

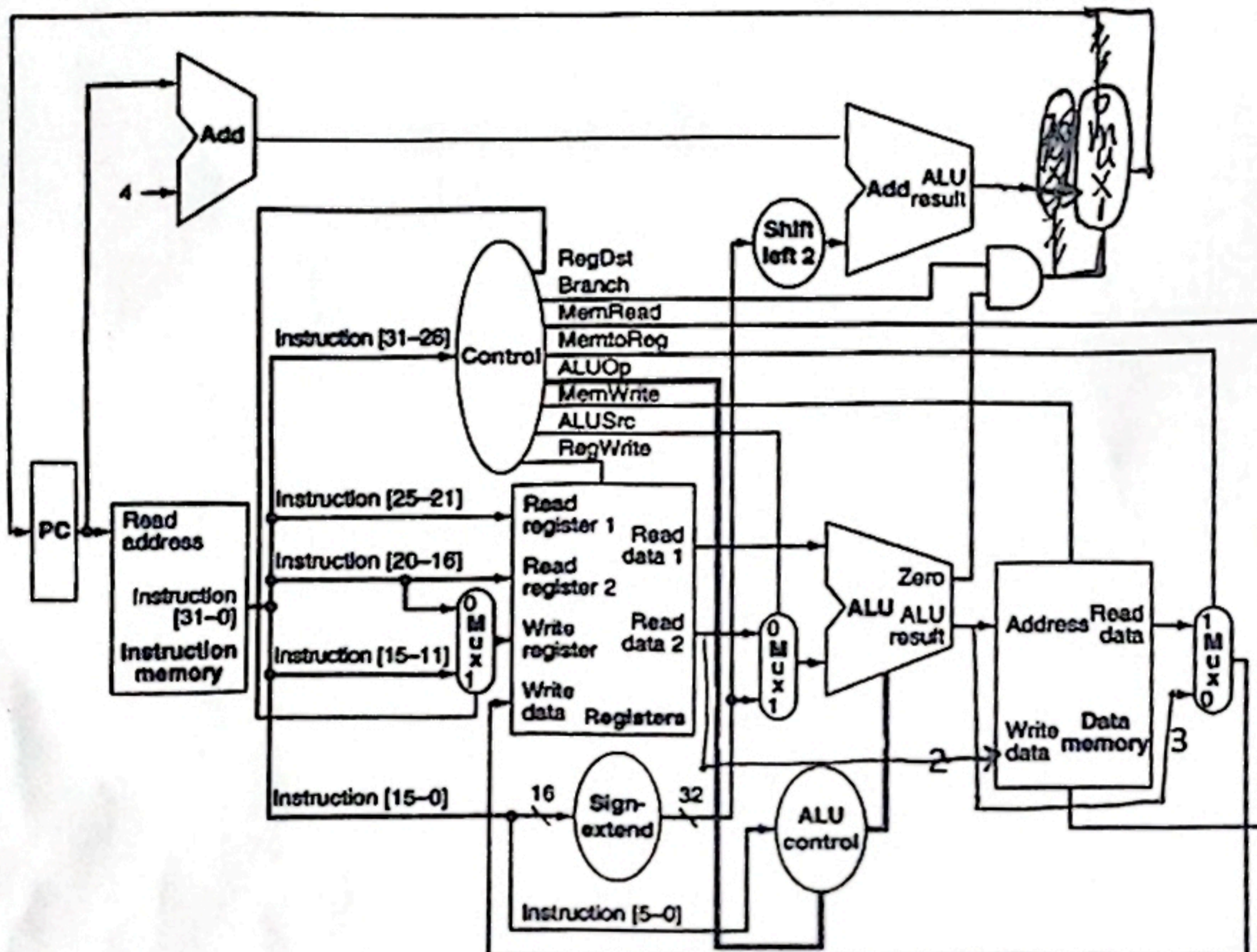
200152013
学号: 姓名: 黄奕林

1. 对以下非流水的 MIPS 数据通路

1) 补充完整图中标注为 1~3 的部分;

2) 当执行 `sw $s1, 100($s2)` 时, 请填写下面表格中的控制信号值:。

3) 如果 $\$s1=100$ 、 $\$s2=55$, 此时 ALU 的输出 ALUresult 信号和数据存储器的 Writedata 输入信号的数值是多少?



(2)

RegDst	Branch	MemRead	MemtoReg	MemWrite	ALUSrc	RegWrite	ALUOp
X	0	0	X	1	1	0	0

(3) ALUresult: 155
Writedata: 100

2. 下面是一段 MIPS 指令序列:

- 1 lw \$t1, 10(\$t0)
- 2 sw \$t1, 20(\$t0)
- 3 sub \$t2, \$s1, \$t3
- 4 add \$t1, \$t1, \$t2
- 5 or \$t1, \$t1, \$t2

假定在一个采用课本中“取指、译码/取数、执行、访存、写回”的五段流水线处理器中执行上述指令序列, 请回答下列问题:

(1) 如果没有前推电路, 以上指令序列中, 哪些指令之间发生数据冒险?

(2) 如果采用“旁路/前推”技术, 是否可以完全解决数据冒险? 不行的话, 需要在何处、加入几条 nop 指令才能使这段指令序列的执行避免数据冒险?

(1) ① 1.2 之间 ② 3.4 之间 ③ 3.5 之间 ④ 4.5 之间

(2) 不能. 因为 1.2 之间是取数操作, 无法使用“旁路/前推”技术解决. 需要在 1.2 之间加入一条 nop 指令才能避免.

3. 表1给出在MIPS汇编语言中,各寄存器的映射和使用约定。表2给出部分MIPS指令的每个字段的值。

以下C赋值语句用于对某个数组元素的值进行增量操作:

$\text{Deposit}[1] = \text{amount} + \text{Deposit}[1];$

其中数组Deposit的基址存放在\$t3中,增量amount存放在\$s6中。

表1. MIPS中寄存器的映射和使用。

寄存器	寄存器号	使用	寄存器	寄存器号	使用
\$a0-\$a3	4 - 7	传递参数	\$gp	28	静态数据的全局指针寄存器
\$v0, \$v1	2 - 3	返回结果值	\$sp	29	堆栈指针寄存器
\$t0-\$t7	8 - 15	临时寄存器	\$fp	30	帧指针寄存器-保存过程帧的第一个字
\$t8-\$t9	24 - 25	临时寄存器	\$ra	31	返回地址寄存器
\$s0-\$s7	16 - 23	保存参数			

表2. MIPS指令的每个字段及其值

指令	格式	Op (6bit)	Rs (5bit)	Rt (5bit)	Rd (5bit)	Shamt (5bit)	Funct (6bit)	Address (16bit)
add	R	0	reg	reg	reg	0	32 ₍₁₀₎	n. a.
sub	R	0	reg	reg	reg	0	34 ₍₁₀₎	n. a.
and	R	0	reg	reg	reg	0	36 ₍₁₀₎	n. a.
or	R	0	reg	reg	reg	0	37 ₍₁₀₎	n. a.
slt	R	0	reg	reg	reg	0		n. a.
addi	I	8 ₍₁₀₎	reg	reg	n. a.	n. a.	n. a.	常数
lw	I	35 ₍₁₀₎	reg	reg	n. a.	n. a.	n. a.	address
sw	I	43 ₍₁₀₎	reg	reg	n. a.	n. a.	n. a.	address
beq	I	4 ₍₁₀₎	reg	reg	n. a.	n. a.	n. a.	address

请回答以下问题:

(1) 编译后生成的MIPS代码是什么?

(2) 请写出指令“add\$t1, \$s5, \$t2”的机器语言二进制代码(机器码请按表2的字段进行分割)

~~(1) add \$t3, \$t3, 4~~

~~add \$t3, \$t3~~

~~(1) add \$t4, \$t3, 4~~

~~lw \$t4, 0(\$t4)~~

~~add \$t4, \$t4~~

(1) add \$t4, \$t3, 4 // 计算Deposit[1]地址

lw \$t5, 0(\$t4) // 读取数据

add \$t5, \$t5, \$s6 // 增量操作

sw \$t5, 0(\$t4) // 存回Deposit[1]

(2) 000000 1010 01010 0100 00000 100000