



1.7 主机：由CPU和主存所组成的计算机硬件

CPU：由ALU与CU组成的计算机硬件，全名中央处理器

主存：计算机存放程序与数据的硬件，是存储器。

存储单元：容量为1 byte ^{或者 byte}，具有地址的存储空间，^是 CPU访问的基本单元

存储元件：可存放1 bit信息的物理元件

存储基元 \Leftrightarrow 存储元 \Leftrightarrow 存储元件

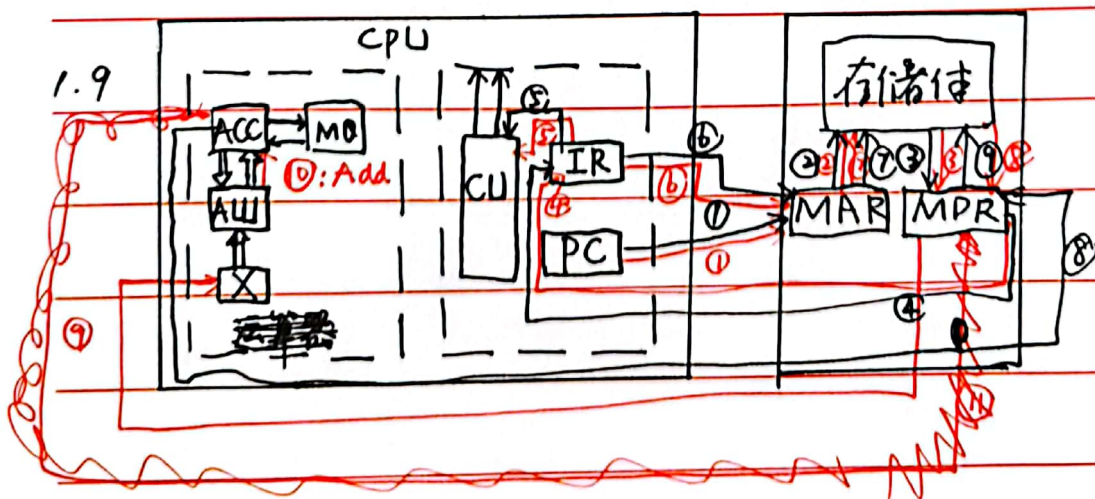
存储字：即 ~~字~~ 放在一个存储单元中的二进制代码组合

存储字长：一个存储单元中存储字位数

存储容量：存储器可存储二进制代码 ^{和数据} 总量

机器字长：CPU一次能处理的二进制位数

指令字长：机器指令中二进制代码总位数



IR, MDR, ACC, X 为32位

256M = 2^{28} bytes

MAR, PC 为28位





中国科学院大学

University of Chinese Academy of Sciences

$$y_{n+1} = \frac{1}{2}(y_n + \frac{x}{y_n})$$

1.10. $\sqrt{x} = \frac{1}{2}(y_n + \frac{x}{y_n})$, $y_0 = 1$ 精度 ϵ

程序如下:

| 主存单元地址 | 指令 | | 注释 |
|--------------|-------------------|-----------------------|--|
| | 操作码 | 地址码 | |
| 0 | 000001 | 0000001000 | 取数至ACC中 → 此行作废. |
| 0 | 000010 | 0000001000 | 取数至ACC中 |
| 1 | 000011 | 0000001000 | 除y得奇, 存于ACC中 |
| 2 | 000000 | 0000001000 | 加y得 $(y + \frac{x}{y})$, 存于ACC |
| 3 | 000001 | 0000001100 | 除2得 $\frac{1}{2}(y + \frac{x}{y})$, 存于ACC中 |
| 4 | 000110 | 0000001000 | 将ACC数存入 指定单元 指定单元 |
| 5 | 000111 | 0000001000 | 减去 ϵ 中数, 得 ϵ' 存入ACC |
| 6 | 001000 | 0000001000 | 与 ϵ 比较大小, 若大则跳到001000, 若小则跳到001001 |
| 7 | 001001 | 0000001000 | 将 $\frac{1}{2}(y + \frac{x}{y})$ 存入 指定单元 指定单元 |
| 8 | 001010 | | 停机 |
| 19 | 001000 | y_0 (初始值) | 原始数据, 同时存放结果 (是动态的) |
| 10 | 001001 | x | 原始数据 |
| 11 | | ϵ | 原始数据 |
| 12 | | 2 | 原始数据 |

操作码

操作性质

~~000010~~

取地址码指定单元中的数到指定单元

| | |
|--------|--|
| 000010 | 取地址码指定单元的数至ACC中 |
| 000011 | 将ACC数与指定单元数相除, 结果存入ACC |
| 000000 | 将ACC数与指定单元数相加, 结果存入ACC |
| 000100 | 将ACC中数存入指定单元 |
| 000111 | 将ACC中数与指定单元数相减, 结果存入ACC |
| 001000 | 将ACC中数与指定单元数比较大小, 若大则跳到001000, 若小则跳到001001 |
| 001001 | 存数至指定单元 |
| 001010 | 停机 |

(加载器、文件、内存)

计算机的解题过程: 高级程序语言 $\xrightarrow{\text{编译、汇编}}$ 机器语言 $\xrightarrow{\text{操作系统}}$ 进程

(CPU调度、进程管理)

(文件、I/O)

OS \rightarrow CPU 执行 \rightarrow Results.



扫描全能王 创建

$$MIPS = \frac{\text{Ins数}}{CPU \text{ Time} \times 10^6} = \frac{1}{\text{平均指令执行时间} \times 10^6}$$



中国科学院大学
University of Chinese Academy of Sciences

$$CPU \text{ Time} = \frac{CPI \times \text{Ins数}}{\text{主频}}$$

1.11 PC提供Address时是指令，指令地址码所取是数据

取指时段是指令，执行指令时是数据

$$9.8 \quad \text{平均指令执行时间} = 0.6 \times 0.35 + 0.8 \times 0.45 + 1.0 \times 0.05 + 1.4 \times 0.15 \mu s \\ = 1.28 \mu s$$

$$\Rightarrow \frac{MIPS}{\cancel{MIPS}} = \frac{1}{1.28} = 0.78 \quad \text{即运行速度.}$$

$$\text{由 } MIPS = \frac{\text{Clock Rate}}{CPI \times 10^6} \Rightarrow MIPS' = \frac{10}{6} \times MIPS = 1.3 \quad \text{即提高主频后运行速度}$$



扫描全能王 创建