# 设计原理

根据公式：被除数=商数·除数+余数，将被除数减去除数，每减一次，商的中间结果加一，直至被除数等于或者小于除数，运算完成，此时，被除数被减剩的结果就是余数。

根据以上的思想，设计由加法器和减法器组成的除法器。

由于是任意四位除法器，所以输入的被除数和除数存在有符号和无符号的情况，设计中引入一个sign信号，当该信号被置为‘1’时，进行的是有符号运算，被置为‘0’时，进行的是无符号的运算。

# 设计流程图

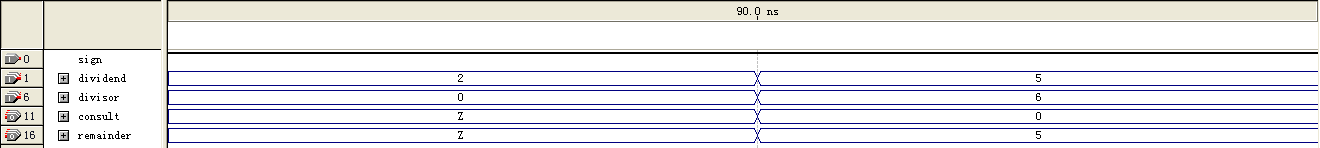


# 有符号运算

当sign信号为‘1’时，运算单元进行的是有符号的运算。

## 除数为零

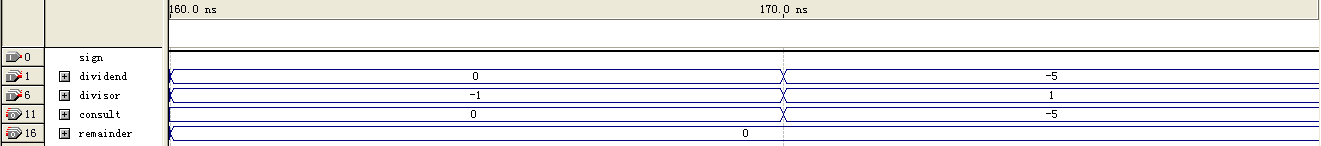
当除数为0，被除数为2时，运算的仿真结果如下：



由于除数为零是不符合除法运算的规则的，所以输出为‘Z’，为高阻态，标志没有输出结果。

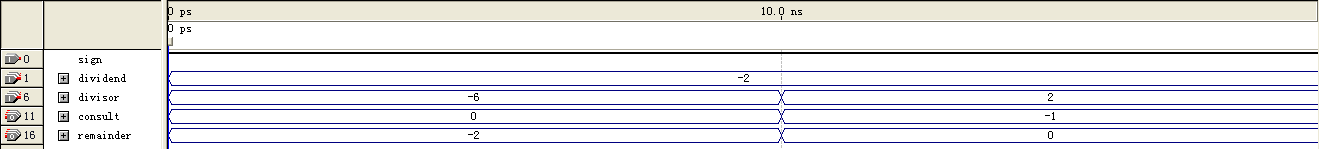
## 被除数为零

当被除数为0，除数为-1时，运算的仿真结果如下：



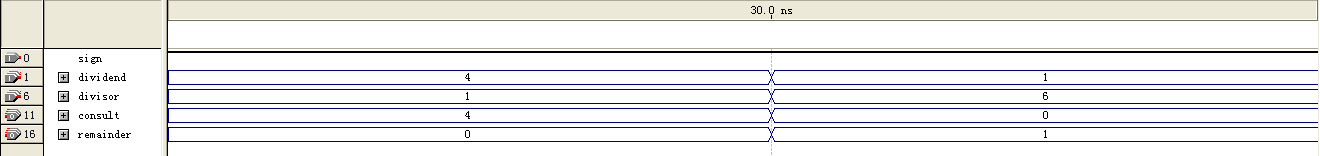
输出的结果为，商为0，余数也为0，结果正确。

## 其他情况



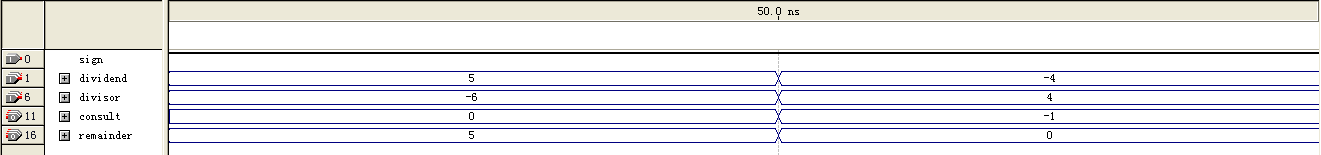
被除数为-2，除数为-6时，商为0，余数也为-2，结果正确。

被除数为-2，除数为2时，商为-1，余数也为0，结果正确。



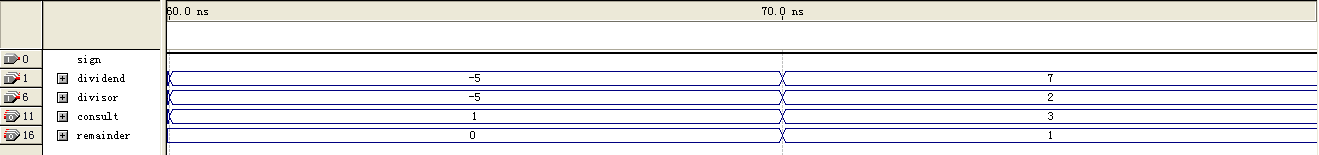
被除数为4，除数为1时，商为4，余数也为0，结果正确。

被除数为1，除数为6时，商为0，余数也为1，结果正确。



被除数为5，除数为-6时，商为0，余数也为5，结果正确。

被除数为-4，除数为5时，商为-1，余数也为0，结果正确。



被除数为-5，除数为-5时，商为1，余数也为0，结果正确。

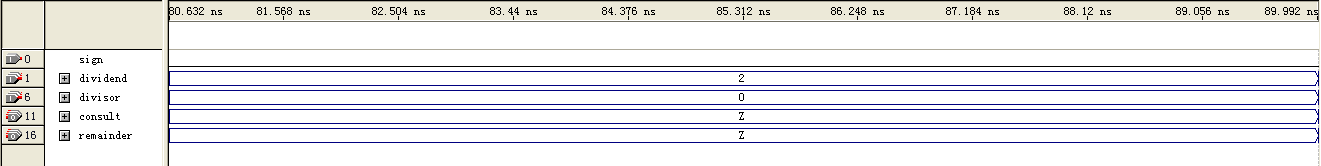
被除数为7，除数为2时，商为3，余数也为1，结果正确。

# 无符号运算

当sign信号为‘0’时，运算单元进行的是有符号的运算。

## 除数为零

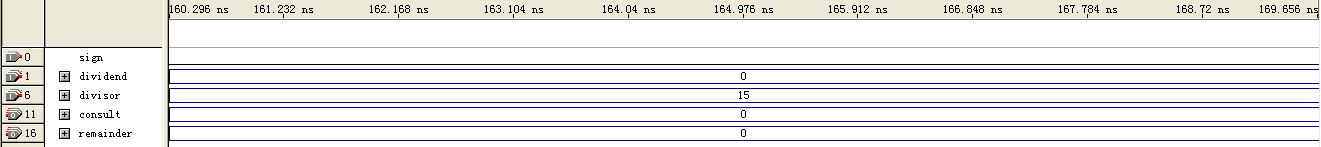
当除数为0，被除数为2时，运算的仿真结果如下：



由于除数为零是不符合除法运算的规则的，所以输出为‘Z’，为高阻态，标志没有输出结果。

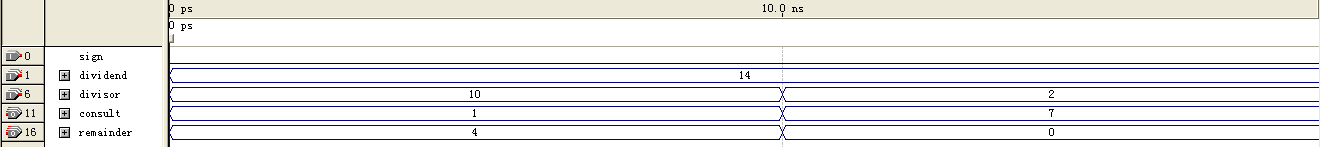
## 被除数为零

当被除数为0，除数为15时，运算的仿真结果如下：



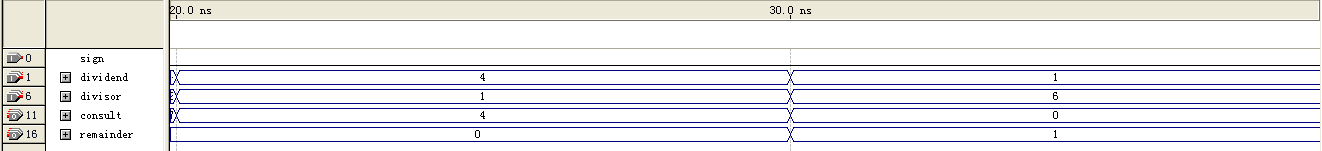
输出的结果为，商为0，余数也为0，结果正确。

## 其他情况



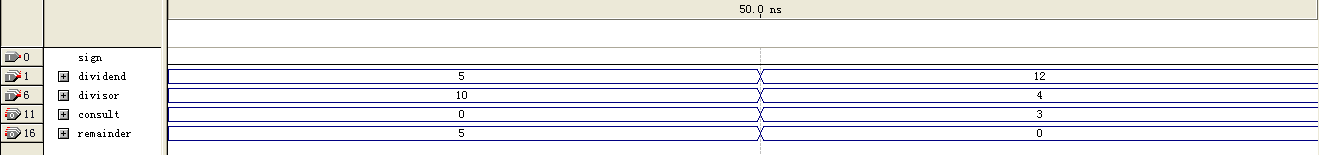
被除数为14，除数为10时，商为1，余数也为4，结果正确。

被除数为14，除数为2时，商为7，余数也为0，结果正确。



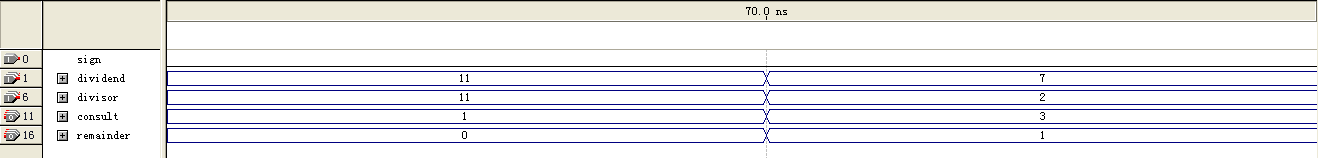
被除数为4，除数为1时，商为4，余数也为0，结果正确。

被除数为1，除数为6时，商为0，余数也为1，结果正确。



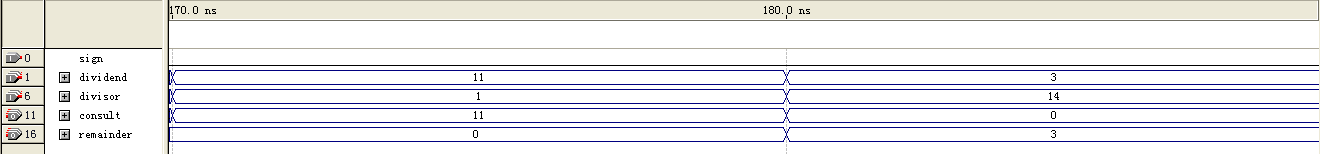
被除数为5，除数为10时，商为0，余数也为5，结果正确。

被除数为14，除数为4时，商为3，余数也为0，结果正确。



被除数为11，除数为11时，商为1，余数也为0，结果正确。

被除数为7，除数为2时，商为3，余数也为1，结果正确。



被除数为11，除数为1时，商为11，余数也为0，结果正确。

被除数为3，除数为14时，商为0，余数也为3，结果正确。