一道简单的笔试题\_格雷码转换

**格雷码**

自然二进制码相邻数据之间可能存在多个bit的变化，例如自然数7和8对应的4bits自然二进制码分别“0111”、“1000的输出从7变到8时，寄存器的每一位都会发生变化，从而造成不稳定态，并且会使得数字电路产生很大的尖峰电流脉冲。而格雷码则没有这一缺点，其中的所有相邻整数在它们的二进制表示中仅有一位不同。

下表给出了4bits自然二进制码、格雷码与十进制整数的对照表：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 十进制 | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
| 二进制码 | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
| 格雷码 | 0000 | 0001 | 0011 | 0010 | 0110 | 0111 | 0101 | 0100 |
| 十进制 | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
| 二进制码 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |
| 格雷码 | 1100 | 1101 | 1111 | 1110 | 1010 | 1011 | 1001 | 1000 |

**1.二进制码转格雷码:**

        称为格雷码的编码，方法是从二进制码的最右边一位（最低位）起，依次将每一位与左边一位进行异或运算，作为对应格雷码该位的值，而最左边高位不变。

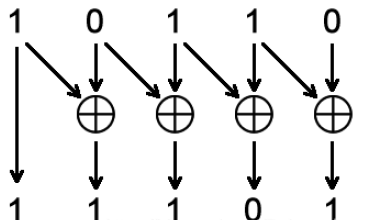
对应公式如下：

g[n] = b[n]，

g[i] = b[i] xor b[i+1] (i∈N,n-1≥i≥0)；

其中g、b分别对应n位的格雷码和二进制码。

例如，将自然二进制码“10110”转换为格雷码，可以形象的用下图表示其转换过程：



用Verilog描述：

assign gray\_value = binary\_value ^ (binary\_value>>1);

**2.格雷码转二进制码:**

        称为格雷码的解码，方法是从格雷码左边第二位（次高位）起，将每一位与其左边一位解码后的值异或，作为该位解码后的值，而最左边一位（最高位）不变就是它本身。

对应公式如下：

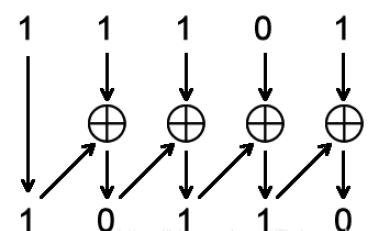
b[n] = g[n]，

b[i] = g[i] xor b[i+1] (i∈N, n-1≥i≥0)

其中g、b分别对应n位的格雷码和二进制码。

例如，将格雷码“11101”转换为自然二进制码，可以形象的用下图表示其转换过程：

        根据格雷码的最高位，得到二进制的最高位，然后，用二进制码的最高位与格雷码的次高位相异或，得到二进制的次高位，依次类推。



用Verilog描述：

module gray2bin #(parameter N = 6)(

input [N-1:0] gray,

output [N-1:0] bin

);

assign bin[N-1] = gray[N-1]; //据格雷码的最高位，得到二进制的最高位

genvar i;

generate

for(i = N-2; i >= 0; i = i - 1) begin: gray\_2\_bin

assign bin[i] = bin[i + 1] ^ gray[i]; //二进制码的最高位与格雷码的次高位相异或，得到二进制的次高位

end

endgenerate

endmodule

**【某笔试题】**

一个四位十六个状态的格雷码计数器，起始值为1001，经过100个时钟脉冲作用之后的值为（）。

解析：

    先计算出100个脉冲后跑了多少个16状态，100/16=6余4；

故需要知道1001后的第四个状态是哪个？

1001转为二进制为1110，1110为十进制14，再后4个数是15，0,1，2；

故第四个数为2，转为格雷码为：0011