

日期 2017.7.4

01章 计算机系统概论

5大部件: ① 运算器 ② 存储器 ③ 控制器 ④ 输入设备 ⑤ 输出设备

MAR 地址寄存器: 对应存储单元个数, 存放指令地址 10位 MAR $2^{10} = 1024$ 个单元

MDR 数据寄存器: 基位数和存储字长相容

冯诺依曼核心: 存储程序

取数指令流程 (将指令地址码指示的存储单元中的操作数取出后送至运算器 ACC 中)

取指令: $PC \rightarrow MAR \rightarrow M \rightarrow MDR \rightarrow IR$

分析指令: $OP(IR) \rightarrow CU$

执行指令: $Ad(IR) \rightarrow MAR \rightarrow M \rightarrow MDR \rightarrow ACC$

$(PC) + 1 \rightarrow PC$

在 CPU 中, IR, MAR, MDR 对各类程序员透明

02章 运算方法和运算器

原码, 补码, 反码, 移码

原码表示法

$$\text{定点小数 } [x]_{\text{原}} = \begin{cases} x & 1 > x \geq 0 \\ 1 - x = 1 + |x| & 0 > x > -1 \end{cases}$$

$$\text{定点整数 } [x]_{\text{原}} = \begin{cases} x & 2^n > x \geq 0 \\ 2^n - x = 2^n + |x| & 0 > x > -2^n \end{cases}$$

补码表示法

$$\text{定点小数 } [x]_{\text{补}} = \begin{cases} x & 1 > x \geq 0 \\ 2 + x = 2 - |x| & 0 > x > -1 \end{cases} \pmod{2}$$

$$\text{定点整数 } [x]_{\text{补}} = \begin{cases} x & 2^n > x \geq 0 \\ 2^{n+1} + x = 2^{n+1} - |x| & 0 > x > -2^n \end{cases} \pmod{2^{n+1}}$$

反码表示法

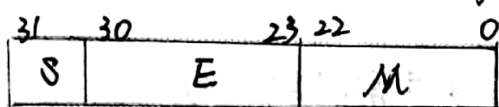
定点小数 $[x]_R = \begin{cases} x & 1 > x > 0 \\ (2-2^{-n}) - x = 1 + |x| & 0 \geq x > -1 \end{cases}$

定点整数 $[x]_R = \begin{cases} x & 2^n > x \geq 0 \\ (2^{n+1}-1) - x & 0 \geq x > -2^n \end{cases}$

移码表示法

IEEE 中阶码字段8位 $[e]_{移} = 127 + e$

IEEE 标准



S 数符 E 阶码 M 尾数

单精度(32位) single

$x = (-1)^S \times (1.M) \times 2^{E-127}$ $e = E - 127$

63	62	52	51	0
S	E			
				M

双精度(64位) double

$x = (-1)^S \times (1.M) \times 2^{E-1023}$ $e = E - 1023$

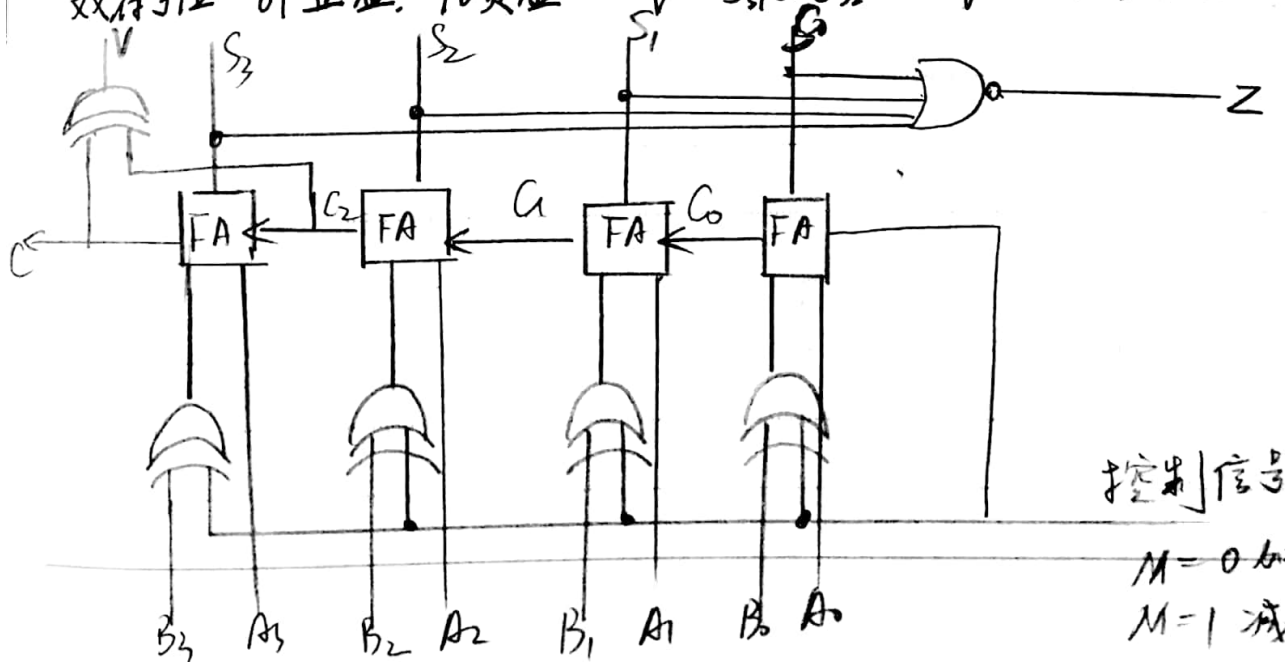
正规格化数 S:0 E:0 < E < 255 M 真值 $2^{E-127} (1.M)$

负规格化数 S:1 E:0 < E < 255 M 真值 $-2^{E-127} (1.M)$

$2^{-26} (1.0) \sim 2^{127} (1.111\dots)$ $[-(1-2^{-24}) \times 2^{128}, -1 \times 2^{126}]$
 $2^{-2^{-23}}$ $[1 \times 2^{-126}, (1-2^{-24}) \times 2^{128}]$

8位有符号整数中, $[-128]_{补} = 1000\ 0000$, 但其原码不存在

双符号位 01 正溢, 10 负溢 $V = S_{s1} \oplus S_{s2}$ $V=0$ 无溢, $V=1$ 溢



$[x]_{补} = 0.101$, $[y]_{补} = 1.001$ 均为定点纯小数

(1) 采用双符号位法算 $x-y$, 判是否溢出

$[x]_{变形补码} = 00.101$ $[y]_{变形补码} = 11.001$

$$x-y = [x]_{补} + [-y]_{补}$$

00.101

两个符号位相反, 故发生溢出

$$\begin{array}{r} 00.101 \\ + 00.111 \\ \hline 01.100 \end{array}$$

(2) 直接补码法算 $x \times y$

(直接补码法就是有括号的那种)

(0).101

(1).001
(0).101

$$\begin{aligned} x \times y &= 0.000101 - 0.101 \\ &= -0.100011 \end{aligned}$$

(0) 000

(0) 000

0 (1) (0) (1)

0 (1) 0 (1) 101

(3) 加减交替法求 x/y

$[x]_{补} = 0.101$

$[y]_{补} = 0.111$

$[-y]_{补} = 1.001$

00.101

11.001 减

11110 < 0 商 0

11100 左移

+00.111

00011 > 0 商 1

00110 左移

+11001

11111 < 0 商 0

11110 左移

+00.111

00101 > 0 商 1

∴ 商 0.101

余数 0.000101

03章 存储系统

* 浮点的加减运算

① 对阶 (小阶向大阶看齐) ② 尾数求和 ③ 规格化 (溢出时右规, 尾数右移, 阶加1)

④ 舍入 ⑤ 溢出判断 (双符号位补码表示7位)

eg. $x = (13/16) \times 2^6$, $y = (23/32) \times 2^5$, 求 $x+y$ (双符号位补码表示5位阶码)

解: $x = 1101 \times 2^{-4} \times 2^6 = 1101 \times 2^2 = 0.1101 \times 2^6$ 00110 0011010

$y = 10111 \times 2^{-5} \times 2^5 = 10111 = 0.10111 \times 2^5$ 00101 0010111

对阶: $E_x - E_y = 00110 + 11011 = 00001 > 0$ y 向 x 看齐

$y = 0.010111 \times 2^6$ 00110 0001011(1)

$E = E_x = 00110$

尾数相加: $M = 00.11010 + 00.01011(1)$

$= 01.00101(1)$

规格化: 溢出需右规, 阶码加1, 右规 $M = 00.10010(1)(1)$

按“0舍1入”规则, 得 $M = 00.10011$

阶码 $E = 00110 + 1 = 00111$

机器数: 001110010011 $x+y = (19/32) \times 2^7 = 76$

$G_i = A_i B_i$, $P_i = A_i \oplus B_i$

① 串行进位 $G_1 = G_1 + P_1 G_0 = A_1 B_1 + (A_1 \oplus B_1) G_0$

$G_2 = G_2 + P_2 G_1 = A_2 B_2 + (A_2 \oplus B_2) G_1$

$G_n = G_n + P_n G_{n-1}$

② 并行进位 $G_1 = G_1 + P_1 G_0$

$G_2 = G_2 + P_2 G_1 = G_2 + P_2 (G_1 + P_1 G_0) = G_2 + P_2 G_1 + P_2 P_1 G_0$

$G_3 = G_3 + P_3 G_2 =$ 代进去算

....

③ 组内并行, 组间 $\left\{ \begin{array}{l} \text{串行} \\ \text{并行} \end{array} \right. G_8, G_{12}, G_{16}$

03章存储系统

BUSY_L变为低电平 \Rightarrow 右 \checkmark 算装波那契, 迭代占内存最少, 递归占内存最多
left

"全相联映像"较多用"按内容寻址"的相联存储器实现

段页式虚拟存储器的段表表目里必须存放:

该段所对应页表在内存中的起始位置

段页式虚拟存储器, 虚存与主存之间交换数据的单位 页

集中式刷新, 刷新时间 64×64 阵列, DRAM 存储周期 $200ns = 0.2\mu s$ 刷新间隔 $2ms$

\Rightarrow 对于 64×64 , 要在 64 个存取周期刷新 64 行, 存储周期为 $0.2\mu s$, $t_{总} = 64 \times 0.2\mu s$

求顺序存储器与交叉存储器带宽:

$$q = \text{字长} \times \text{模块数} = ? \text{ bit}$$

$$\text{顺序存储器: } t_1 = mT$$

$$\text{带宽} = W_1 = \frac{q}{t_1} = \frac{? \text{ bit}}{? s} = \dots \text{ MB/s}$$

$$\text{交叉存储器: } t_2 = T + (m-1)\tau$$

$$\text{带宽} = W_2 = \frac{q}{t_2} = \frac{? \text{ bit}}{? s} = \dots \text{ MB/s}$$

刷新方式:

① 集中式刷新: 每月月底腾出时间专门洗衣服

② 分散式刷新: 每天把今天穿脏的衣服洗一洗

③ 异步式刷新: 这一个月里, 每个周末专门有个时间洗衣服

计算:

$$\text{Cache 命中率 } h = \frac{N_c}{N_c + N_m}$$

$$\text{失效率 } M = 1 - H \text{ (未命中)}$$

$$\text{平均访问时间 } t_a = h t_c + (1-h) t_m \text{ (默认 Cache 与主存同时启动)}$$

$$\text{访问效率 } e = \frac{t_{\text{cache}}}{t_{\text{average}}} = \frac{t_c}{h t_c + (1-h) t_m} = \frac{1}{h + (1-h) \left(\frac{t_m}{t_c} \right)} = \frac{1}{r + (1-h)r}$$

$$\therefore e = \frac{1}{r + (1-h)r} \text{ 即 } \frac{1}{5 + (1-5) \times 0.95} = \frac{1}{5 - 4 \times 0.95} = 83.3\%$$

Cache 与主存的映射方式

- 直接映射
- 全相联映射
- 组相联映射

① 直接映射 (指定行)

tag	行号	字地址
-----	----	-----

$$\text{Cache 行号 } i = \underbrace{\text{内存块号 } j}_{\uparrow} \bmod \text{Cache 中总行数}$$

内存单元地址 / 块大小的字节数

 得到第 n 块

$$\text{Cache 行号} = \text{第 } n \text{ 块内存} / \text{Cache 一共多少行}$$

② 全相联映射: (连内容连地址整个抄过来, 随便放)

tag 为块标记, 即第 n 块内存
(一般从 0 开始编号)

③ 组相联映射:

$$\text{Cache 组号} = \text{内存块号} \bmod \text{Cache 中总组数}$$

在这个组中的任一行 ↑

V 路组相联: 将 Cache 分成 u 组 每组 v 行 Cache 行数 / 每组多少行

$$\text{Cache 总行数: } m = u \times v$$

$$\text{Cache 组号 } q: \text{内存块号 } j \bmod u$$

内存地址格式

tag	组号	字地址
-----	----	-----

页式虚拟存储器: 以页为基本单位 (最后一页的馒头浪费)

段式虚拟存储器: 段按程序的逻辑结构划分

段页式虚拟存储器: 把程序按逻辑结构分段, 每段再划分为固定大小的页
(主存空间也划分为大小相等的页)

快表 TLB

04 章 指令系统

直接寻址 $E = D$

是基址后间接寻址 $E = ((R_n) + D)$

相对寻址 $E = (PC) + D$

变址寻址 $E = (R) + D$

基址寻址 $E = (R_n) + D$

间接寻址 $E = (D)$

05章 中央处理器

RISC一定采用指令流水线技术

单地址双操作数, 常一个指明, 另一个隐含寻址

{ 指令寻址: 确定程序的执行顺序

{ 数据寻址: 确定操作数地址

寄存器间接寻址可简化地址结构

PC对汇编语言程序员可见, 而MAR, MDR对程序员不可见

各种寻址方式有效地址的计算表达式

- 立即寻址 $(EA) = D$
- 直接寻址 $EA = D$
- 间接寻址 $EA = (D)$
- 变址寻址 $EA = (Ri) + D$
- 基址寻址 $EA = (Rb) + D$
- 相对寻址 $EA = (PC) + D$

