

logisim计组实验五 CRC校验电路

原创

Treblez* 于 2020-04-24 22:54:24 发布 3487 收藏 29

分类专栏: [logisim与计算机组成](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/treblez/article/details/105726765>

版权



[logisim与计算机组成](#) 专栏收录该内容

14 篇文章 23 订阅

订阅专栏

文章目录

[CRC \(7,3\) 串行编码电路](#)

[CRC \(21,16\) 并行编码电路](#)

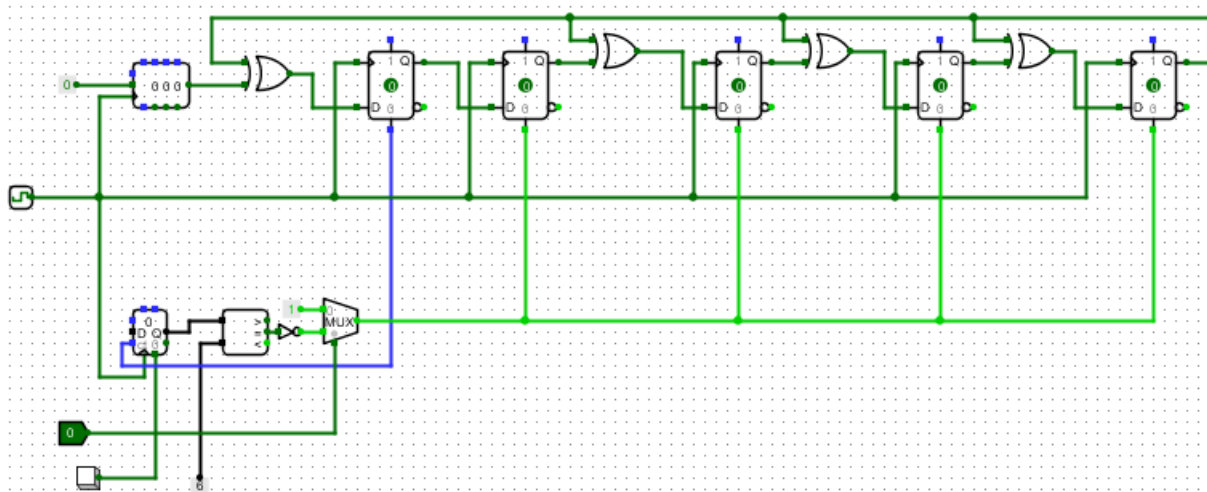
[CRC \(21,16\) 并行解码电路](#)

CRC校验电路的logisim设计, 有什么问题就提出来吧~

电路文件已经托管至Github, 欢迎star: [点这里](#)

CRC (7,3) 串行编码电路

具体设计教材中已经给出, 这里没有验证修改。



<https://blog.csdn.net/treblez>

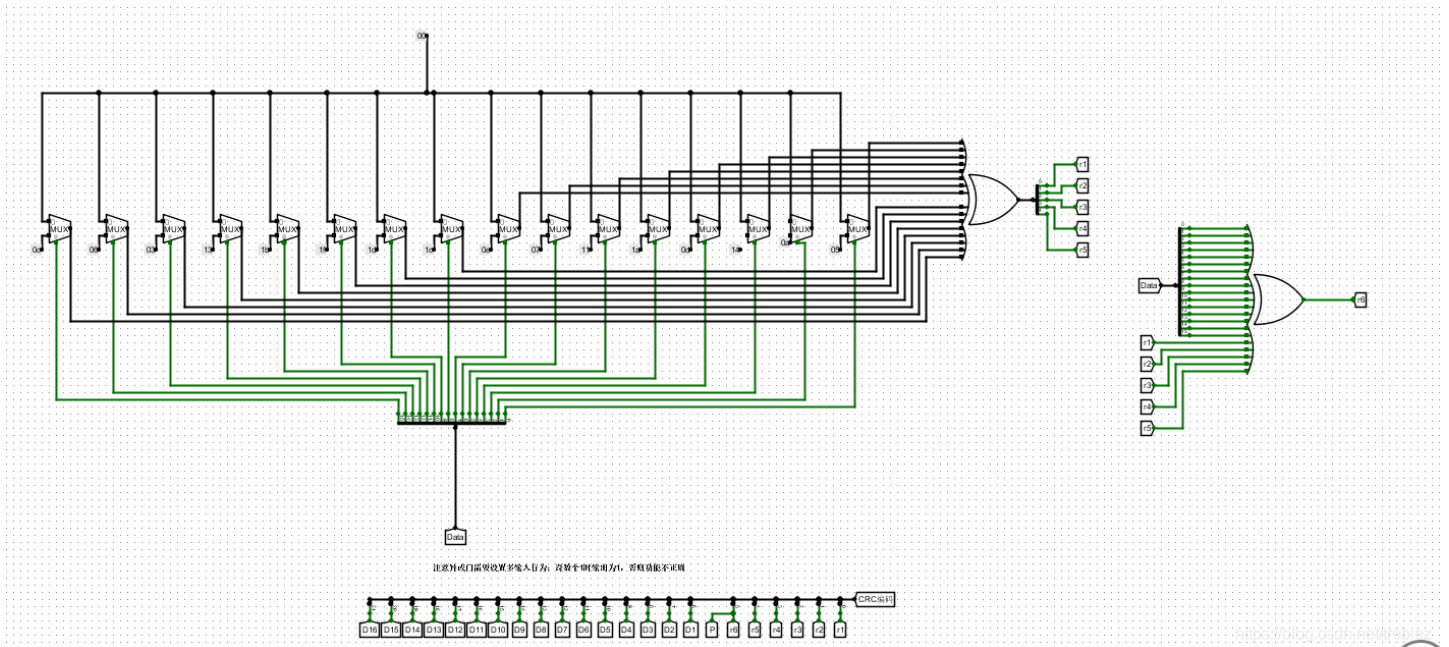
CRC (21,16) 并行编码电路

由公式： $2^r - 1 \geq N = k + r$ $k = 16$ 得 $r = 5$ ，共需要21位。为了后续检验两位错，我们需要另外加一位总偶校验位。因为多项式的 $G(x)$ 最高次幂等于校验位的位数，最低次幂等于0，选择多项式100101。这里和串行电路部分是一样的。并行编码和串行编码不同的地方在于：并行编码使用异或运算的结合律将时序电路转换为了纯逻辑电路，增加了运算速度。

先得出每一位的余数表格，之后数据位为0取0非0取余数

之后就可以设计电路了。

如下图所示：



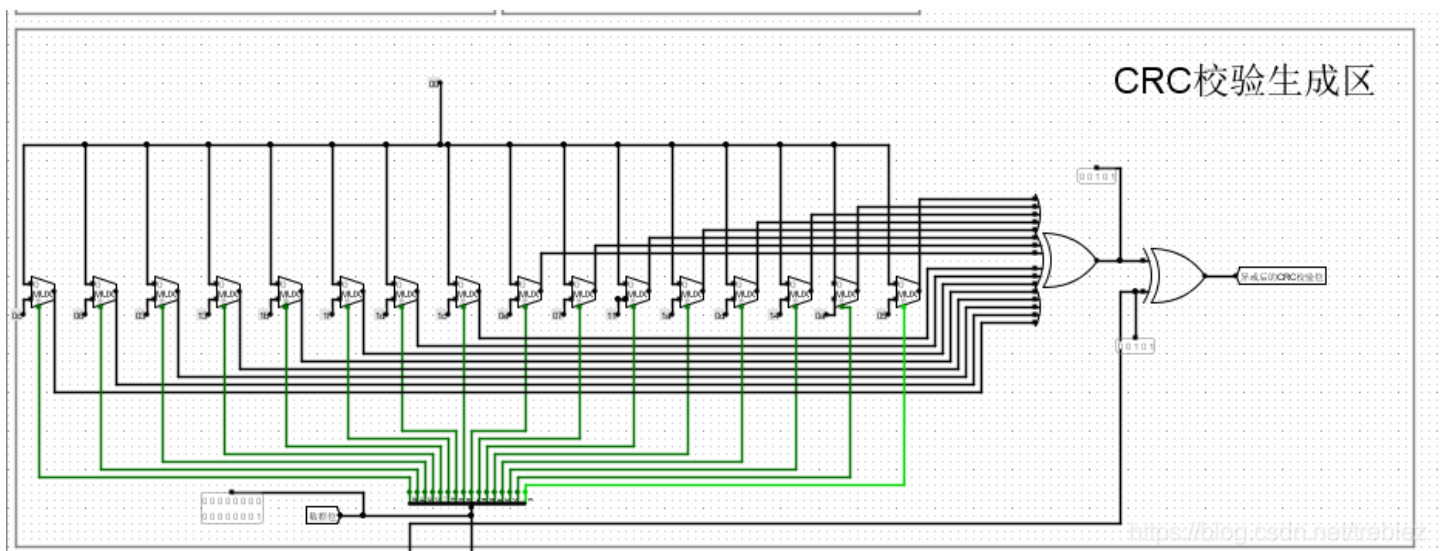
一看就是老电王子

CRC (21,16) 并行解码电路

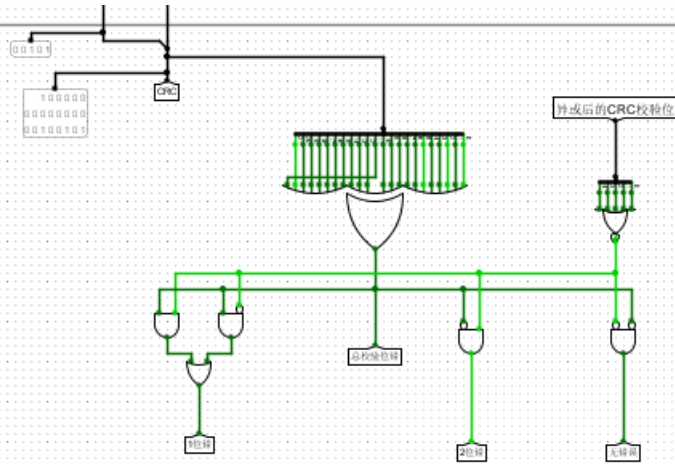
将传输过来的16位数据位与多项式100101进行模2除法，得到的余数再和传输的CRC校验和进行异或，即可得到出错位置相对应的余数。记为W。

将得到的余数W和每一位相对应的余数进行异或。若最后得到的值全为0，即表示这一位出错。

得到的三部分电路如图所示：

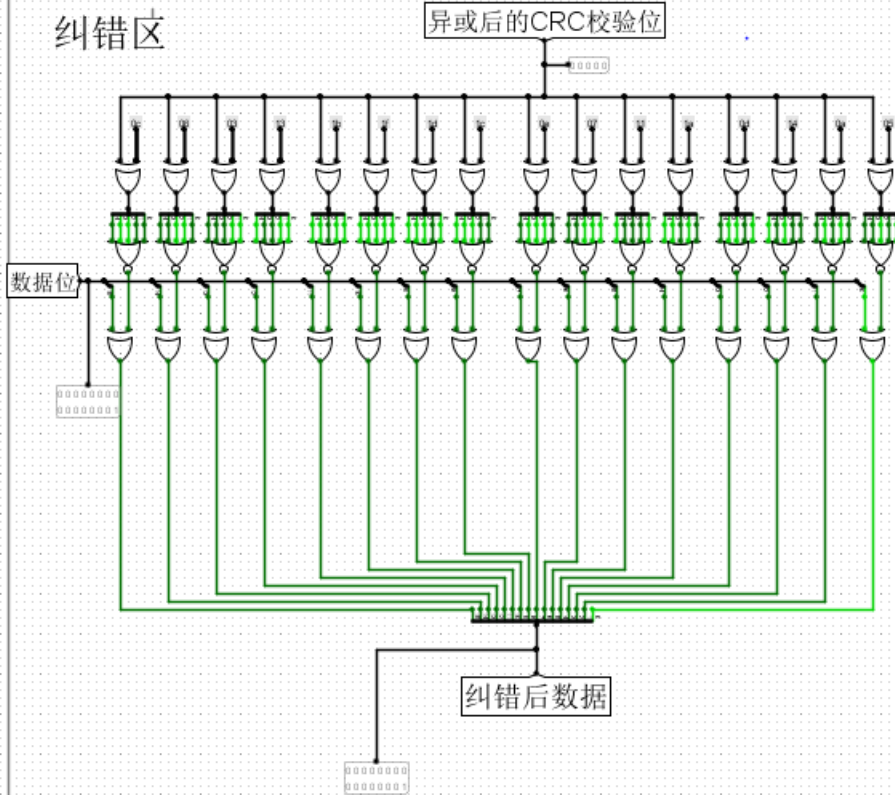


检错区



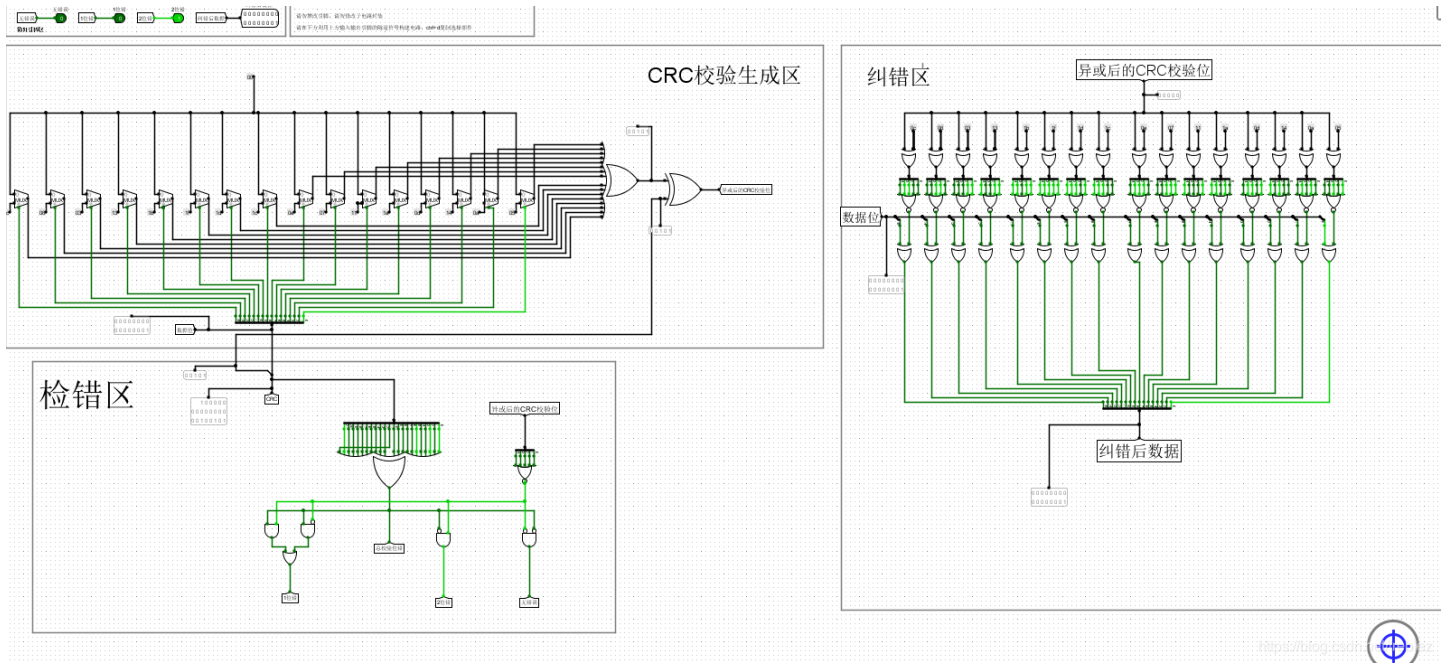
<https://blog.csdn.net/tretblez>

纠错区



<https://blog.csdn.net/tretblez>

总体图如下所示:



难顶啊，做了一整天，有个错误找了三四个小时
emmm最后突然就过了

