

## 第八章 中央处理器

第一次课--课后思考题：11月9日

**本章思路：**设计能执行 8.1 节设定的 8 条指令对应的系统，一旦设计完成，由这 8 条指令构成的任何程序理论上都能在该系统上运行了。

- 1、CPU 按功能模块划分有哪些模块？从设计角度侧重完成哪些部分实现（本章重点）？
- 2、CPU 主要完成哪四个基本功能？
- 3、理解 CPU 指令执行中各部件和数据通路的关系，其中各部件需要的**操作控制信号**的作用（区别于地址线、数据线）；
- 4、理解指令执行的三个基本步骤：取指、译码和执行（可以再细分）；
- 5、本教材模型机的设计需求：有哪几类指令？分别对应哪几条指令？它们采用哪些指令封装？操作码分别是多少？针对不同指令，各种指令封装中各字段的含义是什么？**这部分很重要，牵扯到数据通路的选择和走向等。**
- 6、理解时序电路中**触发**的概念，理解**时钟**在时序电路中的重要性，什么是**时钟周期**（教材里也叫节拍）？什么是**触发**？理解计算机在时钟的基准下工作的大背景。
- 7、什么是传输时间？为什么传输周期  $T$  要去取那个**最大值（快的等慢的）**？
- 8、如何理解 CPU 的数据通路（结合城市间道路设计类比）？总线式和直连式的优缺点是什么？
- 9、明白书上图 8.2 上下两排分别是**时序器件**和**组合器件**，上面器件的时钟信号都省略了，我们约定都是时钟上升沿触发。我们的课程背景，图中 IM 和 DM 是分开的（哈弗结构），其他部件在 CPU 内部，注意 IM 是**只读的**，即对 IM 只有读操作，没有写操作。
- 10、**熟悉图 8.2 中各部件的输入、输出及控制信号或时钟信号的关系。（接下来我们会用他们实现模型机）--重点**

第二次课--课后思考题：11月16日

- 1、不看教材能把各指令的取指、译码和执行的数据通路分别画出来。在本节 8.4.1 学习中淡化了控制信号的作用，强调的是数据和地址的联通，单周期控制信号相对简单，很多就直接用时钟信号了，等学了多周期控制信号的设计后和单周期这部分可以做比较。（数据通路的实现是考点）
- 2、为什么 DM 的读写不能同时进行，而通用寄存器组的读写可以同时进行？
- 3、模型机的指令集设计中，运算类指令支持不支持寄存器和立即数的运算？如果要支持，需要做什么改动？
- 4、理解 CPU 指令执行中各部件和数据通路的关系，其中各部件需要的操作**控制信号**的作用（区别于地址线、数据线）；
- 5、在 beqz 指令的数据通路中为什么设计一个**左移两位**的处理？  
注意我们的前提，内存是字节寻址（一个字节内存单元分配一个内存地址），指令在 IM 中是 4 字节对齐存放的，为了保证生成的新目标地址是 4 字节对齐的，所以要进行左移两位（保证 PC 寄存器的后两位始终为 0）。
- 6、理解图 8.11 的数据通路**整合过程**，其中几个**多路选择器**的作用。在此基础上理解图 8.12（我们第一个较完整的数据通路设计完成）的整合过程，**注意该图中 9 路（11bits）控制信号的作用。**
- 7、从两次整合过程理解，理解 CPU 每增加一种指令，甚至是一种指令封装（如模型机中要支持寄存器和立即数的运算）的实现都会增加 CPU 的设计复杂度和成本。
- 8、理解**主控制器**和**ALU 控制器**的作用，它们的输入输出信号都有哪些？为什么需要 ALUOp？此处的 ALUOp 可不可以改为 1 位？体会指令译码的过程。
- 9、图 8.13 是在图 8.12 数据通路设计基础上**增加了控制器设计**后更完善的一版----**单周期数据通路设计**（单周期的本质是所有指令在一个时钟周期内完成，那就又存在**快的等慢的**的问题，此处的时间粒度是什么？**如何改进以弥补单周期的不足？**）。能将我们设计的三类功能、两种封装、8 条指令的执行过程在图 8.13

中的执行过程走一边，注意各自经历的数据通路、组件和需要的控制信号、状态描述清楚。（不光能看懂，要求会设计）--- **考点!!!**

- 10、学到这里，再去体会 RISC 处理器为什么一般体积小、价格和功耗低？当然它也有缺点，是什么？
- 11、了解主控制器和运算控制器设计思路，着重关心他们的输出（也就是各**控制信号**）在数据通路中的作用。

第三次课--课后思考题：11 月 23 日

- 1、为什么要采用**多周期**实现？它是如何提高 CPU 的**指令执行效率**的？
- 2、在多周期背景下，**指令执行周期 CPI** 引出的意义是什么？
- 3、理解多周期实现机制后，思考 CPU 的**主频提升**依据是什么，主频仍然存在快的等慢的，这里时间粒度是什么？
- 4、理解系统性能和 ISA 设计、数据通路实现、主频大小等有关，这些又相互制约。
- 5、比较单周期数据通路图 8.13 和多周期实现图 8.14 的差异，增加了哪些临时寄存器和控制信号？
- 6、在提高主频上，理解由“单周期”→“多周期”的设计思路，在多周期下“快的等慢的”依然存在，它的粒度单位是什么？
- 7、多周期实现解决了单周期相应问题的同时，增加了哪些成本？周期的划分要考虑什么因素？
- 8、理解“微操作”的**原子性**特点。
- 9、多周期实现中为什么增加了 IRWrite 和 PCWrite 两个控制信号？
- 10、结合表 8.6 和图 8.14，对各指令的多周期的执行过程所经历过的数据通路、组件和需要的控制信号状态能描述清楚。（**重要考点**）  
思考题 1：Load 的写回能不能整合到上一个节拍（理由：这样所有指令就都只需要 4 个节拍，PCWrite 控制就可以用主频时钟 4 分频的时钟信号直接控制了）？  
思考题 2：Load 指令的 PCwrite 写有效能不能放到写回节拍去实现。（理由：其他指令都是在最后一个节拍实现的）
- 11、什么是时钟周期？节拍？和指令周期？理解计算机系统的时序系统设计。（说明：我们的教材里一个微操作就占用一个节拍，但实际中可能一个微操作占用多个节拍完成。）
- 12、结合图 8.15 理解控制器工作原理，它的输入有哪些信号？（如程序状态字 PSW、中断信号、译码结果、时序信号等）输出有哪些信号？（微操作控制信号）
- 13、比较**多周期下的控制信号**（P213）与**单周期下的控制信号**（P203）有什么不同？
- 14、什么是同步和异步时序控制？比较优缺点，CPU 内部为什么采用同步控制？
- 15、理解硬连接方式实现控制器，本质就是通过实现输入（指令操作码字段）和输出（微操作控制信号）的组合和时序电路的实现。
- 16、流水的数据通路本学期不要求掌握，但要理解“流水一定是多周期的，多周期不一定是流水的”，流水机制本质是“**指令级并行**”。

**拓展：**硬件逻辑电路设计出来以后，可以了解 Verilog 或 VHDL 等硬件描述语言来实现该设计，并用仿真软件验证你的设计。