计算机设计与实践

设计文档

班级： 1704103

选题： 字符串匹配

组内成员：陈恒杰、冯家威、张佳琪

一、选题描述

1、功能控制：利用拨码开关控制功能的切换

2、输入：字符串采用键盘输入，输入一个字符，即会利用led显示相应ascii码值

3、输出：在数码管上以十进制显示位置的下标，若不存在匹配的子串，则显示“none”

二、CPU内核总体设计

1、主要性能指标和特性：

（1）地址线宽度：16

（2）数据线宽度：32

（3）I/O编址方式：统一编址

（4）寻址方式：按字节寻址

（5）指令系统概况：RISC、定长指令字长、定长操作码

（6）字节序：小端次序

（7）存储体系：存储字长32位、哈佛结构

（8）是否采用流水线：否

2、构成

控制器、运算器、存储器、时钟

三、CPU指令系统设计

***I型指令：***

1、addi rt, rs, imme

功能：rt<——rs+SignExtImme

2、beq rs, rt, label

功能：if rs==rt then branch

3、bne rs, rt, label

功能：if rs!=rt then branch

4、lw rt, offset(rs)

功能：rt<——mem(rs+offset)

5、sw rt, offset(rs)

功能：mem(rs+offset)<——rt

6、andi rt, rs, imme

功能：rt<——rs&ZeroExtimme

***R型指令：***

1、add rd, rs, rt

功能：rd<——rs+rt

2、sll rd, rt, shamt

功能：rd<——rt<<shamt

3、sub rd, rs, rt

功能：rd<——rs-rt

4、and rd, rs, rt

功能：rd<——rs+rt

5、slt rd, rs, rt

功能：rd<——(rs<rt)?1:0

6、or rd, rs, rt

功能：rd<——rs|rt

7、srl rd，rt，shamt

功能：rd<——rt>>shamt

8、jr rs

功能：PC<——rs

***J型指令:***

1、j target

功能：jump to target

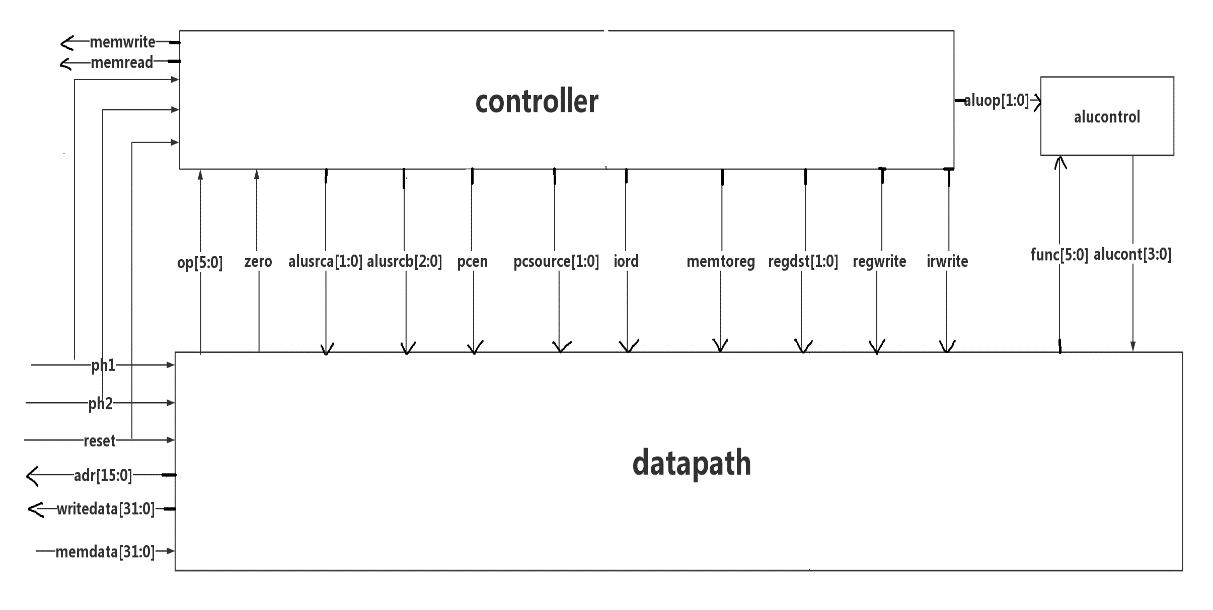
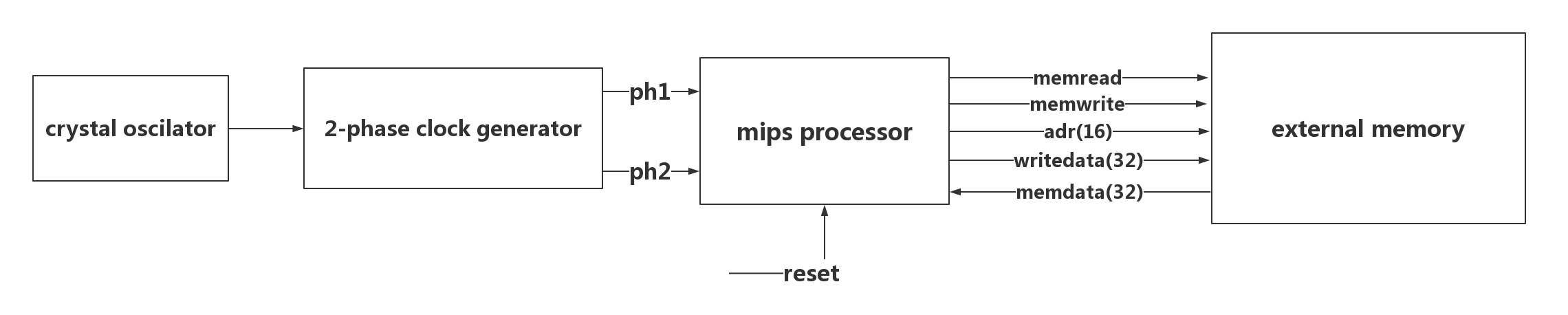
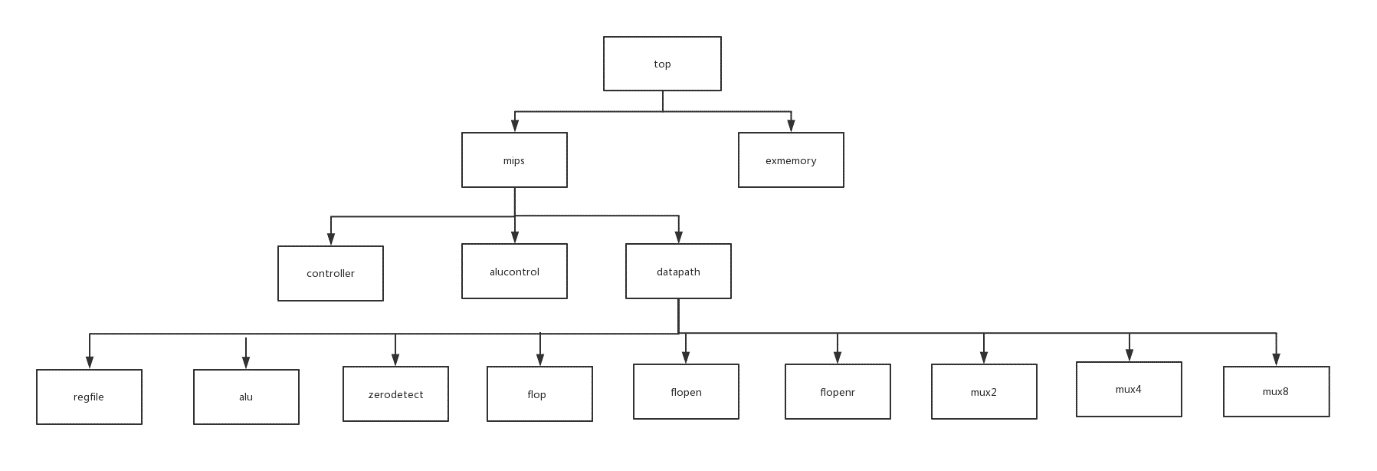
2、jal target

功能：$31<——PC+8; jump to target

四、CPU数据通路图设计

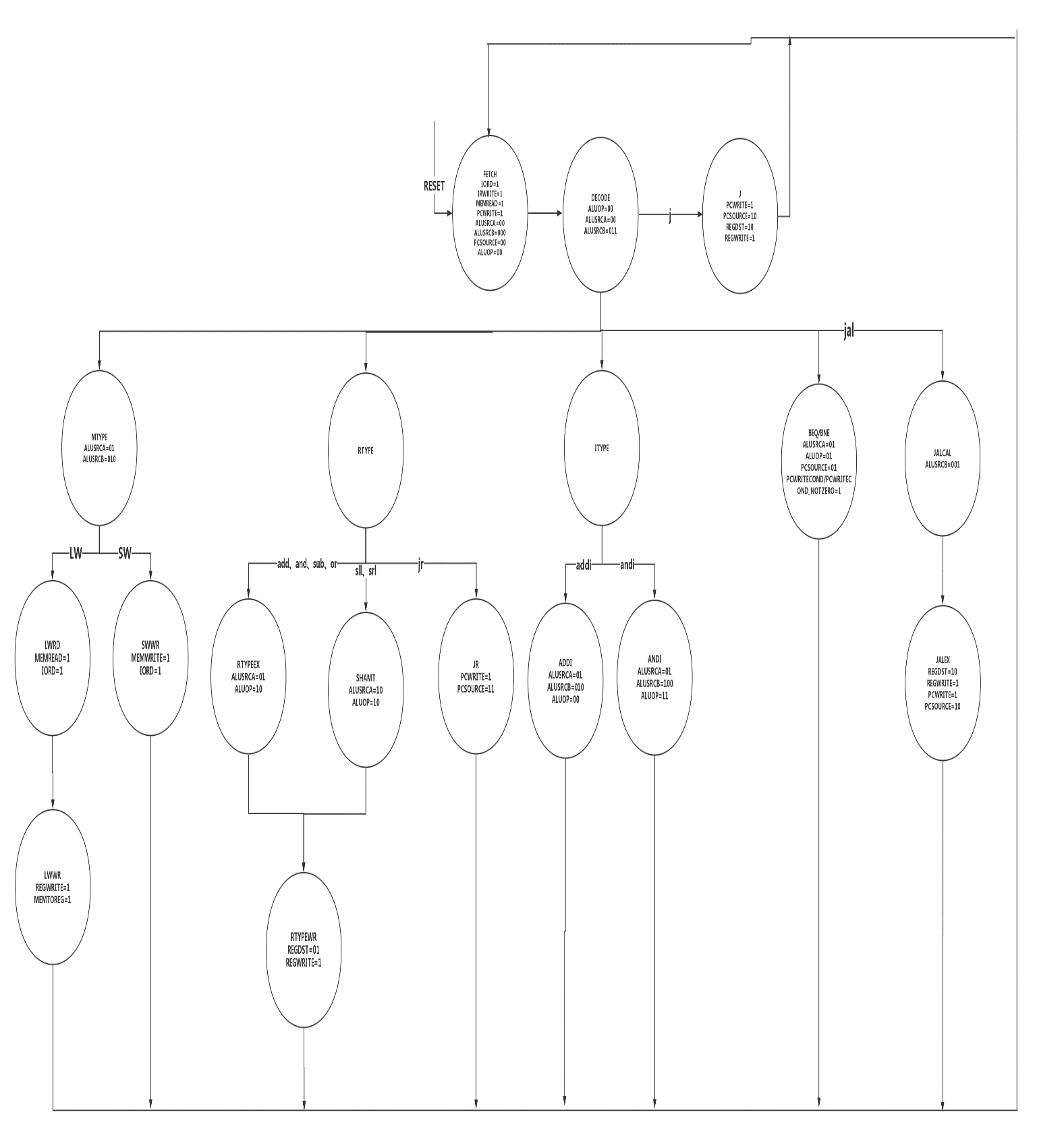
（见压缩包中文件夹“数据通路图”）

五、CPU模块构成层次设计

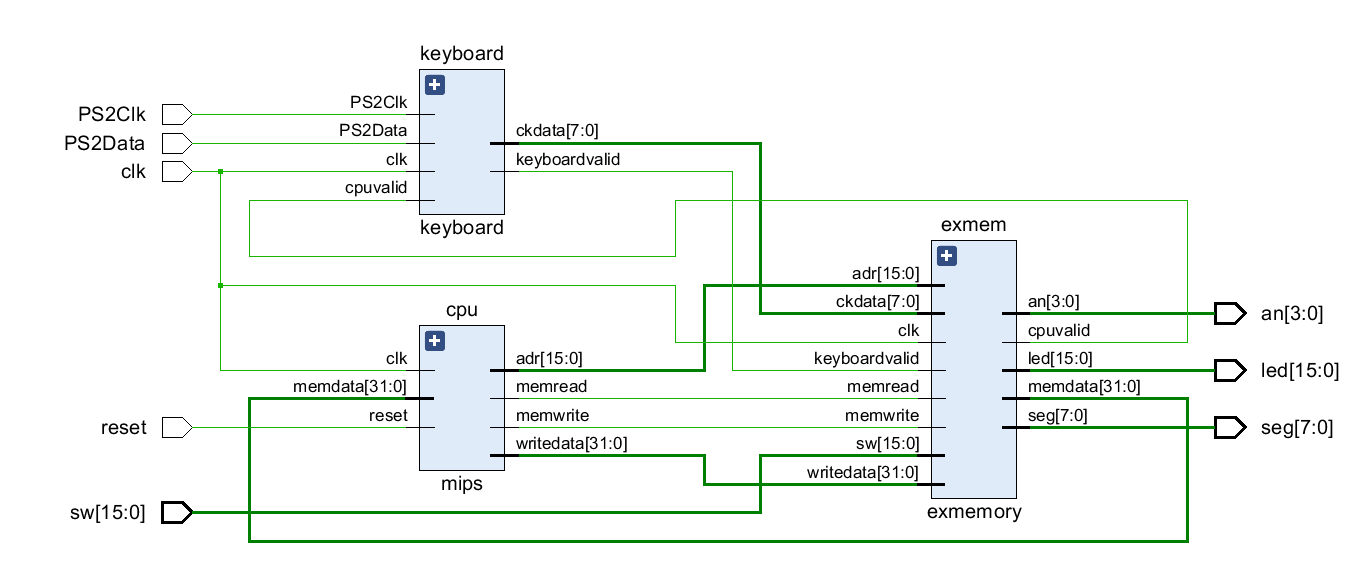
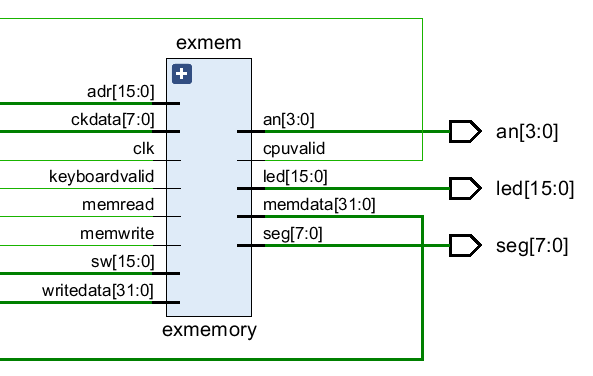


六、多周期操作设计



七、状态转换设计

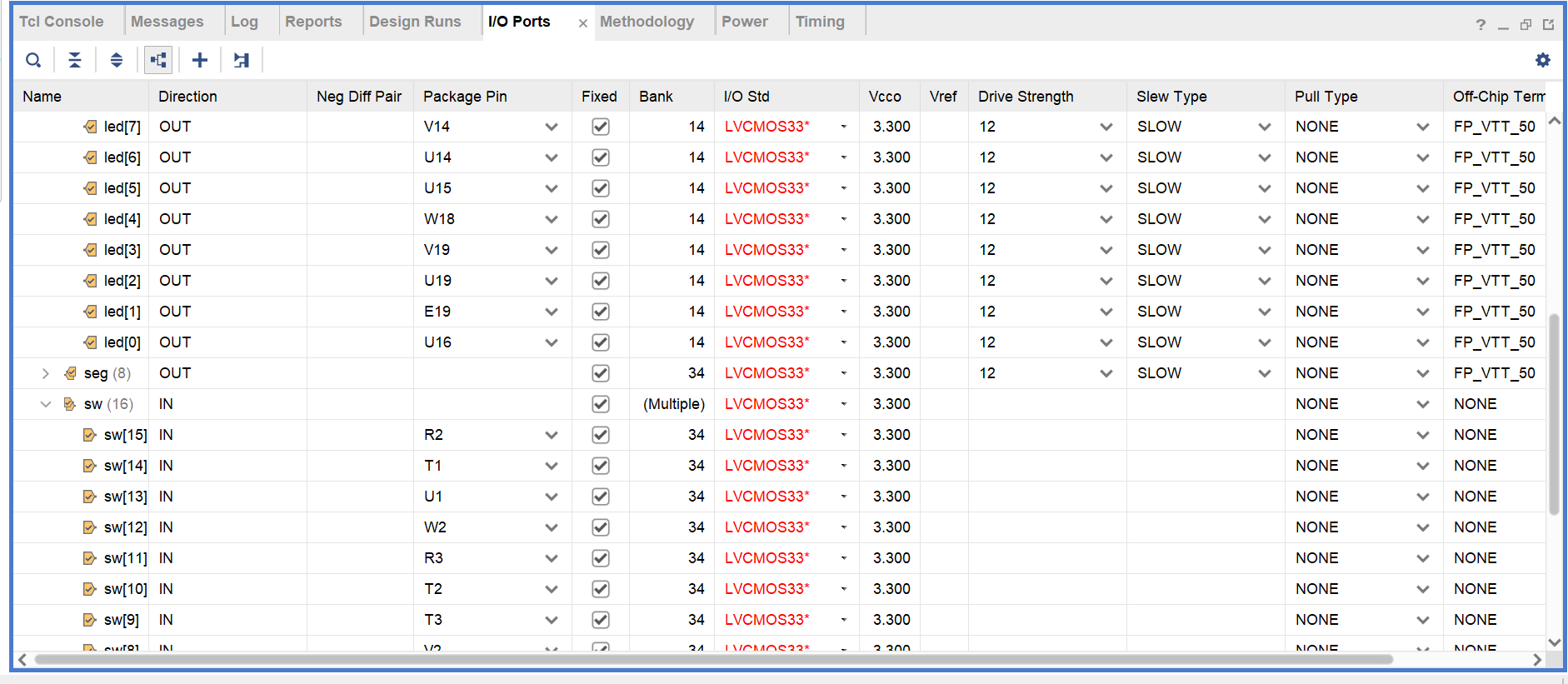
八、SOC设计

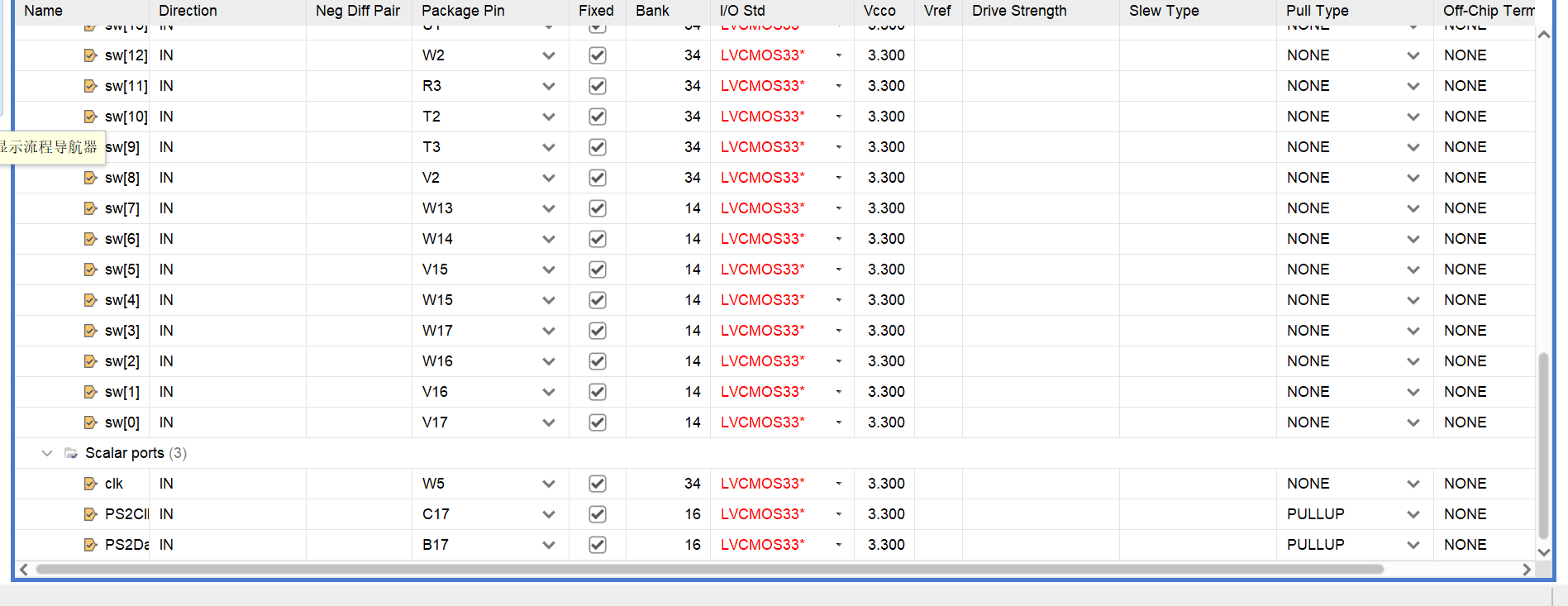
1、外部封装以及存储器设计

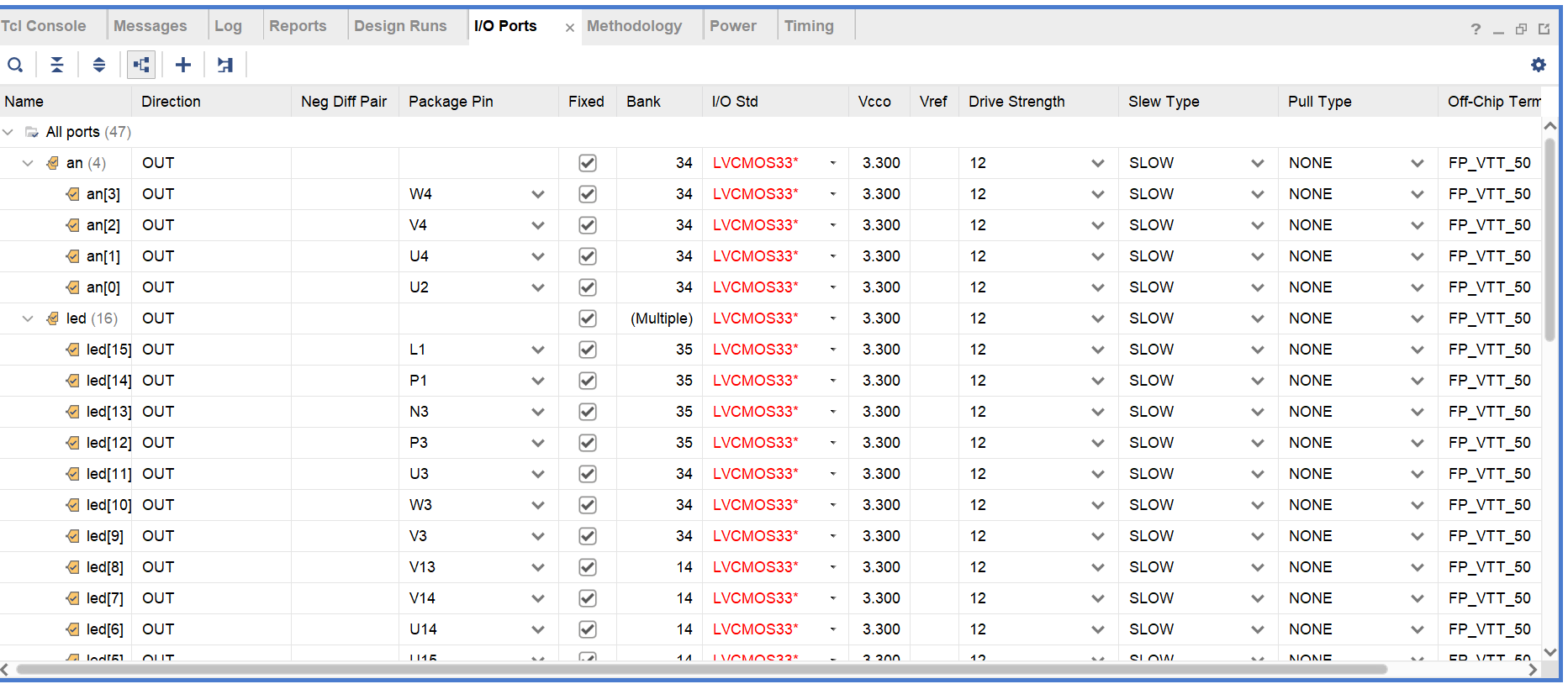
|  |  |  |  |
| --- | --- | --- | --- |
| 信号名 | 位数 | 输入/输出 | 说明 |
| clk | 1 | 输入 | 时钟信号 |
| reset | 1 | 输入 | 复位信号 |
| sw | 16 | 输入 | 拨码开关，作为功能切换的标志 |
| PS2Clk | 1 | 输入 | PS2键盘clock线 |
| PS2Data | 1 | 输入 | PS2键盘data线 |
| an | 4 | 输出 | 数码管位选信号 |
| led | 16 | 输出 | Led灯显示输入字符的ascii码 |
| seg | 8 | 输出 | 数码管段选信号 |

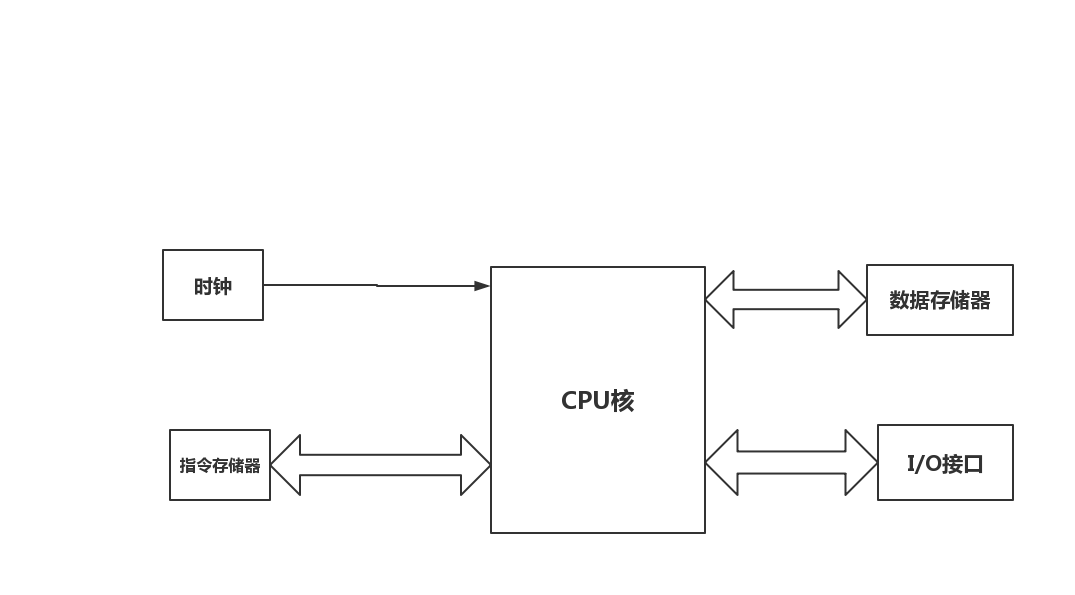
|  |  |  |  |
| --- | --- | --- | --- |
| 信号名 | 位数 | 输入/输出 | 说明 |
| adr | 16 | 输入 | 16位地址输入 |
| ckdata | 8 | 输入 | 键盘数据输入 |
| clk | 1 | 输入 | 时钟信号 |
| keyboardvalid | 1 | 输入 | PS2键盘空闲信号 |
| memread | 1 | 输入 | 存储器读信号，控制存储器读 |
| memwrite | 1 | 输入 | 存储器写信号，控制存储器写 |
| sw | 16 | 输入 | 拨码开关，控制功能的切换 |
| writedata | 32 | 输入 | 写回的数据 |
| an | 4 | 输出 | 数码管位选信号 |
| cpuvalid | 1 | 输出 | CPU空闲信号 |
| led | 16 | 输出 | Led灯显示输入字符的ascii码 |
| memdata | 32 | 输出 | 读出的数据 |
| seg | 8 | 输出 | 数码管段选信号 |

2、引脚配置



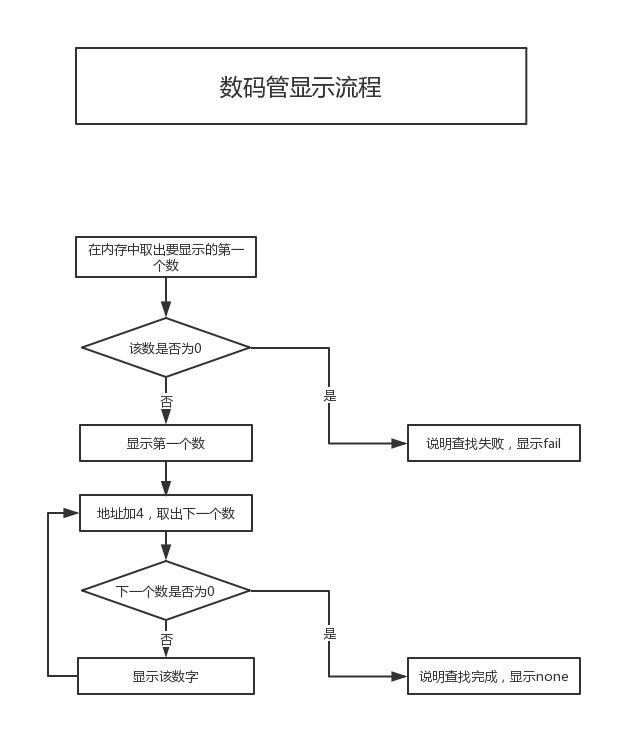




3、内部结构

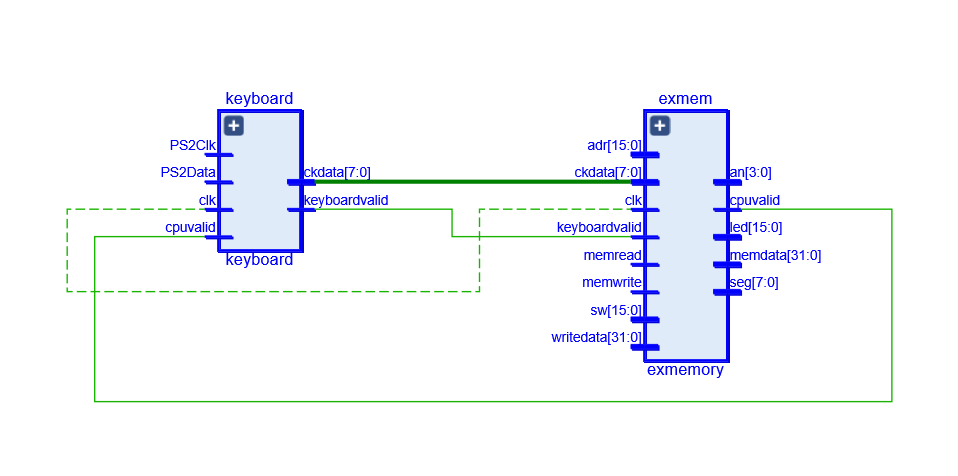
4、数码管设计

本次课程设计中数码管显示部分通过I/O与内存统一编址，使用指令完成对指定地址数值修改的方法来实现数码管的点亮。在内存中段选信号分配的地址为0xfff8，位选信号分配的地址为0xfff9，根据目标数的数值来通过执行指令来实现对这两个地地址所存储数值的修改，然后在exmemory模块中，输出信号seg（段选信号）与an（位选信号）读取该值后，与数码管对应引脚连接，从而实现点亮。点亮流程如下：



注：在实际验证后，发现Basys 3自带时钟频率对于本部分而言过高，无法正常显示，而且本部分没有专门的Verilog模块，不能通过分频来实现降频（若在exmemory中降频讲可能影响其余部分的使用），最终在位选部分加入循环指令，通过循环达到降频延时的效果。

九、外设设计



|  |  |  |  |
| --- | --- | --- | --- |
| 信号名 | 位数 | 输入/输出 | 说明 |
| clk | 1 | 输入 | 开发板时钟信号 |
| reset | 1 | 输入 | 复位信号 |
| PS2Clk | 1 | 输入 | PS2键盘clock线 |
| PS2Data | 1 | 输入 | PS2键盘data线 |
| cpuvalid | 1 | 输入 | cpu对键盘的屏蔽信号 |
| keyboardvalid | 1 | 输出 | 键盘的中断请求信号 |
| ckdata | 8 | 输出 | 键盘数据 |

该键盘接口采用程序查询方式进行工作，CPU使用拨码开关sw[0]作为程序查询程序的入口（即若sw[0] = 1，则CPU进入对键盘设备的查询）。CPU进入查询程序后，不断检查keyboardvalid看键盘的数据是否准备就绪。若keyboardvalid = 1，则

1. CPU首先设置cpuvalid = 1，表示对键盘的屏蔽，键盘接口这时会设置keyboardvalid为0，且不会再发送数据给CPU直到cpuvalid重新置为0。

2. 然后CPU处理ckdata。

3. 最后CPU处理完ckdata后重新设置cpuvalid = 0。

4. 键盘接口又可以发送数据给CPU，CPU回到对键盘准备信号的查询。

十、设计运行结果

（见压缩包中文件加“运行结果”）