

# **数字逻辑实验报告（2**）

|  |  |  |
| --- | --- | --- |
| **数字逻辑实验3** | | |
| **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： 王占成**

**学 号： U201714786**

**班 级： CS1707**

**指 导 教 师： 胡迎松**

**计算机科学与技术学院**

**2019年5月17日**



**数字逻辑实验报告**

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1-1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1-1 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1-2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1-2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1-2 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1-3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1-3 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1-4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1-4 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**

若乘数为a=a3a2a1a0 ,乘数b=b3b2b1b~~0~~ 计算p(7:0)=a×b时具体可看做乘数移位再相加的过程具体过程如下：

P= a3a2a1a0 × b3b2b1b~~0~~

即： a3 a2 a1 a0

× b3 b2 b1 b0

0 0 0 0 b3a0 b2a0 b1a0 b0a0

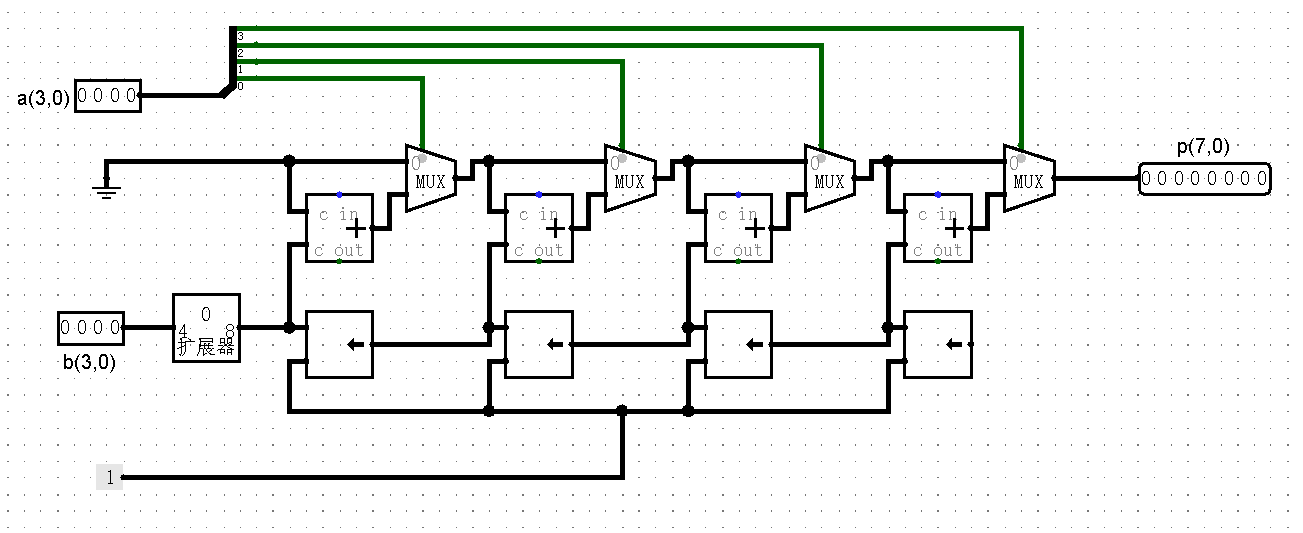
0 0 0 b3a1 b2a1 b1a1 b0a1 0

0 0 b3a2 b2a2 b1a2 b0a2 0 0

0 b3a3 b2a3 b1a3 b0a3 0 0 0

通过上面的运算过程可以知道，四位二进制乘法计算过程可看成乘数b的移位并相加的结果。将b扩展成一个8位二进制数，高四位为0，当与乘数a中的一位相乘之前，就将b向左移位一次。若an为1，则保留本次的加法结果，用来和下一次的移位后的b相加；若an为0，则放弃本次的加法结果，使用上一次的加法结果进行下一次相加。依次进行四次便完成了相应的功能。

Logisim电路图如图1-5所示

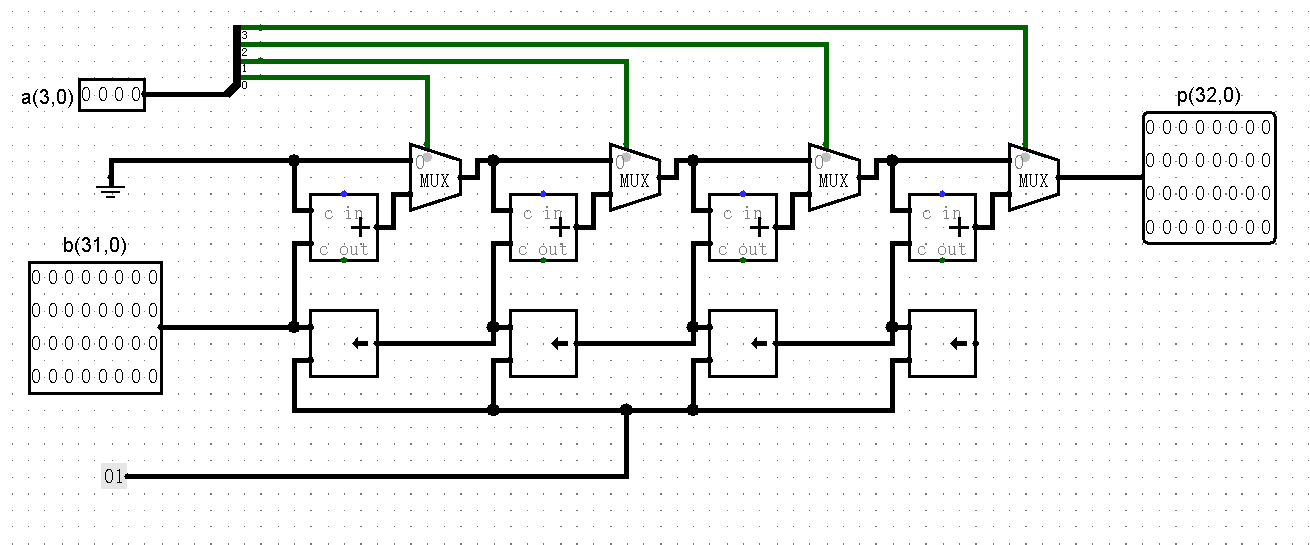


**图1-5 四位乘法器**

**（2）32🞨4乘法器设计**

假定乘数a=a3a2a1a0 ,乘数b= b31b30b29b28**···**b15b14b13b12**···**b3b2b1b0 计算p(31:0)=a×b时计算过程与4×4乘法器相同，都是为b的移位和相加过程。这里，要求乘积p能用32位二进制数表示，且不会发生溢出，所以在移位时不需要考虑将b向高位扩展时会溢出的情况，所以算法逻辑与4×4位乘法器相同。

Logisim电路图如图1-6所示：



**图1-6** **32🞨4乘法器**

**（3）32🞨32乘法器设计**

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)

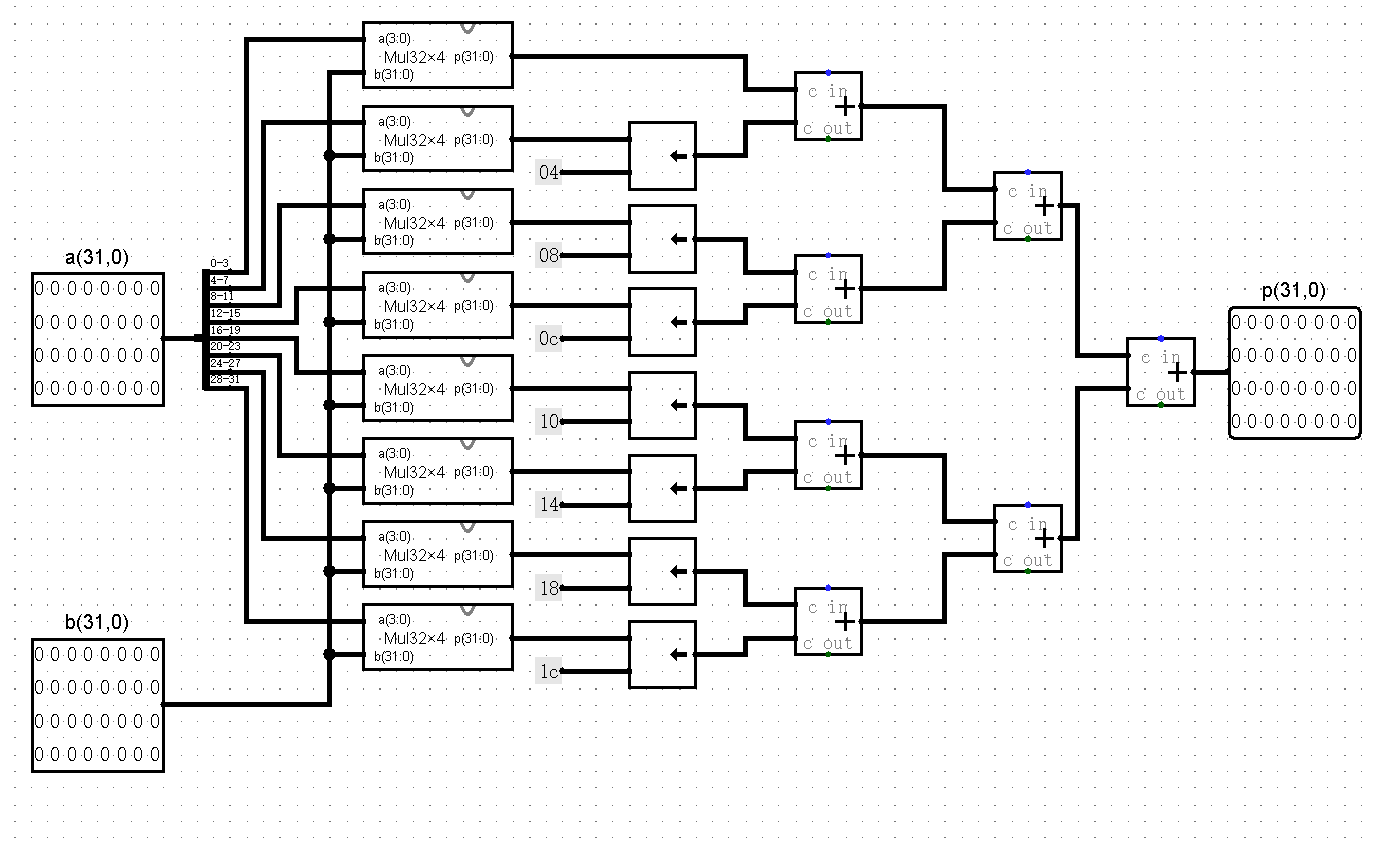
= b(31:0) 🞨((a31a30a29a28)2🞨228+**···**+ **(** a15a14a13a12)2🞨212+**···**+ (a3a2a1a0)2🞨20)

= 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。

Logisim电路图如图1-7所示。



**图1-7 32🞨32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1-1。

表1-1 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 |
| Mul4🞨4 | b=0🞨E | a=0🞨9 | p=0x7E |
| Mul32🞨4 | b=0🞨003ABEF1 | a=0🞨A | p=0x024B 756A |
| Mul32🞨4 | b=0🞨019ABEF1 | a=0🞨7 | p=0x0B3B 3897 |
| Mul32🞨32 | b=0🞨0002BEF1 | a=0🞨00004EF1 | p=0xD8C3 2EE1 |
| Mul32🞨32 | b=0🞨00003EF1 | a=0🞨0003BEF1 | p=0x0EBC5 1EE1 |



**数字逻辑实验报告**

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)

2-1 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)

2-2 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**

除法器Div4的内部逻辑结构框图

四位除法器设计logisim电路图如下列图所示：

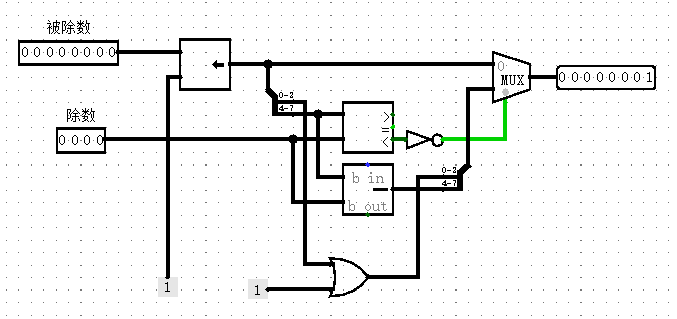


图2-3 四位除法器除法单元

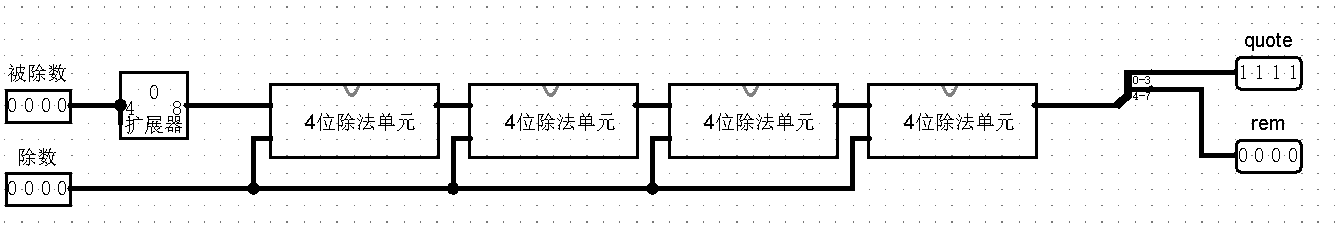


图2-4 四位除法器

**（2）32位除法器Div32设计**

32位除法器Div4算法步骤如下：

1. 设n1="0…0"（共32位），将被除数以n1:n2 的形式拼接，除数为d；
2. 重复32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 。

32位除法器设计logisim电路图如下列图所示：

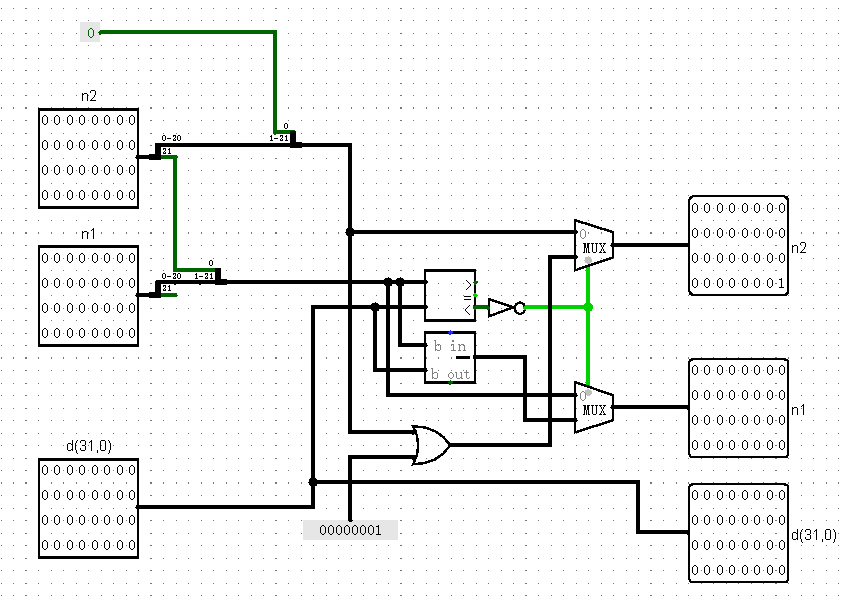


图2-5 32位除法器除法单元

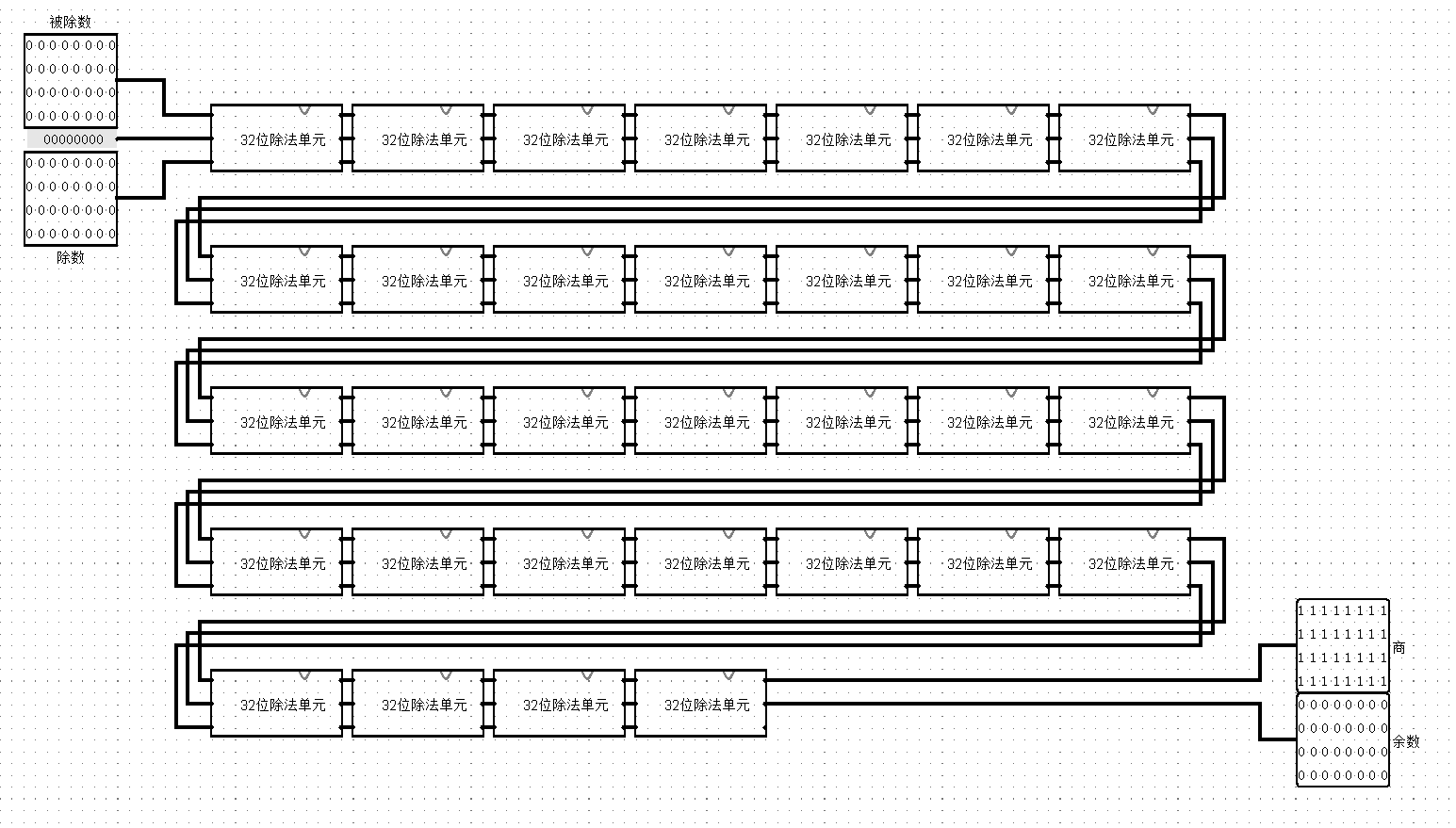


图2-6 32位除法器

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2-1。

表2-1 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 |

三、实验中遇到的问题及解决方法

（1）故障1

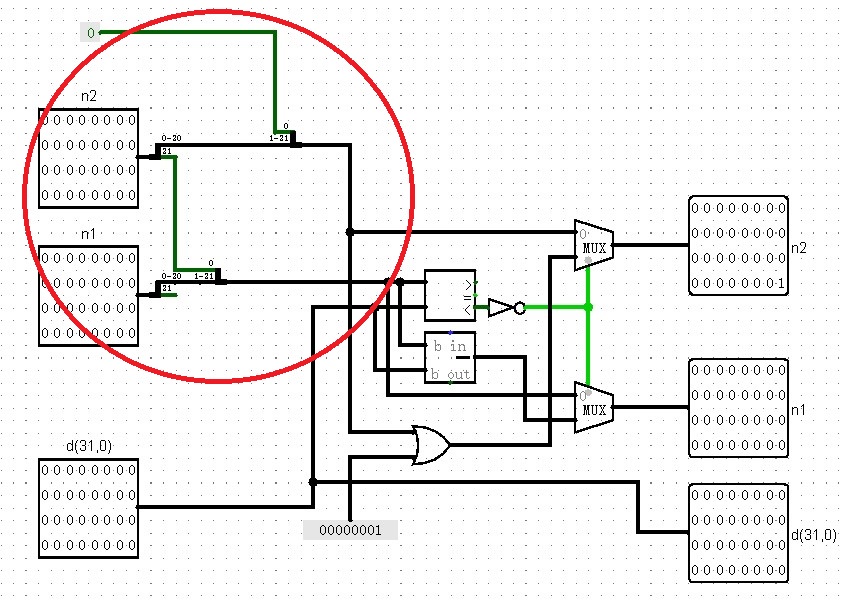
问题描述：32位除法器n1:n2表示为64位，左移并进行比较过程难以实现

问题分析： 将n1:n2左移1位；

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

在这里需要对前32位和后32位进行操作，所以可以将64位左移以后分开实现。

解决方法：在这里可以用分离器交错一位并将最低位补0来实现，然后分为两个输出进行比较。实现如下：



四、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？

答：各器件延时如表4-1所示。设计的乘法器和除法器均为组合逻辑电路。

表2-2 各器件延时表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 器件 | Mul4\*4 | Mul32\*4 | Mul32\*32 | Div4 | Div32 |
| 延时(Δt) | 4 | 4 | 7 | 8 | 32 |

（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？

答：可以，最低5Δt。

（3）目前的除法器Div32性能并不好，你是否有性能更好设计？

答：可以首先比较被除数和除数的大小。若被除数的小于除数，那么商直接置0，余数就为被除数。

若被除数大于除数，可以再次进行本次实验的设计，运用相同的方法计算商以及余数。

五、心得体会、意见与建议

本次电路设计比较简单，根据模板中的内部框架图以及不同运算器之间的关系可以比较容易的设计出本次实验所需要设计的电路。

这次实验中需要注意的是电路的效率以及延迟的考虑，在实际情况下，考虑的东西比我们做实验时考虑的东西更多。譬如并行与串行的考虑，电路门延迟的考虑等，在以后的生活中，我们还要不断提高。