

4.2.6 存储器的校验

在计算机运行过程中,由于种种原因致使数据在存储过程中可能出现差错。为了能及时发现错误并及时纠正错误,通常可将原数据配成汉明编码。

1. 汉明码的组成

汉明码是由 Richard Hanming 于 1950 年提出的,它具有一位纠错能力。

由编码纠错理论得知,任何一种编码是否具有检测能力和纠错能力,都与编码的最小距离有关。所谓编码最小距离,是指在一种编码系统中,任意两组合法代码之间的最少二进制位数的差异。

根据纠错理论得

$$L-1=D+C \quad \text{且 } D \geq C$$

即编码最小距离 L 越大,则其检测错误的位数 D 越大,纠正错误的位数 C 也越大,且纠错能力恒小于或等于检错能力。例如,当编码最小距离 $L=3$ 时,这种编码可视为最多能检错二位,或能检错一位、纠错一位。可见,倘若能在信息编码中增加若干位检测位,增大 L ,显然便能提高检错和纠错能力。汉明码就是根据这一理论提出的具有一位纠错能力的编码。

设欲检测的二进制代码为 n 位,为使其具有纠错能力,需增添 k 位检测位,组成 $n+k$ 位的代码。为了能准确对错误定位以及指出代码没错,新增添的检测位数 k 应满足:

$$2^k \geq n+k+1$$

由此关系可求得不同代码长度 n 所需检测位的位数 k ,如表 4.2 所示。

表 4.2 代码长度与检测位位数的关系

n	k (最小)
1	2
2~4	3
5~11	4
12~26	5
27~57	6
58~120	7

k 的位数确定后,便可由它们所承担的检测任务设定它们在被传送代码中的位置及它们的取值。

设 $n+k$ 位代码自左至右依次编为第 $1, 2, 3, \dots, n+k$ 位,而将 k 位检测位记作 $C_i (i=1, 2, 4, 8, \dots)$,分别安插在 $n+k$ 位代码编号的第 $1, 2, 4, 8, \dots, 2^{k-1}$ 位上。这些检测位的位置设置是为了保证它们能分别承担 $n+k$ 位信息中不同数位所组成的“小组”的奇偶检测任务,使检测位和它所负责检测的小组中 1 的个数为奇数或偶数,具体分配如下:

C_1 检测的 g_1 小组包含 $1, 3, 5, 7, 9, 11, \dots$ 位。

C_2 检测的 g_2 小组包含 $2, 3, 6, 7, 10, 11, 14, 15, \dots$ 位。

C_4 检测的 g_3 小组包含 4,5,6,7,12,13,14,15,⋯位。

C_8 检测的 g_4 小组包含 8,9,10,11,12,13,14,15,24,⋯位。

⋮

其余检测位的小组所包含的位也可类推。这种小组的划分有如下特点：

① 每个小组 g_i 有一位且仅有一位为它所独占，这一位是其他小组所没有的，即 g_i 小组独占第 2^{i-1} 位 ($i=1,2,3,\cdots$)。

② 每两个小组 g_i 和 g_j 共同占有一位是其他小组没有的，即每两小组 g_i 和 g_j 共同占有第 $2^{i-1}+2^{j-1}$ 位 ($i,j=1,2,\cdots$)。

③ 每 3 个小组 g_i 、 g_j 和 g_l 共同占有第 $2^{i-1}+2^{j-1}+2^{l-1}$ 位，是其他小组所没有的。

依次类推，便可确定每组所包含的各位。

例如，欲传递信息为 $b_4b_3b_2b_1$ ($n=4$)，根据 $2^k \geq n+k+1$ ，可求出配置成汉明码需增添检测位 $k=3$ ，且它们位置的安排如下：

二进制序号	1	2	3	4	5	6	7
名称	C_1	C_2	b_4	C_4	b_3	b_2	b_1

如果按配偶原则来配置汉明码，则 C_1 应使 1、3、5、7 位中的“1”的个数为偶数； C_2 应使 2、3、6、7 位中的“1”的个数为偶数； C_4 应使 4、5、6、7 位中的“1”的个数为偶数。

故 C_1 应为 3 位 \oplus 5 位 \oplus 7 位，即 $C_1=b_4 \oplus b_3 \oplus b_1$ ； C_2 应为 3 位 \oplus 6 位 \oplus 7 位，即 $C_2=b_4 \oplus b_2 \oplus b_1$ ； C_4 应为 5 位 \oplus 6 位 \oplus 7 位，即 $C_4=b_3 \oplus b_2 \oplus b_1$ 。

令 $b_4b_3b_2b_1=0101$ ，则

$$C_1=b_4 \oplus b_3 \oplus b_1=0 \oplus 1 \oplus 1=0$$

$$C_2=b_4 \oplus b_2 \oplus b_1=0 \oplus 0 \oplus 1=1$$

$$C_4=b_3 \oplus b_2 \oplus b_1=1 \oplus 0 \oplus 1=0$$

故 0101 的汉明码应为 $C_1C_2b_4b_3b_2b_1$ ，即 0100101。

2. 汉明码的纠错过程

汉明码的纠错过程实际上是对传送后的汉明码形成新的检测位 P_i ($i=1,2,4,8,\cdots$)，根据 P_i 的状态，便可直接指出错误的位置。 P_i 的状态是由原检测位 C_i 及其所在小组内“1”的个数确定的。倘若按配偶原则配置的汉明码，其传送后形成新的检测位 P_i 应为 0，否则说明传送有错，并且还可直接指出出错的位置。由于 P_i 与 C_i 有对应关系，故 P_i 可由下式确定：

$$P_1=1 \oplus 3 \oplus 5 \oplus 7, \text{ 即 } P_1=C_1 \oplus b_4 \oplus b_3 \oplus b_1$$

$$P_2=2 \oplus 3 \oplus 6 \oplus 7, \text{ 即 } P_2=C_2 \oplus b_4 \oplus b_2 \oplus b_1$$

$$P_4=4 \oplus 5 \oplus 6 \oplus 7, \text{ 即 } P_4=C_4 \oplus b_3 \oplus b_2 \oplus b_1$$

设已知传送的正确汉明码（按配偶原则配置）为 0100101，若传送后接收到的汉明码为 0100111，其出错位可按下述步骤确定。

令

二进制序号	1	2	3	4	5	6	7
正确的汉明码	0	1	0	0	1	0	1
接收到的汉明码	0	1	0	0	1	1	1

则新的检测位为

$$P_4 = 4 \oplus 5 \oplus 6 \oplus 7, \text{ 即 } P_4 = 0 \oplus 1 \oplus 1 \oplus 1 = 1$$

$$P_2 = 2 \oplus 3 \oplus 6 \oplus 7, \text{ 即 } P_2 = 1 \oplus 0 \oplus 1 \oplus 1 = 1$$

$$P_1 = 1 \oplus 3 \oplus 5 \oplus 7, \text{ 即 } P_1 = 0 \oplus 0 \oplus 1 \oplus 1 = 0$$

由此可见, 传送结果 P_4 和 P_2 均不呈偶数, 显然出了差错。那么, 错在哪一位呢? 仔细分析发现, 只有第 6 位出错才会同时使 P_4 和 P_2 不呈偶数。同时, P_4 、 P_2 、 P_1 所构成的二进制值恰恰是出错的位置, 即 $P_4P_2P_1 = 110$, 表示第 6 位出错。发现错误后, 计算机便自动将错误的第 6 位“1”纠正为“0”。

又如, 若收到按偶配置的汉明码为 1100101, 则经检测得

$$P_4 = 4 \oplus 5 \oplus 6 \oplus 7, \text{ 即 } P_4 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$$

$$P_2 = 2 \oplus 3 \oplus 6 \oplus 7, \text{ 即 } P_2 = 1 \oplus 0 \oplus 0 \oplus 1 = 0$$

$$P_1 = 1 \oplus 3 \oplus 5 \oplus 7, \text{ 即 } P_1 = 1 \oplus 0 \oplus 1 \oplus 1 = 1$$

即 $P_4P_2P_1 = 001$, 表示第 1 位出错。由于第 1 位不是欲传送的信息位, 而是检测位, 而检测位不参与运算, 故在一般情况下可以不予纠正。

以上均以 $n=4$ 为例, 其实对任意不同 n 位的信息, 均可按上述步骤配置汉明码, 即先求出需增加的检测位位数 k , 再确定 C_i 的位置, 然后, 按奇或偶原则配置 C_i 各位的值即可。值得注意的是: 按奇配置与按偶配置所求得的 C_i 值正好相反, 而新的检测位 P_i 的取值与奇偶配置原则是相对应的, 读者可自行分析。

汉明码常常被用在纠错一位的场合, 若欲实现检错两位, 实用时还得再增添一位检测位。

例 4.4 已知接收到的汉明码为 0110101 (按配偶原则配置), 试问欲传送的信息是什么?

解: 由于要求出欲传送的信息, 必须是正确的信息, 因此不能简单地从接收到的 7 位汉明码中去掉 C_1 、 C_2 、 C_4 这 3 位检测位来求得。首先应该判断收到的信息是否出错。纠错过程如下:

$$P_1 = 1 \oplus 3 \oplus 5 \oplus 7 = 1$$

$$P_2 = 2 \oplus 3 \oplus 6 \oplus 7 = 1$$

$$P_4 = 4 \oplus 5 \oplus 6 \oplus 7 = 0$$

所以, $P_4P_2P_1 = 011$, 第 3 位出错, 可纠正为 0100101, 故欲传送的信息为 0101。

例 4.5 按配奇原则配置 1100101 的汉明码。

解: 根据 1100101, 得 $n=7$ 。根据 $2^k \geq n+k+1$, 可求出需增添 $k=4$ 位检测位, 各位的安排如下:

二进制序号	1	2	3	4	5	6	7	8	9	10	11
汉明码	C_1	C_2	1	C_4	1	0	0	C_8	1	0	1

按配奇原则配置,则

$$C_1 = 3 \oplus 5 \oplus 7 \oplus 9 \oplus 11 = 1$$

$$C_2 = 3 \oplus 6 \oplus 7 \oplus 10 \oplus 11 = 1$$

$$C_4 = 5 \oplus 6 \oplus 7 = 0$$

$$C_8 = 9 \oplus 10 \oplus 11 = 1$$

故新配置的汉明码为 11101001101。

4.2.7 提高访存速度的措施

随着计算机应用领域的不断扩大,处理的信息量越来越多,对存储器的工作速度和容量要求也越来越高。此外,因 CPU 的功能不断增强,I/O 设备的数量不断增多,致使主存的存取速度已成为计算机系统的瓶颈。可见,提高访存速度也成为迫不及待的任务。为了解决此问题,除了寻找高速元件和采用层次结构以外,调整主存的结构也可提高访存速度。

1. 单体多字系统

由于程序和数据在存储体内是连续存放的,因此 CPU 访存取出的信息也是连续的,如果可以在一个存取周期内,从同一地址取出 4 条指令,然后再逐条将指令送至 CPU 执行,即每隔 1/4 存取周期,主存向 CPU 送一条指令,这样显然增大了存储器的带宽,提高了单体存储器的工作速度,如图 4.41 所示。

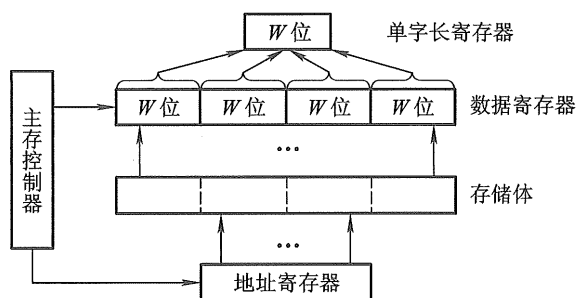


图 4.41 单体四字结构存储器

图中示意了一个单体四字结构的存储器,每字 W 位。按地址在一个存取周期内可读出 $4 \times W$ 位的指令或数据,使主存带宽提高到 4 倍。显然,采用这种办法的前提是:指令和数据在主存内必须是连续存放的,一旦遇到转移指令,或者操作数不能连续存放,这种方法的效果就不明显。