Addition / Subtraction (Again)

ICS3U

Mr. Emmell

Try these, in decimal

```
   5
+  3
-----
```

Try these, in decimal

```
   5
+  3
-----
   8
```

Try these, in decimal

```
   8
+  4
-----
```

Try these, in decimal

```
   8
+  4
-----
  12
```

How did you do that?

When one column overflowed….

It incremented the next column

How did you know it overflowed?

When one column…
tried to have a higher value than our number system allowed!

Then you
- Subtracted ten
- Wrote the remainder
- Added the carry

Try these, in octal

$$
\begin{array}{r}
2_8 \\
+\ 4_8 \\
\hline
\end{array}
$$

Try these, in octal

$$
\begin{array}{r} 2_8 \\ +\ 4_8 \\ \hline 6_8 \end{array}
$$

Try these, in octal

$$
\begin{array}{r} 5_8 \\ +\ 5_8 \\ \hline \end{array}
$$

Try these, in octal

$$\begin{array}{r} 5_8 \\ +\ 5_8 \\ \hline \end{array}$$

5 + 5 = 10!!!!
That's too much

This is base 8 so…

10 – 8 = 2

Try these, in octal

$$\begin{array}{r} 5_8 \\ +\ 5_8 \\ \hline 1\,2_8 \end{array}$$

Try these, in octal

$$\begin{array}{r} 1\,5_8 \\ +\quad 5_8 \\ \hline \end{array}$$

Try these, in octal

$$\begin{array}{r} 1\,5_8 \\ +\quad 5_8 \\ \hline 2\,2_8 \end{array}$$

Try these, in octal

$$\begin{array}{r} 3\ 5_8 \\ +\ 4\ 6_8 \\ \hline \end{array}$$

Try these, in octal

$$\begin{array}{r} 3\ 5_8 \\ +\ 4\ 6_8 \\ \hline 1\ 0\ 3_8 \end{array}$$

Hexadecimal works the same way!

$$
\begin{array}{r}
3_{16} \\
+\ 6_{16} \\
\hline
\end{array}
$$

Hexadecimal works the same way!

$$
\begin{array}{r}
3_{16} \\
+\ 6_{16} \\
\hline
9_{16}
\end{array}
$$

Hexadecimal works the same way!

$$
\begin{array}{r}
7_{16} \\
+\ 6_{16} \\
\hline
\end{array}
$$

Hexadecimal works the same way!

$$
\begin{array}{r}
7_{16} \\
+\ 6_{16} \\
\hline
D_{16}
\end{array}
$$

Hexadecimal works the same way!

$$
\begin{array}{r}
B\ 9_{16} \\
+\quad A_{16} \\
\hline
\end{array}
$$

Hexadecimal works the same way!

$$
\begin{array}{r}
B\ 9_{16} \\
+\quad A_{16} \\
\hline
\end{array}
$$

Remember!
In decimal,
this is:
9
+
10
= 19

Hexadecimal works the same way!

$$
\begin{array}{r}
B\ 9_{16} \\
+ \quad A_{16} \\
\hline
C\ 3_{16}
\end{array}
$$

Hexadecimal works the same way!

$$
\begin{array}{r}
2\ 6_{16} \\
+\ 7\ E_{16} \\
\hline
\end{array}
$$

Hexadecimal works the same way!

$$
\begin{array}{r}
26_{16} \\
+\ 7E_{16} \\
\hline
A4_{16}
\end{array}
$$

Don't forget binary too

$$
\begin{array}{r}
1001_{2} \\
+\ 1011_{2} \\
\hline
\end{array}
$$

Don't forget binary too

$$
\begin{array}{r}
1\,0\,0\,1_2 \\
+\ 1\,0\,1\,1_2 \\
\hline
1\,0\,1\,0\,0_2
\end{array}
$$

Now… how about subtraction?

Good news! We can't… Not really.
We need to learn about complements first

So instead, we add the complement.

Consider a 12 hour clock.

If it is 11:00 now, then three hours later it will be 2:00

$$11 + 3 = 14 \equiv 2 \pmod{12}$$

2 o'clock

So instead, we add the complement.

Similarly, if it is 1:00, then 4 hours ago it was 9 since

$$1 - 4 = -3 \equiv 9 \pmod{12}$$

9 o'clock

So instead, we add the complement.

Notice that subtracting 4 hours on the clock is the same as *adding* 8 hours (12 – 4).

In particular, we could have computed it as follows:

$$1 - 4 \equiv 1 + 8 = 9 \pmod{12}$$

Binary numbers work the same way

Use an 8-bit number as an example

Those numbers range from 0-255

If we have 255 and add one,
then we get zero because it 'cycles back'.

So....?

How do we find the complement for a number in binary?

We know that 200 – 50 is the same as 200 + (255-50)
(Because we roll over after 255!)

HOW TO FIND "TWO's COMPLEMENT"

- Define how many bits you are using! 8-bit?
  - Always need to have leading zeroes.
  - Both numbers must have the same number of bits
- Invert all the bits
- Add one
- BOOM – Two's complement. Add normally using binary addition.

Example

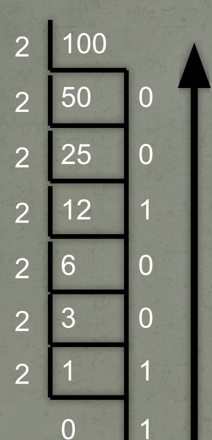
100 – 58 = 42

Now for this operation, we are really saying:

100 + (-58) = 42

Let's convert to binary

Example



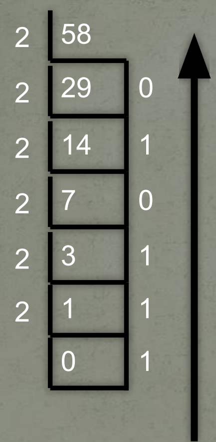
$100_{10} = 1100100_2$

$100_{10} = 01100100_2$

Remember! 8-bits

What about -58?

Example – Start with 58



$58 = 00111010_2$

Remember! 8-bits

What about -58?

Example

00111010_2	//58
11000101_2	//FLIP
+1	//Add one
11000110	

$-58 = 11000110_2$

Now we do the 'addition'

$$\begin{array}{r} 0\,1\,1\,0\,0\,1\,0\,0_2 \\ \underline{1\,1\,0\,0\,0\,1\,1\,0}_2 \end{array}$$

Now we do the 'addition'

$$\begin{array}{r} 0\,1\,1\,0\,0\,1\,0\,0_2 \\ \underline{1\,1\,0\,0\,0\,1\,1\,0}_2 \\ 1\,0\,0\,1\,0\,1\,0\,1\,0_2 \end{array}$$

If there is a leading one, we drop it (remember! 8-bits)

Answer = 00101010_2

(42 !)

What do we mean by “padding”?

Examples:

$101_2 \Rightarrow 00101_2$

$1101_2 \Rightarrow 01101_2$

$1110_2 \Rightarrow 01110_2$

$100_2 \Rightarrow 00100_2$

$10000_2 \Rightarrow 010000_2$

$110_2 \Rightarrow 000110_2$

$1100_2 \Rightarrow 01100_2$

$1000_2 \Rightarrow 01000_2$

Note:

Rather than having our numbers go from 0 to 255 around a clock, we have them go from −128 to 127

NOTE: Use 8-bit representations of every number below

Complete the following in binary

a) $00001110_2 + 00100011_2$ b) $75_{10} + 26_{10}$

Using two's complement

a) $37_{10} - 37_{10}$ b) $55_{10} - 18_{10}$ c) $85_{10} - 65_{10}$

ex: $37_{10} + (-37_{10})$

Using two's complement

a) $-14_{10} + 28_{10}$ b) $20_{10} - 36_{10}$ c) $15_{10} - 50_{10}$