



EDA技术

Electronic Design Automation

LAB1讨论

翟建芳



3—8译码器的几种表达方式

- [case语句](#)
- [if 语句](#)
- [for语句](#)
- [assign语句](#)

阻塞与非阻塞赋值

例1 :

```
module non_block1;
    reg a, b, c, d, e, f;
    initial begin // blocking assignments
        a = #10 1; //
        b = #2 0; //
        c = #4 1; //
    end
    initial begin // non- blocking assignments
        d <= #10 1; //
        e <= #2 0; //
        f <= #4 1; //
    end
    initial begin
        $monitor($ time,," a= %b b= %b c= %b d= %b e= %b f= %b", a, b, c, d, e, f);
        #100 $finish;
    end
endmodule
```

阻塞与非阻塞赋值

输出结果：

0 a= x b= x c= x d= x e= x f = x

2 a= x b= x c= x d= x **e= 0** f = x

4 a= x b= x c= x d= x e= 0 **f = 1**

10 **a= 1** b= x c= x **d= 1** e= 0 f = 1

12 a= 1 **b= 0** c= x d= 1 e= 0 f = 1

16 a= 1 b= 0 **c= 1** d= 1 e= 0 f = 1

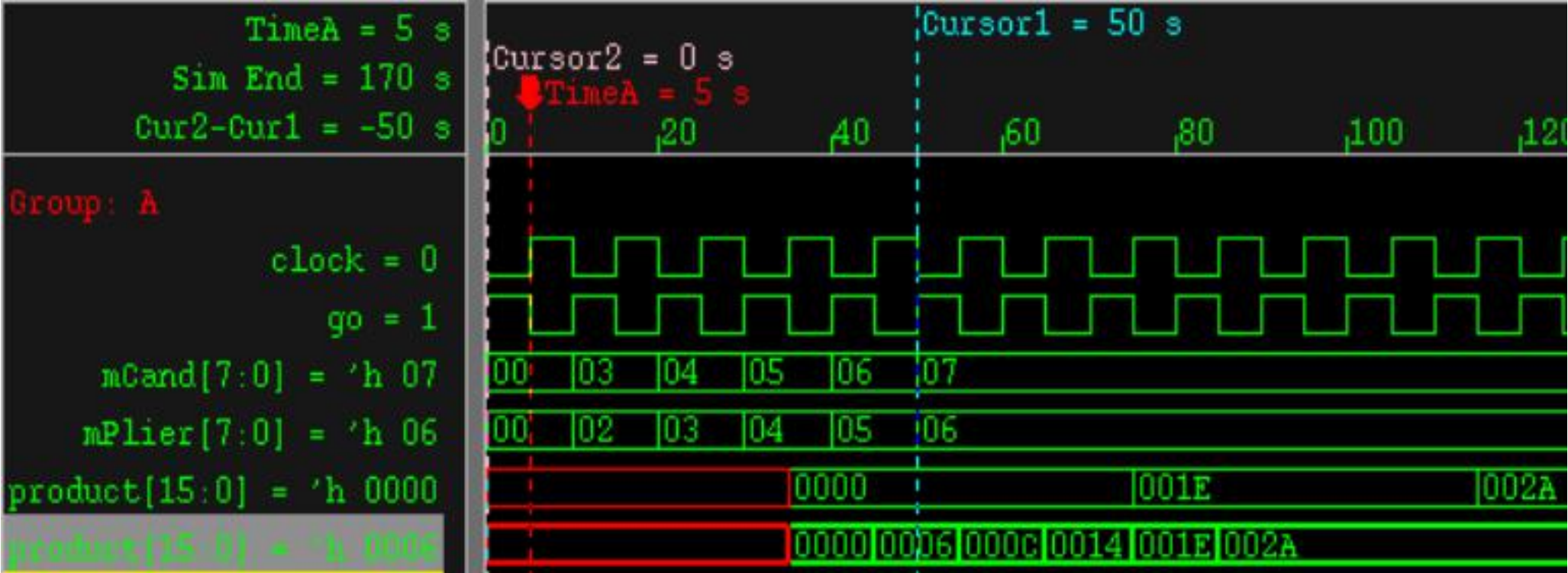
阻塞与非阻塞赋值

例2：

```
module pipeMult(product, mPlier, mCand, go, clock);  
input      go, clock;  
input [7:0] mPlier, mCand;  
output [15:0] product;  
reg [15:0] product;  
always @(posedge go)  
    product = repeat (4) @(posedge clock) mPlier * mCand;  
endmodule
```

```
module pipeMult(product, mPlier, mCand, go, clock);  
input      go, clock;  
input [7:0] mPlier, mCand;  
output [15:0] product;  
reg [15:0] product;  
always @(posedge go)  
    product <= repeat (4) @(posedge clock) mPlier * mCand;  
endmodule
```

阻塞与非阻塞赋值



阻塞与非阻塞赋值

阻塞赋值

- 求出RHS表达式的值并完成赋值行为
- 这个过程不能被其它的Verilog描述打断

非阻塞赋值

- 在一个时间步的开始求RHS表达式的值并在这个时间步结束时用RHS的值替换LHS；
- 在求RHS表达式和替换LHS变量的中间时间段，其它的Verilog语句和非阻塞赋值操作可以被执行。

阻塞与非阻塞赋值

阻塞赋值or非阻塞赋值？

- 当为**时序**逻辑建模，使用“非阻塞赋值”。
- 当为**锁存器**（latch）建模，使用“非阻塞赋值”。
- 当用always块为**组合逻辑建模**，使用“阻塞赋值”。
- 当在同一个always块里面**既为组合逻辑又为时序逻辑建模**，使用“非阻塞赋值”。
- 不要在同一个always块里面**混合**使用“阻塞赋值”和“非阻塞赋值”
- 不要在**两个或两个以上**always块里面对同一个变量进行赋值。
- 使用**\$strobe**以显示已被“非阻塞赋值”的值。
- 不要用**#0**延迟的赋值。

The End