

RISC-V作业

5-31

时间问题导致周二没有讲评
同学有问题可以及时私聊

T1 & T2

回答问题：参考上面的图表，分别写出slli指令和AUIPC指令的控制信号

答案：

	BrEq	BrLT	PCSel	ImmSel	BrUn	ASel	Bsel	ALUSel	MemRW	RegWEn	WBSel
slli	*	*	0(PC+4)	I	*	0	1	sll	read	1	1(ALU)
auipc	*	*	0(PC+4)	U	*	1	1	add	read	1	1(ALU)

本题注意点：

ImmSel

tells immediate generator what type of immediate to generate

slli指令：I-type

auipc指令：U-type 很多同学在auipc指令上使用 ImmSel = I的控制信号

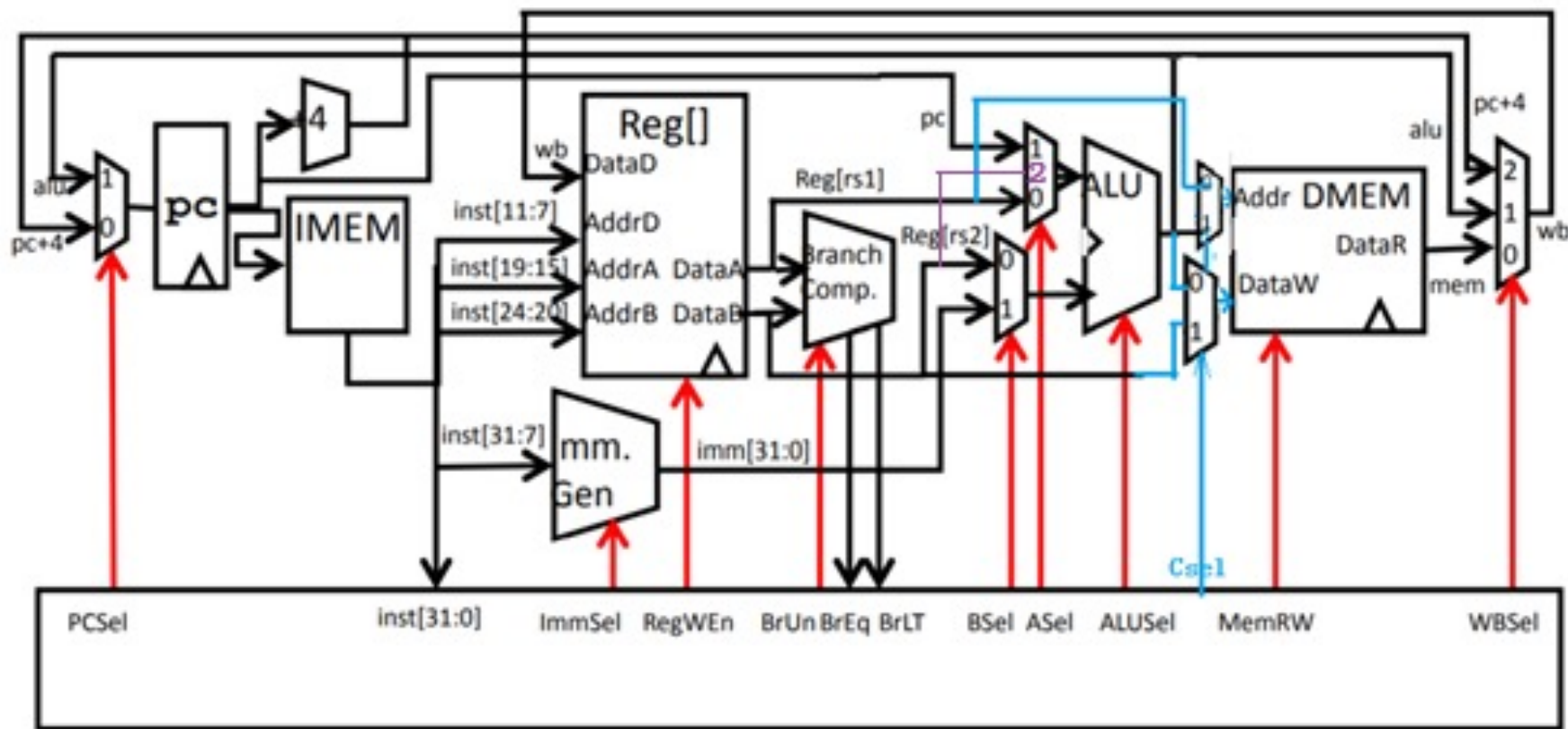
ALUSel 不仅仅只有add和sub

ALUSel has 4 bits: several different operations to choose

在slli指令中很多同学使用add作为ALUSel的控制信号

T1 & T2

尝试添加RISC-V指令：ss rs1, rs2, imm，功能为： $\text{Mem}[\text{Reg}[\text{rs1}]] = \text{Reg}[\text{rs2}] + \text{immediate}$ （存储两数之和）



本题完成的都比较好，要注意对部件进行改造，新加入功能部件时避免对原有的指令造成污染

T3 & T4

- 用到关键路径（critical path）的指令是哪一条？

注意到lw指令涉及了指令执行的全部五个阶段

	IF	ID	EX	MEM	WB
add	X	X	X		X
ori	X	X	X		X
lw	X	X	X	X	X
sw	X	X	X	X	
beq	X	X	X		
jal	X	X	X		X
bltu	X	X	X		

- 最小时钟周期 t_{clk} 是多少？最大时钟频率 f_{clk} 是什么？假设 $t_{clk-to-q} >$ 保持时间（hold time）。

提示： $t_{clk} = \text{PC寄存器的clktoQ} + \text{关键路径（critical path）延迟} + \text{RegFile_Setup}$

$$t_{clk} = t_{pc \text{ clk-to-q}} + t_{critical_path} + t_{regfile_setup}$$

$$= 30 + \quad + 20$$

IF : 250 (IMEM)

ID®: 150 (tRFread)

exe: 25(tmux) + 200(tALU)

dmem. 250(tMEMread)

wb 25(tmux)

T3 & T4

- 这个五阶段流水线处理器的最小时钟周期长度和最大时钟频率分别是多少？
 - 在T3的基础上，五个阶段前后都加上流水段寄存器的时间即可
 - IF : $t_{PC \text{ clk-to-q}} + t_{MEM \text{ read}} + t_{Reg \text{ setup}} = 30 + 250 + 20 = 300 \text{ ps}$
 - ID : $t_{Reg \text{ clk-to-q}} + t_{RF \text{ read}} + t_{Reg \text{ setup}} = 30 + 150 + 20 = 200 \text{ ps}$
 - EX : $t_{Reg \text{ clk-to-q}} + t_{mux} + t_{ALU} + t_{Reg \text{ setup}} + t_{mux} = 30 + 25 + 200 + 20 + 25 = 300 \text{ ps}$
 - MEM : $t_{Reg \text{ clk-to-q}} + t_{DMEM \text{ read}} + t_{Reg \text{ setup}} = 30 + 250 + 20 = 300 \text{ ps}$
 - WB : $t_{Reg \text{ clk-to-q}} + t_{mux} + t_{RF \text{ setup}} = 30 + 25 + 20 = 75 \text{ ps}$
 - 注意最小时钟周期取每个阶段的最大时间
- 相比于单周期处理器，性能加速比（speed up）是多少？为什么加速比会小于5？
 - 需要把时钟周期设置为五段流水线的最大时间段
 - 每一段的流水线都需要添加流水段寄存器clk-to-q 和寄存器set-up时间
 - 如果存在额外的数据冒险，还需要插入nop指令，所以实际的加速比还小于3.2

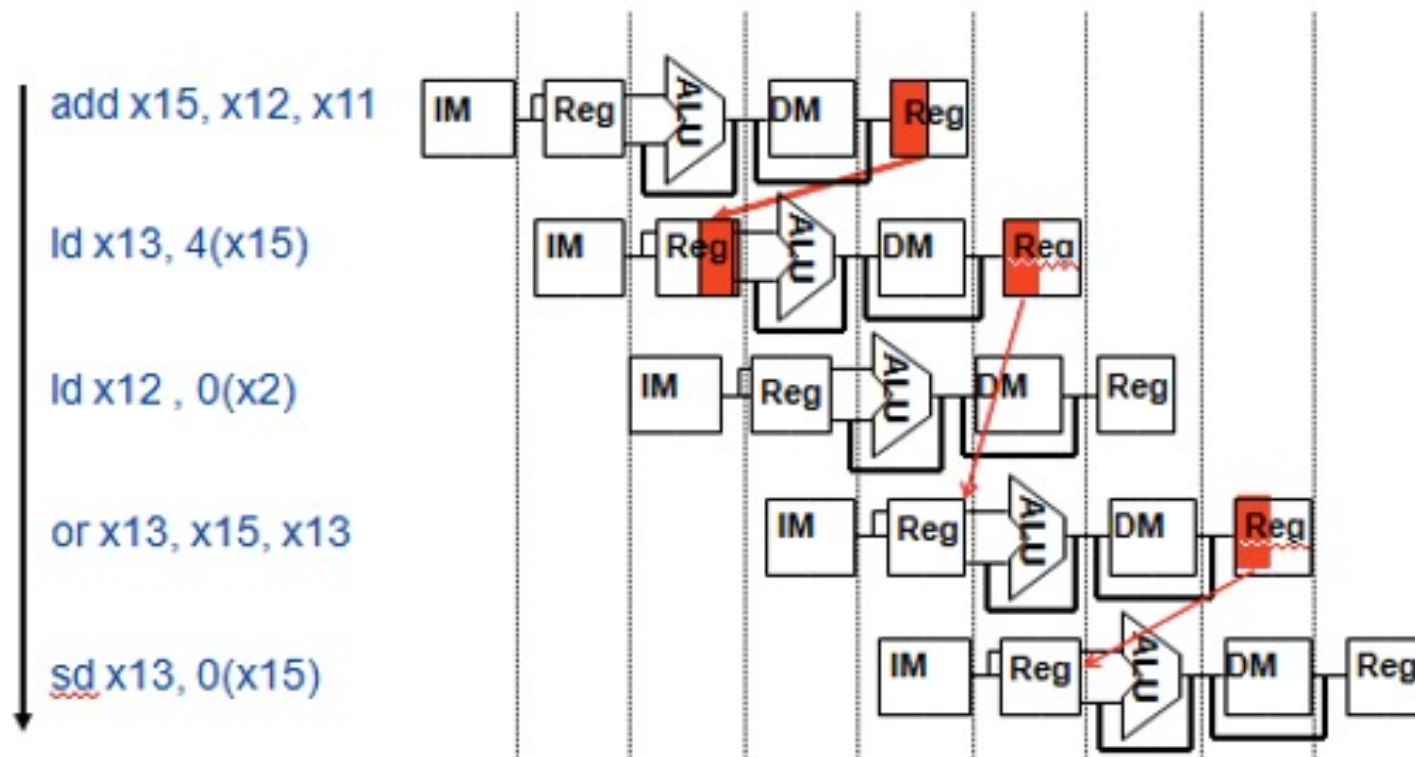
T5

- 假设硬件不检查和处理数据冒险，数据通路没有前向传递（Forwarding）在指令序列中插入空指令 NOP，使得上述指令序列得到正确的执行结果。

将指令流水线按照时间轴画出来。

标出相应的数据依赖关系即可

- add x15, x12, x11
- **NOP**
- **NOP**
- ld x13, 4(x15)
- ld x12, 0(x2)
- **NOP**
- or x13, x15, x13
- **NOP**
- **NOP**
- sd x13, 0(x15)



T5

- 假设对硬件不做任何改变，进行编译优化：对代码的次序重排、寄存器换名，使得插入的空指令最少。
假设寄存器x17可以被用来做临时寄存器。
 - 均无法优化

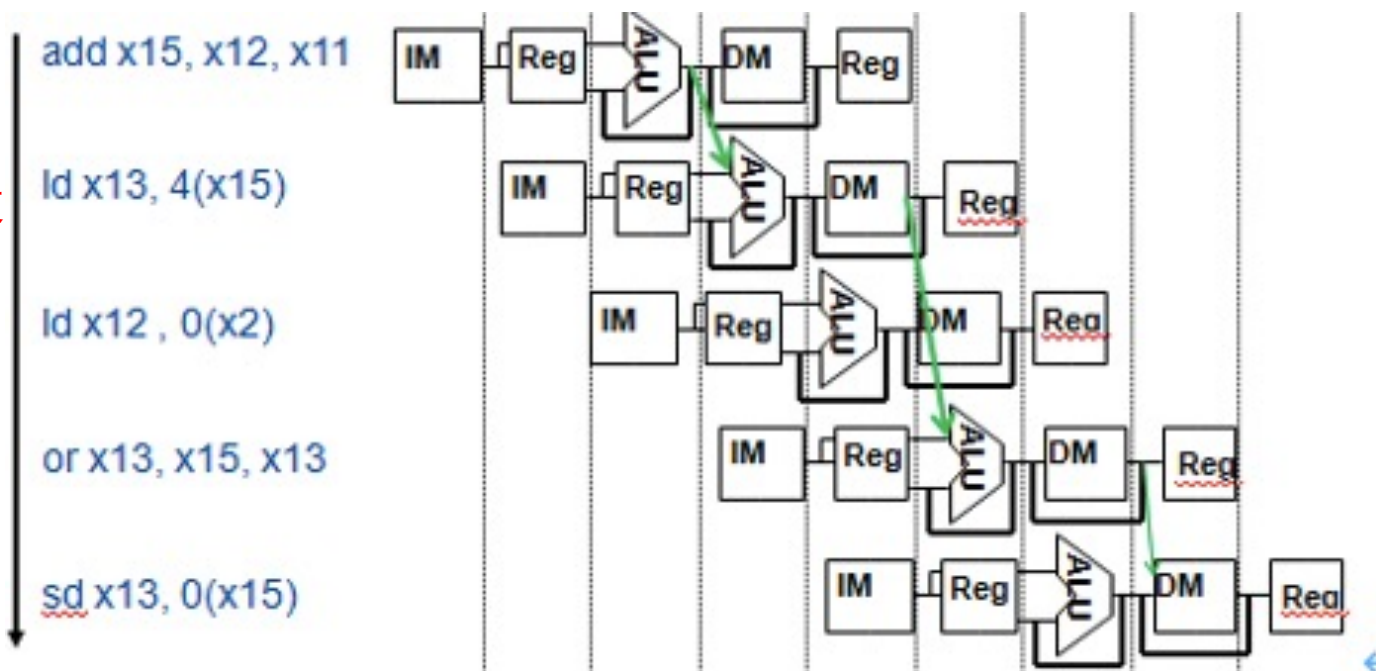
- 假设进行硬件优化：数据通路中增加了前向传递（forwarding），并增加了冒险检测单元。哪些指令之间还是需要停顿？停顿几个周期？

直接在时序图上找数据依赖

本题不需要停顿

注意这里很多同学忽略了

在ALU和DMEM输入端均需要数据转发
才能实现消除指令停顿



T6

- 在一个采用“取指、译码/取数、执行、访存、写回”的五段流水线中，检测结果是否为“零”的操作在执行阶段进行，则因为分支延迟导致阻塞的时钟周期为多少？以下指令序列中，在有数据转发（forwarding）的情况下，哪些指令执行时会发生流水线阻塞？各需要阻塞几个时钟周期？

```
1 loop: add t1, s3, s3
2           add t1, t1, t1
3           add t1, t1, s6
4           lw      t0, 0(t1)
5           bne t0, s5, exit
6           add s3, s3, s4
7           j      loop //这是一条伪指令，是jal指令的变体
8 exit:
```

指令2, 3, 4均可实现数据转发避免停顿

指令5存在典型的load-use需要阻塞一个时钟周期：这里有部分同学阻塞的时钟周期数量不对，需要回头看load-use