1. |, & — bitwise operator

```
assign c = a | b;
```

2. &&, || — Logical operators

```
(state == idle) || (state == op) && (count > 10))
```

3. { } — Concatenation operators

```
wire [3:0] a4;

Wire [7:0] b8, c8;
assign b8 = {a4, a4};
assign d8 = {b8[3:0], c8[3:0]};
  assign c8 = {b8[7:2], 2'b00};
```

4. Conditional Operator

```
assign max = (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c);
```

5. Expression bit-length adjustment

从左看到右，以最长位为标准，将右手边的参数都扩展到最长位，然后执行操作，如右将结果按左手边位数进行调整。

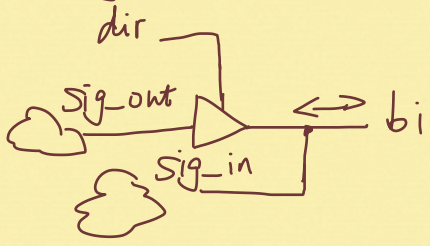
```
wire [7:0] a, b;

Wire [7:0] sum1, sum2 ;
assign sum1 = (a + b + 0) >> 1;
assign sum2 = (a + b) >> 1;
```

0是32位整数，首先对a,b扩展到32位，运算，然后取成8位

sum2都是8位，无须扩展到32位。

对有符号的整数也无差别化重置

6. 关于三



三组态 bidirectional port 有用

注：bi要申明为 inout 类型

```
module bi_demo (
    inout wire bi,
    .....
);
assign sig_out = output_expression;
assign bi = (dir)? sig_out : 1'bz;
assign sig_in = bi;
```