

浙江大学

本科实验报告

课程名称:	计算机组成
姓 名:	段皞一
学 院:	计算机科学与技术学院
专 业:	计算机
邮 箱:	1031004722@qq.com
QQ 号:	1031004722
电 话:	19883161889
指导教师:	施青松
报告日期:	2020 年 3 月 22 日

浙江大学实验报告

实验项目名称： IP 核集成 SOC 设计实验报告

学生姓名： 段皞一 学号： 3190105359

实验地点： 紫金港东四 509 室 实验日期： 2021 年 3 月 22 日

一、操作方法与实验步骤

1.1 实验的顶层架构

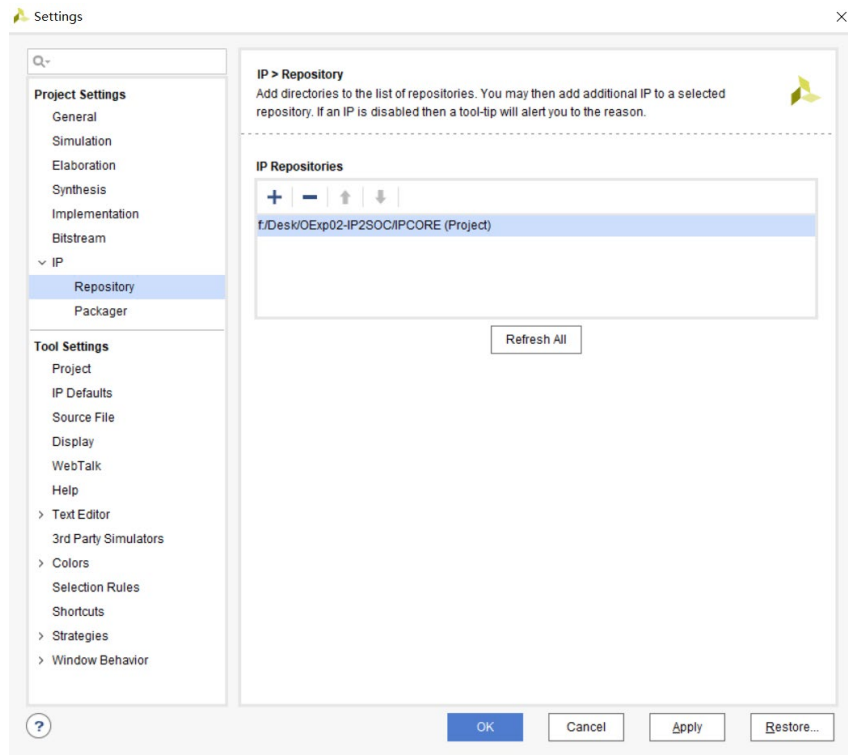


图表 1 实验的顶层架构图

1.2 实验任务：建立 SOC 应用工程

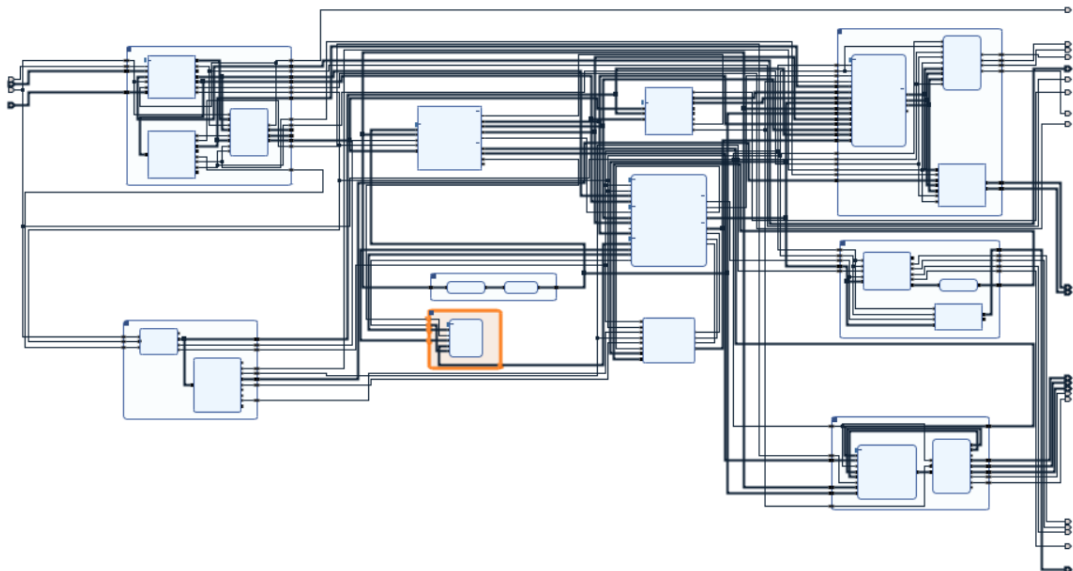
1 新建工程，命名为 OExp02-IP2SOC.

2 添加第三方 IP 核 CPU (U1),总线 (U4),开关去抖模块 (U9),数据输入模块 (M4),PIO (U71)。



图表 2 添加第三方的 IP 核

- 3 添加实验一中自己设计的模块：GPIO, PIO, DESGIO。
- 4 添加通用分频模块 clk_div (U8)。
- 5 构建 ROM (U2)，RAM (U3) IP 核，使用提供的 coe 文件。
- 6 新建顶层模块文件 OExp02_IP2SOC。绘制原理图。



图表 3 实验原理图

1.3 书写引脚约束文件

#系统时钟

```
set_property -dict {PACKAGE_PIN AC18 IOSTANDARD LVCMOS18} [get_ports
clk_100mhz]
```

```
#Reset or CR
set_property -dict {PACKAGE_PIN W13 IOSTANDARD LVCMOS18} [get_ports RSTN]
#SWORD LED:LED 同步串行接口
set_property -dict {PACKAGE_PIN N26 IOSTANDARD LVCMOS33} [get_ports LEDCK]
set_property -dict {PACKAGE_PIN N24 IOSTANDARD LVCMOS33} [get_ports LEDCR]
set_property -dict {PACKAGE_PIN M26 IOSTANDARD LVCMOS33} [get_ports LEDDT]
set_property -dict {PACKAGE_PIN P18 IOSTANDARD LVCMOS33} [get_ports LEDEN]
#SWORD SEG:七段码同步串行接口
set_property -dict {PACKAGE_PIN M24 IOSTANDARD LVCMOS33} [get_ports SEGCK]
set_property -dict {PACKAGE_PIN M20 IOSTANDARD LVCMOS33} [get_ports SEGCR]
set_property -dict {PACKAGE_PIN L24 IOSTANDARD LVCMOS33} [get_ports SEGDT]
set_property -dict {PACKAGE_PIN R18 IOSTANDARD LVCMOS33} [get_ports SEGEN]
#ArrayKeys:阵列键盘: 行
set_property -dict {PACKAGE_PIN V18 IOSTANDARD LVCMOS18} [get_ports {KCOL[0]}]
set_property -dict {PACKAGE_PIN V19 IOSTANDARD LVCMOS18} [get_ports {KCOL[1]}]
set_property -dict {PACKAGE_PIN V14 IOSTANDARD LVCMOS18} [get_ports {KCOL[2]}]
set_property -dict {PACKAGE_PIN W14 IOSTANDARD LVCMOS18} [get_ports {KCOL[3]}]
#ArrayKeys:阵列键盘: 列
set_property -dict {PACKAGE_PIN V17 IOSTANDARD LVCMOS18 PULLUP true} [get_ports {KROW[0]}]
set_property -dict {PACKAGE_PIN W18 IOSTANDARD LVCMOS18 PULLUP true} [get_ports {KROW[1]}]
set_property -dict {PACKAGE_PIN W19 IOSTANDARD LVCMOS18 PULLUP true} [get_ports {KROW[2]}]
set_property -dict {PACKAGE_PIN W15 IOSTANDARD LVCMOS18 PULLUP true} [get_ports {KROW[3]}]
set_property -dict {PACKAGE_PIN W16 IOSTANDARD LVCMOS18 PULLUP true} [get_ports {KROW[4]}]
#Switch:滑动开关
set_property -dict {PACKAGE_PIN AA10 IOSTANDARD LVCMOS15} [get_ports {SW[0]}]
set_property -dict {PACKAGE_PIN AB10 IOSTANDARD LVCMOS15} [get_ports
```

```
{SW[1]}}
set_property -dict {PACKAGE_PIN AA13 IOSTANDARD LVCMOS15} [get_ports
{SW[2]}}
set_property -dict {PACKAGE_PIN AA12 IOSTANDARD LVCMOS15} [get_ports
{SW[3]}}
set_property -dict {PACKAGE_PIN Y13 IOSTANDARD LVCMOS15} [get_ports
{SW[4]}}
set_property -dict {PACKAGE_PIN Y12 IOSTANDARD LVCMOS15} [get_ports
{SW[5]}}
set_property -dict {PACKAGE_PIN AD11 IOSTANDARD LVCMOS15} [get_ports
{SW[6]}}
set_property -dict {PACKAGE_PIN AD10 IOSTANDARD LVCMOS15} [get_ports
{SW[7]}}
set_property -dict {PACKAGE_PIN AE10 IOSTANDARD LVCMOS15} [get_ports
{SW[8]}}
set_property -dict {PACKAGE_PIN AE12 IOSTANDARD LVCMOS15} [get_ports
{SW[9]}}
set_property -dict {PACKAGE_PIN AF12 IOSTANDARD LVCMOS15} [get_ports
{SW[10]}}
set_property -dict {PACKAGE_PIN AE8 IOSTANDARD LVCMOS15} [get_ports
{SW[11]}}
set_property -dict {PACKAGE_PIN AF8 IOSTANDARD LVCMOS15} [get_ports
{SW[12]}}
set_property -dict {PACKAGE_PIN AE13 IOSTANDARD LVCMOS15} [get_ports
{SW[13]}}
set_property -dict {PACKAGE_PIN AF13 IOSTANDARD LVCMOS15} [get_ports
{SW[14]}}
set_property -dict {PACKAGE_PIN AF10 IOSTANDARD LVCMOS15} [get_ports
{SW[15]}}
#Tri-LED:三色指示灯
set_property -dict {PACKAGE_PIN U22 IOSTANDARD LVCMOS33} [get_ports
readn]
set_property -dict {PACKAGE_PIN V22 IOSTANDARD LVCMOS33} [get_ports CR]
set_property -dict {PACKAGE_PIN U21 IOSTANDARD LVCMOS33} [get_ports RDY]
#ARDUNIO-IO for LED
set_property -dict {PACKAGE_PIN AF24 IOSTANDARD LVCMOS33} [get_ports
{LED[0]}}
set_property -dict {PACKAGE_PIN AE21 IOSTANDARD LVCMOS33} [get_ports
{LED[1]}}
set_property -dict {PACKAGE_PIN Y22 IOSTANDARD LVCMOS33} [get_ports
{LED[2]}}
set_property -dict {PACKAGE_PIN Y23 IOSTANDARD LVCMOS33} [get_ports
{LED[3]}}
set_property -dict {PACKAGE_PIN AA23 IOSTANDARD LVCMOS33} [get_ports
```

```
{LED[4]}}
set_property -dict {PACKAGE_PIN Y25 IOSTANDARD LVCMOS33} [get_ports
{LED[5]}}
set_property -dict {PACKAGE_PIN AB26 IOSTANDARD LVCMOS33} [get_ports
{LED[6]}}
set_property -dict {PACKAGE_PIN W23 IOSTANDARD LVCMOS33} [get_ports
{LED[7]}}
#ARDUNIO for Buzzer
set_property -dict {PACKAGE_PIN AF25 IOSTANDARD LVCMOS33} [get_ports
Buzzer]
#ARDUNIO-IO for SEG
set_property -dict {PACKAGE_PIN AB22 IOSTANDARD LVCMOS33} [get_ports
{SEGMENT[0]}}
set_property -dict {PACKAGE_PIN AD24 IOSTANDARD LVCMOS33} [get_ports
{SEGMENT[1]}}
set_property -dict {PACKAGE_PIN AD23 IOSTANDARD LVCMOS33} [get_ports
{SEGMENT[2]}}
set_property -dict {PACKAGE_PIN Y21 IOSTANDARD LVCMOS33} [get_ports
{SEGMENT[3]}}
set_property -dict {PACKAGE_PIN W20 IOSTANDARD LVCMOS33} [get_ports
{SEGMENT[4]}}
set_property -dict {PACKAGE_PIN AC24 IOSTANDARD LVCMOS33} [get_ports
{SEGMENT[5]}}
set_property -dict {PACKAGE_PIN AC23 IOSTANDARD LVCMOS33} [get_ports
{SEGMENT[6]}}
set_property -dict {PACKAGE_PIN AA22 IOSTANDARD LVCMOS33} [get_ports
{SEGMENT[7]}}

set_property -dict {PACKAGE_PIN AD21 IOSTANDARD LVCMOS33} [get_ports
{AN[0]}}
set_property -dict {PACKAGE_PIN AC21 IOSTANDARD LVCMOS33} [get_ports
{AN[1]}}
set_property -dict {PACKAGE_PIN AB21 IOSTANDARD LVCMOS33} [get_ports
{AN[2]}}
set_property -dict {PACKAGE_PIN AC22 IOSTANDARD LVCMOS33} [get_ports
{AN[3]}}

#VGA
set_property -dict {PACKAGE_PIN N21 IOSTANDARD LVCMOS33 SLEW FAST}
[get_ports {Red[0]}}
set_property -dict {PACKAGE_PIN N22 IOSTANDARD LVCMOS33 SLEW FAST}
[get_ports {Red[1]}}
set_property -dict {PACKAGE_PIN R21 IOSTANDARD LVCMOS33 SLEW FAST}
[get_ports {Red[2]}}
```

```
set_property -dict {PACKAGE_PIN P21 IOSTANDARD LVCMOS33 SLEW FAST}
[get_ports {Red[3]}]
set_property -dict {PACKAGE_PIN R22 IOSTANDARD LVCMOS33 SLEW FAST}
[get_ports {Green[0]}]
set_property -dict {PACKAGE_PIN R23 IOSTANDARD LVCMOS33 SLEW FAST}
[get_ports {Green[1]}]
set_property -dict {PACKAGE_PIN T24 IOSTANDARD LVCMOS33 SLEW FAST}
[get_ports {Green[2]}]
set_property -dict {PACKAGE_PIN T25 IOSTANDARD LVCMOS33 SLEW FAST}
[get_ports {Green[3]}]
set_property -dict {PACKAGE_PIN T20 IOSTANDARD LVCMOS33 SLEW FAST}
[get_ports {Blue[0]}]
set_property -dict {PACKAGE_PIN R20 IOSTANDARD LVCMOS33 SLEW FAST}
[get_ports {Blue[1]}]
set_property -dict {PACKAGE_PIN T22 IOSTANDARD LVCMOS33 SLEW FAST}
[get_ports {Blue[2]}]
set_property -dict {PACKAGE_PIN T23 IOSTANDARD LVCMOS33 SLEW FAST}
[get_ports {Blue[3]}]
set_property -dict {PACKAGE_PIN M22 IOSTANDARD LVCMOS33 SLEW FAST}
[get_ports HSYNC]
set_property -dict {PACKAGE_PIN M21 IOSTANDARD LVCMOS33 SLEW FAST}
[get_ports VSYNC]
```

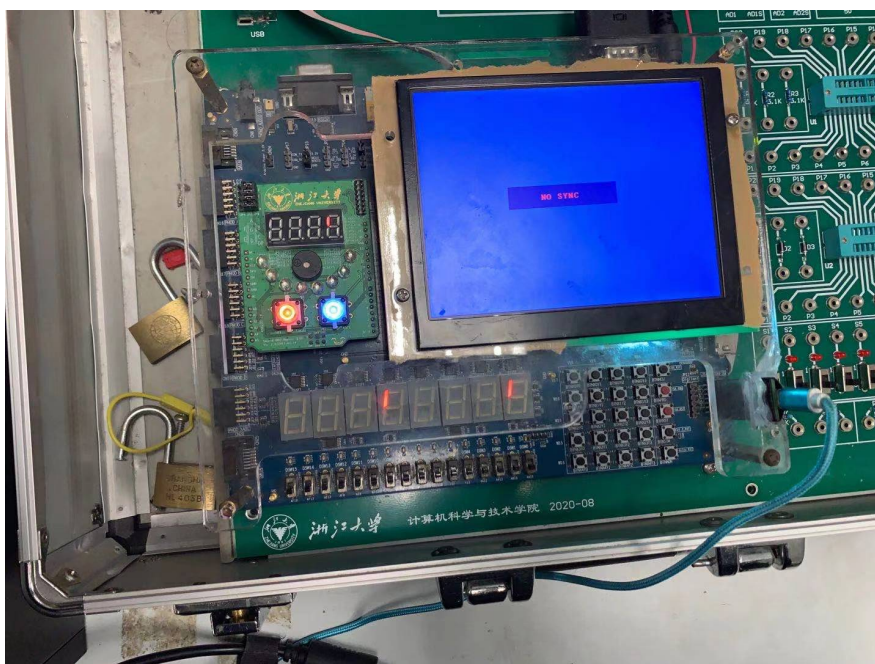
1.4 上板子验证

成功生成 BitStream 以后，就可以上板子验证实验的正确性了。

二、实验结果与分析

2.1 七段码——显示低 16 位

所有卡关掷“0”，产生显示低 16 位的图形七段码。



图表 4 七段码——显示低 16 位示意图

2.2 七段码——显示高 16 位

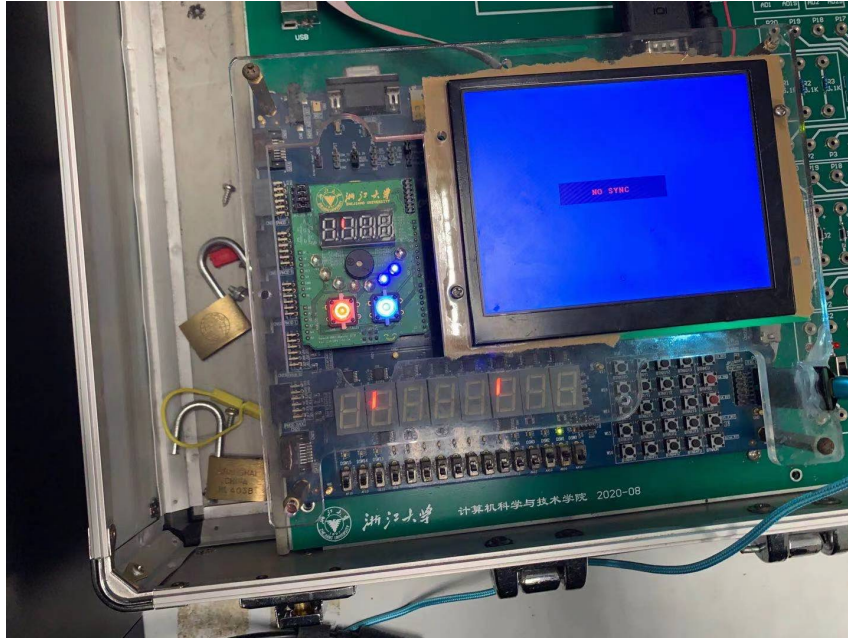
SW[1] = 1, 产生显示高 16 位的七段图形数码。如下图所示。



图表 5 七段码——显示高 16 位示意图

2.3 单步时钟——静止

SW[2] = 1, 产生单步时钟，七段数码管产生的跑马灯停止。

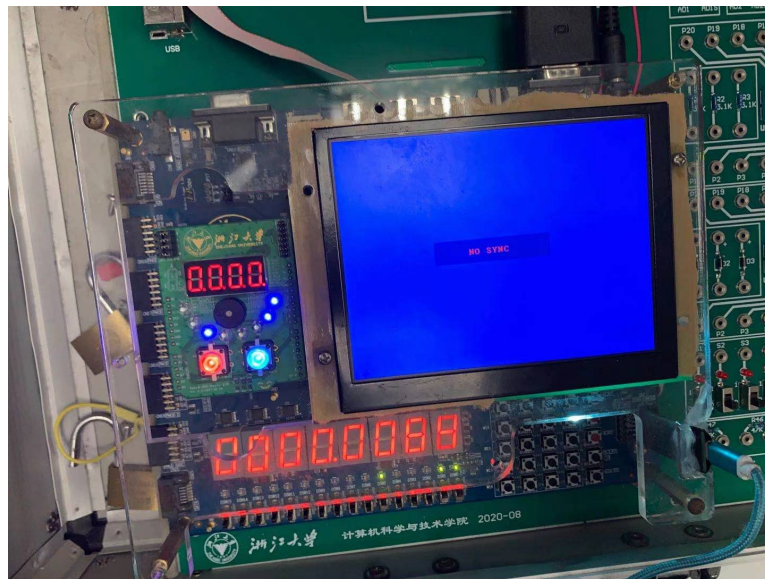


图表 6 单步时钟——静止示意图

2.4 Cpu 指令字地址

$SW[7:5] = 001$.

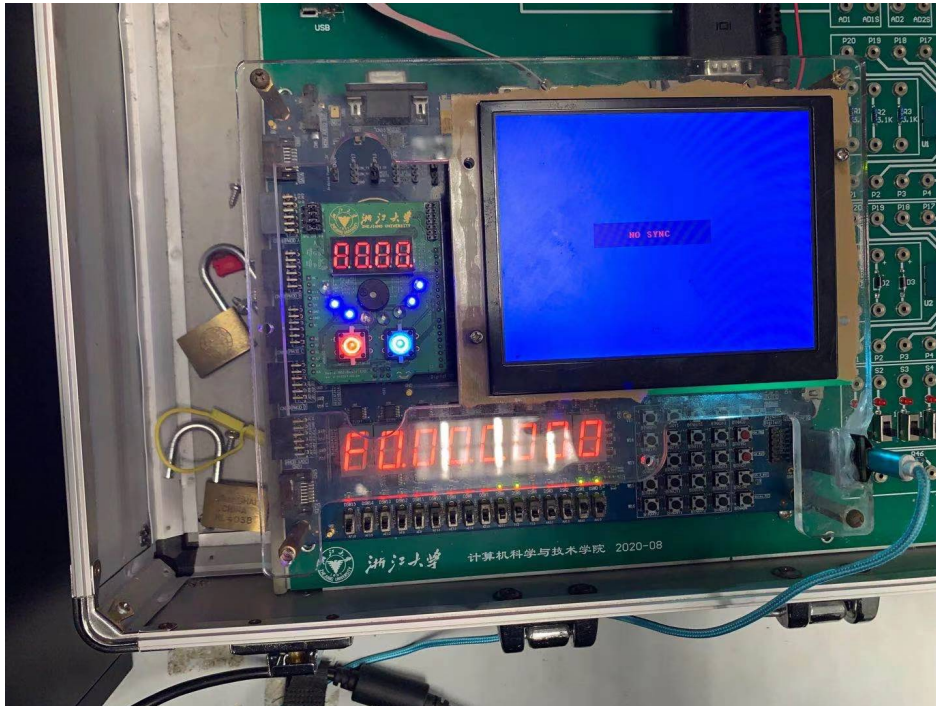
$SW[0]=SW[1]=SW[5] = 1$,产生 Cpu 指令字地址。



图表 7 Cpu 指令字地址示意图

2.5 ROM 指令输出

$SW[7:5] = 011$,产生 ROM 的指令输出。示意图如下。



图表 8 ROM 指令输出示意图

2.6 CPU 数据存储地址 addr—— ALU 输出

SW[7:5] = 100,产生 CPU 数据的存储地址 addr。相关示意图如下所示。



图表 9 CPU 数据存储地址 addr 的示意图

2.7 CPU 数据输出 Datao CPUD

SW[7:5] = 101,产生 CPU 数据输出的 Datao CPUD,相关示意图如下所示。



图表 10 CPU 输出信号示意图

2.8 CPU 数据输入 Datai CPUD

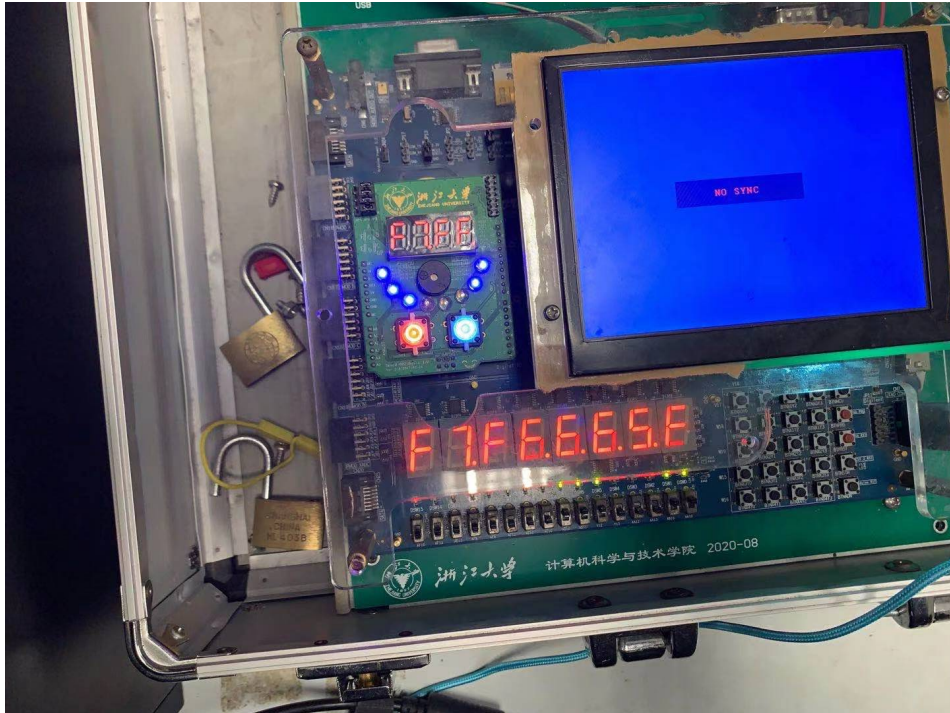
SW[7:5] = 110,产生 CPU 数据输入信号，相关的图像如下所示。



图表 11 CPU 数据输入示意图

2.9 CPU 指令字节地址 PC

SW[7:5] = 111,产生 CPU 指令字节地址 PC，相关示意图如下。



图表 12 CPU 指令字节地址输出示意图

2.10 七段码文本显示——低 16 位

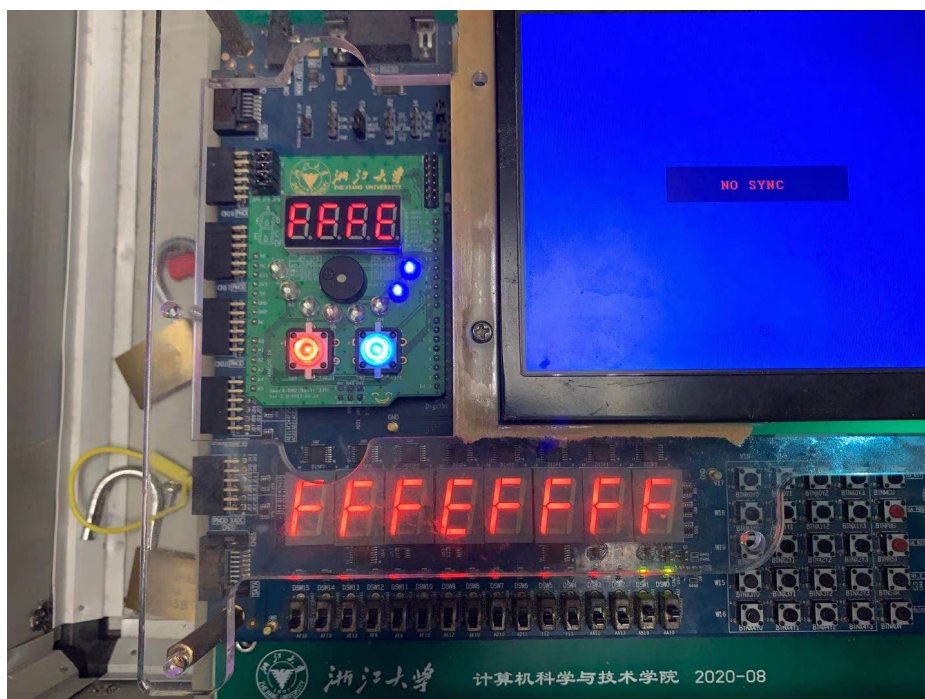
SW[0] = 1 的时候，输出七段码文本显示低 16 位的情况。



图表 12 七段数码管显示低 16 位的示意图

2.11 七段码文本显示——高 16 位

SW[1:0] = 11, 显示七段数码管的文本模式，高 16 位显示在板子上。相关示意图如下。



图表 13 七段数码文本显示——高 16 位示意图

2.12 七段码显示 RAM 数字

SW[3] = 1, 显示 RAM 里存储的数字, 相关示意图如下所示。



图表 14 七段码显示 RAM 数字示意图

2.13 七段码显示累加

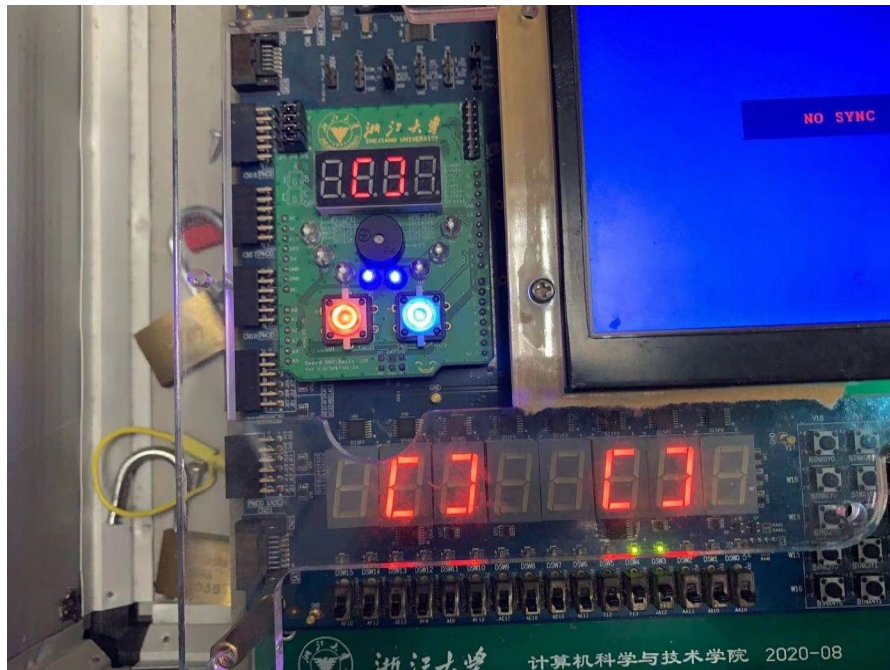
SW[1:0] = 11, SW[4] = 1, 七段码显示累加。相关示意图如下所示。



图表 15 七段码显示累加示意图

2.14 图形界面显示图像

SW[4:3] = 11, 显示动态图像，相关示意图如下。



图表 16 显示动态图像

2.15 VGA 显示模块



VGA DEBUG 显示

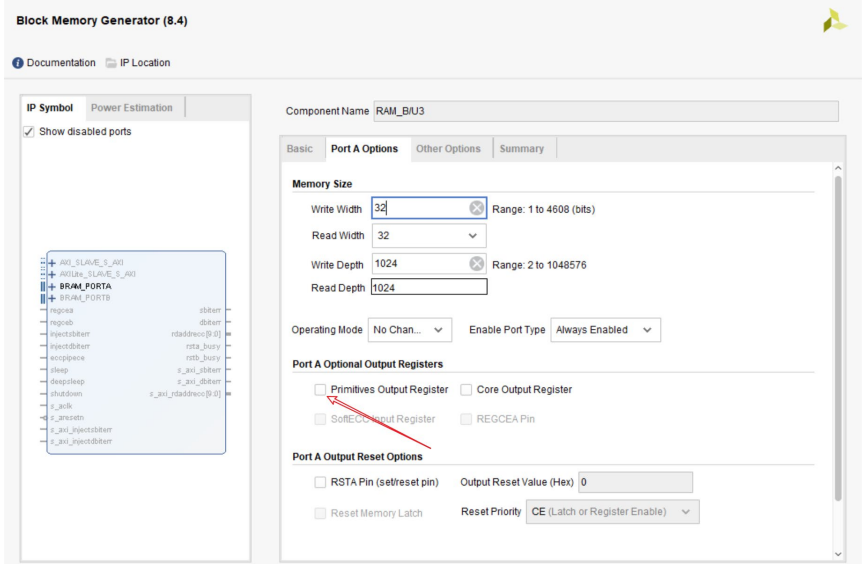


图表 17 VGA 显示模块示意图

三、讨论、心得

本次实验在前面实验的基础上，进一步建立了 CPU 调试测试环境。通过这次实验，我

对于 Vivado 平台更加熟悉了，连线也更加熟练了，没有像上次实验一样出现了连线错误的情况，这也让我能够较早的把重心放在之后的错误的解决上。实验中，同学们都出现了跑马灯跑不起来的情况，估计是 ROM 或者 RAM 在配置的时候出现了问题，因为这两个存储器里存储着关于七段码和跑马灯的.coe 文件，如果读取出现了问题，就无法正确产生跑马灯。最后，经过同学发现，是由于 RAM 在配置的过程中出了一些小问题。



图表 4 RAM 配置图

如图所示，要把图中的“Primitives Output Register”的勾去掉，就可以成功运行了。这个实验也启示我，在实验中，一定要十分心细，对实验的各个步骤的原理都要了解透彻，因为一个小小的问题可能导致灾难性的后果。

通过本次实验，我还进一步加深了 Vivado 引脚文件书写的印象。Vivado 的引脚文件相对于 ISE 平台有所不同，但本质上是一样的，我已经能够自己书写一些基础的引脚代码。

此外，在对 DEMO 进行定性分析的时候，也对加深了对 SOC 的认识，对整个架构有了一个全局性的把握。