

## 5.6 用Verilog HDL描述锁存器和触发器

---

### 5.6.1 时序电路建模基础

### 5.6.2 锁存器和触发器的Verilog建模

## 5.6.1 时序电路建模基础

---

Verilog行为级描述用关键词initial或always, 但initial是面向仿真, 不能用于逻辑综合。

always是无限循环语句, 其用法为:

always@(事件控制表达式 (或敏感事件表))

begin

块内局部变量的定义;

过程赋值语句;

end

## 敏感事件分为电平敏感事件和边沿触发事件:

---

**电平敏感事件（如锁存器）：**

**`always@(sel or a or b)`**

**sel、a、b中任意一个电平发生变化，后面的过程赋值语句将执行一次。**

**边沿敏感事件（如触发器）：**

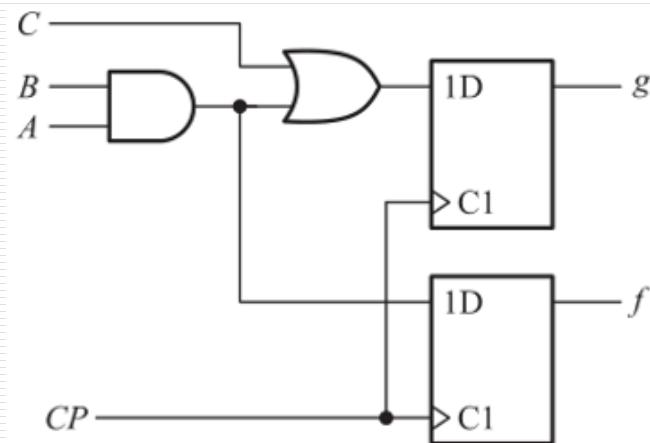
**`always@(posedge CP or negedge CR)`**

**CP的上升沿或CR的下降沿来到，后面的过程语句就会执行。**

## 过程赋值语句有阻塞型和非阻塞型:

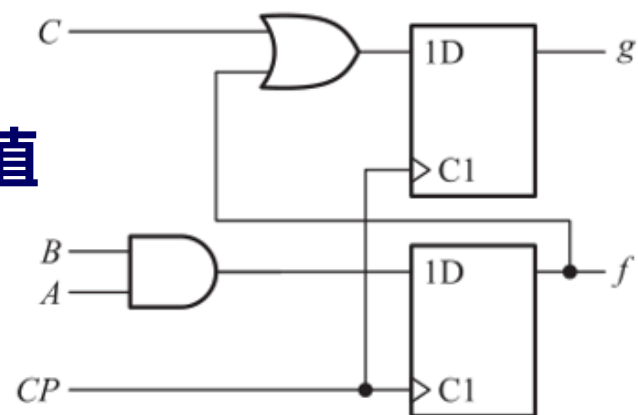
阻塞型用 “=” 表示，多条语句**顺序**执行。

```
begin
  f = A & B;
  g = f | C;
end
```



非阻塞型用 “<=” 表示，语句块内部的语句**并行**执行。

```
begin
  f <= A & B;
  g <= f | C;    // g用CP上升沿前的f值
end
```



## 5.6.2 锁存器和触发器的Verilog建模实例

```
module D_latch (Q, D, E); //D锁存器的描述
    output Q;
    input D, E;
    reg Q;
    always @(E or D)
        if (E) Q <= D;    //Same as: if (E== 1)
endmodule
```

```
module DFF (
    output reg Q,
    input D, CP
); //D触发器的描述
    always @(posedge CP)
        Q <= D;
endmodule
```

**例：分析下面Verilog模块,说明其逻辑功能。**

---

```
module async_set_rst_DFF (  
    output reg Q, QN,  
    input D, CP, Sd, Rd  
);  
always @(posedge CP, negedge Sd, negedge Rd)  
begin  
    if (~Sd || ~Rd)  
        if (~Sd ) begin Q <= 1'b1; QN <= 1'b0; end//异步置数  
        else      begin Q <= 1'b0; QN <= 1'b1; end//异步清零  
    else  
        begin Q <= D; QN <= ~D; end  
    end  
endmodule
```