

2. 单周期处理器在一个周期内完成指令所有的微操作，思考：

- 寻址方式如何实现
不局限于RISC-V架构，直接寻址，间接寻址，寄存器寻址，寄存器间接寻址，立即寻址，隐含寻址，基址寻址，变址寻址，相对寻址，堆栈寻址等。
- 周期宽度如何确定
所有指令中延迟最大的指令执行所需时间。
- 能否“在一个clk内完成”
若一个clk大于所有指令的最大延迟，则可以。
- 能否将两个adder合二为一
可以，使用Mux选择下一条指令的偏移
- 能否将两个memory合二为一
可以，使用两读一写端口的Memory实现

3. 根据图4-22，画出主控制器的PLA实现图

- 先写布尔表达式
- 请参考英文版教材附录C第2节
- 由于PLA规模不小，可以手绘，也可以用软件画完后打印出来

PLA 为主控制器画PLA实现图

主控制器真值表

Input or output	Signal name	R-format	ld	sd	beq	Other
Inputs	I[6]	0	0	0	1	Not R/ ld/ sd/ beq
	I[5]	1	0	1	1	
	I[4]	1	0	0	0	
	I[3]	0	0	0	0	
	I[2]	0	0	0	0	
	I[1]	1	1	1	1	
	I[0]	1	1	1	1	
Outputs	ALUSrc	0	1	1	0	x
	MemtoReg	0	1	X	X	x
	RegWrite	1	1	0	0	0
	MemRead	0	1	0	0	x
	MemWrite	0	0	1	0	0
	Branch	0	0	0	1	0
	ALUOp1	1	0	0	0	x
	ALUOp0	0	0	0	1	x

Input or output	Signal name	R-format	ld	sd	beq
Inputs	I[6]	0	0	0	1
	I[5]	1	0	1	1
	I[4]	1	0	0	0
	I[3]	0	0	0	0
	I[2]	0	0	0	0
	I[1]	1	1	1	1
	I[0]	1	1	1	1
Outputs	ALUSrc	0	1	1	0
	MemtoReg	0	1	X	X
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1

写布尔表达式

AND平面

把n个inputs合成一个布尔信号，再取个名字。
Inputs之间只需要做与操作，称为 AND 平面

$$R\text{-format} = \overline{I[6]} I[5] I[4] \overline{I[3]} \overline{I[2]} I[1] I[0], \text{ Only } I[6] \sim I[0] = 0110011, \text{ R-format} = 1$$

$$LD = \overline{I[6]} \overline{I[5]} \overline{I[4]} \overline{I[3]} \overline{I[2]} I[1] I[0], \text{ Only } I[6] \sim I[0] = 0000011, \text{ ld} = 1$$

$$SD = \overline{I[6]} I[5] \overline{I[4]} \overline{I[3]} \overline{I[2]} I[1] I[0], \text{ Only } I[6] \sim I[0] = 0100011, \text{ sd} = 1$$

$$BEQ = I[6] I[5] \overline{I[4]} \overline{I[3]} \overline{I[2]} I[1] I[0], \text{ Only } I[6] \sim I[0] = 1100011, \text{ beq} = 1$$

Input or output	Signal name	R-format	ld	sd	beq
Inputs	I[6]	0	0	0	1
	I[5]	1	0	1	1
	I[4]	1	0	0	0
	I[3]	0	0	0	0
	I[2]	0	0	0	0
	I[1]	1	1	1	1
	I[0]	1	1	1	1
Outputs	ALUSrc	0	1	1	0
	MemtoReg	0	1	X	X
	RegWrite	1	1	0	0
	MemRead	0	1	0	0
	MemWrite	0	0	1	0
	Branch	0	0	0	1
	ALUOp1	1	0	0	0
	ALUOp0	0	0	0	1

写布尔表达式

OR平面

对与平面得到的布尔型结果，根据outputs的真值表组合运算得到输出。
得到输出只需要做或操作，称为 OR 平面

$$ALUSrc = ld + sd \quad MemtoReg = ld (+sd(+beq))$$

$$RegWrite = R + ld \quad MemRead = ld$$

$$MemWrite = sd \quad Branch = beq$$

$$ALUOp1 = R \quad ALUOp0 = beq$$

![[image-20220612170117433](C:\Users\User\Desktop\3\image-20220612170117433.png)]

