

计算机系统概述

计算机系统层次结构

计算机发展历程

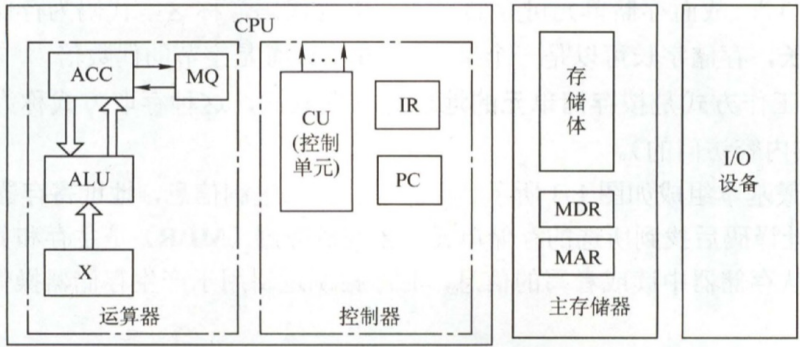
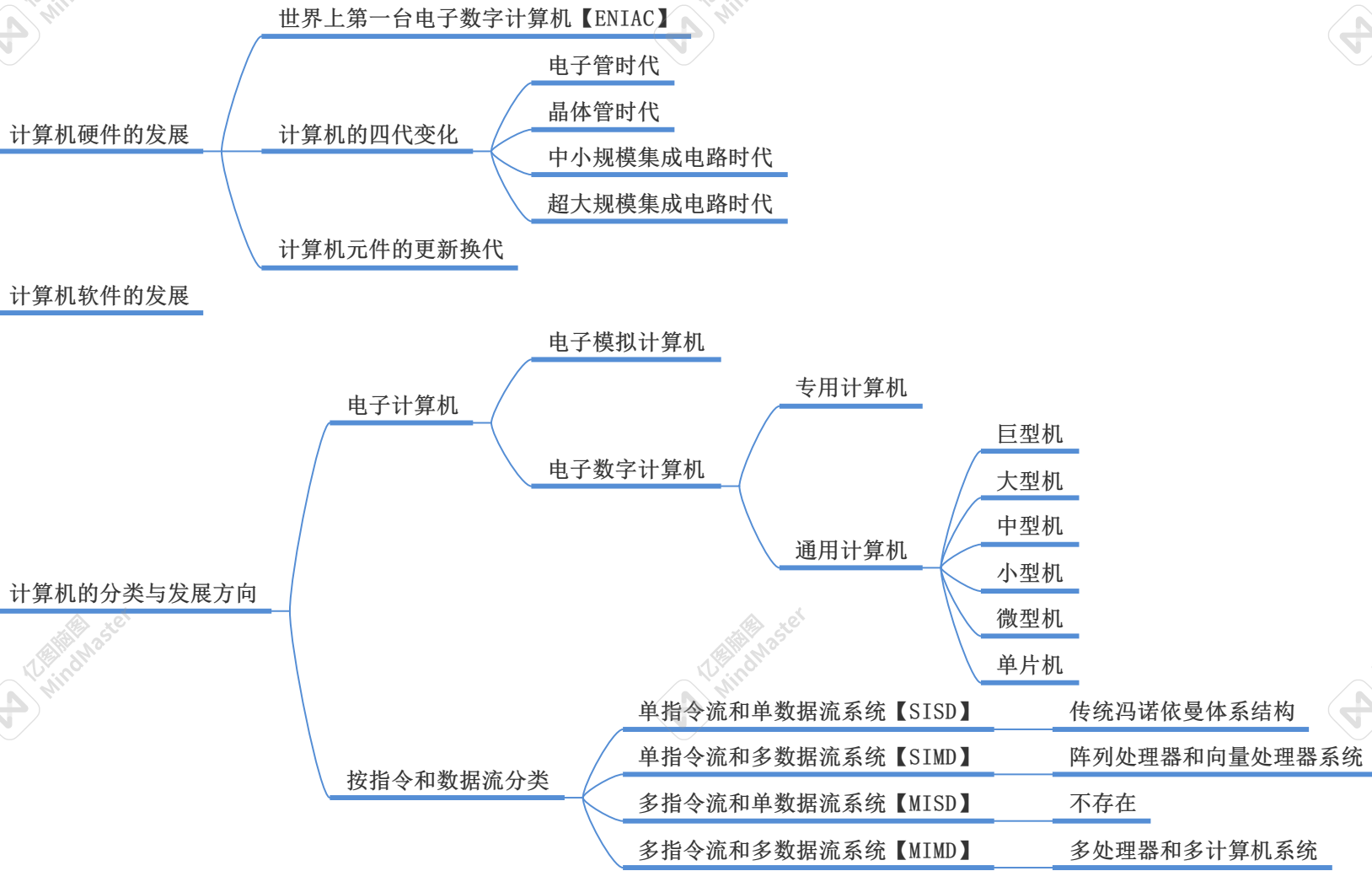


图 1-4 细化的计算机组成框图

计算机系统的组成

- 计算机硬件系统由运算器、存储器、控制器、输入设备和输出设备组成
- 指令和数据以同等地位（二进制形式）存储在存储器中，并按地址寻访，CPU通过寻址方式区分
- 指令由操作码和地址码组成
- 指令在存储器中按顺序存放
- 早期冯诺依曼机以运算器为中心，I/O通过运算器和存储器传送数据
- 控制流驱动方式

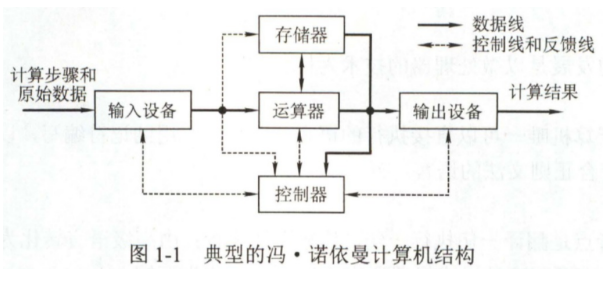


图 1-1 典型的冯·诺依曼计算机结构

计算机硬件的基本组成

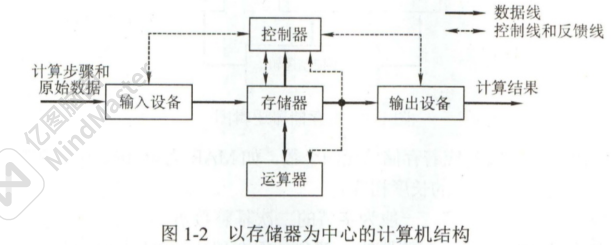


图 1-2 以存储器为中心的计算机结构

现代计算机的组织结构

输入设备

输出设备

存储器

运算器

控制器

计算机的功能部件

概念

主存储器（主存）

辅助存储器（外存储器、辅存）

核心：算术逻辑单元【ALU】

累加器、状态寄存器和通用寄存器组

若干通用寄存器

程序计数器【PC】

指令寄存器【IR】

控制单元【CU】

计算机的存储部件，存放程序和数据

CPU能直接访问

存储单元

存储体

地址寄存器【MAR】

数据寄存器【MDR】

时序控制逻辑

工作方式

外存的信息调入主存才能被CPU访问

位数对应存储单元个数

位数与存储字长相等

按存储单元的地址进程存取

存储一位二进制代码“0”或“1”

存储一串二进制代码，称为存储字，代码位数称为存储字长

存储字长是1B（8bit）或字节的偶数倍

10位代表有2^10个存储单元

可自动+1形成下一条指令地址，与主存的MAR之间有一条直接通路

存放当前的指令，内容来自主存的MDR

存放当前欲执行指令的地址，可自动+1形成下一条指令地址，与主存的MAR之间有一条直接通路

计算机软件的分类

系统软件

应用软件

1、把程序和数据装入主存

2、从程序的起始地址允许程序

3、用程序首地址从存储器取出第一条指令，经过一系列步骤完成指令功能，计算下一条指令的地址

4、用得到的地址继续读第二条指令执行，直到程序结束

每条指令都是在取指、译码和执行的循环中完成的

计算机的工作过程

计算机系统的多级层次结构

三个级别的语言

机器语言

汇编语言

高级语言

二进制编码组成，机器语言是计算机唯一可以直接识别和执行的语言

英文或缩写代替二进制的指令代码，汇编语言要经过汇编操作，将其转化为机器语言才能执行

C++/java/C，编译程序编译成汇编语言，汇编操作得到机器语言才能执行

用编译程序翻译成汇编语言程序

用汇编程序翻译成机器语言程序

用机器语言解释操作系统

软、硬件界面

传统机器M1（用机器语言的机器）

用微程序解释器指令

微程序机器M0（微指令系统）

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令

由硬件直接执行微指令

由微程序解释器指令