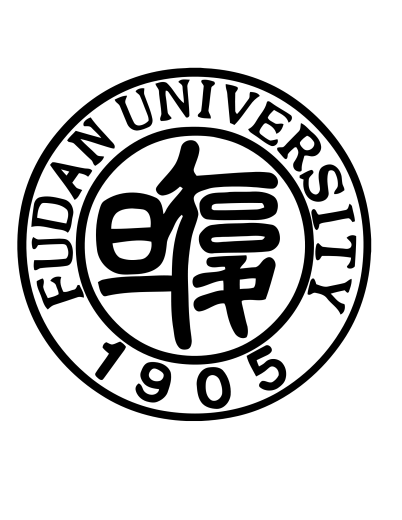
**计算机组成与体系结构**

**实验报告**

**实验2 【32位MIPS多周期处理器设计】**

****

指导教师： 孙晓光

|  |
| --- |
| 学生姓名： 李钧 |
| 学 号：20307130135 |
| 专 业： 信息安全 |

日 期： 2022.4.24

一、实验目的

1、掌握多周期CPU数据通路图的构成、原理以及设计方法

2、学习掌握多周期CPU的实现方法、代码实现方法

3、学习掌握测试多周期CPU的方法

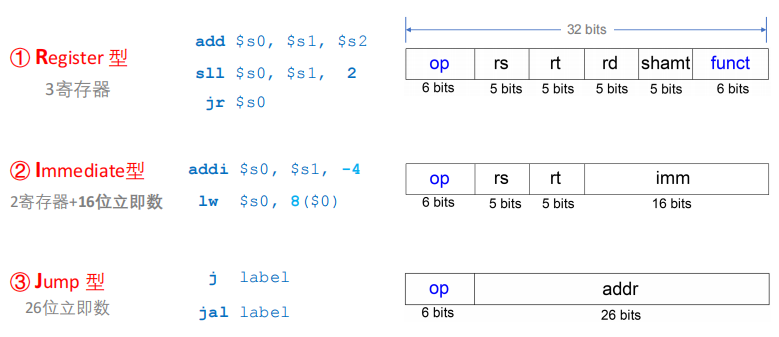
4、在完成任务的过程中感悟百折不挠的可贵，以及一丝不苟的黄金精神

二、实验原理

多周期CPU和单周期CPU的最大不同就是，多周期CPU在一个时钟周期之内有很多个部件可以同时运作，而一个指令（比如lw指令）可以分在多个时钟周期完成处理；单周期CPU则是限制了要在一个时钟周期内完成一个指令，这就得根据最大处理时长的lw指令来确定时钟周期，使得处理效率较低。

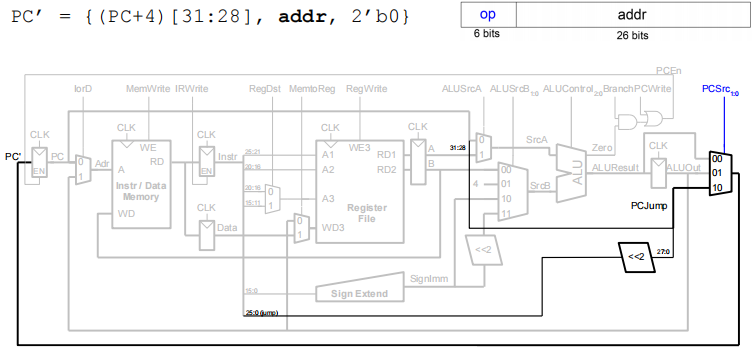
除此之外，单周期CPU需要三个加法器来实现需要的功能，而且单周期CPU需要将指令存储器和数据存储器独立放置，这将会占用较多的芯片面积、增加读写内容的时间消耗。多周期只需要使用一个加法器、一个存储器（指令+数据），另外需要多五个寄存器、两个MUX选择器，相比于单周期CPU可以节省更多面积以及时间。

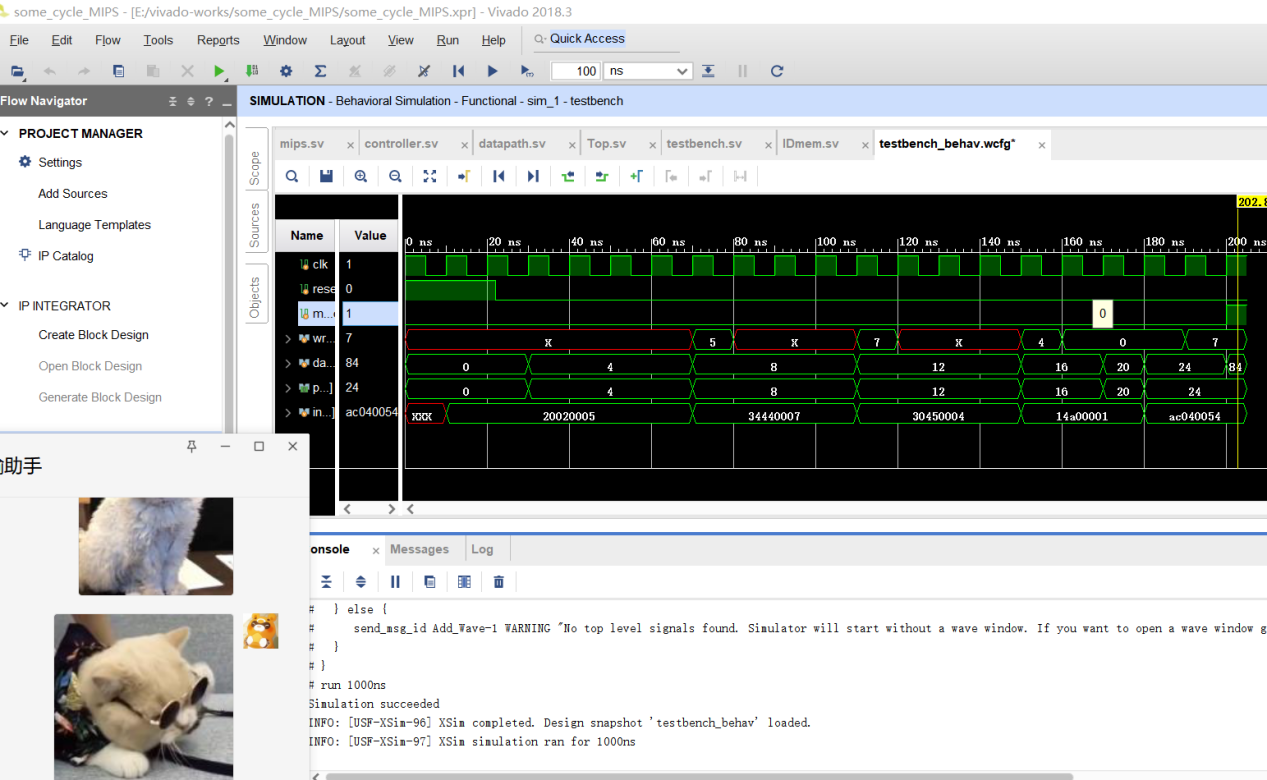
多周期CPU的实现原理和单周期CPU相差无几，指令处理的格式也分为几个类型：

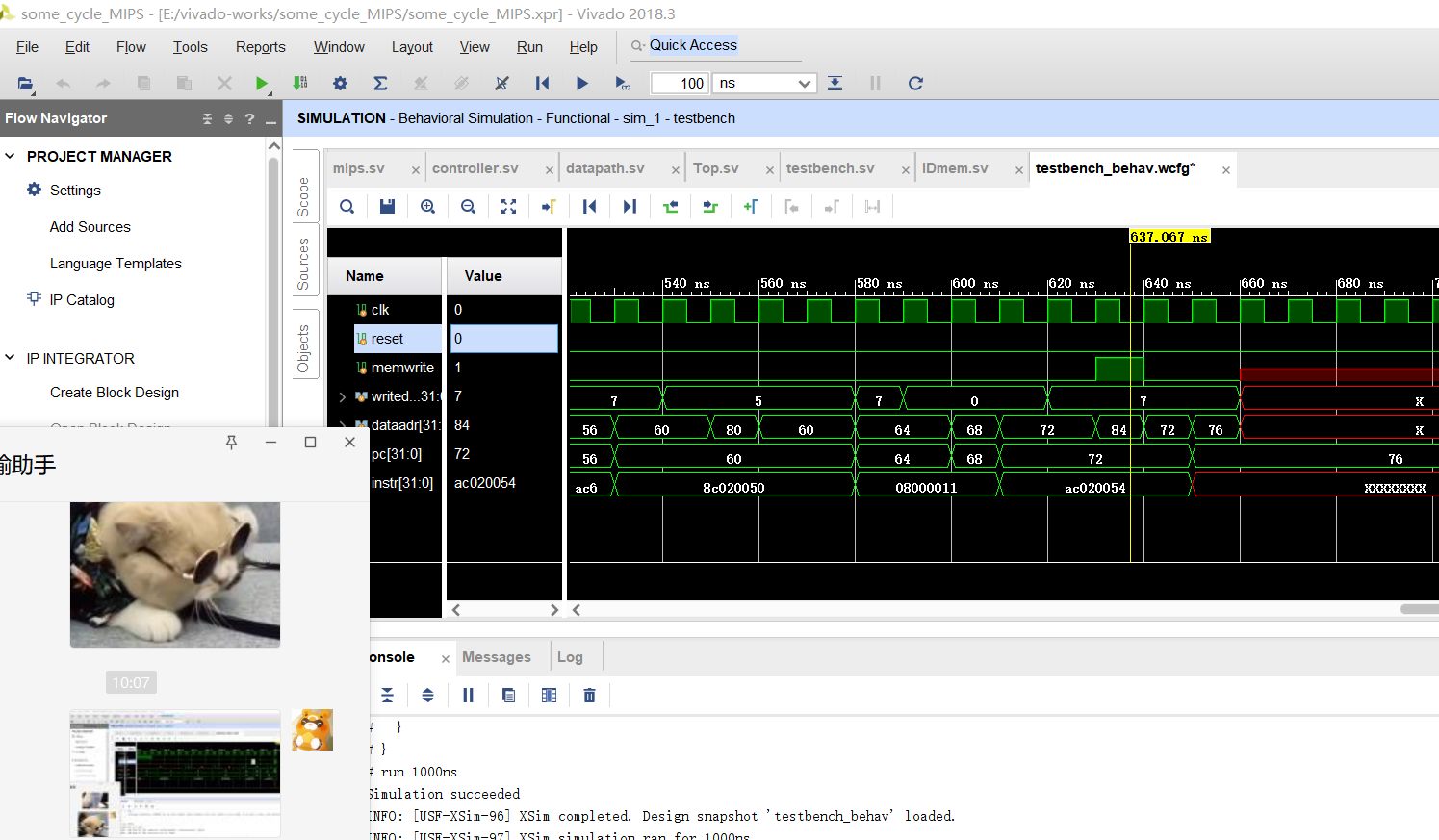


三、实验内容

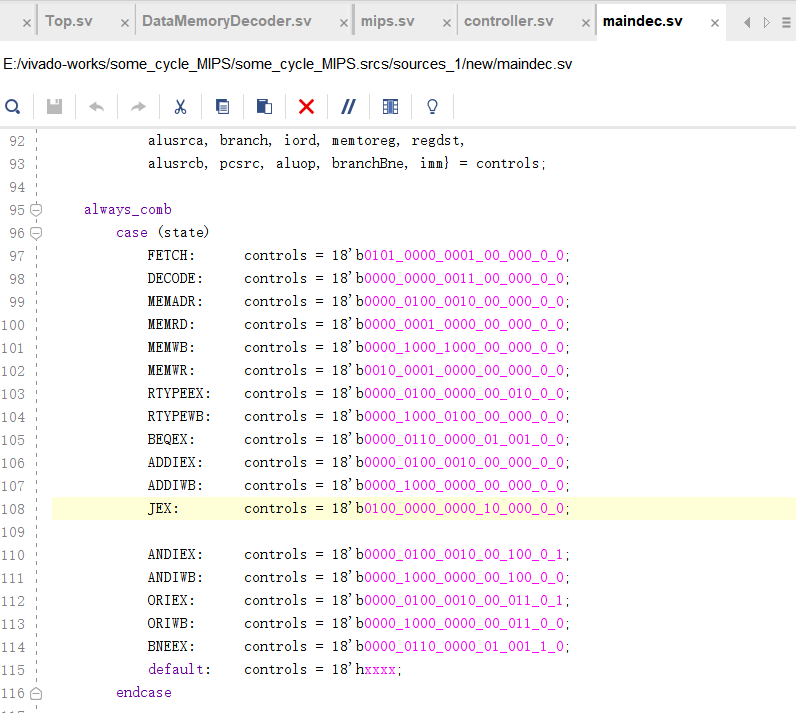
首先我们需要根据书上的说明介绍，参考单周期CPU的部分元器件代码内容将整个项目大致建立起来，在写TOP文件之前先把各个小部件的实现代码编写好以便我们的调用。 然后根据下图进行各元件之间的连线，当然了，在一开始我们没有考虑很多扩展指令，只是将最基本的代码设计好并模拟运行，下面附上头两次模拟成功的图



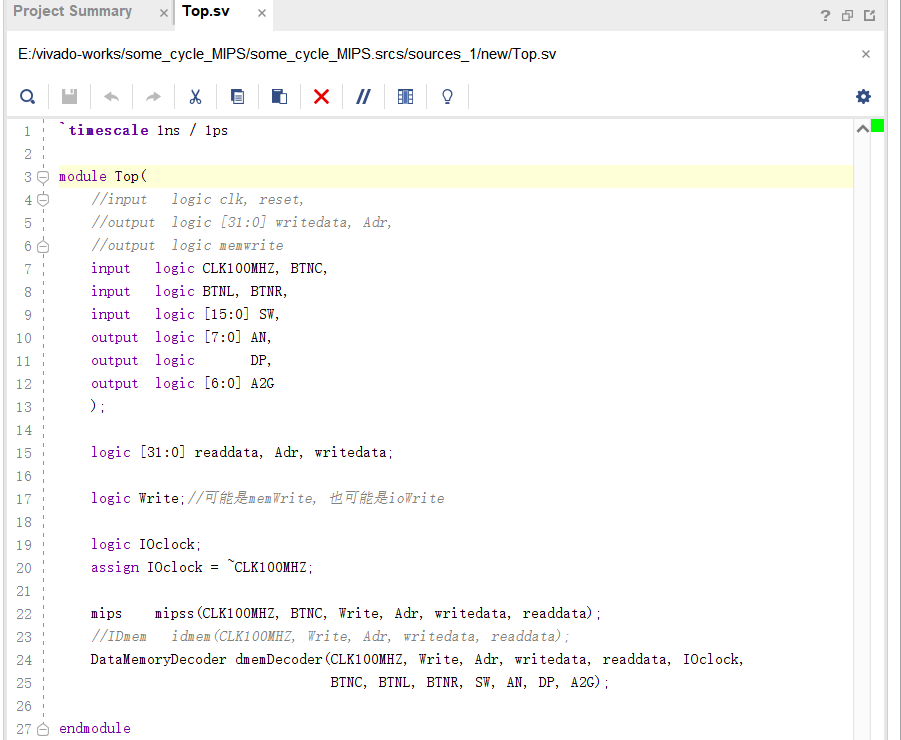




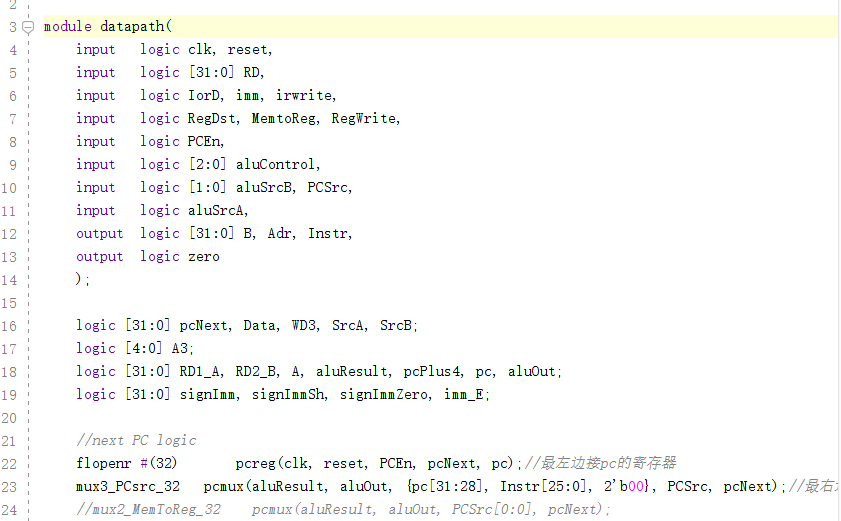
1、FSM状态机的关键代码我认为是最容易出问题的，PPT上给出的代码中这一部分内容是用16进制书写的，我将他翻译成二进制之后再进行拓展的时候就很容易看错看漏。为了解决这一部分的问题，我花了最多的时间。

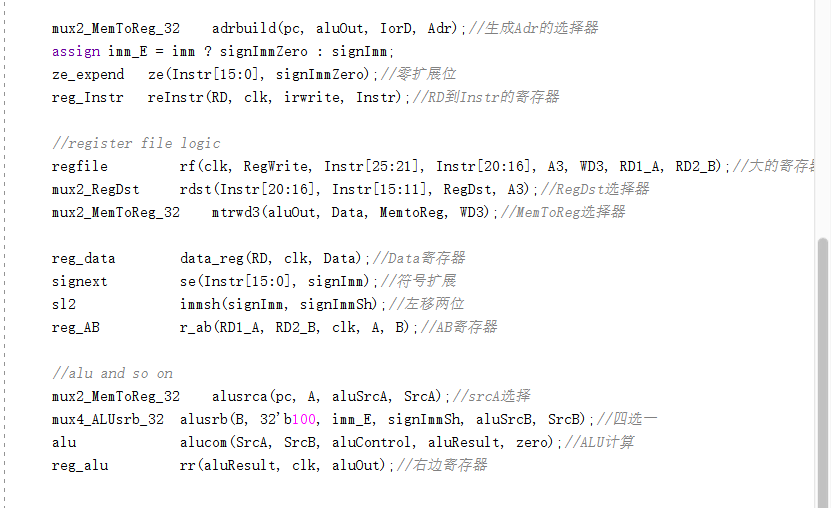


2、头文件TOP统领全局，和单周期CPU有所不同，给出截图如下：



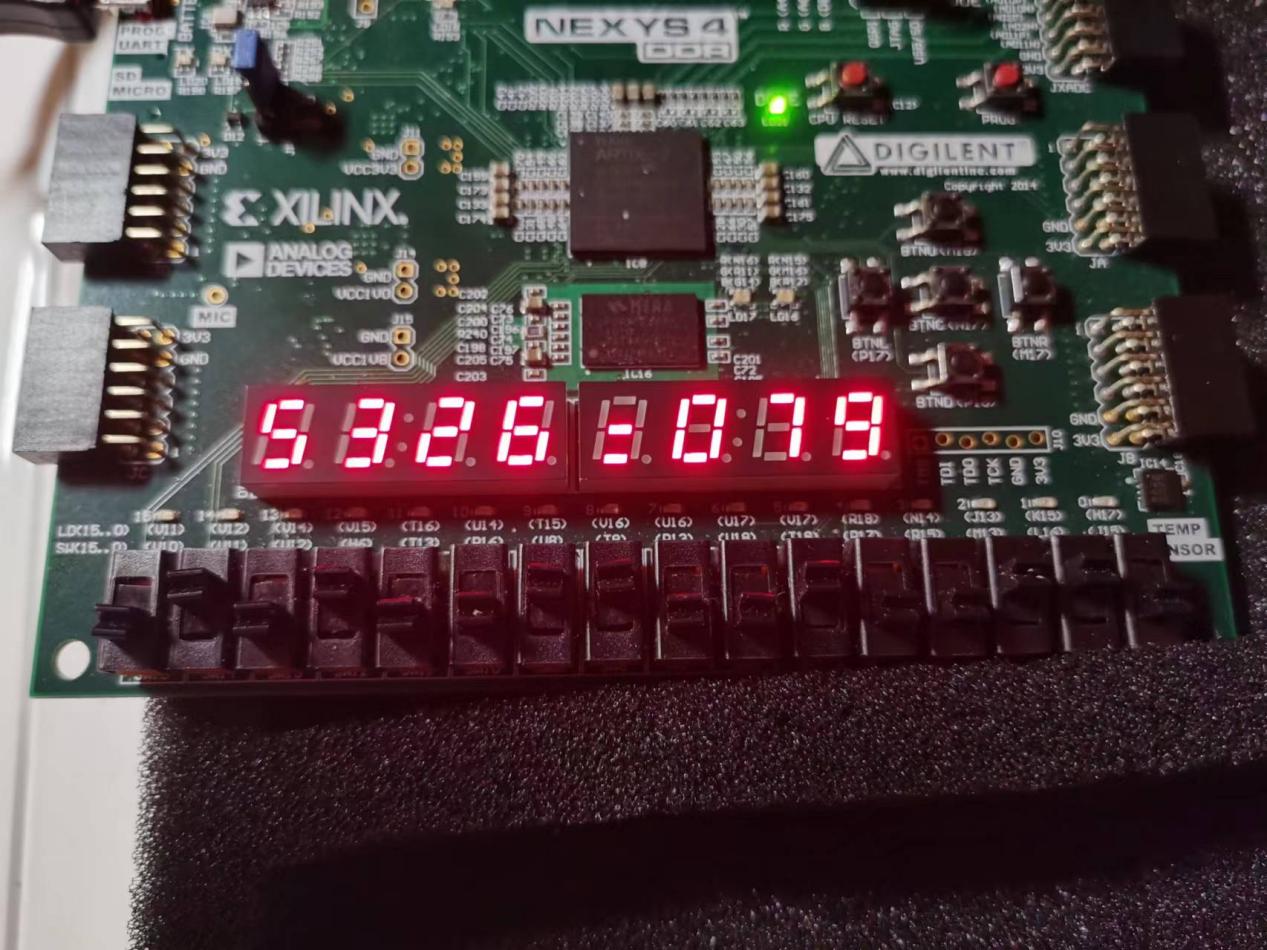
3、多周期CPU的数据路径使用了较多的寄存器、选择器，各个输入输出线也比较容易出错，在编写代码的过程中也使用了很多变量，给出截图如下：





4、在经历了颇为痛苦的找错改错过程之后，我终于还是成功地上板子了：





四、心得体会

相比于做单周期CPU时候的陌生感，本次任务多了几分熟悉，但是正由于这份熟悉，我小瞧了艾宾浩司遗忘曲线的威力——写一半之后就做其他事情去了，几天之后再看自己之前写的代码就看不懂了。这直接导致了我书写的代码比较乱，之后寻找错误更困难。

说说添加IO前后的修改吧，上面我提到的FSM状态机的代码，在添加IO、添加扩展指令之前是没有任何问题的，但是添加扩展指令之后为了适应新的指令，我不得不进行添加和修改。于是乎在这之后的第一次仿真便顺理成章地失败了。没错接下来就是一股脑埋头在代码里面找错——当然，失败了，找不到逻辑上的问题。鏖战一段时间之后，终于顺着仿真出来的线索一步一步找到FSM当中数组开的太小了、10写成了01这样的错误。过程是这样的：发现PC值一直在０和４之间跳转——发现PCEN一直为零——改一点之后发现PC值到了他本不该到的一个值——修改FSM状态机中JUMP的值——解决问题。

认真仔细、一丝不苟、百折不挠，真的是很重要的品质。