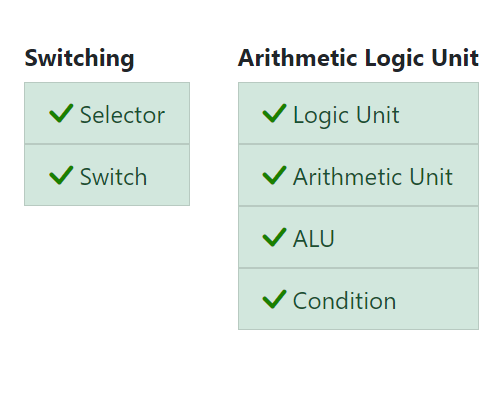
**继续完成nandgame里面的Switching和ArithmeticLogicUnit，记录耗时并总结收获。**

完成Switching和ArithmeticLogicUnit关卡耗时约2小时，其中Switching关卡重点在信号选择和切换逻辑，耗时较少，而ArithmeticLogicUnit关卡涉及算术与逻辑操作的结合，复杂度更高。这次练习让我深入理解了选择器和开关的逻辑控制，熟悉了ALU的核心功能及条件控制的作用。通过NandGame的实践，我强化了动手能力和对抽象逻辑概念的理解，为后续使用Verilog实现类似功能奠定了基础。

****

**自行设计一个简单运算模块，要求如下：**

1、两个输入（ina，inb），分别为8位的二进制数

2、多个输出，分别是这两个二进制数的运算结果，包括

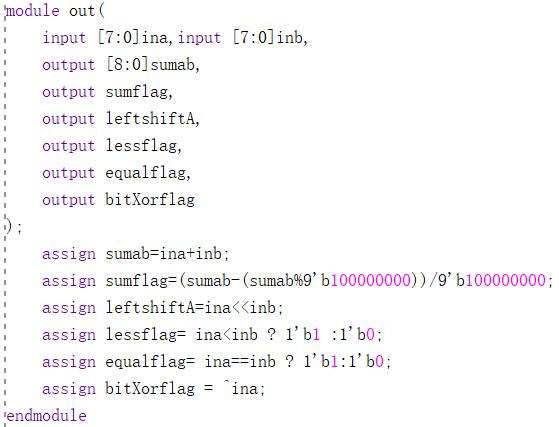
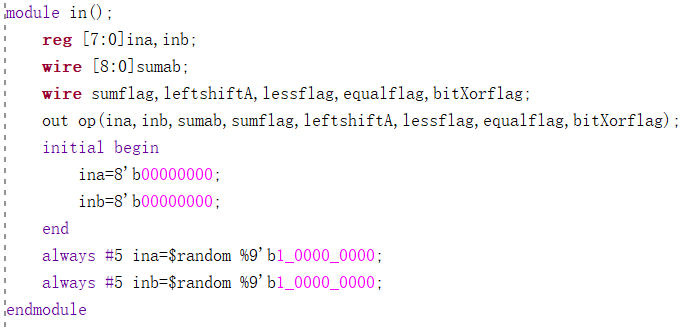
sumab:两个输入之和，sumflag:两个输入相加之后的进位

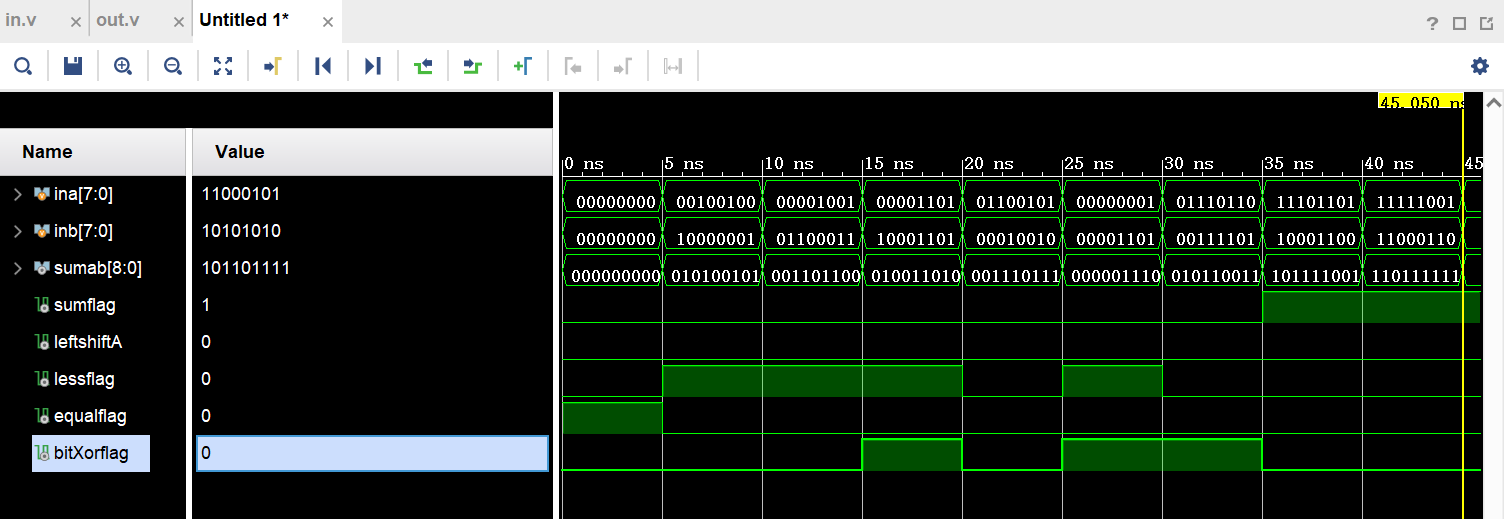
leftshiftA：把ina向左逻辑移位，移动的位数为inb，得到的结果（inb数值小于9）

lessflag:ina小于inb时返回1，否则返回0

equalflag:ina等于inb时返回1，否则返回0

bitXorflag:把ina按位缩减异或之后的结果



结果分析：

### 1.ina和inb的二进制值

ina[7:0]=11000101(十进制：197)

inb[7:0]=10101010(十进制：170)

### 2.sumab[8:0]

11000101(197)+10101010(170)

101101111(结果：367)

加法正确，结果为9位，最高位是进位。

### 3.sumflag

加法结果101101111是9位，最高位是进位1，因此sumflag=1。

### 4.leftshiftA

leftshiftA是ina向左逻辑移位inb位。

ina被完全移出，leftshiftA=0。

移位操作仅支持合理范围（如0到7），因为ina是8位，左移超过其位宽会导致溢出。

### 5.lessflag

197>170，所以lessflag=0。

### 6.equalflag

不相等，所以equalflag=0。

### 7.bitXorflag

把ina从后往前按位异或，bitXorflag=0。

**总结：**

通过本次程序设计和NandGame实践，我深入理解了数字电路设计和逻辑实现的核心概念。同时，通过实践，我提升了动手能力，体会到硬件设计的严谨性。本次学习为后续设计更复杂的实验奠定了基础，也激发了我进一步探索数字电路与硬件开发的兴趣。