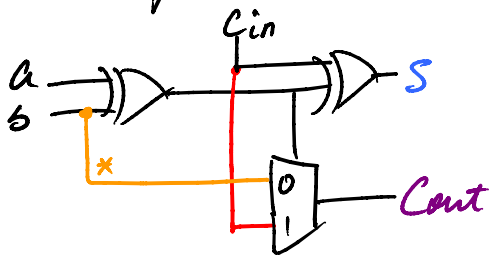
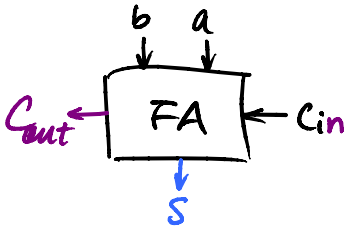
3.2-3 Adder + Subtractor

recall a full adder circuit



Cin	b	a	Cout	S
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



$S = Cin \oplus a \oplus b$

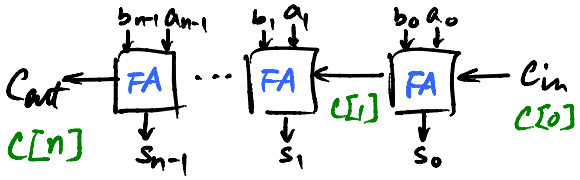
* a_i will work as well

Verilog code for a FA: conditional assignment

{ if a=b, Cout = b
else Cout = Cin

```
module FA (input Cin, a, b, output Cout, S);
  assign S = Cin ^ a ^ b;
  assign Cout = (a==b) ? b : Cin;
endmodule
```

⇒ n-bit ripple-carry adder



Verilog code using a "For" loop

```
module Add_n (Cin, A, B, Cout, S);
  parameter n = 4;
  :                    // A, B, S as [n-1, 0] vectors
  reg [0:n] C;
  always @(*)
    begin
      C[0] = Cin;
      for (k=0; k<n; k=k+1)
        begin
          S[k] = A[k] ^ B[k] ^ C[k];
          C[k+1] = (A[k]==B[k]) ? A[k] : C[k];
        end
      Cout = C[n];
    end
endmodule
```

the "+" operator

$\{C[k+1], S[k]\} = A[k] + B[k] + C[k]$

The "+" operator used on multi-bit vectors

```
module Add_n (Cin, A, B, Cout, S);
  parameter n = 4;
  :                    // A, B, S are [n-1, 0] vectors
  always @(*)
    {Cout, S} = A + B + Cin;
endmodule
```

3.3 Signed numbers (2's complement)

eg. $5+1=6$

$5-1=4$

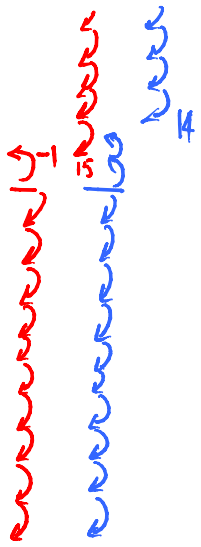
$(0101)+(1111)=(0100)$ } $-1 \rightarrow 1111$

$5-2=3$

$(0101)+(1110)=(0011)$ } $-2 \rightarrow 1110$

4-bit binary

0	0 0 0 0
+1	0 0 0 1
+2	0 0 1 0
+3	0 0 1 1
+4	0 1 0 0
+5	0 1 0 1
+6	0 1 1 0
+7	0 1 1 1
−8	1 0 0 0
−7	1 0 0 1
−6	1 0 1 0
−5	1 0 1 1
−4	1 1 0 0
−3	1 1 0 1
−2	1 1 1 0
−1	1 1 1 1



4 bits signed # : $-8 \rightarrow +7$

n bits signed # : $-2^{n-1} \rightarrow 2^{n-1}-1$

4 bits, to represent $-k$ we do

2^4-k eg. find 2's comp. of −3

$10000 \leftarrow 2^4$ (5 bits)

$-00011 \leftarrow -3$

$01101 \leftarrow$ This is −3 in 2's comp. rep.

not easy to do

let's do this instead:

$2^4-1-k+1$ = $1111 \leftarrow 2^4-1$

$-0011 \leftarrow -3$

1100

$+0001 \leftarrow +1$

$1101 \leftarrow$ 2's complement of −3 using 4-bits

easy to work with

2's complement

all −ve # have "1" as the msb.

Short-cut to find 2's comp

eg. 4-bits, we want to represent −6

+6 0110

1010 ← −6 in 2's comp.

→ going from right (LSB) to left (MSB)

→ copy down all "0's and the 1st "1" that you see

→ complement the remaining bits.

Let's make an adder into a adder/subtractor

$A - B \Rightarrow A + (-B)$

$-B$: 2's comp of $-B$

This is how we can find the 2's comp of $-B$

use an example where B is 3

$2^4 - 1 \rightarrow 1\ 1\ 1\ 1$

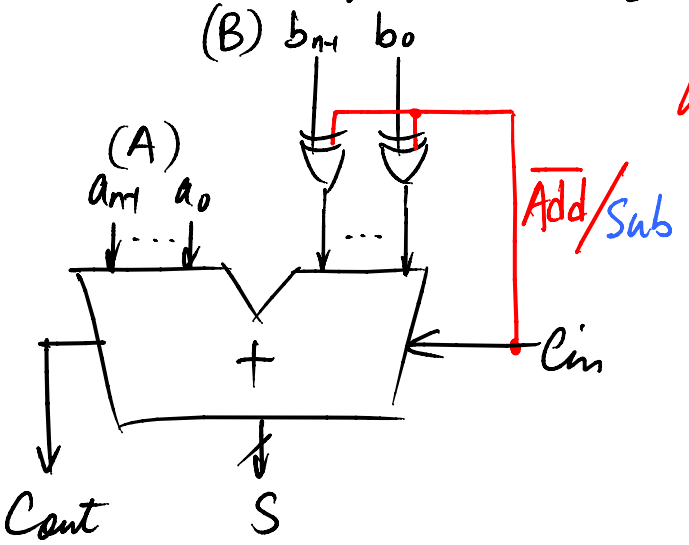
$-3 \quad -0\ 0\ 1\ 1$

(This result is in fact the complemented value of 3) $\rightarrow 1\ 1\ 0\ 0$ ← Step 1: $\overline{B}$

$+1 \quad +0\ 0\ 0\ 1$ ← Step 2: +1

(This is the 2's comp. representation of −3) $\rightarrow 1\ 1\ 0\ 1$ ← the result is 2's comp. of $-B$

in general



When performing addition, Cin is set to 0, outputs of ⏚ are $b_{n-1} \cdots b_0$ hence A+B is performed.

When performing subtraction, Cin is set to 1, outputs of ⏚ will be $\overline{b}_{n-1} \cdots \overline{b}_0$,
hence A + (2's comp of −B) = A − B is performed.