

计算机组织结构

4 数据校验码

任桐炜

2021年9月23日



南京大学
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

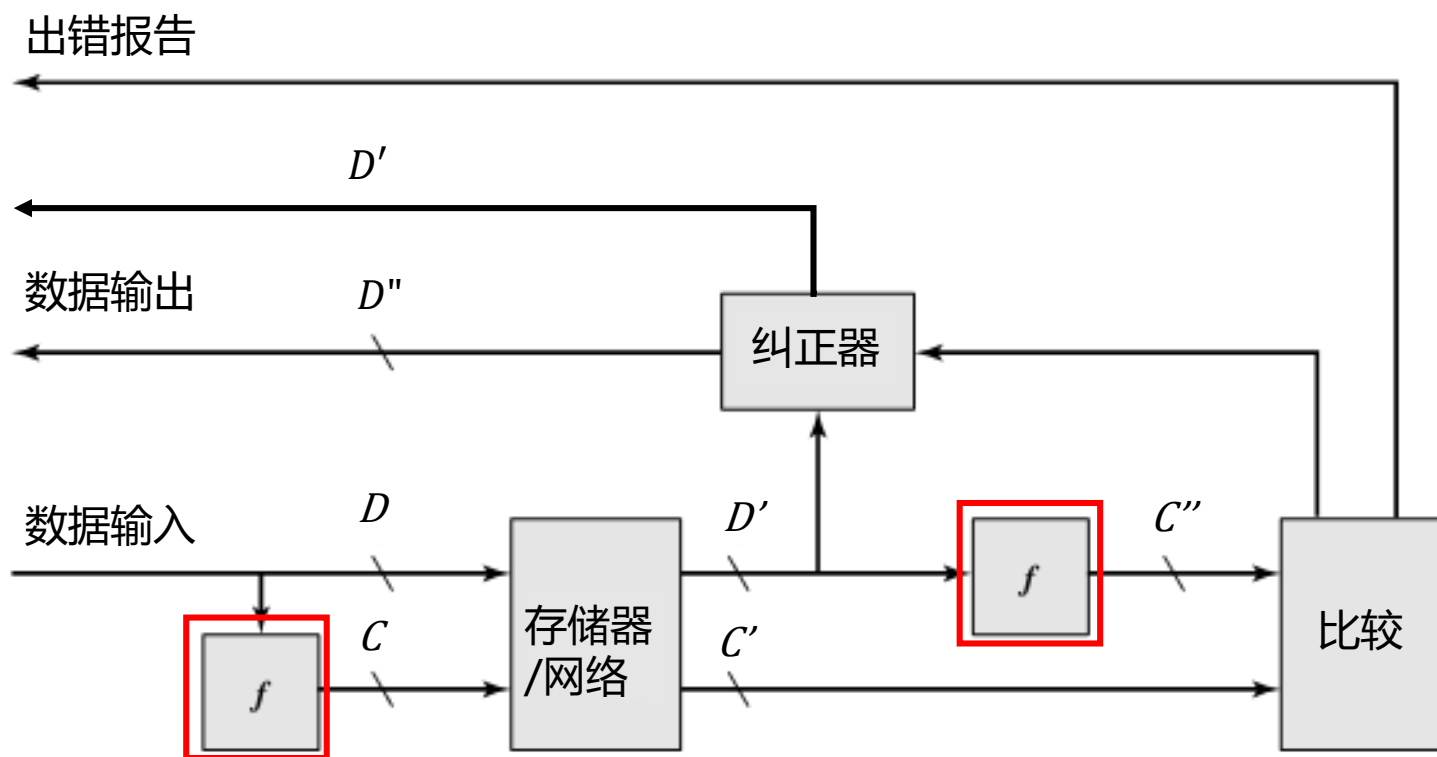


纠错 (Error Correction)

- 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
- 处理过程
 - 假设数据为 $D = D_M \dots D_2 D_1$
 - 数据输入
 - 奇校验: $C = D_M \oplus \dots \oplus D_2 \oplus D_1 \oplus 1$
 - 偶校验: $C = D_M \oplus \dots \oplus D_2 \oplus D_1$
 - 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

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

数据位	单纠错	
	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	D8	D7	D6	D5		D4	D3	D2		D1		
校验位					C4				C3		C2	C1

- 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	D8	D7	D6	D5		D4	D3	D2		D1		
校验位					C4				C3		C2	C1



海明码 (续)

- 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

011001010011



海明码 (续)

- 示例 (续)

- 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: 011**1**01010011

$$D' = 011**1**1010 \rightarrow C'' = 1010, C' = 0011$$

$$S = C'' \oplus C' = 1010 \oplus 0011 = 1001$$

- 情形3: 0110**1**1010011

$$D' = 01101010 \rightarrow C'' = 0011, C' = **1**011$$

$$S = C'' \oplus C' = 0011 \oplus 1011 = 1000$$

[谭琼, 141250122]



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



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 (续)

- 示例

- 假设数据是 100011, 生成多项式为1001
- 校验码是111

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



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

rentw@nju.edu.cn



南京大學
NANJING UNIVERSITY