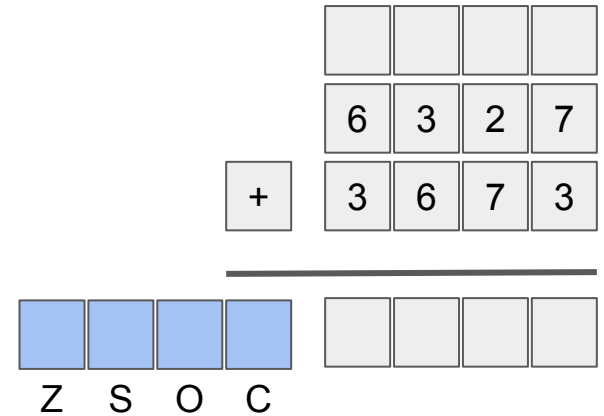
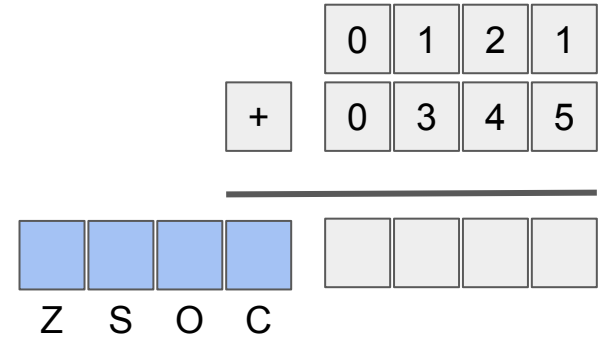
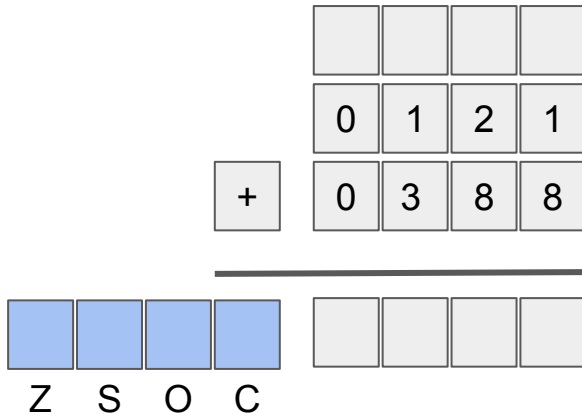


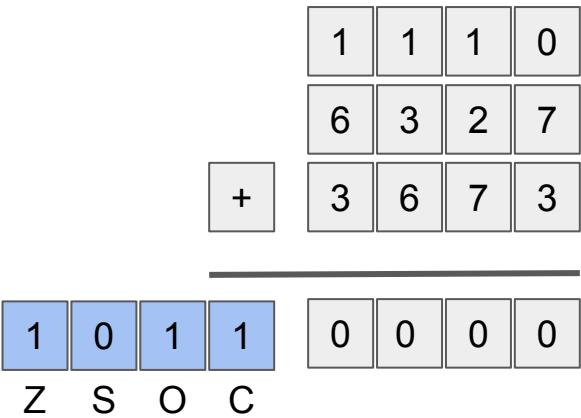
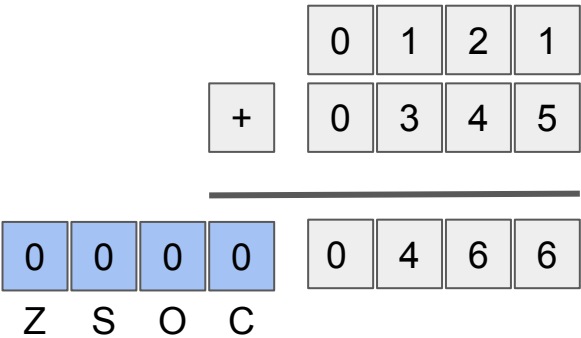
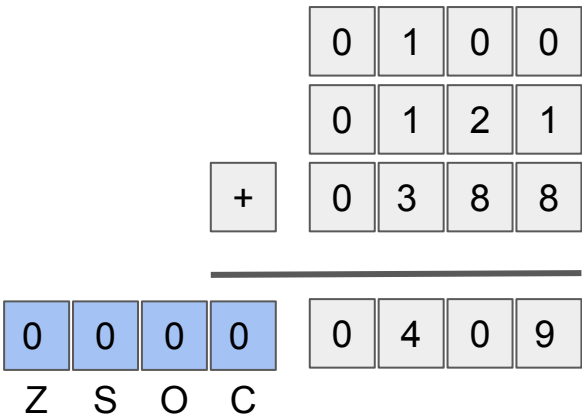
# Review of Mathematical Operations:

- Before
- First, introduce some status values:
  - Zero, Sign, Overflow, Carry
- Assume a register of size 4:



# Review of Mathematical Operations:

- After:
- First, introduce some status values:
  - Zero, Sign, Overflow, Carry
- Assume a register of size 4:



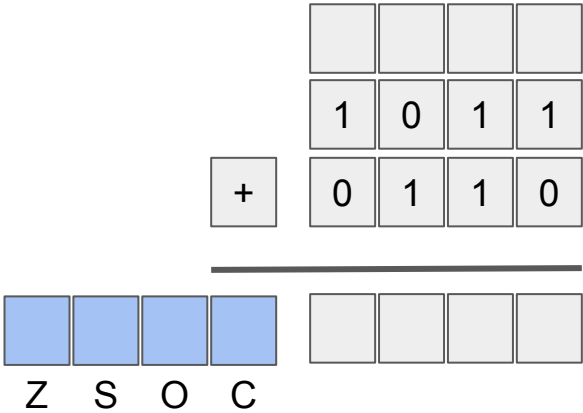
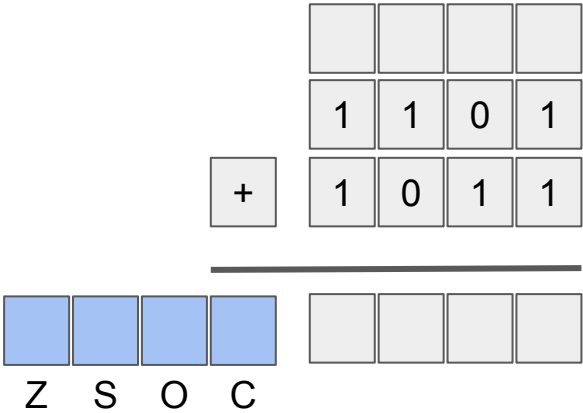
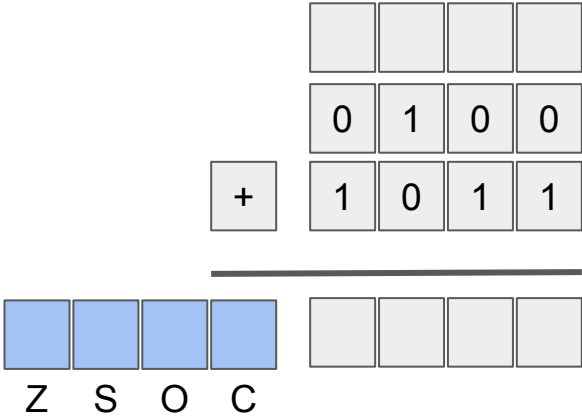
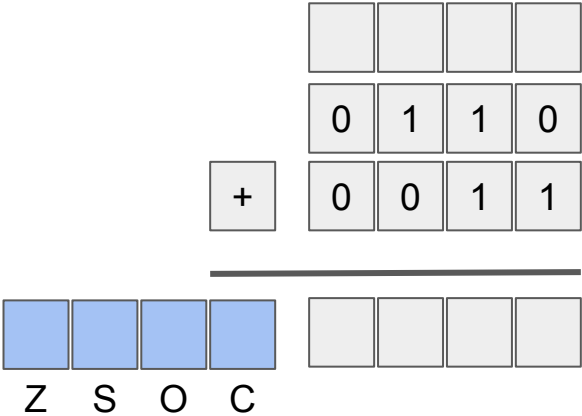
# In Binary (before)

C + A + B = C, S

C	A	B	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0

C + A + B = C, S

C	A	B	C	S
1	0	0	1	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



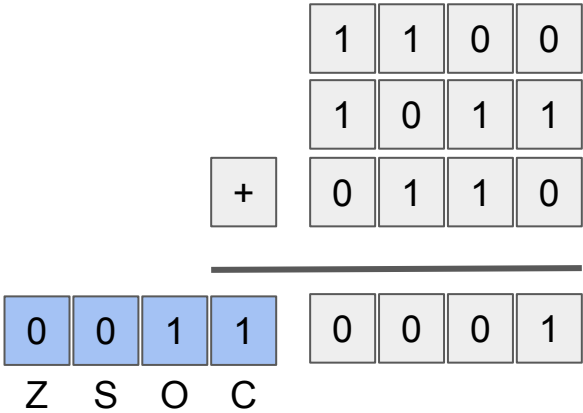
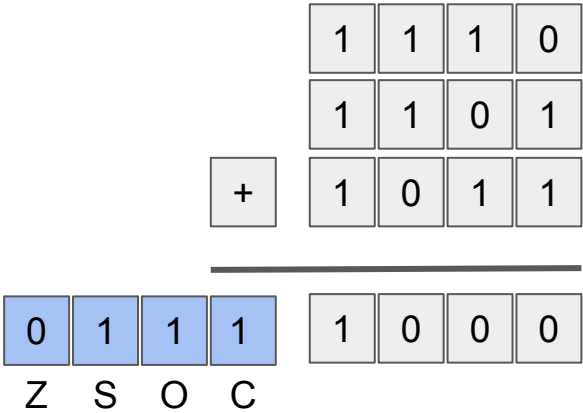
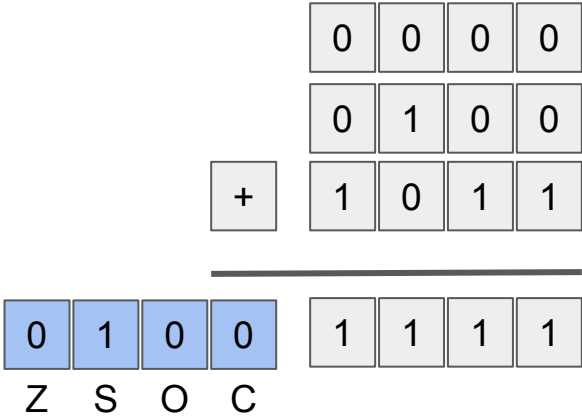
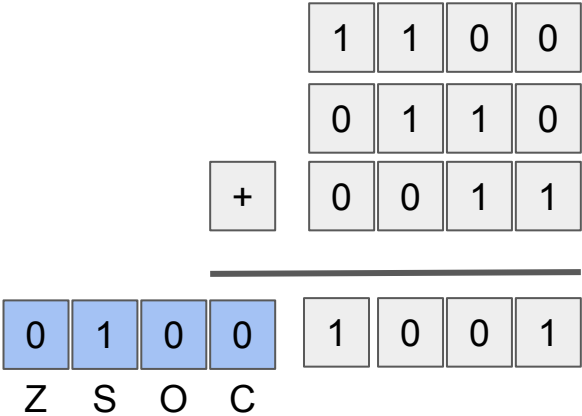
# In Binary (after)

$C + A + B = C, S$

C	A	B	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0

$C + A + B = C, S$

C	A	B	C	S
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

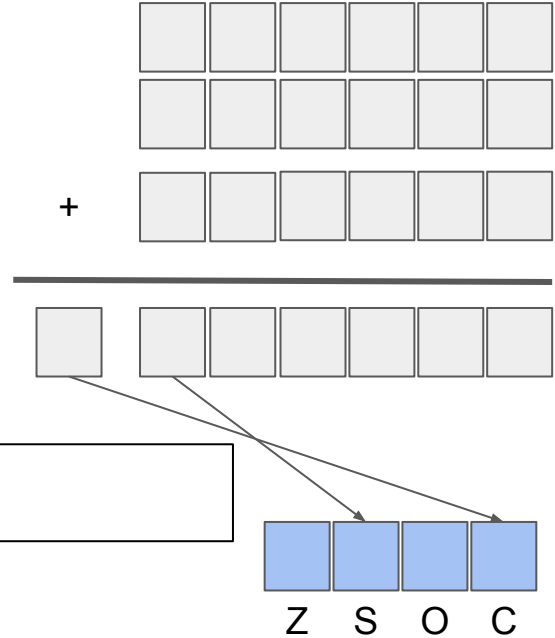


# Practice: Addition and Subtraction (template)

Steps:

1. transform the operation to addition
2. convert  $\text{abs}(X)$  to binary
3. compute 2's complement for negative numbers
4. perform binary addition
5. convert result to decimal
  - if negative result: compute 2's complement first

- $8 + 9$ :
- $7 - 4$ :
- $-5 + 2$ :
- $-16 - 3$ :



# Practice: Addition and Subtraction

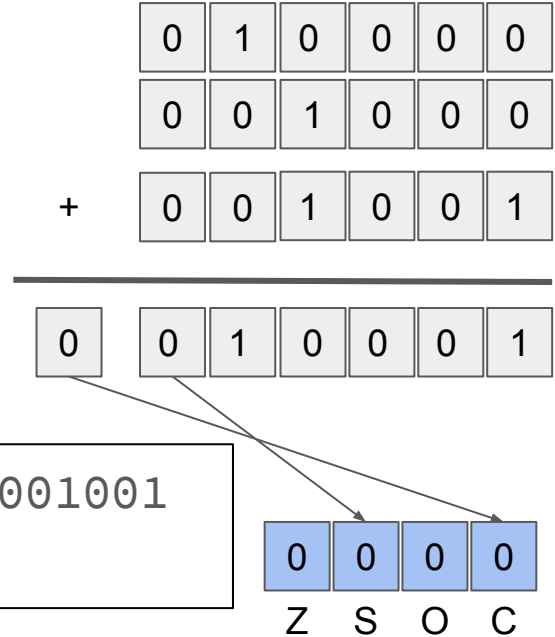
Steps:

1. transform the operation to addition
2. convert  $\text{abs}(X)$  to binary
3. compute 2's complement for negative numbers
4. perform binary addition
5. convert result to decimal
  - if negative result: compute 2's complement first

➤ 8 + 9:

- 7 - 4:
- -5 + 2:
- -16 - 3:

2.  $\text{abs}(8) = 001000$ ,  $\text{abs}(9) = 001001$   
5. 17 :  $8+9 = 17$  (correct)



# Practice: Addition and Subtraction

Steps:

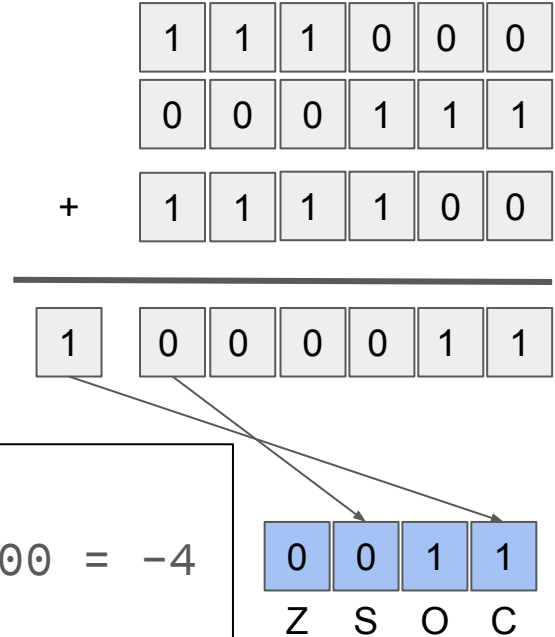
1. transform the operation to addition
2. convert  $\text{abs}(X)$  to binary
3. compute 2's complement for negative numbers
4. perform binary addition
5. convert result to decimal
  - if negative result: compute 2's complement first

- 8 + 9:
- 7 + (-4):
- -5 + 2:
- -16 - 3:

2. 000111, 000100

3. 111011+1 = 111100 = -4

5. 3: 7 - 4 = 3 (correct)



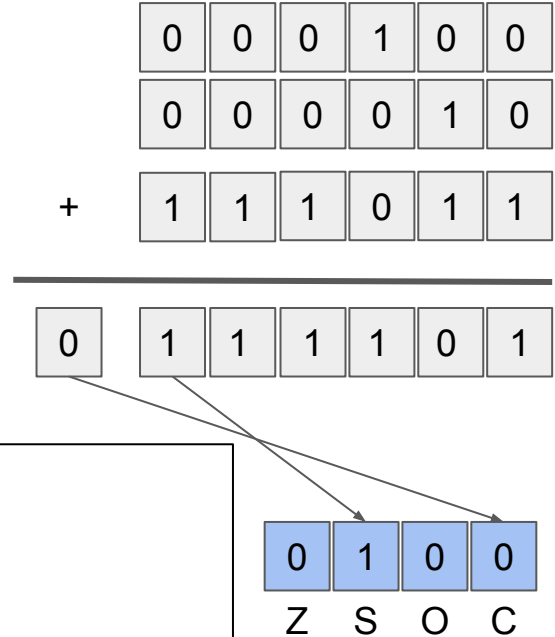
# Practice: Addition and Subtraction

Steps:

1. transform the operation to addition
2. convert  $\text{abs}(X)$  to binary
3. compute 2's complement for negative numbers
4. perform binary addition
5. convert result to decimal
  - if negative result: compute 2's complement first

- $8 + 9$ :
- $7 - 4$ :
- $(-5) + 2$ :
- $-16 - 3$ :

2. 5: 000101, 2: 000010  
3. -5: 111010 + 1: 111011  
5. 000010 + 1 = 000011 = 3  
-5 + 2 = -3 (correct)





# Practice: Addition and Subtraction

Steps:

1. transform the operation to addition
2. convert  $\text{abs}(X)$  to binary
3. compute 2's complement for negative numbers
4. perform binary addition
5. convert result to decimal
  - if negative result: compute 2's complement first

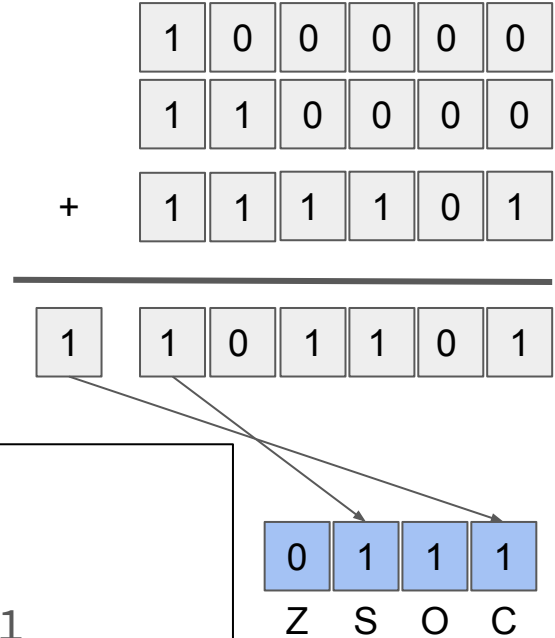
• 8 + 9:

• 7 - 4:

• -5 + 2:

➤ (-16) + (-3):

2.	16:	010000	3:	000011
3.	-16:	110000	-3:	111101
5.	-19:	$\text{abs}(010011), 101101$		



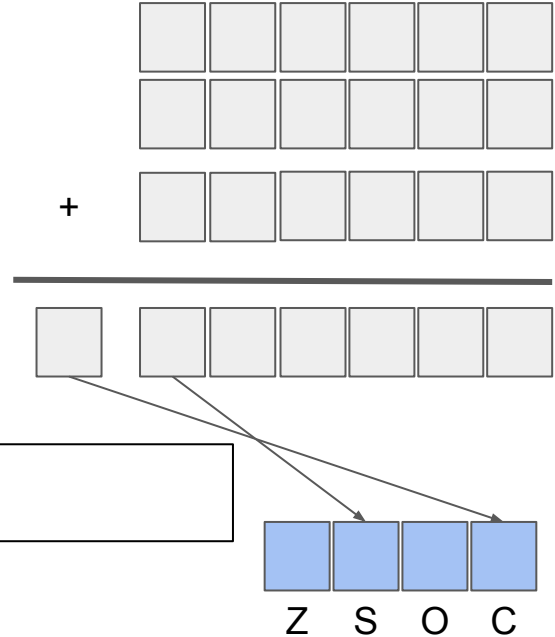
More Practice

# Practice: Addition and Subtraction (template)

Steps:

1. transform the operation to addition
2. convert  $\text{abs}(X)$  to binary
3. compute 2's complement for negative numbers
4. perform binary addition
5. convert result to decimal
  - if negative result: compute 2's complement first

- $4 + 5$ :
- $7 - 6$ :
- $-3 + 5$ :
- $-18 - 4$ :



# Practice: Addition and Subtraction

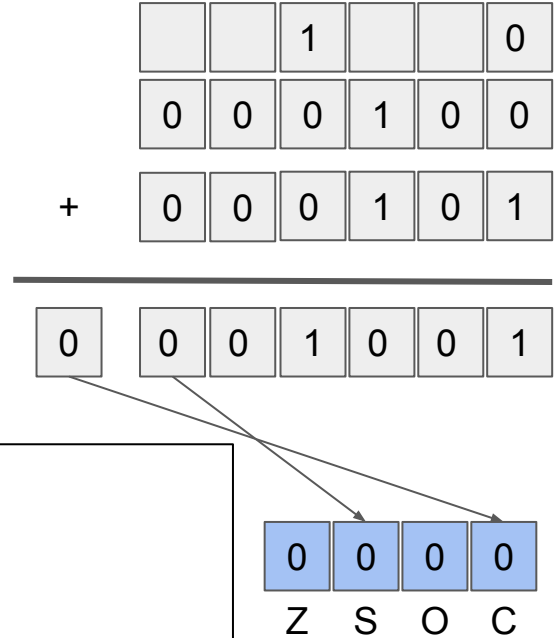
Steps:

1. transform the operation to addition
2. convert  $\text{abs}(X)$  to binary
3. compute 2's complement for negative numbers
4. perform binary addition
5. convert result to decimal
  - if negative result: compute 2's complement first

- 4 + 5:
- 7 - 6:
  - -3 + 5:
  - -18 - 4:

5. 1001 == 9

4 + 5 = 9 (correct)



# Practice: Addition and Subtraction

Steps:

1. transform the operation to addition
2. convert  $\text{abs}(X)$  to binary
3. compute 2's complement for negative numbers
4. perform binary addition
5. convert result to decimal
  - if negative result: compute 2's complement first

- 4 + 5:

➤ 7 + (-6):

- -3 + 5:

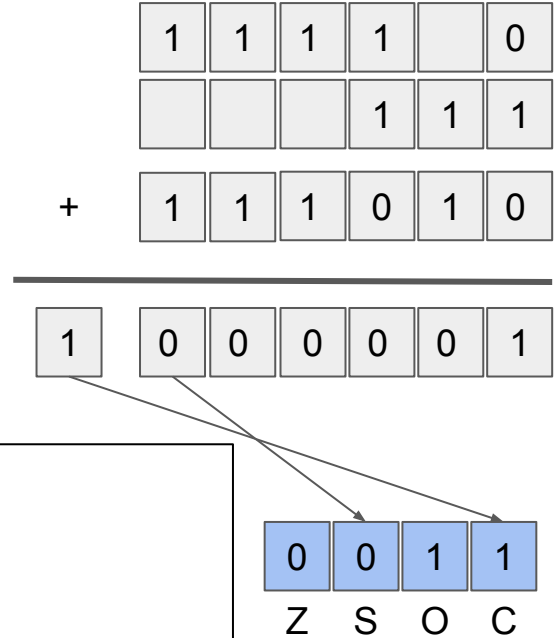
- -18 - 4:

2. 000110 (6)

3. 111001 + 1 == 111010

5. 1

7 + -6 = 1 == 1 (correct)



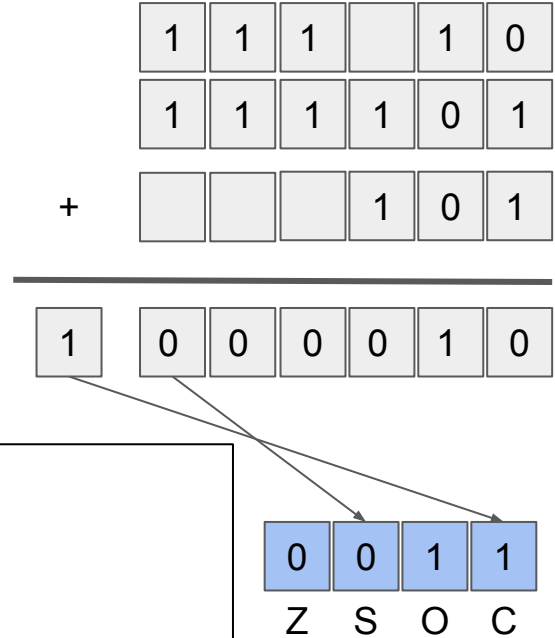
# Practice: Addition and Subtraction

Steps:

1. transform the operation to addition
2. convert  $\text{abs}(X)$  to binary
3. compute 2's complement for negative numbers
4. perform binary addition
5. convert result to decimal
  - if negative result: compute 2's complement first

- 4 + 5:
- 7 + - 6:
- (-3) + 5:
- -18 - 4:

2. 000011  $\text{abs}(3)$   
3. 111100 + 1 = 111101  
5. 2  
-3 + 5 == 2 (correct)



# Practice: Addition and Subtraction

Steps:

1. transform the operation to addition
2. convert  $\text{abs}(X)$  to binary
3. compute 2's complement for negative numbers
4. perform binary addition
5. convert result to decimal
  - if negative result: compute 2's complement first

- $4 + 5$ :
- $7 + -6$ :
- $-3 + 5$ :
- $(-18) + (-4)$ :

2.  $\text{abs}(-18) : 010010$

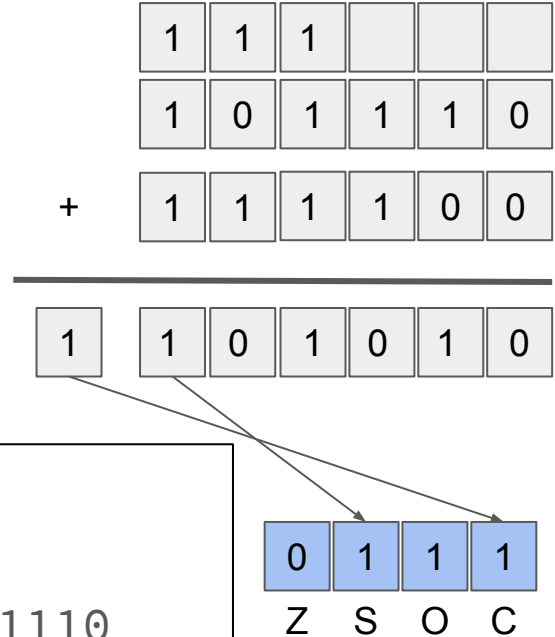
$\text{abs}(-4) : 000100$

3.  $-18 : 101101 + 1 == 101110$

$-4 : 111011 + 1 == 111100$

5.  $101010 \rightarrow 010101 + 1 \rightarrow 010110$

$010110 + 001110 = 1011010$



# BCD: Addition

- Addition performed on the nibble level: 6+7

$$\begin{array}{r} \begin{array}{|c|c|c|c|} \hline 1 & 1 & 0 & 0 \\ \hline \end{array} \\ \begin{array}{|c|c|c|c|} \hline 0 & 1 & 1 & 0 \\ \hline \end{array} \\ + \begin{array}{|c|c|c|c|} \hline 0 & 1 & 1 & 1 \\ \hline \end{array} \\ \hline \begin{array}{|c|c|c|c|c|} \hline 0 & 1 & 1 & 0 & 1 \\ \hline \end{array} \end{array}$$

if (overflow or invalid code ) then

$$\begin{array}{r} + \begin{array}{|c|c|c|c|} \hline 0 & 1 & 1 & 0 \\ \hline \end{array} \\ \hline \begin{array}{|c|c|c|c|c|} \hline 1 & 0 & 0 & 1 & 1 \\ \hline \end{array} \end{array}$$

N	Code	N	Code
0	0000	8	1000
1	0001	9	1001
2	0010		1010
3	0011		1011
4	0100		1100
5	0101	x	1101
6	0110		1110
7	0111		1111





# Multiplication

- Problem:  $109 \times 13 = 1417$
- Approach: Successive Additions
  - Consider:  $9 + 9 + 9 \dots + 9$  (13 times) = ?
  - What is carry value for the 10's column?
- Approach: Long Multiplication

```

      013      (A)
*     109      (B)
-----
      117      (A*9)*1
      0000      (A*0)*10
+ 01300      (A*1)*100
-----
     1417

```

```

      013      (A)
*     109      (B)
-----
      117      (A*9)*1
+   0000      (A*0)*10
-----
      117
   01300      (A*1)*100
-----
      1417

```

Requires (at worst)  $10^N$  additions

The diagram illustrates the multiplication of 11 by 7. At the top, there are two boxes containing the digits 1 and 1, followed by a box containing the digit 7. Below these, there is a large empty box on the left for the result. To the right of this box, the number 11 is written above the number 7. A vertical arrow on the right side of the diagram points downwards, with the label 'x13' next to it. Below the arrow, there is a row of boxes containing the digits 1, 4, 1, and 7. A dashed line is drawn below the boxes containing 1, 4, and 1.

# Algorithm for Multiplication

```
sum = 0;
for (d = 0 ; d < 4 ; d ++ ) {
    sum += A * B[d];
    A = A * 10 ; // Shift to the left
}
```

```
// B[0] = 9
// B[1] = 0
// B[2] = 1
```

original:

	013	(A)
*	109	(B)
-----		
	117	(A*9)*1
+	000	(A*0)*10
-----		
	117	
	01300	(A*1)*100
-----		
	1417	

reframe:

	013	(A)
*	109	(B)
-----		
	117	(A*1) *9
+	000	(A*10) *0
-----		
	117	
	01300	(A*100)*1
-----		
	1417	

# Algorithm for Binary Multiplication

```
sum = 0;
for (d = 0 ; d < 4 ; d ++ ) {
    if (B[d] == 1) {
        sum += A * B[d];
    }
    A = A * 2 ; // Shift to the left
    A << 1 ;
}
```

reframe:

010	(A = 2)
* 101	(B = 5)
-----	
010	(A*1) *1
<del>+ 0000</del>	<del>(A*2) *0</del>
-----	
0010	
01000	(A*4) *1
-----	
01010	(A*B = 10)

original:

010	(A = 2)
* 101	(B = 5)
-----	
010	(A*1)*1
+ 0000	(A*0)*2
-----	
0010	
01000	(A*1)*4
-----	
01010	(A*B = 10)

Requires *word\_size* additions