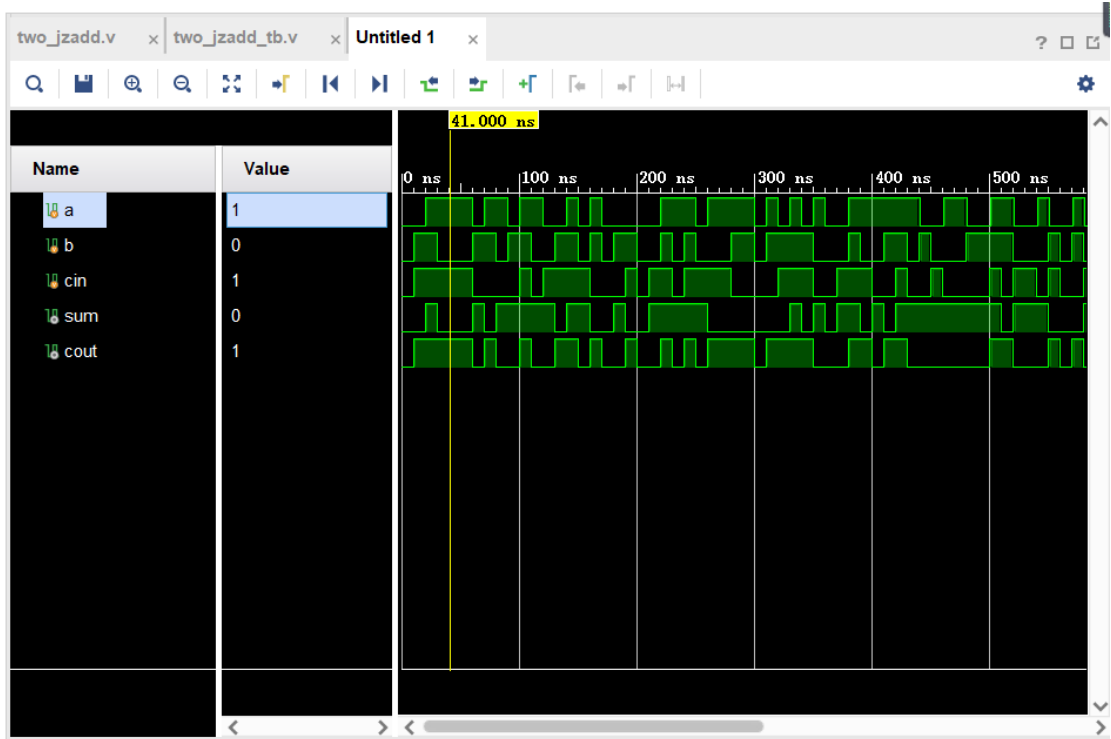


# 一、构建基本组合电路器件

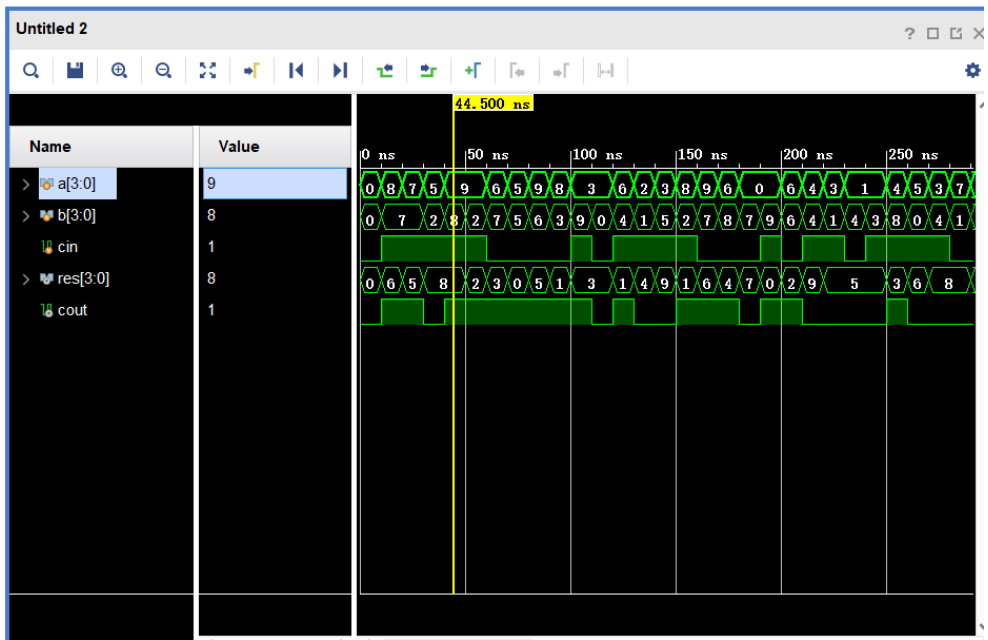
## 1.1 加法器

### (1) 二进制加法器



### (2) 十进制加法器

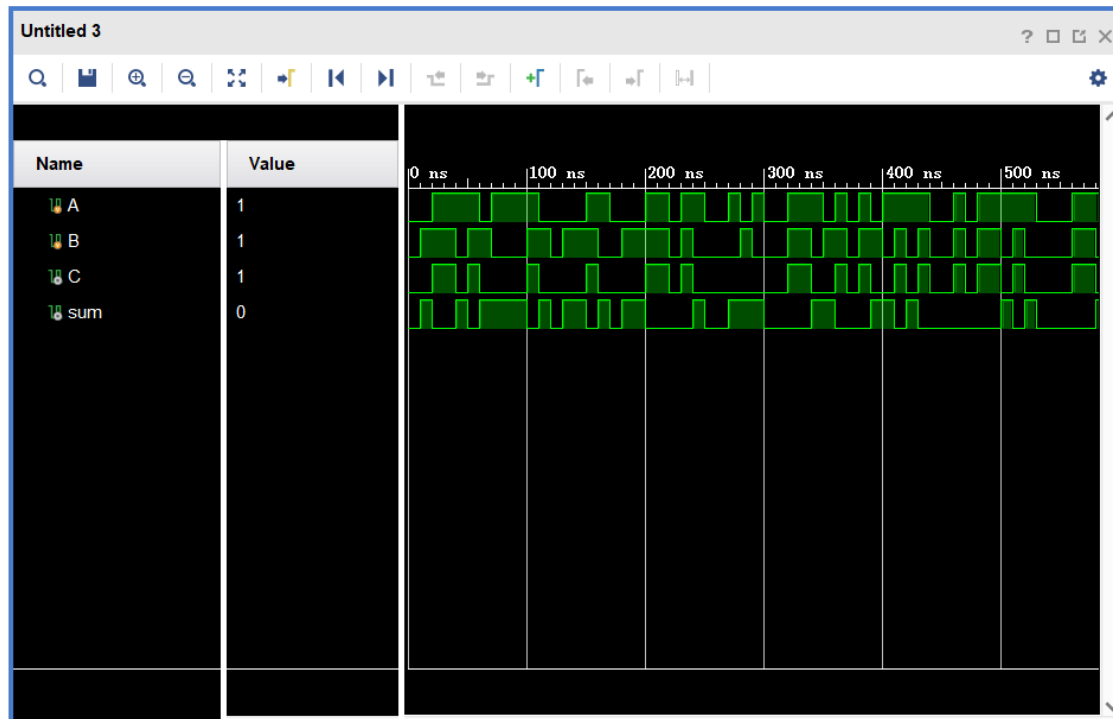
我们使用 bcd 码来表示十进制数，当出现大于 9 的数值我们需要+6 修正，使用四位加法器原理首先得到十六进制的结果再判断，修正；输入输出为 4 位进位为 1 位



## 1.2 全加器

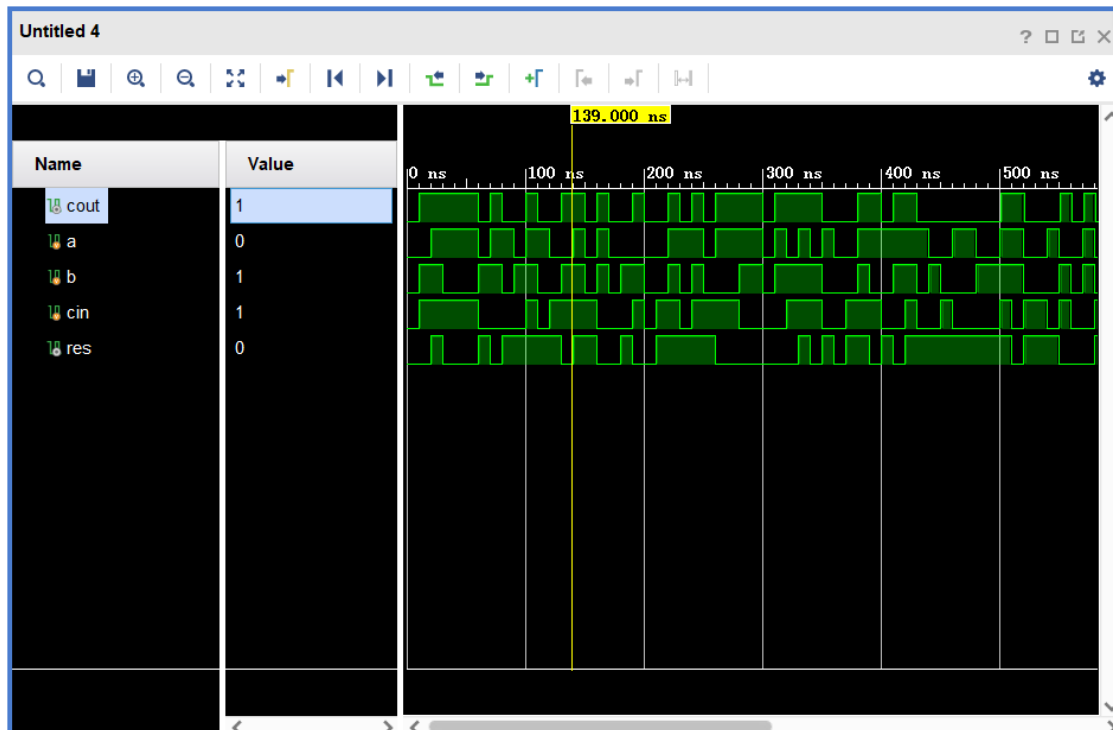
### (1) 半加器

根据真值表画出逻辑图即可



### (2) 一位全加器

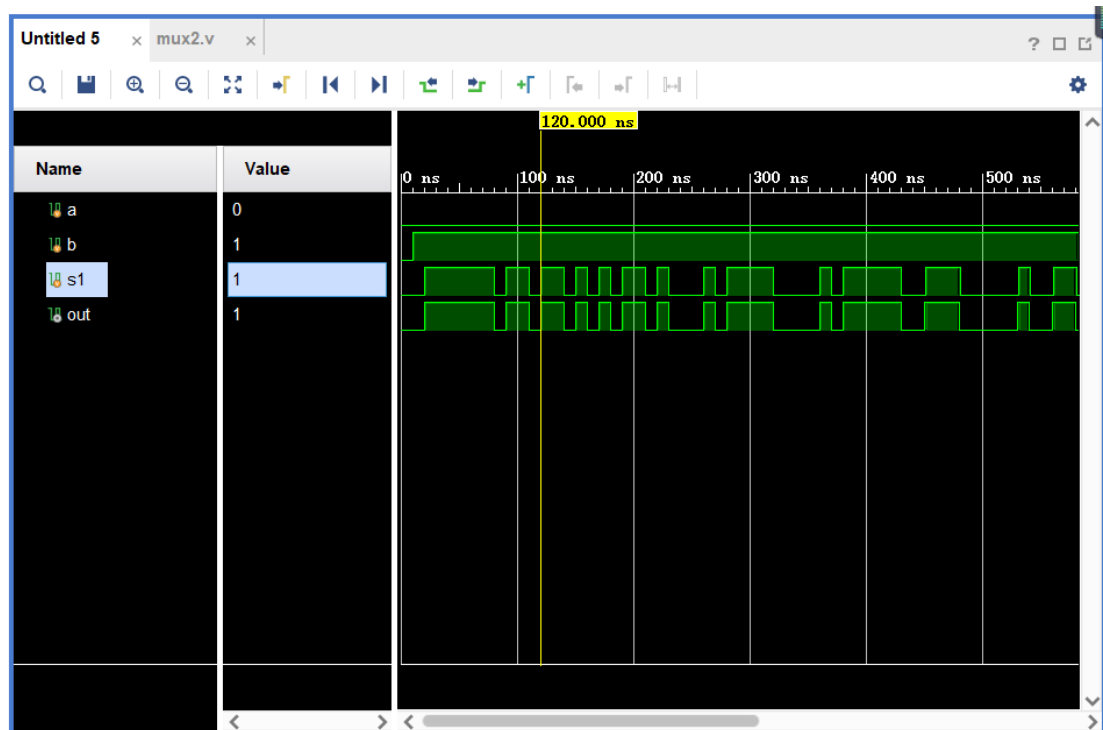
使用了封装的半加器 ip 核



## 1.3 选择器

### (1) 二选一数选器

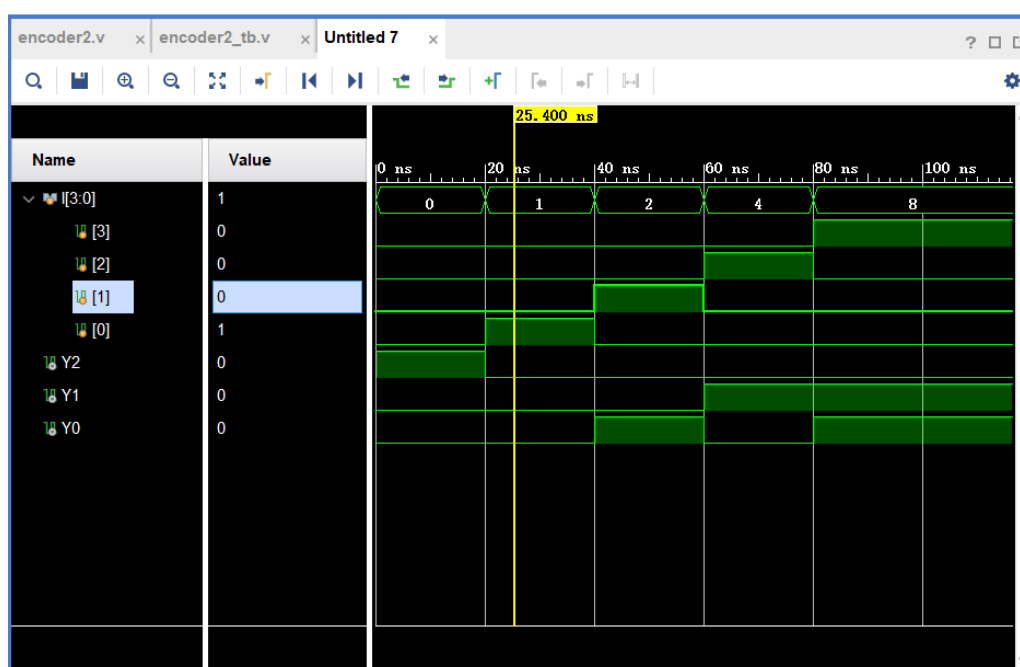
根据输入的 s1 的值来判断选 a 还是 b, 如果 s1 为 0 那么就选 a, 如果 s1 为 1 就选 b; 其中 a, b 的值我给的是固定的分别为 0, 1



## 1.4 编码器

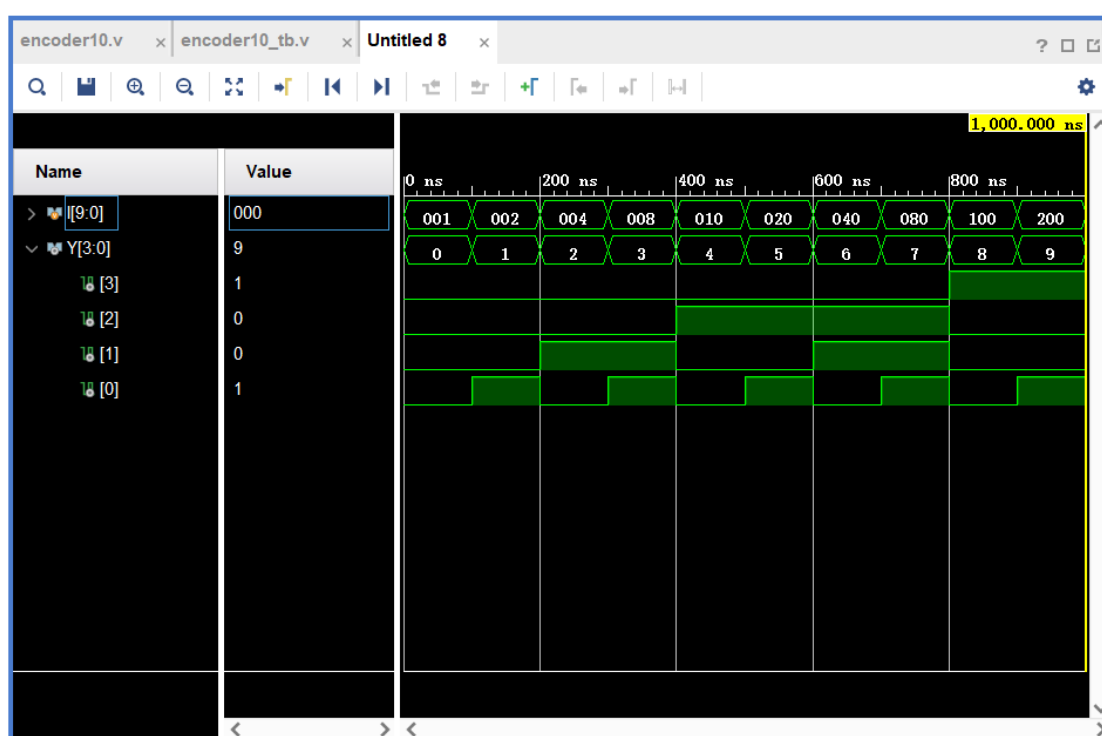
### (1) 二进制编码器

写出真值表则可得逻辑关系, 其中 Y2 为是否有信号的标志, Y1Y0 则组成对应的序号, 如图中若 I[0] 为 1 则 Y1Y0 为 00, Y2 为 0 表示有信号。设置变量值时需注意一次最多有一个 1



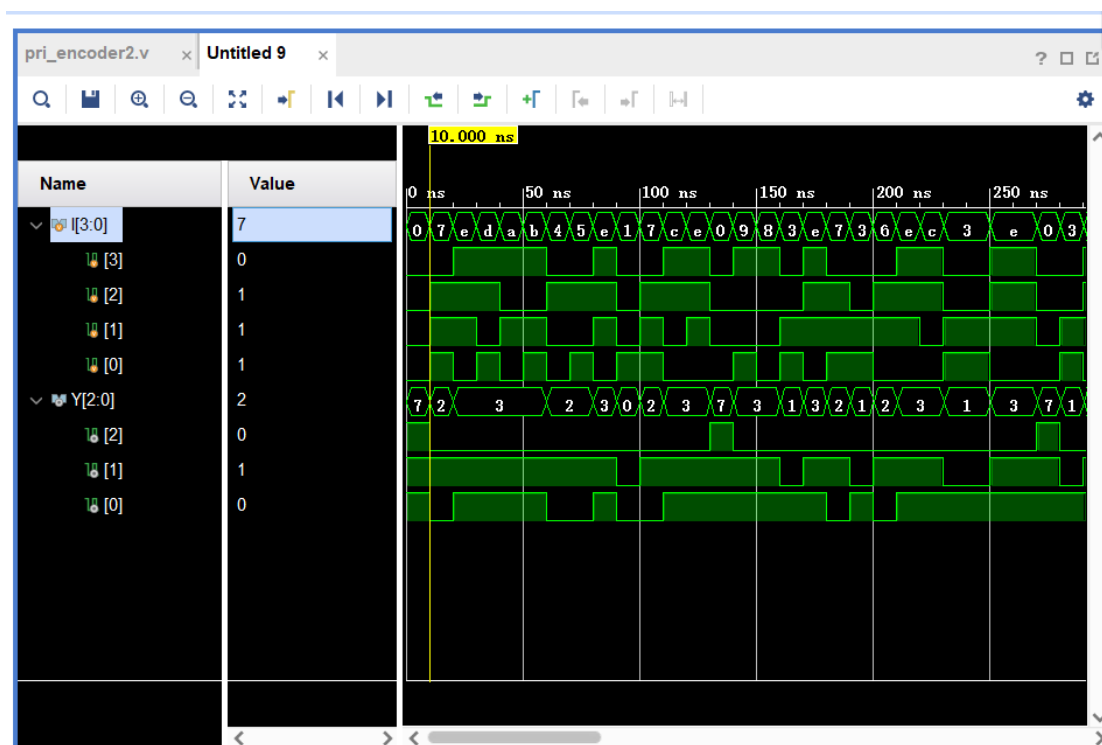
## (2) 二-十进制编码器

与 (1) 类似，也是写出真值表得到逻辑关系



## (3) 优先编码器

设计的是一个四位二进制优先编码器。这里输入的值可以为随机数，处理时我们以高位优先，从高向低判断是否为 1，为 1 就显示该输入的下标，比如图中。I[2]为 1，则无论下面是不是 1 结果都是 2；如果全部都是 0 则没有信号，输出直接为 7

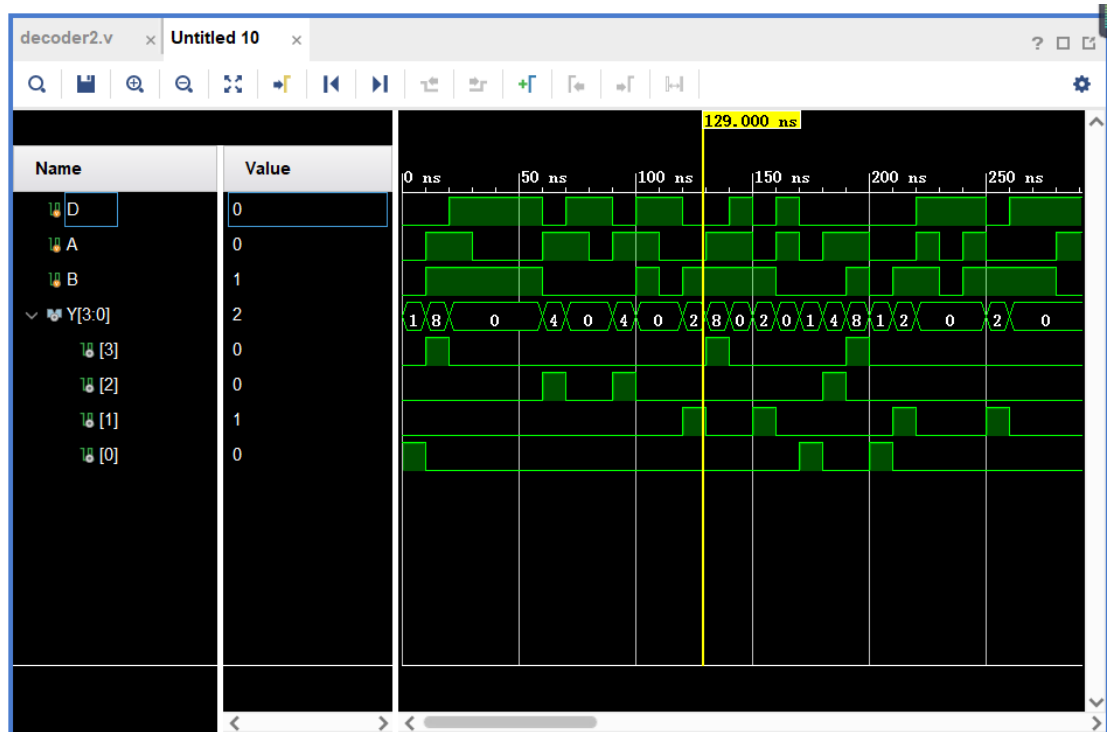


## 1.5 译码器

### (1) 二进制译码器

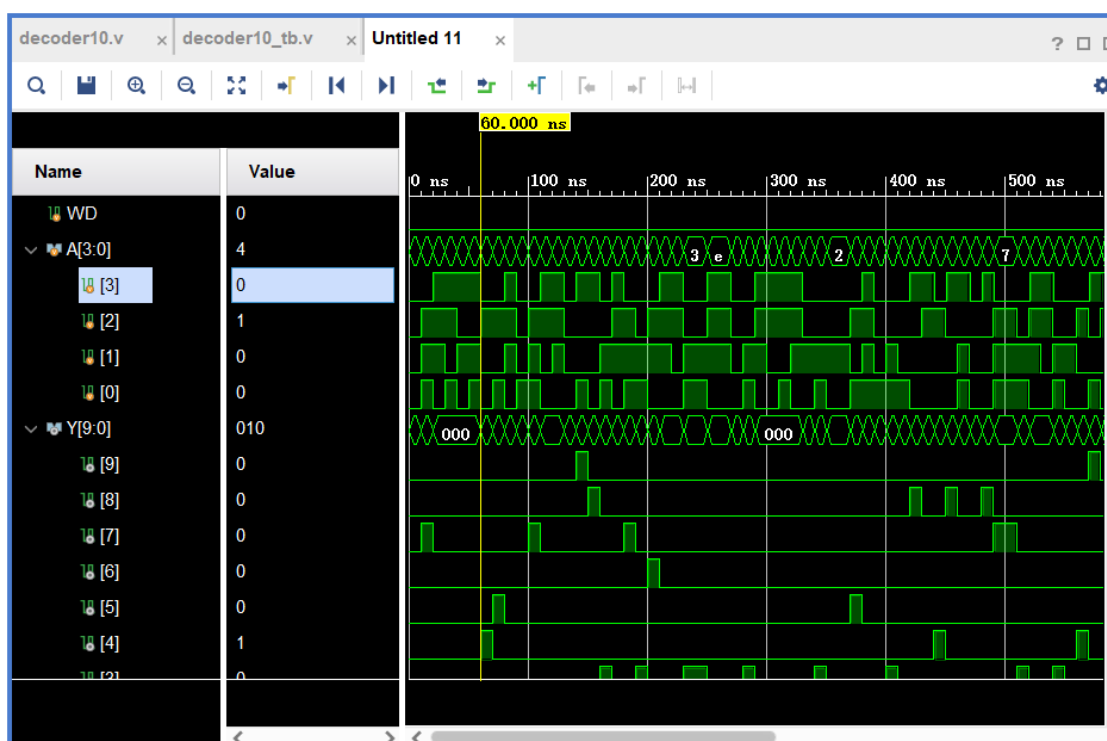
列出真值表则可得逻辑关系图

其中D是低电平有效的



### (2) 二-十进制译码器

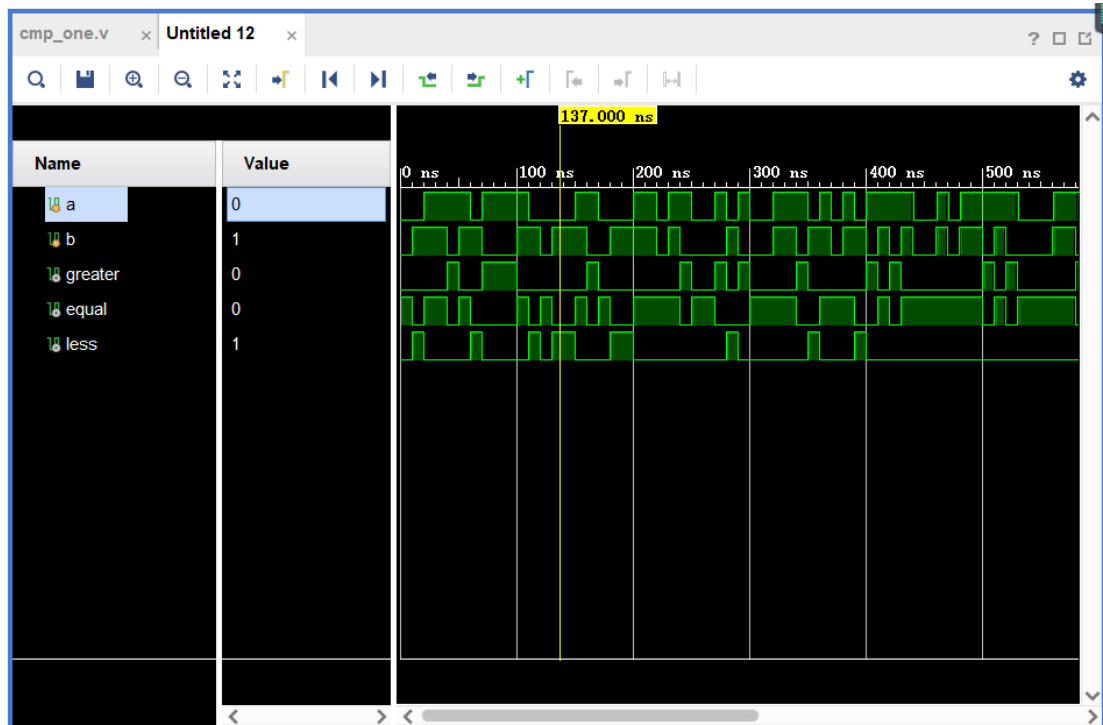
与 (1) 同理



## 1.6 数值比较器

### (1) 一位数值比较器

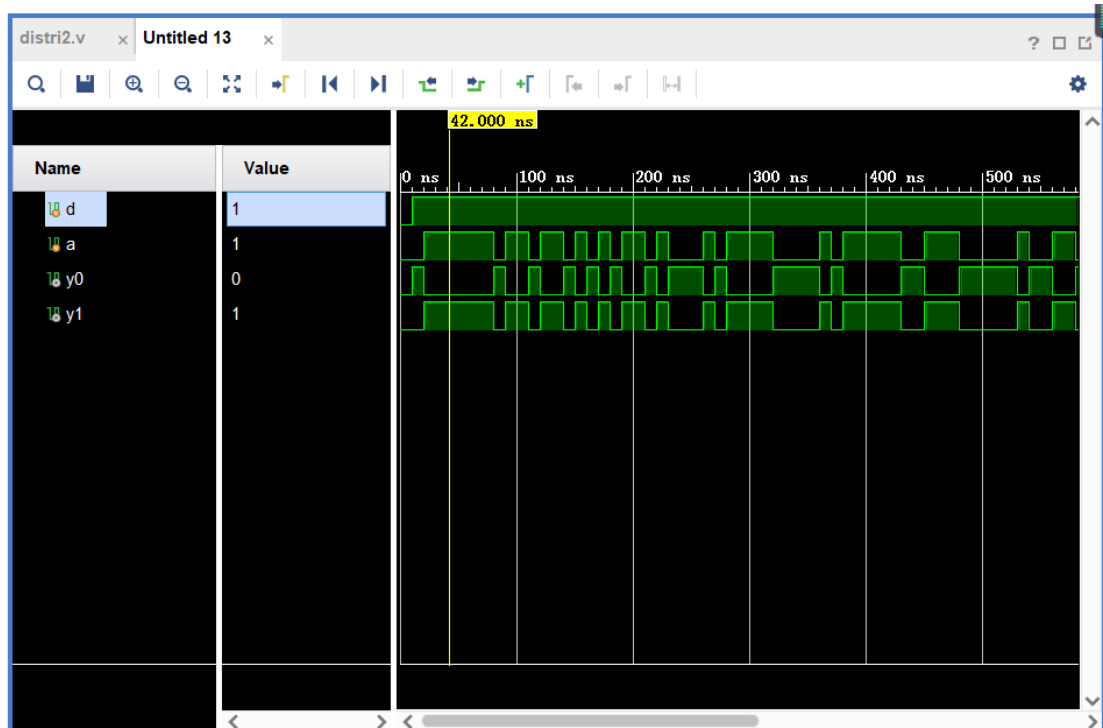
列出真值表即可得逻辑关系式进行编写, greater, equal, less 分别代表大于等于小于



## 1.7 数值分配器

### (1) 二路分配器

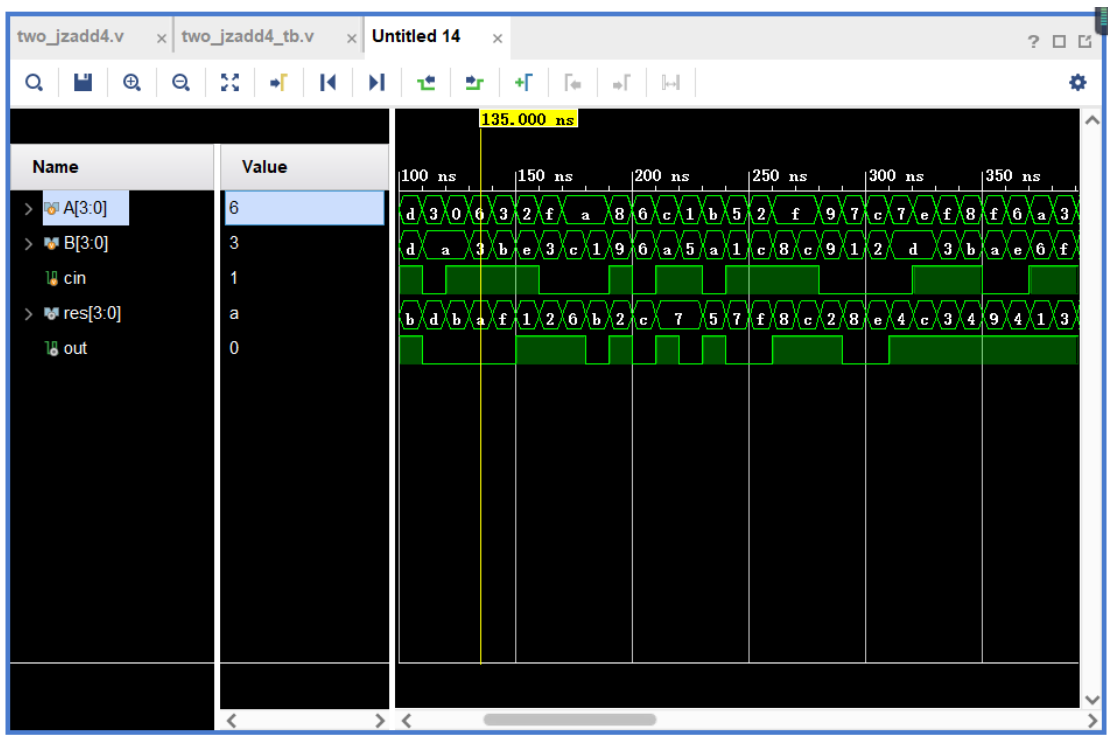
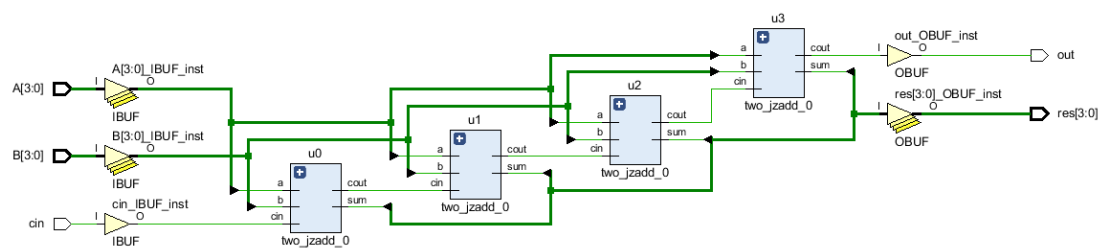
根据 a 的值决定将 d 的值分配给 y1 还是 y0, a 为 0 则将 d 分配给 y0, 为 1 分配给 y1



# 二、扩充功能

## 2.1 加法器

(1) 四位二进制加法器  
调用一位二进制加法器的 ip 核，四位每位对应相加，低位向高位进位  
对应逻辑图



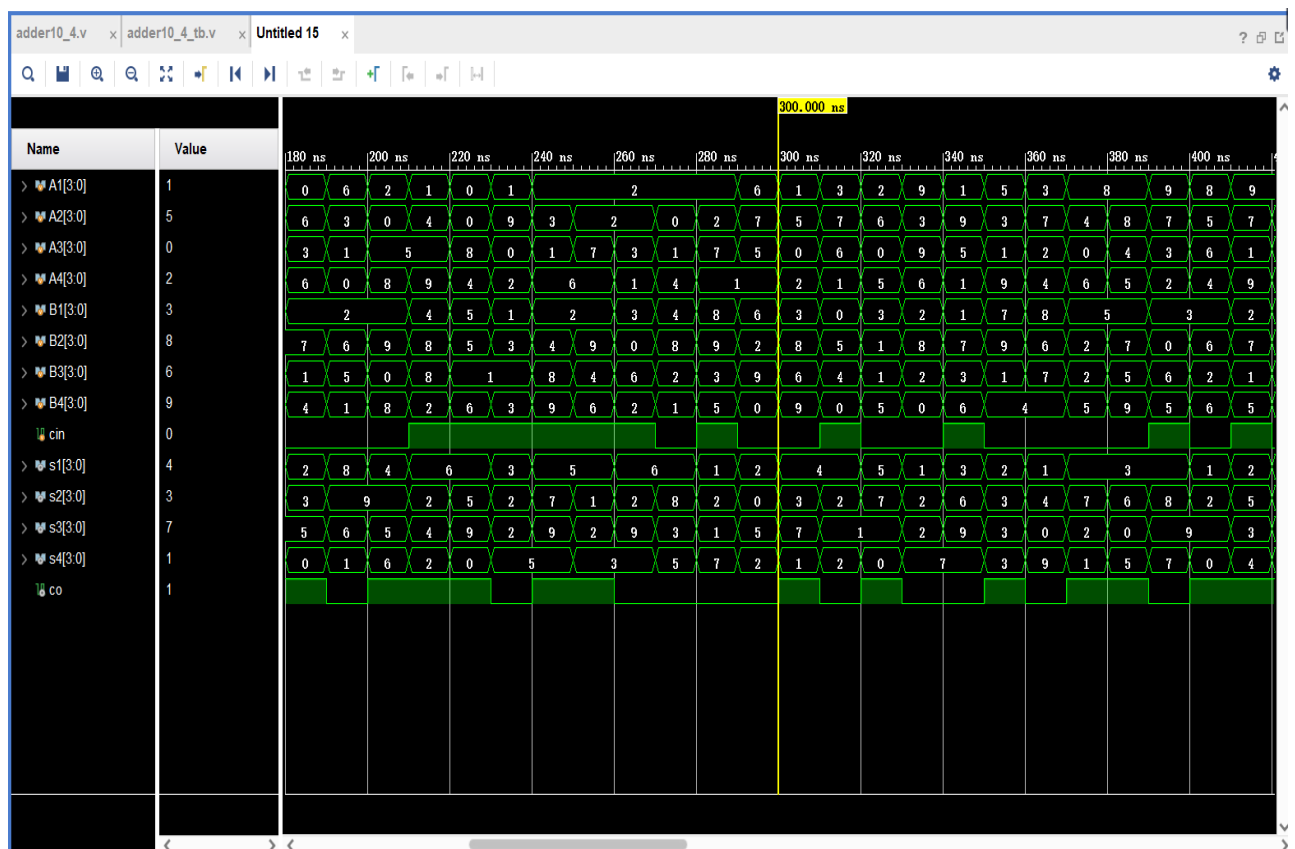
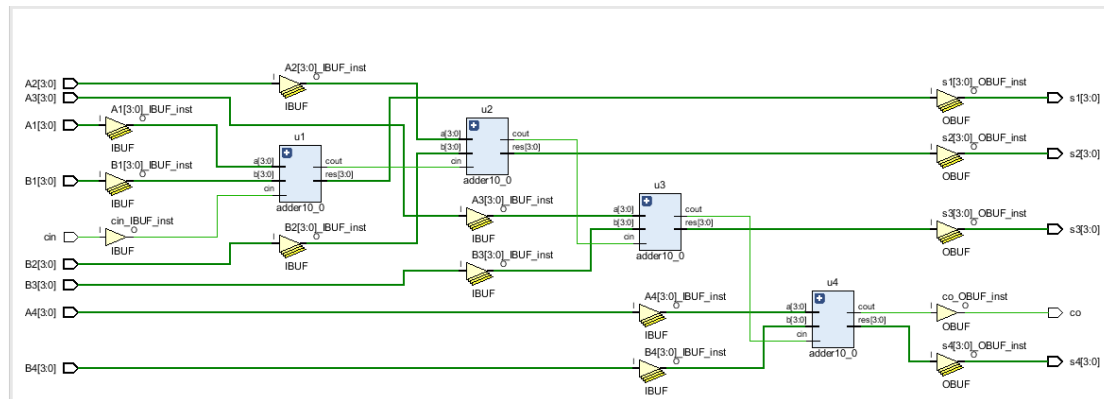
与 (1) 同理，调用一位十进制加法器 ip 核，这里输入和输出的每一位都是四位二进制，进位还是一位

逻辑图

与 (1) 同理, 调用一位十进制加法器 ip 核, 这里输入和输出的每一位都是四位二进制, 进位还是一位

逻辑图

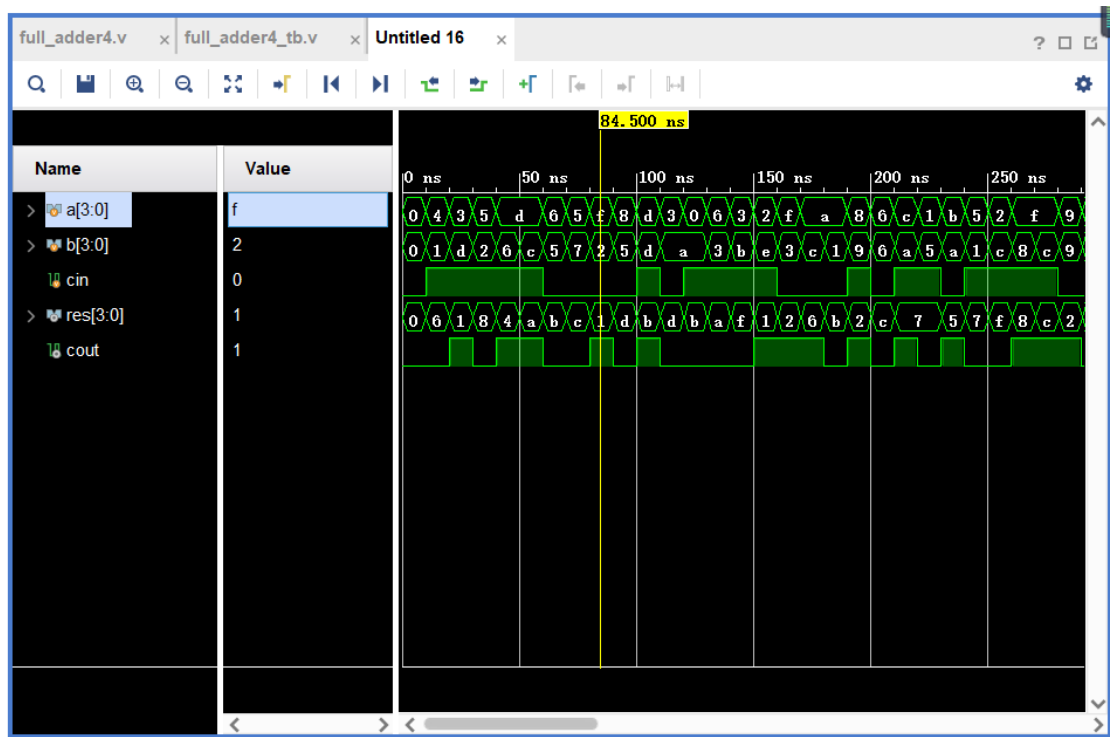
逻辑图





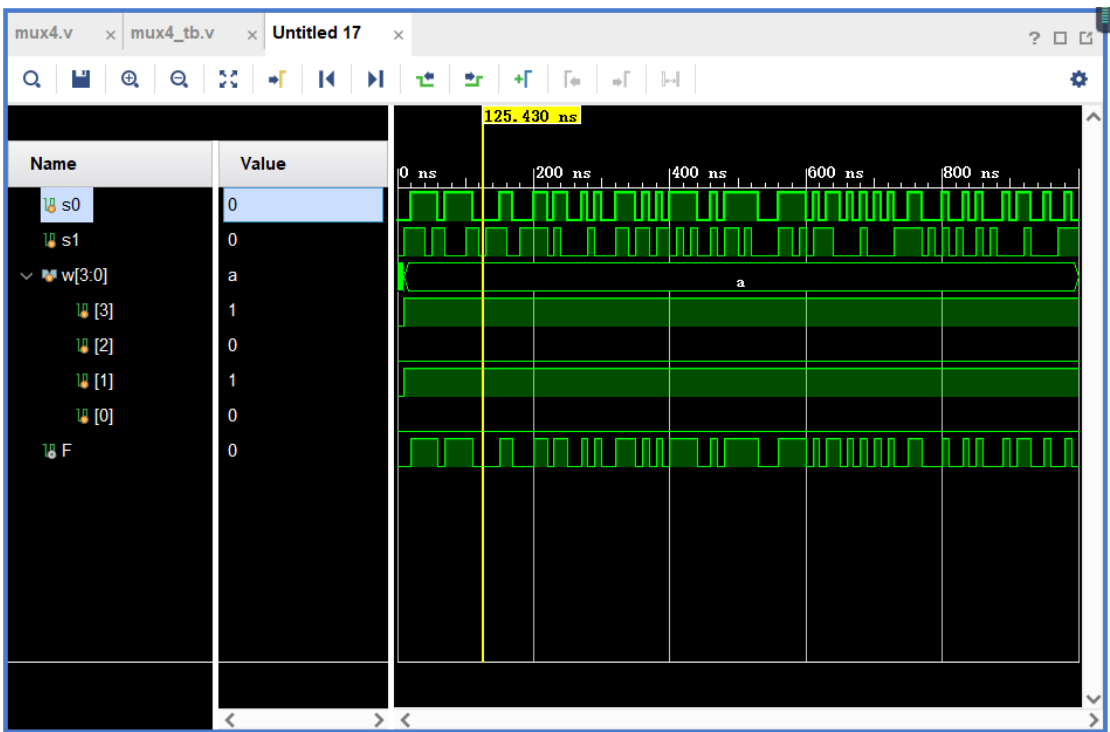
## 2.2 全加器

(1) 四位全加器  
这里用了模块例化的方式，调用一位全加器模块，逻辑图与四位二进制加法器相同

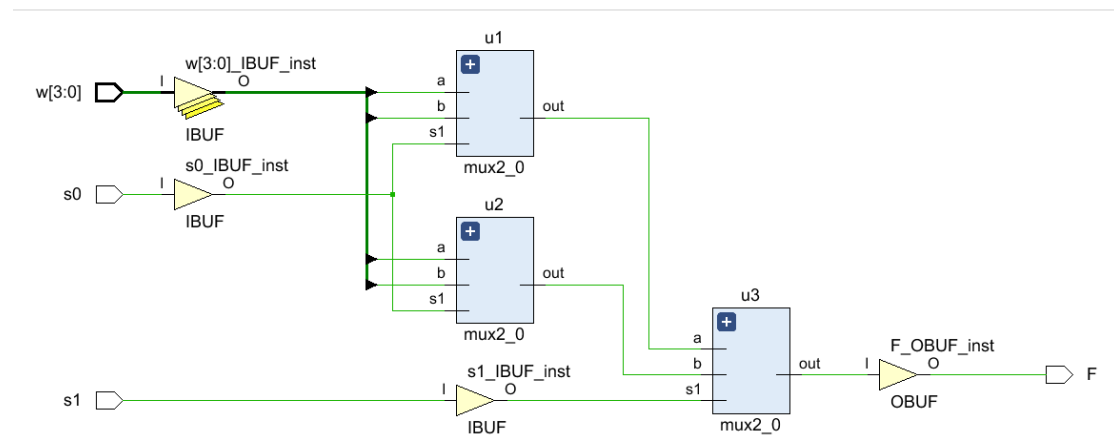


## 2.3 选择器

(1) 四选一数选器  
根据 `s1s0` 的值来判断输出选择输入的哪位数，如为 00 则选择 `w[0]`，11 选择 `w[3]`



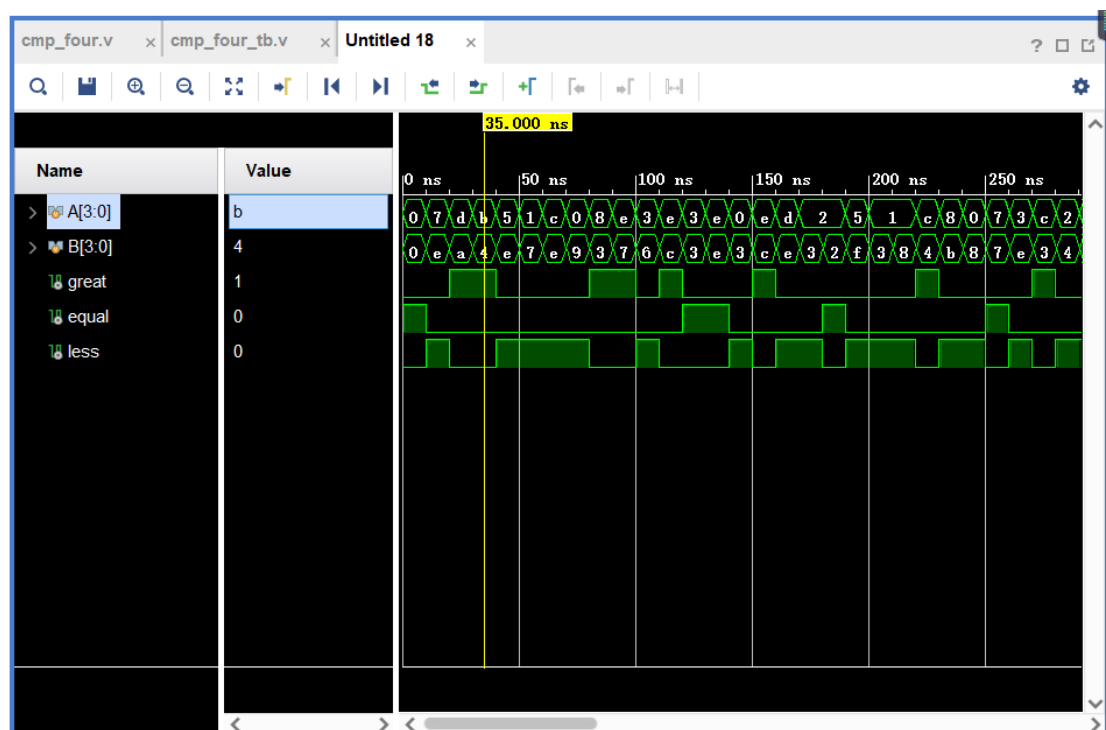
逻辑图：



## 2.4 数值比较器

### (1) 四位数值比较器

调用一位数值比较器，对每一位作比较，然后若 equal 全为 1 则说明相等，比较每位大于小于所得值，来判断是否为大于关系，小于关系则可看是否为大于关系或等于关系来判断  
具体见代码

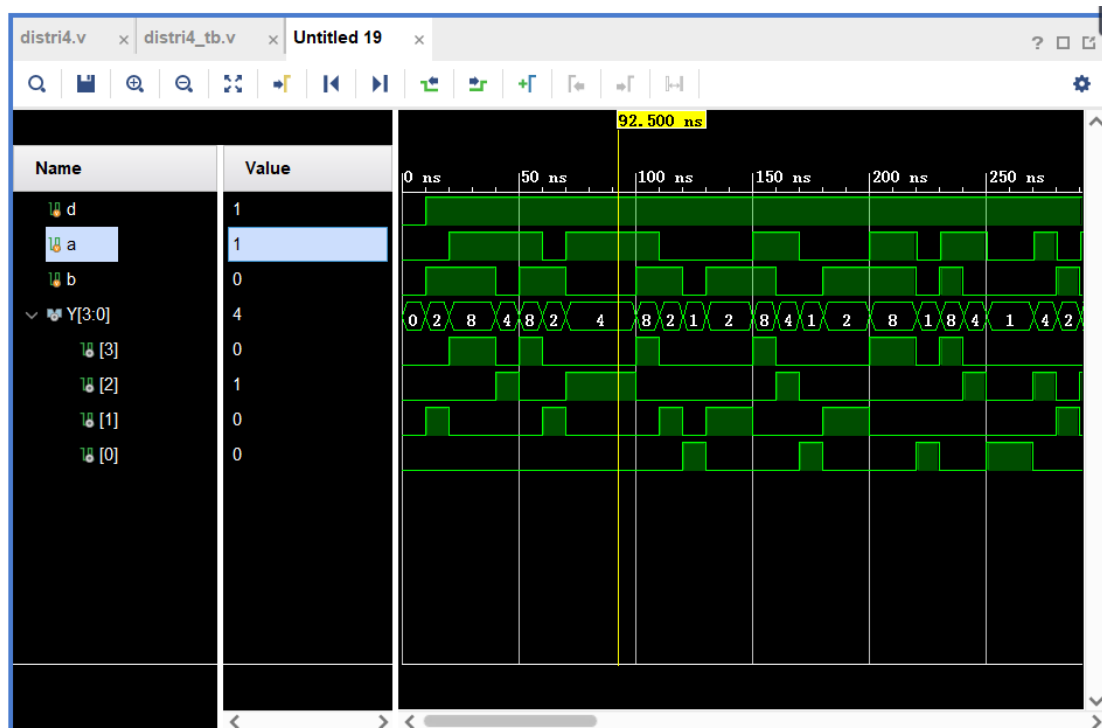
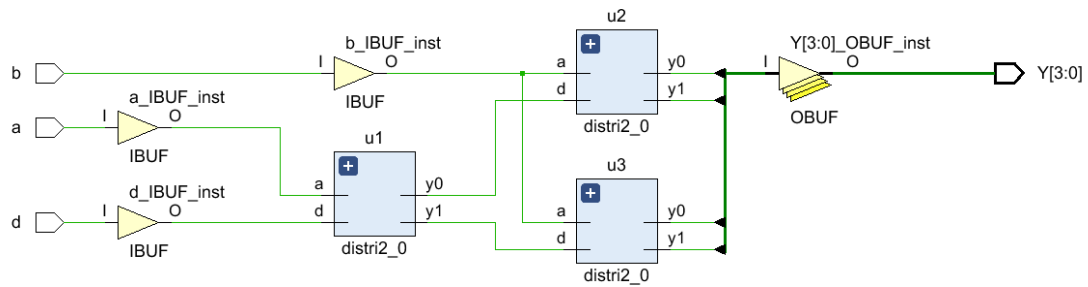


## 2.5 数据分配器

### (1) 四路分配器

根据 a,b 的值来判断选择将 d 值分配到 Y[3...0]之间一个数，如 10 则分配给 Y[2]，Y[2]为 1  
也是调用二路分配器 ip 核

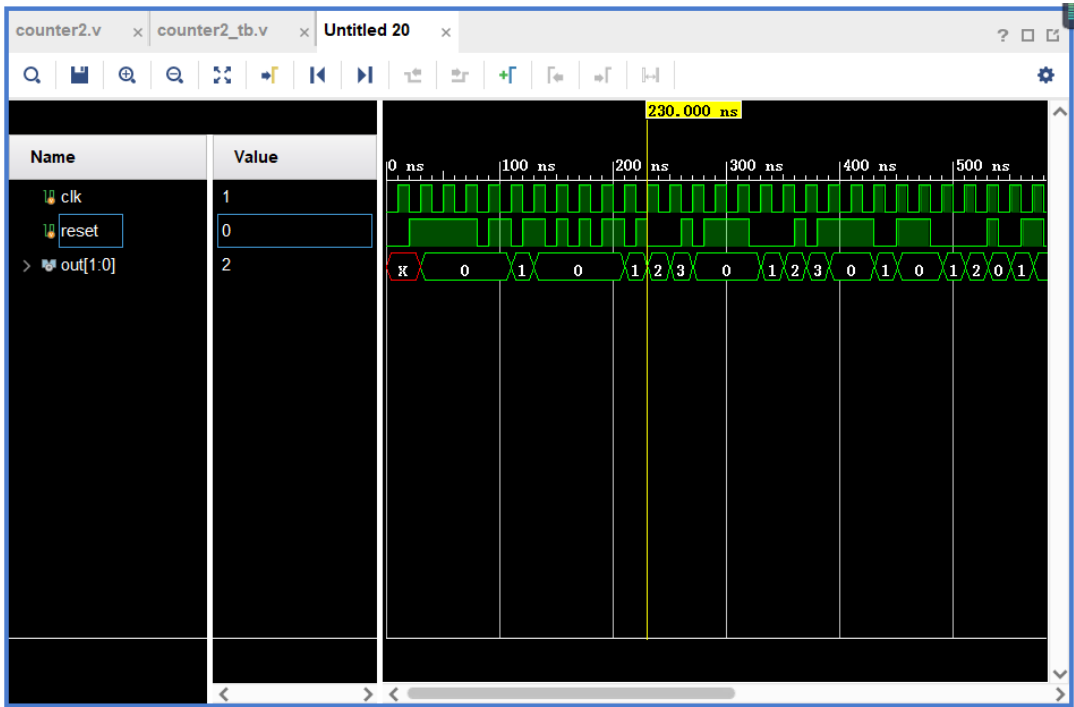
逻辑图：



### 三、常用功能器件

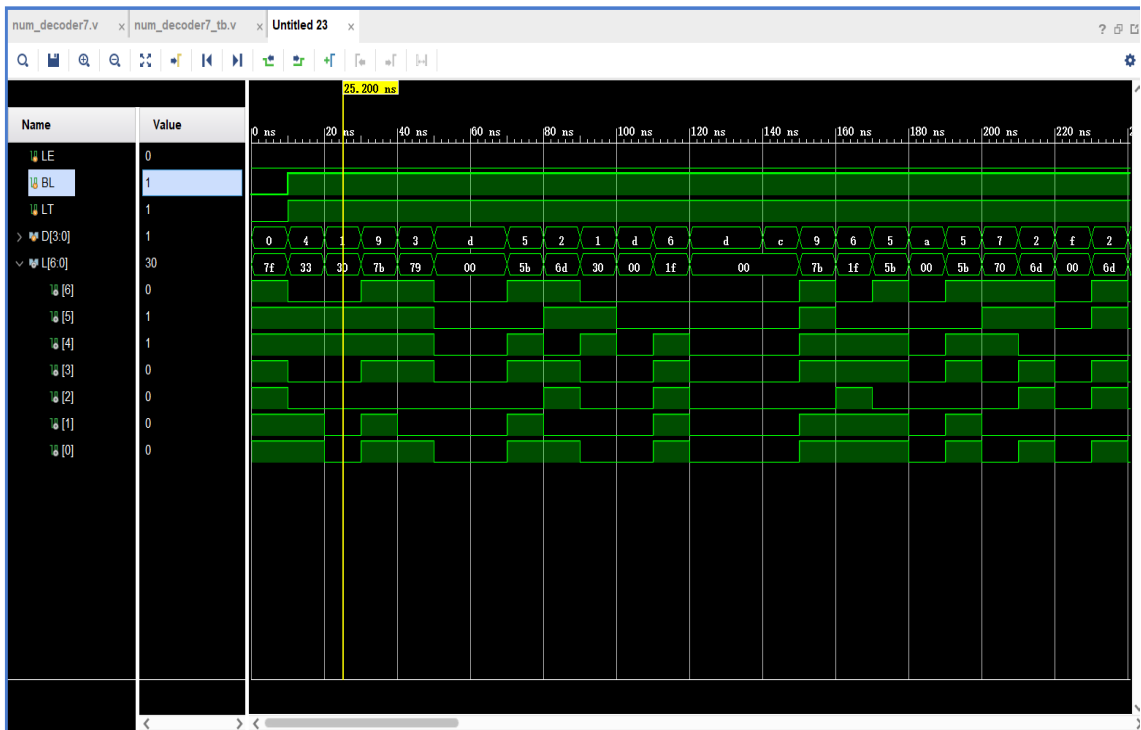
#### 3.1 二位计数器

其中 clk 为时钟周期，该程序是上升沿有效，reset 是置零操作，不置零则 out 就会+1



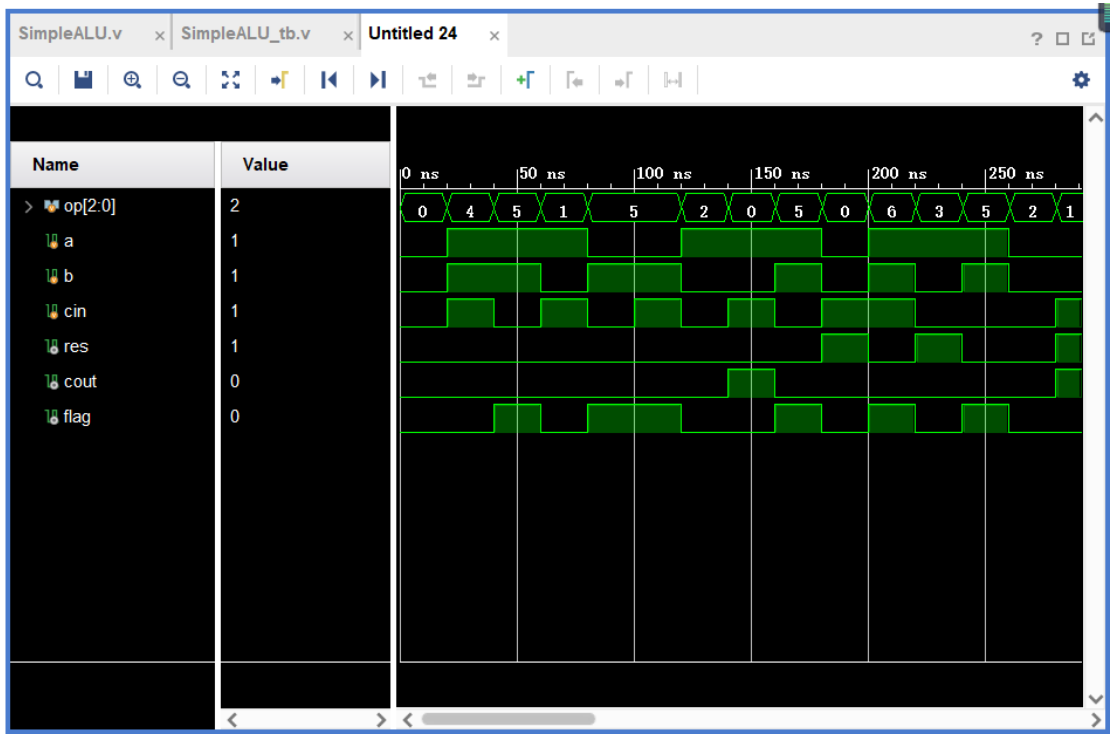
#### 3.2 七段数字显示译码器

查资料得各数字对应显示管真值表，L 对应的是显示管 a,b,c,d,e,f,g



### 3.3 简单算数逻辑单元

说明：其中 op 代表着操作类型，为 0 时是加操作，为 1 时是减操作，此时 cout 为是否借位；为 2 时是与操作，为 3 时是或操作，为 4 时是对 a 取反，这些情况 flag 为 0；其他值的情况对应 flag 为 1，表示没有操作信号



### 3.4 三状态二进制格雷码

根据格雷码的定义我们将二进制码从最右边一位起，依次将每一位与左边一位异或，最左边一位不变，可利用 verilog 中的右移和异或来实现

