

计算机组织结构

10 数据校验码

刘博涵

2023年11月9日



南京大學
NANJING UNIVERSITY

教材对应章节



第2章 数据的机器级表示



第5章 内部存储器

差错 (Error)

- **数据**在计算机内部进行计算、存取和传送过程中，由于**元器件故障或噪音干扰**等原因，会出现**差错**
- **以存储为例**
 - **硬故障 (hard failure)**：永久性的物理故障，以至于受影响的存储单元不能可靠地存储数据，成为固定的“1”或“0”故障，或者在0和1之间不稳定地跳变
 - 由恶劣的环境、制造缺陷和旧损引起
 - **软故障 (soft error)**：随机非破坏性事件，它改变了某个或某些存储单元的内容，但没有损坏机器
 - 由电源问题或 α 粒子引起
- **解决方案**
 - 从计算机硬件可靠性入手，在电路、电源、布线等方面采取必要的措施，提高计算机的抗干扰能力
 - 采取数据**检错**和**校正**措施，自动发现并纠正错误



身份证中的校验码

6位 地址码	8位 出生日期码	3位 顺序码	1位 校验码
--------	----------	--------	--------

- 1. 将身份证号码从左至右标记为 a_1, a_2, \dots, a_{18} ; a_{18} 即为校验码;
- 2. 计算权重系数 $W_i = 2^{18-i} \bmod 11$; 其中 \bmod 表示求余数。

所以:

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
W_i	7	9	10	5	8	4	2	1	6	3	7	9	10	5	8	4	2	1

- 3. 计算 $S = \sum_{i=1}^{17} a_i \cdot W_i$
- 4. $a_{18} = (12 - (S \bmod 11)) \bmod 11$

为什么 $\bmod 11$: 使检出以下情况的概率超过90%

1. 有一个数填错了。
2. 相邻两位填反了



纠错 (Error Correction)

- 基本思想

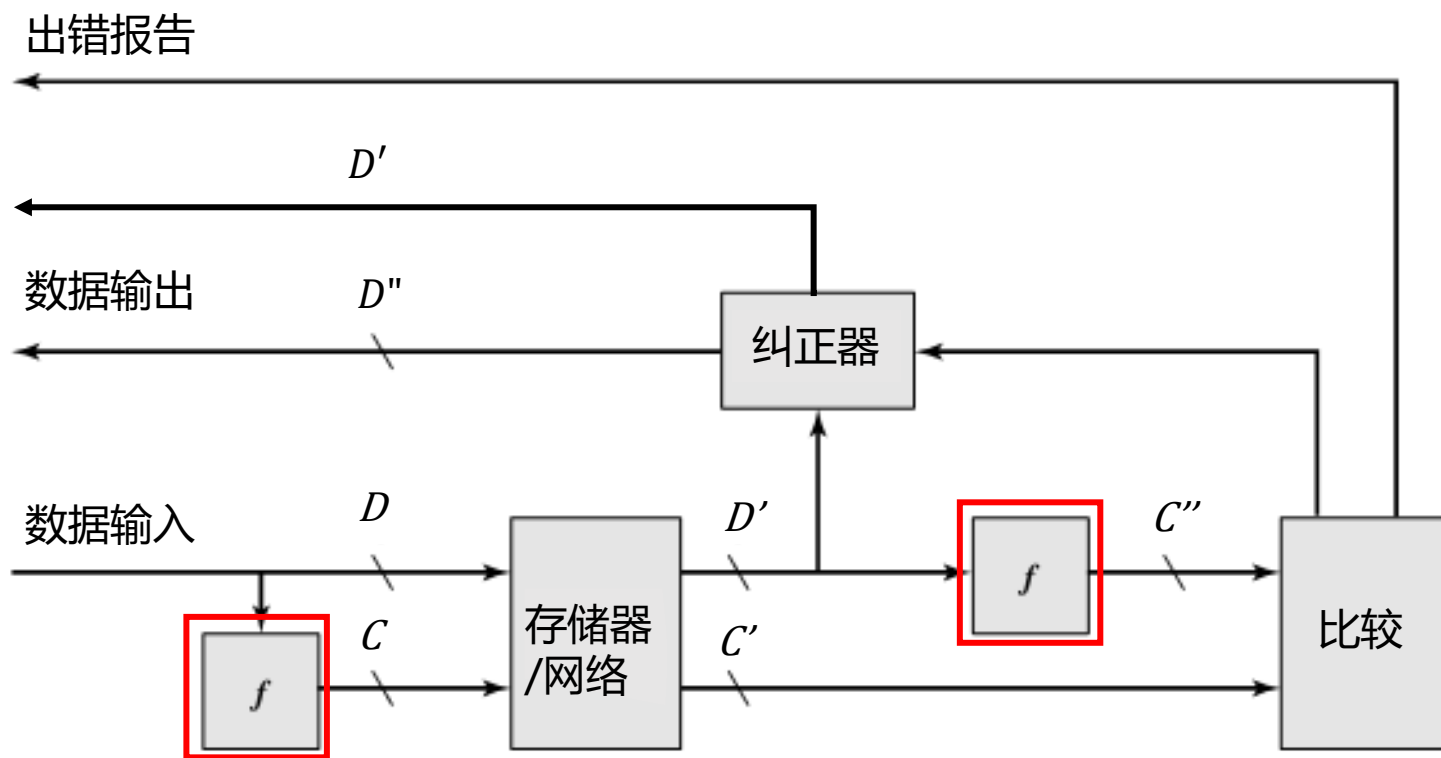
- 存储额外的信息以进行检错和校正

- 处理过程

- 数据输入：使用函数 f 在 M 位数据 D 上生成 K 位校验码 C
 - 数据输出：使用函数 f 在 M 位数据 D' 上生成新的 K 位代码 C'' ，并与取出的 K 位码 C' 进行比较
 - 没有检测到差错：使用数据 D'
 - 检测到差错且可以校正：校正数据 D' 来生成数据 D'' ，并用数据 D''
 - 检测到差错但无法纠正：报告



纠错的处理过程



奇偶校验码

- 基本思想

- 增加**1位校验码**来表示数据中**1的数量**是**奇数**还是**偶数**
 - 奇校验：使传输的数据（数据位+校验位）中有**奇数**个1
 - 偶校验：使传输的数据（数据位+校验位）中有**偶数**个1

- 处理过程

- 假设数据为 $D = D_M \dots D_2 D_1$
- 数据输入
 - 奇校验： $C = D_M \oplus \dots \oplus D_2 \oplus D_1 \oplus 1$ 1的数量为**奇数**个，校验码为0
 - 偶校验： $C = D_M \oplus \dots \oplus D_2 \oplus D_1$ 1的数量为**偶数**个，校验码为0
- 数据输出
 - 奇校验： $C'' = D'_M \oplus \dots \oplus D'_2 \oplus D'_1 \oplus 1$
 - 偶校验： $C'' = D'_M \oplus \dots \oplus D'_2 \oplus D'_1$



奇偶校验码 (续)

- 处理过程 (续)

- 检错: $S = C'' \oplus C'$
 - $S = 0$: 正确 / 数据中出错的位数为偶数
 - $S = 1$: 数据中出错的位数为奇数

- 优点

- 代价低 (只需要1位额外数据, 计算简单)

- 缺点

- 不能发现出错位数为偶数的情形
- 发现错误后不能校正
- 适用于对较短长度 (如1字节) 的数据进行检错



海明码

- 基本思想

- 将数据分成几组，对每一组都使用奇偶校验码进行检错

- 处理过程

- 将 M 位数据分成 K 组
 - 数据输入：为数据 D 中每组生成1位校验码，合并得到 K 位校验码 C
 - 数据输出：为数据 D' 中每组生成1位校验码，合并得到新的 K 位校验码 C''
 - 检错：将校验码 C'' 和取出的校验码 C 按位进行异或，生成 K 位**故障字**
(syndrome word)



海明码 (续)

- 校验码长度

- 假设最多1位发生错误
- 可能的差错
 - 数据中有1位出现错误: M
 - 校验码中有1位出现错误: K
 - 没有出现错误: 1

- 校验码的长度

$$2^K \geq M + K + 1$$



根据 M 求出最小的 K →

	单纠错	
数据位	校验位	增加的百分率 (%)
8	4	50
16	5	31.25
32	6	18.75
64	7	10.94
128	8	6.25
256	9	3.52



海明码 (续)

- 故障字的作用

- 每种取值都反映一种情形 (数据出错 / 校验码出错 / 未出错)
- 规则
 - 全部是0: 没有检测到错误
 - 有且仅有1位是1: 错误发生在校验码中的某一位, 不需要纠正
 - 有多位为1: 错误发生在数据中的某一位, 将 D' 中对应数据位取反即可纠正 (得到 D'')



海明码 (续)

- 数据位划分

- 假定数据位为8位 $D = D_8 \dots D_2 D_1$, 校验码为4位 $C = C_4 C_3 C_2 C_1$
- 数据位/校验码与故障字的关系

故障字	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
数据位	D_8	D_7	D_6	D_5		D_4	D_3	D_2		D_1		
校验位					C_4				C_3		C_2	C_1

- 数据位划分

$$C_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7$$

$$C_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7$$

$$C_3 = D_2 \oplus D_3 \oplus D_4 \oplus D_8$$

$$C_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_8$$



海明码 (续)

- 位安排
 - 将位设置在与其故障字值相同的位置

位的编号	12	11	10	9	8	7	6	5	4	3	2	1
位置编号	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
数据位	D_8	D_7	D_6	D_5		D_4	D_3	D_2		D_1		
校验位					C_4				C_3		C_2	C_1



海明码：示例

- 假定8位数据字为 $D=01101010$ ，在生成海明码时采用偶校验
- 校验位计算如下

$$C_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7 = 0 \oplus 1 \oplus 1 \oplus 0 \oplus 1 = 1$$

$$C_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7 = 0 \oplus 0 \oplus 1 \oplus 1 \oplus 1 = 1$$

$$C_3 = D_2 \oplus D_3 \oplus D_4 \oplus D_8 = 1 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$C_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_8 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$$

- 存储时的12位内容

0110**0**101**0011**



海明码：示例

- 011001010011在获取时的12位内容
 - 情形1: 011001010011
$$D'=01101010 \rightarrow C''=0011, C'=0011$$
$$S = C'' \oplus C' = 0011 \oplus 0011 = 0000$$
 - 情形2: 011101010011
$$D'=01111010 \rightarrow C''=1010, C'=0011$$
$$S = C'' \oplus C' = 1010 \oplus 0011 = 1001 \rightarrow D_5 \text{ 出错}$$
 - 情形3: 011011010011
$$D'=01101010 \rightarrow C''=0011, C'=1011$$
$$S = C'' \oplus C' = 0011 \oplus 1011 = 1000 \rightarrow C_4 \text{ 出错}$$



码距和纠错理论

- 码距：同一编码中，任意两个合法编码之间不同二进制数位数的最小值
 - {0000, 0001, 0010, 0011} 码距为1
 - {0000, 0011} 码距为2
 - 纠错理论： $L - 1 = D + C, D \geq C$
 - L 是码距， D 是检错位数， C 是纠错位数
 - 奇偶校验的码距是2， 1位能检错，不能纠错
 - 海明码的码距是3， 1位能检错和纠错
- 海明码能检测2位纠错0位吗？



补充阅读：SEC-DED

- SEC-DED

- 单纠错，双检错
- 可以找到两个位产生的错误，并纠正一个位的错误
- 添加了一个额外的校验位

$$C_5 = D_1 \oplus D_2 \oplus D_3 \oplus D_5 \oplus D_6 \oplus D_8$$

- 如果一位数据发生错误，三位校验码将被更改
- 故障字
 - 都是0：没有检测到错误
 - 1位为1：在5个校验位中有一个发生了错误，不需要修正
 - 2位为1：有2位数据和校验位出现错误，但找不到错误的位置
 - 3位为1：8位数据位中有1位发生了错误，该错误可以被纠正
 - 3位以上均为1：严重情况，检查硬件



补充阅读：SEC-DED (续)

- SEC-DED (续)
 - 纠错码以增加复杂性为代价来提高存储器的可靠性
 - 主存的实际容量比用户见到的容量要大

数据位	单纠错		单纠错/双检错	
	校验位	增加的百分率 (%)	校验位	增加的百分率 (%)
8	4	50	5	62.5
16	5	31.25	6	37.5
32	6	18.75	7	21.875
64	7	10.94	8	12.5
128	8	6.25	9	7.03
256	9	3.52	10	3.91

$$C_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7$$

$$C_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7$$

$$C_3 = D_2 \oplus D_3 \oplus D_4 \oplus D_8$$

$$C_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_8$$

$$C_5 = D_1 \oplus D_2 \oplus D_3 \oplus D_5 \oplus D_6 \oplus D_8$$



循环冗余校验

- 奇偶校验问题
 - 额外成本很大
 - 要求将数据分成字节
- 循环冗余校验 (Cyclic Redundancy Check, CRC)
 - 适用于以流格式存储和传输大量数据
 - 用数学函数生成数据和校验码之间的关系



循环冗余校验 (续)

- 基本思想

- 假设数据有M位，左移数据K位（右侧补0），并用K+1位生成多项式除它（模2运算）
- 采用K位余数作为校验码
- 把校验码放在数据（不含补的0）后面，一同存储或传输

- 校错

- 如果M+K位内容可以被生成多项式除尽，则没有检测到错误
- 否则，发生错误



循环冗余校验：示例

- 假设数据是 100011, 约定的多项式为1001 ($x^3 + 1$)
- 校验码是111
- 发出的数据是100011111
- 接收方也对1001做模2运算, 余数为0则无误

无借位减, 等价于异或运算

$$\begin{array}{r} 100111 \\ 1001 \overline{) 100011000} \\ \underline{1001} \\ 0011 \\ \underline{0000} \\ 0111 \\ \underline{0000} \\ 1110 \\ \underline{1001} \\ 1110 \\ \underline{1001} \\ 1110 \\ \underline{1001} \\ 111 \end{array}$$



循环冗余校验：补充阅读

- 为什么叫循环冗余？
 - CRC是基于有限域**GF(2)**（即除以2的同余）的多项式环。即所有系数都为0或1（又叫做二进制）的多项式系数的集合，并且集合对于所有的代数操作都是封闭的。
 - 例如乘法运算 $(x^2+x)(x+1) = x^3 + 2x^2 + x \equiv x^3 + x \pmod{2}$
 - 例如除法运算 $\frac{x^3+x^2+x}{x+1} = (x^2+1) - \frac{1}{(x+1)}$
 - $(x^2+x+1)x = (x^2+1)(x+1) - 1$
↓ 通用的构造方法
 - $M(x) \cdot x^n = Q(x) \cdot K(x) - R(x)$
 - $M(x)$ 是原数据的信息多项式， $K(x)$ 是构造的多项式， $Q(x)$ 是商的多项式， $R(x)$ 是余数的多项式
 - 所以 $M(x) + R(x)$ 就是发送的数据。



循环冗余校验：补充阅读

常用CRC（按照ITU-IEEE规范） [编辑]

名称	多项式	表示法：正常或者翻转
CRC-1	$x + 1$ (用途：硬件，也称为奇偶校验位)	0x1 or 0x1 (0x1)
CRC-5-CCITT	$x^5 + x^3 + x + 1$ (ITU G.704标准)	0xB (0x??)
CRC-5-USB	$x^5 + x^2 + 1$ (用途：USB信令包)	0x5 or 0x14 (0x9)
CRC-7	$x^7 + x^3 + 1$ (用途：通信系统)	0x9 or 0x48 (0x11)
CRC-8-ATM	$x^8 + x^2 + x + 1$ (用途：ATM HEC, PMBUS (参见SMBUS.org[1] (页面存档备份, 存于互联网档案馆)))	0x7或0xE0 (0xC1)
CRC-8-CCITT	$x^8 + x^7 + x^3 + x^2 + 1$ (用途：1-Wire 总线)	0x8D
CRC-8-Dallas/Maxim	$x^8 + x^5 + x^4 + 1$ (用途：1-Wire bus)	0x31或0x8C
CRC-8	$x^8 + x^7 + x^6 + x^4 + x^2 + 1$	0xD5 (0x??)
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x + 1$	0x233 (0x????)
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$ (用途：通信系统)	0x80F或0xF01 (0xE03)
CRC-16-Fletcher	参见Fletcher's checksum	用于Adler-32 A & B CRC
CRC-16-CCITT	$x^{16} + x^{12} + x^5 + 1$ (X25, V.41, Bluetooth, PPP, IrDA)	0x1021或0x8408 (0x0811)
CRC-16-IBM	$x^{16} + x^{15} + x^2 + 1$ (用途：Modbus)	0x8005或0xA001 (0x4003)
CRC-16-BBS	$x^{16} + x^{15} + x^{10} + x^3$ (用途：XMODEM协议)	0x8408 (0x????)
CRC-32-Adler	参见Adler-32	参见Adler-32
CRC-32-MPEG2	参见IEEE 802.3	参见IEEE 802.3
CRC-32-IEEE 802.3	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	0x04C11DB7或0xEDB88320 (0xDB710641)
CRC-32C(Castagnoli)	$x^{32} + x^{28} + x^{27} + x^{26} + x^{25} + x^{23} + x^{22} + x^{20} + x^{19} + x^{18} + x^{14} + x^{13} + x^{11} + x^{10} + x^9 + x^8 + x^6 + 1$	0x1EDC6F41或0x82F63B78 (0x05EC76F1)
CRC-64-ISO	$x^{64} + x^4 + x^3 + x + 1$ (用途：ISO 3309)	0x0000000000000001B或0xD800000000000000 (0xB000000000000001)
CRC-64-ECMA-182	$x^{64} + x^{62} + x^{57} + x^{55} + x^{54} + x^{53} + x^{52} + x^{47} + x^{46} + x^{45} + x^{40} + x^{39} + x^{38} + x^{37} + x^{35} + x^{33} + x^{32} + x^{31} + x^{29} + x^{27} + x^{24} + x^{23} + x^{22} + x^{21} + x^{19} + x^{17} + x^{13} + x^{12} + x^{10} + x^9 + x^7 + x^4 + x + 1$ (参见ECMA-182 (页面存档备份, 存于互联网档案馆) p.63)	0x42F0E1EBA9EA3693或0xC96C5795D7870F42 (0x92D8AF2BAF0E1E85)
CRC-128	IEEE-ITU标准。被MD5 & SHA-1取代	
CRC-160	IEEE-ITU标准。被MD5 & SHA-1取代	



总结

- **纠错**
 - 数据出错的原因，纠错的原理和处理过程
- **常用的数据校验码**
 - 奇偶校验码
 - 海明码
 - 循环冗余校验码



谢谢

bohanliu@nju.edu.cn



南京大學
NANJING UNIVERSITY