

Verilog HDL语言

# Verilog程序设计方法

---- 时序电路

主讲：卢 萍

华中科技大学计算机科学与技术学院

# 主要内容

- ❑ 分频器
- ❑ 计数器
- ❑ 只读存储器
- ❑ 实验3 简单时序电路设计

# 参数parameter

有时候希望模块成为一般化的模块，即希望端口位数可选。

parameter可实现此功能，在调用模块时可改变该参数的值

```
module mux2_1(out1, a, b, sel) ;  
    parameter N=2; //本模块内不可变  
    output [N-1:0] out1;  
    input [N-1:0] a, b;  
    input sel;  
  
    assign out1= sel ? b : a;  
endmodule
```

**parameter:** 是个常量  
范围 -> 本module内有效  
作用 -> 模块间参数传递

在顶层模块中：

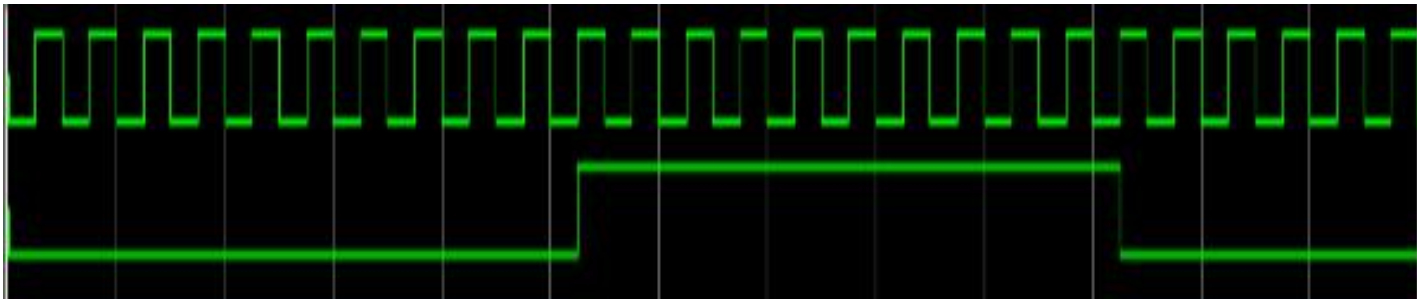
```
mux2_1 #(4) dut0( out, x,y,s ); //实例化时参数N值为4
```

# 分频器

**分频：**将信号的频率降低为原来的 $1/N$ ，就叫N分频。

**分频器：**实现分频的电路。

**分频的实现：**通过计数实现分频，当计数器从0计数到 $(N/2-1)$ 时，输出时钟翻转，计数器清零



20分频

# 分频器



```
module divider(input clk, output reg clk_N);
```

```
    parameter N=20;    // 参数说明，默认值20
```

```
    reg [31:0] count;
```

```
    always @( posedge clk) begin
```

```
//带参数值的模块引用
```

```
divider #(100) (clk100M,clk1M);
```

```
or
```

```
divider #(100) D1(clk100M,clk1M);
```

```
//或者在调用divider的上层模块中用//defparam  
来改写参数值
```

```
defparam D1.N=100;
```

```
divider  D1(clk100M,clk1M);
```

```
    end
```

```
endmodule
```

# 简单加法计数器

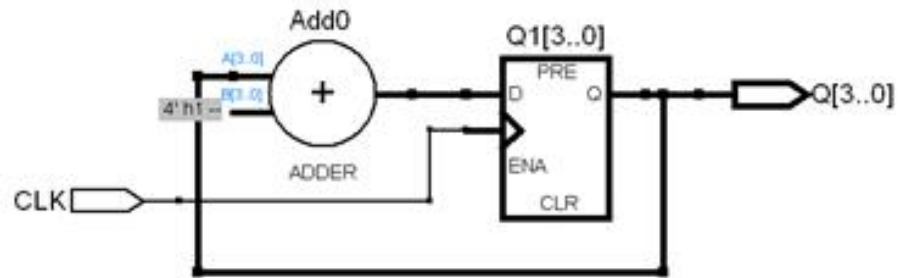


图 5-17 4 位加法计数器 RTL 电路图

```
module counter(clk, out);  
input clk;           // 计数时钟  
output [2:0] out;     // 计数值  
  
always @(posedge clk) begin // 在时钟上升沿计数器加1  
    |                 // 功能实现  
end  
endmodule
```

# 只读存储器

```
module rom8x4(addr, data);  
input [2:0] addr;           // 地址  
output [3:0] data;          // 地址addr处存储的数据  
  
reg [3: 0] mem [7: 0];      // 8个4位的存储器  
  
initial begin               // 初始化存储器  
    ....  
end  
...                          // 读取addr单元的值输出  
endmodule
```