数字系统高级设计技术

主 讲：何宾

Email：hebin@mail.buct.edu.cn

2019.06

逻辑复制和复用--逻辑复制

扇出是指某一器件输出驱动与之相连的后续器件的能力。

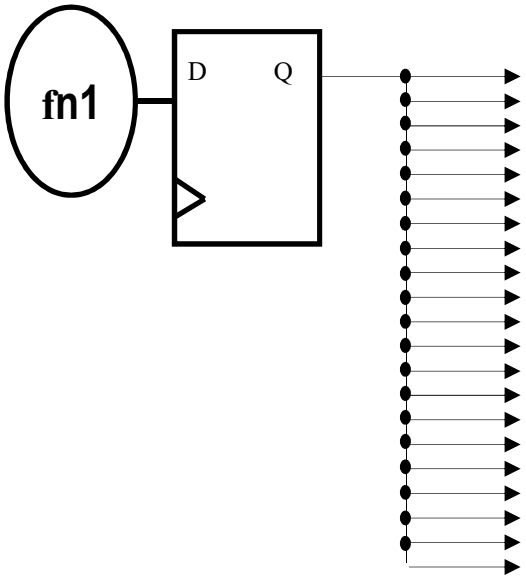
- **一个器件的扇出数是有限制的。扇出数目越多，所要求的驱动能力越高。**
- **在FPGA芯片内，如果一个逻辑单元的扇出数过多的话，会降低其工作速度，并且会对布线造成困难。**
- **在FPGA逻辑资源允许的情况下，要尽量降低扇出数。**

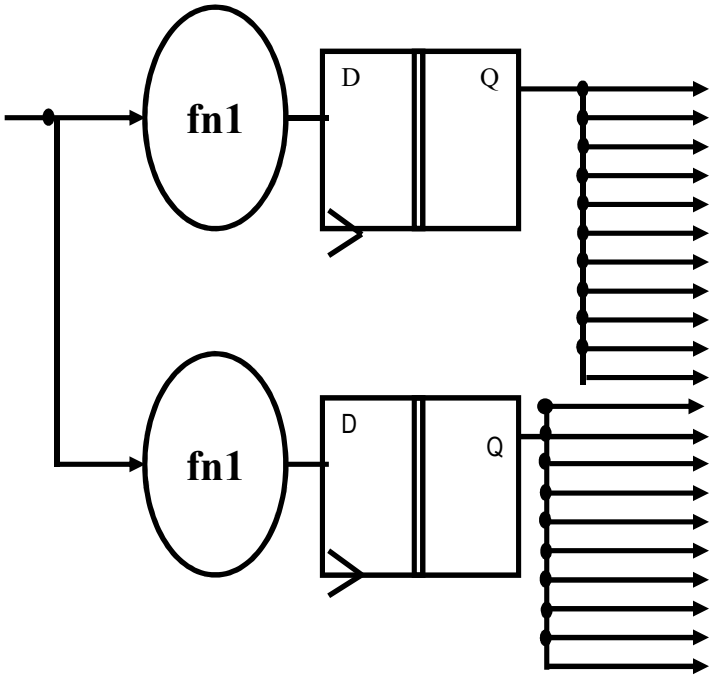
逻辑复制和复用
--逻辑复制

逻辑复制是通过增加面积而改善设计时序的优化方法，经常用于调整信号的扇出。

- 如果信号具有高的扇出，则要添加缓存器来增强驱动能力，但这会增大信号的时延。
- 通过逻辑复制，使用多个相同的信号来分担驱动任务。这样，每路信号的扇出就会变低，就不需要额外的缓冲器来增强驱动，即可减少信号的路径延迟。

逻辑复制和复用
--逻辑复制





逻辑复制和复用
--逻辑复制

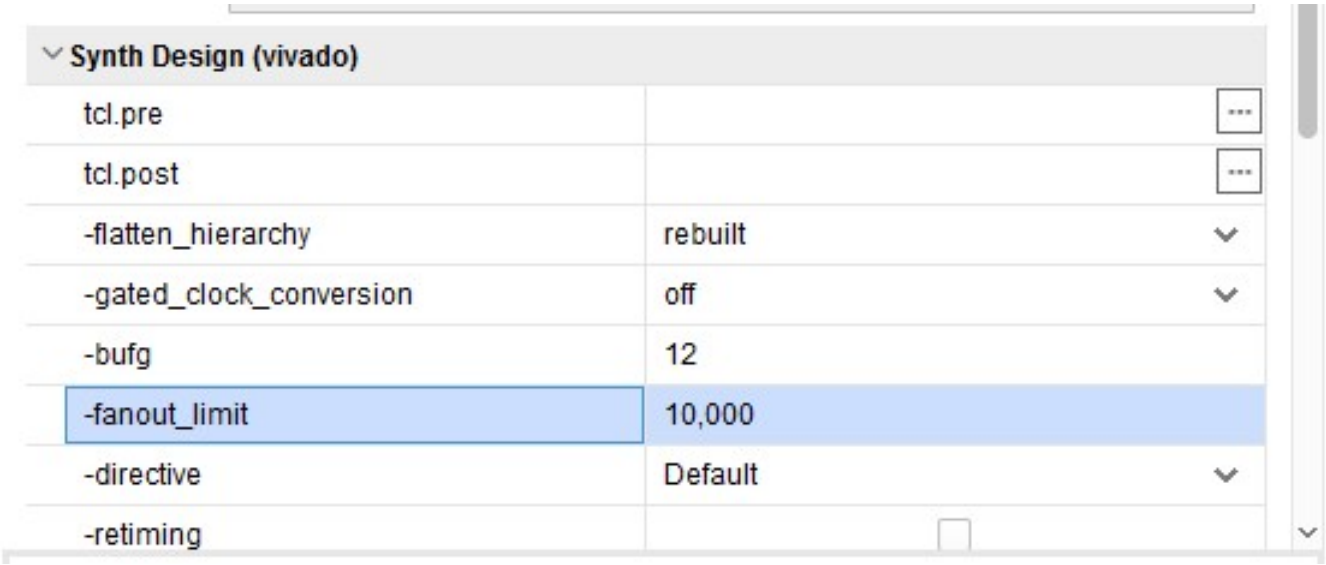
通过逻辑单元的复制，减少扇出数，可以解决下面两个方面的问题

- 减少网络延迟。
- 多个器件分布在不同的区域，这样可以大大降低布线阻塞情况的发生。

注：在使用增加器件减少扇出数目的时候，必须要注意的是，如果是异步单元的话，必须对该单元进行同步处理。

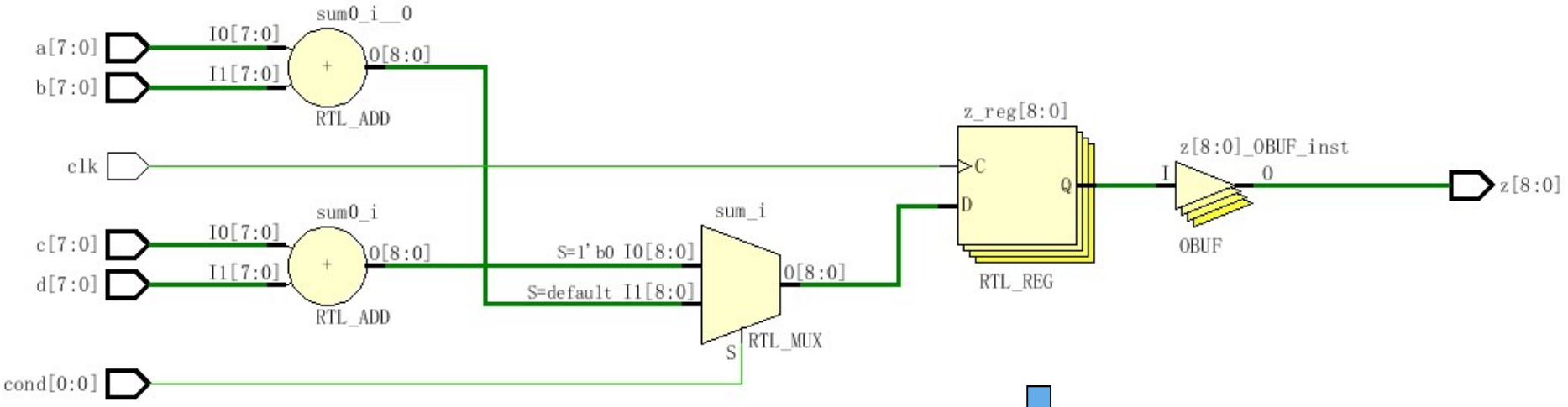
逻辑复制和复用
--逻辑复制

在Vivado工具的综合属性设置选项中，提供了调整扇出个数的功能。



逻辑复制和复用技术
--逻辑复用

逻辑复用是指在完成相同的功能下，尽量减少所使用的逻辑单元的数目。

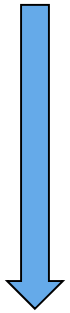


先加后选择的结构

对应的Verilog HDL代码

逻辑复制和复用技术
--逻辑复用

```
module top(
    input clk,
    input [7:0] c,
    input [7:0] d,
    input [0:0] cond,
    input [7:0] a,
    input [7:0] b,
    output reg[8:0] z
    );
reg [8:0] sum;
```



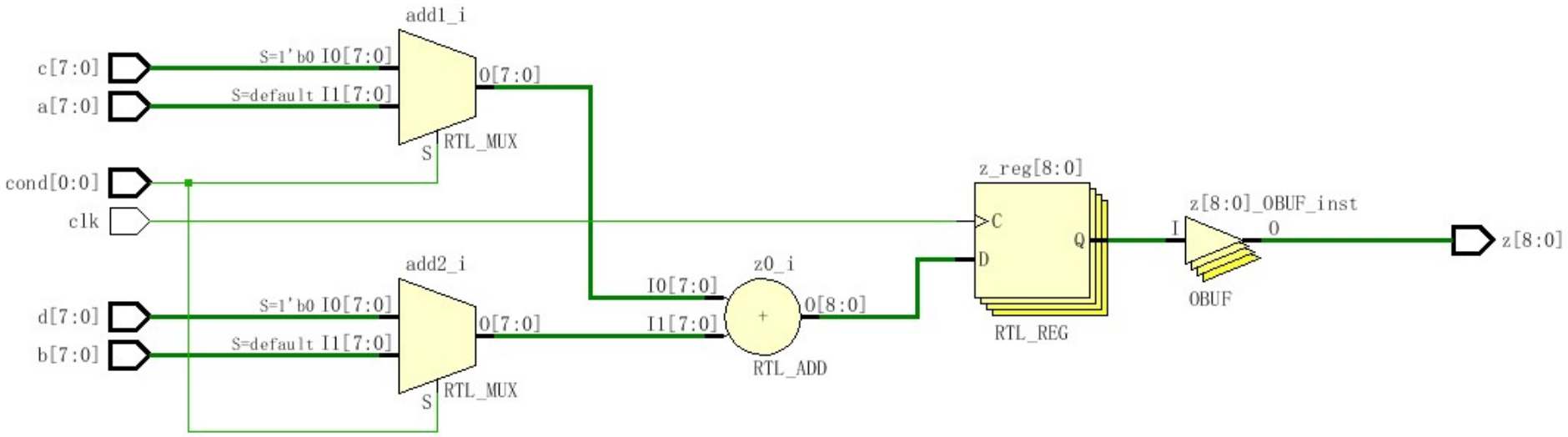
继续下一页

逻辑复制和复用技术
--逻辑复用

```
always@(*)
begin
case (cond)
  1'b0    :  sum=c+d;
  default :  sum=a+b;
endcase
end

always@(posedge clk)
begin
  z<=sum;
end
endmodule
```

逻辑复制和复用技术
--逻辑复用

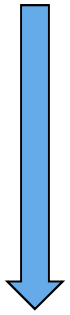


先选择后加的结构

对应的**Verilog HDL代码**

逻辑复制和复用技术
--逻辑复用

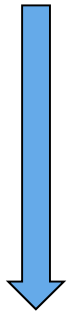
```
module top(
    input clk,
    input [7:0] c,
    input [7:0] d,
    input [0:0] cond,
    input [7:0] a,
    input [7:0] b,
    output reg[8:0] z
    );
reg [7:0] add1;
reg [7:0] add2;
```



继续下一页

逻辑复制和复用技术
--逻辑复用

```
always@(*)
begin
 case (cond)
  1'b0    : begin
                add1=c;
                add2=d;
            end
  default : begin
                add1=a;
                 add2=b;
            end
 endcase
end
```



继续下一页

逻辑复制和复用技术
--逻辑复用

```
always@(posedge clk)
begin
    z<=add1+add2;
end
endmodule
```