

Dependable memory hierarchy

2019年5月20日 14:24

Dependency Measurement:

MTTF

MTTR

MTBF=MTTF + MTTR

Availability = MTTF / (MTTF + MTTR)

Improving Availability

- ◆ Increase MTTF: fault avoidance, fault tolerance, fault forecasting
- ◆ Reduce MTTR: fault detection, fault diagnosis and fault repair

Dependency:

SEC code

Hamming distance

Distance = 2 时 eg. 奇偶校验

Distance = n 时 Detect n-1 Correct $\left\lfloor \frac{n-1}{2} \right\rfloor$

Hamming code

- Parity bits = 0000 indicates no error
- Parity bits = 0101 indicates bit 10 was flipped

Bit position		1	2	3	4	5	6	7	8	9	10	11	12
		0	1	1	1	0	0	1	0	1	1	0	
Encoded date bits		p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8
Parity bit coverate	p1	X		X		X		X		X		X	
	p2		X	X			X	X			X	X	
	p4				X	X	X	X					X
	p8								X	X	X	X	X

✓ 0

X 1

✓ 0

X 1

Can detect (Hamming distance-1)

How to repair ($\left\lfloor \frac{n-1}{2} \right\rfloor$) bits error

附:

o 计算SEC编码

- 1 计算需要的校验位个数：对一个长度为d的字符串 假设需要p个校验位
 - 有如下关系： $p+d \leq 2^p$
- 2 预留校验位位置：把校验位按照 2^n 放在相应的第 2^n 位置上 记录为 $p2^n$
 - 然后把数据放入剩下的空间 记录为 d_i
- 3 计算校验位数值：把位置转化为二进制
 - P1 编码二进制下位置 最后一位为1 调整P1使得求和为0
 - P2 编码二进制下位置 倒数第二位为1 调整P2使得求和为0
 - P4 编码二进制下位置 倒数第三位为1 调整P4使得求和为0
 - ...
- 4 利用SEC编码进行校验和修正
 - 校验：计算位置在二进制编码后 倒数第一/二/...位置上的数字之和 看求和是否都为0
 - 找到出错位置并修正：假设接收端接收到的第10个bit有错
 - ◆ 此时校验位p2, p8会不通过 (求和结果为1)
 - ◆ 那么把校验位的bit串写出来：p8,p4,p2,p1 -> 1010 -> 10
 - ◆ 从bin转化为dec 即为出错的位置
 - ◆ 修正：将此处的数值翻转即可
- 评价SEC编码的效果
 - 此时的最小汉明距离：为3
 - ◆ 从图上可以看出 对于每个data bit 最少会影响两个parity bit
 - ◆ 如果一个data bit发生改变 那么其自身 和影响的两个parity bit也会改变
 - ◆ 因此最小汉明距离为3
 - 因此最多可以检测2错 修正1错
 - ◆ 实际上对于检测出来出错 都会当作只有1个错误修正
 - ◆ 因为就算是两个bit错误 也没办法
- 改进：增加一个整个串的parity bit
 - 此时最小汉明距离提升到4
 - 这个时候可以区分到底是1个错误还是2个错误
 - ◆ 如果是1个错误 可以修正
 - ◆ 如果是2个错误 可以知道错误数量为2 但是无法修正
- 实例：ECC DRAM：对每64 data bit 用 8bit的 SEC/DED编码

改进后的叫做：SEC/DED CODE