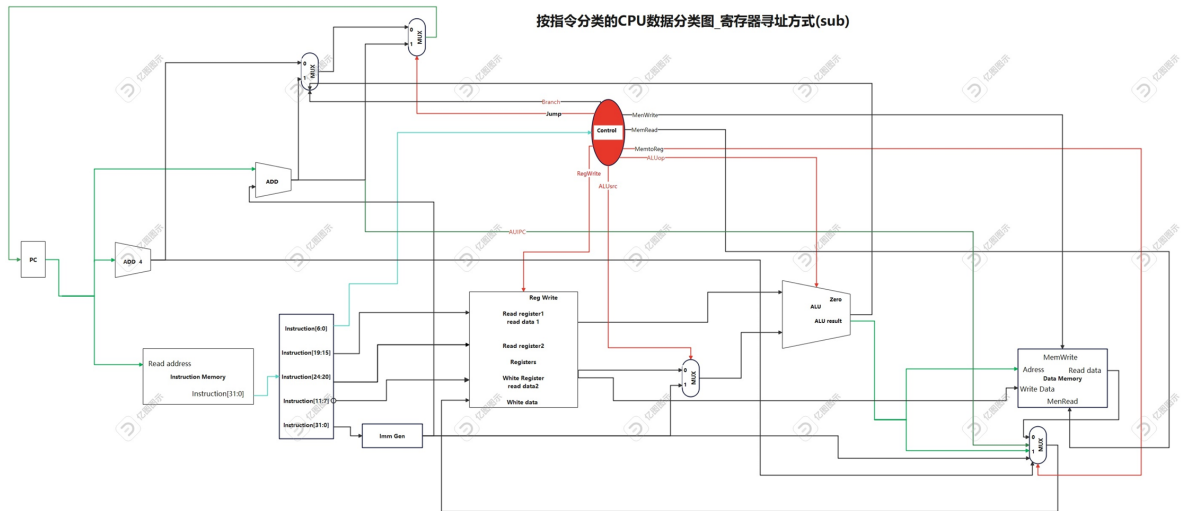


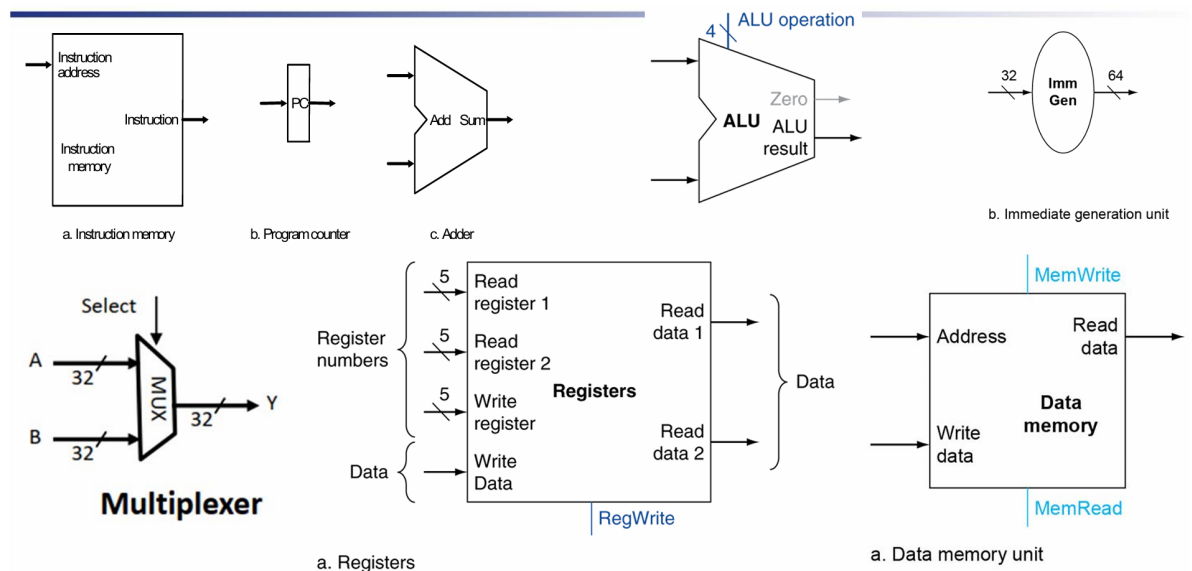
chapter4-1 单周期 CPU

这里主要讲解每个元件的功能、每类指令的执行方法，因为考试要考，理解之后对实验也较有帮助

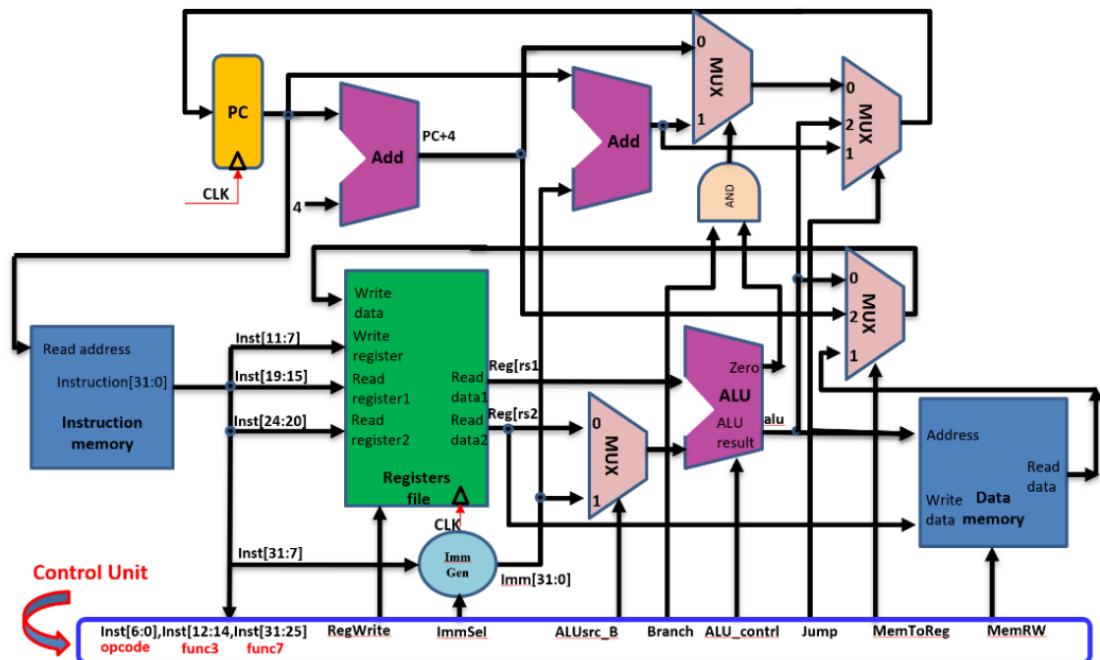


想必经过课程，大家都应该理解了执行过程

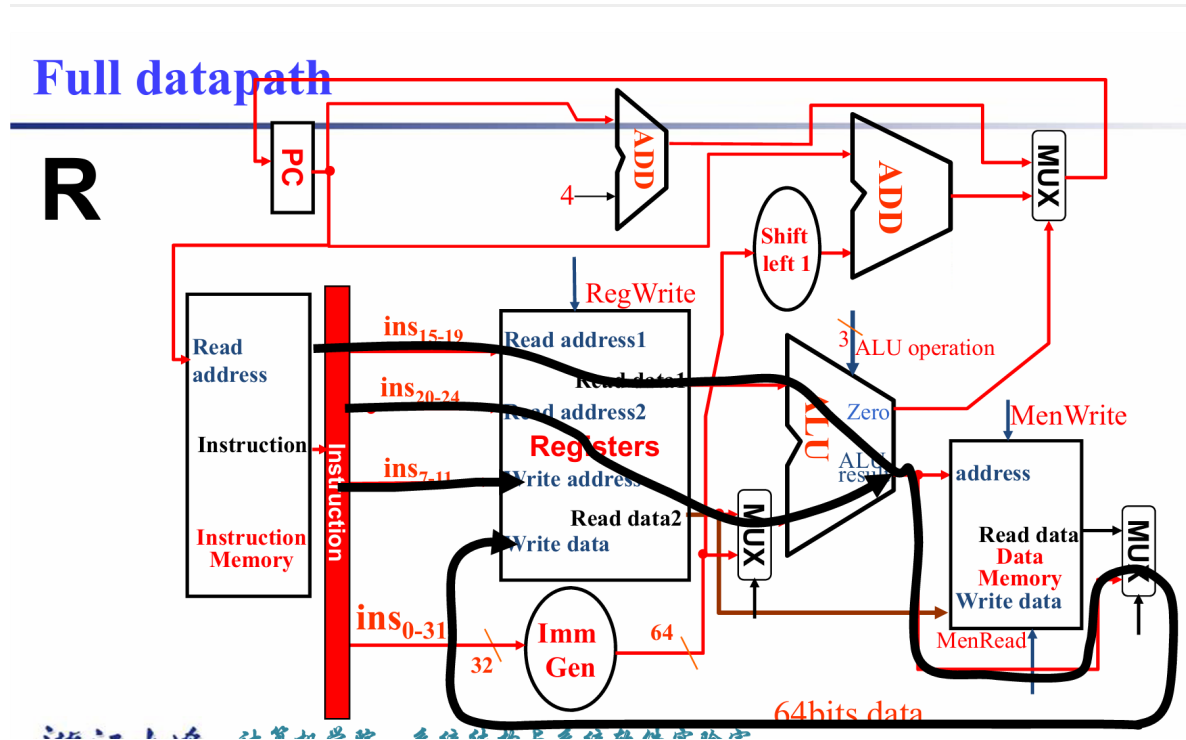
数据通路



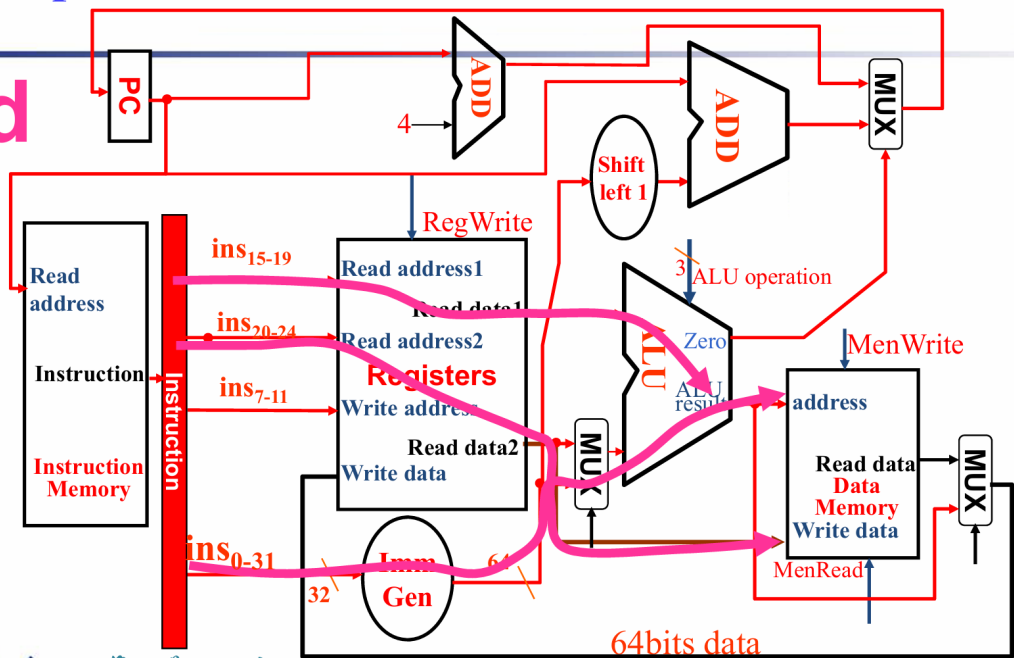
由于这张图似乎在考场上较为常见我们以此为例



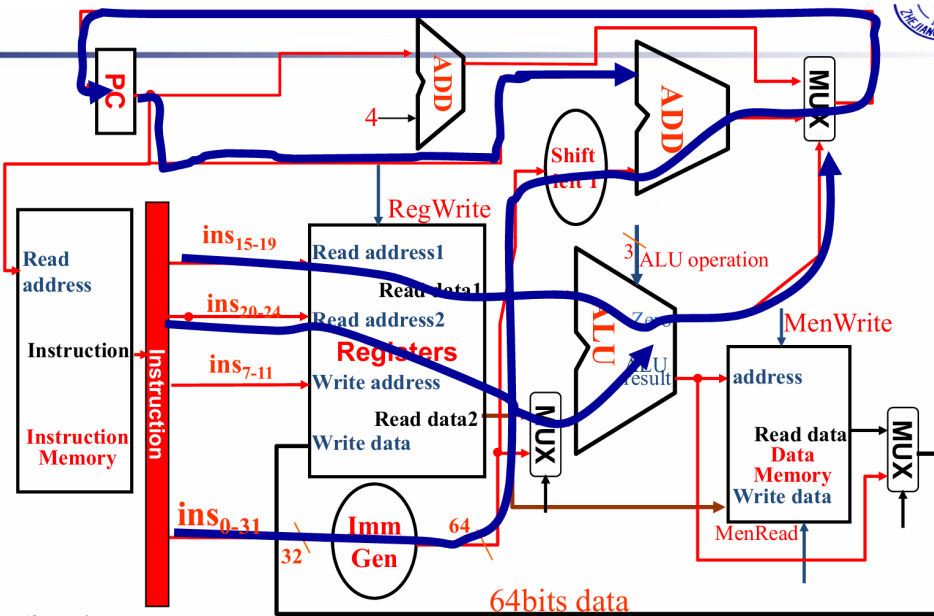
指令需要的数据通路



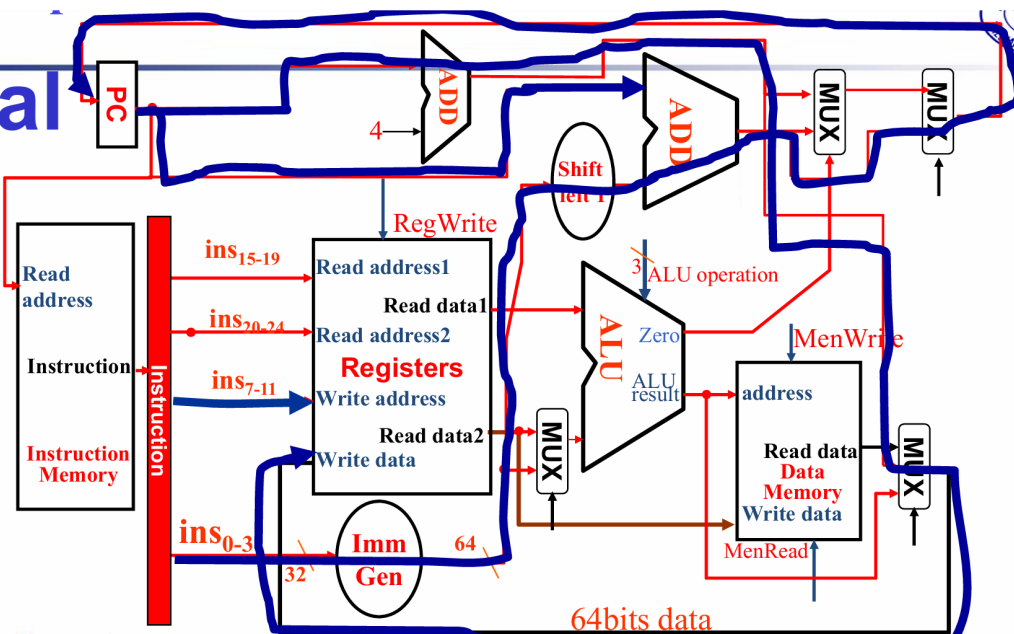
S-sd



SB-beq



UJ-jal



控制信号

指令类型	reg write	ALU op	ALU srcB	Branch	Jump	memory to reg	memory read	memory write	经过元件数量
R-type	1	10	0	0	0	00	0	0	4
load	1	00	1	0	0	01	1	0	5
立即数计算	1	11	1	0	0	00	0	0	4
jalr	1		1	0	2	10	0	0	4
store	0	00	1	0	0	xx	0	1	4
branch	0	01	0	1	0	xx	0	0	3
Jal	1		x	0	1	10	0	0	3
lui	1		x	0	0	11	0	0	3
auipc	1		x	0	0	4	0	0	

经过的元件有助于计算最长耗时的指令，从而计算出最短时钟周期

trap

参考了 [鹤翔万里 特权级 ISA](#)

RISC-V 将能引起当前程序中断，使 CPU 转到特定代码的事件称为**陷阱 (Trap)**，其分为两类，**中断 (Interrupt)** 和**异常 (Exception)**：

- 中断是硬件产生的，异步处理，是正常事件。包括软件中断、时钟中断、外部中断等。
- 异常是软件产生的，同步处理，是非正常事件，可能会导致程序终止。例如指令异常等。

RISC-V 指令集有三种特权模式，分别是 Machine (M)、Supervisor (S) 和 User (U)，除此之外还有 Hypervisor (H) 模式，不过貌似不常用。

等级	编码	名称	缩写
0	00	User/Application	U
1	01	Supervisor	S
2	10	Hypervisor	H
3	11	Machine	M

CSR寄存器

名称	位置	功能
mstatus	0x300	中断类型
mie	0x304	不同中断使能
mip	0x344	中断待处理
mtvec	0x305	trap 处理程序位置
mepc	0x341	原先断点，中断 PC+4，异常 PC
mcause	0x342	异常原因
mtval	0x343	异常指令或者地址

中断处理流程

RISC-V 中外部中断必须通过 CSR 来开启，开启中断由两个步骤：

- mstatus[MIE] 是全局中断使能位，设置为 1 才会全局开启中断
- mie 寄存器中是针对各种中断类型的使能位，要将需要的位设置为 1

中断响应程序的入口地址由 mtvec 寄存器指定，如前面写到的，它分为两种模式：

- 直接模式（Direct），所有 trap 都跳转到 mtvec 寄存器指定的地址进行处理
- 向量化模式（Vectored），中断将根据中断类型跳转到不同偏移位置的中断响应程序，异常仍使用同一个响应程序

当中断发生时，CPU 会：

- 将发生异常的指令（或下一条指令）的地址保存到 mepc 寄存器
- 将中断类型码保存到 mcause 寄存器
- 如果中断带有附加信息，将其保存到 mtval 寄存器
- 如果是外部引发的中断，令 mstatus[MPIE] = mstatus[MIE]（保存），然后令 mstatus[MIE] = 0（关闭中断）
- 将当前特权模式保存到 mstatus[MPP] 中

- 将当前特权模式设置为 Machine 模式
- 根据 mtvec 寄存器的设置，跳转到对应中断响应程序

中断处理结束后要使用 mret 指令进行返回，它会：

- 令 `mstatus[MIE] = mstatus[MPIE]`（恢复），然后令 `mstatus[MPIE] = 1`
- 将当前特权模式设置为 `mstatus[MPP]` 中保存的值
- 将 `mstatus[MPP]` 设置为 U 模式
- 将 pc 值设置为 `mepc` 值，即跳转回中断前的程序