

# **数字逻辑实验报告（2**）

|  |  |  |
| --- | --- | --- |
| **数字逻辑实验2** | | |
| **一、无符号数的乘法器设计50%** | **二、无符号数的除法器设计50%** | **总成绩** |
|  |  |  |

评语：（包含：预习报告内容、实验过程、实验结果及分析）

教师签名

**姓 名： 王明明**

**学 号： U201714726**

**班 级： CS1705**

**指 导 教 师： 何云峰**

**计算机科学与技术学院**

**2019 年 5 月 13 日**



**数字逻辑实验报告（2）**

无符号数的乘法器设计

一、无符号数的乘法器设计

1、实验名称

无符号数的乘法器的设计。

2、实验目的

要求使用合适的逻辑电路的设计方法，通过工具软件logisim进行无符号数的乘法器的设计和验证，记录实验结果，验证设计是否达到要求。

通过无符号数的乘法器的设计、仿真、验证3个训练过程，使同学们掌握数字逻辑电路的设计、仿真、调试的方法。

3、实验所用设备

Logisim2.7.1软件一套。

4、实验内容

**（1）四位乘法器设计**

四位乘法器Mul4🞨4实现两个无符号的4位二进制数的乘法运算，其结构框图如图3-1所示。设被乘数为b(3:0)，乘数为a(3:0)，乘积需要8位二进制数表示，乘积为p(7:0)。

Mul4🞨4

a(3:0)

b(3:0)

p(7:0)

图3-1 四位乘法器结构框图

四位乘法器运算可以用4个相同的模块串接而成，其内部结构如图3-2所示。每个模块均包含一个加法器、一个2选1多路选择器和一个移位器shl。

图3-2中数据通路上的数据位宽都为8，确保两个4位二进制数的乘积不会发生溢出。shl是左移一位的操作，在这里可以不用逻辑器件来实现，而仅通过数据连线的改变（两个分线器错位相连接）就可实现。



图3-2 四位乘法器内部结构

**（2）32🞨4乘法器设计**

32🞨4乘法器Mul32🞨4实现一个无符号的32位二进制数和一个无符号的4位二进制数的乘法运算，其结构框图如图3-3所示。设被乘数为b(31:0)，乘数为a(3:0)，乘积也用32位二进制数表示，乘积为p(31:0)。这里，要求乘积p能用32位二进制数表示，且不会发生溢出。

Mul32🞨4

a(3:0)

b(31:0)

p(31:0)

图3-3 32🞨4乘法器结构框图

在四位乘法器Mul4🞨4上进行改进，将数据通路上的数据位宽都改为32位，即可实现Mul32🞨4。

**（3）32🞨32乘法器设计**

32🞨32乘法器Mul32🞨32实现两个无符号的32位二进制数的乘法运算，其结构框图如图3-4所示。设被乘数为b(31:0)，乘数为a(31:0)，乘积也用32位二进制数表示，乘积为p(31:0)。这里，要求乘积p能用32位二进制数表示，且不会发生溢出。

Mul32🞨32

a(31:0)

b(31:0)

p(31:0)

图3-4 32🞨32乘法器结构框图

用32🞨4乘法器Mul32🞨4作为基本部件，实现32🞨32乘法器Mul32🞨32。

设被乘数为b(31:0)=(b31b30b29b28**···**b15b14b13b12**···**b4b3b2b1b0)2

乘数为a(31:0)=(a31a30a29a28**···**a15a14a13a12**···**a3a2a1a0)2

=(a31a30a29a28)2🞨228+**···**+ **(** a15a14a13a12)2🞨212+**···**+ (a3a2a1a0)2🞨20

所以，

p(31:0)= b(31:0) 🞨 a(31:0)

= b(31:0) 🞨((a31a30a29a28)2🞨228+**···**+ **(** a15a14a13a12)2🞨212+**···**+ (a3a2a1a0)2🞨20)

= b(31:0) 🞨(a31a30a29a28)2🞨228 +**···**+ b(31:0) 🞨 **(** a15a14a13a12)2🞨212 +**···**

+ b(31:0) 🞨 (a3a2a1a0)2🞨20

从上述推导可知，Mul32🞨32可以用8个Mul32🞨4分组相乘，然后通过4的倍数位的左移（相当于乘2i），再将左移结果两两相加得到。

5、实验设计方案

**（1）四位乘法器设计**

（1）设计思路：

可将两个四位二进制数相乘看作是加法运算和移位运算，由于输出端位宽为八，且不会溢出，因此不妨将被加数由四位扩展为八位，使用多路选择器对加数的每一位进行判断，如果加数的一位是0，则被加数加上常量0，并左移一位；如果加数的一位是1，则被加数加上上一步的结果。

假设被加数是(a3a2a1a0)2，加数是(b3b2b1b0)2，那么乘法算式为：(a3a2a1a0)2\* b0+(a3a2a1a0)2(1🡨)\* b1+(a3a2a1a0)2(2🡨)\* b2+(a3a2a1a0)2(3🡨)\* b3

（2）使用logism软件绘制电路图：

图3-5为四位乘法器电路图

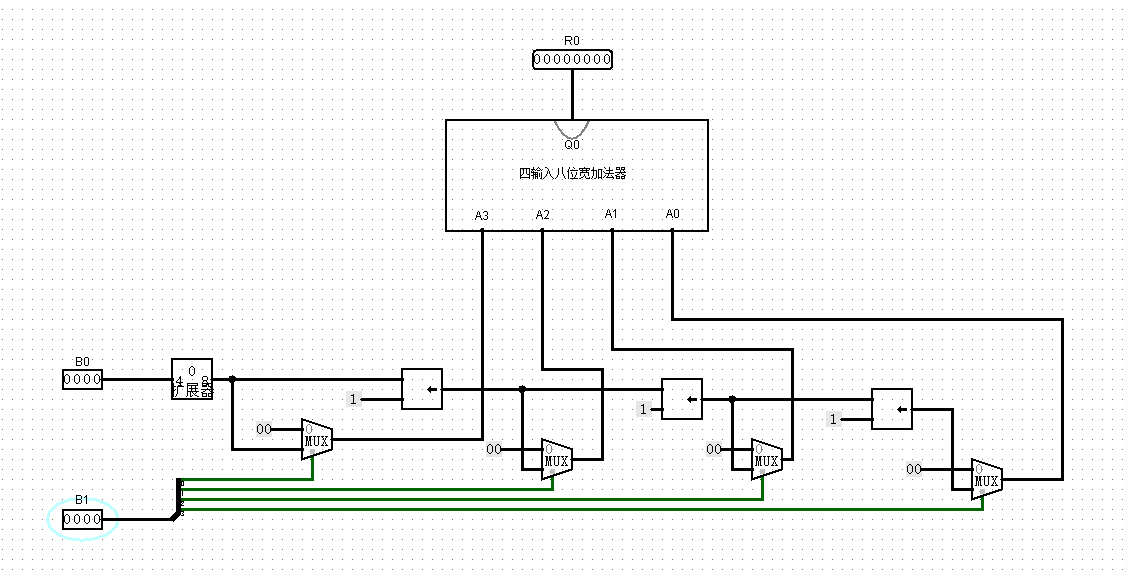


图3-5 无符号四位乘法器

**（2）32🞨4乘法器设计**

（1）设计思路：

在无符号四位乘法器的基础上进行改进，不妨设被乘数设被乘数为b(31:0)：(b31b30b29b28**···**b15b14b13b12**···**b4b3b2b1b0)2乘数为a(3:0)：(a3a2a1a0)2乘积也用32位二进制数表示，乘积为p(31:0)，不考虑溢出。与两个四位二进制数相乘类似，被加数每次左移1位，根据乘数0或者1进行选择操作。

乘法算式为：

(b31b30b29b28**···**b15b14b13b12**···**b4b3b2b1b0)2 \* a0 + (b31b30b29b28**···**b15b14b13b12**···**b4b3b2b1b0)2 (1🡨)\* a1 + (b31b30b29b28**···**b15b14b13b12**···**b4b3b2b1b0)2 (2🡨)\* a2 + (b31b30b29b28**···**b15b14b13b12**···**b4b3b2b1b0)2 (3🡨)\* a3

（2）使用logisim软件绘制电路图。

图3-6为32\*4位乘法器电路图：

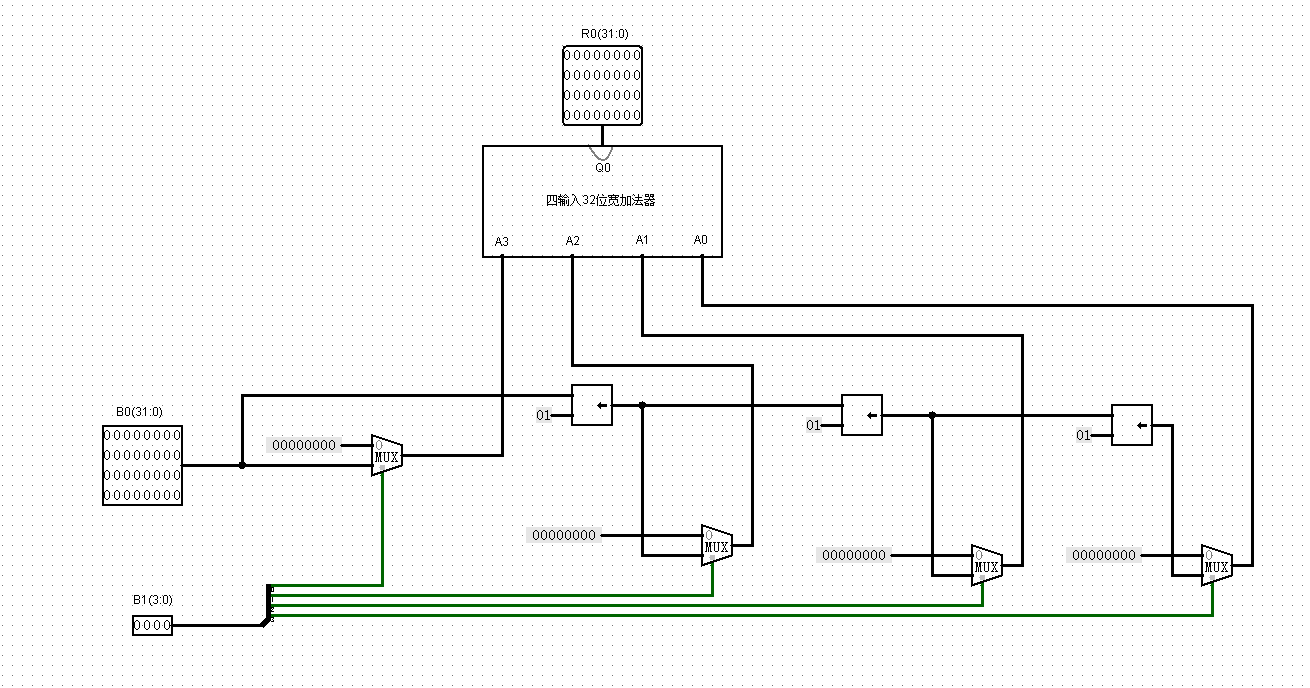


图3-6 无符号32\*4乘法器

**（3）32🞨32乘法器设计**

（1）设计思路：

用32🞨4乘法器Mul32🞨4作为基本部件，实现32🞨32乘法器Mul32🞨32。

设被乘数为b(31:0)=(b31b30b29b28**…**b15b14b13b12**…**b4b3b2b1b0)2

乘数为a(31:0)=(a31a30a29a28**…**a15a14a13a12**…**a3a2a1a0)2

=(a31a30a29a28)2🞨228+**…**+ **(** a15a14a13a12)2🞨212+**…**+ (a3a2a1a0)2🞨20

所以，

p(31:0)= b(31:0) 🞨 a(31:0)

= b(31:0) 🞨((a31a30a29a28)2🞨228+**…**+ **(** a15a14a13a12)2🞨212+**…**+ (a3a2a1a0)2🞨20)

= b(31:0) 🞨(a31a30a29a28)2🞨228 +**···**+ b(31:0) 🞨 **(** a15a14a13a12)2🞨212 +**…**+ b(31:0) 🞨 (a3a2a1a0)2🞨20

可知，Mul32🞨32可以用8个Mul32🞨4分组相乘，然后通过4的倍数位的左移，再将左移结果两两相加得到。

（2）使用logisim软件绘制电路图。

图3-7为32\*32位乘法器电路图：

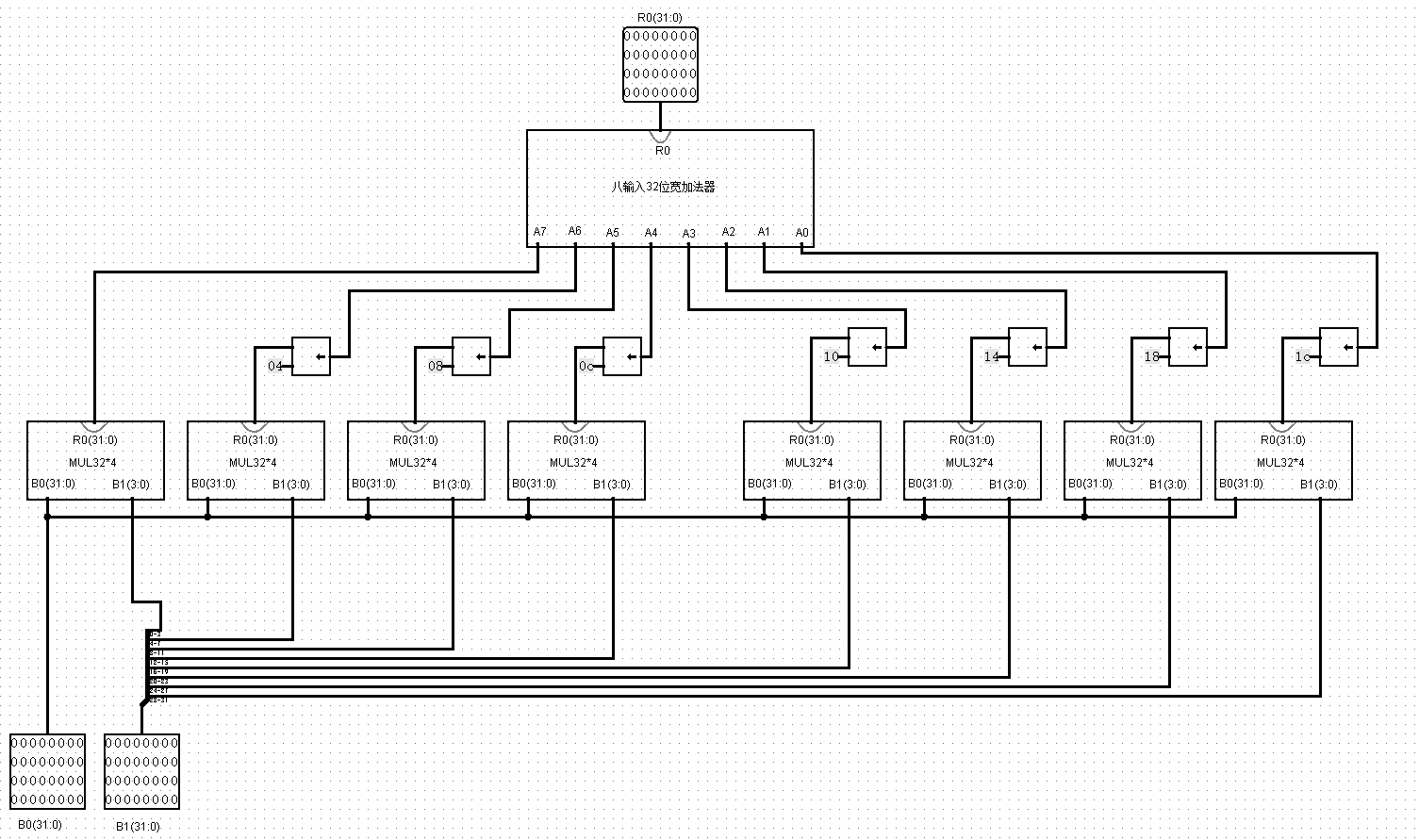
****

图3-7 32\*32位乘法器

6、实验结果记录

根据实验方案设计要求，对于相应的乘法器和除法器，在给定的输入条件下，填写表3-1。

表3-1 无符号数的乘法器实验结果记录表

|  |  |  |  |
| --- | --- | --- | --- |
| 电路 | 输入1（16进制） | 输入2（16进制） | 输出（16进制） |
| Mul4🞨4 | b=0🞨A | a=0🞨A | p=0X64(100) |
| Mul4🞨4 | b=0🞨E | a=0🞨9 | p=0X7E(126) |
| Mul32🞨4 | b=0🞨003ABEF1 | a=0🞨A | p=0X024B756A |
| Mul32🞨4 | b=0🞨019ABEF1 | a=0🞨7 | p=0X0B3B3897 |
| Mul32🞨32 | b=0🞨0002BEF1 | a=0🞨00004EF1 | p=0XD8C32EE1 |
| Mul32🞨32 | b=0🞨00003EF1 | a=0🞨0003BEF1 | p=0XEBC51EE1 |

当Mul4🞨4输入b为0🞨A，a为0🞨A时，输出为0X64(100)

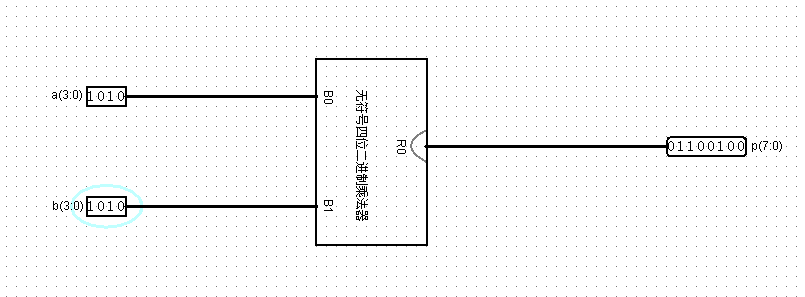


图3-8 Mul4🞨4截图

当Mul4🞨4输入b为0🞨E，a为0🞨9时，输出为0X7E(126)

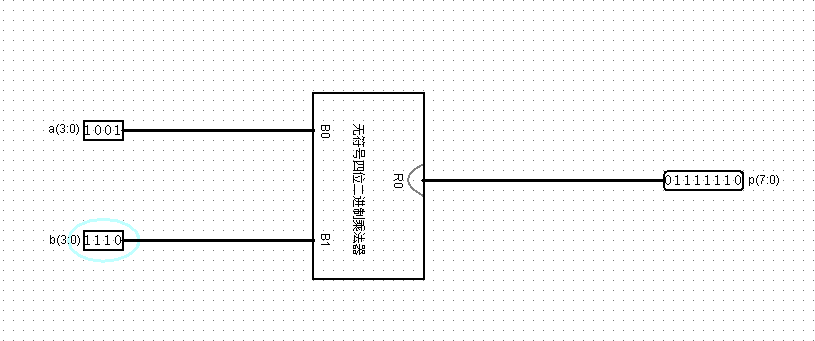


图3-9 Mul4🞨4截图

当Mul32🞨4输入b为0🞨003ABEF1，a为0🞨A时，输出为0X024B756A

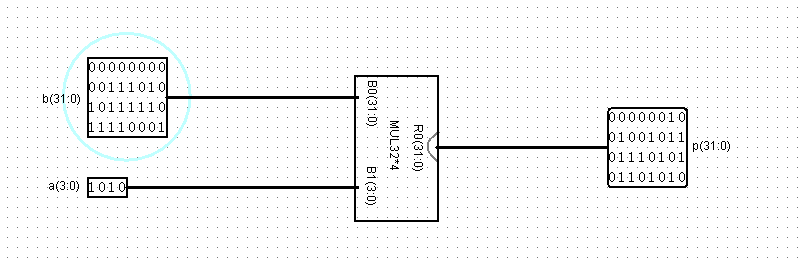


图3-10 Mul32🞨4截图

当Mul32🞨4输入b为0🞨019ABEF1，a为0🞨7时，输出为0X0B3B3897

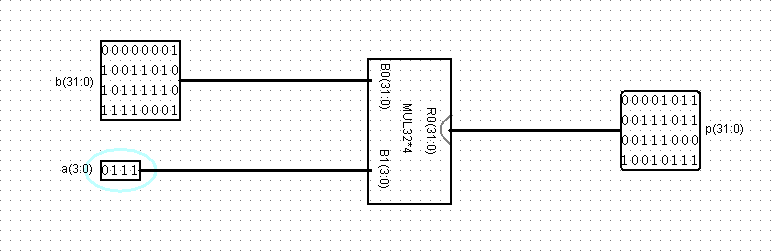


图3-11 Mul32🞨4截图

当Mul32🞨32输入b为0🞨0002BEF1，a为0🞨00004EF1时，输出为0XD8C32EE1

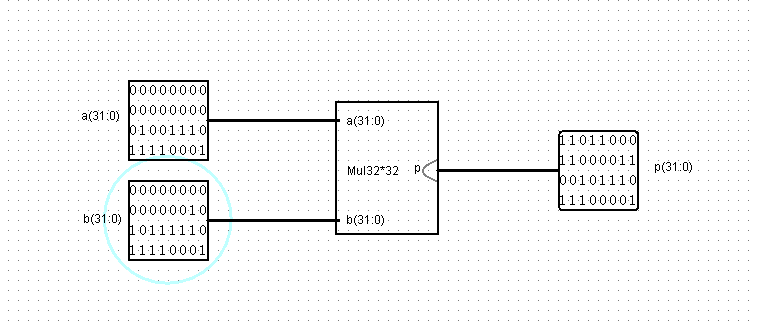


图3-12 Mul32🞨32截图

当Mul32🞨32输入b为0🞨00003EF1，a为0🞨0003BEF1时，输出为0XEBC51EE1

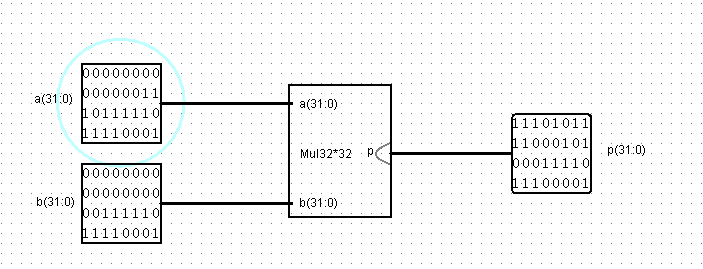


图3-13 Mul32🞨32截图



**数字逻辑实验报告（3）**

无符号数的除法器设计

二、无符号数的除法器设计

1、实验名称

无符号数的除法器的设计。

2、实验目的

要求使用合适的逻辑电路的设计方法，通过工具软件logisim进行无符号数的除法器的设计和验证，记录实验结果，验证设计是否达到要求。

通过无符号数的除法器的设计、仿真、验证3个训练过程，使同学们掌握数字逻辑电路的设计、仿真、调试的方法。

3、实验所用设备

Logisim2.7.1软件一套。

4、实验内容

**（1）四位除法器设计**

四位除法器实现两个无符号的4位二进制数的除法运算，其结构框图如图2-1所示。设被除数为n2(3:0)，除数为d(3:0)，商为quot(3:0)，余数为rem(3:0)。

Div4

n2(3:0)

d(3:0)

quot(3:0)

rem(3:0)

图4-1 四位除法器结构框图

四位除法器Div4算法步骤如下：

1. 设n1="0000"，将被除数以n1:n2 的形式拼接，除数为d；
2. 重复4次：

将n1:n2左移1位；

if (n1>d) begin n1= n1-d; n2 (0)=1 end

1. 商和余数的结果为：quot= n2 ；rem= n1 。

四位除法器也可以用4个相同的模块串接而成。每个模块均包含一个减法器、两个2选1多路选择器、一个比较器和一个移位器shl。请参照四位乘法器的设计思路，实现两个无符号的4位二进制数的除法器。

**（2）32位除法器设计**

32位除法器Div32实现两个无符号的32位二进制数的除法运算，其结构框图如图2-2所示。设被除数为n(31:0)，除数为d(31:0)，商为quot(31:0)，余数为rem(31:0)。

Div32

n(31:0)

d(31:0)

quot(31:0)

rem(31:0)

图4-2 32位除法器结构框图

对四位除法器Div4中4个相同的模块之一进行改进，将数据通路上的数据位宽都扩展为32位，得到一个Div1。将32个Div1拼接起来即可实现Div32。

5、实验设计方案

**（1）四位除法器设计**

设计思路：首先将被除数扩展为8位并左移一位，将其高四位与除数进行比较，如果被除数高四位小于除数，则商0，新的8位数不变；如果被除数高四位小于除数，则商1，新产生的高四位为原先的高四位减去除数，低四位为原先的低四位；紧接着左移一位重复上述的比较操作。

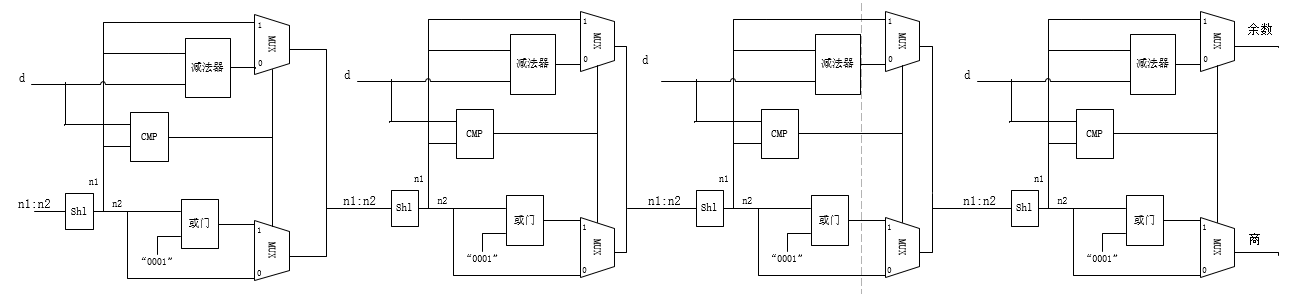


图4-3 四位除法器内部模块

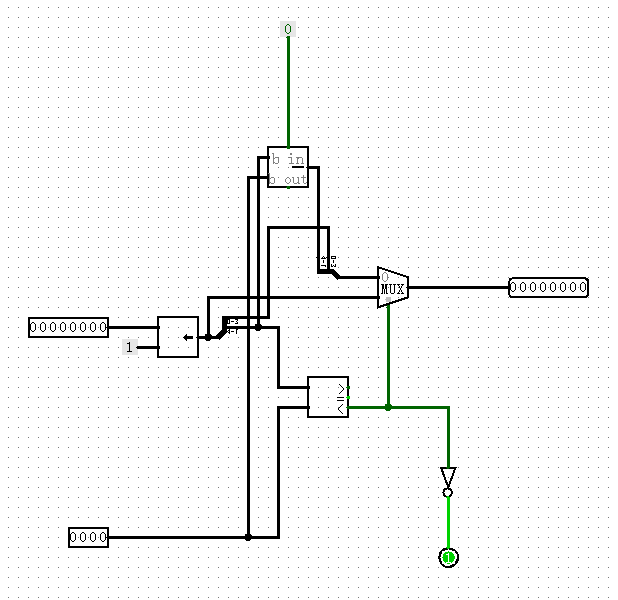


图4-4 四位除法器内部模块截图

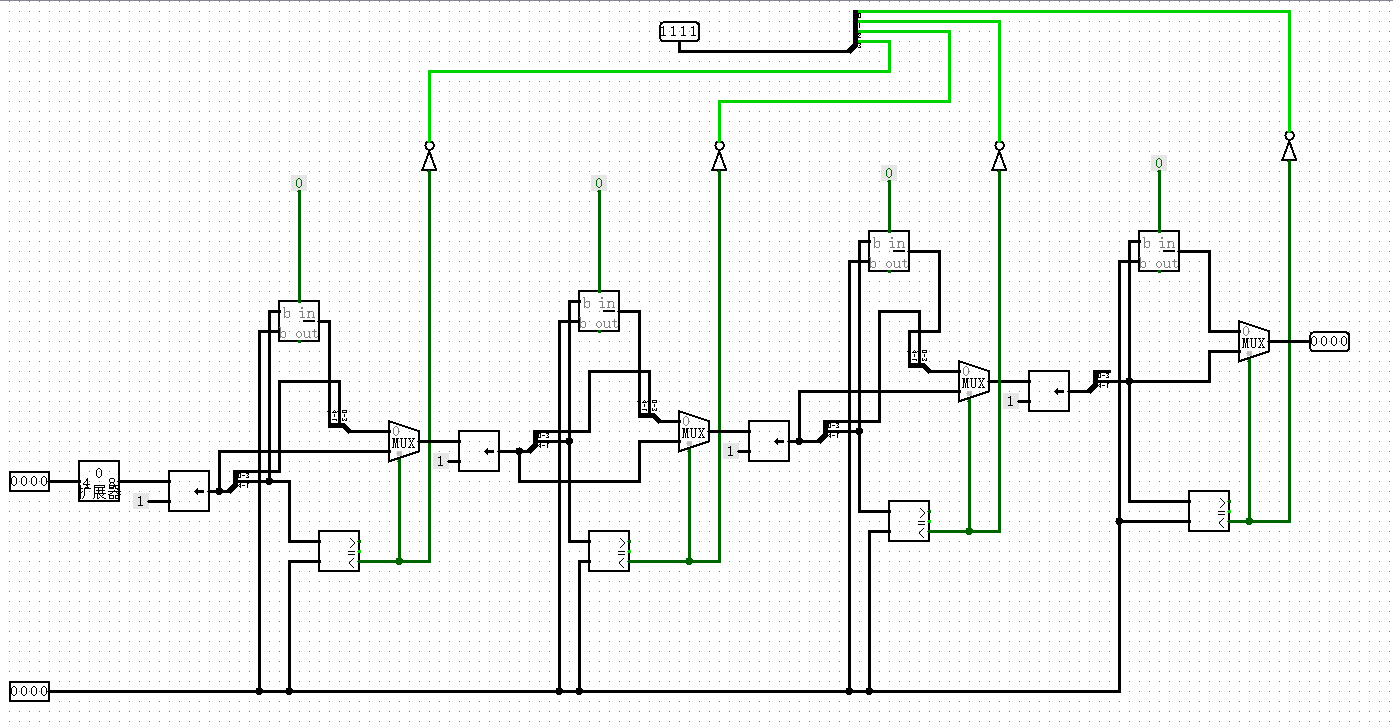


图4-5 无符号四位二进制除法器

**（2）32位除法器Div32设计**

设计思路：由于器件最多支持32位，若按照四位除法器的思路，则先应该将32位扩展为34位，为了解决这个问题，可考虑拼接。即将32位全为0的常量作为被除数的高32位，取被除数的最高位作为32位常量0的最低位，以此拼接成32位的效果，通过左移操作，不断地将被除数的高32位和低32位进行更新变动，比较操作与四位除法器类似。

DIV1模块如下：

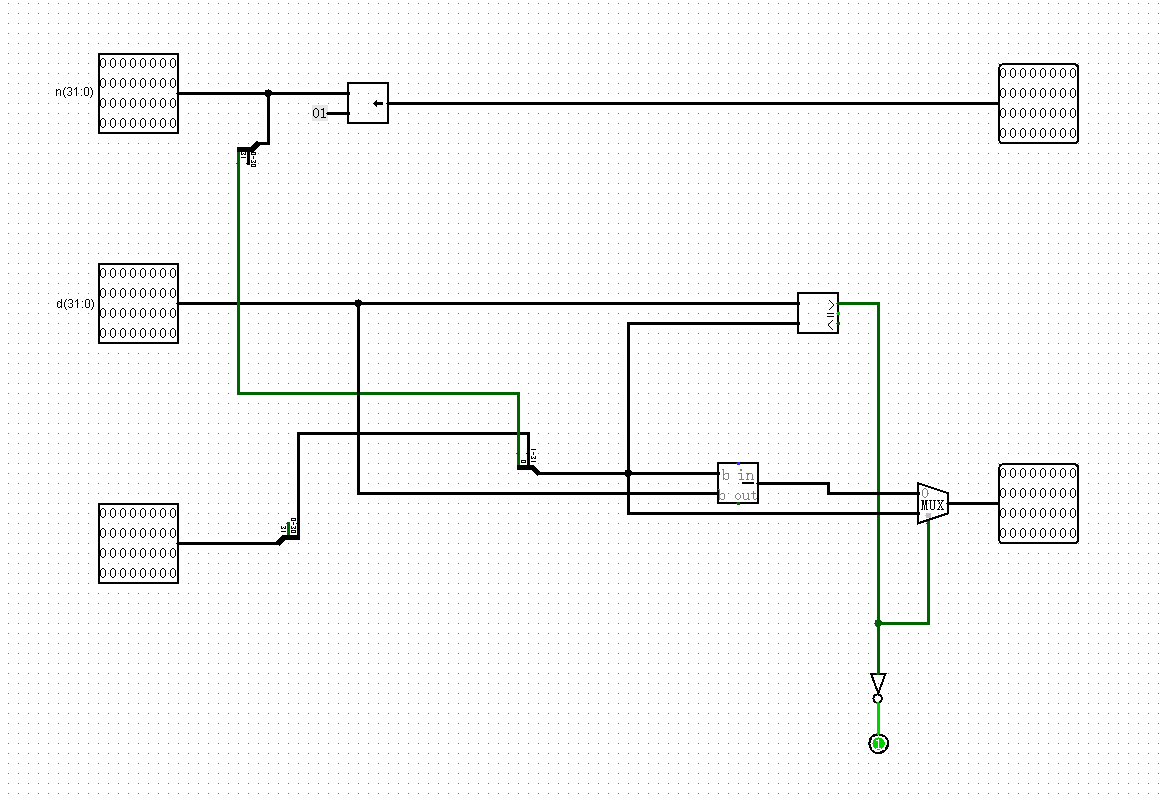


图4-6 无符号32位除法器内部模块

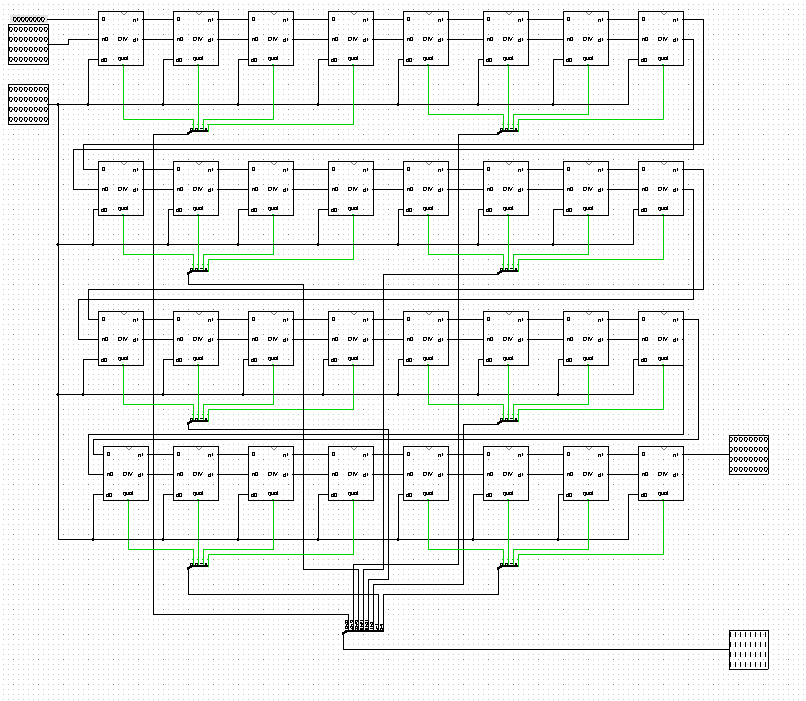


图4-7 无符号32位除法器

6、实验结果记录

根据实验方案设计要求，对于相应的乘法器和除法器，在给定的输入条件下，填写表4-1。

表4-1 无符号数的乘法器实验结果记录表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 电路 | 输入1（16进制） | 输入2（16进制） | 输出（16进制） | |
| Div4 | n2=0🞨E | d=0🞨9 | quot=0x1 | rem=0x5 |
| Div4 | n2=0🞨E | d=0🞨0 | quot=0xF | rem=0xE |
| Div32 | n=0🞨019ABEF1 | d=0🞨00004EF1 | quot=0x534 | rem=0xFD |
| Div32 | n=0🞨A0504EF1 | d=0🞨019ABEF1 | quot=0x63 | rem=0x17877BE |

当DIV4电路，输入n2为0🞨E，d为0🞨9时，输出quot为0x1，rem为0x5，如图4- 所示

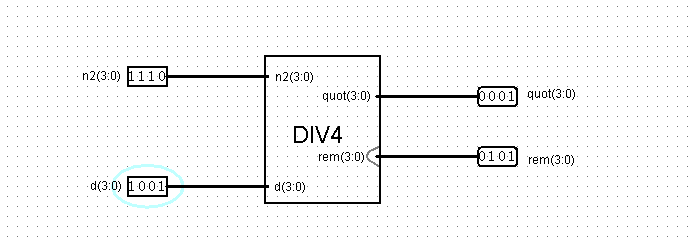


图4-8 Div4截图

当DIV4电路，输入n2为0🞨E，d为0🞨0时，输出quot为0xF，rem为0xE，如图4- 所示

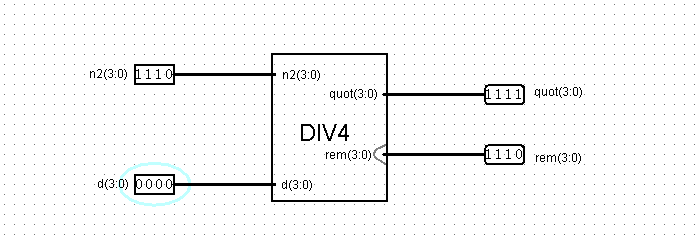


图4-9 Div4截图

当Div32电路，输入n2为0🞨019ABEF1，d为0🞨00004EF1时，输出quot为0x534，rem为0xFD，如图4- 所示

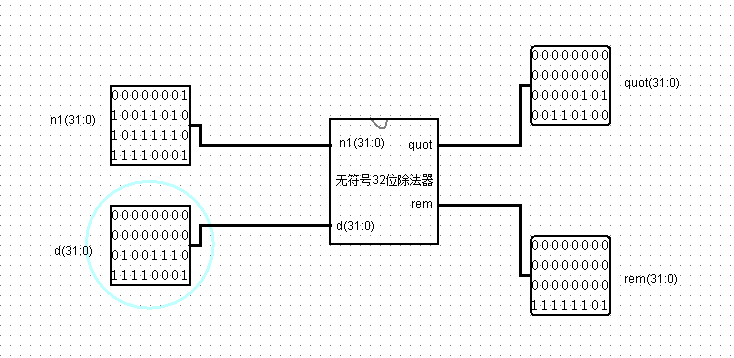


图4-10 Div32截图

当Div32电路，输入n2为0🞨A0504EF1，d为0🞨019ABEF1时，输出quot为0x63，rem为0x17877BE，如图4- 所示

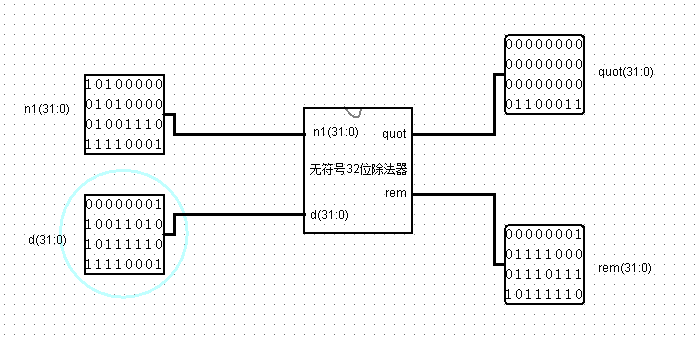


图4-11 Div32截图

7、实验中遇到的问题及解决方法

（1）故障1

问题描述：比较器默认设置为“关于2的补码”，导致最高位为1时出错

问题分析：由于出错比较有特点，均是当两个要比较的数字可以作为有符号时出错，即最高位为1时才出错。

解决方法：将比较器设置为“无符号”。

（2）故障2

问题描述：32位除法器存在明显震荡

问题分析：这个问题在QQ群里出现了很多次，即使有些同学没有使用多路选择器也会出现明显震荡，因此分析这个问题比较普遍。

解决方法：后来在室友的帮助下，得知Ctrl+E可以消除震荡。

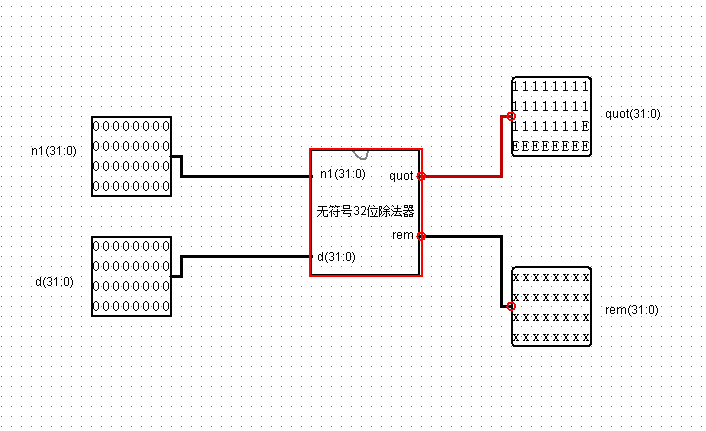


图4-12 明显震荡的封装

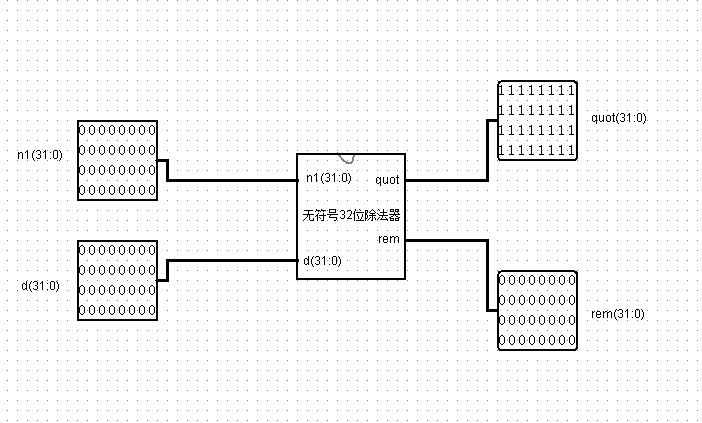


图4-13 消除震荡后的电路封装

8、思考题

（1）乘法器/除法器中的延时主要取决于加法器/减法器的延时，其它组件延时可忽略不计。假设每个加法器/减法器的延时都为Δt，你所设计的乘法器Mul4🞨4、Mul32🞨4、Mul32🞨32、除法器Div4、Div32的延时各是多少？它们是组合逻辑电路、同步时序逻辑电路还是异步时序逻辑电路？

无符号四位乘法器中有四个小模块串行连接，因此延迟为4Δt；无符号32\*4乘法器也是四个模块串联，因此延迟也为4Δt；无符号32\*32乘法器是四个32\*4模块串联再接加法器，因此延迟为7Δt。

无符号四位除法器是4个模块串联，延迟为4Δt；无符号32位除法器是32个DIV1小模块串联，因此延迟为32Δt。

（2）通过改变设计，乘法器Mul32🞨32的延时能不能再减少？如果能减少，它的最小值是多少？

应该可以通过将串行连接改为并行连接的方式降低延迟，类似于无符号32\*32乘法器，延迟最低降为7Δt。

（3）目前的除法器Div32性能并不好，你是否有性能更好设计？

因为32位除法器需要有32个DIV1模块串联，不仅会有延迟，还会产生震荡问题，因此性能并不太好。猜测如果使用并行连接方式可能会提高性能，降低延迟。

9、心得体会、意见与建议

这次实验相比于第一次实验，难度提升了很多。但是在课本上有介绍无符号四位二进制乘法器的例子，并且有比较详细的讲解，为了完成实验，我翻了很多次课本，加深了自己对乘法器的理解。与无符号四位乘法器类似，另外两个乘法器的设计就相对变得简单了。在设计乘法器的过程中，我多次向同学请教，十分感谢同学们对我的帮助。除法器与乘法器类似，相应的加法变成了减法。个人觉得最难的是32位除法器的设计，因为器件本身最大仅为32位，按照之前的设计思路很容易进入死胡同。感谢学委的帮助，提醒我可以用常量表示被除数的高32位。

另外，在这次实验之后，我觉得在进行设计之后应该积极检查自己的设计是否正确，而不是到检查的时候再由老师检查，因为到了检查的时候如果自己出错了很难在短时间内改出来，正确做法应该是自己在检查之前做好测试，早发现错误早改正，节省时间。