**Verilog中单if语句和多if语句**

1. 单if语句（if-else if-else if-else）

单if语句综合出来的电路有优先级，从电路的角度来说，单if语句综合出来的电路类似于数据选择器，else路选通的条件是前面所有的if条件都不满足。

always@(\*) begin

z = 0;

if(sel1)

z = a;

else if(sel2)

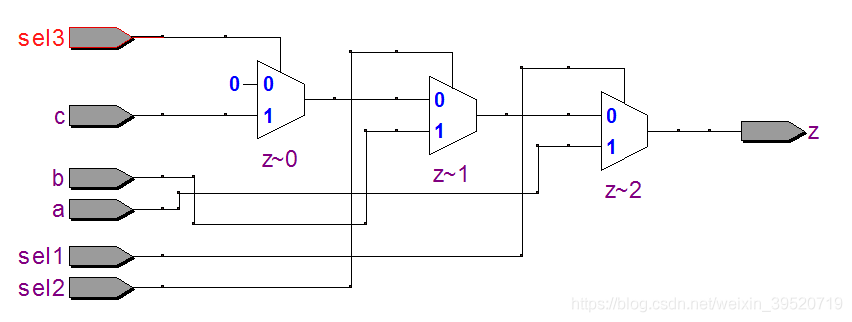
z = b;

else if(sel3)

z = c;

end

综合后生成的RTL为：



此时，最后一级if的优先级最低，在电路中表现为越靠前。由于综合器的不同，可能综合成的电路有所差距。

2、 多if语句（if-if-if）

多if语句综合出来的电路也是有优先级的，越靠后的if对电路的输出起决定性作用最大。若if条件不互斥，即使前面的if条件满足，若最后一级if满足，也将选择最后一级if的内容输出。

always@(\*) begin

z = 0;

if(sel1)

z = a;

if(sel2)

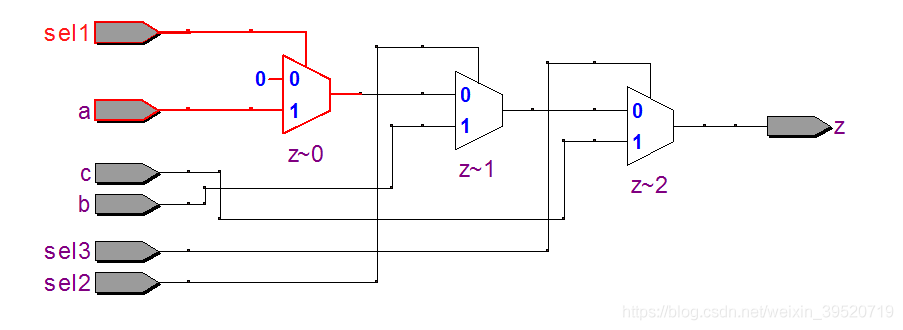
z = b;

if(sel3)

z = c;

end

综合后生成的电路为：



此时最后一级if的优先级最高，在电路中表现为越靠近输出（越靠后）。

3、case语句

case语句一般生成无优先级的判断结构，与if语句的区别是，它的判断条件一般互斥，多用于译码电路。

在组合电路中，如果不是所有分支都列出，且未使用default，则会生成latch。

当case语句的条件分支不互斥，这时会生成有优先级的判断结构，加上synopsis parallel case原语后，则是告诉综合器，所有的条件互斥，为并行结构无优先级。

always@(sel) begin

{int2, int1, int0} = 3'b0;

casez(sel) //synopsys parallel case

3'b1??: int2 = 1'b1;

3'b?1?: int1 = 1'b1;

3'b??1: int0 = 1'b1;

endcase

end

参考资料链接：<https://blog.csdn.net/weixin_39520719/article/details/107809003>