



## **第10章 模拟仿真**

### **10.1 模拟的目的与方法**

### **10.2 逻辑模拟模型**

### **10.3 逻辑模拟算法**

#### **10.3.1 事件表驱动的模拟算法**

#### **10.3.2 事件冲突和惯性延迟的处理**

#### **10.3.3 三值模拟的作用及算法**

### **10.4 VHDL模拟**



# 一. 模拟的目的和方法

- ◆ 1. 设计与验证
- ◆ 验证 (Verification):
  - 电路描述或设计结果的正确性检查。
- ◆ 三类方法:
  - 模拟 (simulation)
  - 形式验证 (Formal verification)
  - 设计规则检查

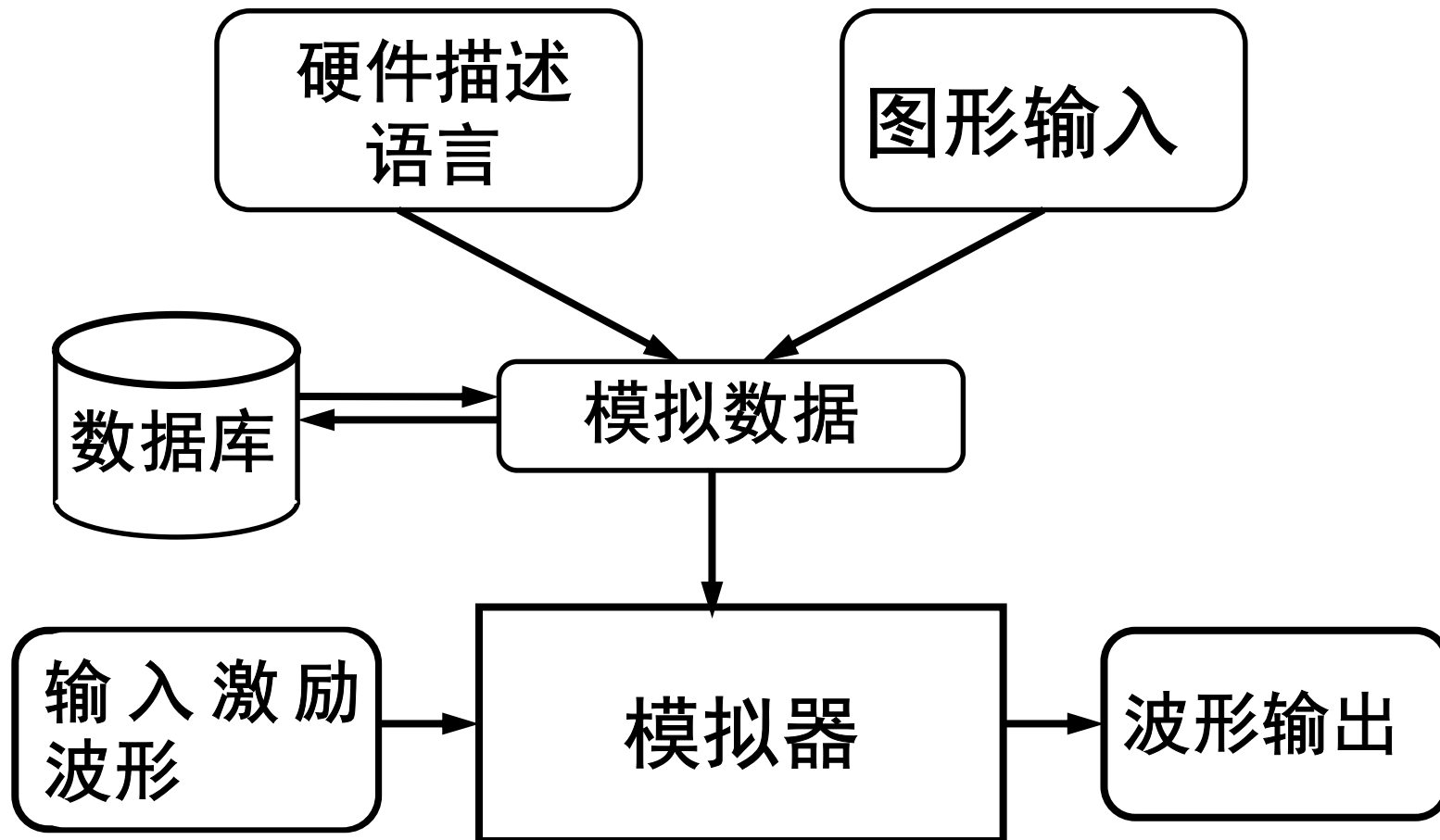


## 2. 模拟级别

级别	描述	特点	算法	工具例
电路级模拟	晶体管电路	电压电流	解方程	Spice
开关级模拟	开关晶体管电 路网表	逻辑值	信号传播	MOSSIM, RSIM
逻辑模拟	逻辑电路网表	逻辑值	信号传播	SIM
寄存器传输 级模拟	VHDL, CDL, DDL	数据流、 控制流	信号传播	DDL
高层次模拟	VHDL, Verilog	算法、 行为	程序执行	VSIM



### 3. 模拟系统的基本组成





## 二. 逻辑模拟模型

- ◆ 电路网表 (Netlist)
- ◆ 信号状态值: 2值, 3值, 多值, 强度
- ◆ 延迟模型: 零延迟, 单位延迟, 标准延迟, 上升下降延迟, 模糊延迟;
  - 惯性延迟与传输延迟
- ◆ 元件模型: 基本门, 三态门, 传输门, 功能块与子线路



## 信号状态值

- ◆ 2值模型: 0, 1
- ◆ 3值模型: 0, 1, X
- ◆ 高阻值Z与4值模型0, 1, X, Z
- ◆ 信号值强度与带强度的信号值



## 信号值的强度

- ◆ 强度表示信号的驱动能力.
- ◆ 常用强度:

F	R	W	Z	U
强	电阻	弱级	高阻	未定
- ◆ 带强度的信号值:
  - 信号值一般取0, 1, X,
  - 每个值都区分为各种强度
- ◆ IEEE Std-1164 标准逻辑系统
  - U, X, 0, 1, Z, W, L, H, -

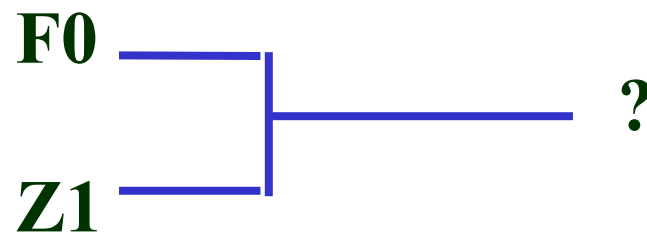


## 带强度信号值的运算：

### 强度取胜原则

$F0 \text{ wor } R1 = F0, \quad F1 \text{ wor } F0 = F1,$   
 $F1 \text{ wor } F1 = F1;$

$F0 \text{ and } R1 = ?$   
 $F0 \text{ and } Z1 = ?$







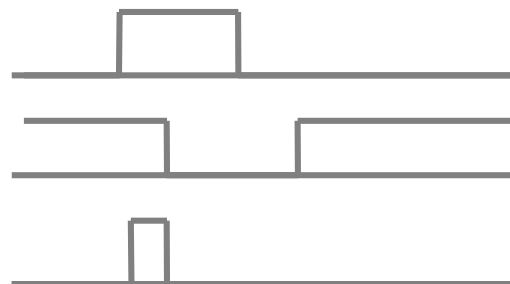
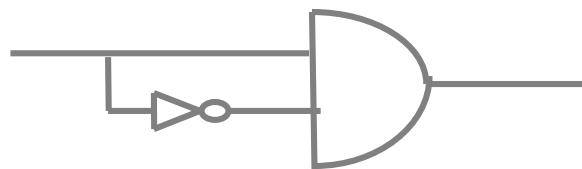
# 延迟模型

## (1) 元件延迟时间值

◆ 零延迟：无延迟，不能处理反馈信号

◆ 单位延迟：延迟时间均为1（无单位）

■ 可处理一些特殊情况：



■ 不适用延迟差别较大的情况（功能模块）



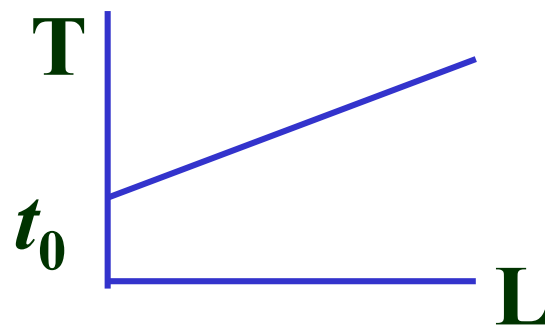
## 延迟模型

### (1) 元件延迟时间值 (续)

◆ 标准延迟：手册给定的值

■ 时间单位：实际时间单位；相对时间单位

■ 负载延迟  $t = t_0 + k \cdot l$

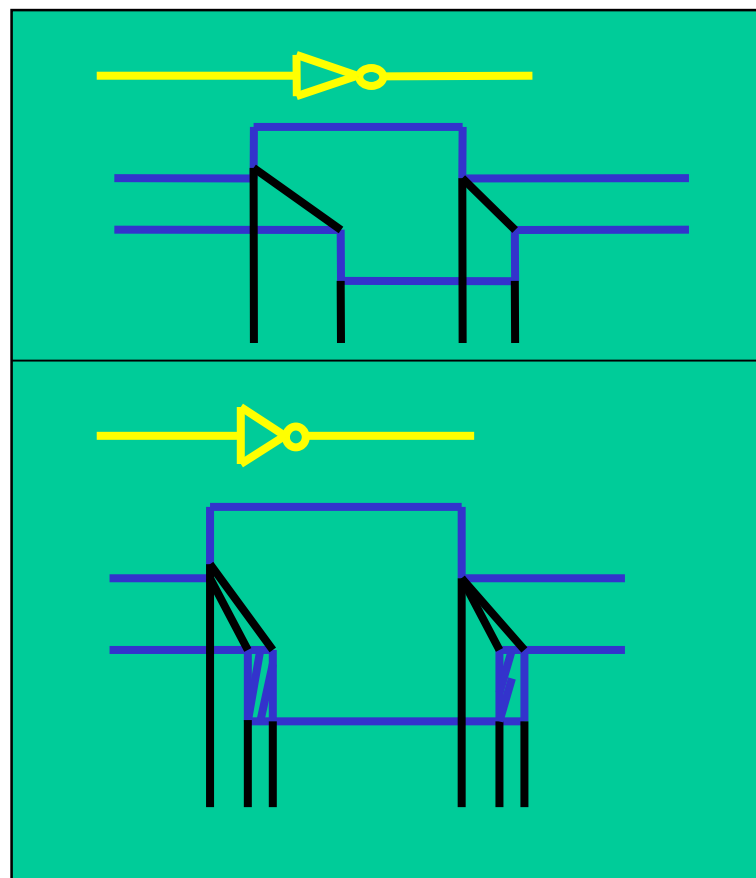




# 延迟模型

## (1) 元件延迟时间值 (续2)

- ◆ 上升/下降延迟
- ◆ 模糊延迟  
(最小最大延迟  
区间延迟  
最坏延迟)

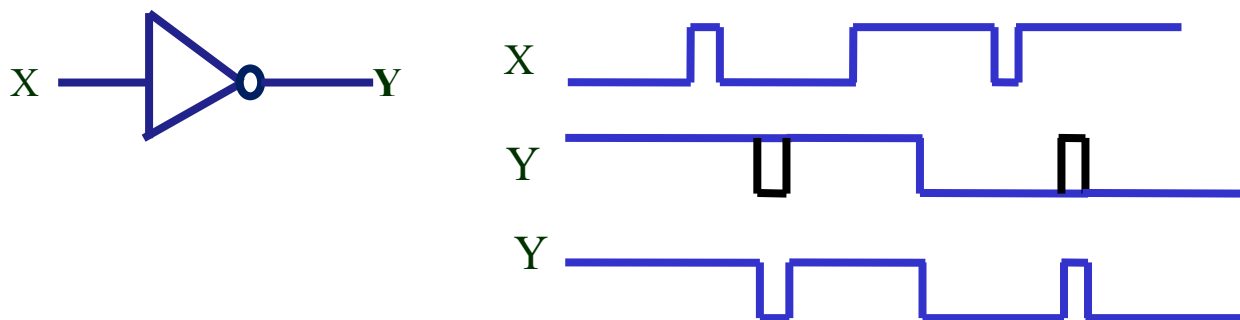




# 延迟模型

## (2) 两种延迟方式

- ◆ **传输延迟 (transport delay):**  
输出端响应输入端的微小变化;
- ◆ **惯性延迟 (inertial delay):**  
输入端信号保持一定时间, 输出端才有响应。





# 元件模型

## ◆ 基本门

AND, OR, NAND, NOR, XOR, EXOR

一个输出端，输入端对称，各端延迟相同

## ◆ 三态门

TRI-NOT, TRI-AND, ...

一个输出端，有一个使能端（允许端），其它输入端对称，输出有高阻状态。



## 元件模型(续)

### ◆ 传输门

接通后相当于短路，可有延迟



— 单向

— 双向 (正通, 负通, 对称):

由两边的强度决定其方向。



— 延迟线: 有延迟





## 元件模型：功能块与子电路

- 功能块：  
具有一定功能的部分电路，其模型为功能描述。
- 子电路：  
具有任意功能的部分电路，其模型为结构描述。
  - 特点：若干输入，若干输出，可以有内部信号。
  - 功能块可以有内部的时间限制检查。
- ◆ 预定义功能块与自定义功能块。



### 三. 逻辑模拟算法

- ◆ 模拟算法的主要组成：
  - 计算元件的顺序安排—**调度** (schedule)
    - 主要实现**并行操作**，和**不同延迟时间**的安排
  - 元件模型的**计算** (evaluation)
    - 求**将来值**和**延迟时间**
- ◆ 模拟算法—与延迟模型密切相关
  - 零延迟——编排级数法
  - 单位延迟——下一事件方式
  - 标准延迟——**时间映射的事项处理表**  
(事件驱动算法)





# 事件驱动算法

- ◆ 事项处理:  $e = (s, v, t)$ 
  - 信号  $s$  将要在将来的  $t$  时刻得到值  $v$
  - 事件: 信号的一次变化

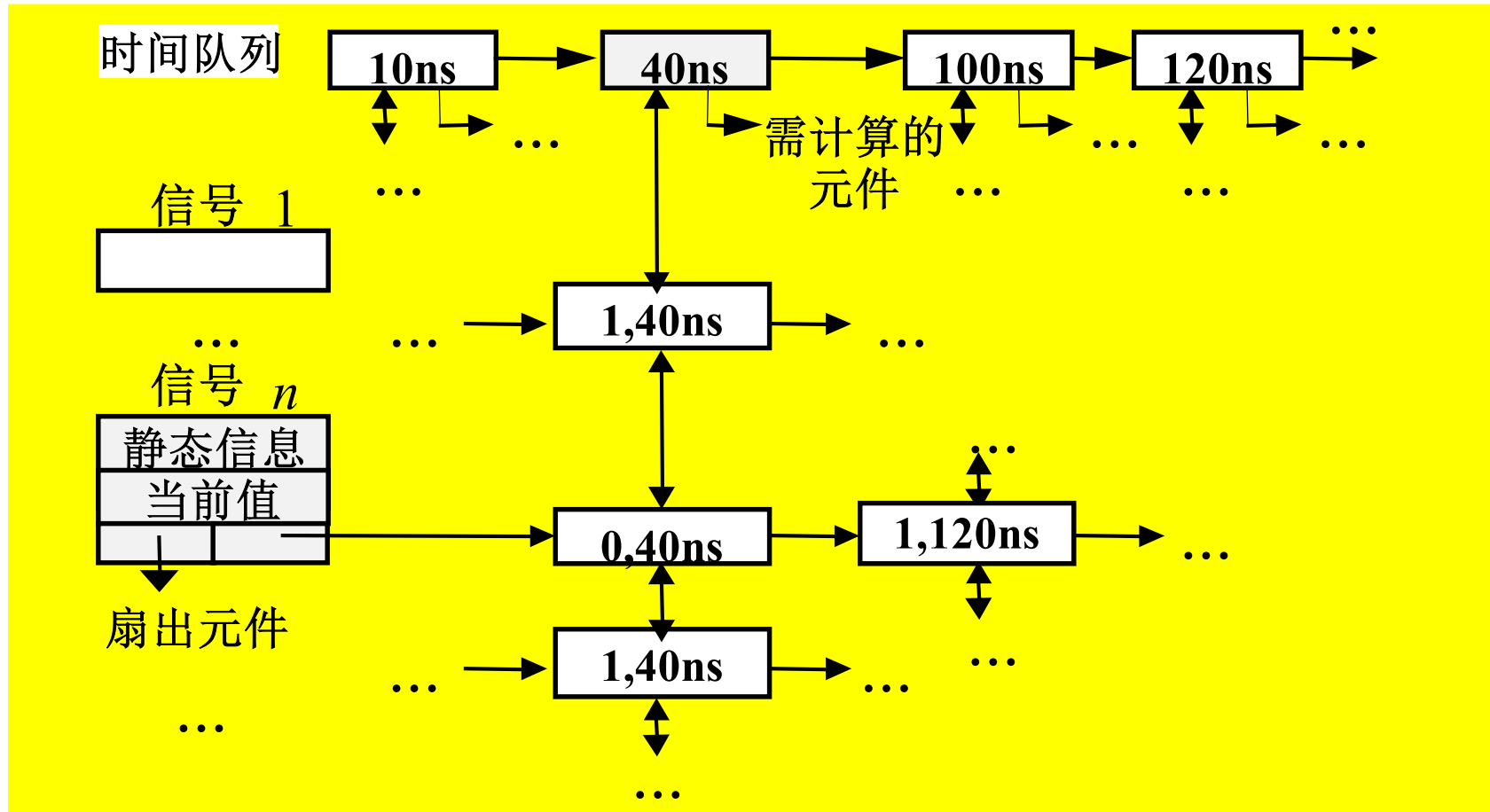
- ◆ 模拟周期



- ◆ 用事项处理链表保存和管理事项处理



## 动态数据结构示意图



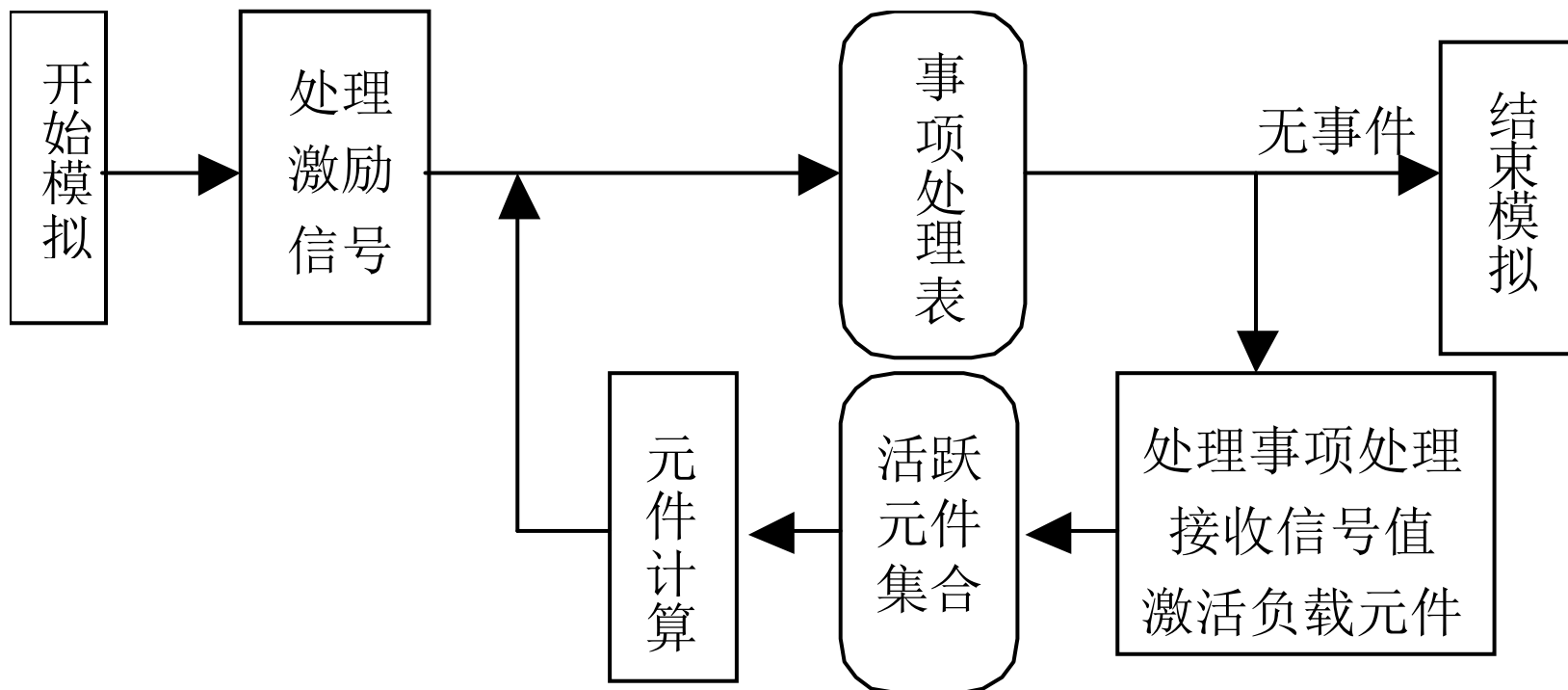


## 模拟过程

1. 外部信号事件和信号初始值加入事项处理表中。
2. 若已不存在事项处理或已经到达最大限定时刻，则退出，否则处理各当前事项处理：
  - 若信号值未变，删除该事项处理。
  - 若信号值变化，即为一事件，接受新值，并将其扇出元件放入活动元件表中。
3. 处理完各事项处理后，删除当前信号事项处理。
4. 计算活动元件；得到新的事项处理，插入事项处理表中。
5. 计算完后活动元件集合置空。
6. 模拟时钟走到下一时刻，修改事项处理表中的相对时刻，返回2。



## 模拟周期示意图





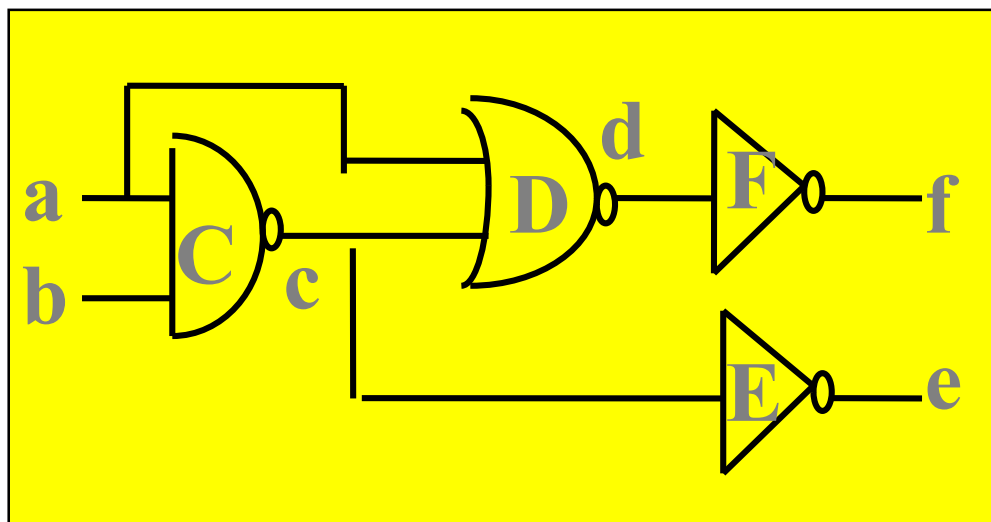
## 示例

设NAND,NOR:

延迟 = 20 ns ;

NOT: 延迟 = 10 ns ;

初值:  $a=1, b=0, c=1,$   
 $d=0, e=0, f=1$

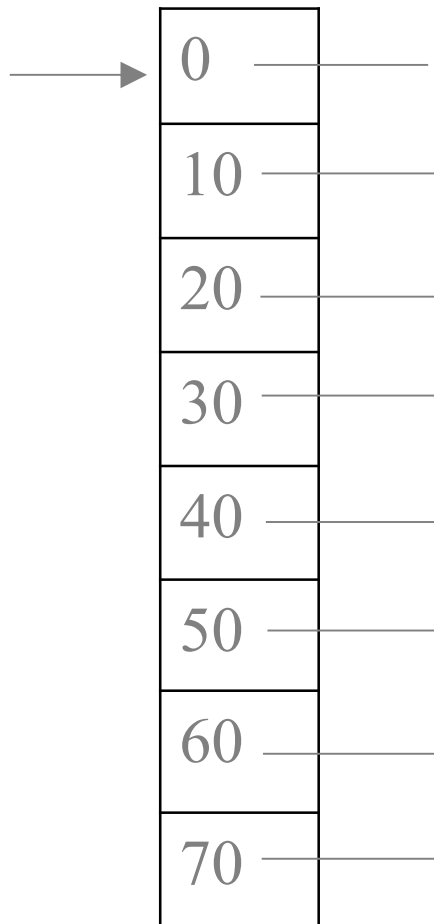


输入激励波形





## 模拟过程

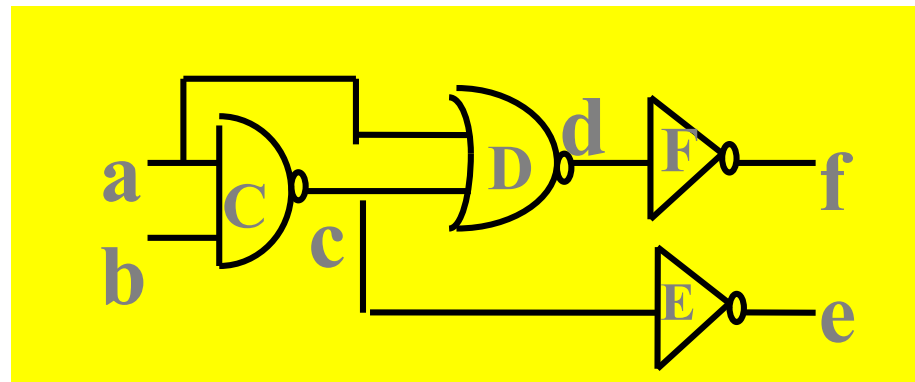


(a,1)(b,0)(c,1)(d,0)(e,0)(f,1)

(a,0)(b,1) (e,0)(f,1)

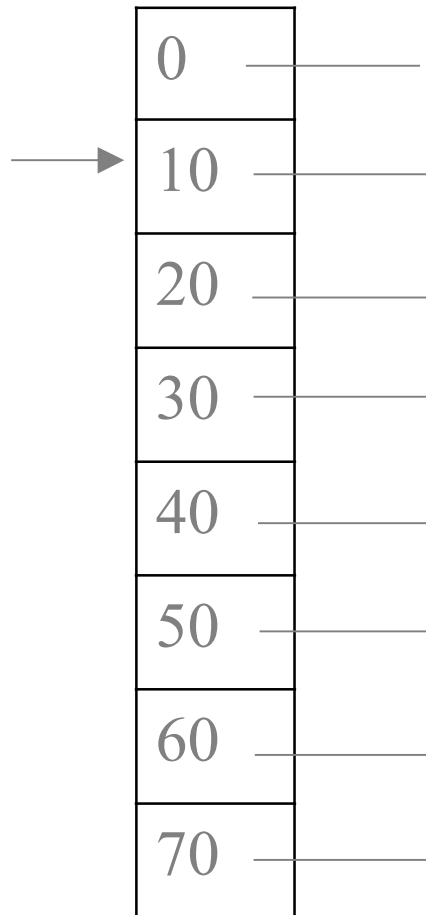
(a,1) (c,1) (d,0)

a	b	c	d	e	f
1	0	1	0	0	1





## 模拟过程

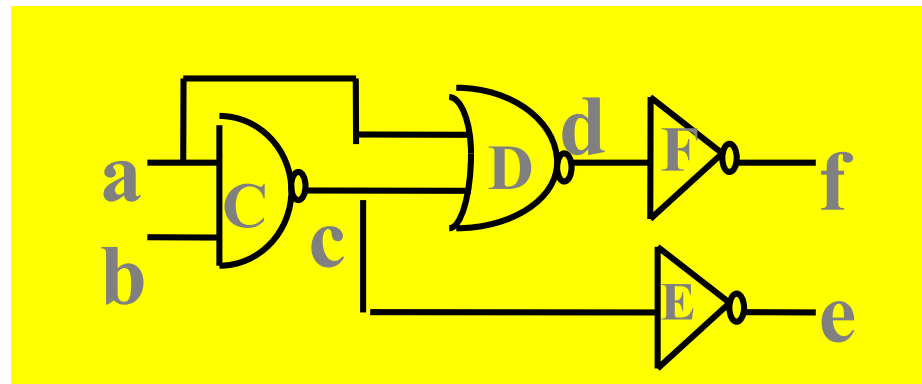


$(a,0)(b,1)(e,0),(f,1)$

$(a,1)(c,1)(d,0)$

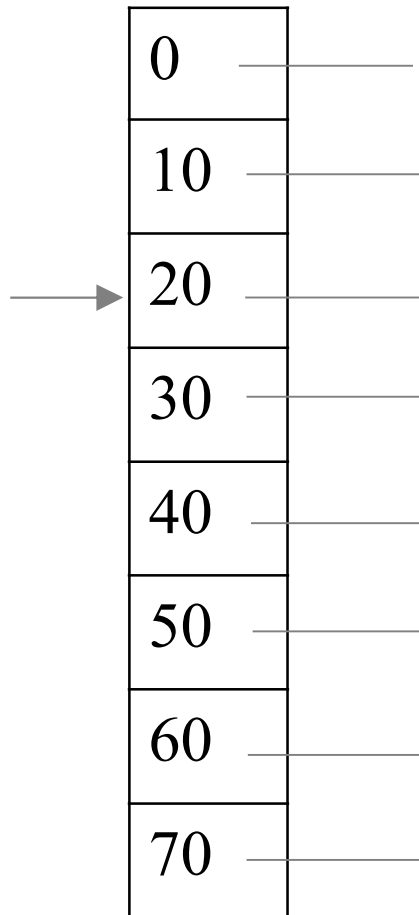
$(c,1)(d,0)$

a	b	c	d	e	f
0	1	1	0	0	1





## 模拟过程

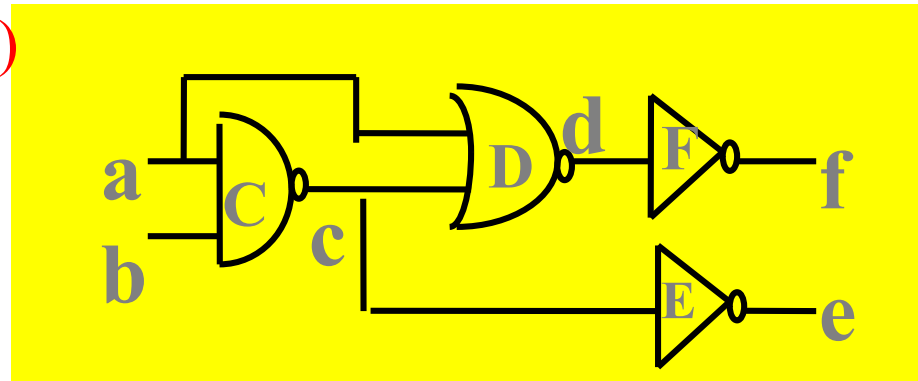


$(a, 1)(\cancel{c, 1})(\cancel{d, 0})$

$(c, 1)(d, 0)$

$(c, 0)(d, 0)$

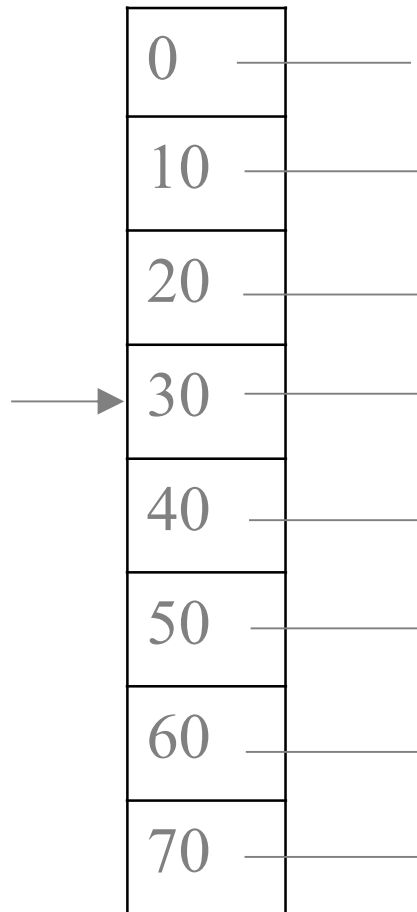
a	b	c	d	e	f
1	1	1	0	0	1







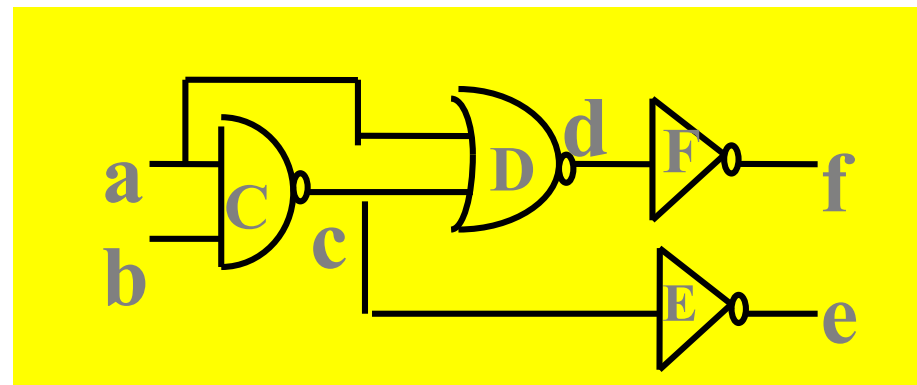
## 模拟过程



a	b	c	d	e	f
1	1	1	0	0	1

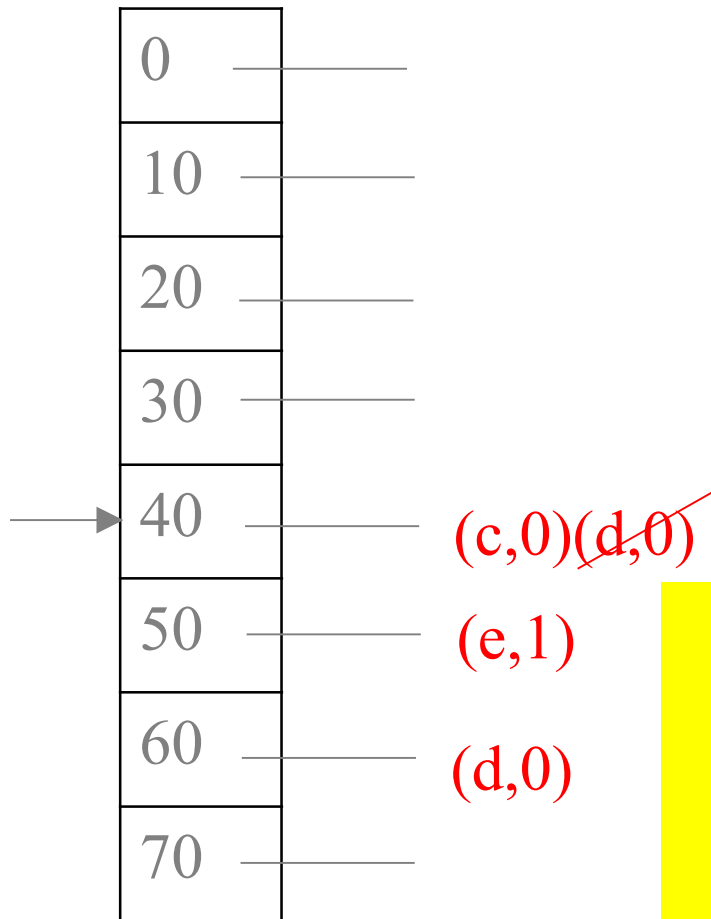
~~(e,1)(d,0)~~

(c,0)(d,0)

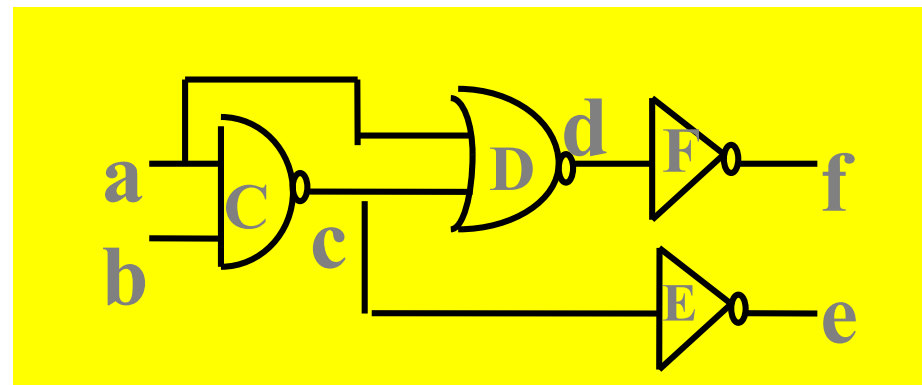




## 模拟过程

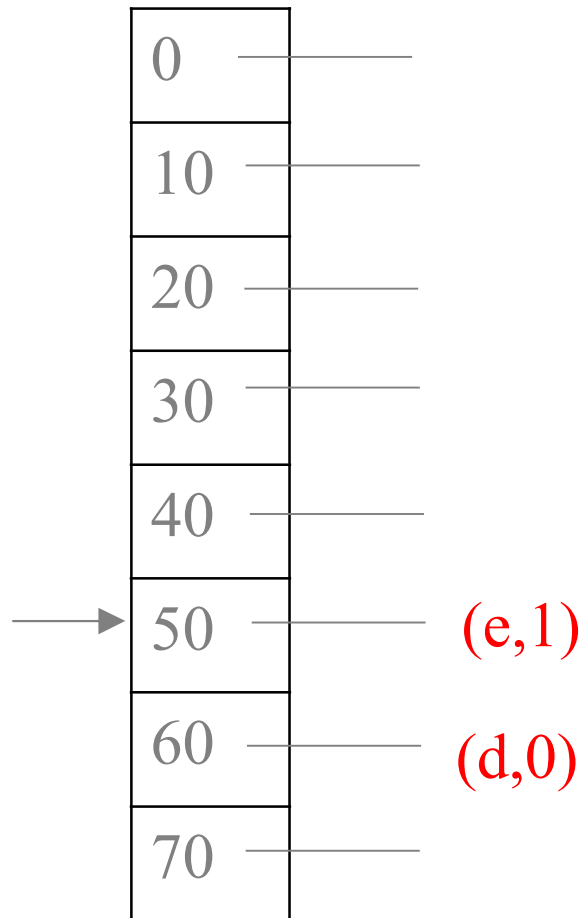


a	b	c	d	e	f
1	1	0	0	0	1

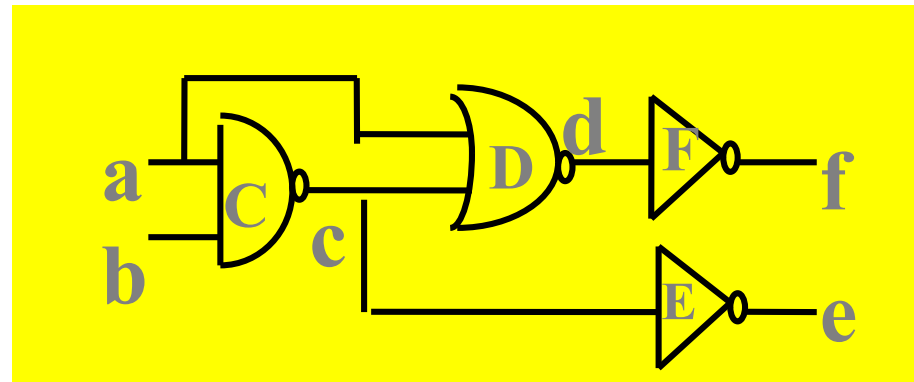




## 模拟过程

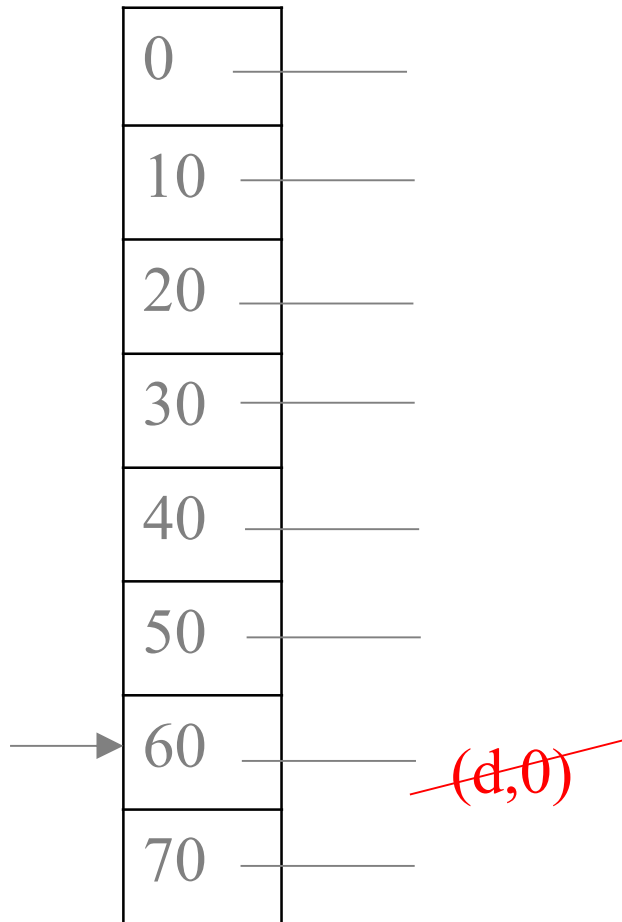


a	b	c	d	e	f
1	1	0	0	1	1

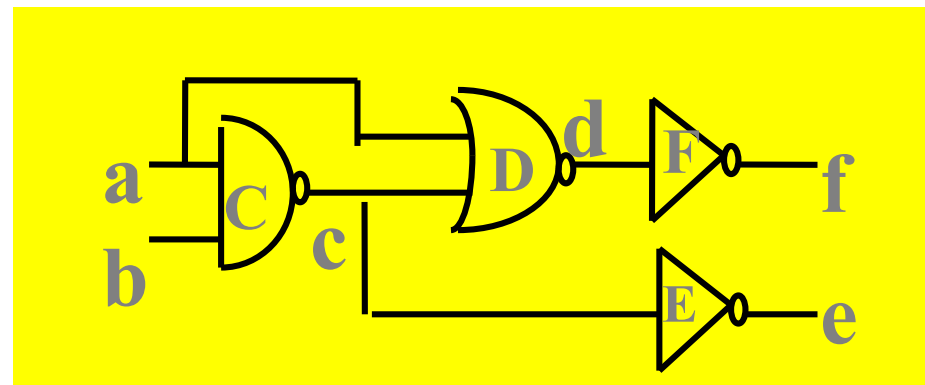




## 模拟过程

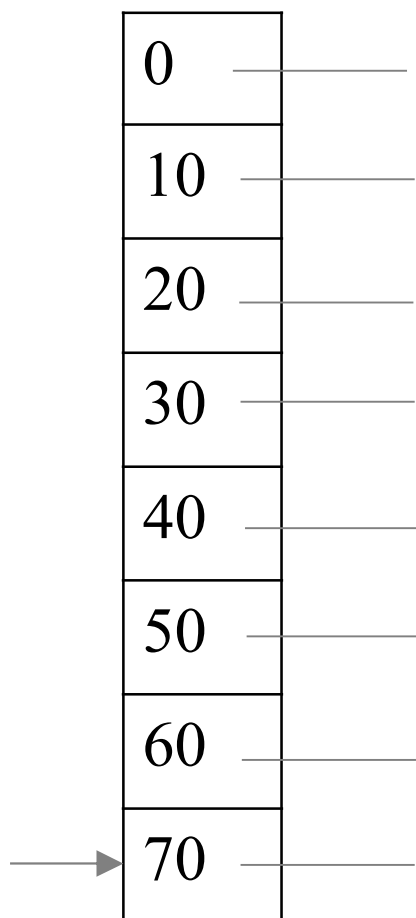


a	b	c	d	e	f
1	1	0	0	1	1

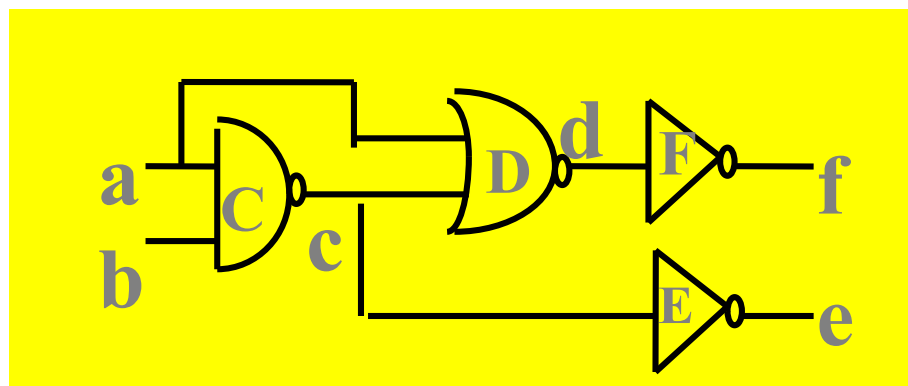




## 模拟过程

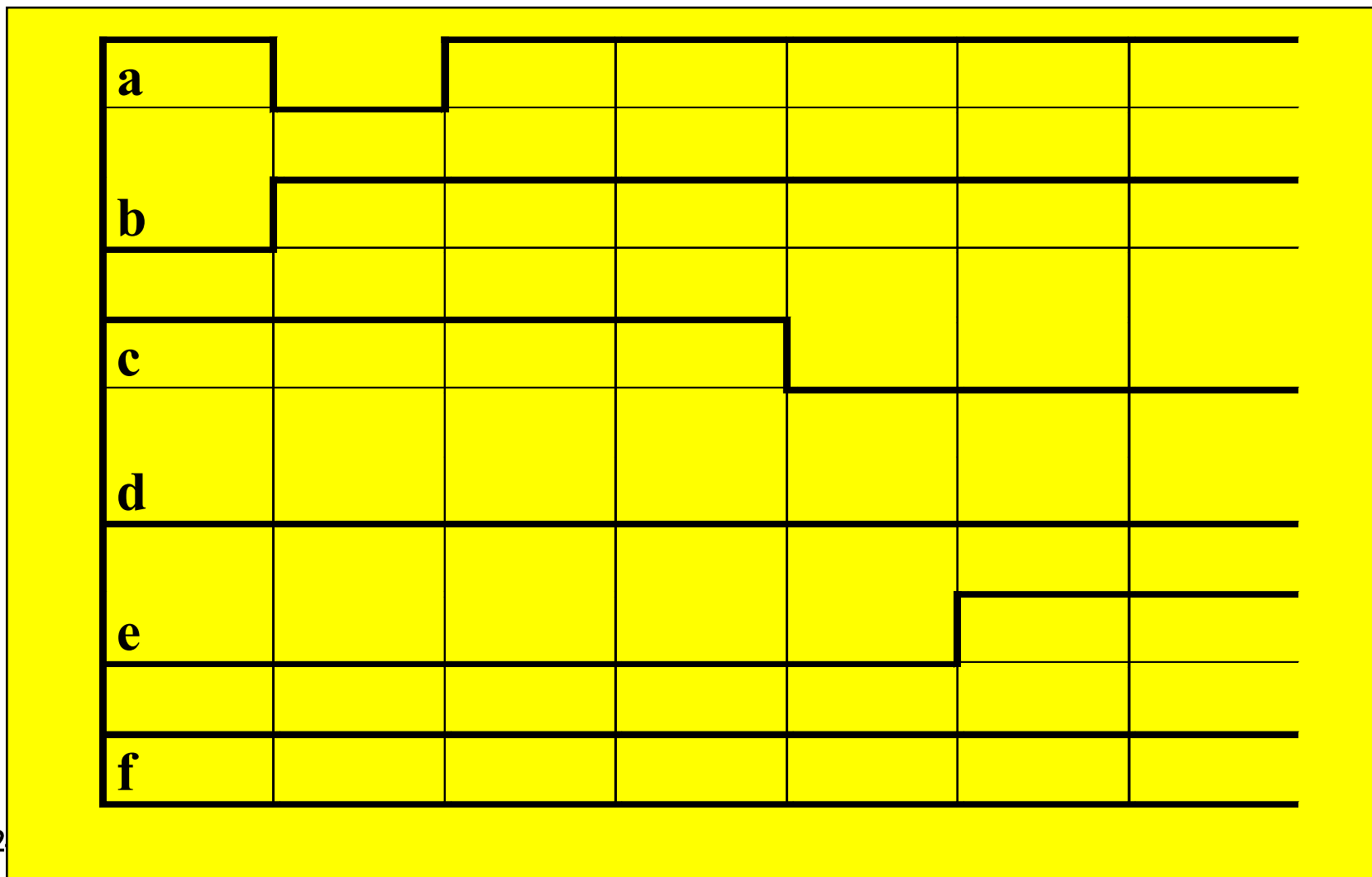


a	b	c	d	e	f
1	1	0	0	1	1





## 模拟结果波形图

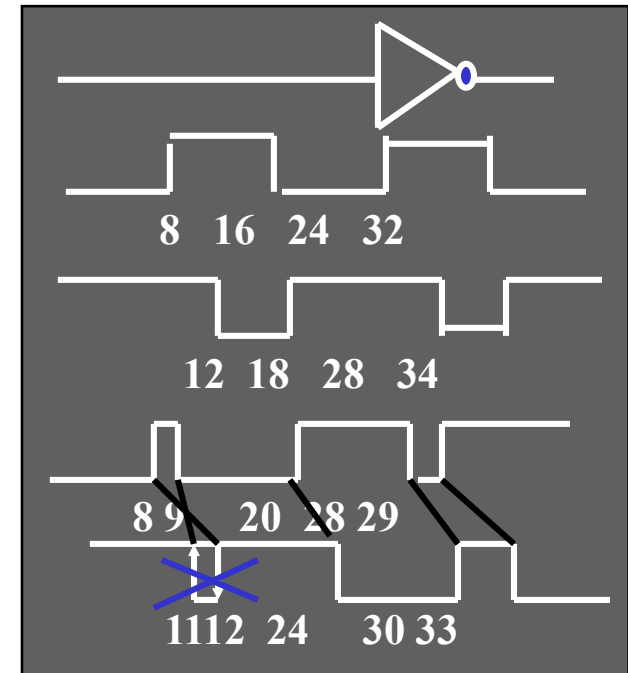




## 事件冲突的处理

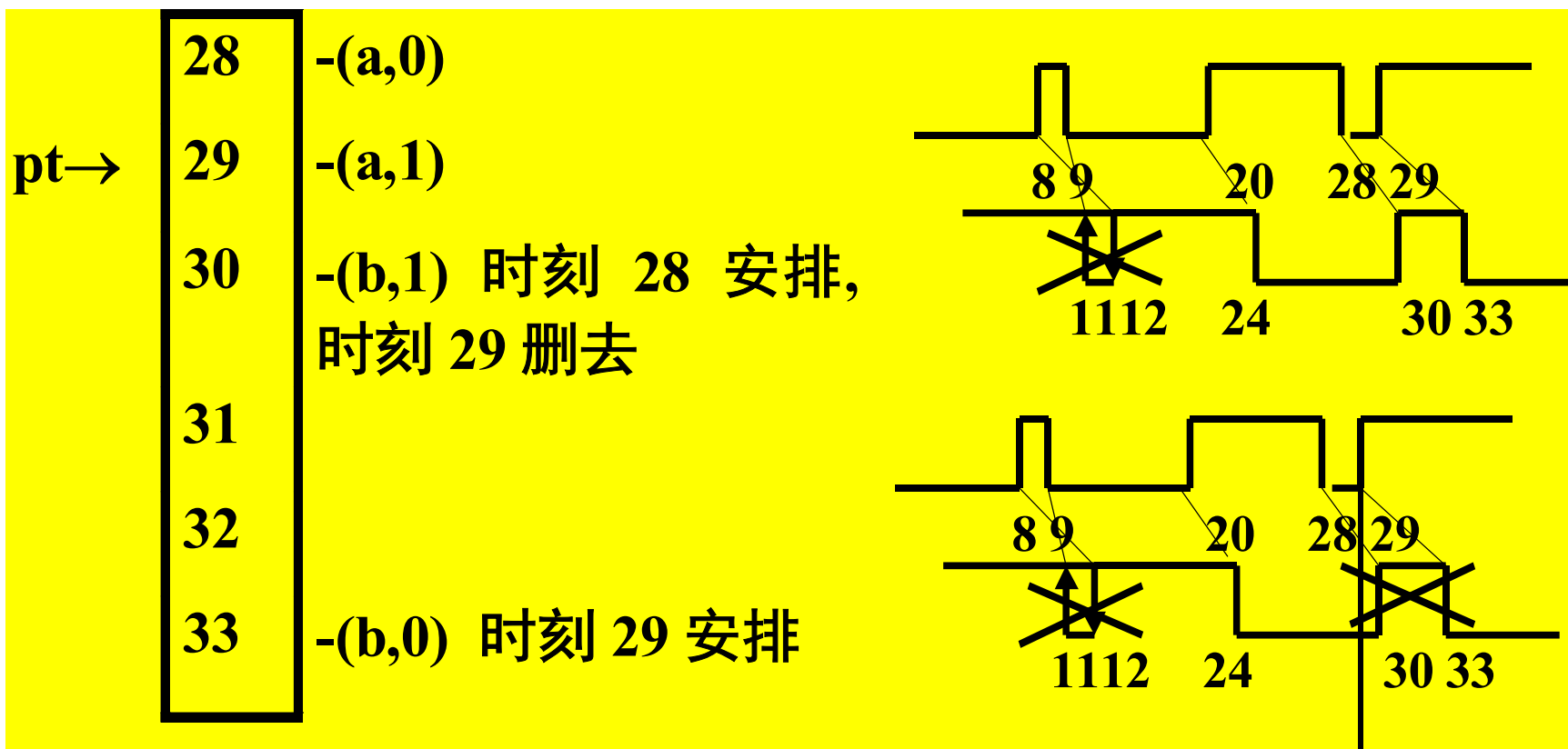
pt→	8	-(a,1)	设:上升延迟 = 2 ns; 下降 延迟= 4 ns;
	9	-(a,0)	
	10		
	11	-(b,1) 时刻 9 安排	
	12	-(b,0) 时刻 8 安排,	
		时刻 9 删去	
	-		

在时刻9安排(b, 1, 11)时,  
将后面的(b, 0, 12)删去。





## 惯性延迟的处理



在29时刻, 安排 (b,0,33), 删去(b,1,30)