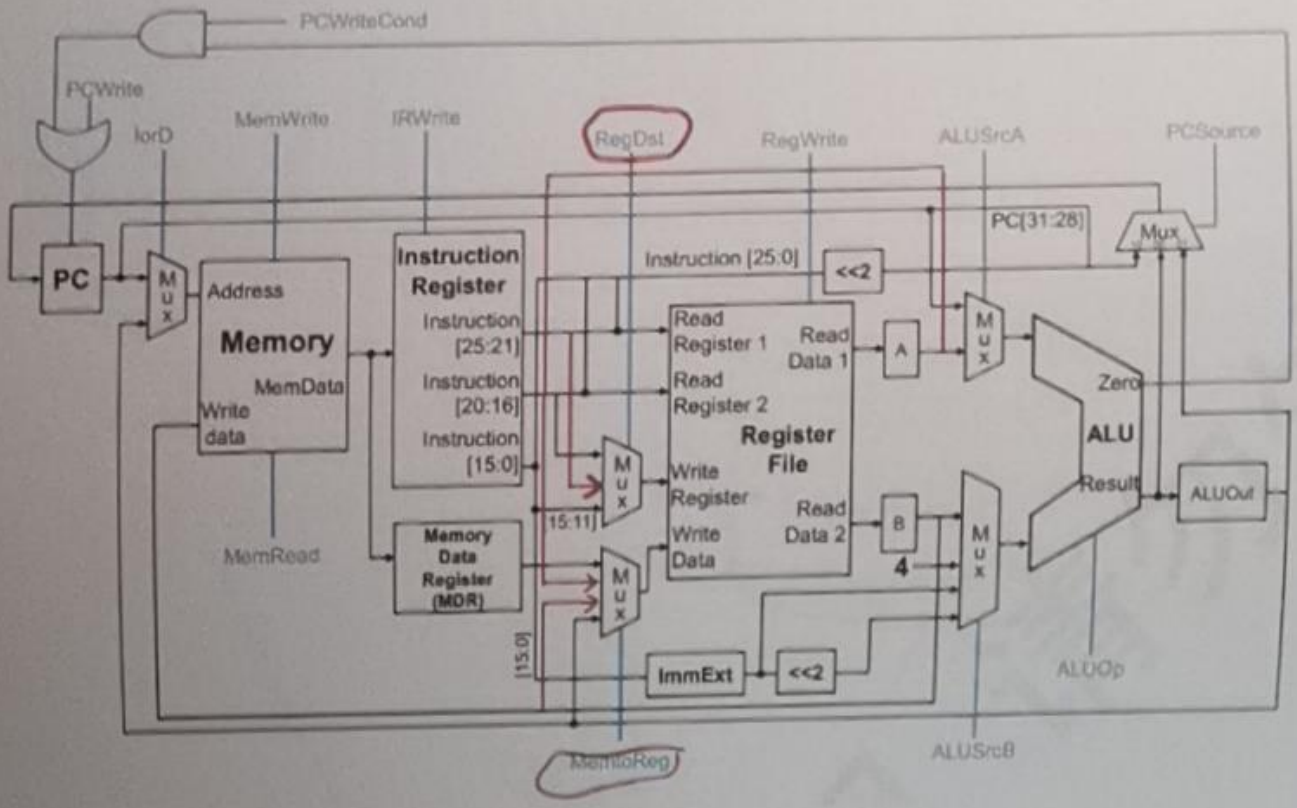
无04 2019012137 张鸿琳

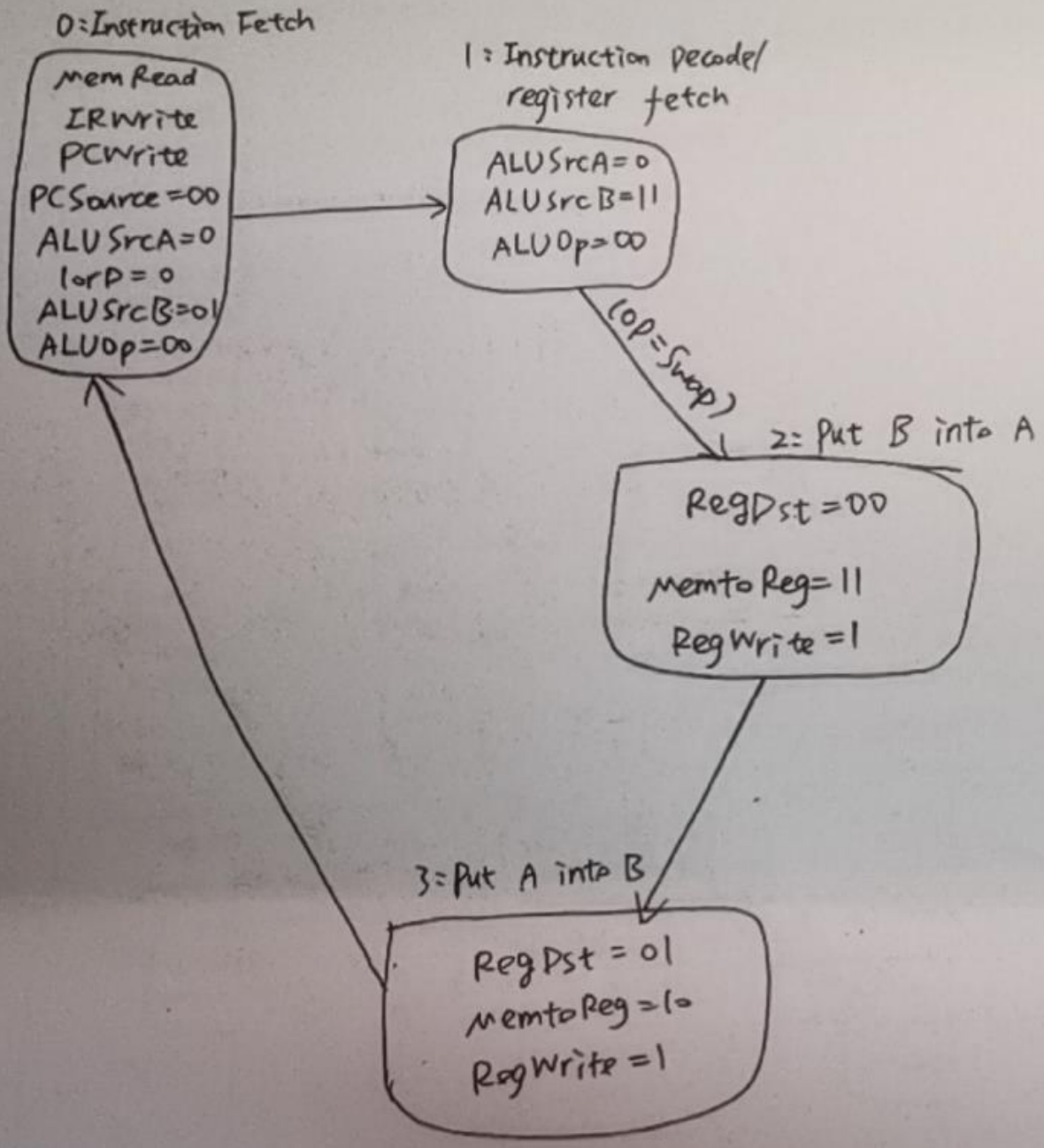
7.(b)

多周期处理器可以完成这条指令。需要修改的组合逻辑数据通路见下图：



红线为修改的部分，其中MemtoReg变为2bit，其等于00时选择MDR中数据，等于01时选择ALUOut中数据，等于10时选择A中数据，等于11时选择B中数据。另外，RegDst也变为2bit，其等于00时选择Instruction[25:21]的地址，等于01时选择Instruction[20:16]的地址，等于10时则选择Instruction[15:11]的地址。

执行过程如下图：（假设指令Swap中[25:21]和[20:16]是需要交换的两个寄存器的地址）



9.(a)

至少需要5条指令来完成：

- $A(nand)A = \overline{A}$
- $B(nand)B = \overline{B}$
- $A(nand)B = \overline{AB}$
- $\overline{A}(nand)\overline{B} = (A+B)$
- $(A+B)(nand)(\overline{AB}) = A \odot B$

200+50+150+200+10 = 610 ps

9.(b)

复杂ALU的时钟周期为（按耗时最长的指令lw计算）$200 + 20 + 50 + 150 + 200 + 10 = 630ps$，而简单ALU的时钟周期为$200 + 20 + 50 + 100 + 200 + 10 = 580ps$，假设XNOR的比例为x，为了使复杂ALU具有性能优势，假如其他所有指令只需要一个时钟周期，则
$630 \times 1 < 580 \times (1 - x + 5x)$，得到$x > 2.1\%$。

200+50+100+200+10 = 560 ps

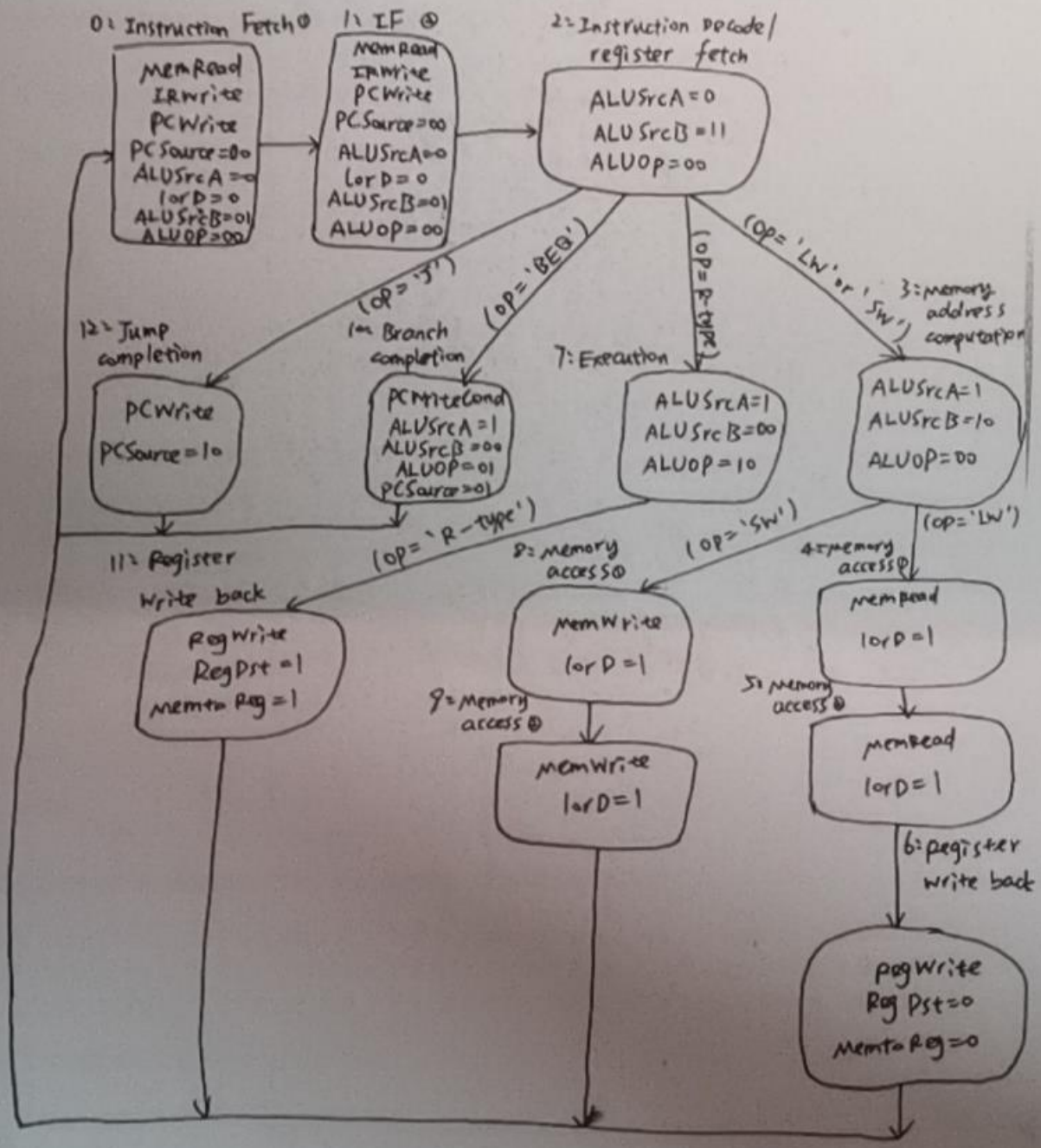
14.

- RegWrite=0：R型指令无法将数据存入RF；lw无法将数据存入RF。

610×1 < 560×(1−x+5x) ⇒ x > 2.23%

- MemRead=0：R型指令，lw，sw，beq，j均无法正常执行，因为指令无法从存储器中取出。
- MemWrite=0：sw无法将寄存器中数据存入存储器。
- IRWrite=0：R型指令，lw，sw，beq，j均无法执行，因为从存储器中读出的指令无法存到IR寄存器中。
- PCWrite=0：R型指令，lw，sw，j无法更新PC。
- PCWriteCond=0：beq无法在判定两个寄存器数据一样后立刻更新PC。

19.(a)



19.(b)

① i 处理器：

- R型：4个时钟周期，800ps
- lw：5个时钟周期，1000ps
- sw：4个时钟周期，800ps
- beq：3个时钟周期，600ps
- J型：3个时钟周期，600ps

② ii 处理器：

- R型：5个时钟周期，500ps
- lw：7个时钟周期，700ps
- sw：6个时钟周期，600ps
- beq：4个时钟周期，400ps
- J型：4个时钟周期，400ps

可以看出，ii 处理器的各项指令延时均小于 i 处理器。

20.

根据分析可以得到各种不同指令的时钟周期，见下表：

指令类型	M1周期数	M2周期数	M3周期数
R型或I型	4	3	3
J型	3	3	3
beq类	3	3	3
lw类	5	4	3
sw类	4	4	3

所给指令集中不同指令类型的比例如下表：

指令类型	出现比例(%)
R型或I型	48
J型	1
beq类	13
lw类	26
sw类	10

那么根据所给指令测试集的各种指令出现比例，可以算出各自CPI为：

- M1: $CPI_{int} = (4 \times 48 + 3 \times 1 + 3 \times 13 + 5 \times 26 + 4 \times 10)/100 = 4.04$
- M2: $CPI_{int} = (3 \times 48 + 3 \times 1 + 3 \times 13 + 4 \times 26 + 4 \times 10)/100 = 3.3$
- M3: $CPI_{int} = (3 \times 48 + 3 \times 1 + 3 \times 13 + 3 \times 26 + 3 \times 10)/100 = 2.94$

故而可以得到平均指令执行时间如下表：

	平均指令执行时间(ns)
M1	1.154
M2	1.031
M3	1.050

可见在该指令测试集下，M2最快。

存在不同的指令测试集可以使另一台机器更快，因为不同处理器对不同指令的处理速度各有快慢，如下表：

指令类型	M1执行时间(ns)	M2执行时间(ns)	M3执行时间(ns)
R型或I型	1.143	0.938	1.071
J型	0.857	0.938	1.071
beq类	0.857	0.938	1.071
lw类	1.429	1.25	1.071
sw类	1.143	1.25	1.071

可以看到M1执行J型和beq类指令最快，而M2执行R型或I型指令最快，M3执行lw和sw类最快，只需要调整比例，三种处理器均有可能是最快的。比如假如执行的工作主要是数据的搬运工作，代码基本上全是lw和sw类，假设lw类占40%，sw类占30%，R型或I型占30%，那么M1平均执行时间为1.2574ns，M2为1.1564ns，而M3只有1.071ns，M3变为最快的；如果程序中含有大量的指令跳转，使得J型和beq型指令的比例占到60%，而R型或I型占20%，lw类占5%，sw类占15%，则可算出M1平均执行时间为0.9857ns，M2为1.0004ns，M3为1.071ns，M1又变为最快的了。