

1.7

解释下列概念:

主机、CPU、主存、存储单元、存储元件、存储基元、存储元、存储字、存储字长、存储容量、机器字长、指令字长

主机 计算机用于放置主板及其他主要部件的容器。通常包括中央处理器、存储器和输入输出接口。

CPU 中央处理器, 主要功能是解释计算机指令以及处理计算机软件中的数据。

主存 主存储器, 与 CPU 直接交换数据的内部存储器, 用于存放数据和指令并能被 CPU 直接随机存取。

存储单元 用于存储数据的单位, 目前最基本的存储单元是字节。

存储元件 组成存储单元的物理事件。

存储基元 能存储一位二进制的存储元件。

存储元 存储器的最小存储单元, 一般是一位二进制。

存储字 存放在一个存储单元中的二进制代码组合, 数据的机器表示, 可以代表一个二进制数, 也可以代表一串字符。

存储字长 内存中存储字的长度。

存储容量 存储器能够存储数据的大小, 一般用字节或二进制位表示。

机器字长 中央处理器指令一次能处理的数据长度, 用二进制位表示。

指令字长 机器指令中二进制代码的位数。

1.8

解释下列英文缩写的含义:

CPU, PC, IR, CU, ALU, ACC, MQ, X, MAR, MDR, I/O, MIPS, CPI, FLOPS

CPU Central Processing Unit, 中央处理器, 主要包括 CU 和 ALU。

PC Program Counter, 程序计数器, 指向下一条将要执行的指令。

IR Instruction Register, 指令寄存器, CPU 里 CU 的一部分, 保存当前正在执行或解码的指令。

CU Control Unit, 控制单元 (控制器), CPU 的一部分, 它与计算机系统的输入输出设备合作获得所有操作的代码, 并通过时序和控制信号将操作传递给其他单元。

ALU Arithmetic Logic Unit, 算术逻辑单元 (运算器), CPU 的一部分, 进行二进制算术或逻辑运算。

ACC Accumulator, 累加器, CPU 中的一个用于存放加法操作数和运算结果的特殊用途寄存器。

MQ Multiplier-Quotient, 乘商寄存器, CPU 中的一个长度为两倍字长的用于存放乘法和除法运算结果的特殊用途寄存器.

X 操作数寄存器.

MAR Memory Address Register, 存储器地址寄存器, CPU 中的一个用于保存将要读取或写入数据的存储器地址的特殊用途寄存器.

MDR Memory Data Register, CPU 的控制单元中用于保存将要保存到存储器或刚从存储器中读取的数据的寄存器, 作用类似缓冲.

I/O Input/Output, 输入/输出, 计算机 (信息处理系统) 与外界 (如人或另一个信息处理系统) 的通讯.

MIPS Million Instructions per Second, 每秒百万条指令, 计算机运算速度的量化单位之一.

CPI Cycle per Instruction, 执行一条指令所需要的时钟周期数, 处理器性能的量化单位之一.

FLOPS Floating Point Operations per Second, 每秒浮点运算次数, 计算机浮点运算 (或科学计算) 执行效能的量化单位之一.

1.11

指令和数据都保存在存储器中, 计算机如何区分它们?

从二进制表示上 CPU 无法区分指令和数据.

区分指令和数据是程序员的工作, 自从计算机启动, 程序就应当保证对 CPU 的 PC 寄存器设置正确的值, 只指向机器指令代码而非数据.

现代操作系统采用页面方式处理主存储器内容, 并通过 “Executable”(可执行) 位来标记一个页是否属于程序指令.

1.5

冯·诺依曼计算机的特点是什么?

- 包括中心运算器, 中心控制器, 保存数据和指令的存储器, 外部存储器, 输入输出设备.
- 存储程序并执行, 将程序指令和数据放在读写随机存储器 (RAM) 中.
- 指令和数据用二进制码表示.

References

1. <http://baike.baidu.com/view/23880.htm>
2. <http://bsr1983.iteye.com/blog/942531>
3. <https://secure.wikimedia.org/wikipedia/zh/wiki/%E4%B8%AD%E5%A4%AE%E5%A4%84%E7%90%86%E5%99%A8>
4. <http://baike.baidu.com/view/1372797.htm>
5. <http://zh.wikipedia.org/zh-cn/%E4%B8%BB%E8%A8%98%E6%86%B6%E9%AB%94>
6. <http://baike.baidu.com/view/1223079.htm>

7. <http://baike.baidu.com/view/1846096.htm>
8. <http://baike.baidu.com/view/3403221.htm>
9. <http://baike.baidu.com/view/1338851.htm>
10. <http://baike.baidu.com/view/1094346.htm>
11. <http://encyclopedia2.thefreedictionary.com/stored+word>
12. <http://zhidao.baidu.com/question/74953580>
13. <http://baike.baidu.com/view/4277307.htm>
14. https://secure.wikimedia.org/wikipedia/en/wiki/Instruction_register
15. https://secure.wikimedia.org/wikipedia/en/wiki/Control_unit
16. https://secure.wikimedia.org/wikipedia/en/wiki/Arithmetic_logic_unit
17. <http://baike.baidu.com/view/331459.htm>
18. <http://www.answers.com/topic/multiplier-quotient-register>
19. https://secure.wikimedia.org/wikipedia/en/wiki/Memory_address_register
20. https://secure.wikimedia.org/wikipedia/en/wiki/Memory_data_register
21. <https://secure.wikimedia.org/wikipedia/en/wiki/Input/output>
22. https://secure.wikimedia.org/wikipedia/en/wiki/Million_instructions_per_second#Million_instructions_per_second
23. https://secure.wikimedia.org/wikipedia/en/wiki/Cycles_per_instruction
24. <https://secure.wikimedia.org/wikipedia/zh/wiki/FLOPS>
25. <http://stackoverflow.com/questions/2022489/how-instructions-are-differentiated-from-data>
26. <http://uk.answers.yahoo.com/question/index?qid=20110126112213AA51MWP>
27. https://secure.wikimedia.org/wikipedia/en/wiki/Von_Neumann_architecture