



# 计算机系统贯通课程-计算机系统 I

——Project-1 单周期 CPU 数据通路设计

洪奕迅 3230102930@zju.edu.cn

史璐欣 3220104390@zju.edu.cn

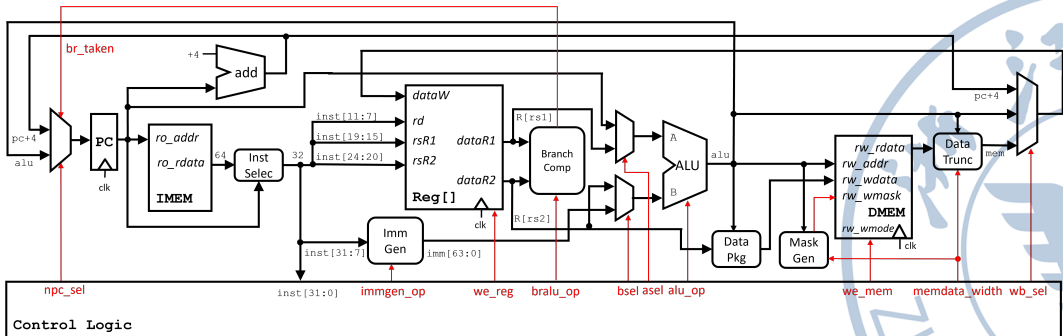
计算机学院  
浙江大学

2025 年 5 月 8 日

# 注意事项

- 本次实验一直持续到本学期最后一周，每周实验课都可以验收，因此不再接受 ddl 以内的线上验收（最后一周实验课结束后会再验一会，课后则全部视为迟交）
- 本次实验非常非常非常重要，单周期 CPU 的代码大家会一直修修改改用到 sys3（甚至于说你们下学期的 Lab0 就是这学期的 project，这学期写好了可以白拿分），当然写不好的话后续所有硬件实验几乎都无法完成，因此请大家认真对待，并且尽量把代码写得规范一些（如果你不想下学期看不懂自己的代码）

# CPU 数据通路



在使用 CPU 过程中我们向 CPU 输入指令从而操控并修改 CPU 内部的部件，因此一条指令在输入后可以被分为两部分：

- 红线：控制流，事实上就是这条指令，Controller 把指令译码为一系列的控制信号
- 黑线：数据流，CPU 内各种信号的流动（这届实验课要讲的内容）

# 关于 Memory

我们提供了一个相当复杂的接口，但这次实验几乎不需要管这些东西，你只需要注意访存地址和取回访存数据即可，这次实验的 mem 可以说是完全同步返回的，因此不需要对 valid-ready PTSD，各个通道复杂的握手信号在这次实验中并不重要：

Time	
raddr[63:0] = 0000000000000000F	+ 00000000+ 00000000000000001 00000000+ 00000000+ 00000000+
rdata[63:0] = 0020873300700113	0020873300+ 0030009300100193 00100093+ 00300093+ 187718E3+
raddr[63:0] = 00000000000000007C	+ 00000000+ 00000000+ 00000000+ 00000000+ 00000000+ 00000000+
rdata[63:0] = 00F0009300600193	+ 0020A13300D00113 147114E300100393 0100009300500193

图：上下两条分别是 d/imem 的读通道，可以看到访存数据的发回和访存地址的发送是同步的

# CPU 会做什么-普通指令

我们以经过部分最多的存储指令为例，假设有命令 `sw a0, 16(sp)`，我们来看一下数据流在流动过程中究竟在干什么：

- IF 阶段：取出指令，且不是跳转指令 PC 只需要 +4
- ID 阶段：进入 Controller 译码，获得指令相关控制信号，这节课假设这步是从天而降的，同时在 Regfile 中读取寄存器的值
- EXE 阶段：根据控制信号进行算术运算，例如这条指令我们将使用 ALU 计算访存地址  $sp + 16$ ，因此此前从 Regfile 中取出的第一个数并不参与运算（但 `add a2, a0, a1` 明显不是这样，这就是控制信号的作用），此外这条指令不涉及条件跳转，因此 Cmp 会被设置为不进行运算
- MEM 阶段：根据 EXE 阶段计算得到的地址  $sp + 16$  访存得到要写回寄存器的数据
- WB 阶段：把 MEM 阶段读到的数据写回寄存器 a0

# CPU 会做什么-影响控制流的指令

有些指令会对控制流产生影响，即会修改 PC 的值来进行跳转，假设有命令 `beq x1, x2, label`，我们来看一下数据流在流动过程中究竟在干什么：

- IF 阶段：取出指令，且 PC 根据 Cmp 行为的结果来修改为  $PC+4$  或 `label`（这只适用于现在设计的单周期 CPU，因为过程都是组合电路，因此我们可以在 IF 阶段直接知道下一个 PC 的值）
- ID 阶段：进入 Controller 译码，获得指令相关控制信号，这节课假设这步是从天而降的，同时在 Regfile 中读取寄存器的值
- EXE 阶段：根据控制信号进行算术运算，例如这条指令我们将使用 ALU 计算跳转地址 `label`（事实上是当前  $PC + \text{偏移量}$ ），Cmp 则会比较传入操作数是否相等，若相等则给出信号使 PC 跳到计算出来的地址，否则 PC 正常  $+4$
- MEM 阶段：这条指令不与 `dmem` 交互，因此这一阶段在干什么无所谓
- WB 阶段：这条指令也不需要写回