MINI QUIZ: BINARY CONVERSION OF TWOS COMPLEMENT

CONVERT THE FOLLOWING NUMBERS TO 8-BIT TWO'S COMPLEMENT:

- 127
- -128
- -1
- 1
- -14

CONVERT THE FOLLOWING 8-BIT TWOS COMPLEMENT NUMBERS TO DECIMAL

- 10000011
- 11000100

COMPUTE BINARY ADDITION

```
01111111
10000000
---------------
```

SHORT ANSWER

What is the value of the most significant bit in 8-bit twos complement? What about 32-bit twos complement?

How do you negate a number in Twos Complement?

How can we compute subtraction of twos complement numbers?

It can be beneficial for our hardware to be able to detect overflow in two's complement. To do so, we'd need a rule for determining—based solely on bit patterns—if overflow has occurred. Can you describe such a rule? Consider the following examples:

```
// In this case there is a carry out, but the result is correct
110000000 (carry row)
 11000000 (-64)
 01000000 (64)
--------------
 00000000 (0, but there was a carry out!)
```

```
// In this case the result is incorrect but there is no carry out
010000000 (carry row)
 01000000 (64)
 01000000 (64)
```

```
--------------
 10000000 (-128)
```

```
// In this case there is a carry out and the result is incorrect
100000000 (carry row)
 10000000 (-128)
 10000000 (-128)
----------------
 00000000 (0)
```