设计者学号：

设计者姓名：

1. **ALU设计（需要设计者完成）**

为了设计ALU的控制信号，首先根据指令功能进行归纳总结，完成下表。

|  |  |  |
| --- | --- | --- |
| 指令 | 操 作 | ALU功能 |
| Addu | 将寄存器rs的值与寄存器rt的值相加，结果写入rd寄存器中。 | 加 |
| Addiu | 将寄存器rs的值与**有符号扩展**至32位的立即数imm相加，结果写入rt寄存器中。 | 加 |
| Mul | 有符号乘法，寄存器rs的值乘以寄存器rt的值，乘积的低32位写入rd寄存器。 | 乘 |
| Sub | 将寄存器rs的值与寄存器rt的值相减，结果写入rd寄存器中。如果产生溢出，则触发整型溢出例  外（IntegerOverflow） | 减 |
| And | 寄存器rs中的值与寄存器rt中的值按位逻辑与，结果写入寄存器rd中 | 与 |
| Andl | 寄存器rs中的值与0扩展至32位的立即数imm按位逻辑与，结果写入寄存器rt中 | 与 |
| Lui | 将16位立即数imm写入寄存器rt的高16位，寄存器rt的低16位置0 | 左移16位 |
| Or | 寄存器rs中的值与寄存器rt中的值按位逻辑或，结果写入寄存器rd中。 | 或 |
| Ori | 寄存器rs中的值与0扩展至32位的立即数imm按位逻辑或，结果写入寄存器rt中 | 或 |
| Xor | 寄存器rs中的值与寄存器rt中的值按位逻辑异或，结果写入寄存器rd中 | 异或 |
| Xori | 寄存器rs中的值与0扩展至32位的立即数imm按位逻辑异或，结果写入寄存器rt中 | 异或 |
| Sll | 由立即数sa指定移位量，对寄存器rt的值进行逻辑左移，结果写入寄存器rd中。 | 左移 |
| Srl | 由立即数sa指定移位量，对寄存器rt的值进行逻辑右移，结果写入寄存器rd中 | 右移 |
| Beq | 如果寄存器rs的值**等于**寄存器rt的值则转移，否则顺序执行。转移目标由立即数offset左移2位  并进行**有符号扩展**的值加上该分支指令对应的延迟槽指令的PC计算得到。 | 减 |
| Bne | 如果寄存器rs的值**不等于**寄存器rt的值则转移，否则顺序执行。转移目标由立即数offset左移2  位并进行**有符号扩展**的值加上该分支指令对应的延迟槽指令的PC计算得到。 | 减 |
| Bgtz | 如果寄存器rs的值**大于0**则转移，否则顺序执行。转移目标由立即数offset左移2位并进行**有**  **符号扩展**的值加上该分支指令对应的延迟槽指令的PC计算得到。 | 减 |
| J | 无条件跳转。跳转目标由该分支指令对应的延迟槽指令的PC的最高4位与立即数instr\_index左移  2位后的值拼接得到。 | 无 |
| Jal | 无条件跳转。跳转目标由该分支指令对应的延迟槽指令的PC的最高4位与立即数instr\_index左移  2 位后的值拼接得到。同时将该分支对应延迟槽指令之后的指令的PC值保存至第31号通用寄存  器中。 | 无 |
| Jr | 无条件跳转。跳转目标为寄存器rs中的值。 | 无 |
| Lb | 将base寄存器的值加上**符号扩展**后的立即数offset得到访存的虚地址，据此虚地址从存储器中读  取1个字节的值并进行符号扩展，写入到rt寄存器中 | 加 |
| Lw | 将base寄存器的值加上**符号扩展**后的立即数offset得到访存的虚地址，如果地址不是4的整数倍  则触发地址错例外，否则据此虚地址从存储器中读取连续4个字节的值，写入到rt寄存器中。 | 加 |
| Sb | 将base寄存器的值加上**符号扩展**后的立即数offset得到访存的虚地址，据此虚地址将rt寄存器的  **最低字节**存入存储器中。 | 加 |
| Sw | 将base寄存器的值加上**符号扩展**后的立即数offset得到访存的虚地址，如果地址不是4的整数倍  则触发地址错例外，否则据此虚地址将rt寄存器存入存储器中。 | 加 |

# ALU的控制信号设计

从上面表来看，一共 8种操作，使用 独热码，在下表中给出定义。

|  |  |
| --- | --- |
| 操作 | 编码 |
| 加 | 0000\_0001 |
| 减 | 0000\_0010 |
| 乘 | 0000\_0100 |
| 与 | 0000\_1000 |
| 或 | 0001\_0000 |
| 异或 | 0010\_0000 |
| 左移 | 0100\_0000 |
| 右移 | 1000\_0000 |

1. **控制单元分成两个部分，一个是cu，一个是alucontrol。**
2. 控制信号定义（仿照教材，所以直接给出）

|  |  |  |
| --- | --- | --- |
| 信号名 | “0”时定义 | “1”时定义 |
| RegWrite | 无 | 往寄存器里写数据 |
| ALUSrc | 第二个操作数来自寄存器 | 第二个操作数是立即数扩展 |
| MemRead | 无 | 读存储器 |
| MemWrite | 无 | 写存储器 |
| SaFrom | Sa字段作为sa输入 | 5’d16作为sa输入 |
| MemByte | 读取或者写入四个字节 | 读取或者写入单个字节 |
| Extension | 与0扩展立即数 | 符号扩展立即数 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 信号名 | 00 | 01 | 10 | 11 |
| MemtoReg | ALU输出作为结果寄存器输入 | 存储器输出作为结果寄存器输入 | Pc+8作为寄存器输入 | 无 |
| RegDst | 写入寄存器来自rt | 写入寄存器来自rd | 写入31号寄存器 | 无 |
| ALUSrc | 第二个操作数来自寄存器 | 第二个操作数是立即数扩展 | 第二个操作数为0 | 无 |
| Extension | 与0扩展立即数 | 符号扩展立即数 | 立即数offset左移2位  并进行有符号扩展 |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | 000 | 001 | 010 | 011 | 100 | 101 |  |
| PCSrc | PC <-- PC+4 | PC <-- 相等条件跳转分支地址 | PC <-- 不相等条件跳转分支地址 | PC <--大于条件跳转分支地址 | PC <-- 无条件跳转分支地址（jal,j） | PC <-- 无条件跳转间接分支地址（jr） |  |

|  |  |  |
| --- | --- | --- |
| 多选器控制 | 源 | 解释 |
| ForwardA=00 | ID/EX | 第一个ALU操作数来自寄存器堆 |
| ForwardA=10/11 | EX/MEM | 第一个ALU操作数由上一个ALU运算结果转发获得 |
| ForwardA=01 | MEM/WB | 第一个ALU操作数从数据存储器或者前面的ALU结果中转发获得 |
| ForwardB=00 | ID/EX | 第二个ALU操作数来自寄存器堆 |
| ForwardB=10/11 | EX/MEM | 第二个ALU操作数由上一个ALU运算结果转发获得 |
| ForwardB=01 | MEM/WB | 第二个ALU操作数从数据存储器或者前面的ALU结果中转发获得 |

1. 针对指令的cu输入/输出真值表

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 指令 | 操作码 | 类型 | RegDst | RegWrite | ALUSrc | PCSrc | MemRead | MemWrite | MemtoReg | SaFrom | MemByte | Extension | ALUop |
| Addu | 000000 | R | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1000 |
| Addiu | 001001 | I | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 001 |
| Mul | 011100 | R | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 011 |
| Sub | 000000 | R | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1000 |
| And | 000000 | R | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1000 |
| Andi | 001100 | I | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 100 |
| Lui | 001111 | I | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 110 |
| Or | 000000 | R | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1000 |
| Ori | 001101 | I | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 101 |
| Xor | 000000 | R | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1000 |
| Xori | 001110 | I | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0111 |
| Sll | 000000 | R | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1000 |
| Srl | 000000 | R | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1000 |
| Beq | 000100 | I | 0 | 0 | 0 | 001 | 0 | 0 | 0 | 0 | 0 | 10 | 010 |
| Bne | 000101 | I | 0 | 0 | 0 | 010 | 0 | 0 | 0 | 0 | 0 | 10 | 010 |
| Bgtz | 000111 | I | 0 | 0 | 10 | 011 | 0 | 0 | 0 | 0 | 0 | 10 | 010 |
| J | 000010 | J | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 11 | 0 |
| Jal | 000011 | J | 10 | 1 | 0 | 100 | 0 | 0 | 10 | 0 | 0 | 11 | 0 |
| Jr | 000000 | R | 0 | 0 | 0 | 101 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Lb | 100000 | I | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| Lw | 100011 | I | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| Sb | 101000 | I | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| Sw | 101011 | I | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |

1. 针对指令的aluctrl输入/输出真值表

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 指令 | 操作码 | 类型 | ALUop | Function字段 | ALU操作 | ALU控制码 |
| Addu | 000000 | R | 1000 | 100001 | 加 | 0000\_0001 |
| Addiu | 001001 | I | 1 |  | 加 | 0000\_0001 |
| Mul | 011100 | R | 11 |  | 乘 | 0000\_0100 |
| Sub | 000000 |  | 1000 | 100010 | 减 | 0000\_0010 |
| And | 000000 | R | 1000 | 100100 | 与 | 0000\_1000 |
| Andl | 000101 | I | 100 |  | 与 | 0000\_1000 |
| Lui | 100011 | I | 110 |  | 左移16位 | 0100\_0000 |
| Or | 101011 | R | 1000 | 100101 | 或 | 0001\_0000 |
| Ori | 001101 | I | 101 |  | 或 | 0001\_0000 |
| Xor | 000000 | R | 1000 | 100110 | 异或 | 0010\_0000 |
| Xori | 001110 | I | 111 |  | 异或 | 0010\_0000 |
| Sll | 000000 | R | 1000 | 000000 | 左移 | 0100\_0000 |
| Srl | 000000 | R | 1000 | 000010 | 右移 | 1000\_0000 |
| Beq | 000100 | I | 10 |  | 减 | 0000\_0010 |
| Bne | 000101 | I | 10 |  | 减 | 0000\_0010 |
| Bgtz | 000111 | I | 10 |  | 减 | 0000\_0010 |
| J | 000010 | J | 0 |  | 无 | 0 |
| Jal | 000011 | J | 0 |  | 无 | 0 |
| Jr | 000000 | R | 0 | 001000 | 无 | 0 |
| Lb | 100000 | I | 1 |  | 加 | 0000\_0001 |
| Lw | 100011 | I | 1 |  | 加 | 0000\_0001 |
| Sb | 101000 | I | 1 |  | 加 | 0000\_0001 |
| Sw | 101011 | I | 1 |  | 加 | 0000\_0001 |