



Bookmarks

- ▶ [Hello! Computer Organization!](#)
- ▶ [基础知识](#)
- ▶ [Logisim](#)
- ▶ [Verilog-HDL与ISE](#)
- ▶ [MIPS指令集及汇编语言](#)
- ▼ [P0-Logisim简单部件与状态机](#)

课下测试

P0-Logisim简单部件与状态机 > 课下测试 > CRC校验码计算电路 的设计与测试

CRC校验码计算电路 的设计与测试

[Bookmark this page](#)

在这个部分中，你将从需求开始一步一步搭建我们所要求的 8 位 CRC校验码计算电路。我们希望在这个简单的例子中，你能体会到方法性的东西，并应用到之后的设计中去。这对之后的学习和课上测试都是很有帮助的，请各位谨记。

CRC校验码简介

CRC校验是数据通信领域中最常用的一种查错校验方式，它对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

在了解这种校验码怎么计算之前，我们需要先了解一种特殊的除法：“模二除法”。它与算术除法类似，但在做减法时既不向上借位，也不比较除数和被除数相同位数值的大小；它的运算法则为 $1-1=0$ ， $0-1=1$ ， $1-0=1$ ， $0-0=0$ ，例如 $1100-1001=0101$ 。对于模二除法，我们以被除数为1011,除数为10为例，运算过程如下：

$$\begin{array}{r} 101 \\ 10 \overline{) 1011} \\ \underline{10} \\ 01 \\ \underline{00} \\ 11 \\ \underline{10} \\ 1 \end{array}$$

10 最高位为 1,为了消去最高位,需要商 1, 即 $10-10*1=10-10=00$

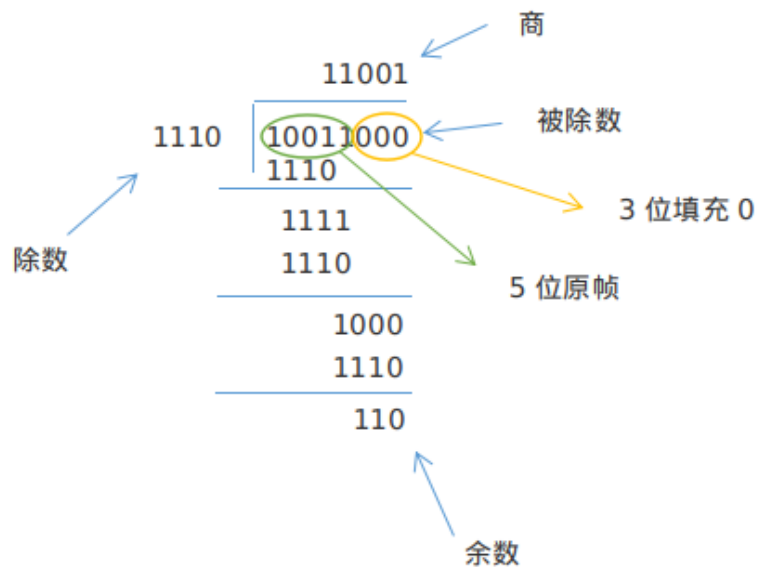
01 最高位为 0,为了消去最高位,应该商 0,即 $01-10*0=01-00=01$

11 最高位为 1, 商 1 , 即 $11-10*1=11-10=01$

剩下的位数小于除数的位数, 作为余数

如果你细心的话，可以发现，模二除法中的减法和异或的操作是完全相同的，所以模二除法可以用异或来完成。

知道了模二除法的计算过程，CRC校验码的计算就很简单了。我们只需要将原帧补上(除数位数-1)个0作为被除数，然后进行模二除法即可。举个例子，我们要发送的帧A为10011，发送端和接收端共同选定的除数B为1110。因为B是4位二进制数，我们需要在A的后面补上3个0，从而得到A'=10011000。我们将A'作被除数，B作除数，进行“模二除法”。如下图：



最后得到的余数是一个三位数(注意如果不是三位，也要在前面补零来凑齐三位)，这就是要求的校验码。我们将得到的校验码110拼接在原数据帧的后面，就得到了要发送的新帧A"=10011110。这样就完成了CRC校验码的生成。

CRC校验码计算电路的定义

在普通的设计流程中，一般都是用汉语描述电路的功能，需要我们自己来进行形式化的定义，不过由于我们这个例子相对简单，因此直接给出我们8位CRC校验码计算电路的端口定义。

信号名	方向	描述
A[7:0]	I	8位原数据帧
B[3:0]	I	4位除数
C[10:0]	O	8位原数据帧+3位校验码

分析目标并层次化

初拿到如上表的需求，可以发现这是一个没有状态的部件，自身内部不存储信息，单纯的输入就决定了输出，因此有一种相当暴力的思路就是直接画出真值表，输入总共11位2进制码(考虑到除数的最高位一定为1)，输出11位2进制码，我们需要动2048下手指才可以把真值表画出来——似乎繁琐了点？我们需要更加简单的方法。但是希望大家明白真值表法永不过时，在输入输出比较小时是非常实用的方法，并且可以使用相关工具自动生成电路。

我们仔细分析一下，其实类似除法的计算方法，我们可以把这个计算过程分解为多次进行被除数为4位的除法计算。因此我们引入“4位模二除法器”这个间接层，这样问题就变成了两个部分：

- 设计四位模二除法器
- 使用四位模二除法器搭建8位CRC校验码计算电路

为了方便理解,我们的给出了一种四位模二除法器的端口定义(这只是为了讲解时参考，你在设计时不一定要遵循这个定义)

信号名	方向	描述
A[3:0]	I	4 位被除数
B[3:0]	I	4 位除数
C[2:0]	O	3 位余数
D	O	1 位商

我们知道，模二除法和异或运算是等价的，所以对于一个四位除四位的模二除法器，如果被除数的最高位是1,则商 1 ,余数使用异或门来将四位被除数和四位除数异或即可。当然，如果被除数的最高位是0,则商为 0,余数直接等于被除数了。

这个电路搭建很简单，我们只需要判断下被除数的最高位，然后分情况输出即可。

搭建完四位的除法器，我们再来看看怎么使用这个简单的电路来搭建我们所需要的复杂电路。类似于普通除法的过程，我们计算模二除法时也是每次从被除数中取出一定的位数(对于该问题来说是四位)来和除数相除，除得的余数再补上一位被除数后继续与除数相除。如此，计算过程就相当于进行多次四位的模二除法了。我们要做的就是将上一个四位模二除法器的余数输出，拼接一位被除数作为下一个四位模二除法器的被除数输入(除数始终是同一个数),如此反复直到被除数所有的位都被使用。

具体搭建（LOGISIM）

在Logisim搭建上述电路时还需注意：

- 可以使用Tunnel来表示中间变量简化布线
- 可以改变门电路输入数简化电路
- 这个电路并不是最简的，可以使用逻辑方法优化

综上，我们可以在完成了设计工作的基础上使用Logisim进行电路搭建啦！

测试与验证

正如前文所述，开发者的自我测试在开发过程中是十分重要，这里我们就以刚搭建的8位CRC校验码计算电路为例来说明如何进行测试样例的设计。对于我们这里的组合电路，测试样例要覆盖所有的输入。因此最简单直接的想法就是将所有的二进制输入依次尝试一遍。在我们的电路中，共需尝试2048组输入。这种方法的好处就是保证彻底的覆盖性，缺点也很明显，需要过多的操作。在面对更复杂电路时，我们需要更加合理的样例。一种比较合理的方法是根据功能来设计相关的样例：

- 覆盖所有测试功能正常进行的样例（如A: 11001010 B: 1011, A: 01001110 B:1100）
- 极端情况的样例（A: 00000000 B: 1000）
- 异常情况的样例（如：A: 110X010X B: XXXX, 这个X的输入在Logisim也是支持的）

在设计出相应的测试样例后，只需进行相关的输入观测现象即可。

Discussion

Topic: P0: Lab0 - 课下测试 / CRC校验码计算电路 的设计与测试

Show Discussion