计算机组成原理　第六周作业　3 月 20 日 周五

PB18151866 龚小航

4.1.

在基本的单周期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]

4.1.1 [5] < 4.1 > 对上述指令而言，图 4-2 中的控制单元将产生哪些控制信号?

4.1.2 [5] < 4.1 > 对上述指令而言，将用到哪些功能单元?

4.1.3 [10] < 4.1 > 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?

不同单元有不同的延迟时间。在图 4-2 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）上各单元的延迟时间决定了该指令的最小延迟。假设各单元的延迟时间如下表所示，回答下列 3 个问题。

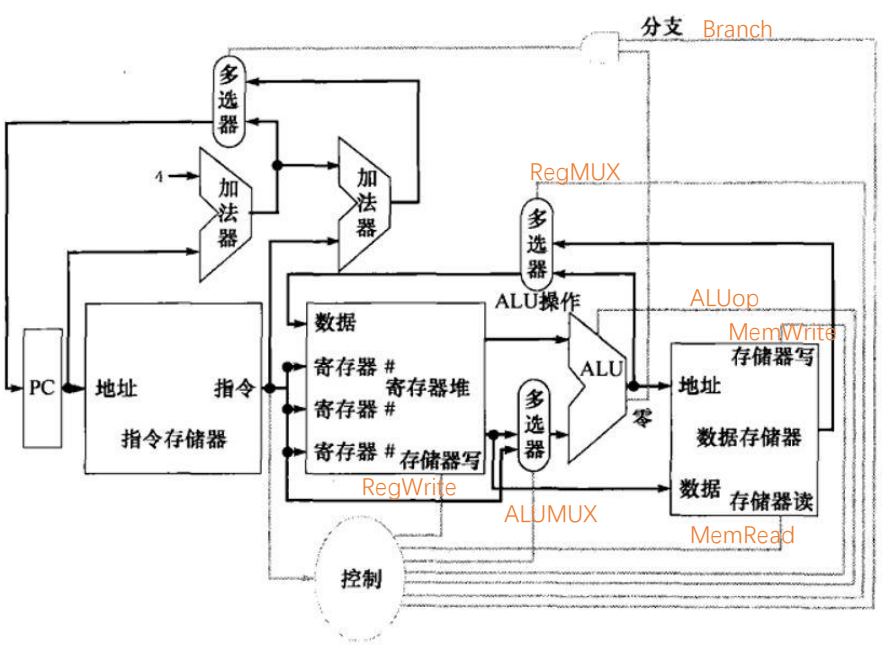
	指令存储器	加	多选器	ALU	寄存器堆	数据存储器	控制
a.	400 ps	100 ps	30 ps	120 ps	200 ps	350 ps	100 ps
b.	500 ps	150 ps	100 ps	180 ps	220 ps	1000 ps	65 ps

4.1.4 [5] < 4.1 >对一条 MIPS 的与指令(AND)而言，关键路径是什么?

4.1.5 [5] < 4.1 >一条 MIPS 的装载指令(LD)而言，关键路径是什么?

4.1.6 [10] < 4.1 >对一条 MIPS 的相等则分支指令(BEQ)而言，关键路径是什么?

图 4-2:



解：(1) 将会产生的信号列于下表：各信号已在图中以橙色标出

	ALUop	ALUMUX	RegWrite	MemWrite	RegMUX	MemRead	Branch
$a.$	0010(add)	0(寄存器)	1	0	1(ALU)	0	0
$b.$	0010(add)	1(立即数)	1	0	0(Mem)	1	0

(2) 对于指令$a.$，使用了上图中数据存储器和分支部分以外的所有功能单元。

对于指令$b.$，使用了上图中分支部分以外的所有功能单元。（还有一个寄存器读端口未用）

(3) 对于指令$a.$，分支部分的加法器产生了某个值，但没有被用到，在多选器处不被选中；
数据存储器不产生任何输出。

对于指令$b.$，分支部分的加法器产生了某个值，但没有被用到，在多选器处不被选中；且寄存器组的第二个读取端口输出了某个寄存器的值，但是没有被用到，未被 ALUMUX 选中。
所有的功能单元均产生了输出。

(4) 对于一条 AND 指令，关键路径：（$a.b.$两种情况相同）

取指→读取寄存器堆→经过 ALUMUX→ALU 计算→经过 RegMUX→存回寄存器堆

(5) 对于一条 LD 指令，关键路径：（$a.b.$两种情况相同）

取指→读取寄存器堆→经过 ALUMUX→ALU 计算→写数据存储器→经过 RegMUX

(6) 对于一条 BEQ 指令，关键路径：（$a.b.$两种情况相同）

取指→读取寄存器堆→经过 ALUMUX→ALU 计算→经过分支多选器

4.9.

本习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。

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

4.9.1 [10] < 4.4 > 指令字的值是多少?

4.9.2 [10] < 4.4 > 提供给寄存器堆“读寄存器 1" 端口的寄存器号是多少? 该寄存器真的被读了吗? 对于“读寄存器 2"呢?

4.9.3 [10] < 4.4 > 供给寄存器堆“写寄存器”端口的寄存器号是多少? 该寄存器真的被写了吗?

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 4-24)。

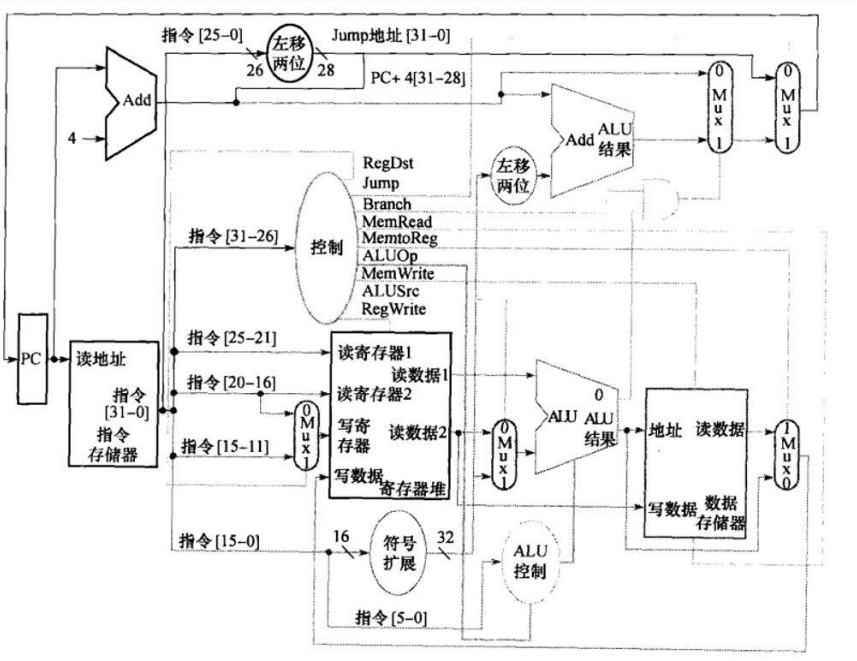
	控制信号 1	控制信号 2
a.	RegDst	MemRead
b.	RegWrite	MemRead

4.9.4 [20] < 4.4 >对该指令而言，这两个控制信号的值应该是多少?

4.9.5 [20] < 4.4 >对图 4-24 中的数据通路而言，画出控制单元中实现第一个信号的部分电路图。假设我们仅需支持 lw、SW、beq、add 和 j (jump)指令。

4.9.6 [20] < 4.4 >对图 4-24 中的数据通路而言，画出控制单元中实现第二个信号的部分电路图

图 4-24：



解：(1) 直接写出这两条指令对应的机器码:

a. lw $1, 40($6) ⟹ 100011 00110 00001 0000000000101000 = 0x 8CC1 0028

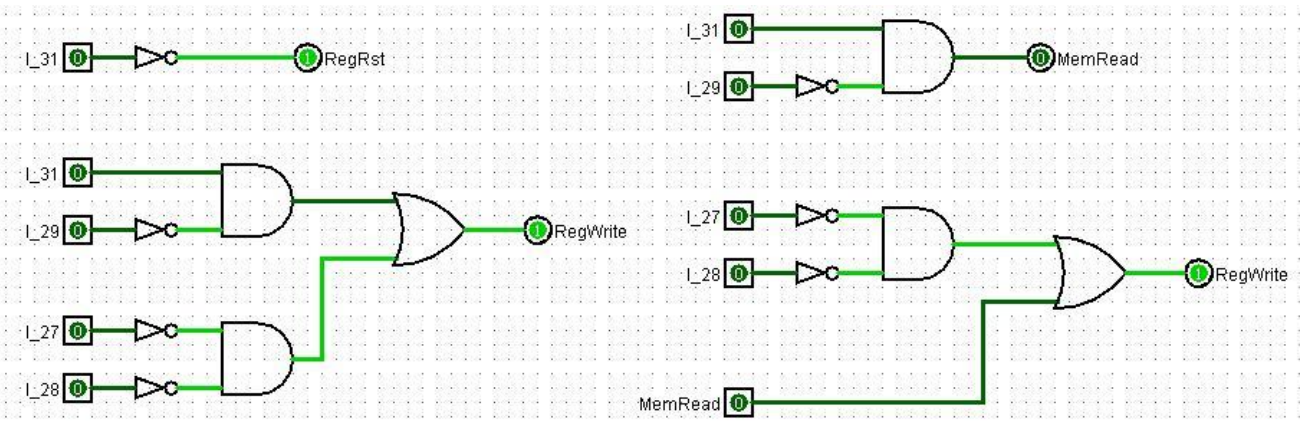
b. bne $1,$2,Label ⟹ 000101 00001 00010 1111111111111111 = 0x 1422 FFFF

(2) 对于指令*a.*，提供给“读寄存器 1“ 的寄存器号为 00110，该寄存器真的被读了。
提供给“读寄存器 2“ 的寄存器号为 00001，该寄存器真的被读了。
对于指令*b.*，提供给“读寄存器 1“ 的寄存器号为 00001，该寄存器真的被读了。
提供给“读寄存器 2“ 的寄存器号为 00010，该寄存器真的被读了。

(3) 从图 4-24 中可以看出，写寄存器端口连接了 MUX，由 RegDst 控制。
对于指令*a.*“写寄存器“得到的寄存器号为 00001，该寄存器真的被写了。
对于指令*b.*“写寄存器“得到的寄存器号为 00010或11111（RegDst未知），该寄存器没有被写。

(4) 根据图 4-24.可以得到:
对于指令*a.* RegDst = 0， MemRead = 1
对于指令*b.* RegWrite = 0， MemRead = 0

(5) *and* (6) 先写出控制信号的逻辑关系，再画简单电路图:
对指令的[31:26]位分析，记每一位分别为I_{26}~I_{31}.分析真值，可得:
a. RegRst = $\overline{I_{31}}$， *b.* RegWrite = $\overline{I_{28}}\,\overline{I_{27}} + I_{31}\overline{I_{29}}$
a. MemRead = $I_{31}\overline{I_{29}}$， *b.* RegWrite = $\overline{I_{28}}\,\overline{I_{27}}$ + MemRead



附：分析 MIPS 三种类型指令的多周期设计方案中每个周期所用到的功能部件。

解：直接列表：

	取指	译码	执行	访存	写回
*R*型	PC 指令存储器	寄存器	ALU	/	寄存器
*I*型	PC 指令存储器	寄存器	ALU	数据存储器	寄存器
*J*型	PC 指令存储器	寄存器	ALU	/	PC