

# 基于 Nios II 实现多类型 LCD 屏幕彩条显示

## 电子信息虚拟仿真实验期末汇报

陈文轩

杭州电子科技大学卓越学院

2025 年 5 月 12 日



杭州电子科技大学  
HANGZHOU DIANZI UNIVERSITY

## ① 实验基本原理

## ① 实验基本原理

## Qsys/NiosII 简介

- 利用 Qsys 系统集成工具，通过 IP 核简单搭建 SOPC 系统，并自动创建 IP 核（如 SDRAM 的控制）之间的互联逻辑（例化、互相通信等）。

## Qsys/NiosII 简介

- 利用 Qsys 系统集成工具，通过 IP 核简单搭建 SOPC 系统，并自动创建 IP 核（如 SDRAM 的控制）之间的互联逻辑（例化、互相通信等）。
- NiosII 是 Altera 为 FPGA 设计的一种 RSIC 架构的嵌入式软核处理器。NiosII 作为处理器核心，是 Qsys 中可以使用的众多 IP 之一。

## Qsys/NiosII 简介

- 利用 Qsys 系统集成工具，通过 IP 核简单搭建 SOPC 系统，并自动创建 IP 核（如 SDRAM 的控制）之间的互联逻辑（例化、互相通信等）。
- NiosII 是 Altera 为 FPGA 设计的一种 RSIC 架构的嵌入式软核处理器。NiosII 作为处理器核心，是 Qsys 中可以使用的众多 IP 之一。
- Qsys 提供图形化界面设计系统，NiosII 则提供软件开发环境运行程序（嵌入式 C 语言开发）

## 实验完成大致流程 1

- 根据正点原子 NiosII 教程，先利用 Qsys 工具部署 IP 核，并进行连接。

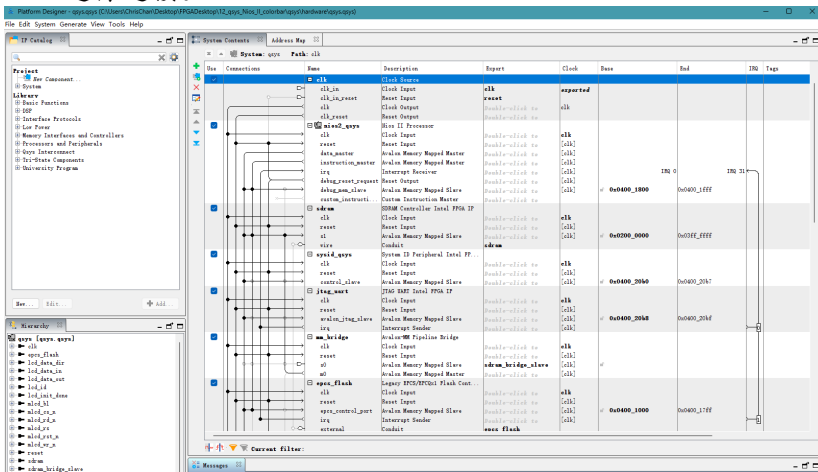
## 实验完成大致流程 1

- 根据正点原子 NiosII 教程，先利用 Qsys 工具部署 IP 核，并进行连接。



# 实验完成大致流程 1

- 根据正点原子 NiosII 教程，先利用 Qsys 工具部署 IP 核，并进行连接。



## 实验完成大致流程 2

- 根据正点原子 NiosII 嵌入式设计教程，设计嵌入式 C 程序

## 实验完成大致流程 2

- 根据正点原子 NiosII 嵌入式设计教程，设计嵌入式 C 程序

## 实验完成大致流程 2

- 根据正点原子 NiosII 嵌入式设计教程，设计嵌入式 C 程序

```
28 #include "unistd.h"
29 #include <string.h>
30 #include "App/mculcd.h"
31 #include "sys/alt_cache.h"
32
33 extern _lcd_dev lcddev; //管理LCD重要参数
34 _lcd_gui lcdgui;
35
36 //SDRAM显存的地址
37 alt_ul6 *ram_disp = (alt_ul6 *) (SDRAM_BASE + SDRAM_SPAN - 2049000);
38
39 int main()
40 {
41     int i,j;
42     MY_LCD_Init(); //LCD初始化
43     lcdgui.width = lcddev.height;
44     lcdgui.height = lcddev.width;
45     //向sdrum中写数据，
46     for(i=0;i<lcdgui.width;i++){
47         for(j=0;j<lcdgui.height;j++){
48             if(j<lcdgui.height/5)
```

# Qsys 实验原理简介

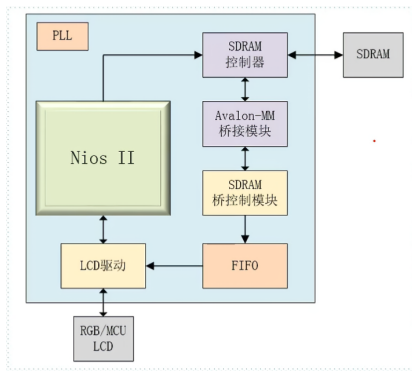


图 3: 实验框图

## Qsys 主要 IP 核介绍

- 由于 LCD 模块是部署在 NiosII 之外的模块，但也需要读取 SDRAM 的彩条控制信息

# Qsys 实验原理简介

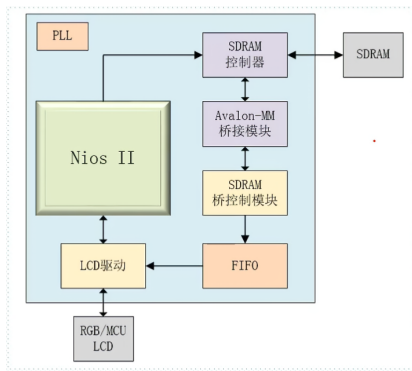


图 3: 实验框图

## Qsys 主要 IP 核介绍

- 由于 LCD 模块是部署在 NiosII 之外的模块，但也需要读取 SDRAM 的彩条控制信息
- 利用 SDRAM 控制器 IP 核的一个 Avalon 的端口引出

# Qsys 实验原理简介

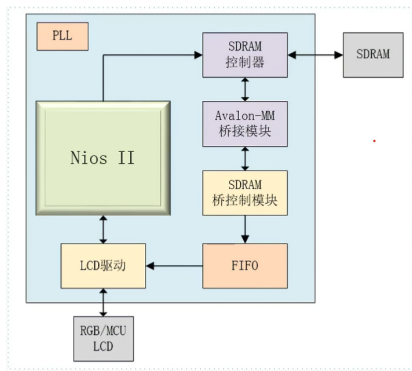


图 3: 实验框图

## Qsys 主要 IP 核介绍

- 由于 LCD 模块是部署在 NiosII 之外的模块，但也需要读取 SDRAM 的彩条控制信息
- 利用 SDRAM 控制器 IP 核的一个 Avalon 的端口引出
- 通过 Avalon-MM 模块的突发传输功能提高数据吞吐量

## Qsys 实验原理简介

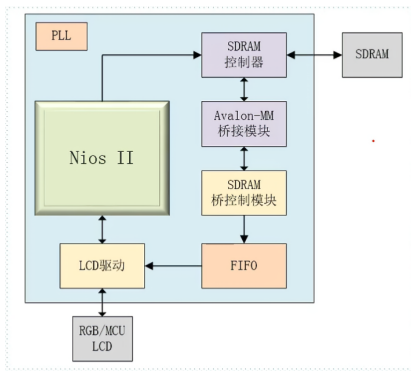


图 3: 实验框图

### Qsys 主要 IP 核介绍

- 由于 LCD 模块是部署在 NiosII 之外的模块，但也需要读取 SDRAM 的彩条控制信息
- 利用 SDRAM 控制器 IP 核的一个 Avalon 的端口引出
- 通过 Avalon-MM 模块的突发传输功能提高数据吞吐量
- 最终实现 LCD 模块可以高效访问外部 SDRAM，即 SDRAM 控制模块由 NiosII 和 SDRAM 桥控制模块两个主机进行控制



## LCD 驱动实验原理简介

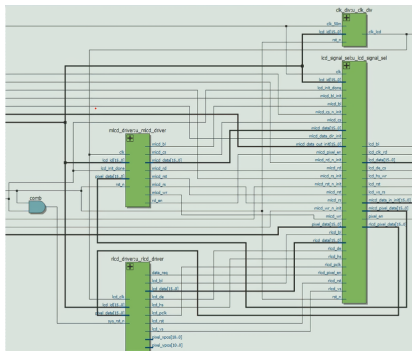


图 4: 实验框图

## LCD 驱动介绍

- 根据 LCDID[15:0] 信号, 判断出当前的 LCD 屏幕类型 (像素点多少、所需时钟频率等), 进行针对化驱动

## LCD 驱动实验原理简介

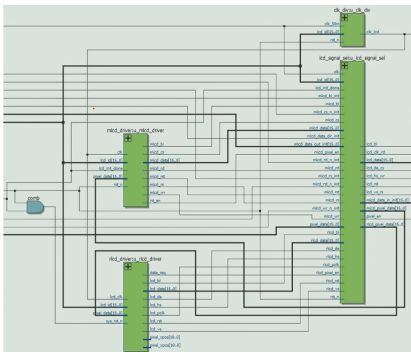


图 4: 实验框图

## LCD 驱动介绍

- 根据 LCDID[15:0] 信号, 判断出当前的 LCD 屏幕类型 (像素点多少、所需时钟频率等), 进行针对化驱动
- LCD 模块采用的是逐行扫描的形式, 具体来说, 它通过行计数器 (hcnt) 和场计数器 (vcnt) 模拟行同步和场同步信号, 从左到右、从上到下依次写入像素数据到 LCD 的 GRAM 中, 确保图像按顺序刷新显示。

# 实验程序示例

```
28 #include "unistd.h"
29 #include <string.h>
30 #include "App/mcoulod.h"
31 #include "sys/alt_cache.h"
32
33 extern _lcd_dev lcddev; //管理LCD重要参数
34 _lcd_gui lcdgui;
35
36 //SDRAM显存的地址
37 alt_u16 *ram_disp = (alt_u16 *) (SDRAM_BASE + SDRAM_SPAN - 2049000);
38
39 int main()
40 {
41     int i,j;
42     MY_LCD_Init(); //LCD初始化
43     lcdgui.width = lcddev.height;
44     lcdgui.height = lcddev.width;
45     //向sdram中写数据,
46     for(i=0;i<lcdgui.width;i++){
47         for(j=0;j<lcdgui.height;j++){
48             if(j<lcdgui.height/5)
49                 *(ram_disp++) = 0xf800; //红色
50             else if(j<(lcdgui.height/5*2))
51                 *(ram_disp++) = 0xffff; //白色
52             else if(j<(lcdgui.height/5*3))
53                 *(ram_disp++) = 0x0; //黑色
54             else if(j<(lcdgui.height/5*4))
55                 *(ram_disp++) = 0x07e0; //绿色
56             else
57                 *(ram_disp++) = 0x001f; //蓝色
58         }
59         alt_dcacheflush_all();
60     }
61     return 0;
62 }
63
64
```

- 该程序初始化 LCD 并设置显存地址。
- 使用嵌套循环向 SDRAM 写入彩条数据。
- 彩条颜色依次为红、白、黑、绿、蓝。
- 使用 'alt\_dcacheflush\_all' 刷新数据缓存。

图 5: 实验 C 代码

*Thanks!*