**单周期CPU分工**

**一、Introdution**

为了实现模块化的效果，将每一部分单独写成一个文件，尽量减少各文件之间的关联。

**二、Module divided**

（X）1.ALU（任务一已实现）

（X）2.Data Memory，用来储存程序运行过程中的数据

输入：读写控制，数据输入

输出：数据输出

工作量：1

（C）3.ROM，储存指令，大小需要自己确定。该部分还需要将汇编代码写出并编译成机器码存入ROM

输入：只有读地址

输出：只有读出的数据

工作量：8

（X）4.Peripheral，包括定时器、开关、LED三个外设。

输入：数据地址等

输出：中断LED开关等控制信号

工作量：6

（X）5.UART（春季学期已完成）

需要对原来的代码进行整合

工作量：3

（X）6.RAM，包含Peripheral、Data Memory、UART三个部分，组成整体的随机访问存储器。

输入：所有与外设有关的信号

输出：外设输出

工作量：1

（D）7.RegisterFile，寄存器堆

输入：读写控制和数据输入

输出：数据输出

工作量：4

（D）8.Control，根据instruction[31 : 25]和function码产生控制信号

输入：instruction[31 : 25]

输出：大量控制信号和ALUFun

工作量：8

（X）9.Adder，加法器，在电路中需要两个额外的32位加法器实现地址的计算

输入：两个被加数

输出：加法结果

工作量：2

（D）10.MUXChooseSignal，对于电路中需要多个源的输入（控制信号控制）的地方需要使用多种选择器。

还包括拓展方式的选择。

还包括移位寄存器的实现。

输入：多个控制信号，多个源信号

输出：当前应该产生的输入信号

工作量：5

（D）11.CLOCK，用于产生时钟分频信号

**三、Use GITHUB or not？**

**四、Pipeline debug**

需要先画出流水线的数据通路图