一、题目

- 在基本的单周期MIPS实现中，不同的指令使用不同的硬件单元。
- 根据如下指令回答下列3个问题。

	指令	解释
a.	add Rd,Rs,Rt	Reg[Rd]=Reg[Rs]+Reg[Rt]
b.	lw Rt,Offs(Rs)	Reg[Rt]=Mem[Reg[Rs]+Offs]

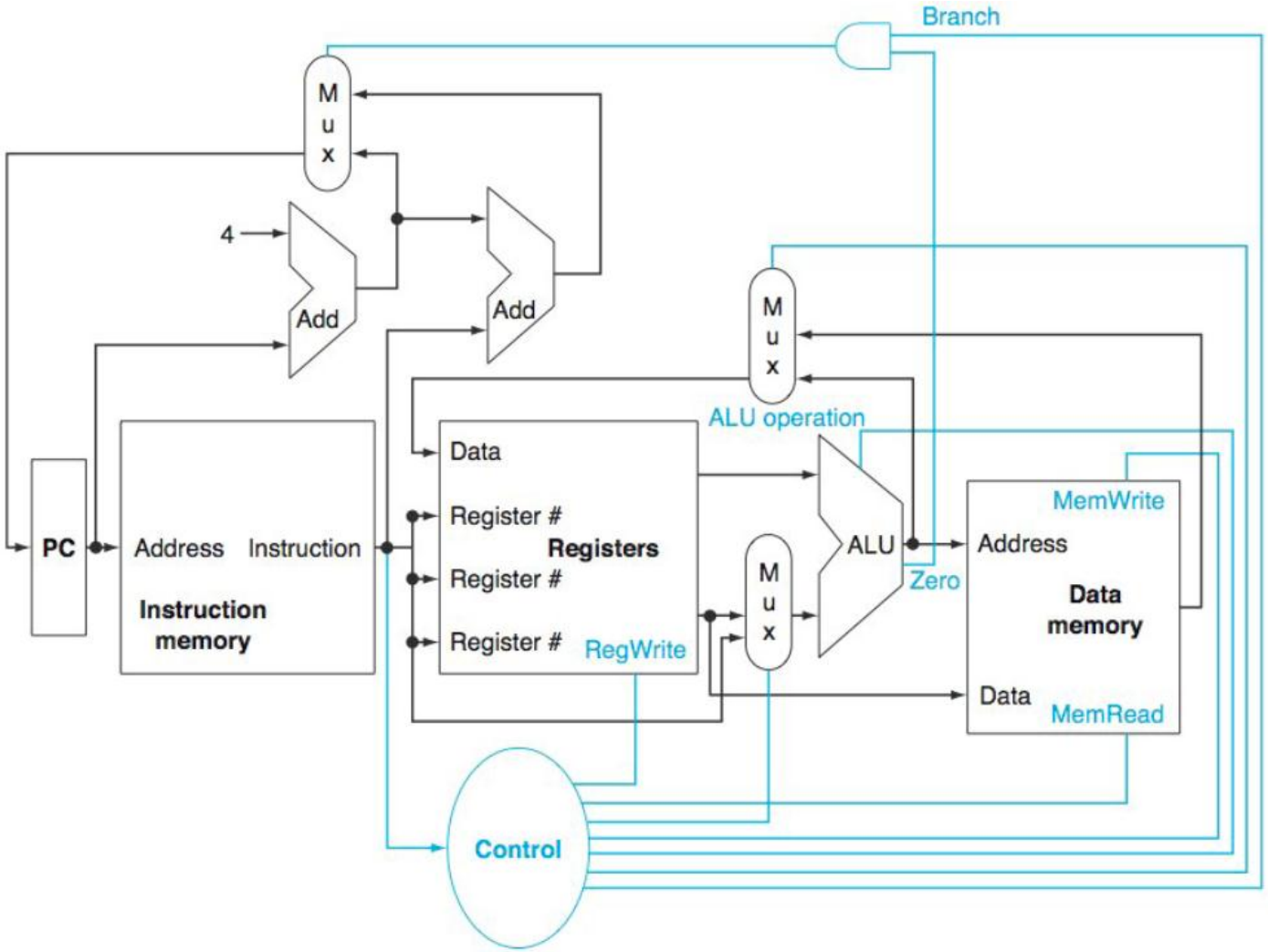
① 对上述指令而言，图1中的控制单元要产生哪些控制信号？

② 对上述指令而言，要用到哪些功能单元？

③ 哪些功能单元会产生输出，但输出不会被以上指令用到？对以上指令而言，哪些功能单元不产生任何输出？

一、题图

❖ 图1


Branch
M
u
x
4
Add
Add
M
u
x
ALU operation
PC
Address
Instruction
Instruction
memory
Data
Register #
Registers
Register #
Register #
RegWrite
M
u
x
ALU
Zero
MemWrite
Address
Data
memory
Data
MemRead
Control

一、解答 1

	指令	解释
a.	add Rd,Rs,Rt	Reg[Rd]=Reg[Rs]+Reg[Rt]
b.	lw Rt,Offs(Rs)	Reg[Rt]=Mem[Reg[Rs]+Offs]

① **对上述指令而言，图1中的控制单元要产生哪些控制信号？**

- **控制单元产生的信号为：**

	RegWrite	MemRead	ALUMux	MemWrite	ALUOp	RegMux	Branch
a.	1	0	0(Reg)	0	Add	0(ALU)	0
b.	1	1	1(Imm)	0	Add	1(Mem)	0

- **ALUMux代表连接在ALU输入端的多选器的控制信号。其中， 0(Reg)代表多选器选择的是寄存器堆的输出；1(Imm)代表多选器选择的是指令存储器中的立即数。**
- **RegMux代表连接在寄存器堆的Data输入端的多选器的控制信号。其中，0(ALU)代表多选器选择的是ALU的输出，1(Mem)代表多选器选择的是数据存储器的输出。**

一、解答2

	指令	解释
a.	add Rd,Rs,Rt	Reg[Rd]=Reg[Rs]+Reg[Rt]
b.	lw Rt,Offs(Rs)	Reg[Rt]=Mem[Reg[Rs]+Offs]

② **对上述指令而言，会用到哪些功能单元？**

- **对于指令a. 除数据存储器和分支加法器(图1中靠右侧的Add)之外的所有单元**
- **对于指令b. 除分支加法器之外的所有单元**

一、解答3

	指令	解释
a.	add Rd,Rs,Rt	Reg[Rd]=Reg[Rs]+Reg[Rt]
b.	lw Rt,Offs(Rs)	Reg[Rt]=Mem[Reg[Rs]+Offs]

③ **哪些功能单元会产生输出，但输出不会被以上指令用到？对以上指令而言，哪些功能单元不产生任何输出？**

	产生输出但没有用到	没有输出
a.	分支加法器	数据存储器
b.	分支加法器，寄存器堆的第二个输出端 (图1中寄存器堆两个输出端中下面的一个)	无

一、题目

- 不同单元有不同的延迟时间。在图1中有七种主要单元。
- 对一条指令而言，关键路径(产生最长延迟的那条路径)上各个单元的延迟时间决定了该指令的最小延迟。
- 假设个单元的延迟时间如下表所示，回答下列3个问题。

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	控制
a.	400ps	100ps	30ps	120ps	200ps	350ps	100ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	65ps

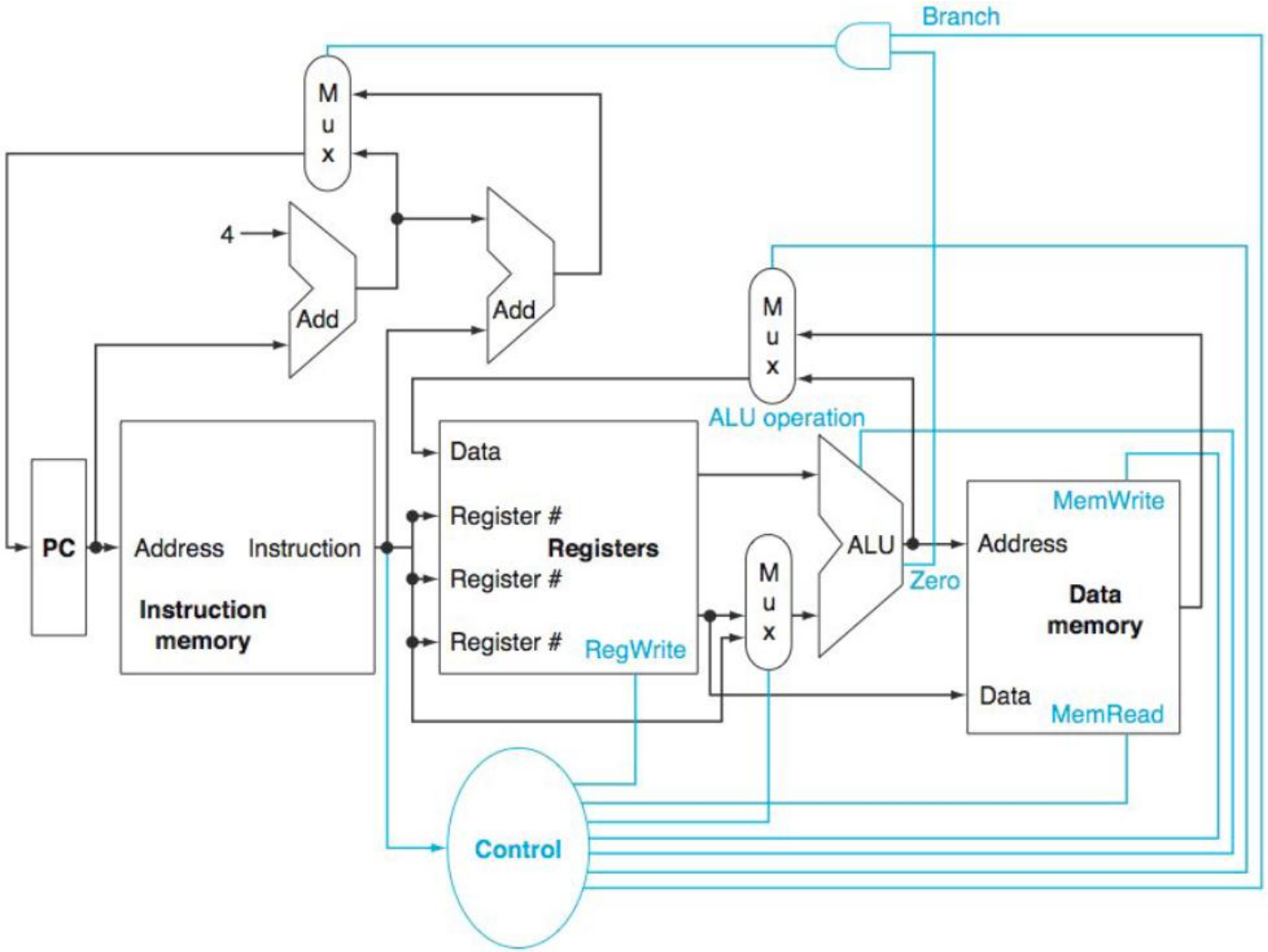
④ 对一条MIPS的与指令(AND)而言，关键路径是什么？

⑤ 对一条MIPS的装载指令(LW)而言，关键路径是什么？

⑥ 对一条MIPS的相等则分支指令(BEQ)而言，关键路径是什么？

一、题图

❖ 图1


Branch
M
u
x
4
Add
Add
M
u
x
ALU operation
PC
Address
Instruction
Instruction
memory
Data
Register #
Registers
Register #
Register #
RegWrite
M
u
x
ALU
Zero
MemWrite
Address
Data
memory
Data
MemRead
Control

一、解答4

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	控制
a.	400ps	100ps	30ps	120ps	200ps	350ps	100ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	65ps

④ 对一条MIPS的与指令(AND)而言，关键路径是什么？

- **a.关键路径为：I-Mem、Regs(Read)、Mux、ALU、Mux、Regs(Write)**
- **b.关键路径为：I-Mem、Regs(Read)、Mux、ALU、Mux、Regs(Write)**
- **解析：对于AND指令(and rd, rs, rt)，存在这样一条长路径：读指令、读寄存器堆、通过ALUMux多选器、进行ALU运算、通过RegMux多选器、写寄存器堆(即I-Mem、Regs(Read)、Mux、ALU、Mux、Regs(Write))。另外一条长路径与之类似，但是这条路径是在寄存器堆进行读操作时通过控制器的，即：I-Mem、Control、Mux、ALU、Mux、Regs(Write)，但由于控制器的速度快于寄存器堆，因而前者为关键路径，其它的路径都短于这两条路径。**

一、解答5

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	控制
a.	400ps	100ps	30ps	120ps	200ps	350ps	100ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	65ps

⑤ **对一条MIPS的装载指令(LW)而言，关键路径是什么？**

- **a.关键路径为：I-Mem、Regs(Read)、Mux、ALU、D-Mem(Read)、Mux、Regs(Write)**
- **b.关键路径为：I-Mem、Regs(Read)、Mux、ALU、D-Mem(Read)、Mux、Regs(Write)**
- **解析：对于LW指令，存在这样一条长路径：读指令、读寄存器堆获得基址、使用多选器选择立即数作为ALU的输入、使用ALU计算地址、访问数据存储器、使用多选器选择存储器输出作为寄存器的数据输入、写寄存器堆，故有路径I-Mem、Regs(Read)、Mux、ALU、D-Mem(Read)、Mux、Regs(Write)。还有一条与之类似的长路径，但是这条路径是通过控制器而不是寄存器堆的 (用于生成ALUMux的控制信号)，由于控制器的速度快于寄存器堆，于是前者为关键路径，除这两条之外的路径都是比较短的路径。**

一、解答6

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	控制
a.	400ps	100ps	30ps	120ps	200ps	350ps	100ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	65ps

⑥ **对一条MIPS的相等则分支指令(BEQ)而言，关键路径是什么？**

- **a.由于控制器的速度快于寄存器堆，故关键路径为：I-Mem、Regs(Read)、Mux、ALU、Mux**
- **b.由于控制器的速度快于寄存器堆，故关键路径为：I-Mem、Regs(Read)、Mux、ALU、Mux**
- **解析：这条指令有两种长路径——决定分支条件以及计算新PC值。对于决定分支条件的路径，需要读指令、读寄存器堆或使用控制单元、使用ALUMux、使用ALU比较两个值、使用ALU的零输出端来控制选择新PC值的多选器。对于计算新PC值的路径，其中一条是PC值加4(Add)、加偏移量offset(Add)、选择这个值作为新的PC值(Mux)；另一条是读指令(为了取得偏移量)、使用分支加法单元和相应的多选器。但是这两条计算PC值中的路径都比决定分支条件的路径要短，这是因为从表中可以看到指令存储器的速度要慢于执行PC+4的加法器、ALU的速度要慢于分支加法器。**

二、题目

- 图1中基本的单周期MIPS实现仅能实现某些指令。
- 可以在这个指令集中加入新的指令，但决定是否加入取决于给处理器的数据通路和数据通路增加的复杂度。
- 对于下表中的新指令而言，试回答下列3个问题。

	指令	解释
a.	add3 Rd,Rs,Rt,Rx	Reg[Rd]=Reg[Rs]+Reg[Rt]+Reg[Rx]
b.	sll Rt,Rd,Shift	Reg[Rd]=Reg[Rt]<<Shift(左移)

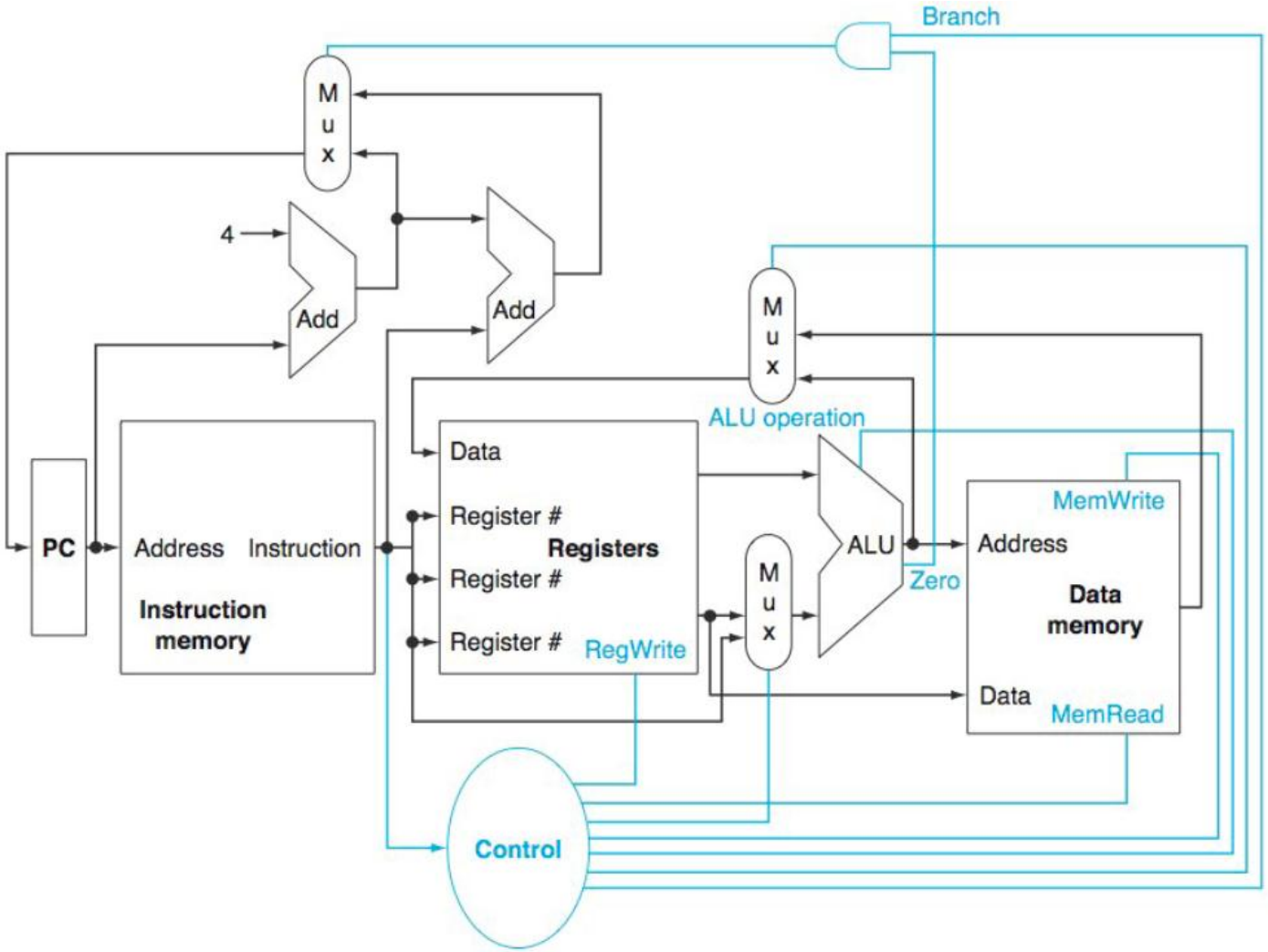
① 对上述指令而言，哪些已有的单元还可以被使用？

② 对上述指令而言，还需要增加哪些功能单元？

③ 为了支持这些指令，需要在控制单元增加哪些信号？

二、题图

❖图1


Branch
M
u
x
4
Add
Add
M
u
x
ALU operation
PC
Address
Instruction
Instruction
memory
Data
Register #
Registers
Register #
Register #
RegWrite
M
u
x
ALU
Zero
MemWrite
Address
Data
memory
Data
MemRead
Control

二、解答1

	指令	解释
a.	add3 Rd,Rs,Rt,Rx	Reg[Rd]=Reg[Rs]+Reg[Rt]+Reg[Rx]
b.	sll Rt,Rd,Shift	Reg[Rd]=Reg[Rt]<<Shift(左移)

① **对上述指令而言，哪些已有的单元还可以被使用？**

- **a.除分支加法器、数据存储器之外的所有单元**
- **b.除分支加法器、数据存储器之外的所有单元，此外，ALU可以选择继续使用也可以选择不继续使用，具体要视第②问的回答，如果增加一个专门的移位器，则不需要使用ALU**

二、解答2

	指令	解释
a.	add3 Rd,Rs,Rt,Rx	Reg[Rd]=Reg[Rs]+Reg[Rt]+Reg[Rx]
b.	sll Rt,Rd,Shift	Reg[Rd]=Reg[Rt]<<Shift(左移)

② **对上述指令而言，还需要增加哪些功能单元？**

- **a.寄存器堆增加一个输出端 和一个相应的读地址输入端，以读出Rx；增加一个加法器(或为现有的ALU增加一个输入端)，加法器的一个输入端连接至寄存器堆新增的输出端，另一个输入端连接至ALU的输出端。如果采用增加一个加法器的方案，则还需增加寄存器堆数据输入选通器的一个输入端，并连接至加法器的输出端**
- **b.增加一个移位器(或为现有的ALU增加移位运算功能)，移位器的输入端连接至寄存器堆的一个输出端。如果采用增加一个移位器的方案，则还需增加寄存器堆数据输入选通器的一个输入端，并连接至移位器的输出端**

二、解答3

	指令	解释
a.	add3 Rd,Rs,Rt,Rx	Reg[Rd]=Reg[Rs]+Reg[Rt]+Reg[Rx]
b.	sll Rt,Rd,Shift	Reg[Rd]=Reg[Rt]<<Shift(左移)

③ 为了支持这些指令，需要在控制单元增加哪些信号？

- **a.如果增加一个加法器，则需增加寄存器堆数据输入选通器的控制信号，以实现数据通道3选1；如果为现有的ALU增加一个输入端，则需增加针对三输入端的ALU的功能控制信号定义，使其可控制新增的ADD3操作**
- **b.如果增加一个移位器，则需增加寄存器堆数据输入选通器的控制信号，以实现数据通道3选1(如同时考虑a和b中的两条指令，则为4选1)；如为现有的ALU增加移位运算功能，则需增加ALU的功能控制信号定义，使其可控制新增的移位操作**

二、题目

- 当设计者考虑改进处理器数据通路时，往往要考虑性能与成本的折中。假设我们从图1的数据通路出发，其中指令存储器(Instruction Memory)、加法器(Add)、多选器(Mux)、ALU、寄存器堆(Registers)、数据寄存器(Data Memory)和控制单元(Control)的延迟分别为400ps、100ps、30ps、120ps、200ps、350ps和100ps，相应的成本分别为1000、30、10、100、200、2000和500。试根据表中的改进分别回答下列问题。

	改进	延迟	成本	优势
a.	更快的加法器	加法单元-20ps	每个加法单元+20	把已有的加法器用更快的加法器替代
b.	更大的寄存器堆	寄存器堆 +100ps	寄存器堆+200	需要更少的load和store指令。这将导致指令数减少5%

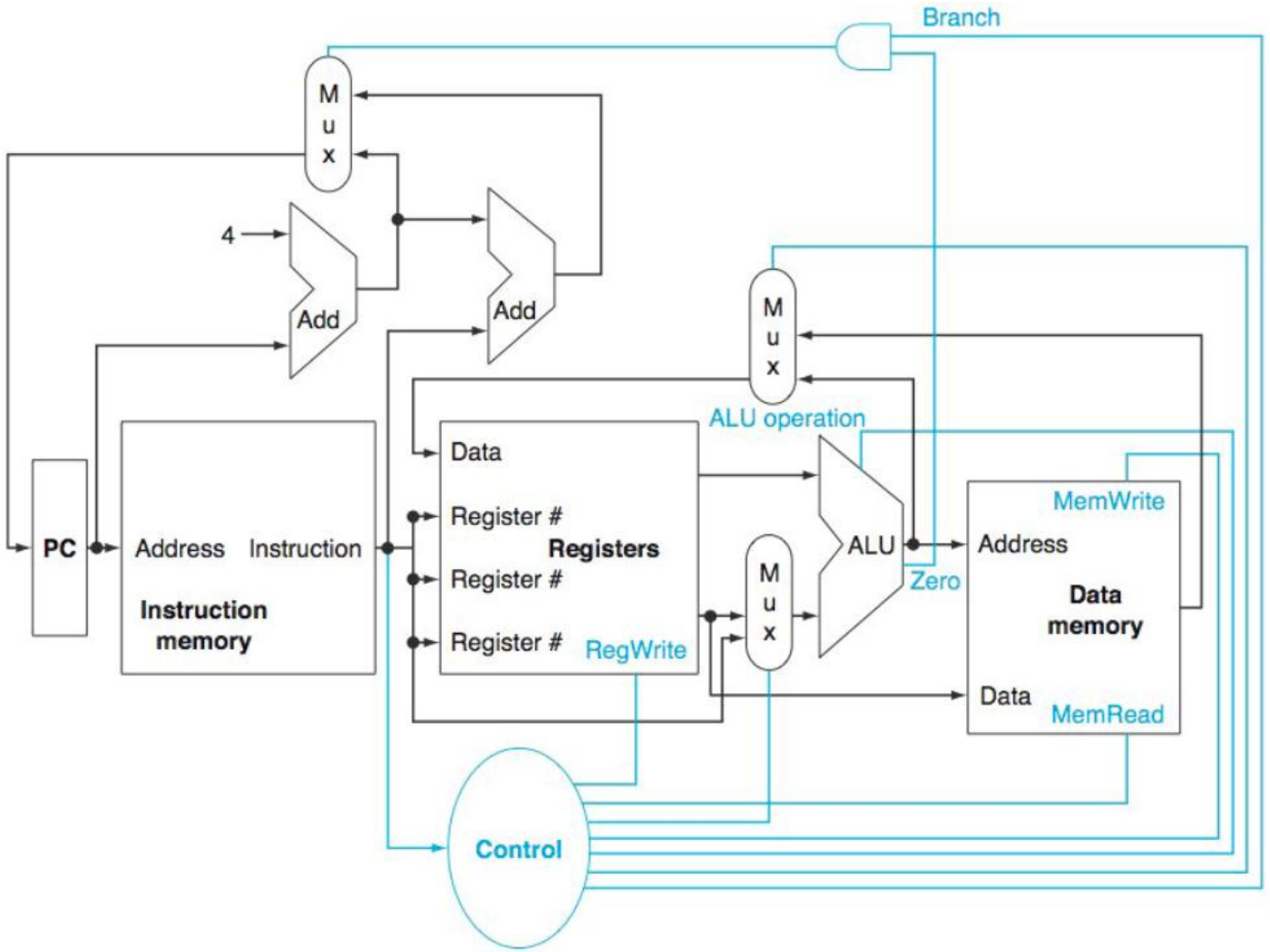
④ 改进前后的时钟周期分别是多少？

⑤ 改进后将获得多大的加速比？

⑥ 比较改进前后的性能/价格比，进行这样的改进是否有意义？

二、题图

❖ 图1


Branch
M
u
x
4
Add
Add
M
u
x
ALU operation
PC
Address
Instruction
Instruction
memory
Data
Register #
Registers
Register #
Register #
RegWrite
M
u
x
ALU
Zero
MemWrite
Address
Data
memory
Data
MemRead
Control

二、解答4

	改进	延迟	成本	优势
a.	更快的加法器	加法单元-20ps	每个加法单元+20	把已有的加法器用更快的加法器替代
b.	更大的寄存器堆	寄存器堆+100ps	寄存器堆+200	需要更少的load和store指令。这将导致指令数减少5%

④ 改进前后的时钟周期分别是多少？

- **a. 由于加法单元不在关键路径上，因此对加法器的改进不影响时钟周期。**
- **b. 寄存器堆位于关键路径上，因而，使用更大的寄存器堆后，时钟周期变为1330ps + 2 × 100ps = 1530ps。**
- **解析：时钟周期是由关键路径决定的，这里的关键路径为：I-Mem(读指令)、Regs(Read)(由于寄存器堆的延迟大于控制器，因而寄存器堆位于关键路径上)、Mux(选择ALU的输入)、ALU、Data Memory(Read)、Mux(选择存储器写入到寄存器堆中的数据)、Regs(Write)(数据写入寄存器堆)，该路径的延迟为400ps + 200ps + 30ps + 120ps + 350ps + 30ps + 200ps = 1330ps。**

二、解答5

	改进	延迟	成本	优势
a.	更快的加法器	加法单元-20ps	每个加法单元+20	把已有的加法器用更快的加法器替代
b.	更大的寄存器堆	寄存器堆+100ps	寄存器堆+200	需要更少的load和store指令。这将导致指令数减少5%

⑤ **改进后将获得多大的加速比？**

- **a.加速比由时钟周期本身的变化以及需要执行的时钟周期数目共同决定，对加法器的改进不影响时钟周期，并且，需要执行的时钟周期数目也不变，因此加速比为1.000**
- **b.需要的指令数减少5%，需要的时钟周期数目也相应减少5%，同时，时钟周期由1330ps增加为1530ps，因而加速比为 (1/0.95)×(1330/1530) = 0.915**

二、解答6

	改进	延迟	成本	优势
a.	更快的加法器	加法单元-20ps	每个加法单元+20	把已有的加法器用更快的加法器替代
b.	更大的寄存器堆	寄存器堆+100ps	寄存器堆+200	需要更少的load和store指令。这将导致指令数减少5%

⑥ **比较改进前后的性能/价格比，进行这样的改进是否有意义？**

- **a. 原来的处理器的总成本为1000(I-Mem) + 200(Regs) + 500(Control) + 100(ALU) + 2000(D-Mem) + 2×30(2个加法单元) + 3×10(3个多选器) = 3890，更换加法器之后的总成本为3890 + 2 × 20 = 3930，相对成本为3930 / 3890 = 1.010，性能/价格比为1.000 / 1.010 = 0.990，成本增加但性能没有提升。**
- **b. 使用更大的寄存器堆的成本为3890 + 200 = 4090，相对成本为4090 / 3890 = 1.051，性能/价格比为 0.915 / 1.051 = 0.871，说明用更大的投入反而换来了性能的下降。**

三、题目

- 下表给出了实现处理器数据通路的逻辑单元延迟。试根据下表的两种情况分别回答下列问题。

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	符号扩展	左移两位
a.	400ps	100ps	30ps	120ps	200ps	350ps	20ps	2ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps

① 如果处理器只需做连续取指这一件事(见图2)，那么时钟周期是多少？

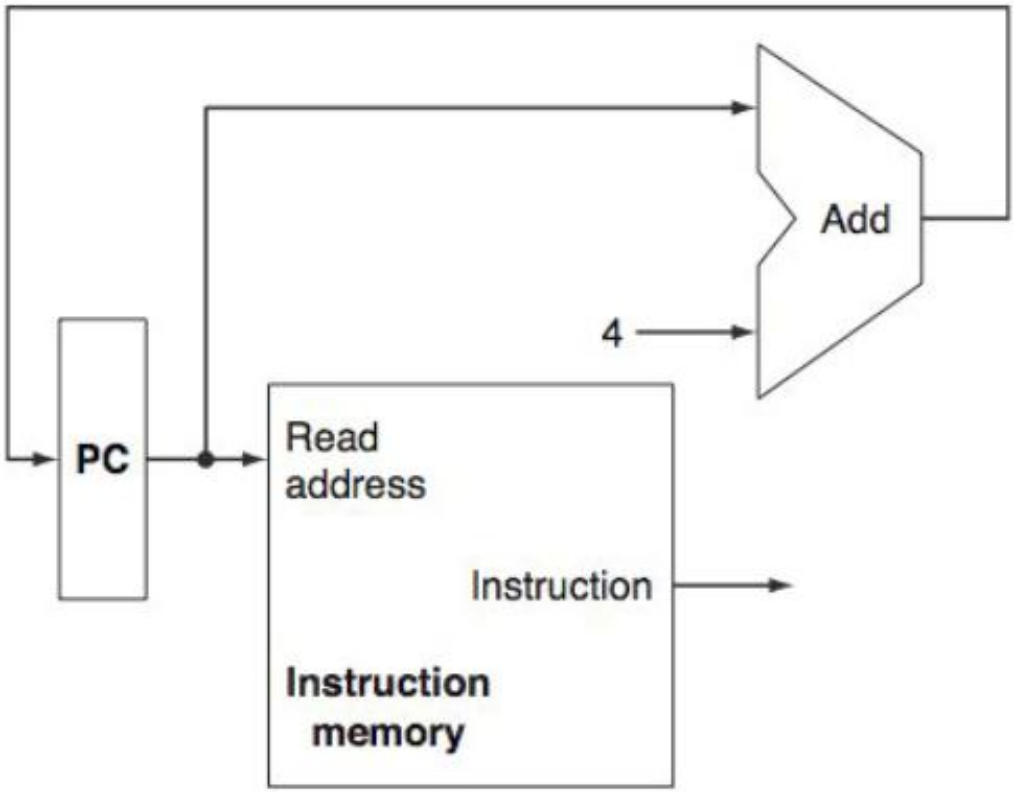
② 考虑一个与图3类似的数据通路，但是假设处理器只需处理无条件相对跳转指令，那么时钟周期是多少？

③ 同样考虑一个与图3类似的数据通路，但这次假设只需处理有条件相对跳转指令，那么时钟周期是多少？(请注意图3中ALU的零输出端不是与数据存储器连接，该输出与选择PC值来源的多选器的控制有关)

- 提示：图3中靠右侧的加法器延迟应当按照ALU来计算

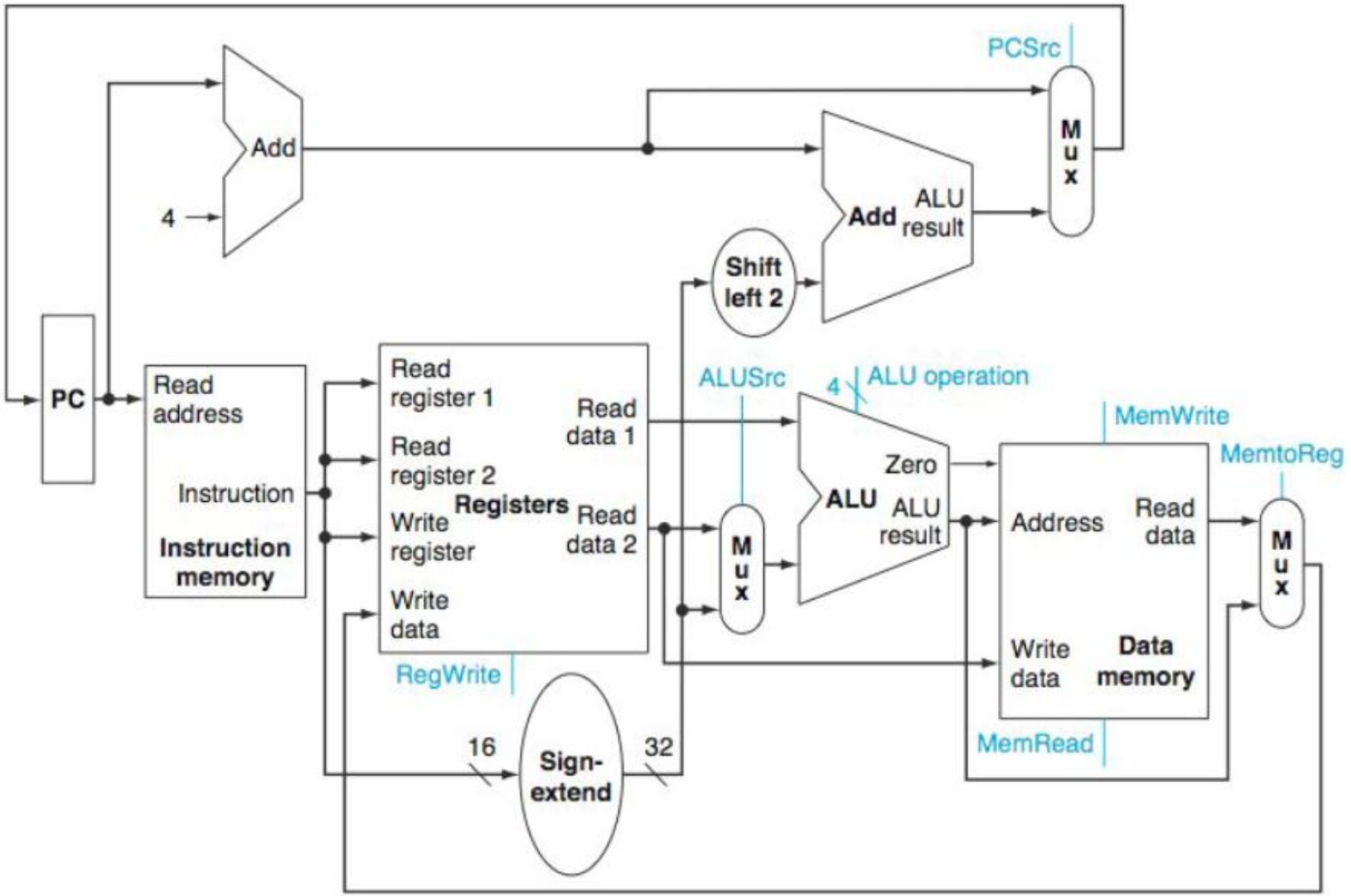
三、题图

❖图2



三、题图

❖图3

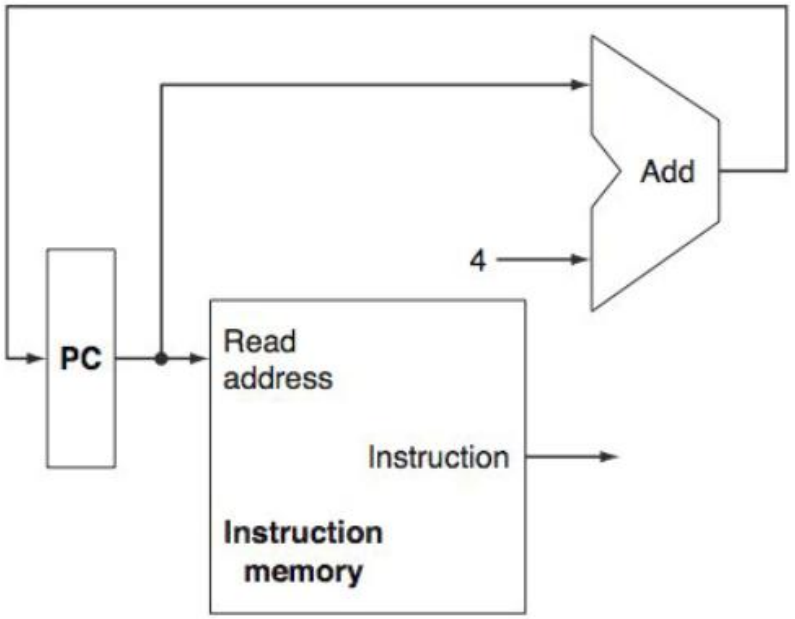

PCSrc
Add
4
M
u
x
Add
ALU
result
Shift
left 2
PC
Read
address
Instruction
Instruction
memory
Read
register 1
Read
register 2
Registers
Write
register
Write
data
Read
data 1
Read
data 2
RegWrite
ALUSrc
4
ALU operation
M
u
x
ALU
Zero
ALU
result
MemWrite
MemtoReg
Address
Read
data
Write
data
Data
memory
MemRead
M
u
x
16
Sign-
extend
32

三、解答1

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	符号扩展	左移两位
a.	400ps	100ps	30ps	120ps	200ps	350ps	20ps	2ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps

① **如果处理器只需做连续取指这一件事(见图2)，那么时钟周期是多少？**

- **a. 由于指令存储器慢于加法器，因此，时钟周期决定于指令存储器的延迟，时钟周期为400ps。**
- **b. 时钟周期为500ps。**



三、解答2

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	符号扩展	左移两位
a.	400ps	100ps	30ps	120ps	200ps	350ps	20ps	2ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps

② **考虑一个与图3类似的数据通路，但是假设处理器只需处理无条件相对跳转指令，那么时钟周期是多少？**

- **a.关键路径为I-Mem、Sign-extend、Shift-left-2、Add(ALU)、Mux，因此，时钟周期为400ps + 20ps + 2ps + 120ps + 30ps = 572ps。**
- **b.时钟周期为500ps + 90ps + 20ps + 180ps + 100ps = 890ps。**

三、解答3

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	符号扩展	左移两位
a.	400ps	100ps	30ps	120ps	200ps	350ps	20ps	2ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps

③ **同样考虑一个与图3类似的数据通路，但这次假设只需处理有条件相对跳转指令，那么时钟周期是多少？(请注意图3中ALU的零输出端不是与数据存储器连接，该输出与选择PC值来源的多选器的控制有关)**

- **a.根据题目中给出的延迟，后者为关键路径，因此时钟周期为 400ps + 200ps + 30ps + 120ps + 30ps = 780ps。**
- **b.根据题目中给出的延迟，后者为关键路径，因此时钟周期为500ps + 220ps + 100ps + 180ps + 100ps = 1100ps。**
- **解析：对于有条件相对跳转指令，除存在长路径I-Mem、Sign-extend、Shift-left-2、Add(ALU)、Mux外，还存在长路径I-Mem、Registers(Read)、Mux、ALU、Mux，关键路径为这两条路径中较长的一个。**

三、题目

❖ 根据下表的两种数据通路的逻辑单元，分别回答下列问题。

	单元
a.	执行加4的加法器(对PC)
b.	数据存储器

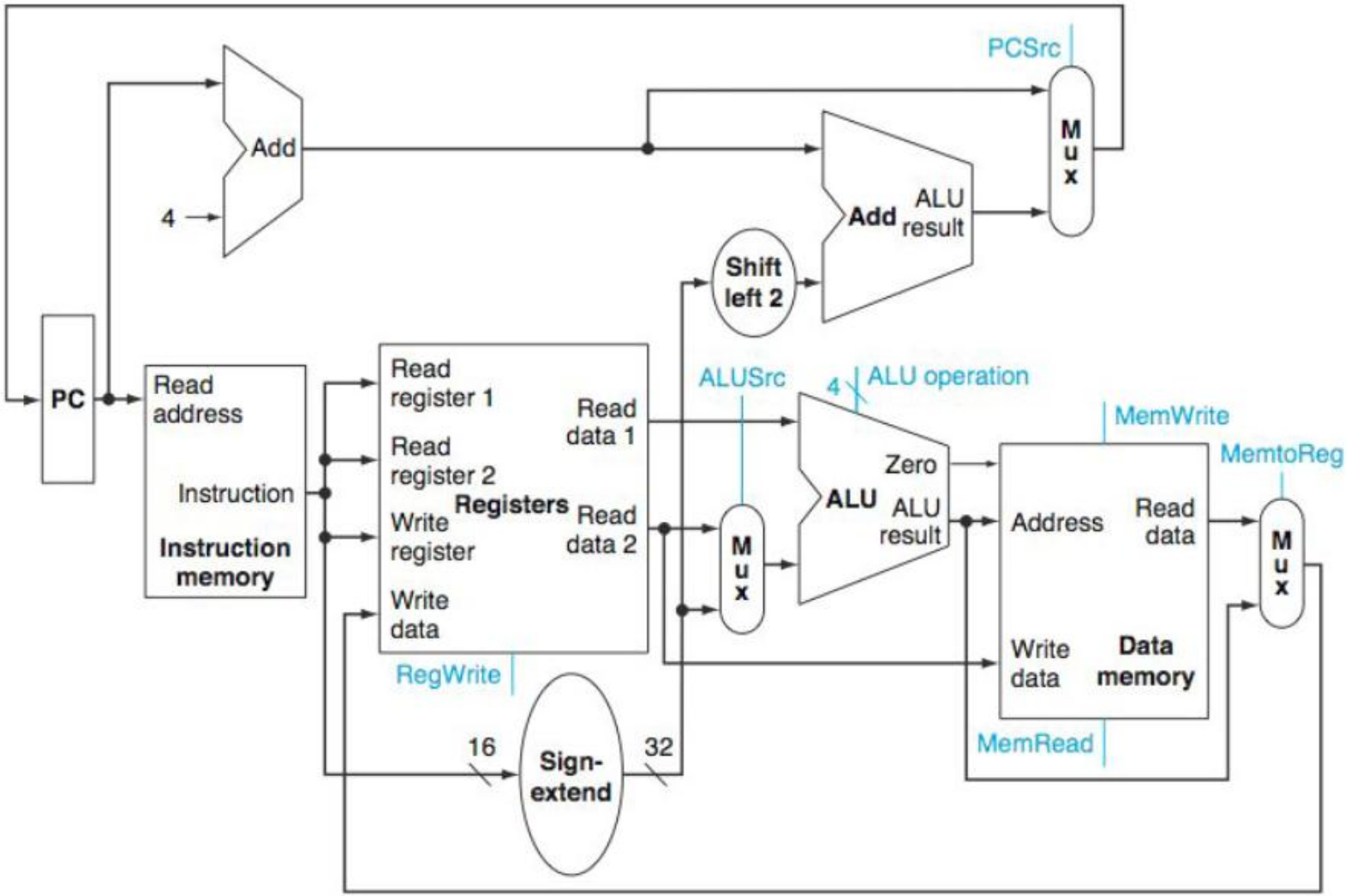
④ 哪些类型的指令需要该单元？

⑤ 对哪些类型的指令而言，该单元位于关键路径上？

⑥ 假设仅需支持beq指令和add指令，讨论该单元的延迟变化对处理器时钟周期的影响。假设其他单元的延迟不变。

三、题图

❖图3


PCSrc
Add
4
M
u
x
Add
ALU
result
Shift
left 2
PC
Read
address
Instruction
Instruction
memory
Read
register 1
Read
register 2
Registers
Write
register
Write
data
Read
data 1
Read
data 2
RegWrite
ALUSrc
4
ALU operation
M
u
x
ALU
Zero
ALU
result
MemWrite
MemtoReg
Address
Read
data
Write
data
Data
memory
MemRead
M
u
x
16
Sign-
extend
32

三、解答4

	单元
a.	执行加4的加法器(对PC)
b.	数据存储器

④ 哪些类型的指令需要该单元？

- **a.所有指令。**
- **b.与存取有关的指令，如：Lw, Sw等。**

三、解答5

	单元
a.	执行加4的加法器(对PC)
b.	数据存储器

⑤ **对哪些类型的指令而言，该单元位于关键路径上？**

- **a.所有指令的关键路径都不会包含这个加法器，因为指令存储器的速度慢于加法器，而所有的指令都必须执行读指令这一操作。**
- **b.与存取有关的指令，因为只有与存取有关的指令会用到该单元。**

三、解答6

	单元
a.	执行加4的加法器(对PC)
b.	数据存储器

⑥ **假设仅需支持beq指令和add指令，讨论该单元的延迟变化对处理器时钟周期的影响。假设其他单元的延迟不变。**

- **a.beq指令的关键路径为I-Mem、Regs(Read)、Mux、ALU、Mux，延迟为780ps，add指令的关键路径为I-Mem、Regs(Read)、Mux、ALU、Mux、Regs(Write)，延迟为980ps。这里只需讨论该单元延迟变化相比于较长的关键路径的影响，较长的关键路径为add指令，其延迟为980ps，而执行加4的加法器所在的路径为 Add、Add(ALU)、Mux，其延迟为100ps + 120ps + 30ps = 250ps，若要该单元延迟变化而导致该路径成为关键路径，从而影响时钟周期，则执行加4的加法器延迟要大于 980ps – 150ps = 830ps，才会影响时钟周期。**
- **b.数据存储器未被 beq或add指令使用，因此，它的延迟不影响时钟周期。**

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	符号扩展	左移两位
a.	400ps	100ps	30ps	120ps	200ps	350ps	20ps	2ps

四、题目

- 本题讨论数据通路中不同的单元延迟对整个数据通路时钟周期的影响，以及指令如何利用不同的数据通路单元。根据下面的两种延迟情况，分别回答下列问题。

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	符号扩展	左移两位
a.	400ps	100ps	30ps	120ps	200ps	350ps	20ps	0ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps

① 如果仅需支持ALU类指令(如add、and等)，处理器的时钟周期是多少？

② 如果仅需支持lw类指令，时钟周期是多少？

③ 如果必须支持add、beq、lw和sw指令，时钟周期是多少？

四、解答1

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	符号扩展	左移两位
a.	400ps	100ps	30ps	120ps	200ps	350ps	20ps	0ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps

① **如果仅需支持ALU类指令(如add、and等)，处理器的时钟周期是多少？**

- **a.时钟周期为400ps + 200ps + 30ps + 120ps + 30ps + 200ps = 980ps**
- **b.时钟周期为500ps + 220ps + 100ps + 180ps + 100ps + 220ps = 1320ps**
- **解析：关键路径为I-Mem、Registers(Read)、Mux(选择ALU输入)、ALU、Mux(选择寄存器写入端)、Registers(Write)**

四、解答2

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	符号扩展	左移两位
a.	400ps	100ps	30ps	120ps	200ps	350ps	20ps	0ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps

② **如果仅需支持lw类指令，时钟周期是多少？**

- **a.时钟周期为400ps + 200ps + 30ps + 120ps + 350ps + 30ps + 200ps = 1330ps**
- **b.时钟周期为500ps + 220ps + 100ps + 180ps + 1000ps + 100ps + 220ps = 2320ps**
- **解析：关键路径为I-Mem、Registers(Read)、Mux(选择ALU输入)、ALU、D-Mem(Read)、Mux(选择写入寄存器堆的数据)、Registers(Write)**

四、解答3

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	符号扩展	左移两位
a.	400ps	100ps	30ps	120ps	200ps	350ps	20ps	0ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps

③ 如果必须支持add、beq、lw和sw指令，时钟周期是多少？

- **a.时钟周期为1330ps**
- **b.时钟周期为2320ps**
- **解析：lw指令的关键路径最长，相比较而言，sw指令少使用了一个多选器并且不用向寄存器堆写数据，add和beq少使用了数据存储器**

四、题目

- 假设各类型指令所占比例如下表所示，试根据下表的两种情况分别回答下列问题。

	add	addi	not	beq	lw	sw
a.	30%	15%	5%	20%	20%	10%
b.	25%	5%	5%	15%	35%	15%

④ 数据存储器平均用了多少时钟周期？

⑤ 符号扩展电路的输入平均用了多少时钟周期？在未用到该输入的其他时间，符号扩展电路在做什么？

⑥ 如果可以将数据通路上某个单元的延迟减少10%，应该减少哪个单元的延迟？改进后整个处理器的加速比是多少？

四、解答4

	add	addi	not	beq	lw	sw
a.	30%	15%	5%	20%	20%	10%
b.	25%	5%	5%	15%	35%	15%

④ 数据存储器平均用了多少时钟周期？

- a.平均有 20% + 10% = 30% 的时钟周期里，会用到数据存储器
- b.平均有 35% + 15% = 50% 的时钟周期里，会用到数据存储器
- 解析：只有lw和sw指令会用到数据存储器

四、解答5

	add	addi	not	beq	lw	sw
a.	30%	15%	5%	20%	20%	10%
b.	25%	5%	5%	15%	35%	15%

⑤ 符号扩展电路的输入平均用了多少时钟周期？在未用到该输入的其他时间，符号扩展电路在做什么？

- 符号扩展电路实际上在每个周期都有计算结果，但是它的输出在add和not指令中被忽略了，符号扩展电路的输入只在addi指令(提供ALU需要的立即数)、beq指令(提供计算PC需要的偏移量)、lw指令和sw指令(提供寻址过程中需要的偏移量)中是需要的
- a. 结果为 15% + 20% + 20% + 10% = 65%
- b. 结果为 5% + 15% + 35% + 15% = 70%

四、解答6

	add	addi	not	beq	lw	sw
a.	30%	15%	5%	20%	20%	10%
b.	25%	5%	5%	15%	35%	15%

⑥ 如果可以将数据通路上某个单元的延迟减少10%，应该减少哪个单元的延迟？改进后整个处理器的加速比是多少？

- lw指令有最长的关键路径为：I-Mem、Registers(Read)、Mux、D-Mem(Read)、ALU、Mux、Registers(Write)，决定着时钟周期的长度。
- a.在路径中，由于指令存储器的延迟值最大，因此，如将它的延迟从400ps减小到360ps(即减少10%)，则时钟周期由1330ps减小到1290ps，加速比为1330/1290=1.031。
- b.在路径中，由于数据存储器的延迟值最大，因此，如将它的延迟从1000ps减小到900ps(即减少10%)，则时钟周期由2320ps减小到2220ps，于是加速比为2320/2220=1.045。

五、题目

- 在制造硅芯片时，材料的缺陷和制造错误会导致电路失效。一个非常普遍的问题是一根线上的信号会对相邻线上的信号产生影响，这被称为串扰。有一类串扰问题是这样的，某些线上的信号为常值(如电源线)，该线附近的线也被固定为0(stuck-at-0)或1(stuck-at-1)。试根据下表的两种缺陷(信号来自图4)分别回答下列问题。

	有问题的信号
a.	指令存储器，输出信号第7位
b.	控制单元，输出信号MemtoReg

① 设这样测试处理器的缺陷：先给PC、寄存器堆、数据和指令存储器中设置一些值(可以自己选择)，执行一条指令，然后读出PC、寄存器堆和存储器中的值；最后检查这些值以判断处理器中是否存在缺陷。你能设计这样一个方案检查该信号上是否有固定为0缺陷吗？

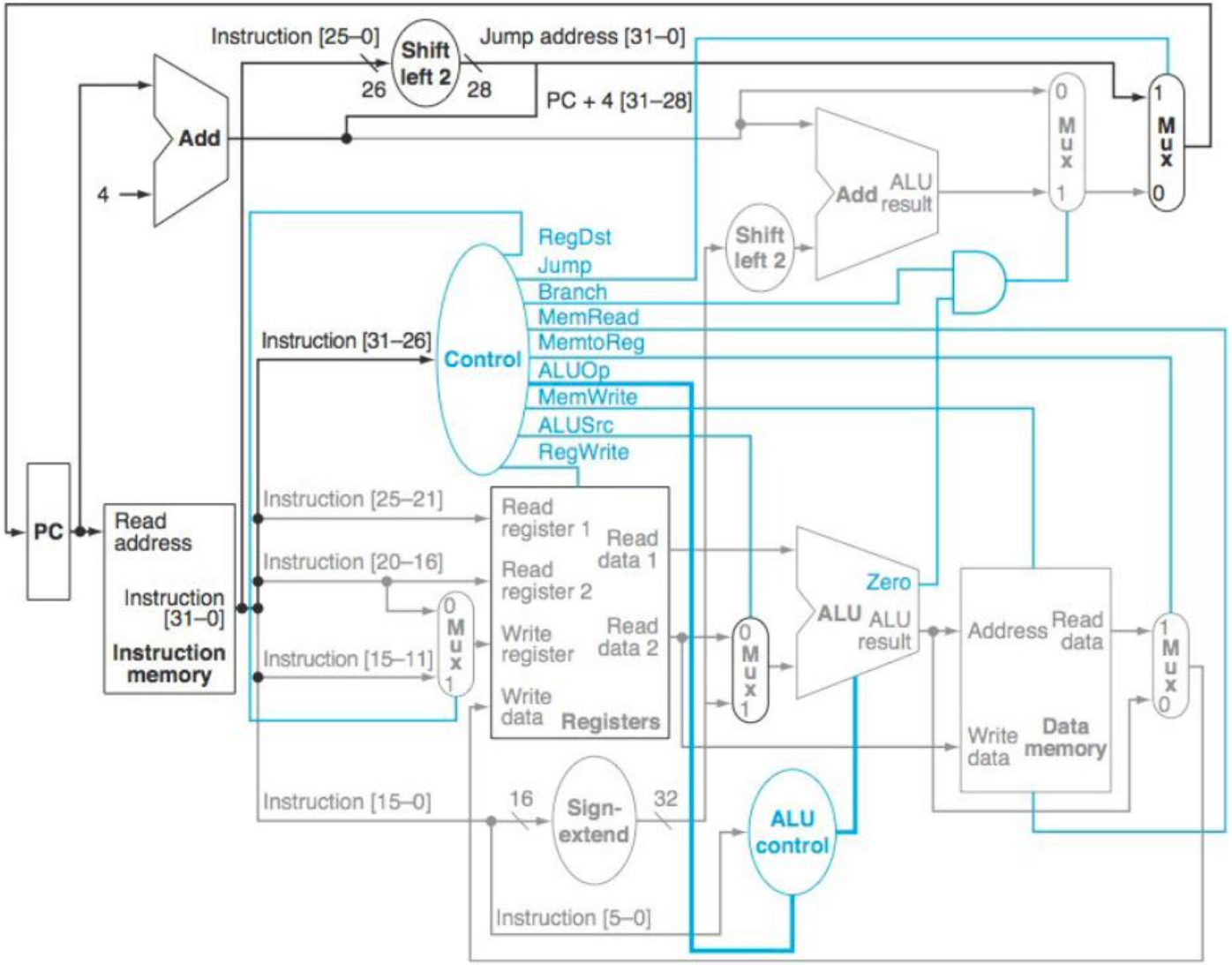
五、题目

② 条件同第①问，但是这次检查固定为1缺陷。你能只设计一个测试方案同时检查固定为0缺陷和固定为1缺陷吗？如果可以，请解释如何实现；如果不能，请说明理由。

③ 如果我们知道一个处理器在该信号上有一个固定为1缺陷，它还能用吗？为了使这个处理器仍然可用，我们必须将原来能在正常MIPS处理器上运行的程序做一些变换，使之可以在这个处理器上运行。假设指令存储器和数据存储器都很大，足够容纳变换后的程序。提示：将因为该缺陷不能用的指令替换为一系列能用的指令，这一系列指令与原指令功能相同。

五、题图

❖ 图4


Instruction [25–0]
Shift left 2
Jump address [31–0]
26
28
PC + 4 [31–28]
Add
4
0
M
u
x
1
1
M
u
x
0
Add
ALU result
Shift left 2
RegDst
Jump
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Instruction [31–26]
Control
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
Zero
ALU
ALU result
Address
Read data
Write data
Data memory
Instruction [15–0]
16
Sign-extend
32
ALU control
Instruction [5–0]

五、解答1

	有问题的信号
a.	指令存储器，输出信号第7位
b.	控制单元，输出信号MemtoReg

① **假设这样测试处理器的缺陷：先给PC、寄存器堆、数据和指令存储器中设置一些值(可以自己选择)，执行一条指令，然后读出PC、寄存器堆和存储器中的值；最后检查这些值以判断处理器中是否存在缺陷。你能设计这样一个方案检查该信号上是否有固定为0缺陷吗？**

- **a.为了测试是否有固定为0缺陷，我们需要一个指令，该指令可以将待测信号置为1，并且与0值有着不同的输出，指令存储器的第7位输出只用于指令的立即数或者偏移量部分，因而可以采用指令ADDI $1,$0,128，该指令可以将128放置到寄存器$1中，如果指令存储器的第7位输出有固定为0缺陷，那么寄存器$1中的值就会为0而不是128。**
- **b.能够将输出信号MemtoReg置为1的只有load类指令，我们可以将数据存储器中的每个字都置为0，然后执行LW $1,1024($0)，如果寄存器$1中的值不是0而是1024，说明输出信号MemtoReg有固定为0缺陷。**

五、解答2

	有问题的信号
a.	指令存储器，输出信号第7位
b.	控制单元，输出信号MemtoReg

② **条件同第①问，但是这次检查固定为1缺陷。你能只设计一个测试方案同时检查固定为0缺陷和固定为1缺陷吗？如果可以，请解释如何实现；如果不能，请说明理由。**

- **检查固定为0缺陷需要能够将该信号置为1的指令，而检查固定为1缺陷需要能够将该信号置为0的指令，由于在一个周期中一个信号不可能既为0又为1，因而不能同时检查固定为0缺陷和固定为1缺陷。若要检查固定为1缺陷，与固定为0缺陷类似，可采用如下方法：**
- **a.执行指令ADDI $1,$0,0，如果指令存储器的第7位输出有固定为1缺陷，那么寄存器$1中的值就会为128而不是0。**
- **b.这个信号的固定为1缺陷不能准确检查出来，因为所有能够将MemtoReg信号设置为0的指令都会同时将MemRead信号设置为0，这样会导致写入到寄存器$1中的数据是不确定的(与我们在检测固定为0缺陷时刻意放置的数据没有关系)，有可能最后出现在寄存器$1中的数据是与原来寄存器中的数据相同的，从而检测不出来固定为1缺陷。**

五、解答3

	有问题的信号
a.	指令存储器，输出信号第7位
b.	控制单元，输出信号MemtoReg

③ 如果我们知道一个处理器在该信号上有一个固定为1缺陷，它还能用吗？为了使这个处理器仍然可用，我们必须将原来能在正常MIPS处理器上运行的程序做一些变换，使之可以在这个处理器上运行。假设指令存储器和数据存储器都很大，足够容纳变换后的程序。提示：将因为该缺陷不能用的指令替换为一系列能用的指令，这一系列指令与原指令功能相同。

- **a.要避开指令存储器输出信号的固定为1缺陷是有可能的，但是比较麻烦，我们必须找到在输出信号第7位上为0的所有指令(一般是I型指令)，然后将这些指令中的立即数进行相应的替代。例如，对于一个第7位为0的LW $1,0($0)指令，需要用LI $1,128、SUB $1,$0,$1和LW $1,128($1)代替。**
- **b. MemtoReg信号的固定为1缺陷是不能回避的，MemtoReg信号的固定为1缺陷，会阻止除load类指令外的所有指令向寄存器堆中写入数据，而load类指令只能将数据从存储器载入到寄存器中，不能模仿ALU的运算操作。**

五、题目

❖ 根据下表的缺陷分别回答下列问题。

	缺陷
a.	固定为1
b.	如果指令的第31~26位全为0，则固定为0，否则无缺陷

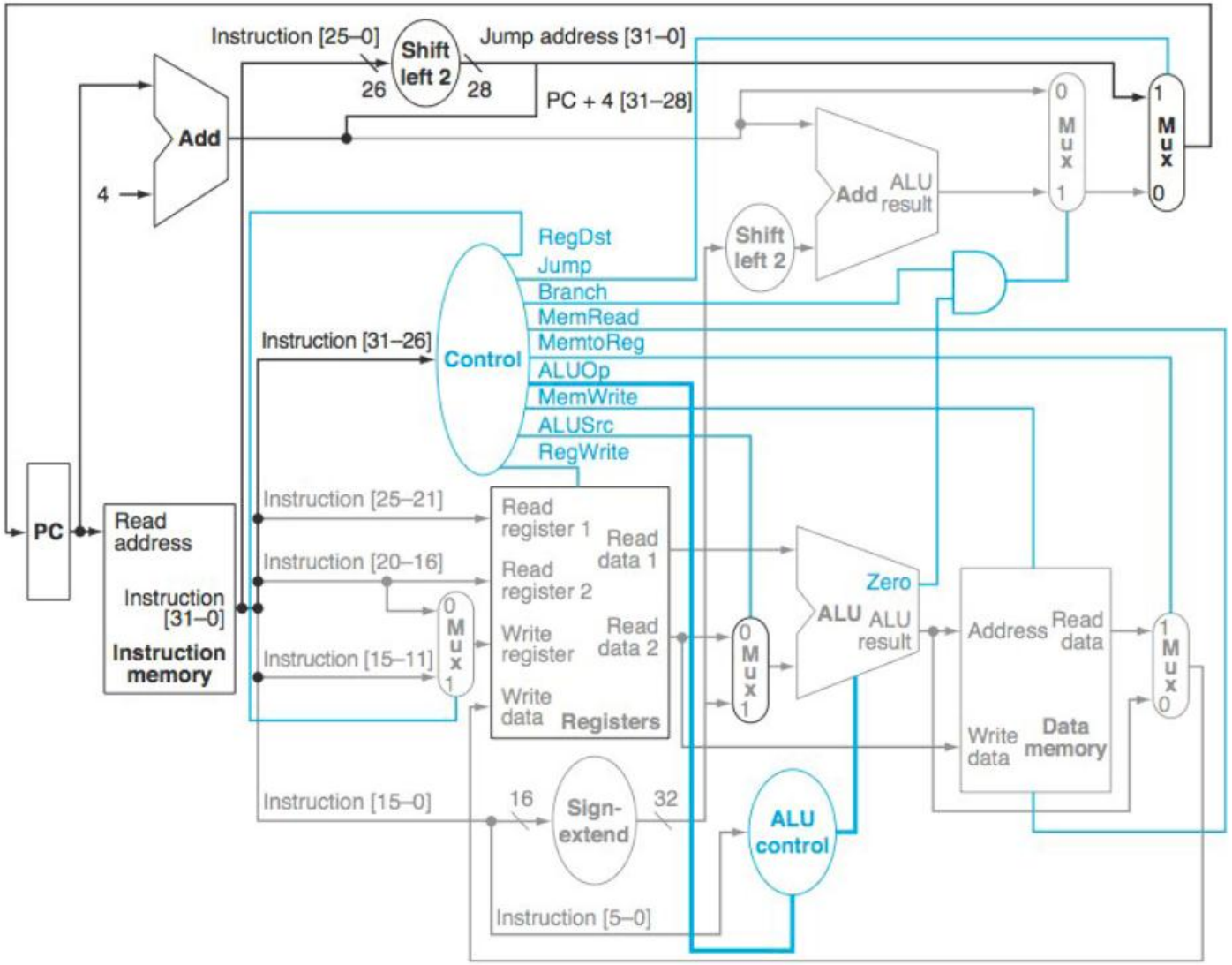
④ 条件同第①问，这次检测控制信号MemRead是否存在上表中的缺陷？

⑤ 条件同第①问，这次检测控制信号Jump是否存在上表中的缺陷？

⑥ 使用第①问中描述的测试方案，可以一次对几个不同的信号进行测试，但一般来说不可能同时测试到所有信号。试着设计一系列方案对所有多选器输出的上表中的缺陷进行测试(五个多选器输出的每一位都要测试到)。尽量使用较少的测试方案。

五、题图

❖图4


Instruction [25–0]
Shift left 2
Jump address [31–0]
26
28
PC + 4 [31–28]
Add
4
0
M
u
x
1
1
M
u
x
0
Add
ALU result
Shift left 2
RegDst
Jump
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Instruction [31–26]
Control
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
Zero
ALU
ALU result
Address
Read data
Write data
Data memory
Instruction [15–0]
16
Sign-extend
32
ALU control
Instruction [5–0]

五、解答4

	缺陷
a.	固定为1
b.	如果指令的第31~26位全为0，则固定为0，否则无缺陷

④ **条件同第①问，这次检测控制信号MemRead是否存在上表中的缺陷？**

- **a.如果MemRead存在固定为1缺陷，那么在每一条指令执行的时候都会读取数据存储器，然而对于非load类指令，从存储器读出的数据会被多选器抛弃，这导致我们无法设计出检测该缺陷的方法，因为即使存在该缺陷，处理器也是正常工作的。**
- **b.为了测试该缺陷我们需要一个操作码为0同时MemRead信号为1的指令，但是操作码为0的指令都是ALU的R型指令，不是load指令，因而它们的MemRead信号都为0，因此，即使存在该缺陷，处理器也是正常工作的，因而我们无法设计出检测该缺陷的方法。**

五、解答5

	缺陷
a.	固定为1
b.	如果指令的第31~26位全为0，则固定为0，否则无缺陷

⑤ 条件同第①问，这次检测控制信号Jump是否存在上表中的缺陷？

- **a.如果Jump信号存在固定为1缺陷，那么每一条指令执行时都会按照执行J指令时的方法更新PC的值，要检测该缺陷，可以将一条非跳转指令(例如ADD $1,$0,$0)放在指令存储器中第一条指令的位置，该指令执行之后PC的值应当是0x00000004，若PC的值变为0x00002080则说明Jump信号存在固定为1缺陷。**
- **b.为了测试该缺陷我们需要一个操作码为0同时Jump信号为1的指令，然而跳转指令的操作码都不是0，因此我们无法设计出检测该缺陷的方法，此时处理器是正常工作的。**

五、解答6

	缺陷
a.	固定为1
b.	如果指令的第31~26位全为0，则固定为0，否则无缺陷

⑥ **使用第①问中描述的测试方案，可以一次对几个不同的信号进行测试，但一般来说不可能同时测试到所有信号。试着设计一系列方案对所有多选器输出的上表中的缺陷进行测试(五个多选器输出的每一位都要测试到)。尽量使用较少的测试方案。**

- **涉及到5个控制信号：RegDst、Jump、Branch、MemtoReg和ALUSrc，分别对每一个信号设计测试方案进行测试即可，但是注意有些信号的缺陷a或者缺陷b是无法检测出来的。**
- **a.RegDst信号的固定为1缺陷可用lw指令测试，首先可以将数据存储器全写为1，执行lw $1,0x1000($0)，如果载入的字出现在了$2中而不是$1中说明RegDst存在固定为1缺陷。**
- **Jump的固定为1缺陷测试方法同(5)a。**

五、解答6

	缺陷
a.	固定为1
b.	如果指令的第31~26位全为0，则固定为0，否则无缺陷

⑥ **使用第①问中描述的测试方案，可以一次对几个不同的信号进行测试，但一般来说不可能同时测试到所有信号。试着设计一系列方案对所有多选器输出的上表中的缺陷进行测试(五个多选器输出的每一位都要测试到)。尽量使用较少的测试方案。**

- **Branch的固定为1缺陷可以用非分支指令测试，也可以用测试Jump时同样的方法，ADD $1,$0,$0指令会使ALU的Zero输出为1，导致分支条件“满足”，若执行之后PC的值变为0x00002084而不是0x00000004则说明Branch存在固定为1缺陷。**
- **MemtoReg信号的固定为1缺陷无法准确检测，原因参见(2)b。**
- **ALUSrc的固定为1缺陷也可以用ADD $1,$0,$0指令测试，若执行之后$1的值为0x00000820而不是0，则说明存在缺陷。**

五、解答6

	缺陷
a.	固定为1
b.	如果指令的第31~26位全为0，则固定为0，否则无缺陷

⑥ **使用第①问中描述的测试方案，可以一次对几个不同的信号进行测试，但一般来说不可能同时测试到所有信号。试着设计一系列方案对所有多选器输出的上表中的缺陷进行测试(五个多选器输出的每一位都要测试到)。尽量使用较少的测试方案。**

- **涉及到5个控制信号：RegDst、Jump、Branch、MemtoReg和ALUSrc，分别对每一个信号设计测试方案进行测试即可，但是注意有些信号的缺陷a或者缺陷b是无法检测出来的。**
- **b.操作码为0的指令都是ALU的R型指令，对于这些指令Jump、Branch、MemtoReg以及ALUSrc本就应该为0，所以这些信号的缺陷无法检测。**
- **RegDst的缺陷可以用ADD $1,$2,$3来测试，若执行完之后$2+$3的结果出现在了$3而不是$1中，说明存在缺陷。**

六、题目

- 本题讨论特定指令在单周期数据通路中的操作。根据下表中的MIPS指令分别回答下列问题(参考MIPS汇编作业中的附录)。

	指令
a.	lw $1,40($6)
b.	Label: bne $1,$2,Label

① 该指令字的值是多少？

② 提供给寄存器堆“Read register 1”端口的寄存器号是多少？该寄存器真的被读了吗？对于“Read register 2”呢？

③ 提供给寄存器堆“Write register”端口的寄存器号是多少？该寄存器真的被写了吗？

六、解答1

	指令
a.	lw $1,40($6)
b.	Label: bne $1,$2,Label

① 该指令字的值是多少？

	二进制	十六进制
a.	100011 00110 00001 0000000000101000	8CC10028
b.	000101 00001 00010 1111111111111111	1422FFFF

六、解答2

	指令
a.	lw $1,40($6)
b.	Label: bne $1,$2,Label

② **提供给寄存器堆“Read register 1”端口的寄存器号是多少？该寄存器真的被读了吗？对于“Read register 2”呢？**

	Read register 1	是否被读？	**Read register 2**	是否被读？
a.	$6(00110_2)$	是	$1(00001_2)$	是
b.	$1(00001_2)$	是	$2(00010_2)$	是

寄存器堆只要提供了寄存器号，相应的寄存器就会被读取，但是读出的数据不一定会用到

六、解答3

	指令
a.	lw $1,40($6)
b.	Label: bne $1,$2,Label

③ **提供给寄存器堆“Write register”端口的寄存器号是多少？该寄存器真的被写了吗？**

	Write register	是否被写？
a.	1(00001_2)	是
b.	2(00010_2)或者31(11111_2)(因为RegDst未知)	否

六、题目

- 不同的指令需要设置数据通路上不同的控制信号。根据下表的两种控制信号情况分别回答下列问题(参考图4)。

	控制信号1	控制信号2
a.	RegDst	MemRead
b.	RegWrite	MemRead

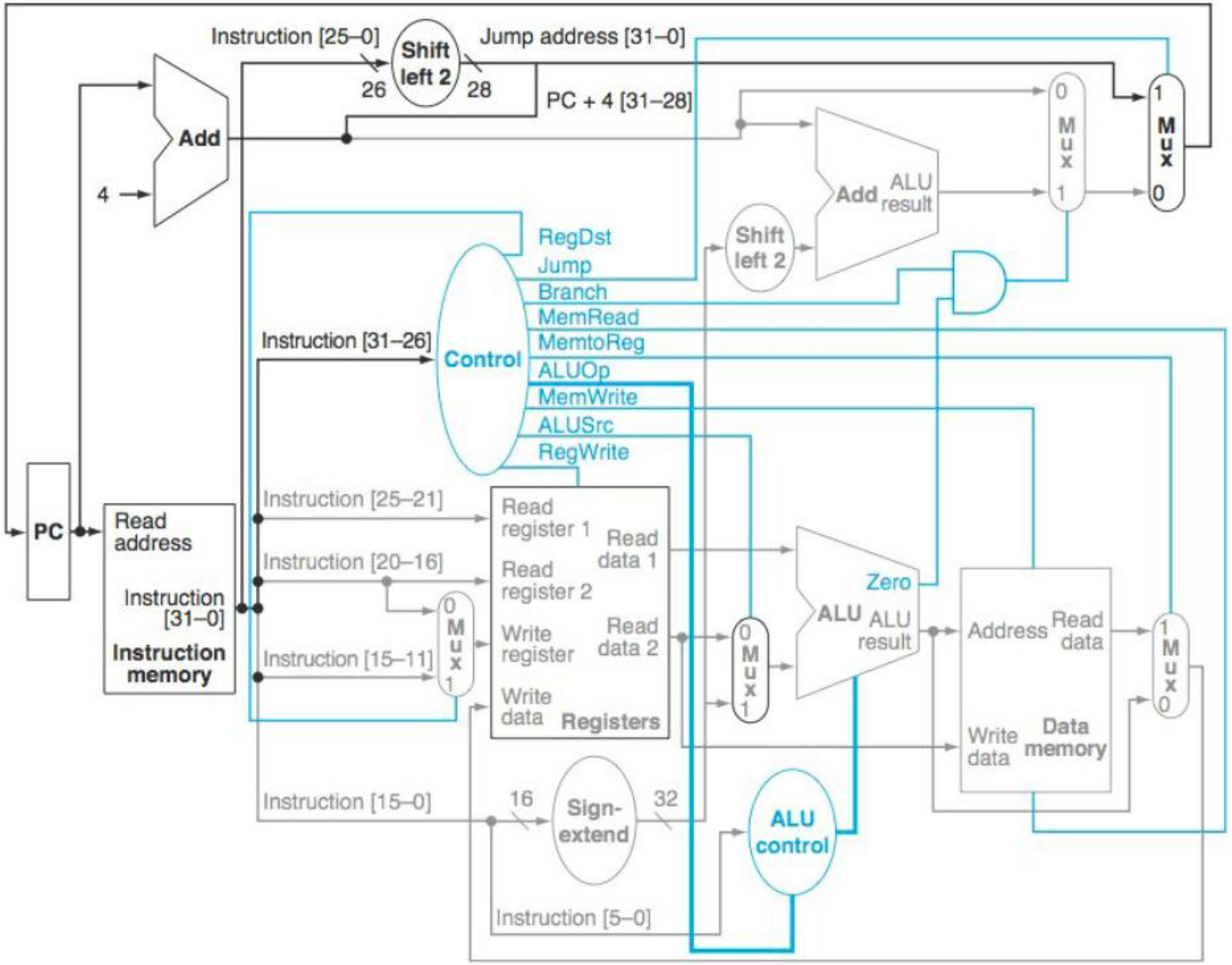
④ 对于前面的指令而言，这两个控制信号的值应该是多少？

	指令
a.	lw $1,40($6)
b.	Label: bne $1,$2,Label

⑤ 对图4中的数据通路而言，写出控制单元中实现这两个信号的逻辑表达式。假设我们仅需支持lw、sw、beq、add和j(jump)指令。

六、题图

❖图4


Instruction [25–0]
Shift left 2
Jump address [31–0]
26
28
PC + 4 [31–28]
Add
4
0
M
u
x
1
1
M
u
x
0
Add
ALU result
Shift left 2
RegDst
Jump
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Instruction [31–26]
Control
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Instruction [5–0]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
16
Sign-extend
32
ALU control
Zero
ALU
ALU result
Address
Read data
Write data
Data memory

六、解答4

	控制信号1	控制信号2
a.	RegDst	MemRead
b.	RegWrite	MemRead

④ 对于前面的指令而言，这两个控制信号的值应该是多少？

	指令
a.	lw $1,40($6)
b.	Label: bne $1,$2,Label

	控制信号1	控制信号2
a.	RegDst = 0	MemRead = 1
b.	RegWrite = 0	MemRead = 0

六、解答5

	控制信号1	控制信号2
a.	RegDst	MemRead
b.	RegWrite	MemRead

⑤ 对图4中的数据通路而言，写出控制单元中实现这两个信号的逻辑表达式。假设我们仅需支持lw、sw、beq、add和j(jump)指令。

- 假设OP5~OP0代表指令中的[31:26]各个位，则有：
- a. $\mathbf{RegDst} = \overline{\mathbf{OP5}}\ \overline{\mathbf{OP4}}\ \overline{\mathbf{OP3}}\ \overline{\mathbf{OP2}}\ \overline{\mathbf{OP1}}\ \overline{\mathbf{OP0}}$

  $\mathbf{MemRead} = \mathbf{OP5}\ \overline{\mathbf{OP4}}\ \overline{\mathbf{OP3}}\ \overline{\mathbf{OP2}}\ \mathbf{OP1}\ \mathbf{OP0}$
- b. $\mathbf{RegWrite} = \mathbf{OP5}\ \overline{\mathbf{OP4}}\ \overline{\mathbf{OP3}}\ \overline{\mathbf{OP2}}\ \mathbf{OP1}\ \mathbf{OP0} + \overline{\mathbf{OP5}}\ \overline{\mathbf{OP4}}\ \overline{\mathbf{OP3}}\ \overline{\mathbf{OP2}}\ \overline{\mathbf{OP1}}\ \overline{\mathbf{OP0}}$

  $\mathbf{MemRead} = \mathbf{OP5}\ \overline{\mathbf{OP4}}\ \overline{\mathbf{OP3}}\ \overline{\mathbf{OP2}}\ \mathbf{OP1}\ \mathbf{OP0}$

六、解答5

	控制信号1	控制信号2
a.	RegDst	MemRead
b.	RegWrite	MemRead

- **解析：**
- **首先写出这几条指令的Opcode：lw：100011，sw：101011，beq：000100，add：000000，j：000010，并列出相应的控制信号的值如下表，操作码为1的位取原变量，操作码为0的位取反变量，根据最小项推导法可以写出相应的逻辑表达式。**

		lw	sw	beq	add	j
操作码	OP5	1	1	0	0	0
	OP4	0	0	0	0	0
	OP3	0	1	0	0	0
	OP2	0	0	1	0	0
	OP1	1	1	0	0	1
	OP0	1	1	0	0	0
a.	RegDst	0	X	X	1	X
	MemRead	1	0	0	0	0
b.	RegWrite	1	0	0	1	0
	MemRead	1	0	0	0	0

七、题目

- 本题讨论处理器时钟周期与控制单元设计之间的相互影响。根据下表的两种数据通路单元延迟情况分别回答下列问题。

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	符号扩展	左移两位	ALU控制
a.	400ps	100ps	30ps	120ps	200ps	350ps	20ps	0ps	50ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps	55ps

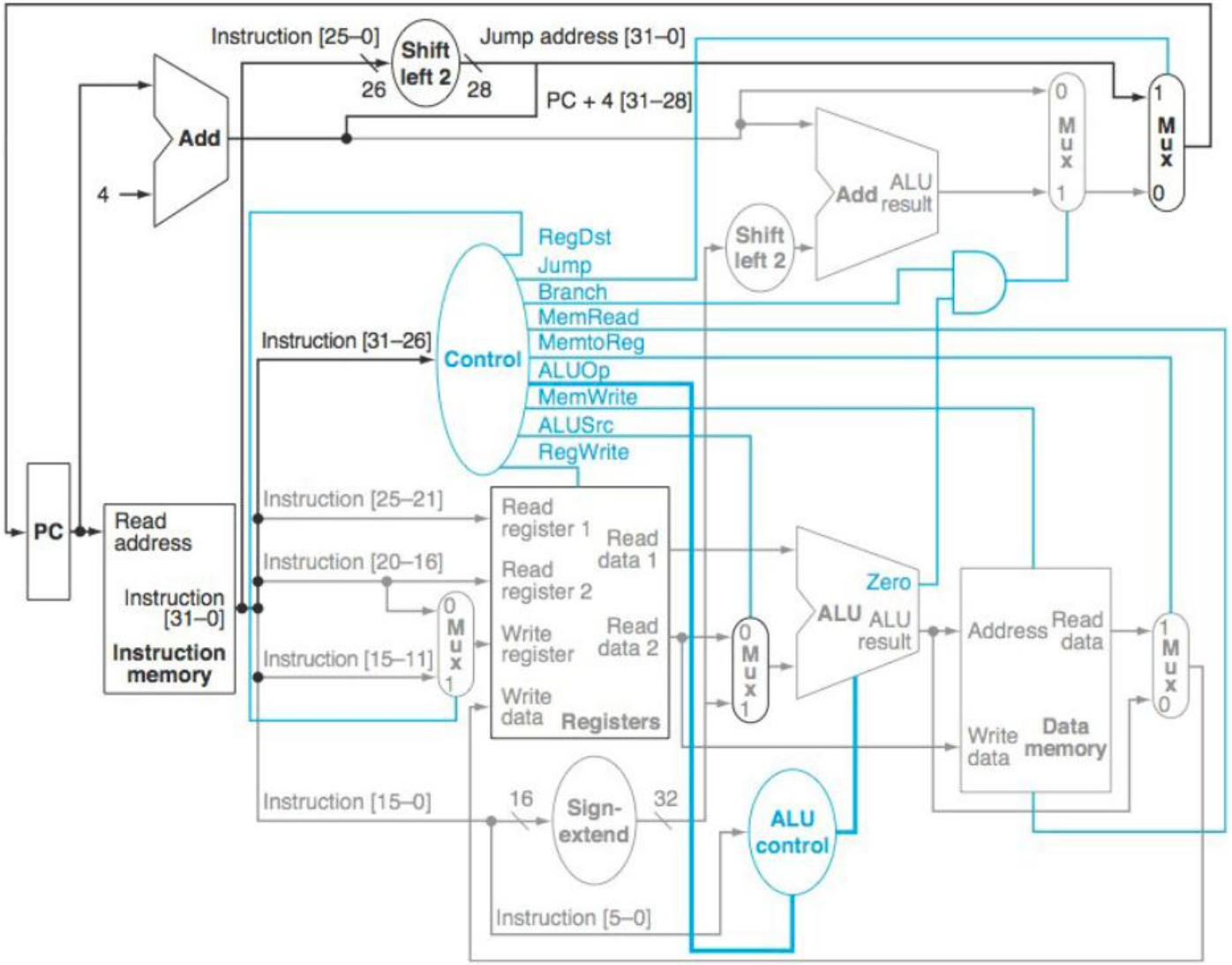
① 为了避免增加图4中数据通路的关键路径长度，留给控制单元产生MemWrite信号的时间有多少？

② 图4中哪个控制信号最不关键，控制单元需要在多长时间内产生该信号以避免其成为关键路径？

③ 图4中哪个控制信号最关键，控制单元需要在多长时间内产生该信号以避免其成为关键路径？

七、题图

❖图4


Instruction [25–0]
Shift left 2
Jump address [31–0]
26
28
PC + 4 [31–28]
Add
4
M u x
0
1
Add
ALU result
Shift left 2
RegDst
Jump
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Instruction [31–26]
Control
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Instruction [5–0]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
16
Sign-extend
32
ALU
Zero
ALU result
ALU control
Address
Read data
Write data
Data memory

七、解答1

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	符号扩展	左移两位	ALU控制
a.	400ps	100ps	30ps	120ps	200ps	350ps	20ps	0ps	50ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps	55ps

① 为了避免增加图4中数据通路的关键路径长度，留给控制单元产生MemWrite信号的时间有多少？

- a.关键路径延迟为400ps + 200ps + 30ps + 120ps + 350ps + 30ps + 200ps = 1330ps，留给控制单元产生MemWrite信号的最长时间为1330ps – 400ps – 350ps = 580ps
- b.关键路径延迟为500ps + 220ps + 100ps + 180ps + 1000ps + 100ps + 220ps = 2320ps，留给控制单元产生MemWrite信号的最长时间为2320ps – 500ps – 1000ps = 820ps
- 解析：假设控制单元的延迟为0，则lw具有最长的关键路径，于是得到关键路径I-Mem、Regs(Read)、Mux、ALU、D-Mem(Read)、Mux、Regs(Write)。控制单元在读取完指令存储器之后才可以产生MemWrite控制信号，并且必须在时钟周期结束之前产生MemWrite信号，但是，由于MemWrite信号是数据存储器的写使能信号，因此在产生该信号之后还应当至少留出写存储器的时间D-Mem(Write)。

七、解答2

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	符号扩展	左移两位	ALU控制
a.	400ps	100ps	30ps	120ps	200ps	350ps	20ps	0ps	50ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps	55ps

② **图4中哪个控制信号最不关键，控制单元需要在多长时间内产生该信号以避免其成为关键路径？**

- **a.Jump信号具有最长的松弛时间，为1330ps – 400ps – 30ps = 900ps**
- **b.Jump信号具有最长的松弛时间，为2320ps – 500ps – 100ps = 1720ps**
- **解析：所有的控制信号都必须在指令读取之后生成，同时一个信号最晚必须在时钟周期结束之前到来，对于MemWrite、RegWrite和Jump信号，由于它们都只在时钟周期的最后才会用到，因而相比于其它控制信号会拥有更长的松弛时间，由于两种情况下均是数据存储器的延迟>寄存器堆>多选器，因而Jump具有最长的松弛时间。**
- **这个题目里面没有考虑PC的延迟。**

七、解答3

	指令存储器	加法器	多选器	ALU	寄存器堆	数据存储器	符号扩展	左移两位	ALU控制
a.	400ps	100ps	30ps	120ps	200ps	350ps	20ps	0ps	50ps
b.	500ps	150ps	100ps	180ps	220ps	1000ps	90ps	20ps	55ps

③ 图4中哪个控制信号最关键，控制单元需要在多长时间内产生该信号以避免其成为关键路径？

	最关键信号	产生该信号可用的时间
a.	ALUOp (50ps > 30ps)	200ps + 30ps – 50ps = 180ps
b.	ALUSrc (100ps > 55ps)	220ps

- **解析：为了不影响关键路径，控制信号必须在数据到达之前产生以便不影响数据的通过，最早出现在关键路径上的控制信号是ALUOp和ALUSrc，ALUSrc的松弛时间为Regs(Read)，ALUOp的松弛时间为Regs(Read) + Mux – ALU Ctrl，拥有较小松弛时间的信号更为关键，可见二者对于ALU计算的影响取决于ALU Ctrl与Mux的延迟大小。**

七、题目

- 假设控制单元产生控制信号的时间如下表所示，试根据表中的两种情况回答下列问题(各部件的延迟与前面相同)。

	RegDst	Jump	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite
a.	720ps	730ps	600ps	400ps	700ps	200ps	710ps	200ps	800ps
b.	1600ps	1600ps	1400ps	500ps	1400ps	400ps	1500ps	400ps	1700ps

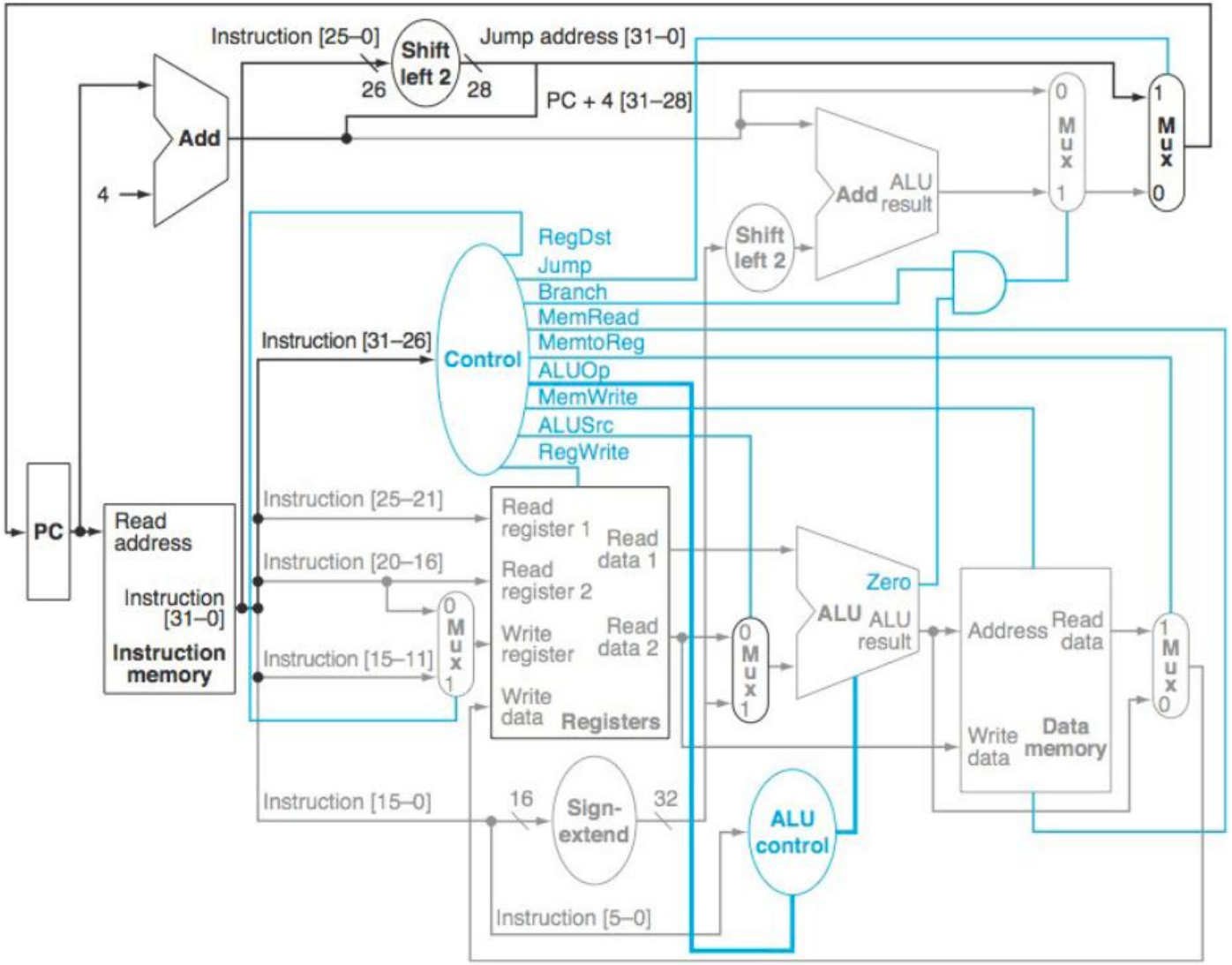
④ 处理器的时钟周期为多少？

⑤ 如果你可以加速控制信号的产生，但加快一个控制信号5ps的代价是处理器成本增加1元。那么为了最大化性能你会加速哪些控制信号？这种性能改进的最小代价是多少？

⑥ 如果一个处理器的成本已经很高，那么我们需要在维持处理器性能的同时降低其成本，而不是像第⑤问中所作的那样为提高它的性能而买单。如果你可以使用更慢的逻辑来实现对信号的控制，并且单个控制信号每减慢5ps，处理其成本就可以节省1元，那么在保持处理器性能的同时，你会减慢哪些控制信号，并且减慢多少来降低成本？

七、题图

❖图4


Instruction [25–0]
Shift left 2
Jump address [31–0]
26
28
PC + 4 [31–28]
Add
4
M u x
0
1
Add
ALU result
Shift left 2
RegDst
Jump
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Instruction [31–26]
Control
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Instruction [5–0]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
16
Sign-extend
32
ALU
Zero
ALU result
ALU control
Address
Read data
Write data
Data memory

七、解答4

	RegDst	Jump	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite
a.	720ps	730ps	600ps	400ps	700ps	200ps	710ps	200ps	800ps
b.	1600ps	1600ps	1400ps	500ps	1400ps	400ps	1500ps	400ps	1700ps

④ 处理器的时钟周期为多少？

	对时钟周期影响最大的控制信号	理想的时钟周期(由第(1)问得来)	实际的时钟周期
a.	MemWrite (+130ps)	1330ps	1460ps
b.	MemWrite (+680ps)	2320ps	3000ps

- 解析：为了便于后面的计算，我们首先计算各个控制信号的松弛时间，如下表所示：

	RegDst	Jump	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite
a.	700ps	900ps	870ps	350ps	700ps	180ps	580ps	200ps	730ps
b.	1500ps	1720ps	1620ps	500ps	1500ps	265ps	820ps	220ps	1600ps

- 若实际产生信号的时间小于松弛时间，则该信号的产生不会影响时钟周期，反之，该信号会影响时钟周期，并且显然是会使时钟周期变大，由此可以计算出新的时钟周期(下表中的数据为实际时间减去松弛时间)。

	RegDst	Jump	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite
a.	20ps	-170ps	-270ps	50ps	0ps	20ps	130ps	0ps	70ps
b.	100ps	-120ps	-220ps	0ps	-100ps	135ps	680ps	180ps	100ps

七、解答5

	RegDst	Jump	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite
a.	720ps	730ps	600ps	400ps	700ps	200ps	710ps	200ps	800ps
b.	1600ps	1600ps	1400ps	500ps	1400ps	400ps	1500ps	400ps	1700ps

⑤ **如果你可以加速控制信号的产生，但加快一个控制信号5ps的代价是处理器成本增加1元。那么为了最大化性能你会加速哪些控制信号？这种性能改进的最小代价是多少？**

	对时钟周期有影响的信号	代价
a.	RegDst (+20ps) MemRead (+50ps) ALUOp (+20ps) MemWrite (+130ps) RegWrite (+70ps)	290/5=58
b.	RegDst (+100ps) ALUOp (+135ps) MemWrite (+680ps) ALUSrc (+180ps) RegWrite (+100ps)	1195/5=239

- **解析：应当只加速影响了时钟周期的控制信号，目标是将这些控制信号对时钟周期的影响至少变为0。**

七、解答6

	RegDst	Jump	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite
a.	720ps	730ps	600ps	400ps	700ps	200ps	710ps	200ps	800ps
b.	1600ps	1600ps	1400ps	500ps	1400ps	400ps	1500ps	400ps	1700ps

⑥ 如果一个处理器的成本已经很高，那么我们需要在维持处理器性能的同时降低其成本，而不是像第⑤问中所作的那样为提高它的性能而买单。如果你可以使用更慢的逻辑来实现对信号的控制，并且单个控制信号每减慢5ps，处理其成本就可以节省1元，那么在保持处理器性能的同时，你会减慢哪些控制信号，并且减慢多少来降低成本？

	RegDst	Jump	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite
a.	20ps	-170ps	-270ps	50ps	0ps	20ps	130ps	0ps	70ps
b.	100ps	-120ps	-220ps	0ps	-100ps	135ps	680ps	180ps	100ps

- 上表中具有最大正值的信号决定了周期，则在保证性能的前提之下，可以将所有的信号都减慢到具有跟该信号相同的值，于是最小化成本的方法是按照下表减慢响应信号的速度：

	RegDst	Jump	Branch	MemRead	MemtoReg	ALUOp	MemWrite	ALUSrc	RegWrite
a.	110ps	300ps	400ps	80ps	130ps	110ps	0ps	130ps	60ps
b.	580ps	800ps	900ps	680ps	780ps	545ps	0ps	500ps	580ps

八、题目

- 本题讨论单周期数据通路中指令的执行细节。根据表中的两种指令字情况回答下列问题(参考MIPS汇编作业中的附录)。

	指令字
a.	10001100 01000011 00000000 00010000
b.	00010000 00100011 00000000 00001100

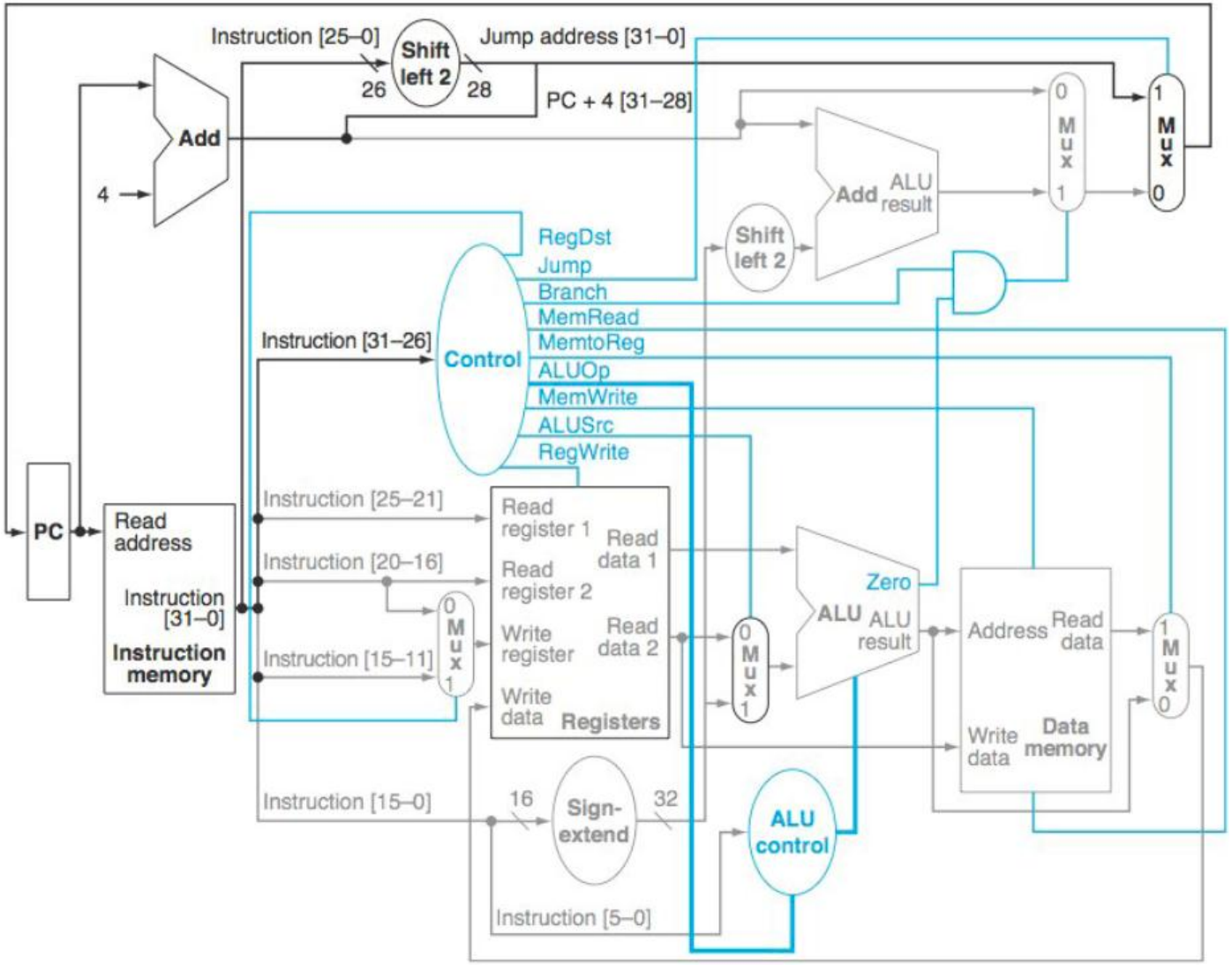
① 对该指令字而言，符号扩展单元和图4左上角的左移两位单元的输出是什么？

② 对该指令字而言，ALU控制单元的输入是什么？

③ 该指令执行后的新PC值是什么？在图4中决定该新PC值的数据通路是什么？

八、题图

❖图4


Instruction [25–0]
Shift left 2
Jump address [31–0]
26
28
PC + 4 [31–28]
Add
4
0 M u x 1
1 M u x 0
Add
ALU result
Shift left 2
RegDst
Jump
Branch
MemRead
MemtoReg
ALUOp
MemWrite
ALUSrc
RegWrite
Instruction [31–26]
Control
PC
Read address
Instruction [31–0]
Instruction memory
Instruction [25–21]
Instruction [20–16]
Instruction [15–11]
Instruction [15–0]
Instruction [5–0]
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
16
Sign-extend
32
ALU control
Zero
ALU
ALU result
Address
Read data
Write data
Data memory

八、解答1

	指令字
a.	10001100 01000011 00000000 00010000
b.	00010000 00100011 00000000 00001100

① **对该指令字而言，符号扩展单元和图4左上角的左移两位单元的输出是什么？**

	符号扩展单元	左移两位单元
a.	00000000 00000000 00000000 00010000	0001 00001100 00000000 01000000
b.	00000000 00000000 00000000 00001100	0000 10001100 00000000 00110000

- **解析：注意符号扩展单元要按符号位进行扩展，同时左移两位单元的输出应为28位。**

八、解答2

	指令字
a.	10001100 01000011 00000000 00010000
b.	00010000 00100011 00000000 00001100

② 对该指令字而言，ALU控制单元的输入是什么？

	ALUOp[1:0]	Instruction[5:0]
a.	00	010000
b.	01	001100

- 解析：可以根据指令字写出指令，如下表所示，在lw指令中，ALU进行加法运算，在beq指令中，ALU执行减法运算。

	指令
a.	lw Rt(00011), Offset(00000000 00010000)(Rs(00010))
b.	beq Rs(00001), Rt(00011), Label(00000000 00001100)

八、解答3

	指令字
a.	10001100 01000011 00000000 00010000
b.	00010000 00100011 00000000 00001100

③ 该指令执行后的新PC值是什么？在图4中决定该新PC值的数据通路是什么？

	新PC值	数据通路
a.	PC + 4	PC、Add(PC + 4)、Mux(branch)、Mux(jump)、PC
b.	如果$1与$3不相等，为PC + 4 否则，为PC + 4 + 4 × 12	PC、Add(PC + 4)、Mux(branch)、Mux(jump)、PC 或者PC、Add(PC + 4)、Add(偏移)、Mux(branch)、Mux(jump)、PC

八、题目

❖ 下列问题假设数据存储器中的值是全零并且寄存器堆中的初值如下表所示，根据表中的两种情况回答下列问题(指令字与前面相同)。

	$0	$1	$2	$3	$4	$5	$6	$8	$12	$31
a.	0	1	2	3	-4	5	6	8	1	-32
b.	0	-16	-2	-3	4	-10	-6	-1	8	-4

④ 对给定的指令字和寄存器堆初值，给出每个多选器数据输出的值。

⑤ 对给定的指令字和寄存器堆初值，给出ALU和两个加法器输入数据的值。

⑥ 对给定的指令字和寄存器堆初值，给出寄存器堆所有输入信号的值。

	指令字
a.	10001100 01000011 00000000 00010000
b.	00010000 00100011 00000000 00001100

八、解答4

	$0	$1	$2	$3	$4	$5	$6	$8	$12	$31
a.	0	1	2	3	-4	5	6	8	1	-32
b.	0	-16	-2	-3	4	-10	-6	-1	8	-4

④ **对给定的指令字和寄存器堆初值，给出每个多选器数据输出的值。**

	WrReg Mux	ALU Mux	Mem/ALU Mux	Branch Mux	Jump Mux
a.	3	16	0	PC + 4	PC + 4
b.	3或者0 (RegDst未知)	-3	X	PC + 4	PC + 4

- **解析：对于b，由于Reg[Rs]!=Reg[Rt]，分支条件不满足，于是不进行跳转，仍有PC=PC+4。**

	指令
a.	lw Rt(00011), Offset(00000000 00010000)(Rs(00010))
b.	beq Rs(00001), Rt(00011), Label(00000000 00001100)

八、解答5

	$0	$1	$2	$3	$4	$5	$6	$8	$12	$31
a.	0	1	2	3	-4	5	6	8	1	-32
b.	0	-16	-2	-3	4	-10	-6	-1	8	-4

⑤ **对给定的指令字和寄存器堆初值，给出ALU和两个加法器输入数据的值。**

	ALU	Add (PC + 4)	Add (Branch)
a.	2和16	PC和4	PC + 4和16×4
b.	-16和-3	PC和4	PC + 4和12×4

八、解答6

	$0	$1	$2	$3	$4	$5	$6	$8	$12	$31
a.	0	1	2	3	-4	5	6	8	1	-32
b.	0	-16	-2	-3	4	-10	-6	-1	8	-4

⑥ 对给定的指令字和寄存器堆初值，给出寄存器堆所有输入信号的值。

	Read Register 1	Read Register 2	Write Register	Write Data	RegWrite
a.	2	3	3	0	1
b.	1	3	X(3或者0)	X	0