

计算机组成原理基本概念汇总

1. 冯·诺依曼思想：用**二进制表示程序和数据**；计算机采用**存储程序的工作方式**；计算机硬件由**存储器、运算器、控制器、输入设备和输出设备**组成。
2. 在计算机中用**数字代码**(二进制代码)表示**各种信息**。
3. 在物理机上用**数字信号**(数字型电信号)表示**数字代码**。
4. 用**电平信号**表示数字代码：**实现并行操作**。
5. 用**脉冲信号**表示数字代码：**实现串行操作**。
6. 存储程序工作方式：事先编制程序；事先存储程序；自动、连续地执行程序。
7. **CPU 由运算器、控制器、寄存器组成**。
8. ALU：通过加法器实现运算操作(由全加器求和、由进位链传递进位信号)。
9. 通用寄存器组：提供操作数，存放运算结果。
10. 选择器：选择操作数；选择控制条件，实现各种算法。
11. 移位器：直接或者移位送出运算结果。
12. **控制器：产生控制命令(微命令)，控制全机操作**。
13. 微命令产生方式：组合逻辑控制方式：由组合逻辑电路产生微命令；**微程序控制方式：由微指令产生微命令**。
14. 存储器：存储信息。
15. 存储体：存放信息的实体。
16. 寻址系统：对地址码译码，选择存储单元。
17. 读/写线路和数据寄存器：完成读/写操作，暂存读/写数据。
18. 控制线路：产生读/写时序，控制读/写操作。
19. 输入/输出设备：转换信息。输入：原始信息→代码，送入主机；输出：处理结果→人们所接受形式(代码)并输出。
20. 总线：能为多个部件分时共享的一组信息传送线路。功能：内总线、局部总线、系统总线、外总线。信息：地址总线、数据总线、控制总线。格式：并行总线、串行总线。时序：同步总线、异步总线。方向：单向总线、双向总线。
21. **基本字长：指操作数的基本位数，影响计算精度、指令功能**。
22. 主存容量：指存储单元个数(决定地址位数)*位数(表明编址单位)。
23. 外存容量：常表示为字节数，外存容量与地址码位数无关。
24. **运算速度：定点/浮点四则运算时间、每秒平均执行的指令条数(MIPS)、CPU 时钟频率(MHz)**。
25. 总线带宽=总线位数*总线时钟频率/8
26. 定点表示法：

类型

无符号数	00000000 (0)	11111111 (255)
定点整数	$\left\{ \begin{array}{l} 11111111_{\text{原}} \\ \quad (-127) \\ 10000000_{\text{补}} \\ \quad (-128) \end{array} \right.$	$\left\{ \begin{array}{l} 01111111_{\text{原}} \\ \quad (127) \\ 01111111_{\text{补}} \\ \quad (127) \end{array} \right.$
定点小数	$\left\{ \begin{array}{l} 1.1111111_{\text{原}} \\ \quad -(1-2^{-7}) \\ 1.0000000_{\text{补}} \\ \quad (-1) \end{array} \right.$	$\left\{ \begin{array}{l} 0.1111111_{\text{原}} \\ \quad (1-2^{-7}) \\ 0.1111111_{\text{补}} \\ \quad (1-2^{-7}) \end{array} \right.$

27. 浮点表示法:

浮点数真值: $N = \pm R^E \times M$

浮点数机器格式:

E_f	E_1	\dots	E_m	M_f	M_1	\dots	M_n
-------	-------	---------	-------	-------	-------	---------	-------

阶符
阶码
数符
尾数

R: 阶码底, 隐含约定。

E: 阶码, 为定点整数, 补码或移码表示。

其位数决定数值范围; 阶符表示数的大小。

M: 尾数, 为定点小数, 原码或补码表示。

其位数决定数的精度; 数符表示数的正负。

尾数规格化: $1/2 \leq |M| < 1$ 最高有效位绝对值为1

28. 浮点表示范围:

阶符1位, 阶码m位, 补码表示, 以2为底;

数符1位, 尾数n位, 补码表示, 规格化。

最小浮点数: $\begin{cases} \text{阶码为最大数: } 2^m - 1 \\ \text{尾数为绝对值最大的负数: } -1 \end{cases}$

最大浮点数: $\begin{cases} \text{阶码为最大数: } 2^m - 1 \\ \text{尾数为最大数: } 1 - 2^{-n} \end{cases}$

最小浮点正数: $\begin{cases} \text{阶码为最小数: } -2^m \\ \text{尾数为最小正数: } 2^{-1} \end{cases}$

29. 指令: 指示计算机执行某类操作的信息的集合。

30. 指令基本格式: 操作码+地址码。

31. 指令字长: 定长指令格式: 便于控制。变长指令格式: 合理利用存储空间。

32. 指令提供地址的方式: 显地址方式: 指令中明显指明地址(直接或间接给出); 隐地址方式: 地址隐含约定, 不出现在指令中。

33. 单独编址: 编址到寄存器: 为每个寄存器(I/O 端口)分配独立的端口地址; I/O 指令中给出端口地址。I/O 地址空间不占用主存空间, 可与主存空间重叠。

34. 统一编址: 编址到寄存器: 为每个寄存器(I/O 端口)分配总线地址; 访问外设时, 指令中给出总线地址。I/O 端口占据部分主存空间。常将存储空间的低端分配给主存单元, 高端分配给 I/O 端口, 以示区分。

35. 微命令发生器: 产生全机所需的各种微命令。分为电位型和脉冲型。

36. 微命令: 控制最基本的操作(微操作)的命令。

37. 指令计数器 PC: 指示指令在 M 中的位置。顺序执行: PC+1; 转移执行: PC 先加 1, 再用转移地址修改 PC。

38. 指令寄存器 IR: 存放现行指令。

39. 状态寄存器 PSW: 指示程序运行方式, 反映程序运行结果。

40. 优先级: 为现行程序赋予优先级别, 以决定是否响应外部中断请求。

41. 时序线路: 控制操作时间和操作时刻。

42. 总线周期长度可变, 时钟周期长度不变。

43. 若干微命令编制成一条微指令, 控制实现一步操作; 若干微指令组成一段微程序, 解释执行一条机器指令; 微程序事先存放在控制存储器中, 执行机器指令时再取出。

44. 控制存储器 CM: 存放微程序, CM 属于 CPU, 不属于主存储器。

45. 微指令寄存器 μIR : 存放现行微指令。
46. **垂直型微指令**: 一条微指令定义并执行一种基本操作。优点: **微指令短**、简单、规整, 便于编写微程序。缺点: **微程序长**, 执行速度慢; 工作效率低。
47. **水平型微指令**: 一条微指令定义并执行几种并行的基本操作。优点: **微程序短**, 执行速度快。缺点: **微指令长**, 编写微程序较麻烦。
48. 混合型微指令: 在垂直型的基础上增加一些不太复杂的并行操作。微指令不长, 便于编写; 微程序不长, 执行速度加快。
49. **同步控制, 用统一微指令周期控制各条微指令执行。**
50. **正数补码移位规则: 数符不变** (单: 符号位不变; 双: **第一符号位不变**), **空位补 0** (右移时**第二符号位移至尾数最高位**)。
51. **负数补码移位规则: 数符不变** (单: 符号位不变; 双: **第一符号位不变**), **左移空位补 0**, **右移空位补 1** (**第二符号位移至尾数最高位**)。
52. 原码一位乘法: 操作数、结果用原码表示; 绝对值运算, 符号单独处理; 被乘数(B)、累加和(A)取双符号位; 乘数末位(C_n)为判断位, 其状态决定下步操作; 作 n 次循环 (累加、右移)。
53. 补码一位乘法: A、B 取双符号位, 符号参加运算; C 取单符号位, 符号参加移位, 以决定最后是否修正; C 末位设置附加位 C_{n+1} , 初值为 0, $C_n C_{n+1}$ 组成判断位, 决定运算操作; 作 n 步循环, 若需作第 $n+1$ 步, 则不移位, 仅修正。
54. 定点除法运算: 若干余数与除数加减、移位。
55. 原码恢复余数法: A、B 双符号位, X、Y 绝对值, $|X| < |Y|$ 。运算结束后, 余数乘以 2^{-n} , 与被除数同号。
56. 原码不恢复余数法 (加减交替法): A、B 取双符号位, X、Y 取绝对值运算, $|X| < |Y|$ 。根据余数的正负决定商值及下一步操作。求 n 位商, 作 n 步操作; 若第 n 步余数为负, 则第 $n+1$ 步恢复余数, 不移位。
57. 补码不恢复余数法 (加减交替法): A、B 取双符号位, 符号参加运算, 并且 $|X| < |Y|$ 。根据余数与除数的符号决定商值及下一步操作。求 $n-1$ 位商, 作 n 步操作 (求出 r_n)。对商校正 (商符变反, 第 n 位商恒置 1)。
58. 浮点乘法运算: 检测操作数是否为 0。阶码相加。若阶码用移码表示, 相加后要修正。尾数相乘。相乘前不需对阶。结果规格化。一般左规。
59. 浮点除法运算: 检测操作数是否为 0。 $|AM| < |BM|$? 阶码相减。若阶码用移码表示, 相减后要修正。尾数相除。相除前不需对阶。结果不再规格化。
60. **静态存储器 SRAM 双极型、静态 MOS 型**: **依靠双稳态电路内部交叉反馈的机制存储信息**。功耗较大, 速度快, 作 Cache。
61. **动态存储器 DRAM (动态 MOS 型)**: **依靠电容存储电荷的原理存储信息**。功耗较小, 容量大, 速度较快, 作主存。
62. 存储空间分配: 先安排大容量芯片 (放地址低端), 再安排小容量芯片。
63. 低位地址分配给芯片, 高位地址形成片选逻辑。
64. **动态存储器的刷新: 定期向电容补充电荷。**
65. 存储密度: 面密度: 单位面积中存储的二进制信息量; 道密度: 单位长度上存储的二进制信息量。
66. 存储容量: 格式化容量; 非格式化容量。
67. **寻址时间: 平均寻道时间+平均等待时间。**
68. 数据传输率: 单位时间内从磁表面存储器所读/写的信息量。
69. 归零制 (RZ): 记录 1 时电流正向流动, 记录 0 时电流反向流动。在记录 2 个信息位之

间有一段距离没有电流变化。

70. 不归零制 (NRZ): 记录 1 时电流正向流动, 电流保持到下一个信息到来, 记录 0 时电流反向流动, 并电流保持到下一个信息到来。

71. 不归零-1 制 (NRZ1): 记录 1 时电流改变极性, 使磁记录层的磁化强度方向发生翻转; 记录 0 时保持原来的写电流和磁化强度方向。

72. 调相制 (PM): 在一个记录单元内, 磁头线圈中的写入电流由负到正表示记录信息 1, 由正到负表示记录信息 0。两者的相位相差 180 度。当二进制信息中出现连续两个 1 或连续 2 个 0 时, 为了维持上述规则, 在两个记录单元的交界处也要发生翻转。

73. 调频制 (FM): 在记录单元起始处不论是记录 0 还是 1, 都要改变电流方向, 产生翻转; 在一个记录单元中间点, 记录 1 时改变电流方向, 产生翻转, 记录 0 时不改变电流方向, 不产生翻转。这样记录 1 的频率是记录 0 的频率的 1 倍。

74. 改进型调频制 (MFM): 当二进制信息中出现连续 0 时, 其记录单元的交界处翻转一次。在其它情况下 (0→1, 1→0, 1→1) 其记录单元的交界处不翻转; 在一个记录单元的中间点, 记录 1 时改变电流方向, 产生磁化翻转, 记录 0 时不改变电流方向, 不产生磁化翻转。

75. 各道容量相同, 各道位密度不同, 内圈位密度最高。

76. 码距: 一种编码体制中, 各组合代码间的不同位数称距离, 其最小距离为该编码的码距。

77. 码距作用: 衡量一种编码查错与纠错的能力。

78. 奇偶校验码: 检测依据 (编码规则): 约定校验码中 1 的个数为奇数/偶数。可检测一位错, 不能纠错。用于主存校验。

79. 海明校验码: 检测依据: 多重奇偶校验。

80. 循环校验码 (CRC): 校验码能被某代码除尽。将有效信息与余数拼在一起形成校验码。出错, 余数不为 0。不同出错位对应不同余数。余数循环。

81. 主存: 主要存放 CPU 当前使用的程序和数据。(速度快、容量有限)

82. 辅存: 存放大量的后备程序和数据。(速度较慢、容量大)

83. 高速缓存: 存放 CPU 在当前一小段时间内多次使用的程序和数据。(速度很快、容量小)

84. 半导体存储器: 利用双稳态触发器存储信息 (动态存储器除外)。速度快, 非破坏性读出 (单管动态存储器除外), 信息易失 (只读存储器除外)。作主存、高速缓存。

85. 随机存取: 可按地址访问存储器中的任一单元访问时间与单元地址无关。

86. 随机存取存储器: RAM: 可读可写。ROM: 只读不写, 固存: 用户不能编程; PROM: 用户可一次编程; EPROM: 用户可多次编程 (紫外线擦除); EEPROM: 用户可多次编程 (电擦除)。速度指标: 存取周期或读/写周期 (ns)。

87. 顺序存取存储器 (SAM): 访问时读/写部件按顺序查找目标地址, 访问时间与数据位置有关。两步操作: 等待操作、读/写操作。速度指标: 平均等待时间、数据传输率。

88. 直接存取存储器 (DAM): 访问时读/写部件先直接指向一个小区域, 再在该区域内顺序查找。访问时间与数据位置有关。三步操作: 定位 (寻道) 操作、等待 (旋转) 操作、读/写操作。速度指标: 平均定位 (平均寻道) 时间 (ms)、平均等待 (平均旋转) 时间 (ms)、数据传输率 (位/秒)。

89. 并行接口: 接口与系统总线、接口与外设均按并行方式传送数据。数据各位同时传送。适用于设备本身并行工作, 距主机较近的情况。

90. 串行接口: 接口与系统总线并行传送, 接口与外设串行传送。数据逐位分时传送。适用于设备本身串行工作, 或距主机较远, 或需减少传送线的情况。

91. 同步接口: 接口与系统总线的信息传送由统一时序信号控制。

92. 异步接口: 接口与系统总线的信息传送采用异步应答方式。

93. 向量中断：由硬件提供服务程序入口地址。将服务程序入口(中断向量)组织在中断向量表中；CPU 响应时由硬件直接产生相应向量地址，按地址查表，取得服务程序入口，转入相应服务程序。

94. 中断向量：服务程序入口地址、服务程序状态字。

95. 中断向量表：存放中断向量的表一段存储区)。

96. 向量地址：访问向量表的地址（指向中断向量的首址）。

97. 单级中断：CPU 响应后只处理一个中断源的请求，处理完毕后才能响应新的请求。

98. 多重中断：在某次中断服务过程中，允许响应处理更高级别的中断请求。