

## 单周期CPU主要包含以下几个模块：

PC（程序计数器）：通过时钟跳电平来使寄存器或者计数器计数，以这种方式来搜索指令的地址（其中可包含自增4模块，怎么装主要看寻址方式）

IM（指令储存器）：储存指令，一般用ROM，左A端输入指令的地址，右D端输出对应指令的机器码

Splitter（分位器）：将IM传出的机器码分解，从而获得OP，FUNCT等子码

(G)RF：寄存器堆，主要功能是将输入的寄存器地址的数据导出到RD1，RD2，并且可通过写入的使能向地址为A3的寄存器写入数据WE

ALU（运算器）：一个包含许多功能的运算器，功能的选择可用多路选择器来实现

Controller（主控制器）：通过与或门的方式来将OP和FUNCT两个六位码转化为各种信号，不同的OP和FUNCT转出信号不完全相同

Bitextender（增强版的位扩展器）：因为不同的指令对于立即数扩位的要求不同，所以需要有一个模块来控制是有符号输出还是补零输出

DM（数据储存器）：储存数据，一般用RAM，左A端输入存储的地址，通过Store和Load判断是读取数据还是存入数据，左D为存入的数据，右D为读出的数据

## 可能有的难点：

1、PC程序计数器不会设计，经常出现电路震荡或者电平滞后的错误：首先可通过一个寄存器循环自增来完成自增4模块，可以实现跳到高电平地址增加的功能。P3线下一共有三种不同的寻址，电平自增4，beq跳转，和j跳转可以将这三种地址放入多路选择器中然后通过Controller的信号来判断进行哪种寻址。

另外，初始地址（复位）的设置可参考这个网址的内容：

[【Logisim】带有初始化功能的寄存器 - Elucidator\\_xrb - 博客园 \(cnblogs.com\)](http://cnblogs.com/Elucidator_xrb/)

可以通过给计数器设置最大值1来控制多路选择器，从而达到初始值的效果

2、Controller主控制器模式的选择和输出信号的种类分不清：Controller相当于一个离散化的过程，将两个六位码OP和FUNCT转化为数量较少的信号，从而可以更加简单的处理机器码。Controller设计采用与或门的形式比较好理解，并且输出信号也很清晰，主要输出的信号有：MemRead，MemWrite，Branch，ALUSrc，EXTOp，ZERO，RegWrite等，MemRead主要判断LW，是否从DM中读数据；MemWrite主要判断SW，是否向DM中写数据；PCSEL个人用来判断j，是否有跳转；RegDst判断R指令，是否用ALU进行R指令计算；EXTOp判断i指令，是否用ALU进行立即数计算；Branch判断B指令，是否用ALU进行比较运算；ZERO判断比较运算的结果；ALUSrc判断向ALU中传入的第二个数据是什么（比如ori传入立即数，sw，lw传入偏移量等）。为了方便判断，还可以自己设置其他信号

3、连线一定缕清思路，如果觉得很乱的话可以用隧道tunnel连接，调试代码的时候可以用探测器Probe进行线路测试主要测试存储是否有效，若无效并且main连接没有什么问题的话可以看看Controller和ALU是否连错线

## 输出解释：

Instr: IM输出的机器码

RegWrite: 是否向寄存器堆中写入数据 (RegDst)

RegAddr: 向什么地址的寄存器中写入数据 (A3)

RegData: 向寄存器中写入了什么数据数据 (WD)

MemWrite: 是否向DM中写入数据

MemAddr: 向DM的什么地址存入数据 (RAM的A端)

MemData: 向DM中写入了什么数据 (RAM左D端)

另：

- 1、示例图助教说尽量不放；
- 2、gitee和github上有前辈的CPU可参考；
- 3、CPU最好分模块写，否则太乱debug很难
- 4、可能字写得比较密了QAQ