P0实验报告

1. 关于各模块
2. IFU

(1) 基本描述

IFU 主要功能是完成取指令功能。IFU 内部包括 PC、IM(指令存储器)以及其

他相关逻辑。IFU 除了能执行顺序取值令外，还能根据 BEQ 指令的执行情况决定顺序取值令还是转移取值令。



1. GPR
2. 基本描述

主要由32个寄存器组成，并由多个控制信号控制完成读数和存数的功能。

1. 模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| readRegister1[5:0] | I | 第一个被读寄存器的地址 |
| readRegister2[5:0] | I | 第二个读被寄存器的地址 |
| writeRegister[5:0] | I | 被写入寄存器的地址 |
| writeData[31:0] | I | 要写入的数据 |
| regWrite | I | 是否实现写入功能  1：可以写入  0：不能写入 |
| clk | I | 时钟信号 |
| readData1[31:0] | O | 第一个读出的数据 |
| readData2[31:0] | O | 第二个读出的数据 |

1. 功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名 | 功能描述 |
| 1 | 读出 | 根据输入的读数地址readRegister1和readRegister2输出寄存器里的值 |
| 2 | 读入 | 当写入使能信号为1时，把输入writeData的值写入writeRegister这个地址的寄存器 |

1. ALU
   1. 基本描述

主要进行各种运算，加减，或，以及lui功能。

* 1. 模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| A | I | 第一个运算数 |
| B | I | 第二个运算数 |
| ALUOp1[1:0] | I | 选择计算操作类型 |
| Result | O | 运算结果 |
| zero | O | 运算结果是否为0  1：运算结果为0  0：运算结果不为0 |

* 1. 功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名 | 功能描述 |
| 1 | 加 | 将两个数进行相加 |
| 2 | 减 | 用第一个操作数减去第二个并输出结果 |
| 3 | 与立即数ori | 把两个操作数（其中一个为立即数）进行按位或 |
| 4 | 加载到高位lui | 把一个立即数的低16位加载到32位的高位，并低位补0 |

1. EXT
   1. 基本描述

到操作数进行不同方式的扩展，主要是0扩展和符号扩展

* 1. 模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| imm16 | I | 需要扩展的数 |
| ExtOP | I | 扩展方式 |
| out | O | 32位输出 |

* 1. 功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名 | 功能描述 |
| 1 | 0扩展 | 当ExtOP为0时将imm16扩展成32位并进行高位补0 |
| 2 | 减 | 当ExtOP为1时将imm16扩展成32位并进行高位补上符号码 |

1. DM
   1. 基本描述

主要是存储功能。

* 1. 模块接口

|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| Address[4:0] | I | 写入数据将在DM中存放的地址或要读数据的地址 |
| Writedata[31:0] | I | 要写入的数据 |
| MemWrite | I | 写入使能信号 |
| MemRead | I | 读操作使能信号 |
| Clk | I | 时钟信号 |
| Readdata[31:0] | O | 读出的数据 |

* 1. 功能定义

|  |  |  |
| --- | --- | --- |
| 序号 | 功能名 | 功能描述 |
| 1 | 读入 | 当MemWrite为1时将Writedata写入adderss所指的位置 |
| 2 | 减 | 当MemRead为1, MemWrite为0时将adderss所指位置的数据读出 |

1. Controller
   1. 基本描述

产生各种控制信号。

* 1. 模块接口

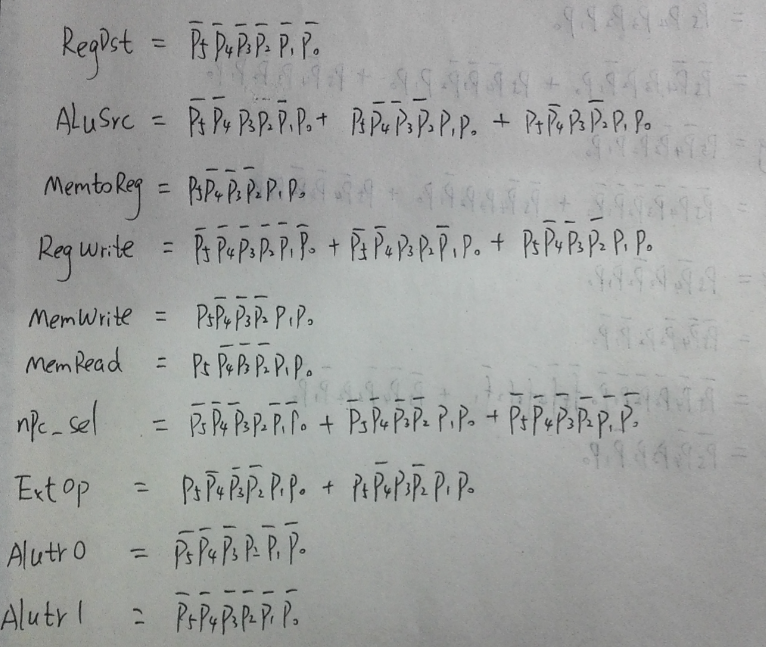
|  |  |  |
| --- | --- | --- |
| 信号名 | 方向 | 描述 |
| op | I | 指令的前6位Option码 |
| RegDst | O | Gpr的writeRegister选择信号 |
| ALUSrc | O | Alu第二个操作数的选择信号 |
| MemtoReg | O | GPR读入数据选择信号 |
| RegWrite | O | GPR读入使能信号 |
| MemRead | O | DM读数使能信号 |
| MemWrite | O | DM写入使能信号 |
| Branch | O | Beq指令标志信号 |
| ALUtrc | O | 传给ALUsontrol的信号 |
| ol | O |
| ExtOp | O | ExtOP功能选择信号 |

* 1. 功能定义

|  |  |  |
| --- | --- | --- |
| 1. 序号 | 功能名 | 功能描述 |
| 1 | 控制信号产生 | 根据op的值产生不各种控制信号，控制各种部件工作 |

2、控制器设计 addu，subu，ori，lw，sw，beq，lui, jal，jr

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Func  op | 100001 | 100011 |  |  |  |  |  |  |  |
| 000000 | 000000 | 001101 | 100011 | 101011 | 000100 | 001111 | 000011 | 000000 |
| addu | subu | ori | lw | sw | beq | lui | jal | jr |
| RegDst | 1 | 1 | 0 | 0 | 0 | 0 | 0 | x | x |
| ALUSrc | 0 | 0 | 1 | 1 | 1 | 0 | 1 | x | x |
| MemtoReg | 0 | 0 | 0 | 1 | 0 | 0 | 0 | x | x |
| RegWrite | 1 | 1 | 1 | 1 | 0 | 0 | 1 | x | x |
| MemWrite | 0 | 0 | 0 | 1 | 0 | 0 | 0 | x | x |
| MemRead | 0 | 0 | 0 | 1 | 0 | 0 | 0 | x | x |
| nPC\_sel | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| ExtOp | x | x | 0 | 1 | 1 | x | 0 | x | x |
| ALUCtr0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | x | x |
| ALUCtr1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | x | x |



五、 测设要求 lui $1,0 #初始化$1为0

ori $t0,$1,0 #初始化$t0为0，$t0用来做地址

ori $t4,$zero,0 #初始化$t4为0，$t4用来计数

ori $t5,$t4,10 #初始化$t5为10，$t5用来判断$t4是否为10

ori $t1,$t4,1 #初始化$t1为1,每次自增1用到

addu $t2,$t4,$t1 #用$t0,$t1相加得到$t2

loop: sw $t2,0($t0) #将$t2,存入DM

addu $t0,$t0,$t1 #$t0自增4

addu $t2,$t2,$t1 #$t2自增1

addu $t4,$t4,$t1 #$t4自增1，表示计数

beq $t4,$t5,out # $t4,$t5相等则退出

beq $t4,$t4,loop #继续

out: lw $t2,-10($t0) #取出前面存的数

lui $t3,3 #把3加载到高位并保存到$t3

subu $t2,$t3,$t2 #让$t3减去$t2

sw $t2,0($t0) #将$t2保存回原位

addu $t0,$t0,$t1 #$t0自增1

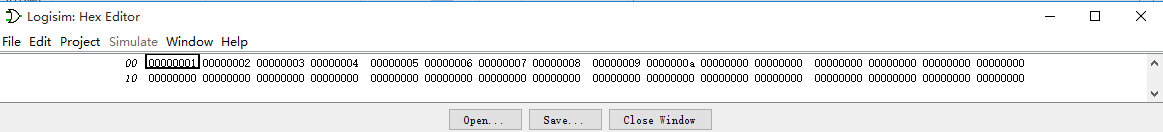
subu $t4,$t4,$t1 #$t4自减1

beq $t4,$0,exit #$t4等于0时退出

beq $t4,$t4,out #不等于0则继续

exit:

测试程序最后会在DM里出现１，２，。。。。，１０。之后还会在ALU的输出端多次出现前16位的最后两位有两个1，接着会有一些明显大于10的数出现在alu输出Result端。



这是将3加载到高位的现象。DM的之后10个数会是把3加载到高位，再分别减去前面保存的1到10的结果。



六、 问答

18、

a)

|  |  |  |  |
| --- | --- | --- | --- |
| 信号 | 与门数 | 或门数 | 非门数 |
| Ori | 6 | 0 | 3 |
| add | 6 | 0 | 6 |
| sub | 6 | 0 | 6 |

|  |  |  |  |
| --- | --- | --- | --- |
| 信号 | 与门数 | 或门数 | 非门数 |
| RegDst | 0 | 1 | 0 |
| ALUScr | 0 | 3 | 0 |
| RegWrite | 0 | 2 | 0 |
| ExtOP | 0 | 1 | 0 |

b)

我更喜欢17那种设计方法，先分析再画图，这样画出的图才简单不太复杂。