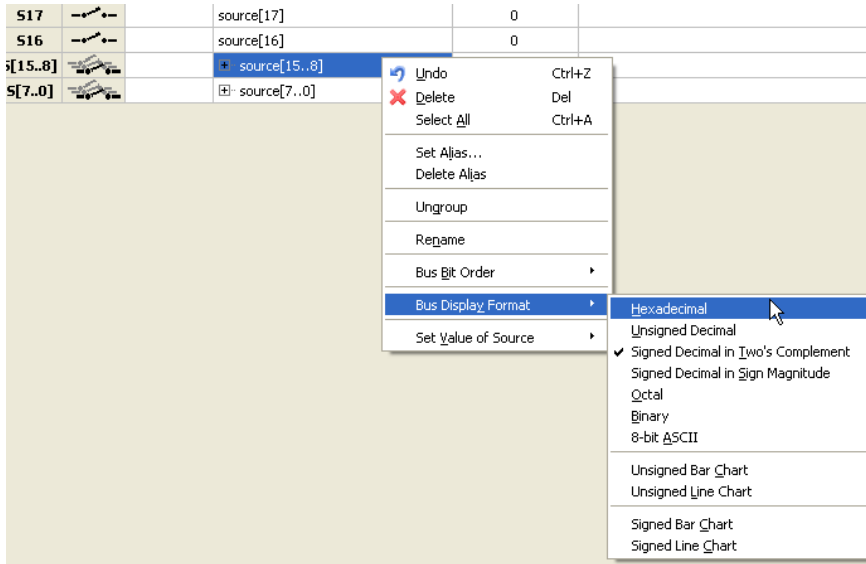


```
wire[7:0] tiic_ab;  
//需要通过 IIC 接口配置 ADV7180 的写入数据  
wire[7:0] tiic_wrdb;  
//需要通过 IIC 接口配置 ADV7180 的读出数据  
wire[7:0] tiic_rddb;  
  
//例化 In-System Sources and Probes  
tviic_debug tviic_debug_inst (  
    .probe ( tiic_rddb ),  
    .source ( {tiic_en, tiic_wr_rdn, tiic_ab, tiic_wrdb} )  
);
```

分组方式如图所示, 例如我们需要将 source 的低 8 位, 级 tiic_wrdb 总线分组, 只要选择他们, 然后单击右键, 在弹出菜单中选择 Group 选项即可。其他信号以同样的方式分组。



如果想切换显示数据的格式, 如图所示, 可以选择信号组后, 点击右键菜单, 找到 Bus Display Format 选项并选择相应的格式。



查看 ADV7180 的 datasheet, 我们可以如下所示, 分别读取一些主要的状态寄存器。

名称: 芯片 ID 版本寄存器

地址: 0x11

功能:

IDENT[7:0]	Description
0x1B	Initial release silicon
0x1C	Improved ESD and PDC fix

读取数据:

Name	Data	-8	-7	-6	-5	-4	-3	-2
* probe[7..0]	18h							18h
source[17]	1							
source[16]	0							
source[15..8]	11h							11h
source[7..0]	00h							00h

名称: 状态寄存器 1

地址: 0x10

功能:



Status 1[7:0]	Bit Name	Description
0	IN_LOCK	In lock (now)
1	LOST_LOCK	Lost lock (since last read of this register)
2	FSC_LOCK	f _{sc} locked (now)
3	FOLLOW_PW	AGC follows peak white algorithm
4	AD_RESULT[0]	Result of autodetection
5	AD_RESULT[1]	Result of autodetection
6	AD_RESULT[2]	Result of autodetection
7	COL_KILL	Color kill active

AD_RESULT[2:0]	Description
000	NTSM M/J
001	NTSC 4.43
010	PAL M
011	PAL 60
100	PAL B/G/H/I/D
101	SECAM
110	PAL Combination N
111	SECAM 525

读取数据:

Name	Data	-8	-7	-6	-5	-4	-3
probe[7..0]	4Fh	4Fh					
source[17]	1						
source[16]	0						
source[15..8]	10h	10h					
source[7..0]	00h	00h					

名称: 状态寄存器 2

地址: 0x12

功能:



Status 2[7:0]	Bit Name	Description
0	MVCS DET	Detected Macrovision color striping
1	MVCS T3	Macrovision color striping protection; conforms to Type 3 if high, Type 2 if low
2	MV PS DET	Detected Macrovision pseudosync pulses
3	MV AGC DET	Detected Macrovision AGC pulses
4	LL NSTD	Line length is nonstandard
5	FSC NSTD	f _{sc} frequency is nonstandard
6	Reserved	
7	Reserved	

读取数据:

Name	Data	-8	-7	-6	-5	-4	-3
probe[7..0]	00h						00h
source[17]	1						
source[16]	0						
source[15..8]	12h						12h
source[7..0]	00h						00h

名称: 状态寄存器 3

地址: 0x13

功能:



Status 3[7:0]	Bit Name	Description
0	INST_HLOCK	Horizontal lock indicator (instantaneous)
1	GEMD	Gemstar detect
2	SD_OP_50Hz	Flags whether 50 Hz or 60 Hz is present at output
3	Reserved	Reserved for future use
4	FREE_RUN_ACT	ADV7180 outputs a blue screen (see the DEF_VAL_EN, Default Value Enable, Address 0x0C[0] section)
5	STD FLD LEN	Field length is correct for currently selected video standard
6	Interlaced	Interlaced video detected (field sequence found)
7	PAL_SW_LOCK	Reliable sequence of swinging bursts detected

读取数据:

Name	Data	-8	-7	-6	-5	-4	-3
probe[7..0]	EDh	EDh					
source[17]	1						
source[16]	0						
source[15..8]	13h	13h					
source[7..0]	00h	00h					

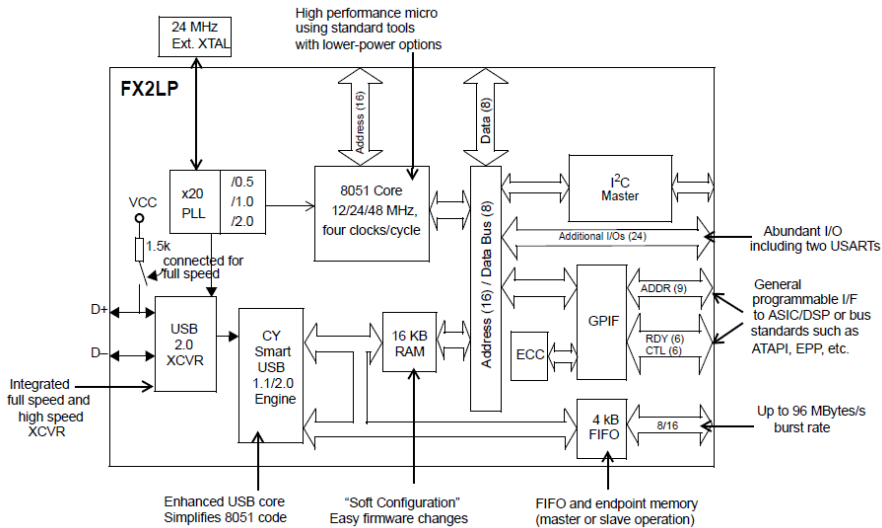


2.11 特权 VIP 例程 11—FX2 硬件和驱动安装 以及 Bulkloop 实验

功能简介

FX2 (CY7C68013) 是一款集成 8051 单片机的灵活的 USB2.0 控制器, 其带宽可以接近 USB2.0 标称的 480Mbps (通常我们使用 FPGA 的 SlaveFIFO 接口进行数据传输可以达到 400Mbps 的带宽)。

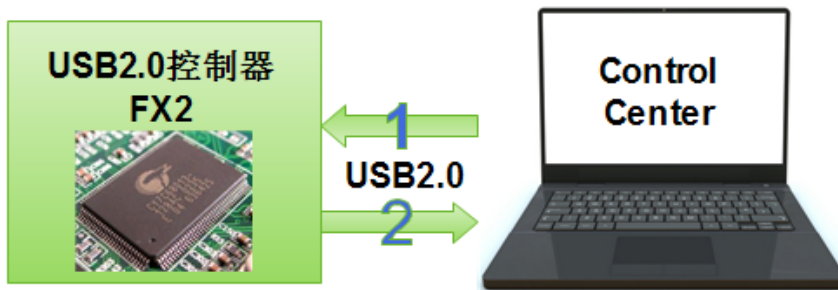
FX2 的内部功能框图如图所示。



我们的系统中, PC 和 SF-VIP FPGA 核心板通过 SF-FX2 外设子板进行连接, 完成 FPGA 和 PC 之间的数据通信。FPGA 和 FX2 之间使用一组灵活的 SlaveFIFO 接口进行数据交互, FPGA 掌握着数据传输的主动权, 只需要配置好 FX2 的传输模式即可。

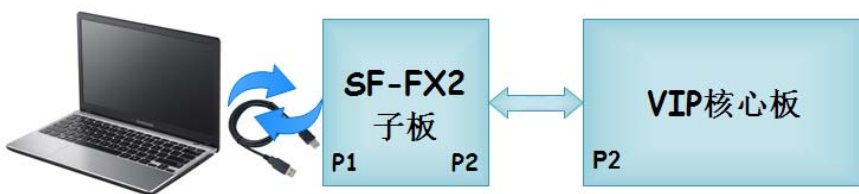


本例程是 FX2 外设子板的第一个实验。我们先不动用 FPGA，而是直接使用 Cypress 官方给出的 bulkloop 例程，PC 端发送一组数据给 FX2，FX2 内部自动将其返回给 PC，整个过程 FX2 自动完成，无需 FPGA 参与。



装配说明

转配连接示意如图所示。



驱动安装

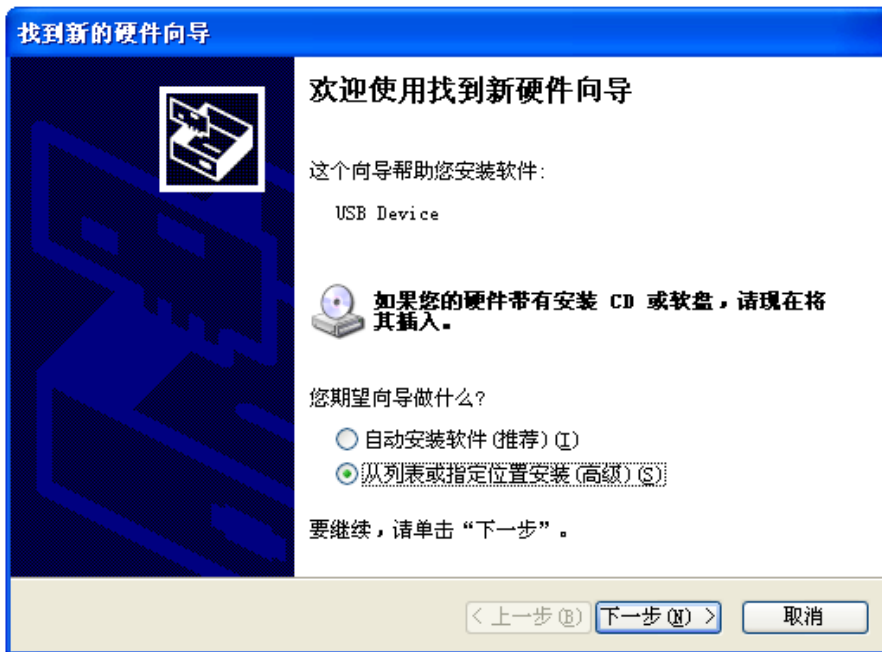
1. 在 FX2 配套光盘下的 prj\vip_ex11 文件夹中双击安装“NetFx20SP1_x86.exe”。
2. 在 FX2 配套光盘下的 prj\vip_ex11 文件夹中双击安装“CY3684Setup.exe”。



建议一路 Next, 默认安装到 C 盘中。FX2 用到的几个软件工具都会同时打包安装上去。安装完毕, 我们可以看到“开始→程序”菜单中多了 Cypress 的多个软件工具。



3. 安装装配说明连接 SF-VIP 核心板和 SF-FX2 子板, USB 线连接好 PC 和 FX2, 给板子上电。
4. 第一次连接 FX2 和 PC, 会弹出 USB 外设驱动安装的提示, 如图, 选择“从列表或指定位置安装”。

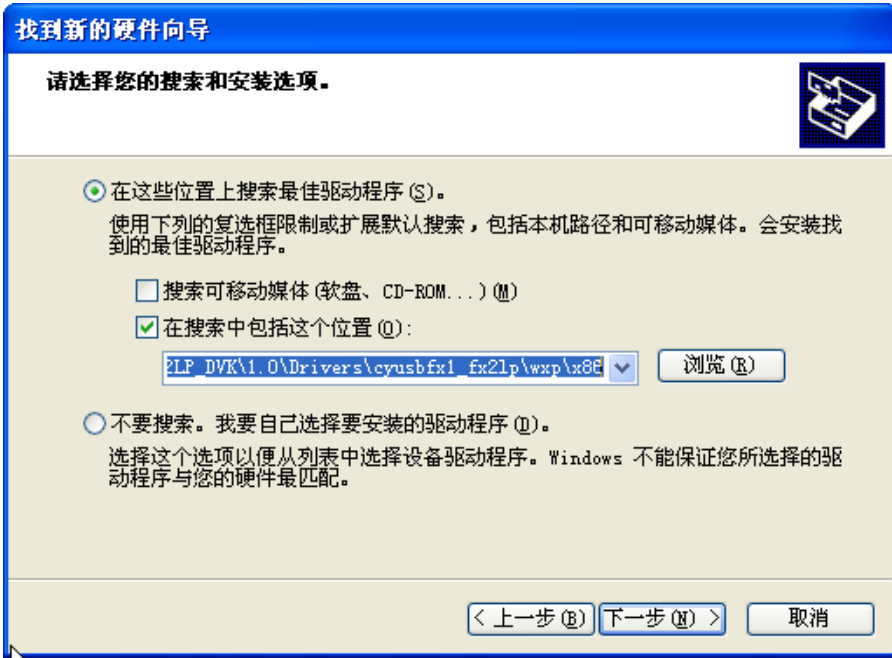


如图, 将驱动安装文件夹定位到:

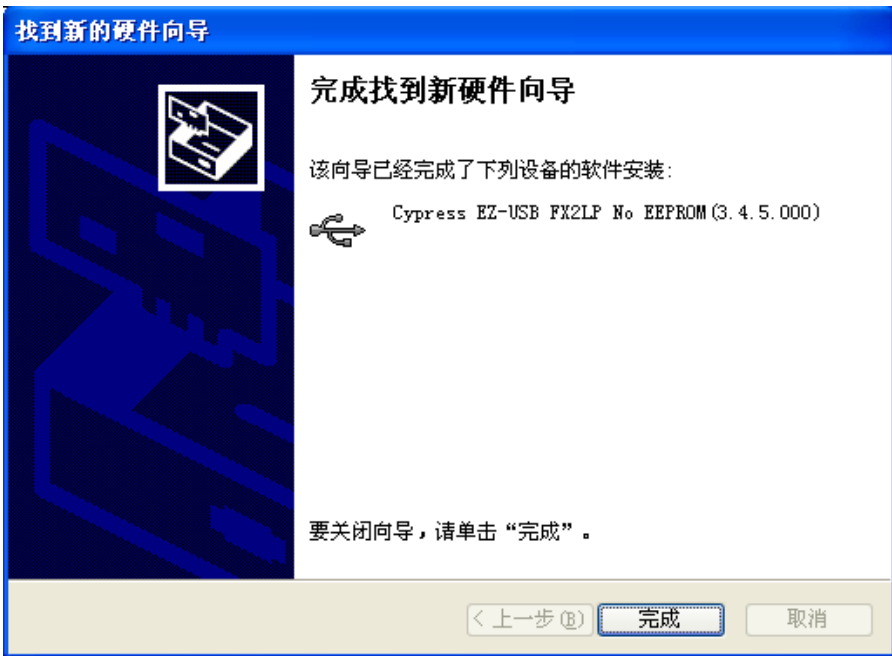
C:\Cypress\USB\CY3684_EZ-USB_FX2LP_DVK\1.0\Drivers\cyusbfx1_fx2lp\wpx
\x86



注意, 在 cyusbfx1_fx2lp 文件夹下有 win7\vista\XP 的驱动, 大家根据实际的电脑系统进行选择。本实例是 XP。



安装完驱动, 如图所示。





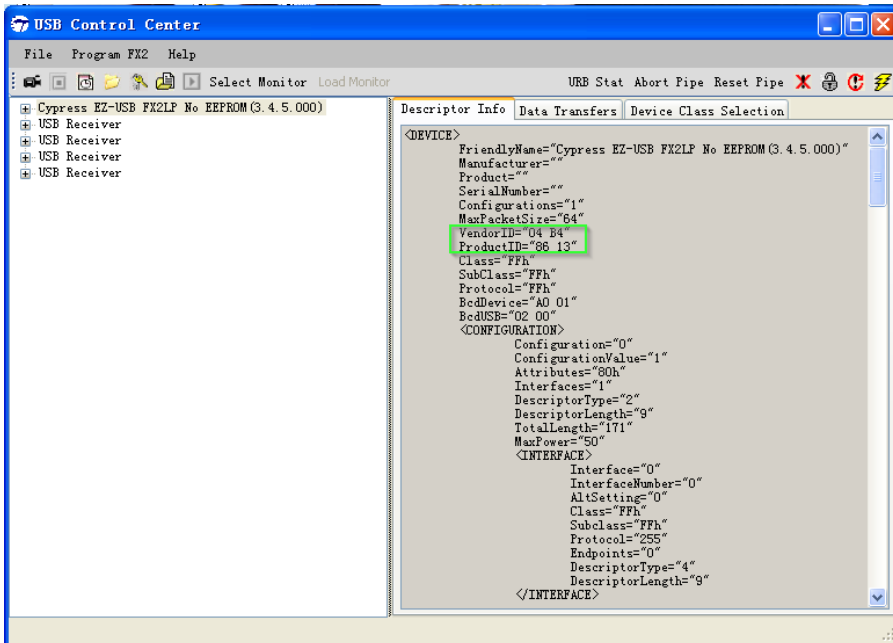
我们也可以打开设备管理器, 如图所示, 这里也认识到了新设备 FX2。



Bulkloop 实验

这里, 我们要使用一个现有的 bulkloop 例程来进行基本的 FX2 收发通信, 该例程的详细说明和操作, 大家也可以参考光盘中 pdf\AN66806 - Getting Started with EZ-USB® FX2LP™ GPIF 文件夹下的文档 AN66806 CN.pdf。这里将实验操作步骤简单的说明

1. 点击“开始→程序→Cypress→Cypress Suite USB 3.4.7→Control Center”。
- 打开后, 我们看到立即识别了我们的 FX2 设备, Descriptor Info 中也列出了 VID 和 PID, 分别为 0x04B4 和 0x8613。

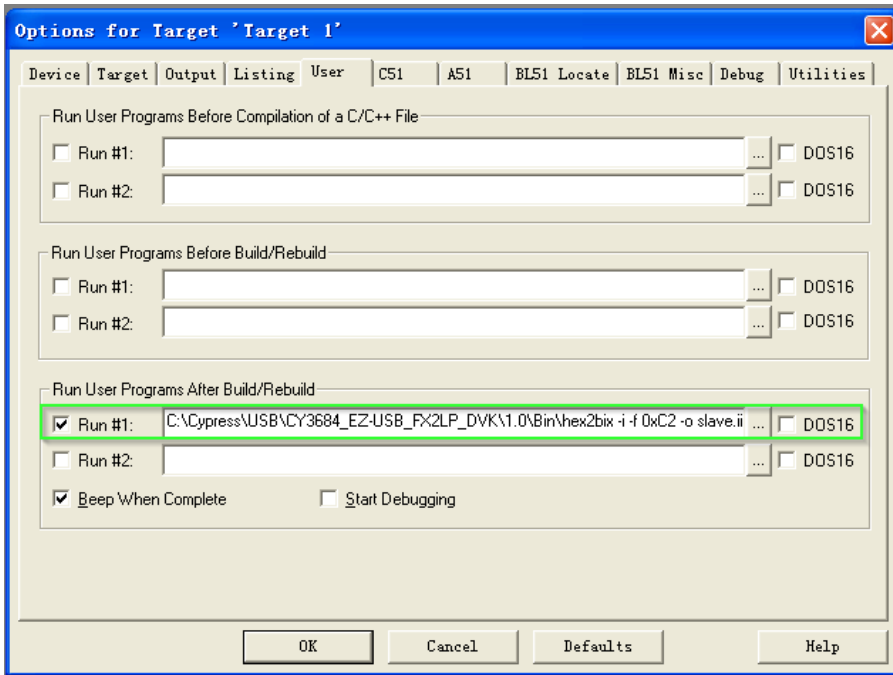


《圣经》箴言九 11“敬畏耶和华是智慧的开端, 认识至胜者便是聪明。”



2. 我们点击 USB Control Center 的菜单“Program FX2→RAM”，定位到文件夹“C:\Cypress\Cypress Suite USB 3.4.7\Firmware\Bulkloop”下的 bulkloop.hex 文件，将其选中烧录到 FX2 中。注意这里选择了 RAM，是在线烧录，而我们的 SF-FX2 板载了一颗 EEPROM，大家若选择下载到 Small EEPROM 选项（对应的下载文件后缀为.iic），则重新上电后，EEPROM 中的程序将被执行。

这里顺便提一下.iic 文件的生成。在 Keil 中打开 firmware 工程后，进入“Target Options ...”。如图所示，在 User 选项卡中勾选“Run User Programs After Build/Rebuild”下的“Run #1”。



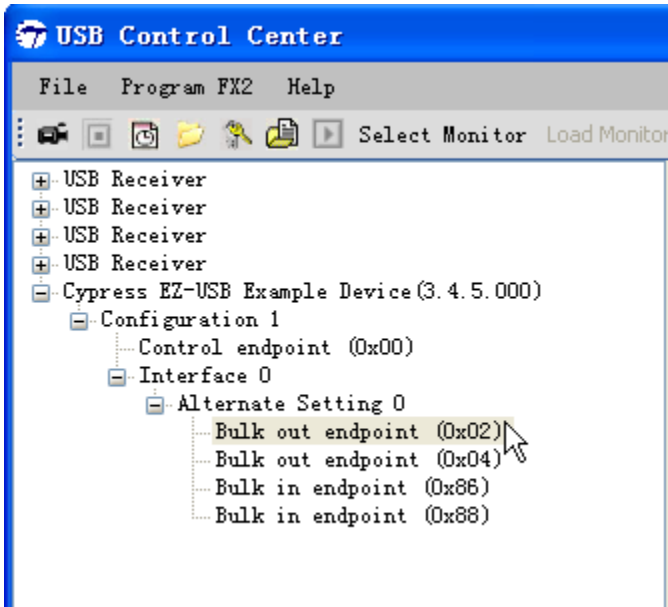
路径和脚本如下：

```
C:\Cypress\USB\CY3684_EZ-USB_FX2LP_DVK\1.0\Bin\hex2bix -i -f 0xC2 -o slave.iic slave.hex
```

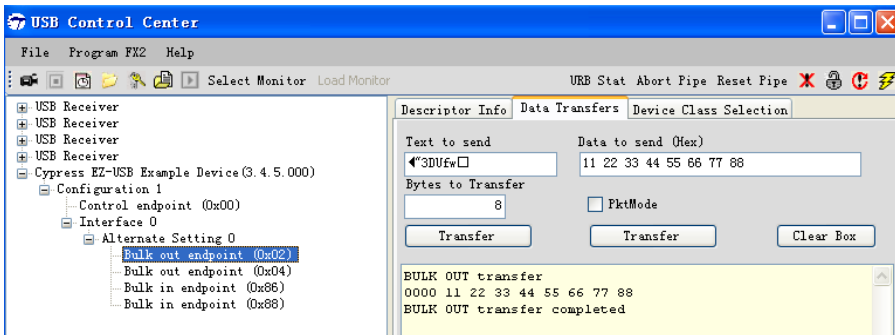
3. 如图，此时 USB Control Center 中识别到的设备变成了“Cypress EZ-USB Example Device”，展开后有 4 个可用端点。这里 0x02 和 0x86 是一对《圣经》箴言九 11“敬畏耶和华是智慧的开端，认识至胜者便是聪明。”



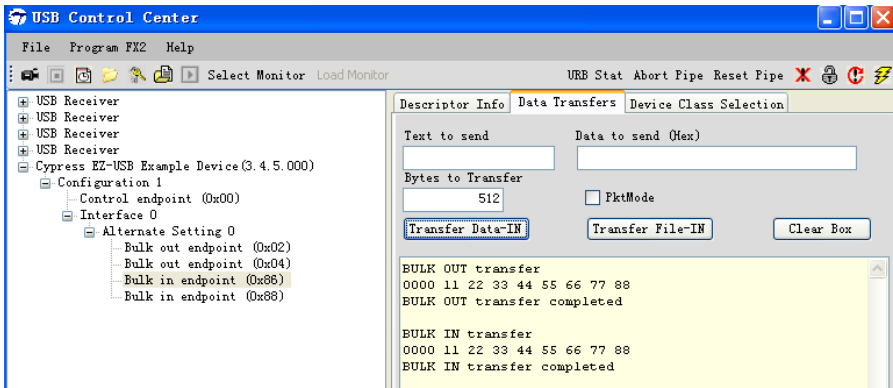
收发端口, 0x04 和 0x88 也是一对。



4. 这里我们来做个简单的测试。如图所示, 选中端口 0x02 后, 在“Byte to send”下发输入“11 22 33 44 55 66 77 88”这 8 个 Byte 数据, 然后点击“Transfer”, 随后打印出了数据发送出去的信息。



接下来, 我们如图所示, 选中端口 0x86, 点击“Transfer Data-In”, 随后在刚才打印发送数据的下方, 出现了接收到的数据信息, 和所发送的数据完全一致。若使用 0x04 和 0x88 端口也是一样的, 大家可以试试。而在没有先发送就想接收的情况下, 是收不到任何数据的。





2.12 特权 VIP 例程 12—基于 FX2 和 FPGA 的 Loopback/StreamIN/StreamOUT 实例

功能简介

本实例有 4 个工程，一个 8051 MCU 的工程，位于光盘\prj\vip_ex12\fx2_firmware 文件夹下。另外 3 个都是 FPGA 工程，分别位于 \prj\vip_ex12\fx2_loopback、\prj\vip_ex12\fx2_StreamIN 和 \prj\vip_ex12\fx2_StreamOUT 文件夹下。

fx2_loopback 工程主要是 PC 端发送数据到 FX2，FX2 通过指示位 flaga 告知 FPGA 有数据待读取，FPGA 端便通过 SlaveFIFO 接口读取 PC 端发送过来的数据缓存到 FPGA 内部的 FIFO 中，FPGA 在完成读取操作后，发起一次 SlaveFIFO 的写入操作，将接收到的数据通过 FX2 最终返回到 PC 端。整个数据的收发过程，我们在 FPGA 内部可以通过在线逻辑分析仪 SignalTap II 抓取 SlaveFIFO 接口的所有信号进行查看。



fx2_StreamIN 工程执行的操作比 loopback 要简单得多，FPGA 端只要读取数据写入满标志位 flagb 一直处于高电平（未滿）状态，就持续的写入 256*16bit 的数据到 FX2 中等待 PC 端读取。PC 端读取一次，则 FPGA 端再次写入新的数据。整个数据的写入过程，我们在 FPGA 内部可以通过在线逻辑分析仪 SignalTap II 抓取 SlaveFIFO 接口的所有信号进行查看。

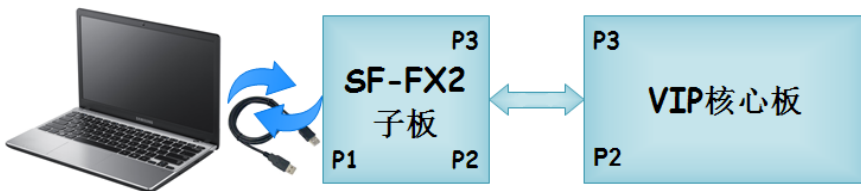


fx2_StreamOUT 工程的数据发起者是 PC 端, PC 端写入的任何数据, 都可以通过 FX2 最终传输到 FPGA 中。整个数据的读取过程, 我们在 FPGA 内部可以通过在线逻辑分析仪 SignalTap II 抓取 SlaveFIFO 接口的所有信号进行查看。同时我们在 FPGA 内部也例化了一个 onchip RAM, 将最新读取到的一帧数据缓存到 RAM 上, 同时在 Quartus II 中, 我们可以查看已经读取的更新的一帧数据。



装配说明

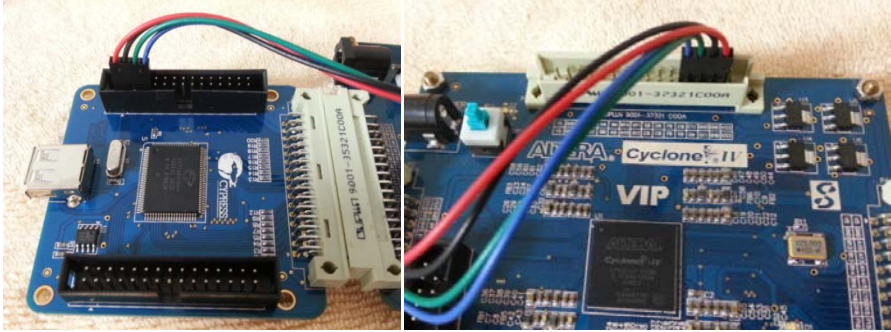
本实例需要连接 VIP 核心板和 SF-FX2 子板, 除了正常的 OUPLLN 连接器外, 还需要额外的 4PIN 杜邦线连接, 此外, SF-FX2 子板上的标准 USB-A 端口和 PC 之间也需要 USB 线连接。



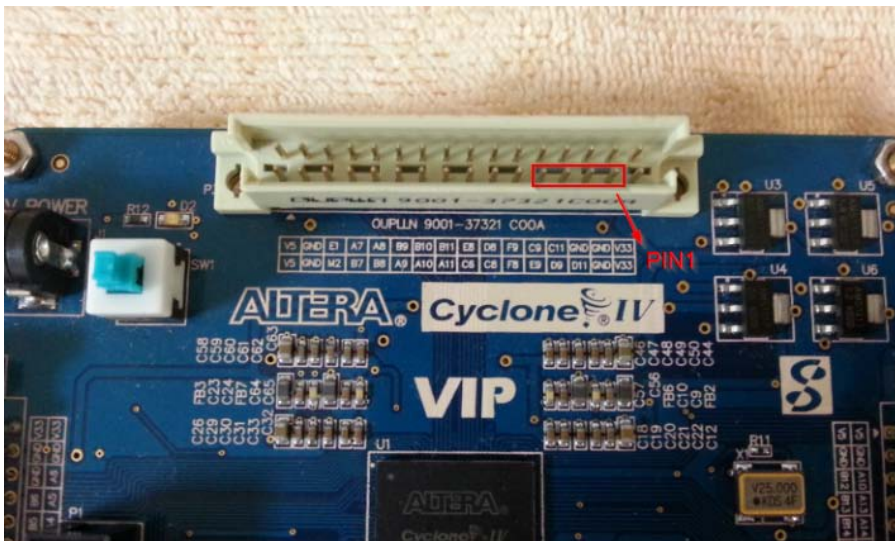
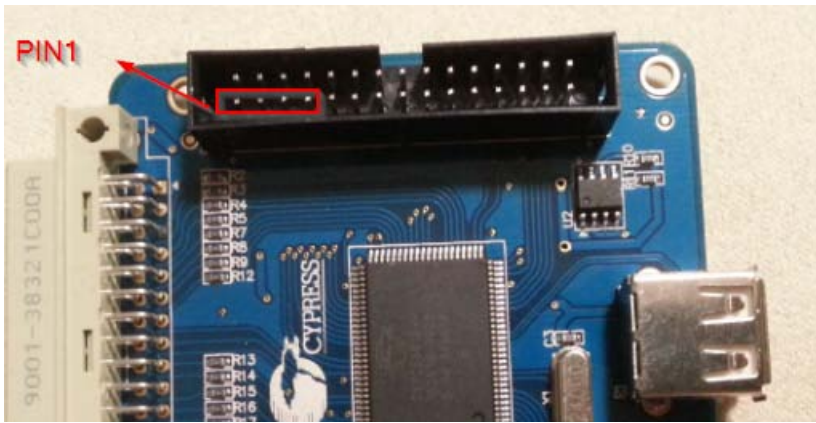
我们重点关注一下 4PIN 杜邦线的连接方式。如图所示, 4PIN 杜邦线分别连接到 VIP 核心板以及 SF-FX2 子板的连接器 P3 上。

《圣经》箴言九 11“敬畏耶和华是智慧的开端, 认识至胜者便是聪明。”

特权制造 本教程只适用于特权开发制作的各类学习套件使用
淘宝店链接: <http://myfpga.taobao.com/>



具体的连接位置如图所示。请大家特别注意 PIN1 的位置，不要连接错了。



《圣经》箴言九 11“敬畏耶和华是智慧的开端，认识至胜者便是聪明。”



引脚分配

FX2 与 FPGA 之间的接口引脚分配定义如下所示。

信号名	SF-FX2 子板连接 器	SF-VIP 核心板连 接器	FPGA 引脚
SLCLK	P2-19	P2-19	A2
SLCS#	P2-16	P2-16	C3
SLOE#	P2-14	P2-14	D3
SLRD#	P2-26	P2-26	B6
SLWR#	P2-27	P2-27	A6
SLPKTEND#	P2-15	P2-15	D5
SLFLAGA	P2-13	P2-13	E6
SLD0	P2-5	P2-5	G2
SLD1	P2-6	P2-6	F1
SLD2	P2-7	P2-7	F2
SLD3	P2-8	P2-8	D1
SLD4	P2-9	P2-9	C2
SLD5	P2-10	P2-10	B1
SLD6	P2-11	P2-11	G5
SLD7	P2-12	P2-12	F3
SLD8	P2-17	P2-17	D6
SLD9	P2-18	P2-18	E7
SLD10	P2-20	P2-20	B3
SLD11	P2-21	P2-21	A3
SLD12	P2-22	P2-22	B4

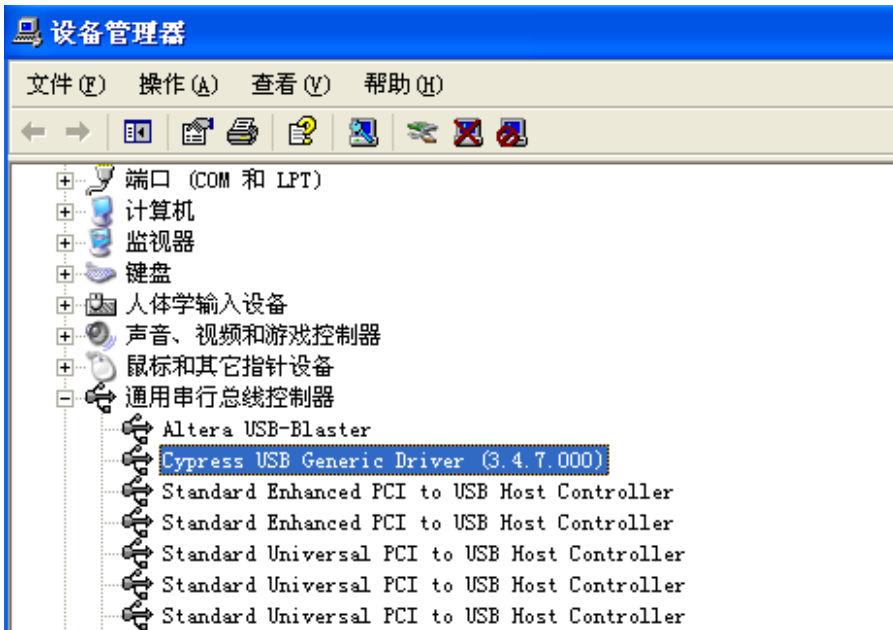


SLD13	P2-23	P2-23	A4
SLD14	P2-24	P2-24	B5
SLD15	P2-25	P2-25	A5
SLA0	P3-5	P3-25	D9
SLA1	P3-3	P3-27	D11
SLFLAGB	P3-7	P3-23	E9
SLFLAGC	P3-9	P3-21	F8

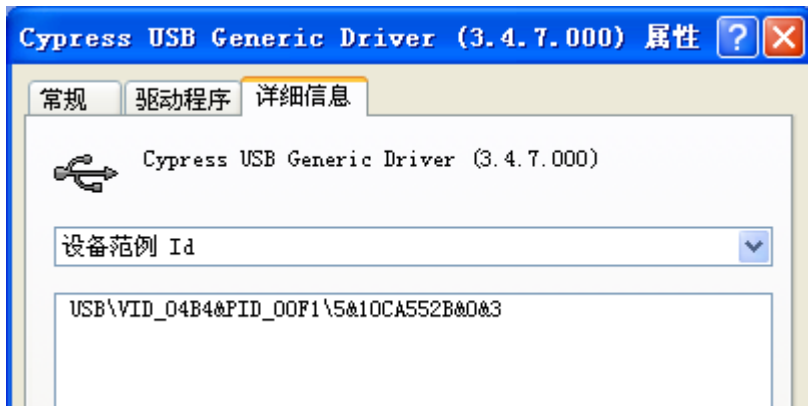
Loopback 实验说明

1. 安装装配说明连接好各个功能模块，给板子上电。
2. 参照上一节在 USB Control Center 中将光盘中的 “...\prj\
vip_ex12\fx2_firmware\slave.hex” 文件下载到 FX2 的 RAM 中。
3. 通常在我们下载完 slave.hex 文件后，会弹出对话框提示我们安装 USB 设备的硬件驱动，或者即便没有弹出驱动安装提示，我们也可以到设备管理器中查看一下是否有未安装驱动的设备。

如图所示，这里我们的新设备已经失败到了，若未识别，则会出现绿色的叹号。



假设这里还没有识别到硬件驱动, 我们可以打开这个设备的属性, 选择“详细信息”一栏, 这里可以看到“设备范例 Id”下有 VID_04B4&PID_00F1 的信息, 记住这两个号码, 分别代表供应商 ID 和产品 ID。



随后我们可以来到文件夹

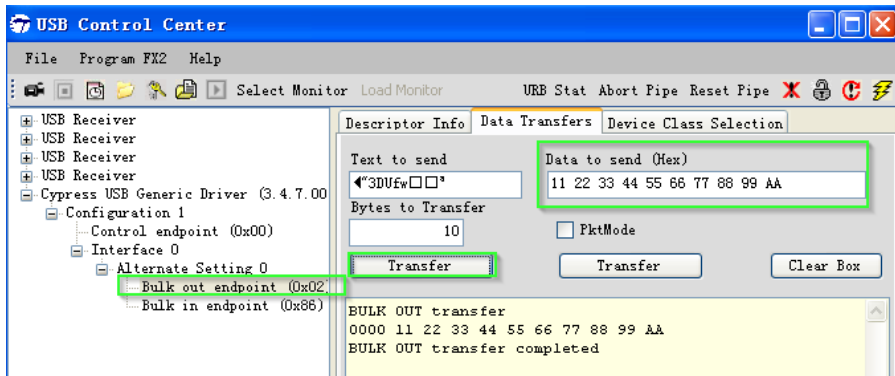
“C:\Cypress\USB\CY3684_EZ-USB_FX2LP_DVK\1.0\Drivers\cyusbfx1_fx2lp”
下, 找到对应的操作系统, 如本实例使用的是 32bit 的 XP 系统, 则继续定位到 “...\wxp\x86” 文件夹下, 打开 cyusbfx1_fx2lp.inf 文件, 将以下 9 处



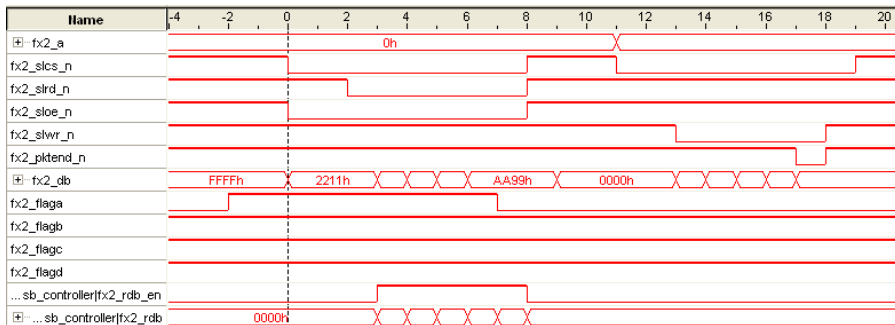
的“VID_04B4&PID_8613”替换为前面刚刚获得的“VID_04B4&PID_00F1”。

接着右键单击未识别驱动的设备，将驱动文件夹路径定位到刚刚修改了 VID&PID 的文件路径下完成驱动安装。

4. 将“\prj\ vip_ex12\fx2_loopback\output_files\vip.sof”文件下载到 VIP 板的 FPGA 中。
5. 如图所示，在 USB Control Center 中，首先选择“Bulk out endpoint (0x02)”，在“Data to send”下面输入一串 16 进制数据“11 22 33 44 55 66 77 88 99 AA”，再点击“Transfer”按钮，在下面的信息栏中便会打印出刚刚发送的这串数据。



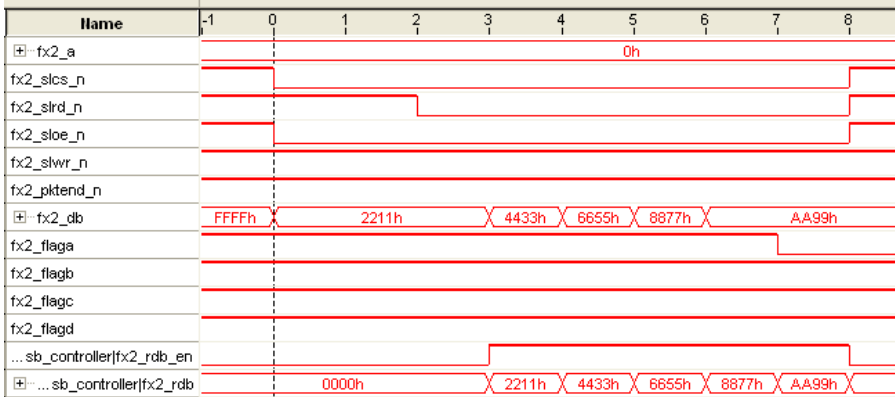
6. 在前面发送数据之前，我们便将在线逻辑分析仪 SignalTap II 设定好，当 fx2_slcs_n 下降沿便触发采样。这里我们抓到了先是 SlaveFIFO 的读时序，然后是写时序。



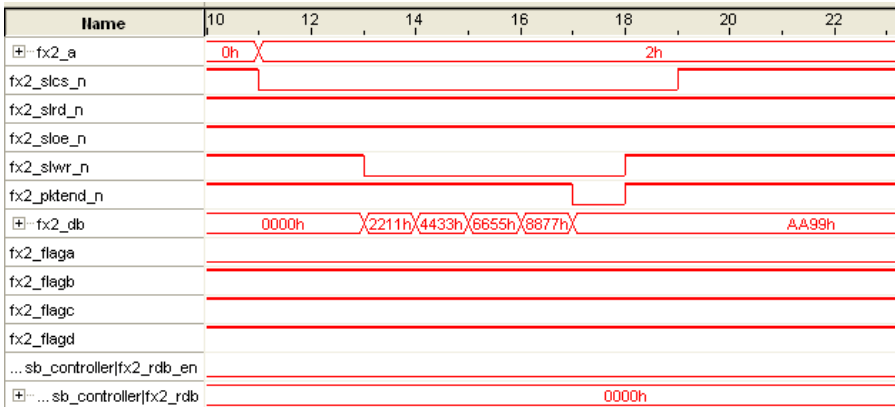
先来看读时序，随着 fx2_slcs_n/fx2_slrd_n/fx2_slwe_n 以及地址



fx2_a=2'b00, 在他们的配合下, PC 端发送的数据便出现在总线上了。FPGA 内部信号 fx2_rdb_en 拉高表示读取到了有效的数据, 对应的数据 fx2_rdb 便是我们在 PC 端发送的数据。



再来看数据的写入操作。这里对应 fx2_slwr_n 信号的拉低, 5 个 16bit 数据先后出现在了数据总线上, 并且最后一个数据对应了 fx2_pktend_n 信号的拉低 (由此表示数据发送结束)。



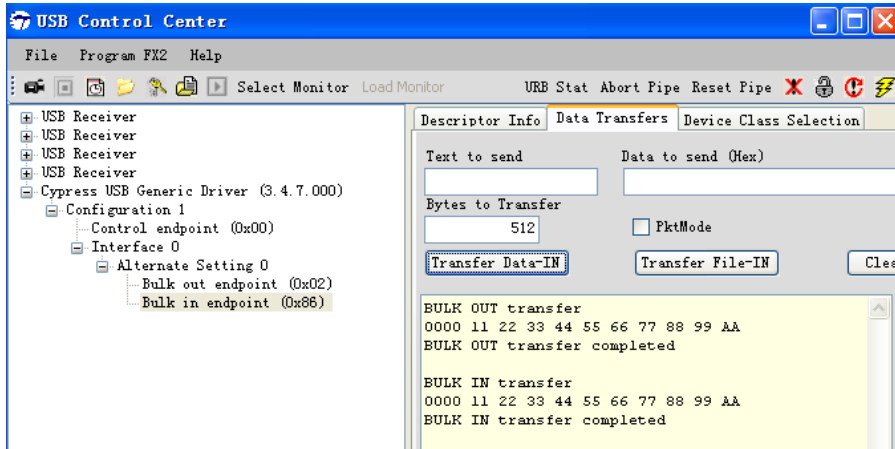
虽然 SlaveFIFO 中已经接收到了 FPGA 端要发送的数据帧, 但是这些数据并不是实时的传输到 PC 端, 因为 PC 端为主机, 它必须发起读取的动作, 那么 FX2 才会将要送给 PC 端的数据传送过去。

7. 最后, 我们回到 USB Control Center 中, 选择“Bulk in endpoint (0x86)”。

点击“Transfer Data-IN”按钮, 则回传的数据打印出来了。我们可以对《圣经》箴言九 11“敬畏耶和华是智慧的开端, 认识至胜者便是聪明。”

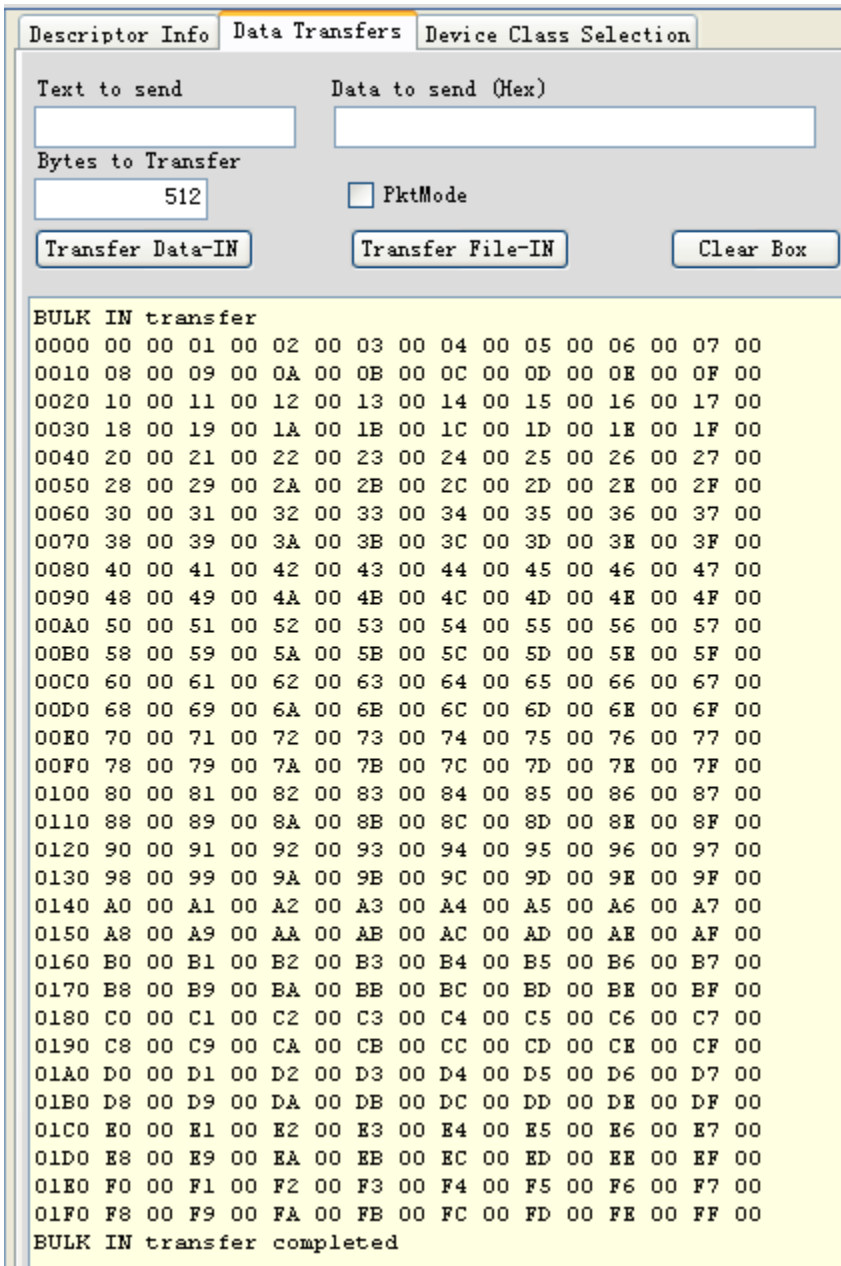


照“BULK OUT transfer”和“BULK IN transfer”的数据，完全一致。



StreamIN 实验说明

1. 安装装配说明连接好各个功能模块，给板子上电。
2. 在 USB Control Center 中将光盘中的“...\prj\vip_ex12\fx2_firmware\slave.hex”文件下载到 FX2 的 RAM 中。
3. 若需要安装驱动，参考“loopback 实验说明”。
4. 将“\prj\ vip_ex12\fx2_StreamIN\output_files\vip.sof”文件下载到 VIP 板的 FPGA 中。
5. 此时 FPGA 端已经开始往 FX2 的 SlaveFIFO 中送数据了，但是 PC 端必须读取走这些数据，否则 FPGA 端将会被“堵”住。在 PC 端，我们在 USB Control Center，选择“Bulk in endpoint (0x86)”，然后点击“Transfer Data-IN”按钮，512Byte 递增的数据便打印出来。



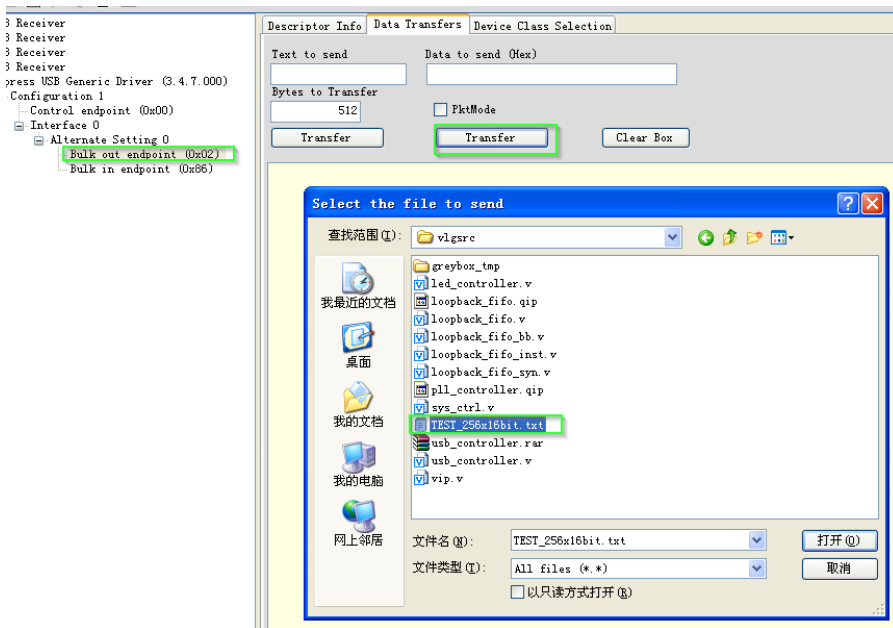
6. 大家可以打开在线逻辑分析仪 SignalTap II 并设置好触发条件, 查看相应的数据发送时序。

StreamOUT 实验说明

《圣经》箴言九 11“敬畏耶和华是智慧的开端, 认识至胜者便是聪明。”



1. 安装装配说明连接好各个功能模块, 给板子上电。
2. 在 USB Control Center 中将光盘中的 “...\prj\
vip_ex12\fx2_firmware\slave.hex” 文件下载到 FX2 的 RAM 中。
3. 若需要安装驱动, 参考 “loopback 实验说明”。
4. 将 “\prj\vip_ex12\fx2_StreamOUT\output_files\vip.sof” 文件下载到 VIP
板的 FPGA 中。
5. 在 USB Control Center, 选择 “Bulk out endpoint (0x02)”, 然后点击第二
个 “Transfer” 按钮, 随后弹出文件夹我们定位到文件 ““\prj\
vip_ex12\fx2_StreamOUT\vlgsrc\TEST_256x16bit.txt”, 发送这里已经预
存储在 TXT 文本中的 256*16bit 数据。



6. 在 FPGA 端, 通过 Quartus II 软件, 我们不仅可以使使用在线逻辑分析仪
SignalTap II 查看 SlaveFIFO 的读时序, 也可以使用 In-system Memory
Content Editor 查看已经接收到的一帧数据。

如图所示, 这是 PC 端发送并打印的数据帧。

《圣经》箴言九 11“敬畏耶和华是智慧的开端, 认识至胜者便是聪明。”



```
Descriptor Info  Data Transfers  Device Class Selection

Text to send      Data to send (Hex)
[Empty]          [Empty]

Bytes to Transfer
512

[ ] PktMode

[Transfer]      [Transfer]      [Clear Box]

BULK OUT transfer
0000 10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
0010 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0020 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0030 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0040 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0050 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0060 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0070 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0080 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0090 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
00A0 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
00B0 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
00C0 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
00D0 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
00E0 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
00F0 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0100 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0110 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0120 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0130 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0140 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0150 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0160 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0170 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0180 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
0190 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
01A0 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
01B0 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
01C0 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
01D0 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
01E0 A5 A5 A5 A5 5A 5A 5A 5A A5 A5 A5 A5 5A 5A 5A 5A
01F0 F0 F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF
BULK OUT transfer completed
```

如图所示, 这是在 In-system Memory Content Editor 查看到的 FPGA 端接收的数据帧。这里的数据和发送端的完全一致。



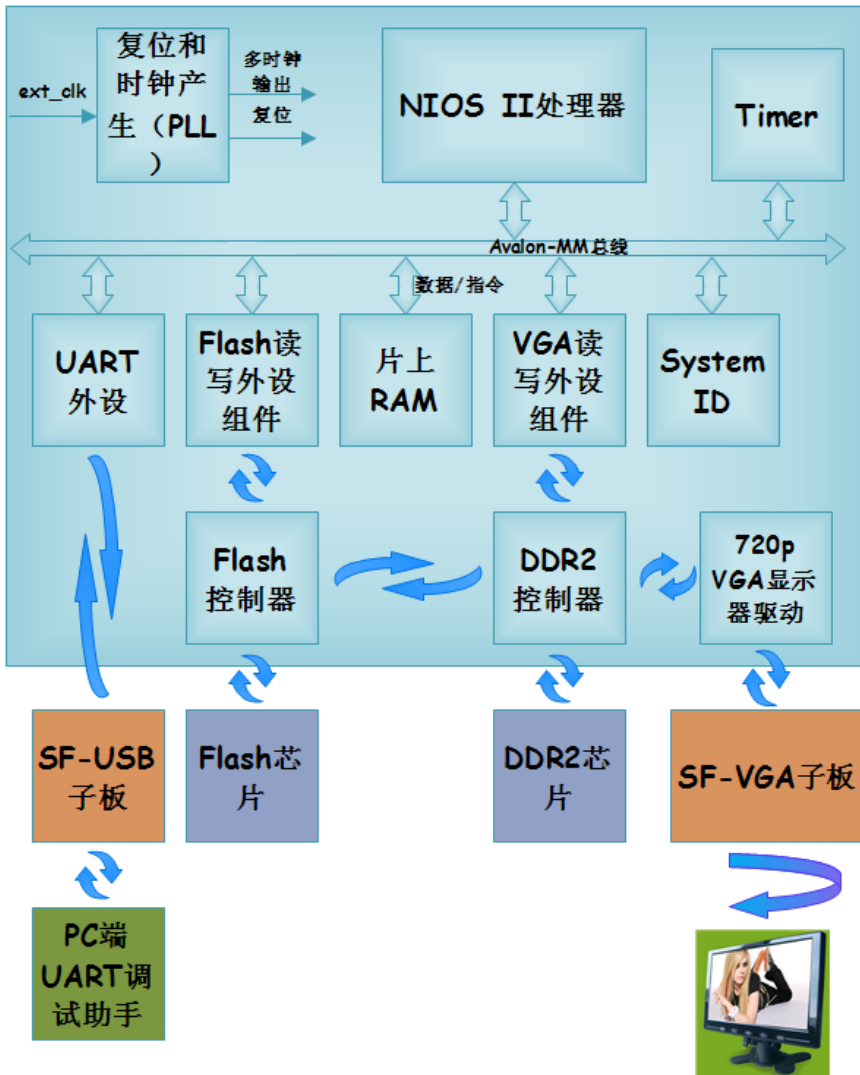
Instance 0: NONE																				
000000	11	10	13	12	15	14	17	16	19	18	1B	1A	1D	1C	1F	1E	A5	A5	A5	A5
00000a	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A
000014	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5
00001e	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A
000028	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5
000032	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A
00003c	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5
000046	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A
000050	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5
00005a	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A
000064	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5
00006e	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A
000078	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5
000082	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A
00008c	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5
000096	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A
0000a0	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5
0000aa	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A
0000b4	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5
0000be	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A
0000c8	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5
0000d2	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A
0000dc	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5
0000e6	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A
0000f0	A5	A5	A5	A5	5A	5A	5A	5A	A5	A5	A5	A5	5A	5A	5A	5A	F1	F0	F3	F2
0000fa	F5	F4	F7	F6	F9	F8	FB	FA	FD	FC	FF	FE								



2.13 特权 VIP 例程 13—基于 VGA 驱动的 720p 广告机实例

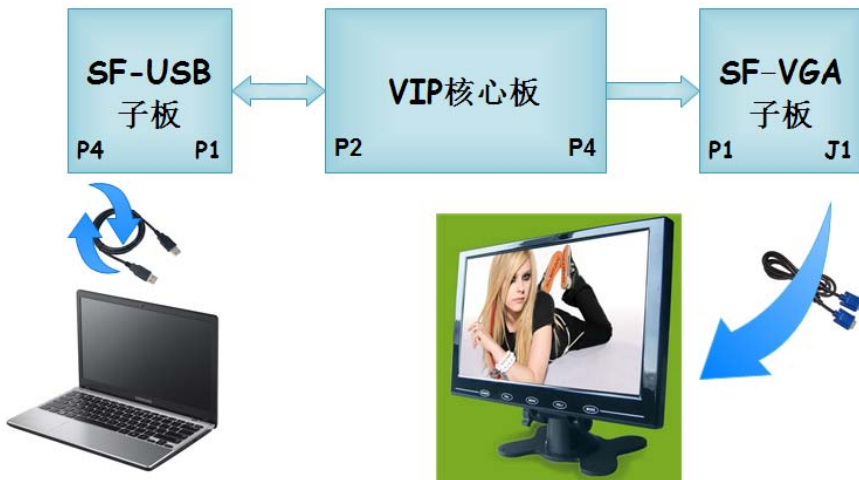
功能简介

本实例的基本架构和电子点菜单项目实例的基本一致。只是将电子点菜单实例中的 7 寸 LCD 换成了通用的 VGA 显示器(很多液晶电视基本也都支持 VGA 输出功能, 本实例演示就用了一台电视的 VGA 进行显示)。显示的分辨率也有电子点菜单的 800*480 提升到了 1080*720(720p)。这个实例中, 不仅要实现 720p 图片的下载存储操作, 同时在正常工作中, 循环播放预存储在 NAND Flash 芯片中的广告图片。整个工程实例的功能框图如下图所示。



装配说明

本实例工程除了VIP核心板，还需要SF-USB子板的UART串口实现图片的下载操作，以及SF-VGA子板用于驱动VGA显示器。整个系统的装配连接如图所示。



引脚分配

本工程实例由电子点菜单工程移植而来，因此只要将原先的 LCD 接口信号的引脚分配删除，并对 VGA 接口的引脚进行分配即可。VGA 接口引脚的定义如下所示。

信号名	VIP 板插座号	引脚号	备注
R0	P4-5	B12	
R1	P4-6	A12	
R2	P4-7	B13	
R3	P4-8	A13	
R4	P4-9	B14	
G0	P4-10	A14	
G1	P4-11	B16	
G2	P4-12	A15	
G3	P4-13	C16	



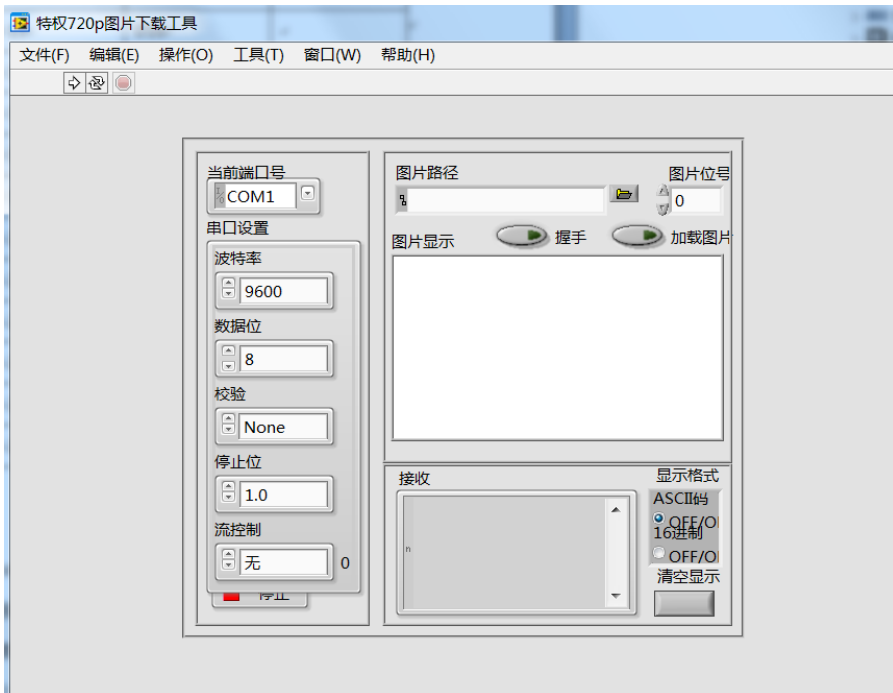
G4	P4-14	C15	
G5	P4-15	D16	
BLK	P4-16	D15	
SYN	P4-17	F16	
B0	P4-18	F15	
B1	P4-19	G16	
B2	P4-20	G15	
B3	P4-21	E11	
B4	P4-22	E10	
CLK	P4-23	C14	
VSY	P4-26	D14	
HSY	P4-27	F13	

实验说明

1. 按照装配说明将 SF-VIP 核心板、SF-VGA 子板和 SF-USB 子板连接好, 同时将 VGA 显示器连接到 SF-VGA 子板上, 连接好 SF-VIP 核心板和 PC 之间的 USB-Blaster 下载线, 连接好 SF-USB 子板和 PC 之间的 USB 线, 最后需要给整个系统上电。
2. 将 “...\prj\vip_ex13\vip_vga\output_files\vip.sof” 文件下载到 VIP 板的 FPGA 中。
3. 双击 “...\prj\vip_ex13\VIP_LVPRJ_UART720P\builds\特权 720p 图片下载工具 (VIP 套件专用) 安装程序\Volume” 下的 “setup.exe” 安装图片下载工具。
4. 安装完成, 在 “开始→程序” 中打开刚刚安装好的工具软件, 打开软



件节目如图所示。



5. 参照电子点菜单实例，确认 FT232RL 的驱动安装完成，如图所示，在设备管理器中，我们的 USB Serial Port 设备为 COM3 端口。



6. 回到打开的“特权 720p 图片下载工具”界面。选择“当前端口号”为 COM3；“接收”的显示格式为“16 进制”，波特率为“115200”，其他设置默认即可。



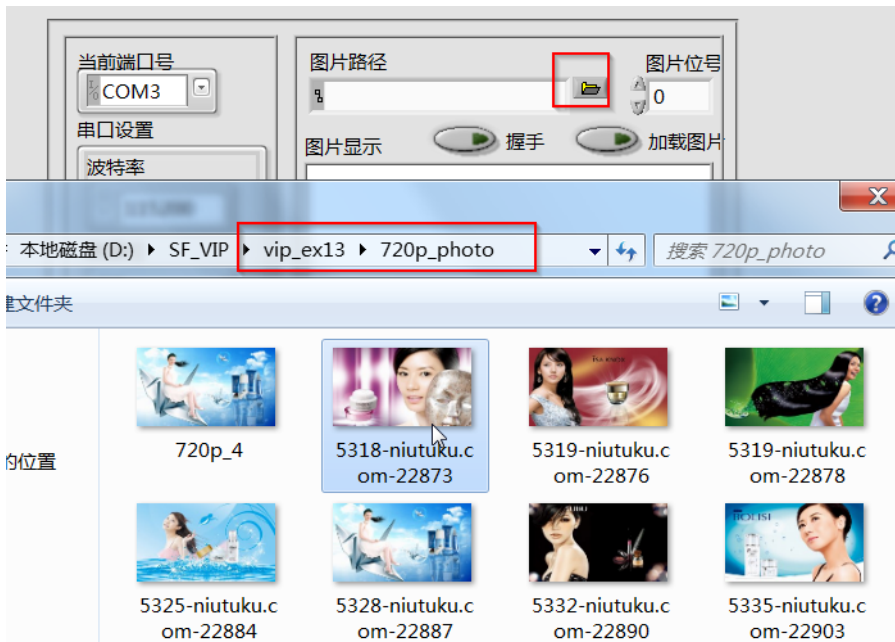
7. 点击“运行”按钮。



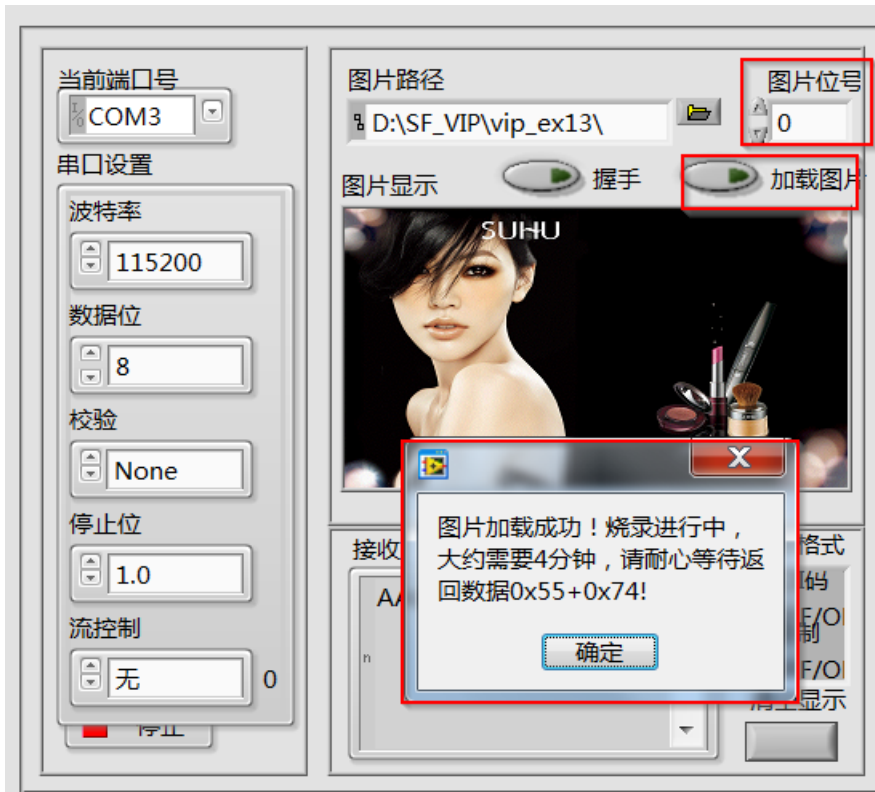
8. 点击“握手”按钮，直到“接收”打印“AA00”的返回数据，这说明 PC 端和 FPGA 端已经建立的 UART 通信。接着我们可以开始准备图片的下载操作了。



9. 点击“图片路径”后面的按钮，选择“...\vip_ex13\720p_photo”路径下的任意一张图片。这便是选择我们需要下载的图片。若大家希望使用自己的图片，请将图片在画图板中转成 1280*720 分辨率的 24bit 的 bmp 格式图片即可加载。其他格式或分辨率的图片加载将会出现报错。

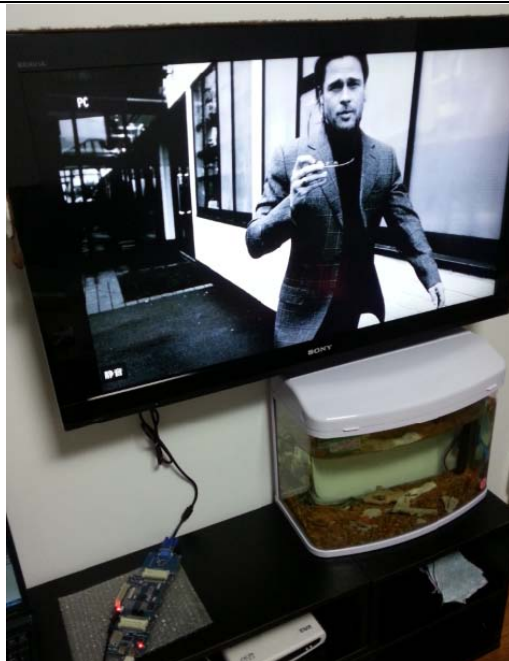


10. 选择好图片后, 可以更改“图片位号”, 本实例将使用到 0-9 位号的图片, 对他们进行循环播放, 所以大家要选择 10 张不同的图片分别设定图片位号为 0-9 进行下载。我们点击“加载图片”后, 会弹出一个对话框, 提示烧录进行需要 4 分钟, 并且在“接收”打印“55 74”数据后表示图片烧录完成, 此时我们也可以注意在“图片显示”中已经出现了我们所选择图片的预览。单击弹出对话框的“确认”按钮, 将启动烧录操作。



11. 当完成一幅图片的烧录后, 建议大家再次单击“握手”按钮, 确认当前 FPGA 端和 PC 端通信正常, 然后继续下一幅图的下载操作。
12. 完成 10 副图片的下载操作后, 我们便可以看到 VGA 显示器在不断的循环播放我们的广告图片。显示效果如图所示。



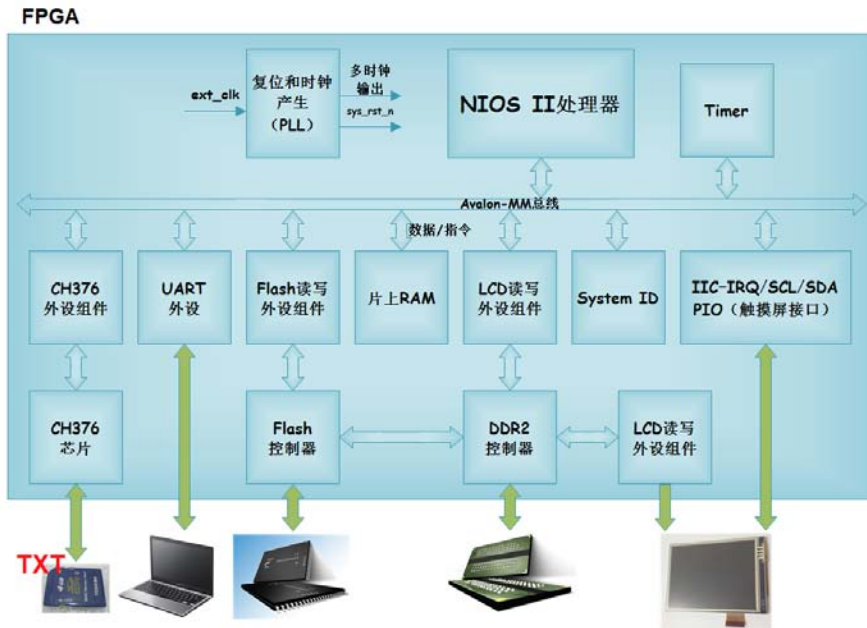




2.14 特权 VIP 例程 14—TXT 文本阅读器

功能简介

本实例也是基于例程 8（电子菜单）的基础上，增加了 Flash 字库的存储和显示，同时也增加了 CH376 芯片的控制外设组件，用于实现 SD 卡中 TXT 文本的读取操作。本实例的基本功能框图如下所示。



本实例需要实现的基本功能如下：

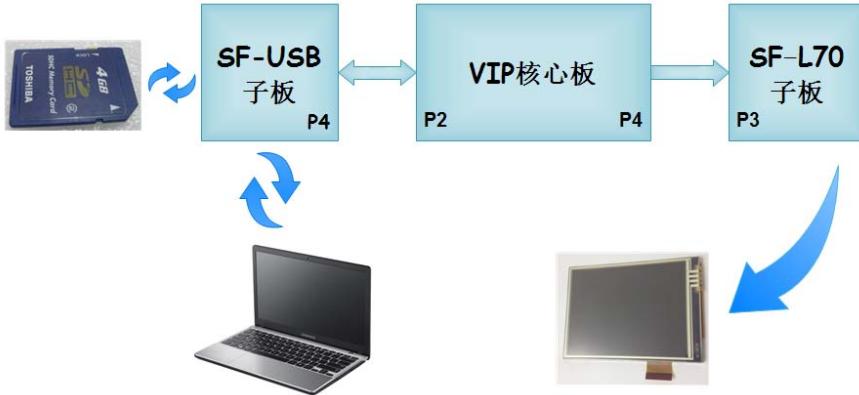
- 使用 UART 传输 3 张图片到 Flash 中，用于作为 TXT 文本显示的背景图像；
- 使用 UART 传输 16*16、32*32、64*64 三种大小的国标汉字到 Flash 中，用于液晶屏上的字符显示；
- Nand Flash 和 DDR2 SDRAM 的读写操作；
- NIOS II 系统及其基本组件的构建；
- 800*480 的 7 寸 LCD 显示驱动。

《圣经》箴言九 11“敬畏耶和华是智慧的开端，认识至胜者便是聪明。”



装配说明

本实例的基本连接如图所示。一图胜千言，高智商的童鞋们都懂。



引脚分配

这里只有 CH376 芯片连接部分的引脚是新添加进来的，其他都和例程 8 一样。因此给出 CH376 部分的引脚分配如下所示。

```
set_location_assignment PIN_D1 -to usb_a0
set_location_assignment PIN_B3 -to usb_cs_n
set_location_assignment PIN_D6 -to usb_db[7]
set_location_assignment PIN_E7 -to usb_db[6]
set_location_assignment PIN_D5 -to usb_db[5]
set_location_assignment PIN_C3 -to usb_db[4]
set_location_assignment PIN_E6 -to usb_db[3]
set_location_assignment PIN_D3 -to usb_db[2]
set_location_assignment PIN_G5 -to usb_db[1]
set_location_assignment PIN_F3 -to usb_db[0]
set_location_assignment PIN_C2 -to usb_int_n
set_location_assignment PIN_F2 -to usb_rd_n
set_location_assignment PIN_B1 -to usb_wr_n
```



实验说明

1. 首先按照装配说明, 完成各种子板、外设的连接, 并且完成各种必要的驱动安装 (前面章节都有介绍, 这里不赘述了)。当然了, 还有 USB-Blaster 下载线和电源不要忘记。给整板上电。
2. 打开 ex14/ vip_ex_book 文件夹下的工程, 将 output_files/vip.sof 文件烧录到 FPGA 中。
3. 打开 ex14/ uart_tools 文件夹下的软件 UART_TOOLS。

如图所示, 打开硬件管理器, 看到识别到虚拟串口的端口号为 COM7。



如图所示, 因此设定端口号为 COM7、选择 M070S65、点击“打开串口” (点击完后变成“关闭串口”图标)、然后选择波特率为 115200。这里需要严格按照上面的步骤操作 (具体也可参考实例 8)。一会我们会用到该工具进行 3 张图片和字库的烧录下载。

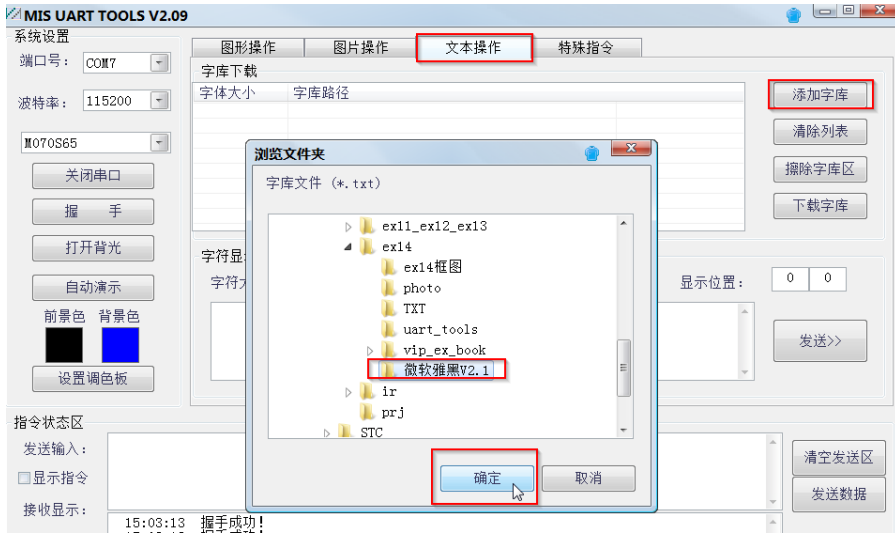


4. 开启 ED5, import 软件工程(software 文件夹下), 并且将其 Run 到 FPGA 中。
5. 此时我们可以回到 UART TOOL 界面中, 点击“握手”按钮, 可以看到“接收显示”窗口中提示“握手成功”。那么说明我们的硬件和软件的确都已经在 FPGA 中 Run 起来了。

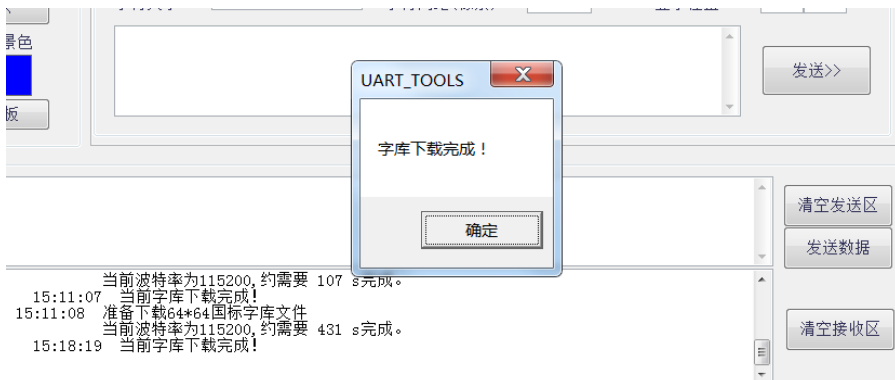




- 参考实例 8, 将 ex14/photo 文件夹下的 3 张图片分别下载到位号为 0、1、2 的 Flash 中。
- 如图所示, 在“文本操作”界面中, 点击“添加字库”按钮, 然后将浏览文件夹定位到 ex14/微软雅黑 V2.1 目录下, 单击“确定”完成字库的加载。

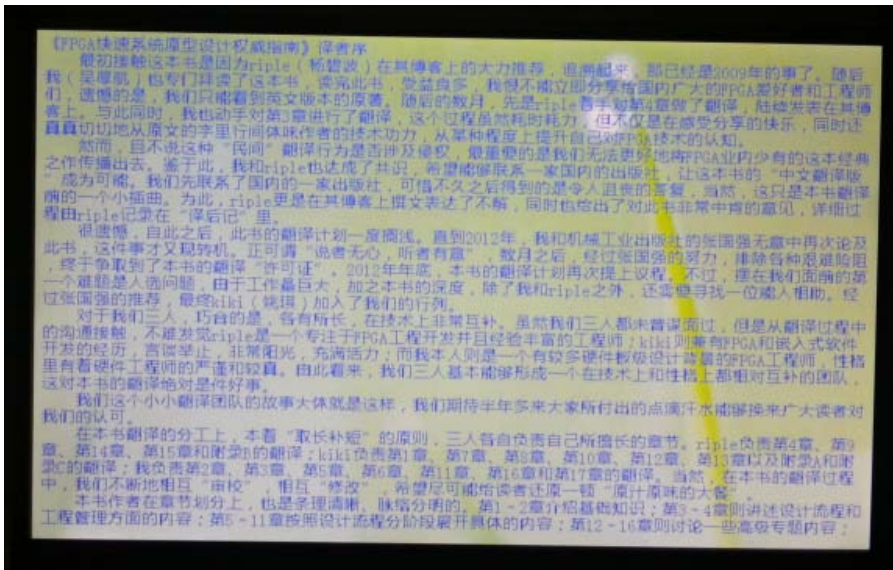


- 接着先单击“擦除字库区”按钮, 然后点击“下载字库”按钮, 进行字库的烧录下载。整个过程大约需要 10 分钟的时间, 大家可以去泡杯茶先。
- 烧录完成后, 则会弹出如图所示的对话框。

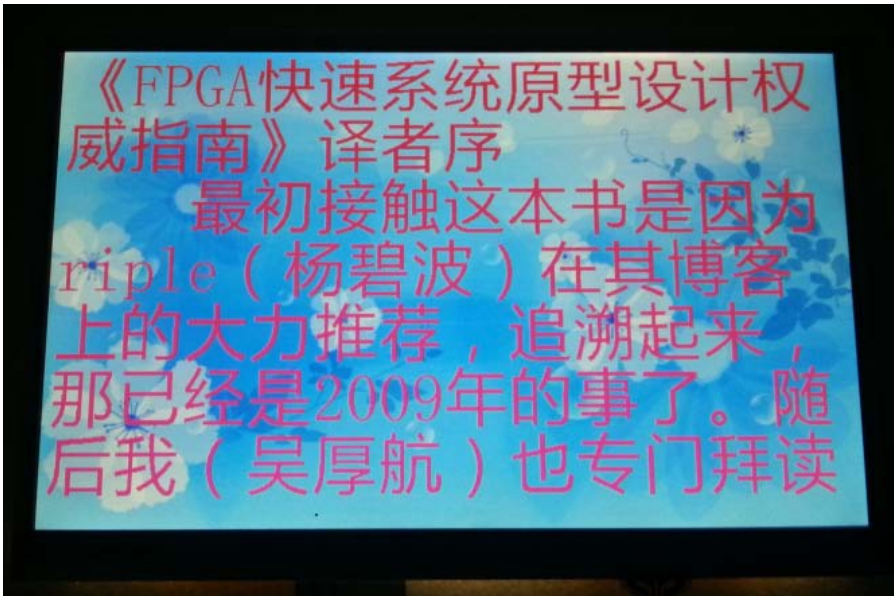




- 接着我们可以找一张 SD 卡来, 往里面拷贝 ex14/TXT 文件夹下的 READ.txt 文本。接着插入到 SF-USB 子板的 SD 卡槽 (P2) 中。
- 如果整个系统不下电, 我们则可以直接再次运行软件工程, 此时我们可以看到当前 SD 卡中的 TXT 文本“跃然屏上”。



《圣经》箴言九 11“敬畏耶和华是智慧的开端, 认识至胜者便是聪明。”



12. 在软件工程的 `mydef.h` 文件中, 以下的宏定义可以设定显示字符的前景和背景色 (背景色不建议修改, `0x0000` 表示以背景图片做背景色)、`x` 和 `y` 坐标的字符间距。

```
#define LCD_XDISTANCE0 //x字符间距
#define LCD_YDISTANCE0 //y字符间距
#define LCD_FCOR 0x001f //字体前景色
#define LCD_BCOR 0x0000 //字体背景色
```

Main 函数中调用函数 `mu_photo_display` 的第二个参数可以更改背景图片的位号。

```
mu_photo_display(0x70, 1, 0, 0); //显示主菜单
```

Main 函数中调用函数 `page_chinese_display` 则可以更改字体大小。

```
page_chinese_display(16, book_page); //16*16 字体大小显示文本
//page_chinese_display(32, book_page); //32*32 字体大小显示文本
//page_chinese_display(64, book_page); //64*64 字体大小显示文本
```



附录 A 实例与工程映射

实例名	硬件目录	软件目录	硬件平台
特权 VIP 例程 01--PLL 例化和 LED 闪烁	...\prj\vip_ex1	—	SF-VIP1
特权 VIP 例程 02--DDR2 控制器读写测试	...\prj\vip_ex2	—	SF-VIP1
特权 VIP 例程 03--Qsys 系统与软件实例	...\prj\vip_ex3	...\prj\vip_ex3\software	SF-VIP1
特权 VIP 例程 04--NAND FLASH 读写测试	...\prj\vip_ex4	...\prj\vip_ex4\software	SF-VIP1
特权 VIP 例程 05-- VGA /SVGA /720p /1080p 多分辨率显示驱动	...\prj\vip_ex5	—	SF-VIP1 + SF-VGA
特权 VIP 例程 06—7 寸 800*480 LCD 显示驱动	...\prj\vip_ex6	—	SF-VIP1 + SF-L70
特权 VIP 例程 07—触摸屏驱动	...\prj\vip_ex7	...\prj\vip_ex7\software	SF-VIP1 + SF-L70
特权 VIP 例程 08—电子点菜单	...\prj\vip_ex8	...\prj\vip_ex8\software	SF-VIP1 + SF-L70 + SF-USB
特权 VIP 例程 09—200W 像素 CMOS 摄像头采集显示	...\prj\vip_ex9	—	SF-VIP1 + SF-L70 + SF-SENSOR + MT9D111
特权 VIP 例程 10—AV(PAL/NTSC)视频采集显示	...\prj\vip_ex10	—	SF-VIP1 + SF-L70 + SF-AV
特权 VIP 例程 11—FX2 硬件和驱动安装以及 Bulkloop 实验	...\prj\vip_ex11	—	SF-VIP1 + SF-FX2
特权 VIP 例程 12—基于 FX2 和 FPGA 的	...\prj\vip_ex12	...\prj\vip_ex12\fx2_firmware	SF-VIP1 + SF-FX2



Loopback/StreamIN/StreamOUT 实例		...\prj\vip_ex12\fx2_loopback ...\prj\vip_ex12\fx2_StreamIN ...\prj\vip_ex12\fx2_StreamOUT	
特权 VIP 例程 13—基于 VGA 驱动的 720p 广告机实例	...\prj\vip_ex13	...\prj\vip_ex13\software	SF-VIP1 + SF-USB + SF-VGA
特权 VIP 例程 14—TXT 文本阅读器	...\prj\vip_ex14	...\prj\vip_ex14\software	SF-VIP1 + SF-USB + SF-L70



附录 B 套件淘宝购买链接

套件名称	主要配件	价格	淘宝链接
SF-VIP1 核心板	焊接并调试好的 VIP 电路板 资料光盘	RMB499	http://item.taobao.com/item.htm?spm=0.0.0.0.oJ1o4l&id=39542308218
SF-USB 子板	焊接并调试好的 SF-USB 子板 USB 线一条	RMB129	http://item.taobao.com/item.htm?id=18757370673
SF-L70 子板	焊接并调试好的 SF-L70 子板 40PIN FPC 一条 带触摸屏的 7 寸液晶屏一块	RMB398	http://item.taobao.com/item.htm?spm=0.0.0.0.u5CFVf&id=40437200629
SF-SENSOR 子板	焊接并调试好的 SF-SENSOR 子板 OV7670 CMOS 摄像头一个 超声波测距模块一个 3V 纽扣电池一个	RMB199	http://item.taobao.com/item.htm?spm=a1z10.5.w4002-6129022074.26.6O6izY&id=17838581080
MT9D111 摄像头	MT9D111 摄像头一个	RMB150	http://item.taobao.com/item.htm?spm=a1z10.5.w4002-6129022074.16.o0h12w&id=39303073016
SF-AV 子板	焊接并调试好的 SF-AV 子板	RMB120	http://item.taobao.com/item.htm?spm=0.0.0.0.oFXFdi&id=41186607353

特权制造 本教程只适用于特权开发制作的各类学习套件使用
淘宝店链接: <http://myfpga.taobao.com/>



SF-FX2 子板	焊接并调试好的 SF-FX2 子板 USB 线一条	RMB90	http://item.taobao.com/item.htm?spm=a1z10.5.w4002-8255776496.18.6Y7wQK&id=41898221959
SF-VGA 子板	焊接并调试好的 SF-VGA 子板	RMB119	http://item.taobao.com/item.htm?spm=a1z10.5.w4002-8255776496.19.cxDU9x&id=25407728030

《圣经》箴言九 11 “敬畏耶和华是智慧的开端，认识至胜者便是聪明。”