Lecture 13 CS1110/CS5019 24/2/20

We will approach the operation of Subtraction by Converting the second operand into it negative equivalent and then adding:

for example: $7 - 2 \equiv 7 + (-2)$

But how do we negate a number? What do we have to do to 2 to get -2?

In two's Complement representation:

(i) Invert each bit

(ii) Add 1

0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	-8
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	-1

For 2: 2 = 0010

(i) Invert each bit → 1101

(ii) Add 1 → 1110 = -2

Negate -5: $-5 = 1011$

(i) Invert → 0100

(ii) add 1 → 0101 = 5

etc

0000	0	→	
0001	1	→ 1111	-1
0010	2	→ 1110	-2
0011	3	→ 1101	-3
0100	4	→ 1100	-4
0101	5	→ 1011	-5
0110	6	→ 1010	-6
0111	7	→ 1001	-7
1000	-8	→ 1000	-8
1001	-7	→ 0111	7
1010	-6	→ 0110	6
1011	-5	→ 0101	5
1100	-4	→ 0100	4
1101	-3	→ 0011	3
1110	-2	→ 0010	2
1111	-1	→ 0001	1
		0000	0

To negate a number x:

Get the bit-wise complement of x ($\bar{x}$) and add 1.

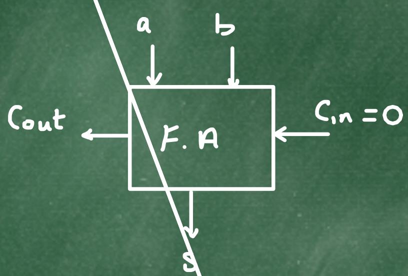
Note: zero, and the max negative number (-2^n, i.e., -8 in our example) are unaffected by this transformation.

In general:

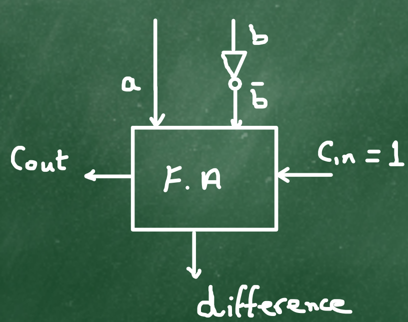
$$a \overset{\text{minus}}{-} b = a \overset{\text{plus}}{+} (\overset{\text{minus}}{-} b) = a \overset{\text{plus}}{+} (\overset{\text{not}}{\bar{b}} \overset{\text{plus}}{+} 1)$$

We see that we have reduced subtraction to a logical Inversion and the addition of 3 things.

This Puts us in mind of the F.A., which also add 3 things

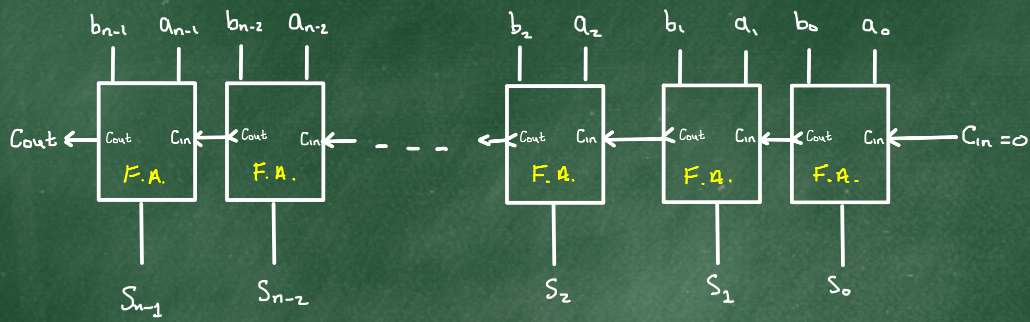


Since $a - b = a + (\bar{b} + 1)$ We can reuse our F.A. to do Subtraction by Inverting the b Input and by setting the C_{in} to 1, Instead of 0:



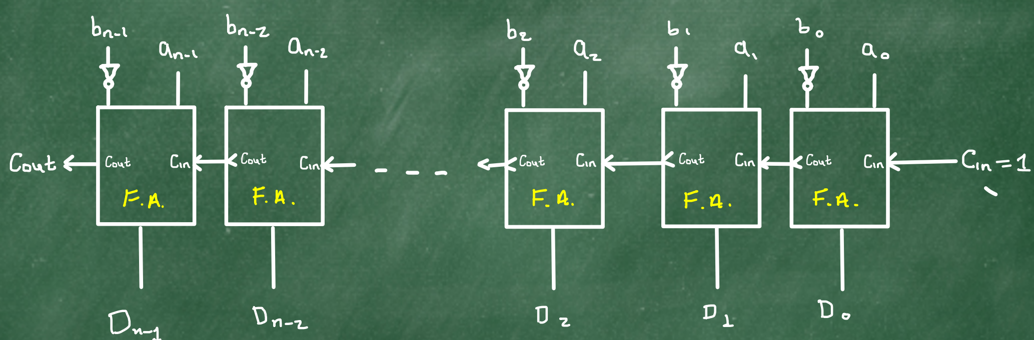
Mult-bit Subtraction

Recall



An n-bit multidigit Binary Ripple-Carry Addition

$$
\begin{array}{rrl}
 & a = & a_{n-2}\ a_{n-1}\ ---\ a_2\ a_1\ a_0 \\
+ & b = & b_{n-2}\ b_{n-1}\ \dots\ b_2\ b_1\ b_0 \\
 & & \qquad\qquad\qquad\qquad 0 \\
\hline
 & S = & \text{Cout}\ s_{n-2}\ s_{n-1}\ ---\ s_2\ s_1\ s_0
\end{array}
$$



An n-bit Ripple-Carry adder being used for Subtraction

$$
\begin{array}{rrl}
 & a = & a_{n-2}\ a_{n-1}\ ---\ a_2\ a_1\ a_0 \\
- & b = & \overline{b_{n-2}}\ \overline{b_{n-1}}\ \dots\ \overline{b_2}\ \overline{b_1}\ \overline{b_0} \\
 & & \qquad\qquad\qquad\qquad 1 \\
\hline
 & d & \text{Cout}\ d_{n-2}\ d_{n-1}\ \dots\ d_2\ d_1\ d_0
\end{array}
$$

Example Subtractions

① $5 - 6 = 5 + (-6) = 5 + ((\text{Inverse of } 6) + 1)$

$5 = 0101 \quad (\equiv a_3 a_2 a_1 a_0)$

$6 = 0110 \quad (= b_3 b_2 b_1 b_0)$

Inverse of $6 = 1001 \quad (= \bar{b}_3 \bar{b}_2 \bar{b}_1 \bar{b}_0)$

There $5-6$

$$
\begin{array}{r}
0101 \\
+\ 1001 \\
+\ \ \ \ \ \ 1 \\
\hline
1111
\end{array}
\quad (= C_{in} = 1)
$$

$= -1$, as required

② $-3 - 2 = -3 + (-2) = -3 + ((\text{Inverse } 2) + 1)$

$3 = 0011$,

$-3 = 1100 + 1 = 1101$

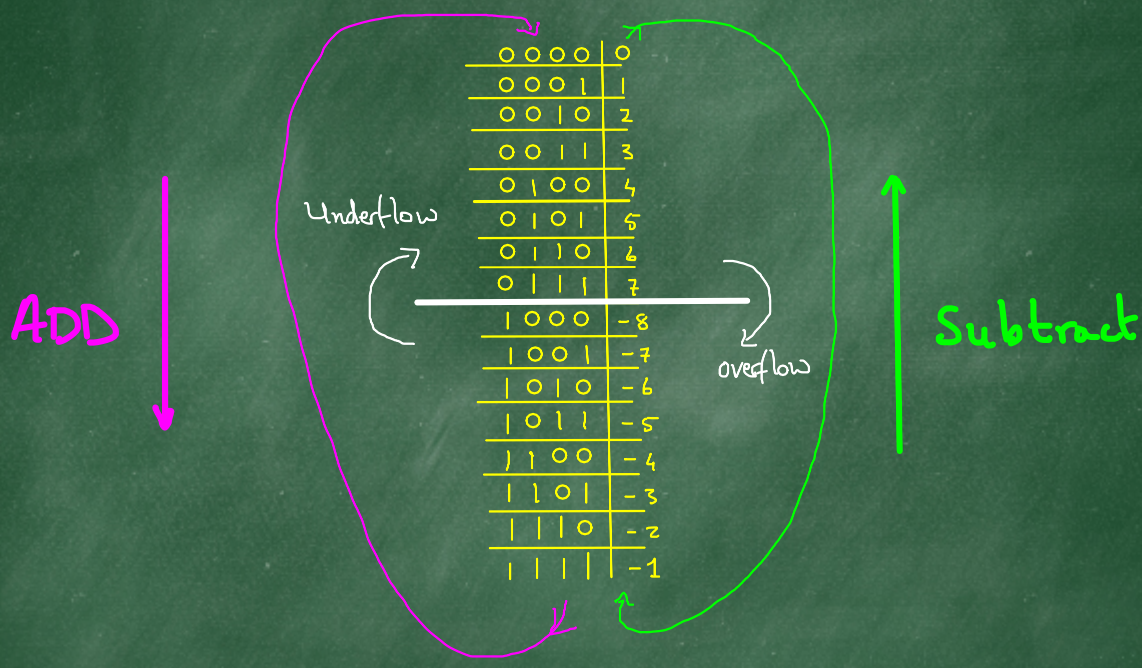
$2 = 0010$

Inverse of $2 = 1101$

Therefore $-3 - 2$

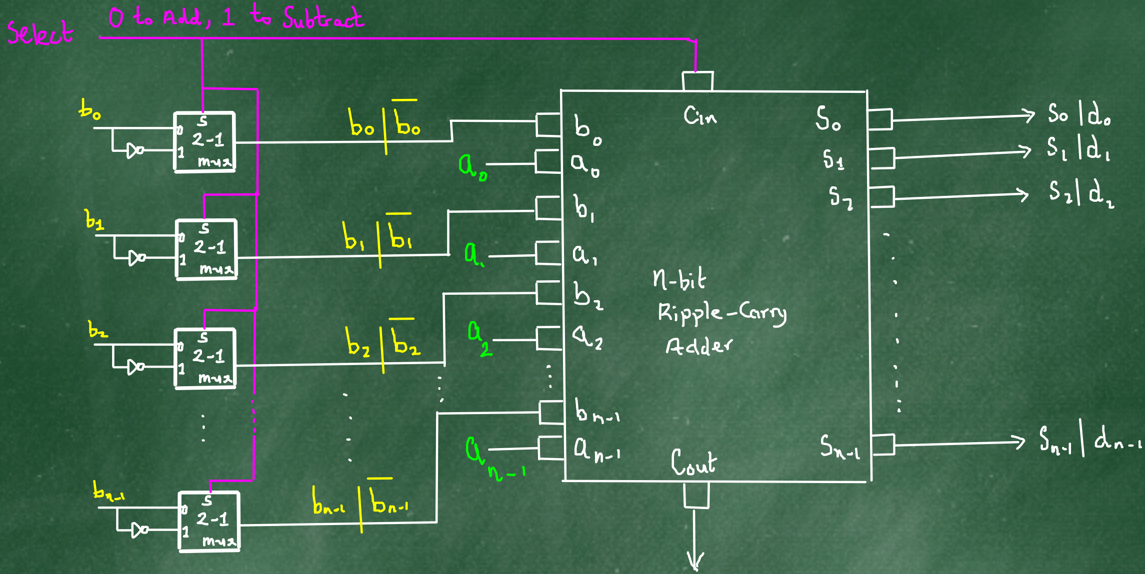
$$
= \begin{array}{r}
1101 \\
+\ 1101 \\
+\ \ \ \ \ \ 1 \\
\hline
1\ \ \underbrace{1011}_{= -5}
\end{array}
$$

$= -5$, as required.



ADD
Subtract
Underflow
overflow
0000 | 0
0001 | 1
0010 | 2
0011 | 3
0100 | 4
0101 | 5
0110 | 6
0111 | 7
1000 | -8
1001 | -7
1010 | -6
1011 | -5
1100 | -4
1101 | -3
1110 | -2
1111 | -1

A Circuit to both add and to Subtract



Select acts as the Select line for the multiplexers and as the C_{in} bit for the n-bit adder.

If Select = 0 we get $a+b$

If Select = 1 we get $a-b$

Therefore, Select can be viewed as and Instruction — choosing different Pathways through the Circuit.

Select = 0 ⇒ ADD a, b

Select = 1 ⇒ Sub a, b