P0（Logisim练习）

▼ 目录

设计电路模块

如何规范优雅的设计电路模块

电路设计目标

端口定义表

Aa 信号名	方向	描述
A[7:0]	I	第一个加数
B[7:0]	I	第二个加数
C0	I	初始进位
SUM[7:0]	O	两数相加之和
Overflow	O	溢出标志位 Overflow = C8

8位加法器，信号语法类似verilog

对于更复杂的电路会再多一张功能跟那个定义表，更好的理清电路设计思路，也可以应对同一模块在不同情况与输入下有不同的功能的场景，

模块接口

信号名	方向	描述
Clk	I	MIPS-C 处理器时钟
Reset	I	复位信号
RS1[4:0]	I	读寄存器文件时的第 1 个寄存器下标
RS2[4:0]	I	读寄存器文件时的第 2 个寄存器下标
RD[4:0]	I	写寄存器文件时的寄存器下标
RegWrite	I	寄存器文件写使能
WData[31:0]	I	寄存器文件写入数据
RData1[31:0]	O	读寄存器文件时的第 1 个寄存器的输出
RData2[31:0]	O	读寄存器文件时的第 2 个寄存器的输出

功能定义

序号	功能名称	功能描述
1	读寄存器	RData1 输出 RS1[4:0]所寻址的寄存器。 RData2 输出 RS2[4:0]所寻址的寄存器。
2	写寄存器	当时钟上升沿到来时，并且 RegWrite 有效时，WData 被写入 RD[4:0]所寻址的寄存器
3	复位	reset=1 时，使 32 个 gpr 单元清零

模块接口

信号名	方向	描述
Clk	I	MIPS-C 处理器时钟
Reset	I	复位信号
RS1[4:0]	I	读寄存器文件时的第 1 个寄存器下标
RS2[4:0]	I	读寄存器文件时的第 2 个寄存器下标
RD[4:0]	I	写寄存器文件时的寄存器下标
RegWrite	I	寄存器文件写使能
WData[31:0]	I	寄存器文件写入数据
RData1[31:0]	O	读寄存器文件时的第 1 个寄存器的输出
RData2[31:0]	O	读寄存器文件时的第 2 个寄存器的输出

功能定义

序号	功能名称	功能描述
1	读寄存器	RData1 输出 RS1[4:0]所寻址的寄存器。 RData2 输出 RS2[4:0]所寻址的寄存器。
2	写寄存器	当时钟上升沿到来时，并且 RegWrite 有效时，WData 被写入 RD[4:0]所寻址的寄存器
3	复位	reset=1 时，使 32 个 gpr 单元清零

分析目标并划分层次

> "在计算机科学中的任何问题都可以通过增加一个间接层来解决“（Any problem in computer science can be solved with another layer of indirection. ）

分层思想，将大化整，以 32 位加法器为例，初拿到手，可能会觉得无所适从，输入是两个 32 位数，输出是一个 32 位数，关系虽然抽象上简单，但是并不能简单地用门电路概括。但是如果我们添加一个“一位加法器”的间接层。使用 1 位加法器拼出一个 32 位加法器比较简单，单独拼出一个 1 位加法器也很简单。如此，问题就解决

电路搭建与测试

> 💡 注重测试的重要性

测试的基本原则就是根据我端口定义和功能定义表，对各种可能的输入情况进行排查，观察是否有与定义违背的情况，或者在输入未定义信号的时候会不会产生一些非常危险的bug

---

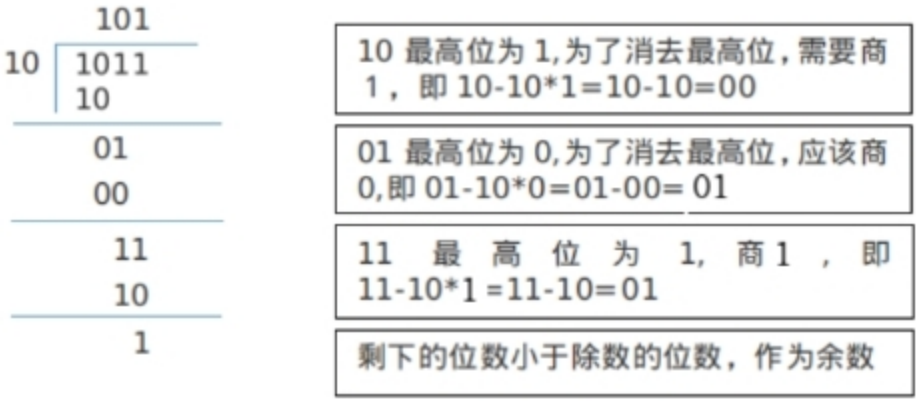
CRC校验码计算电路

（一道例题）

CRC 校验码简介

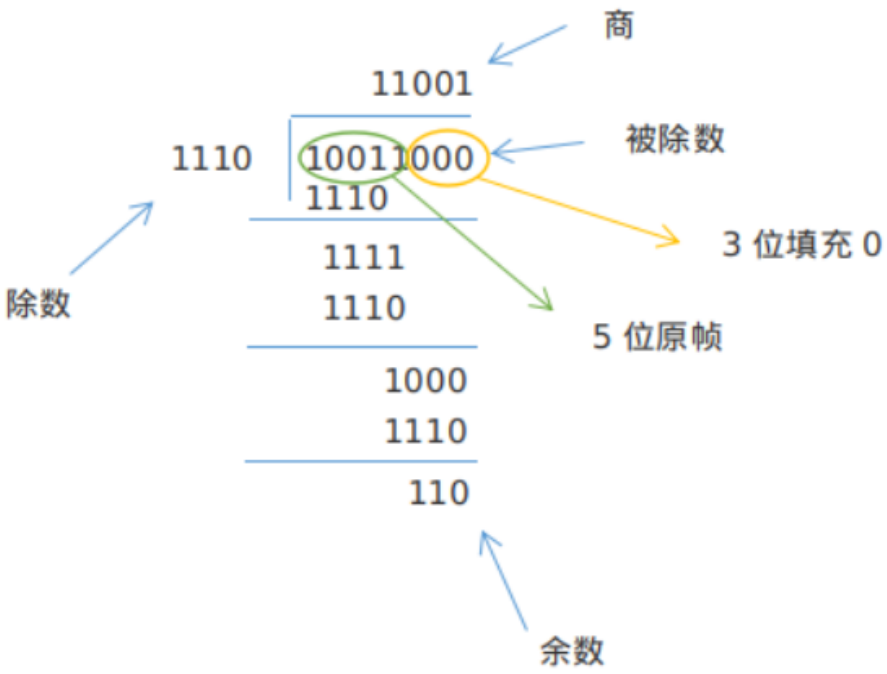
CRC 校验是数据通信领域中最常用的一种查错校验方式，它对数据进行多项式计算，并将得到的结果附在帧的后面，接收设备也执行类似的算法，以保证数据传输的正确性和完整性。

在了解这种校验码怎么计算之前，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`。用除数得到商时，只关心能把最高位消掉就行。对于模二除法，以被除数为 1011，除数为 10 为例，运算过程如下：



不难发现模二除法和异或的操作完全相同，因此可以用**异或**实现

由此，我们得到了CRC校验码的验证方法，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（如果不是三位则需要补前导0），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。

计算电路的定义

Name	Direction	Description
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

分析目标层次化

1. **真值表法永不过时！**

   本身不需要存储状态，输入决定输出（说明了一定可以用真值表）把2^8个原数据帧与对应2^3（除数最高位比为1可以不用管）分别得到的2^{10}个CRC依次输入到仿真，自动生成即可即可

2. 模块化构造

   将这个题目分成两个部分

   1. 设计四位模二除法器
   2. 使用四位模二除法器搭载8位CRC校验码计算电路

   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。

   因此对于这个电路只需要判断被除数最高位然后分情况输出

   后续需要将两个模块连接起来，

   类似于普通除法的过程，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。

课上测试回顾

1. 状态图好好画，把每种状态对应每个输入都会转移到哪交代明白（目前阶段步子还是不能迈的那么大）
2. 时序逻辑掌握不熟练，特别是多个变量的更新上容易犯迷糊