P3.logisim设计单周期CPU

一、实验目的

使用logisim设计一个能执行七条指令的单周期CPU。

二、设计要求

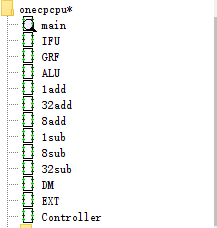
1.CPU使用logisim设计。

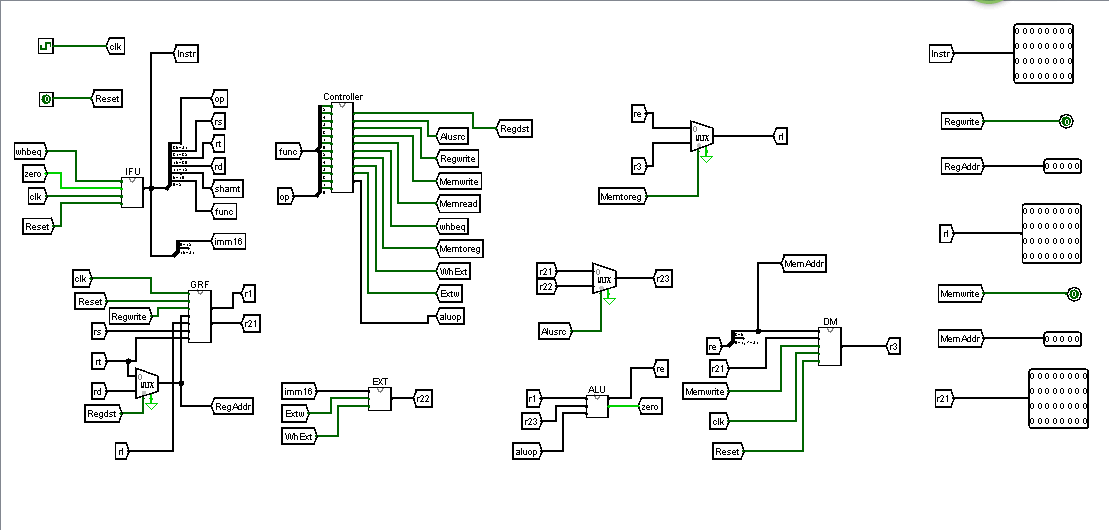
2.CPU能支持addu、subu、ori、lw、sw、beq、lui、nop指令。addu、subu 可以不支持实现溢出。

3.CPU为单周期设计

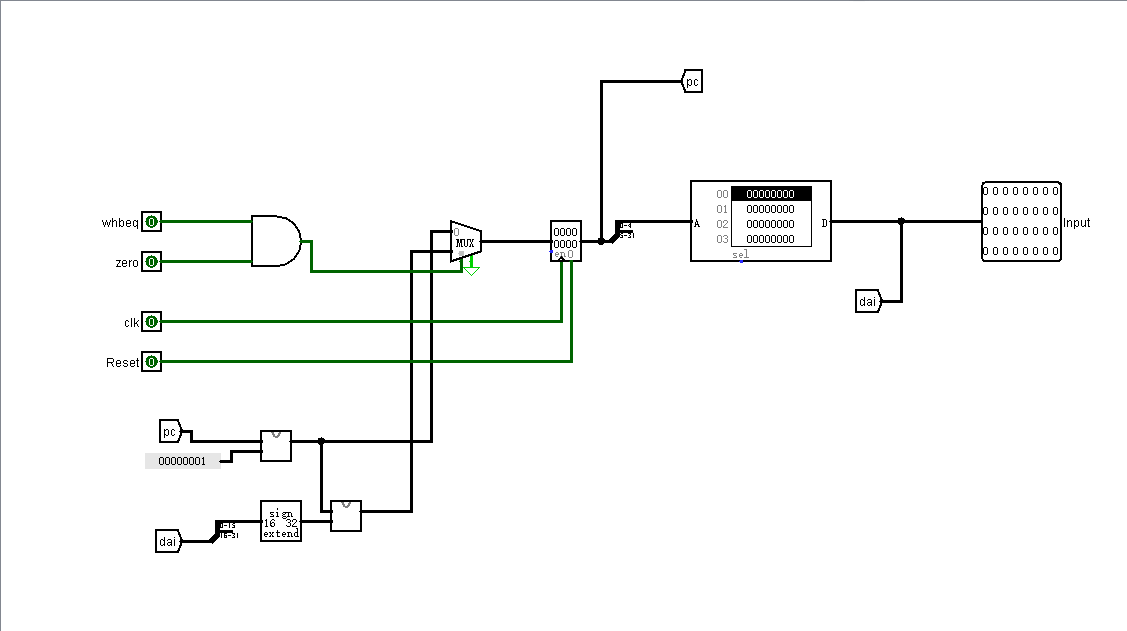
三、实验设计

模块定义：



主电路：

四、各模块介绍

1.IFU模块

（1）基本描述

IFU主要功能是完成取指令功能，IFU包括PC（程序计数器）和IM（指令存储器）以及其他逻辑部件，IFU除了可以顺序执行取指令外，还可以根据BEQ指令的执行情况决定顺序取指令还是转移取指令。

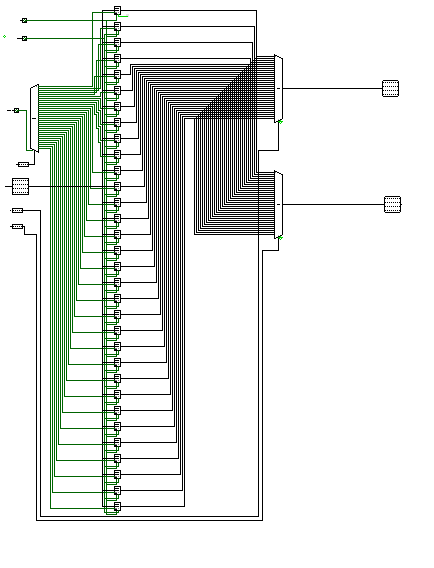
（2）模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| whbeq | I | 判断当前指令是否为beq指令。  1：当前指令是beq。  0：当前指令不是beq。 |
| zero | I | 判断当前ALU计算结果是否为0。  1：计算结果为0。  0：计算结果不是0。 |
| clk | I | 时钟信号。 |
| Reset | I | 复位信号。  1：复位信号有效。  0：复位信号无效。 |
| Input[31:0] | O | 当前的32位MIPS指令。 |

（3）功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 复位 | 复位信号有效时，PC被设置为0x00000000。 |
| 2 | 取指令 | 根据PC中的数字取出指令。 |
| 3 | 计算下一条指令地址 | 如果当前指令不是BEQ指令，或者当前指令是BEQ指令并且Zero为0，则PC←PC+1。 如果当前指令是BEQ指令并且Zero为1，则PC←PC+1+sign\_ext[当前指令15：0]。  [注]：PC取地址为4字节，低两位地址可以去除。 |

2.GRF模块



（1）基本描述

GPR主要功能是利用寄存器以实现对数据的取出和存入操作。通过一个32位MIPS指令对指令中的指定寄存器的值进行读或写操作，以实现对这些数据的后续操作

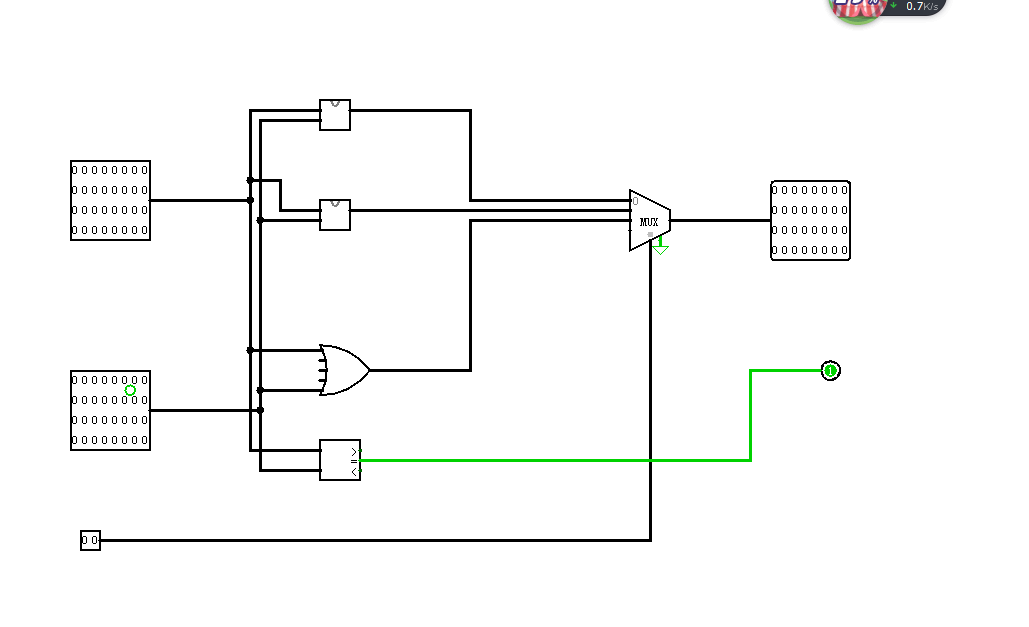
（2）模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| Regwrite | I | 判断当前是否向寄存器堆写入值。  1：向寄存器里存值。  0：不向寄存器里存值。 |
| wd[4:0] | I | 写寄存器地址。 |
| clk | I | 时钟信号。 |
| Reset | I | 复位信号。  1：复位信号有效。  0：复位信号无效。 |
| Input[31:0] | I | 写入数据的输入。 |
| Rd1 | I | 读入数据的寄存器地址1。 |
| Rd2 | I | 读入数据的寄存器地址2。 |
| Output1[31:0] | O | 地址中的数据输出1。 |
| Output2[31:0] | O | 地址中的数据输出2。 |

（3）功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 读数 | 取出指定寄存器中存储的数据 |
| 2 | 写数 | 将指定的32位数据写入指定的寄存器中。 |

3.ALU模块



（1）基本描述

ALU的功能是对指定的两个32位数进行无符号加、无符号减、或以及对两个数进行大小比较。

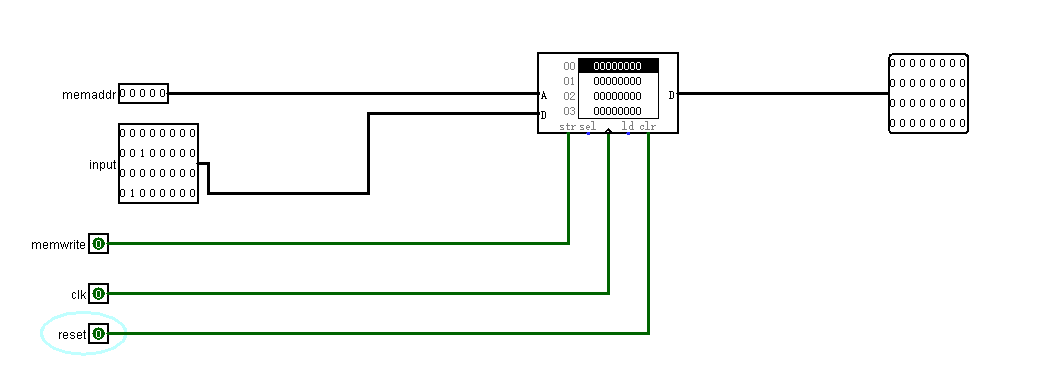
（2）模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| Input1[31:0] | I | 第一个32位数据的输入。 |
| Input2[31:0] | I | 第二个32位数据的输入。 |
| Aluop[1:0] | I | 输出的控制信号。  00：输出加法结果。  01：输出减法结果。  10：输出或运算结果。 |
| Zero | O | 判断两个数是否相等。 |
| output[31:0] | O | 写入数据的输入。 |

（3）功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 加法运算 | 对两个32位数据进行无符号加法运算。 |
| 2 | 减法运算 | 对两个32位数据进行无符号减法运算。 |
| 3 | 或运算 | 对两个32位数据进行或运算。 |
| 4 | 判断是否相等 | 判断两个32位数据是否相等 |

4.DM模块



（1）基本描述

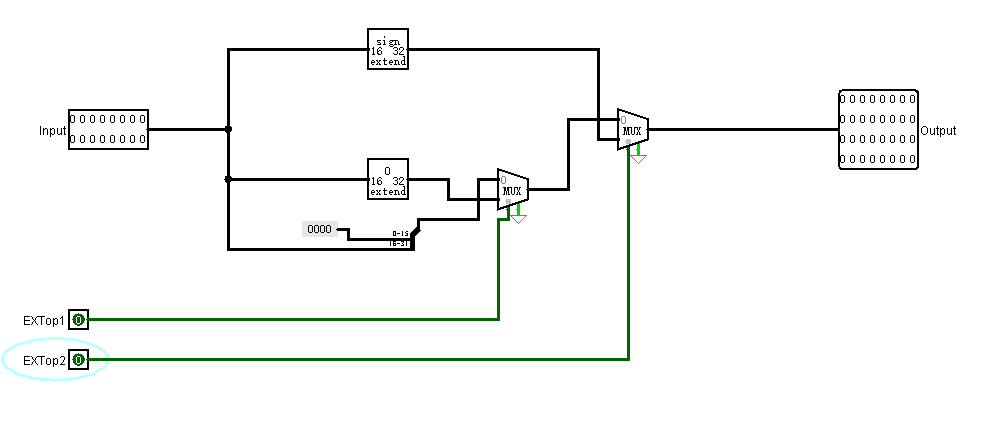
DM（数据存储器）的功能是对指定的存储器地址里的数据进行操作。并根据信号判断进行读操作还是写操作。

（2）模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| Input [31:0] | I | 执行写操作所需的32位数据。 |
| memaddr[4:0] | I | 存储32位数据的地址。 |
| clk | I | 时钟信号 |
| Reset | I | 复位信号。 |
| memwrite | I | 判断执行写操作还是读操作。  0：读操作  1：写操作 |
| Output[31:0] | O | 读入指定地址的32位数据 |

（3）功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名称 | 功能描述 |
| 1 | 读数 | Memwrite信号为0时，读出指定地址中的数据。 |
| 2 | 写数 | Memwrite信号为1时，将32位写入指定地址中。 |

5.EXT模块

（1）基本描述

EXT的功能是对一个十六位二进制数进行扩展，并根据两个扩展信号判断进行符号扩展还是零扩展。

（2）模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| Input [15:0] | I | 需要扩展的16位数据 |
| memaddr[4:0] | I | 存储32位数据的地址。 |
| EXTop1 | I | 判断哪里补零。  0：低位补零。  1：高位补零。 |
| EXTop1 | I | 选择需要输出的32位数据  0：已经补零的数据。  1：已经符号扩展的数据。 |
| Output[31:0] | O | 输出扩展完成的32位数据。 |