1 请画出课本（单发射）数据通路中寄存器文件的示意图，标出输出输出信号和控制信号

2写出课本中基本数据通路中控制电路的输入输出信号

输入信号Instruction[ ]

输出信号：

3 （共30分）写出下列指令的机器码（add/lw/sw操作码分别为35/43/32，t0寄存器号为8，s0寄存器号为16）。（18分）

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | 6bit（高位） | 5bit | 5bit | 5bit | 5bit | 6bit(低) |
| 0x0200 | add $t1,$t2,$t3 |  |  |  |  |  |  |
| 0x0204 | lw $t0,(0x1000)$s1 |  |  |  |  |  |  |
| 0x0208 | sw $t2,(0x1004)$s1 |  |  |  |  |  |  |

如果此时CPU的PC寄存器输出为0x0204，s1=0x100，内存单元0x1100上保存的机器字=055，01104地址上保存的是0x66。请给出处理其中以下信号的值。

指令存储器输出的Instruction:

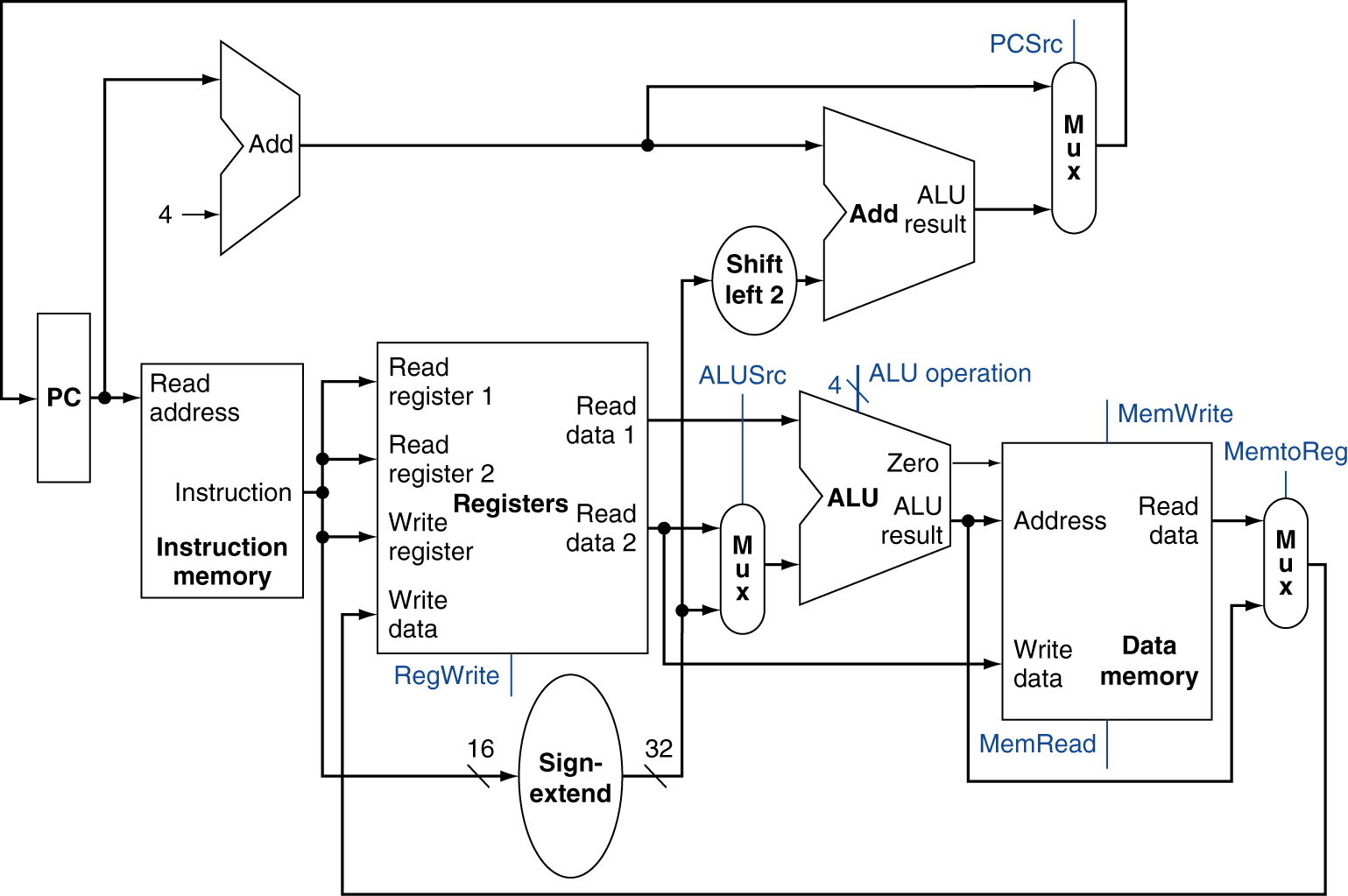
寄存器文件的readdata1:

寄存器文件的writedata:

Sign-extend输出的32位信号：:

ALU输出的ALU result信号：

PC的输入信号：



上述的数据通路，能否支持j指令的功能？如果能，请说明理由，如果不能请画出修改部分的电路结构示意图。

4 请按下图方式完成流水线的时空图，如果有流水线气泡，请注明”stall”

sll r8,r17,2

addi r21,r0,a

add r8,r8,r19

addi r22,r0,b

sll r8,r8,3

addi r32,r0,c

add r0,r8,r21

IF

ID

EXE

MEM

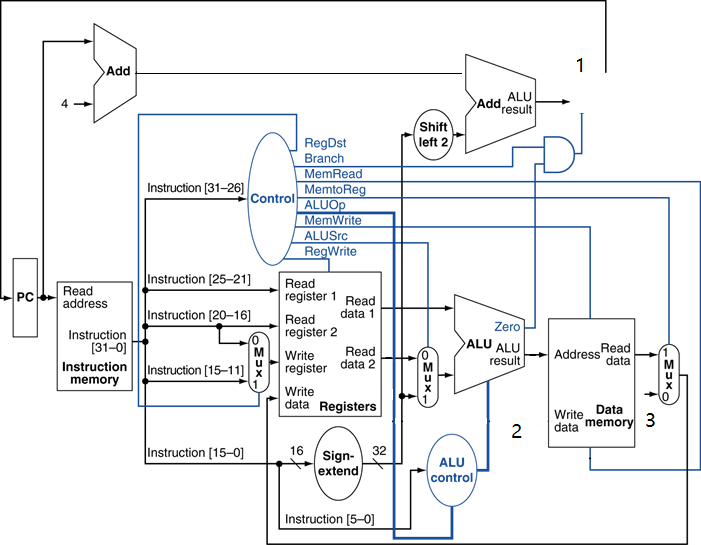
WB

1. 写出流水数据通路上可以检测数据冒险中EX冒险的判定条件逻辑表达式（其中的RS和RT相关的复选器控制信号分别为ForwardA和ForwardB）。
2. 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？



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| RegDst | Branch | MemRead | MemtoReg | MemWrite | ALUSrc | RegWrite | ALUOp |
|  |  |  |  |  |  |  |  |

7. 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？

8. 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：

Deposit [1] = amount + Deposit [1];

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

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）

8 如果非流水MIPS数据通路的时钟周期为100ns，五级流水的MIPS数据通路的时钟周期为20ns，执行完成10条无数据冒险的R型指令，各自需要多少时间？

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 |