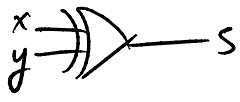
Recall adding two bits.

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$\begin{array}{r} x \\ +\,y \\ \hline C\,S \end{array} \qquad \begin{array}{r} 0 \\ +\,0 \\ \hline 0\,0 \end{array} \quad \begin{array}{r} 0 \\ +\,1 \\ \hline 0\,1 \end{array} \quad \begin{array}{r} 1 \\ +\,0 \\ \hline 0\,1 \end{array} \quad \begin{array}{r} 1 \\ +\,1 \\ \hline 1\,0 \end{array}$$

$C = xy$

$S = \bar{x}y + x\bar{y} = x \oplus y$ (XOR)



3-input XOR

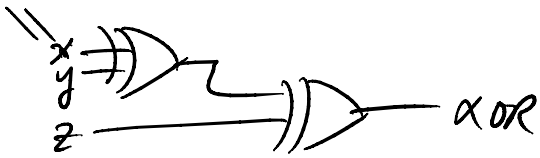
x	y	z	XOR
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$= x \oplus y \oplus z$

$= (x \oplus y) \oplus z$

$= x \oplus (y \oplus z)$

associative



(output = 1 when there is odd numbers of 1 on the inputs)

extend to adding multi-bit #s. $A = a_2 a_1 a_0$ and $B = b_2 b_1 b_0$

eg. $A = 010$ add to $B = 111$

$$\begin{array}{rrrr} & 1 & 1 & 0 \\ A & 0 & 1 & 0 \\ B + & 1 & 1 & 1 \\ \hline & 1 & 0 & 0 \end{array} \qquad \begin{array}{rrrr} & c_2 & c_1 & c_0 \\ & a_2 & a_1 & a_0 \\ + & b_2 & b_1 & b_0 \\ \hline & c_3 & s_2 & s_1 \quad s_0 \end{array}$$

c_i	b_i	a_i	c_{i+1}	s_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
→ 0	1	1	1	0
1	0	0	0	1
→ 1	0	1	1	0
→ 1	1	0	1	0
→ 1	1	1	1	1

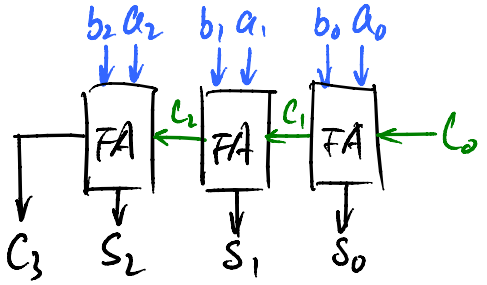
$$\begin{array}{r} c_{i+1}\,c_i \\ a_i \\ +)\ b_i \\ \hline s_i \end{array}$$

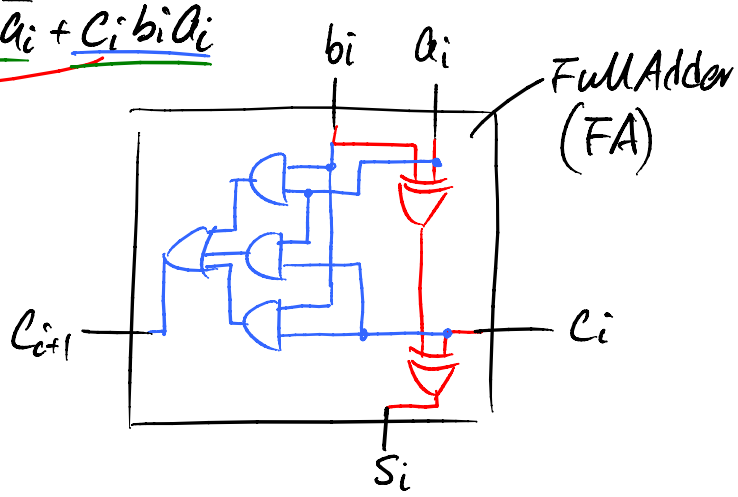
Logic expression for outputs.

$s_i = a_i \oplus b_i \oplus c_i$

$c_{i+1} = \bar{c}_i b_i a_i + c_i \bar{b}_i a_i + c_i b_i \bar{a}_i + c_i b_i a_i$

$= b_i a_i + c_i a_i + c_i b_i$





```
module FA (input a, b, Cin, output S, Cout);
    assign S = a ^ b ^ Cin;
    assign Cout = (a&b) | (a&Cin) | (b&Cin);
end module
```

FA in Verilog.

```
module adder (A, B, Cin, S, Cout);
    input [2:0] A, B;
    input Cin;
    output [2:0] S;
    output Cout;
    wire C1, C2;
    FA Bit0 (A[0], B[0], Cin, S[0], C1);
    FA Bit1 (A[1], B[1], C1, S[1], C2);
    FA Bit2 (A[2], B[2], C2, S[2], Cout);
end module
```