

16b18b

编码原理

16b/18b编码，其控制字符的定义可能因不同的应用和规范而有所不同。因此，在16b/18b编码的Excel表中可能没有提供控制字符的定义，因为这些定义通常需要根据具体的通信标准或规范来确定（个人猜测）。在控制字符的设定当中，可以仿造8b/10b编码的规则来创建16b/18b编解码的控制字符（想法，暂未验证）。

数据分析总结（已验证）

1. 在RD-的数据中，保证了**1的数量比0多**，RD+的数据中，保证了**0的数量比1多**。取值在 正负4、正负2、0
2. RD反映了上一个编码的01数量情况，来指导当前编码的数据选择。RD-是指上一轮编码的0比1多，所以当前编码时候，从RD-取值（RD-的数据都是1比0多）。最后保证了整体编码的01稳定性。
3. 分析该行的两列数据具有什么关系：**除了是相同编码就是反码，此外没有其他关系**。由此可得，在设计映射表的时候可以用技巧节省大量存储空间。
4. 测试得到：**非完美编码不存在RD-和RD+数值相等**的行数据。
5. 测试得到：**完美平衡码存在相等也存在反码关系**
6. 重新验证后，使用**数组+下标**存储编码值不会造成空间浪费。
7. 在RD-中与RD+存在相同的数据，只可能是**RD+RD-同一行相同**的情况，即**完美编码**，确保了编码的**唯一性**，在做解码的时候可以直接对应
8. 编码器， $(512 \times 12\text{bit}) + (128 \times 12\text{bit}) = 6144 + 1536 = 7680\text{bit} = 7.5\text{kb}$
9. 解码器， $(1024 \times 9) + (256 \times 7) = 9216 + 1792 = 11008\text{bit} = 10.75\text{kb}$ 。总1024 有效792 有效率77.3% 总256有效188 有效率73.5%

技术实现

假设分析

假设一：物理层不做控制字符检测，只实现DC平衡的功能，是否可行？

假设二：编解码控制字符是否可以由个人定义？定义过程总遵从的规则能否完全仿造8b/10b的规则？

实现思路

实现大概9个模块

1. top整体控制模块
2. 特殊字符控制模块（欠缺，可能需要自己设计和选择编码表）
3. 9B代码产生模块
4. 7B代码产生模块
5. 极性判断与控制信号产生模块（待测试）
6. 9B/10B编码器模块（finish）
7. 7B/8B编码器模块
8. 整合输出模块

RD切换模块使用状态机实现。

当前状态，下一状态，状态的切换是要根据编码的D来决定。

方案一：D从何而来？ $D = \text{"0"个数} - \text{"1"个数}$ （考虑提前计算出来，然后分类？）D的数值是 0、-2、+2 一共三个数值，可以考虑使用2bit来存放该数据。

那么原来的10bit和8bit数据都需要拓宽2bit来存放相对应的D值。

12bit有1024个，10bit的数据有256个。共占 14.5kb

还有512个9bit数据，128个7bit数据。共占 5.375kb

方案二：类似8b10b编码，总结规律，将编码划分成几类，但仍至少需要增加1位。总体数据存储量估计将减少30% - 40%。

由实验总结后可以得知，编码有3种类别。1.非完美平衡码只存在反码关系。2.完美平衡码存在反码。3.完美平衡码相等的关系

2位可以表示4个类别，可以在选取RD+或者RD-其中某一列的数据，然后增加2bit的flag标签，用来表示该编码的性质。

RD的转换需要依靠该flag：例如，当前初始为RD-，选取9b/10b编码，该编码为相同编码则不需要改变RD性质，如果不是相同编码则需要从RD-列数据中选择对应的编码，而且RD-数据中的1的数量比0多，则RD的极性必然从-变成+。

实施：

1. 对数据分析，算出各个数据的0,1个数，再计算D值。（finish）
2. 分析该行的两列数据是否处于相反的关系（反码）。（finish）
3. 判断和标注哪些是完美平衡码。（finish）
4. 判断和标注完美平衡码中哪些是反码。（finish）

注意：

1. 完美平衡编码中的RD-和RD+值不一定会相等。需要进行分析验证
2. 在不平衡编码中，RD-和RD+的值不一定成反码关系，需要进一步分析验证。

对于编码器模块的例化在RD转换模块实现还是做一个top模块实现，有待考虑

具体实现

- 1.在数据映射表中代码以RD-为基准，增加2bit数据用来表明该行编码的3种关系。

问题

1. 控制编码是否能自己根据规则定义？
2. 物理层不做控制字符检测，只实现DC平衡的功能，是否可行？

