

取指周期

存储器读信号memread有效，取出指令。
实际上设计中并未实现memread这一信号，存储器的读永远有效，只要地址发生变化，存储器就会在两个clk上升沿后将对应数据读出。取指周期存在的意义在于延长存储器读时间，确保能读出地址对应数据。

译码周期

分析指令op字段。
alusrca<=2'b00, alusrcb<=3'b011控制多路选择器选择寄存器A、B的值，计算跳转地址。（计算跳转地址并不一定会在后面用到，只是在译码阶段充分利用cpu资源的一个额外工作）

addi、andi、ori计算周期

alusrca<=2'b01, alusrcb为对应立即数所在数据通路。op为对应指令操作码。计算得到aluout<=A op B。

ltype写周期

regwrite信号有效，reglir[15:11]<=aluout，将计算结果aluout写入对应寄存器。

lw、sw访问地址计算

alusrca<=2'b01, alusrcb<=3'b010。
计算访存地址，下一个clk上升沿到来结果放到aluout。

lw读1、2、3

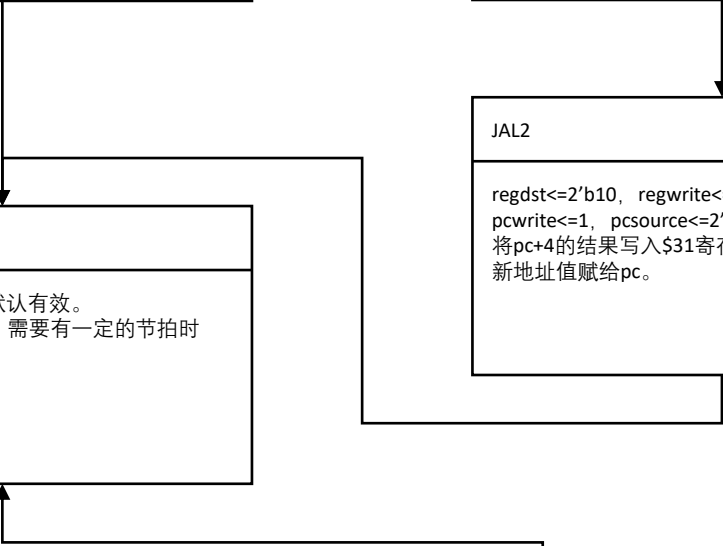
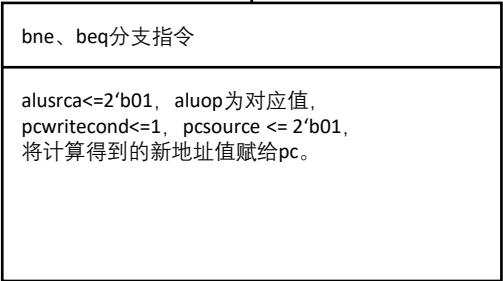
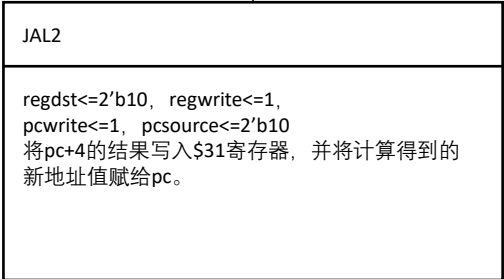
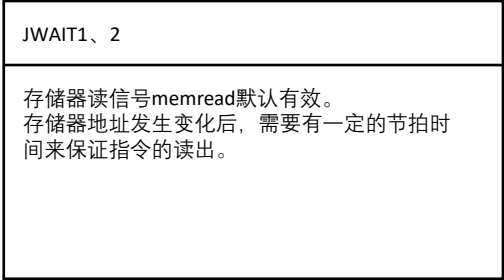
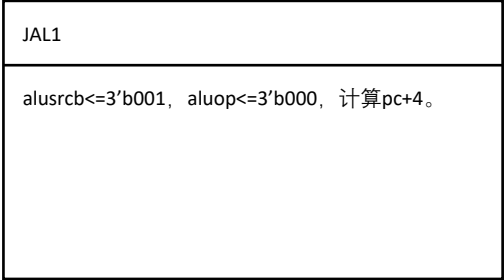
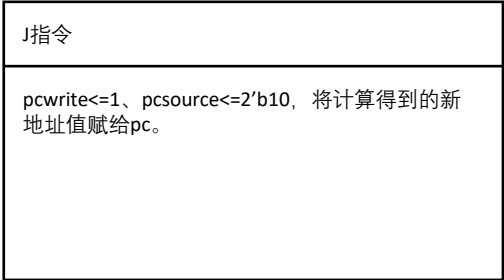
存储器读信号memread默认有效，iord<=1。
lw读1、2节拍内持续令alusrca<=2'b01, alusrcb<=3'b010，保证地址值不变，使得能在两个clk上升沿后将对应存储器数据读出。

sw写入1、2、3

存储器写信号memwrite<=1, iord<=1。
sw写1、2节拍内持续令alusrca<=2'b01, alusrcb<=3'b010，保证地址值不变，使得能在两个clk上升沿后将数据写入对应存储器地址空间。

lw写入寄存器

regwrite<=1, memtoreg<=1，将存储器读出数据写入对应寄存器。



add、addu、slt指令指令执行周期
alusrca<=2'b01, alusrcb<=3'b000, aluop<=3'b111, alu对A、B寄存器内容进行运算，运算结果在下一个clk上升沿到来后写入aluout。

sll、jr指令指令执行周期
sll: alusrca<=2'b10, aluop<=3'b111; jr: alusrca<=2'b01, alusrcb<=3'b110, aluop<=3'b111, pcwrite<=1, 将计算结果alurestult直接赋给pc。

RTYPE指令写
regwrite<=2'b01, regwrite<=1, 将R型指令计算结果写入对应结果寄存器，其中jr作用为跳转，故其在这一节拍周期向\$zero写入数据。