​**MIPS单周期实验报告**

罗翔

17307130191

一. 各部件详解...................................................................................................................................................2

1. Control Unit..........................................................................................................................................2
2. ALU.........................................................................................................................................................3
3. RegisterFile..........................................................................................................................................4
4. Memory.................................................................................................................................................4
5. PC............................................................................................................................................................5
6. ImmExtend...........................................................................................................................................6
7. Dispaly...................................................................................................................................................6
8. Datapath...............................................................................................................................................6

二. 展示................................................................................................................................................................7

1. 使用部件................................................................................................................................................7
2. 部件分配................................................................................................................................................7

三. 遇见的问题及解决方法............................................................................................................................7

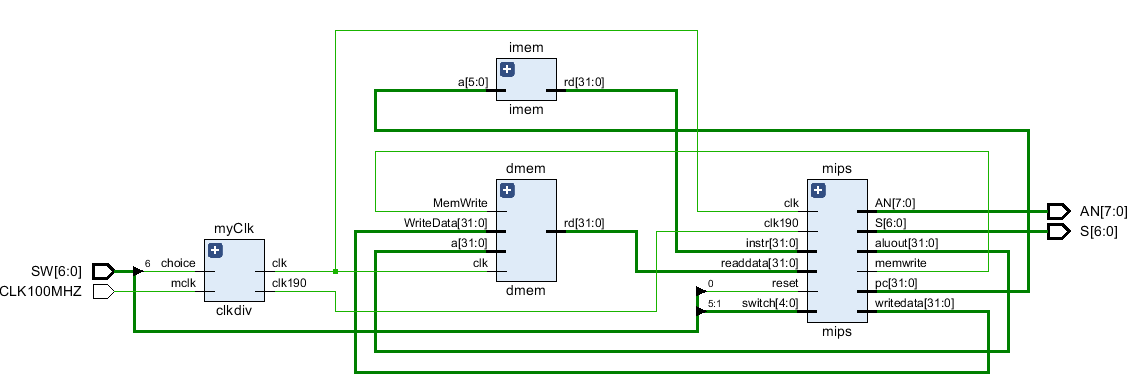
七段数码管闪烁显示...............................................................................................................................7

四．添加指令及模拟测试...............................................................................................................................8

五．参考文献....................................................................................................................................................10

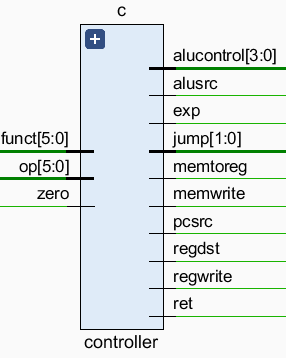
一． 各部件详解

本次MIPS单周期的总体框架如下：



mips为CPU核心，包括控制单元(controller)与数据通路(datapath)；imem为指令存储器,为CPU提供指令；dmem为数据存储器,为CPU提供数据存储空间；myClk为时钟分频器，其中clk为CPU主体的时钟信号，clk190为七段码显示的扫描时钟信号。

1. Control Unit



**功能说明：**

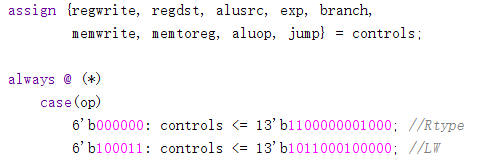
解析指令，根据不同指令设定不同的控制线。需要提的是，branch为两根线分别对应的beq和bne，jump也有两根线分别对应的jump和jal。

设计思路：

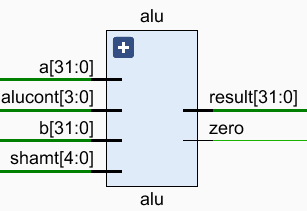
控制码真值表如下：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 操作 | 类型 | op | funct | regwrite | regdst | alusrc | exp | branch | memwrite | memtoreg | aluop | jump |
| add | R-Type | 000000 | 100000 | 1 | 1 | 0 | 0 | 00 | 0 | 0 | 010 | 00 |
| sub | R-Type | 000000 | 100010 | 1 | 1 | 0 | 0 | 00 | 0 | 0 | 010 | 00 |
| and | R-Type | 000000 | 100100 | 1 | 1 | 0 | 0 | 00 | 0 | 0 | 010 | 00 |
| or | R-Type | 000000 | 100101 | 1 | 1 | 0 | 0 | 00 | 0 | 0 | 010 | 00 |
| slt | R-Type | 000000 | 101010 | 1 | 1 | 0 | 0 | 00 | 0 | 0 | 010 | 00 |
| addi | I-Type | 001000 |  | 1 | 0 | 1 | 1 | 00 | 0 | 0 | 000 | 00 |
| andi | I-Type | 001100 |  | 1 | 0 | 1 | 0 | 00 | 0 | 0 | 011 | 00 |
| ori | I-Type | 001101 |  | 1 | 0 | 1 | 0 | 00 | 0 | 0 | 100 | 00 |
| slti | I-Type | 001010 |  | 1 | 0 | 1 | 1 | 00 | 0 | 0 | 101 | 00 |
| sw | I-Type | 101011 |  | 0 | 0 | 1 | 1 | 00 | 1 | 0 | 000 | 00 |
| lw | I-Type | 100011 |  | 1 | 0 | 1 | 1 | 00 | 0 | 1 | 000 | 00 |
| j | J-Type | 000010 |  | 0 | x | x | x | xx | 0 | x | xxx | 10 |
| nop | R-Type | 000000 | 000000 | 0 | 0 | 0 | 0 | 00 | 0 | 0 | 000 | 00 |
| beq | I-Type | 000100 |  | 0 | 0 | 0 | 0 | 10 | 0 | 0 | 001 | 00 |
| bne | I-Type | 000101 |  | 0 | 0 | 0 | 0 | 01 | 0 | 0 | 001 | 00 |
| jal | J-Type | 000011 |  | 0 | x | x | x | xx | 0 | x | xxx | 11 |
| sll | R-Type | 000000 | 000000 | 1 | 1 | 0 | 0 | 00 | 0 | 0 | 010 | 00 |
| srl | R-Type | 000000 | 000010 | 1 | 1 | 0 | 0 | 00 | 0 | 0 | 010 | 00 |
| sra | R-Type | 000000 | 000011 | 1 | 1 | 0 | 0 | 00 | 0 | 0 | 010 | 00 |
| jr | R-Type | 000000 | 001000 | 1 | 1 | 0 | 0 | 00 | 0 | 0 | 010 | 00 |

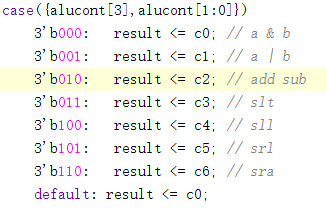
通过将控制线并列、case语句分类的方式，可以十分简练的完成对不同类型指令控制码的确定。



1. ALU

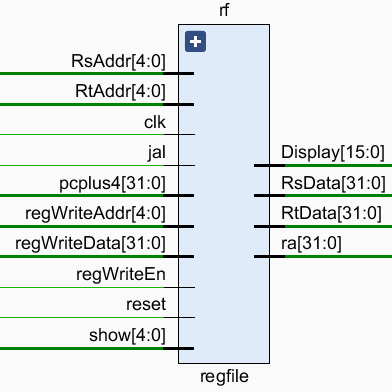


ALU是CPU中的运算部件，可以支持加减、移位和位运算等操作。其中ALU的运算结果主要由{ alucont[3], alucont[1:0] }决定，



而alucont[2]决定是否对b取反，以及作算术运算时是否有进位carry。其中shamt是移位数，zero是零符号位。

1. RegisterFile



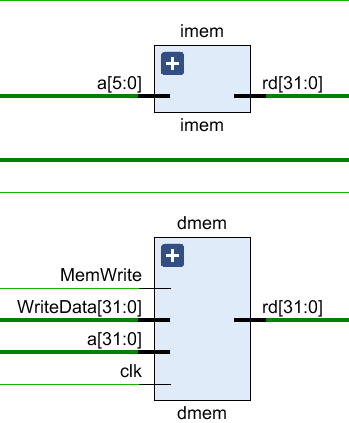
**功能说明：**

为满足CPU高速运行的寄存器，置于芯片内部，通常由快速的静态随机读写存储器实现。且由于造价高昂，所以需要控制可存储空间。

主要用于从寄存器中读写数据，其中读过程为组合逻辑；而写过程为时序逻辑，必须在下一个时钟周期到来时才能写入。

jal是jump and link的标志位。jal为1，PC会跳转至目标地址并且将PC+4存储在$ra(31)中，jal与jr配合可以实现call（调用函数），再加上$sp可以实现栈（stack）操作。向外输出$ra(31)是为了在执行jr指令时，告知PC下一步执行的指令的地址。

1. Memory

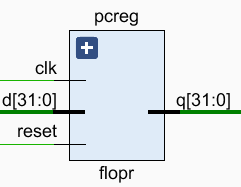


**功能说明：**

dmem为数据存储器，主要是从存储器中读写数据信息，其中读过程可视作组合逻辑，而写过程为时序逻辑。实际运用中从存储器中读写数据的速度远远慢于从寄存器中读写数据，但是存储器的存储空间非常大。

imem为指令存储器。因为MIPS存储器模型是字节寻址且指令编码则是32位的一个字，所以每条指令的地址都是4的倍数，因此每次取值可以直接将PC前30位（即除以4后）传入imem。

1. PC



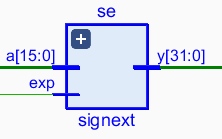
**功能说明：**

用于指示下一条指令的地址，即在时钟信号到来时更新下一条指令的PC地址，使CPU执行下一条指令。

PC一般为顺序执行，即下一条指令的地址一般为PC + ‘1’。但也存在转移执行的情况，比如分支指令beq、bne和跳转指令j。

其中分支指令需要将给定的16位（目标指令与现有指令的偏移量）进行符号扩展加上现有地址得到。而跳转指令则是将给定的26位伪地址头部加上现有指令地址的前4位后左移2位得到实际地址。{PC [31:28], instr [25:0], 2’b00}

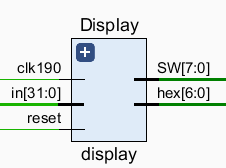
1. ImmExtend



**功能说明：**

由于I型指令的立即数为16位，而实际运算时为32位，所以需要对其进行扩展，其中I型指令中的逻辑运算应为零扩展，而其他I型指令均为符号扩展。所以需要一个控制位决定扩展方式。

1. Display



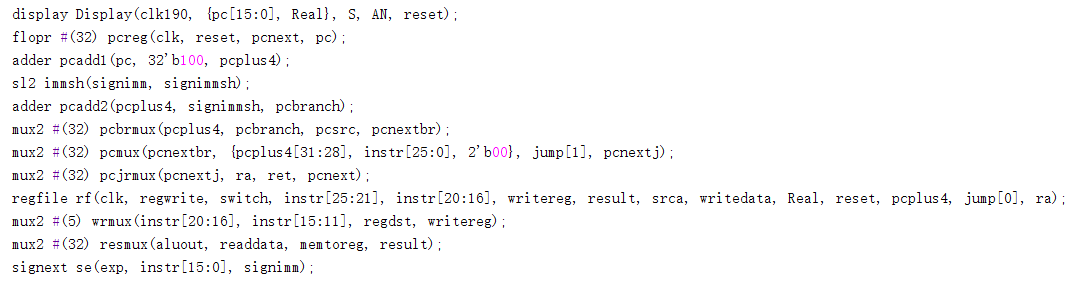
**功能说明：**

因为需要将观察的内容显示在七段数码管上，所以需要一个Display的元件。并且考虑到要同时在8个七段数码管上显示内容，需要新增一个更快的时钟信号来扫描，利用人眼的视觉限制造成8个数码管同时显示的假象。

1. Datapath

**功能说明：**

数据通路通过实例化各元件，将整个CPU串接起来。



二． 展示

1. 使用部件

七段数码管 \* 8、 开关 \* 7

2. 部件分配

（1） 开关1： reset，使pc、regfile、memory复位。

（2） 开关2 – 6： 选择显示的register。

（3） 开关7： 速度选择器（时钟周期T = 0.16s / 1.32s）。

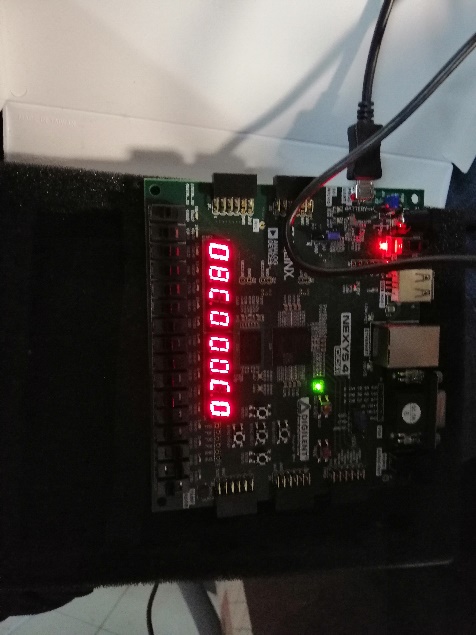
（4） 数码管1 – 4： 显示当前选定的register内容。

（5） 数码管5 – 8： 显示当前pc值。

三． 遇到的问题及解决方法

七段数码管显示问题：

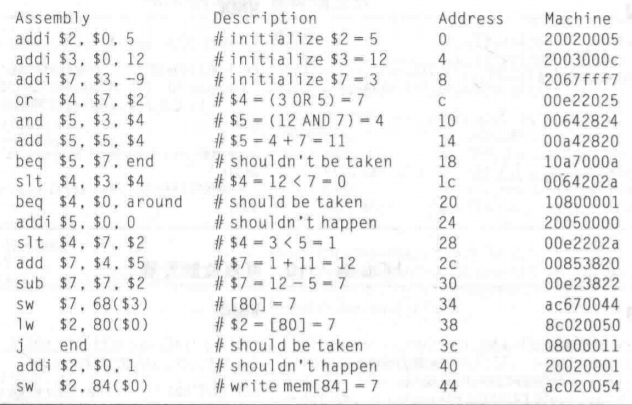
因为要同时在8个数码管上显示内容，故需要扫描8个数码管。考虑人的眼睛的反应周期在0.05s左右。最开始我通过分频设置的扫描周期为q[18](0.005s)，此时七段数码管闪烁明显，考虑到应该是扫描周期过长，使得肉眼观测到了扫描过程，我又将扫描周期置为q[16](0.00125s)，则七段数码管正常显示，没有明显的闪烁。但是在制作过程中，我还遇见了本该正常显示的pc和register，在七段数码管上集体向左偏移一位的情况，如下图。后发现，我的Display元件always语句中全为非阻塞赋值，因此每次被扫描的数码管显示的都是上一个数码管应该显示的内容（即右侧数码管本该显示的内容），故造成了整体左移的情况。



四．添加指令及模拟测试

首先测试基本功能，使用书上的代码进行随机测试：

具体指令如下：

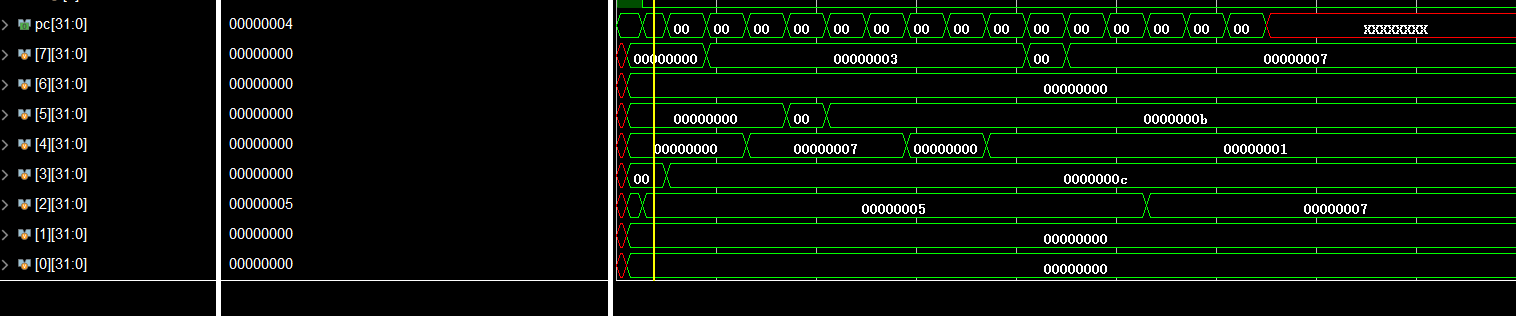


其中$2: 5 → 7

$4: 7 → 0 → 1

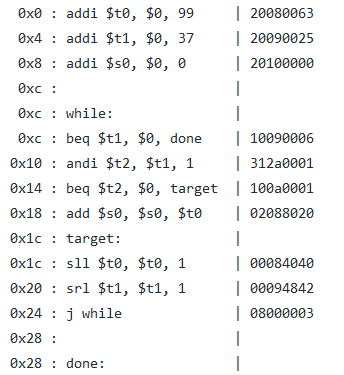
$5: 4 → 11

$7: 3 → 12 → 7



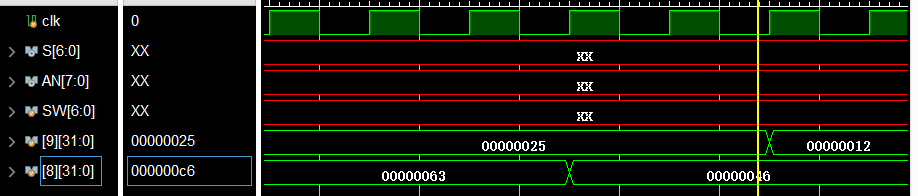
接下来测试添加的srl、sll、beq、j指令，使用的是张作柏同学在github上的代码:

其中指令如下

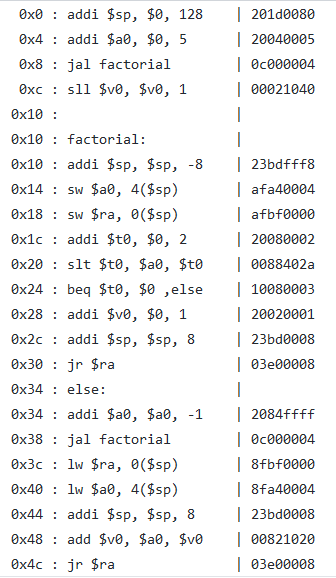


$t0($8): 99 → 198(0x63 → 0xc6) t0 逻辑左移1位

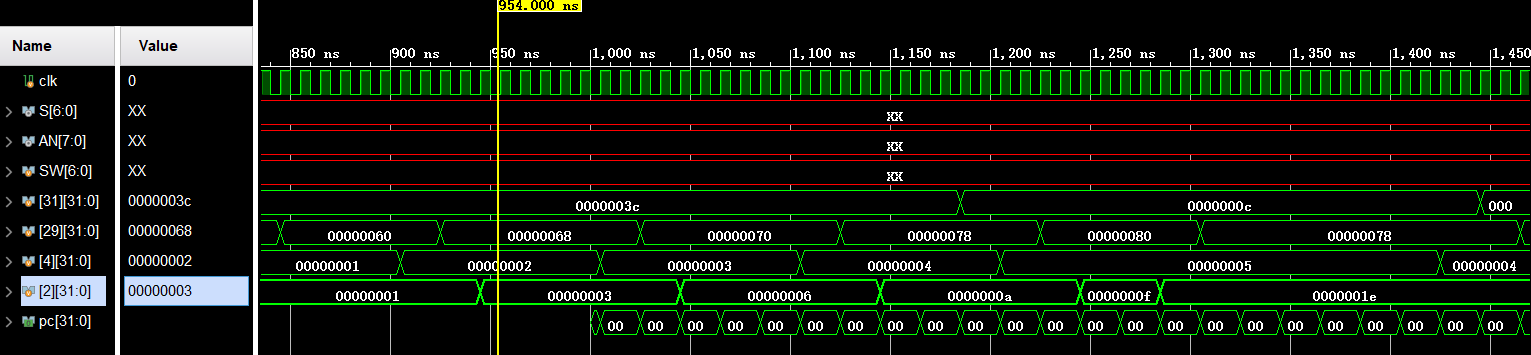
$t1($9): 37 → 18(0x25 → 0x12) t1 逻辑右移1位



接下来测试添加的jal、jr指令，使用的同样也是张作柏同学在GitHub上的累加代码：



$v0($2) 1 → 3 → 6 → 0xa → 0xf



五．参考文献

1. 戴维·莫尼·哈里斯，莎拉 L. 哈里斯. 数字设计和计算机体系结构（原书第2版）

2. https://github.com/Oxer11/MIPS