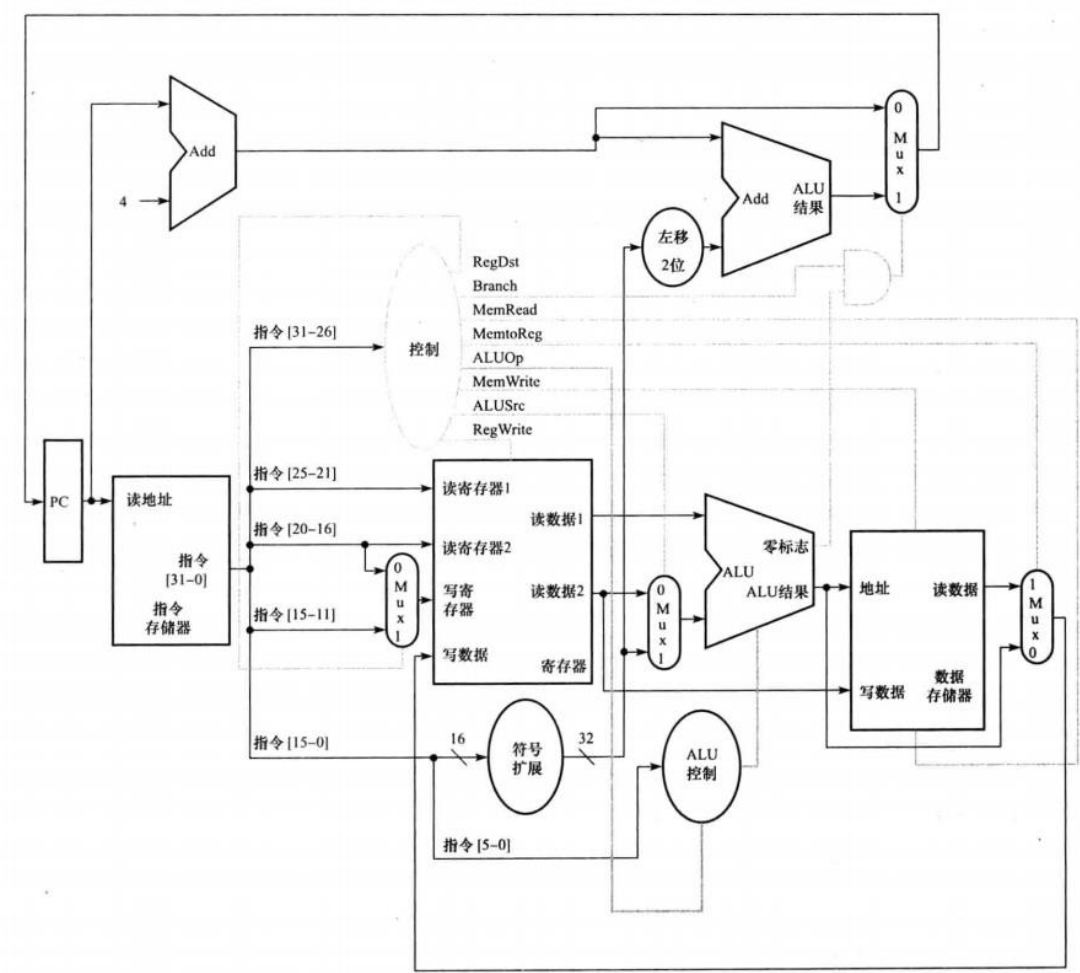


设计者学号： 21231511
设计者姓名：欧祖宇

1. 指定的指令如下

指令	操作码	类型	含义
ori	001101	I	汇编格式: ORI rt, rs, imm 功能描述: 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。 操作定义: $GPR[rt] \leftarrow GPR[rs] \text{ or } Zero_extend(imm)$
lui	001111	I	汇编格式: LUI rt, imm 功能描述: 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。 操作定义: $GPR[rt] \leftarrow (imm \parallel 0^{16})$
addu	000000 Func: 100001	R	汇编格式: ADDU rd, rs, rt 功能描述: 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。 操作定义: $GPR[rd] \leftarrow GPR[rs] + GPR[rt]$
sub	000000 Func: 100010	R	汇编格式: SUB rd, rs, rt 功能描述: 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)。 假定不产生溢出
bne	000101	I	汇编格式: BNE rs, offset 功能描述: 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。 操作定义: $if \text{ condition} \leftarrow GPR[rs] \neq GPR[rt]$
lw	100011	I	汇编格式: LW rt, offset(base) 功能描述: 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。
sw	101011	I	汇编格式: SW rt, offset(base) 功能描述: 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。

2. 带控制信号的数据通路（用书上的即可）



3. 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
ori	op1 和 op2 按位或	或运算
lui	op2 左移 16 位	左移运算
addu	op1 和 op2 相加	加运算
Sub	op1 和 op2 相减	减运算
bne	op1 和 op2 判断相等	减运算后为 0 zero 输出 1
lw	op1 和 op2 相加	加运算
sw	op1 和 op2 相加	加运算

4. ALU 的控制信号设计

从上面表来看，一共 4 种操作，使用 二进制编码，在下表中给出定义。

操作	编码
加运算	00
减运算	01
或运算	10
左移运算	11

5. 控制单元分成两个部分，一个是 **cu**，一个是 **alucontrol**。

(1) 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
RegDst	写入寄存器来自 rt	写入寄存器来自 rd
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
PCSrc	$PC \leftarrow PC+4$	$PC \leftarrow$ 分支地址
MemRead	无	读存储器
MemWrite	无	写存储器
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

(2) 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	ALUop
addu	000000	R	1	1	0	0	0	0	1	000
sub	000000	R	1	1	0	0	0	0	1	000
ori	001101	I	0	1	1	0	0	0	1	110
lui	001111	I	0	1	1	0	0	0	1	111
bne	000101	I	0	0	0	1	0	0	0	010
lw	100011	I	0	1	1	0	1	0	0	001
sw	101011	I	0	0	1	0	0	1	0	101

(3) 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	000	100001	加运算	00
sub	000000	R	000	100010	减运算	01
ori	001101	I	110	*****	或运算	10
lui	001111	I	111	*****	左移运算	11
bne	000101	I	010	*****	减运算	01
lw	100011	I	001	*****	加运算	00
sw	101011	I	101	*****	加运算	00