2021-2022 冬季学期
《计算机组成原理 A(1)》课程报告

严昕宇 20121802

(计算机工程与科学学院)

1 引言

计算机组成原理是一门计算机专业的基础核心课。它的先修课程为数字逻辑，后续课程为计算机体系结构、接口技术、操作系统、编译原理等。作为一门承上启下的专业课程，计算机组成原理占有十分重要的地位和作用。

由于受到疫情影响，考试需以报告形式提交。因此，这份报告应运而生。借此机会，我将在报告中阐述个人在完成计算机组成原理 A(1)学习后的收获、课本相关知识点的思考与分析以及期待学习的其他知识。

2 知识点总结与收获

在冬季学期的学习过程中，通过系统性地学习计算机各个部件，我建立起计算机硬件的基本组成框架，掌握计算机硬件的基本原理。但与此同时，由于计算机组成原理是一门理论性很强的课程，且涉及大量硬件知识。因此在学习过程中，我真切地感受到部分知识点的抽象、难以理解、庞杂而零碎，难度较大，需要利用实验课与课外时间思考并实践，以加深理解。

因此，在学习过程中，我也借助思维导图梳理知识脉络，强化记忆。各章节知识点的思维导图请见**附页** A 与**压缩包**中**【思维导图】**文件夹。

2.1 第1章 计算机系统概论

本章主要对计算机系统做了概括性的介绍，在学习过程中，我逐步建立起计算机总体结构的粗略概念，这为深入学习后面各章打下基础。

2.1.1 知识点总结

- 习惯上所称的“电子计算机”是指电子数字计算机，而非电子模拟计算机。它分为专用计算机和通用计算机两大类。而通用计算机分为超级计算机、大型机、服务器、PC 机、单片机、多核机六类。
- 计算机的硬件是由电子器件构成的，包括运算器、存储器、控制器、适配器、输入输出设备。存储程序并按地址顺序执行，这是冯·诺依曼型结构的设计思想，也是 CPU 自动工作的关键。而哈佛结构则分离了指令与数据，提升了计算机速度。这可以用计算机的性能指标衡量，主要有 CPU 性能指标、存储器性能指标和 I/O 吞吐率。
- 计算机的软件是计算机系统结构的重要组成部分，一般分为系统程序和应用程序两大类。
- 因而计算机系统是一个由硬件、软件组成的多级层次结构，它通常由微程序级、一般机器级、操作系统级、汇编语言级、高级语言级组成，每一级上都能进行程序设计，且得到下面各级的支持。且软件与硬件在逻辑功能上等价。

2.1.2 收获

在学习第 1 章的内容之前，我存在着一个疑惑：集成电路的发展是不是有一定的规律性？而摩尔定律告诉我了一个趋势——“集成电路上可容纳的晶体管数目，约每隔 18 个月便会增加一倍，性能也将提升一倍”。 它不仅揭示了信息技术进步的速度，更在接下来的半个实际中，犹如一只无形大手般推动了整个半导体行业的变革。它可以帮助我更好的理解为什么在计算机，特别是芯片领域的发展是如此日新月异。

除此之外，我也学习到可以计算机性能的参数指标，如 MIPS、FLOPS 等。在购买计算机后，我们常会使用跑分软件进行 Benchmark 基准测试，以测试计算机的性能。但是往往是知其然而不知其所以然，只会使用，而不知道如何得出结论的。通过学习计算机性能指标后，我明白了其背后的评价标准。

最重要的一点则是冯·诺依曼计算机的设计思想。早期的计算机是由各种门电路组成的，这些门电路通过组装出一个固定的电路板，来执行一个特定的程序，一旦需要修改程序功能，就要重新组装电路板，所以早期的计算机程序是硬件化的。但是冯·诺依曼提出一种计算机的设计思想：①存储程序并按地址顺序执行；②采用二进制表示数据与指令；③计算机由五大部件组成。这种结构消除了硬件控制程序的状况，导致硬件和软件的分离，实现了可编程的计算机功能，促进了计算机的发展。因而学习冯·诺依曼结构帮助我更好理解现代计算机的基本设计思想。

2.2 第2章 运算方法和运算器

本章中，我首先学习了数据与文字的表示方法，并通过定点数与浮点数的运算方法，掌握了相应的定点运算器与浮点运算器的组成和硬件实现原理。

2.2.1 知识点总结

- 一个定点数由符号位和数值域两部分组成。按小数点位置不同，定点数有纯小数和纯整数两种表示方法。按 IEEE754 标准，一个浮点数由符号位 *S*、阶码 E、尾数 *M* 三个域组成。其中阶码 *E* 的值等于指数的真值 *e* 加上固定偏移值。
- 数的真值变成机器码时有四种表示方法：原码表示法、反码表示法、补码表示法、移码表示法。其中移码主要用于表示浮点数的阶码 *E*，以利于比较两个指数的大小和对阶操作。
- 为运算器构造的简单性，运算方法中算术运算通常采用补码加、减法，原码乘除法或补码乘除法。为了运算器的高速性和控制的简单性，采用了先行进位、阵列乘除法、流水线等并行技术措施。
- 在运算过程中，可能会出现溢出的情况。对于定点整数可以采用双符号位法（变形补码法）或单符号位法检测。而对于浮点数的尾数溢出可调整阶码。

2.2.2 收获

在刚开始接触 C 语言程序设计的时候，经常会发现当调用 printf()函数后，输出的结果是一串乱码，例如有“锘斤拷”等。在学习了第 2 章中关于字符与字符串的表示方法，与研讨环节的资料查找后，我明白这涉及到字符编码，特别是 GBK 字符集和 Unicode 字符集之间的转换问题。Unicode 和老编码体系的转化过程中，存在一些字，用 Unicode 是没法表示的，Unicode 官方用了一个占位符来表示这些文字，这就是：U+FFFD REPLACEMENT CHARACTER。而 U+FFFD 的 UTF-8 编码出来，恰好是 '\xef\xbf\xbd'。放到中文计算机使用的 GBK/CP936/GB2312/GB18030 环境中显示，最终结果就是常见的乱码——“锘斤拷”：锘（0xEFBF）、斤（0xBDEF）、拷（0xBFBD）。正是通过学习和调研，令我收获了乱码生成背后的原因。

与此同时，在 2.2 定点加法、减法运算小节中，讲述了二进制加法/减法器的逻辑结构。这与数字逻辑课程中所学习到的内容紧密结合。因此，我感受到其他课程知识的在计算机组成元件上的实际应用，同时也感叹其中蕴含的简洁美——通过方式控制 M 信号，可以同时实现加法和减法操作。此设计大大简化了电路的复杂性，拓展了我的视野。

2.3 第4章 指令系统

在本章的学习过程中，我主要了解到指令系统的发展历史和性能要求，学习了指令的一般结构——由操作码字段 OP 与地址码字段 A 组成。同时重点理解了多种寻址方式，指令的分类和功能。

2.3.1 知识点总结

- 一台计算机中所有机器指令的集合，称为这台计算机的指令系统，它是表征一台计算机性能的重要因素指令格式是指令字用二进制代码表示的结构形式，通常由操作码字段和地址码字段组成。操作码字段表征指令的操作特性与功能，而地址码字段指示操作数的地址。目前多采用二地址、单地址、零地址混合方式的指令格式。指令字长度分为：单字长、半字长、双字长三种形式。
- 形成指令地址的方式，称为指令寻址方式。有顺序寻址和跳跃寻址两种，由指令计数器来跟踪。形成操作数地址的方式，称为数据寻址方式。操作数可放在专用寄存器、通用寄存器、内存和指令中。数据寻址方式有隐含寻址、立即寻址、直接寻址、间接寻址、寄存器寻址、寄存器间接寻址、相对寻址、基值寻址、变址寻址、段寻址、堆栈寻址等。按操作数的物理位置不同，有 SS 型、RR 型和 RS 型。
- 不同机器有不同的指令系统。一个较完善的指令系统应当包含数据传送类指令、算术运算类指令、逻辑运算类指令、程序控制类指令、I/O 类指令、字符串类指令、系统控制类指令。
- RISC 指令系统是目前计算机发展的主流，也是 CISC 指令系统的改进，它的最大特点是：①指令条数少；②指令长度固定，指令格式和寻址方式种类少；③只有取数/存数指令访问存储器，其余指令的操作均在寄存器之间进行。

2.3.2 收获

在学习第 4 章指令系统相关知识之前，我曾经产生过一个疑问：“存储器可用来存放数据，那么计算机如何在其中寻找到执行指令时所需要的数据呢？”通过在第 4 章的学习，我认识到在指令执行的过程中，其实存在着三个操作数的来源：①直接由指令中的地址码部分给出；②存放在 CPU 内部的通用寄存器中；③存放在内存中。特别是存放在内存中时，可以直接给出操作数的实际访问地址，也可以在指令的地址字段给出形式地址，再依据变换得到有效地址再取操作数。

正因此其复杂性，便出现了多种操作数的寻址方式，每种方式都有其优缺点和使用场景。而且不同的指令系统会采用不同的方式，例如会把常见的寻址方式组合起来，构成更复杂的复合寻址方式。书本上介绍的是基础，只有掌握好基本的寻址模型，才能更好的理解其组合应用，避免混淆。

2.4 第5章 中央处理器

本章中，我主要学习了 CPU 的功能和基本组成，并了解了指令周期、流水 CPU 和 RISC CPU 的概念。

2.4.1 知识点总结

- CPU 是计算机的中央处理部件，具有指令控制、操作控制、时间控制、数据加工等基本功能。早期的 CPU 由运算器和控制器组成。随着集成电路技术的发展，当今的 CPU 芯片变成运算器、cache 和控制器三大部分，CPU 中至少有六类寄存器：指令寄存器、程序计数器、地址寄存器、数据缓冲寄存器、通用寄存器、状态条件寄存器。
- CPU 从存储器取出一条指令并执行这条指令的时间和称为指令周期。CISC 中，由于各种指令的操作功能不同，各种指令的指令周期是不尽相同的。划分指令周期，是设计操作控制器的重要依据。RISC 中，由于流水执行，大部分指令在一个机器周期完成。时序信号产生器提供 CPU 周期(也称机器周期)所需的时序信号。操作控制器利用这些时序信号进行定时，有条不紊地取出一条指令并执行这条指令。

- 微程序设计技术是利用软件方法设计操作控制器的一门技术，具有规整性、灵活性、可维护性等一系列优点，因而在计算机设计中得到了广泛应用。
- 并行处理技术已成为计算机技术发展的主流。概括起来，主要有三种形式：①时间并行；②空间并行；③时间并行+空间并行。流水 CPU 是以时间并行性为原理构造的处理机。流水技术中的主要问题是资源相关、数据相关和控制相关，为此需要采取相应的技术对策，才能保证流水线畅通而不断流。

2.3.2 收获

在未学习这章的内容之前，我对 CPU 的理解停留在它只是一个高科技芯片。但是通过对 CPU 的详细讲述，我发现其不光有高集成度（包含了运算器、控制器、寄存器、运算器、cache 等），各部件之间也是通过总线 BUS 相互联系的，形成了一种精密的结构体系。不光是计算机组成上存在着如此严密的逻辑关系，在 CPU 的底层实现上也是如此，令我感叹计算机科学技术与集成电路技术的先进与发达。

第五章最后也介绍了流水 CPU，CPU 按流水线方式组织，可实现高速运行。其理念来自于我们生活中的流水线作业，这也使我领悟到设计并非只是天马行空，而是来源于生活。因此在学习之余也需要多思考、多观察、多尝试。

3 难点分析

3.1 超标量流水计算机的实现原理——以Pentium处理器为例

计算机自诞生到现在，人们追求的目标之一是很高的运算速度。早期的计算机基于冯·诺伊曼的体系结构，采用的是串行操作。这种计算机的主要特征是：计算机的各个操作（如读/写存储器，算术或逻辑运算，I/O 操作）只能串行地完成，即任一时刻只能进行一个操作。提高性能的关键是并行处理，其使得以上各个操作能同时进行，从而大大提高了计算机的速度。流水计算机与流水 CPU 正是并行思想的实例。

在第 5 章的流水线 CPU 技术中提到过一个名词——超标量流水计算机，并以 Pentium 微型机作为例子。但是课本上并未介绍其原理，仅给出了一张超标量流水线时空图。因此在好奇心的驱使下，我对这个知识点进行了调研与思考，并将以 Pentium 处理器为例，具体分析超标量流水计算机的实现原理。

3.1.1 指令级并行与指令流水线

在第四章指令系统中曾介绍过，指令是处理器执行的基本单位。多个指令之间可能存在相关，但也存在没有依赖关系的情况。没有相关的多个指令可以同时执行，存在相关的多个指令如果消除相关，也可以同时执行。所以，处理器需要发掘指令之间的并行执行能力，也就是提高处理器内部操作的并行程度，称之为指令级并行（Instruction-Level Parallelism, ILP）。

指令流水线实现了多条指令重叠执行，是指令级并行的一个成熟实现技术。Pentium 系列处理器的前身 80486 整数指令就已经实现了 5 级指令流水线。但在查阅资料后我发现，即使使用指令流水线，其效率也难以达到理想目标。因此，为进一步提高指令级执行的效率，Pentium 系列处理器还引入了超标量技术和动态执行技术。下面将以超标量技术为重点进行分析。

3.1.2 标量与超标量

超标量技术中的“标量”一词在书本上并未进行介绍。在查阅资料后发现，标量（Scalar）数据是指仅含一个数值的量。传统的处理器进行单值数据的标量操作，设计的是进行单个数值操作的标量指令，可以称之为标量处理器。

而超标量（Superscalar）一词是 1987 年造出的，超标量处理器是指为提高标量指令的执行性能而设计的一种处理器。处理器采用超标量技术，是指它的常用指令可以同时启动，并相互独立地执行。这样，处理器采用多条（超）标量指令流水线，就可以实现一个时钟周期完成多条指令的执行，大大提高了指令流水线

的指令流出率，从而实现处理器性能的提高。

3.1.3 Pentium 处理器上的实现原理

Pentium 处理器为了实现超标量技术，设计了两个可以并行操作的执行单元，形成了两条指令流水线。Pentium 的超标量整数指令流水线的各个阶段分成了 5 个步骤，但是其后 3 个步骤可以在它的两个流水线（U 流水线和 V 流水线）同时进行，如图 1 所示。

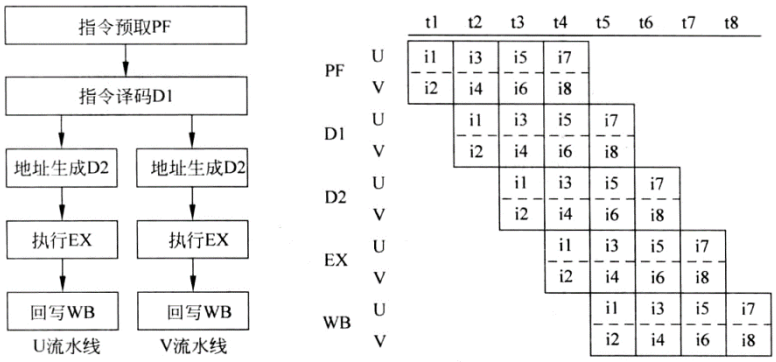

图 3.1 Pentium 的超标量指令流水线

两条流水线独立运行，均含有算术逻辑单元（ALU），且每条流水线含 5 级（取指令、译码、生成地址、执行指令、回写）。不同点在于，U 流水线可执行所有的整数运算指令，而 V 流水线仅执行简单的整数运算和数据交换指令。

查阅资料后，我发现相对于前代 80486，Pentium 设计了两条存储器地址生成（指令译码 2）、执行和回写流水线，其指令预取 PF 和指令译码 D1 步骤可以并行取出、译码两条简单指令，然后分别发向 U 流水线和 V 流水线。这样，在一定条件下，Pentium 允许在一个时钟周期中执行完两条指令，即实现了超标量流水。

3.1.3 超标量与超线程

在 CPU 所采用的技术中，存在着一个相似的名词——超线程（HT, Hyper-Threading）。那么超标量与超线程是同一个概念吗？经过调研与分析，我发现两者是不同的。

超线程是 Intel 公司提出的一种提高 CPU 性能的技术，可以将一个物理 CPU 当作两个逻辑 CPU 使用，使 CPU 可以同时执行多重线程，从而发挥更大的效率。超线程技术通过利用特殊的硬件指令，把两个逻辑内核模拟成两个物理芯片，让单个处理器都能使用线程级并行计算，进而兼容多线程操作系统和应用软件，减少 CPU 的闲置时间，提高 CPU 的运行效率。超线程技术原先只应用于 Xeon 处理器中，当时称为“Super-Threading”。之后陆续应用在 Pentium 4 HT 中。如今，几乎所有的 CPU 都是使用了这项技术。

但是超标量并不是超线程。超线程要求同一个核心有两套执行部件的同时，还有两套保存线程状态的寄存器。超标量是没有的，所以它不能同时执行两个线程上的两个指令，只能执行同一个线程上的两个指令。这两个指令不必是连续，可以是 CPU 的硬件分析出来的一段代码中，两个互相不依赖结果的任意指令。也就是说，例如 ABCD 四条指令，要求 CPU 能够分析出 C 指令的执行不依赖 AC 的结果。

3.1.4 总结

流水 CPU 是以时间并行性为原理构造的处理机，是一种非常经济而实用的并行技术。通过研究 Pentium 处理器的超标量流水的实现原理，我更好的掌握了流水 CPU 的结构，也了解到最新的超标量流水技术。在学习过程中，我也对比了超标量与超线程两者之间的区别。但是由于自身能力和知识面所限，对这部分内容的理解只停留在初步，为了更好掌握，我还希望学习计算机体系结构、微机原理等相关课程与知识内容。

3.2 浅析RISC的发展——以RISC-V为例

此部分请见**压缩包**中的**视频**。

4 结语

个人认为，计算机组成原理是计算机专业所有课程中最重要的一门。因为在本课程中，通过应用数字逻辑、离散数学的知识，可以设计一台模拟人类思维的机器，这为计算机领域的一切发展奠定了基础。

有人可能会问，学习这门课程有什么用呢？在我看来，对于芯片开发或者系统级部件开发而言，这部分知识可以为我们提供工程实践的基本理论支持，同时，该门课程也是整个计算机体系结构的基础，掌握好这门课程对于我们从事计算机软硬件方面的工作非常重要。即使将来从事纯软件开发，如果了解底层的运行原理、工作逻辑，可以设计出更适配的高质量代码，提升效率。

更重要的是，通过本课程的学习，可以掌握不少学习的方法。在计算机组成原理 A(1)课程学习过程中，通过同学间互相探讨，我逐步提高了发现问题、分析问题和解决问题的能力；通过研讨活动，提高了团队合作、沟通交流及表达能力；同时，通过撰写这篇报告，我也提高了文字编辑、排版等方面的能力。

普通人的记忆力是有限制的。大学本科四年学习的专业课，在多年后回想其中的知识点，可能是模糊不清的。但是在学习计算机组成原理课程（甚至其他专业课）中掌握的学习技巧、方法，会流淌在血液中，帮助我们提高各方面的素养，可以更快的重新拾起遗忘的知识，更好的学习其他知识。我想这正是学习的意义所在。

5 致谢

感谢余老师的认真教授，使我掌握了课程的知识点，并在课后耐心地解答我的困惑。感谢同学们对我的帮助。在大家的帮助下，顺利地完成了计算机组成原理 A(1)课程的学习，期待春季学期计算机组成原理 A(2)的学习！

参考文献

[1] 白中英, 戴志涛. 计算机组成原理[M]. 第六版·立体化教材. 北京: 科学出版社, 2019.

[2] 袁春风, 杨若瑜, 王帅, 唐杰. 计算机组成与系统结构[M]. 第2版. 北京: 清华大学出版社, 2015.

[3] Waterman A, Asanović K. The RISC-V Instruction Set Manual[S/OL], 2019-12-13. https://riscv.org/.

[4] 刘畅, 武延军, 吴敬征, 赵琛. RISC-V指令集架构研究综述[J]. 软件学报, 2021,32(12):3992-4024.

[5] 雷思磊. RISC-V架构的开源处理器及SoC研究综述[J]. 单片机与嵌入式系统应用, 2017,17(02):56-60+76.

附录 A 知识点思维导图

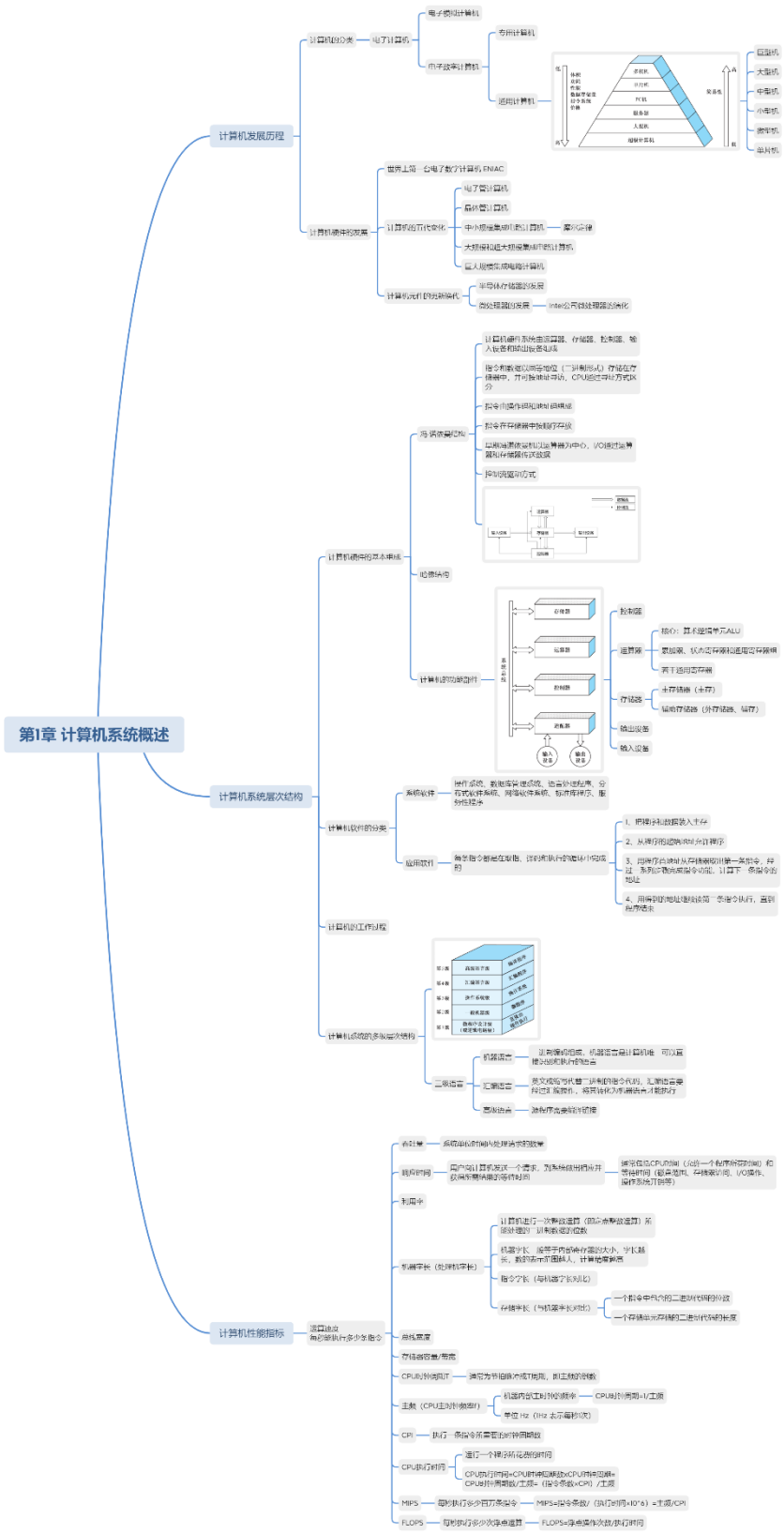

图 A.1 第 1 章计算机系统概述 知识点思维导图

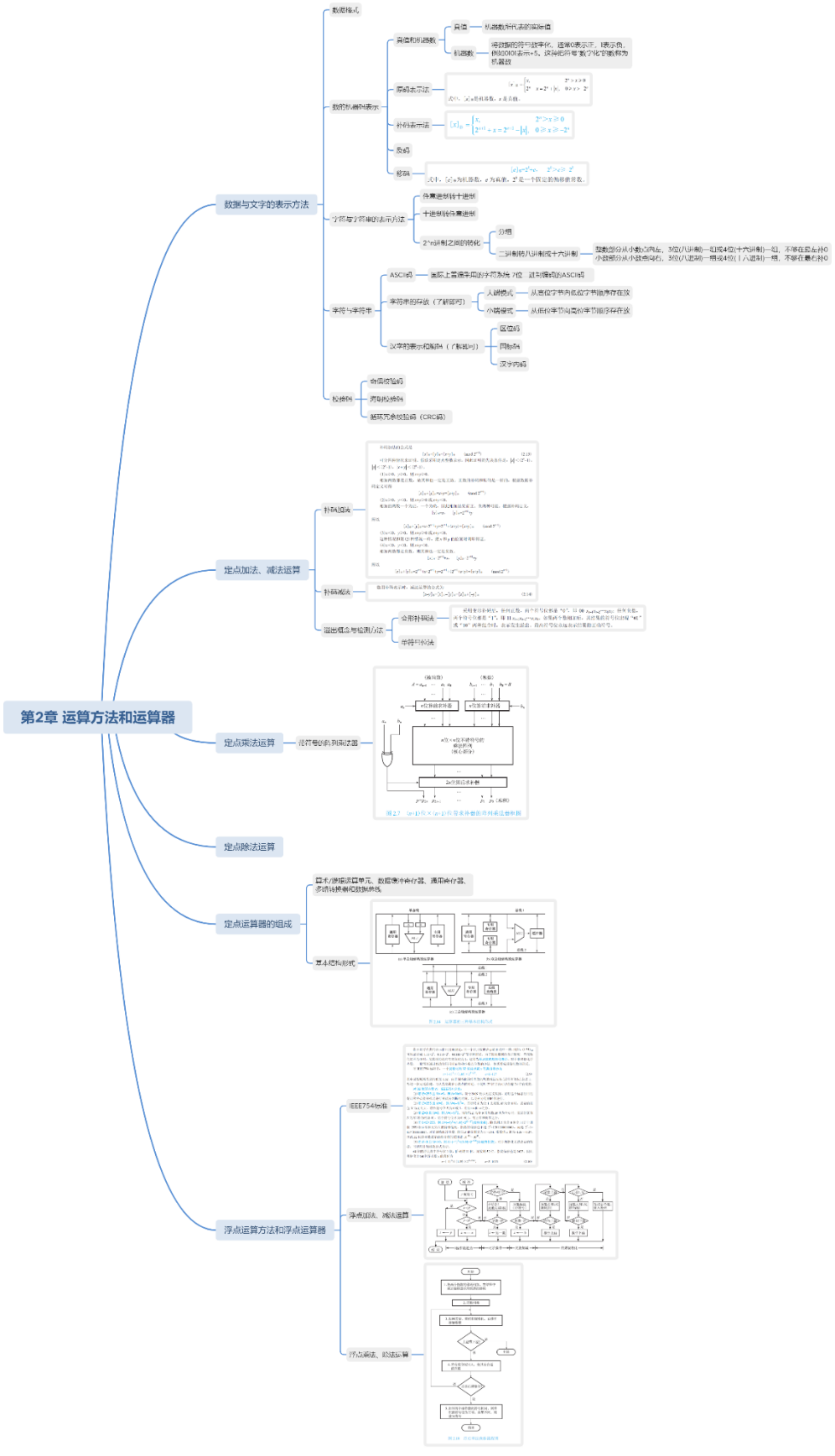

图 A.2 第 2 章运算方法和运算器 知识点思维导图

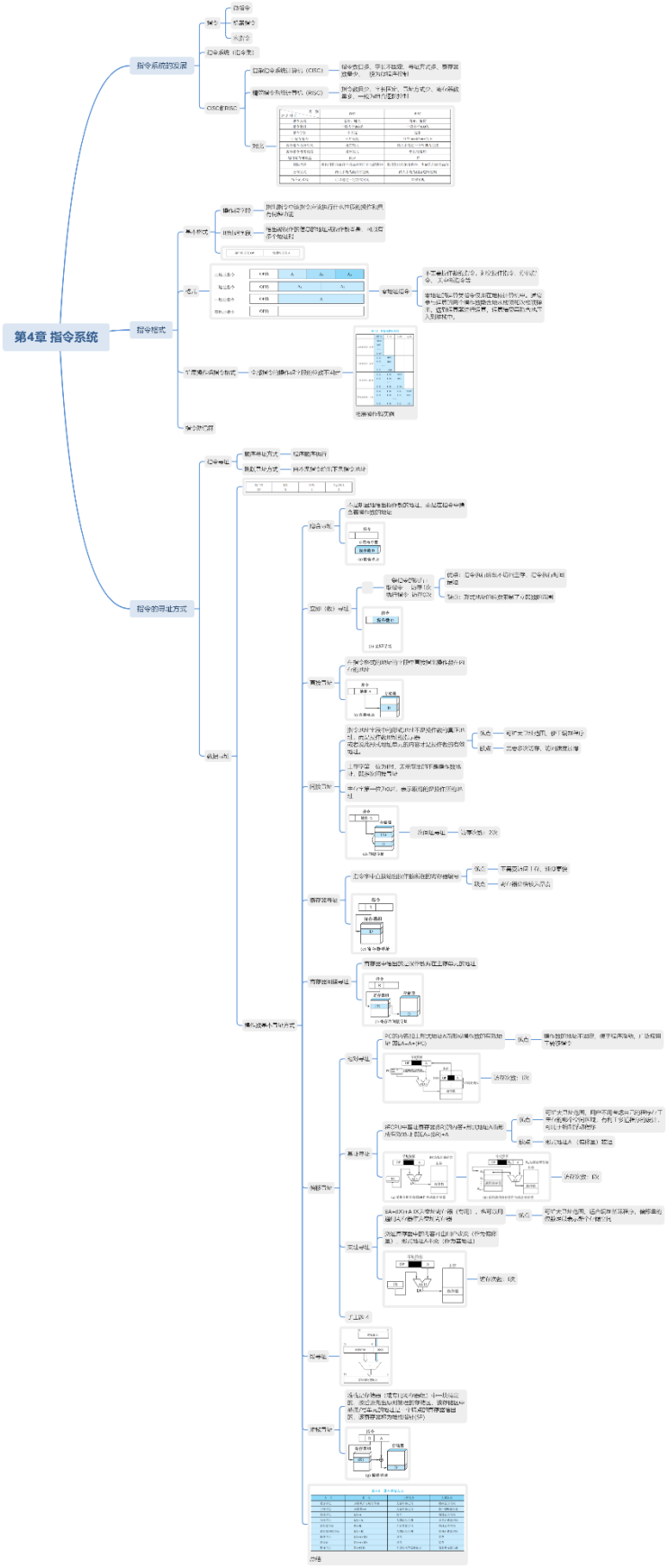

图 A.3 第 4 章指令系统 知识点思维导图

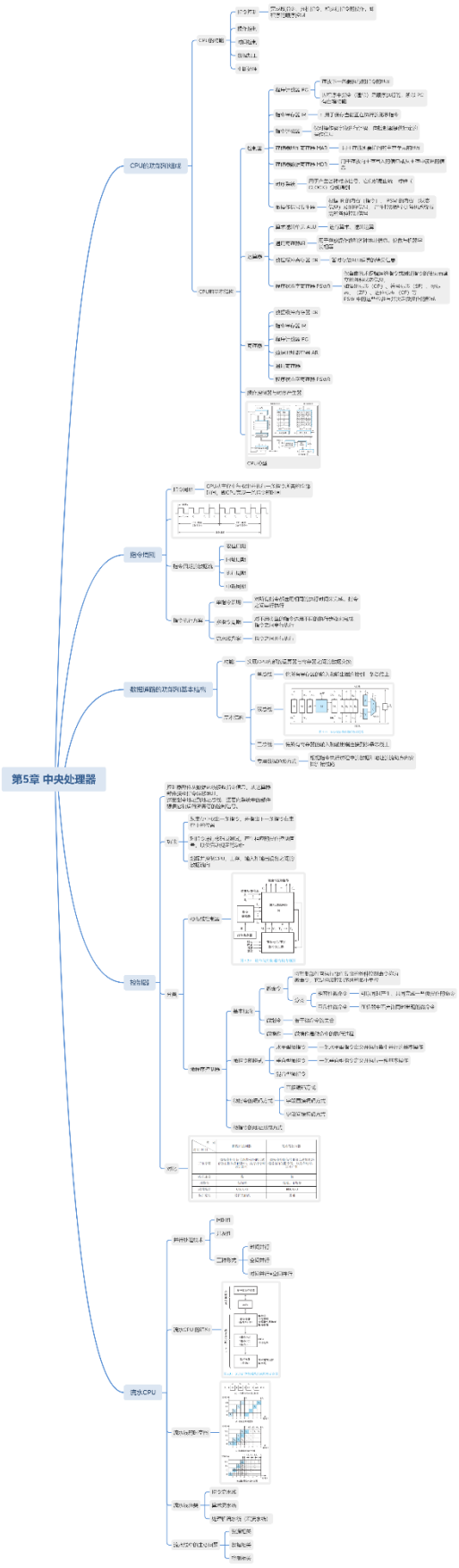

图 A.4 第 5 章中央处理器 知识点思维导图