Signed Magnitude Notation

Wednesday, April 7, 2021 1:54 PM

Unsigned Binary - what you worked with before (ex 0010 as 2)

Signed Magnitude Notation - leftmost bit represents the sign (+/−); other bits represent the magnitude

Ex: Represent +5 using signed magnitude notation (using 4-bits)

sign	Magnitude		
0	1	0	1

↑

+ is 0

Ex: Represent −5 using s.m.n.

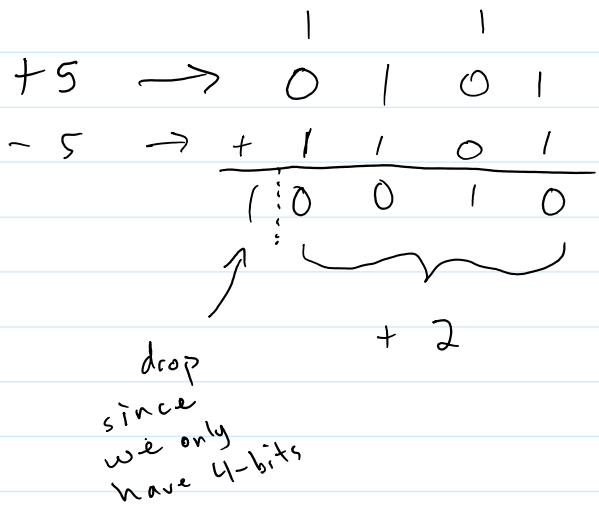
sign	Magnitude		
1	1	0	1

−5 is 1101

Ex: Represent 0 using signed magnitude notation.

sign	magnitude			
0	0	0	0	← +0 } two zeros; not ideal
1	0	0	0	← −0 }

Ex: Add −5 and +5 using S.M.N.



$\Rightarrow 5 + (-5) = 2$

Not good for simple arithmetic

One's Complement Notation

Wednesday, April 7, 2021 2:19 PM

Def: The complement of a binary number is formed by reversing every bit.

Ex: Complement the binary value 0100.

Sol: 1011

Ex: Represent +5 using one's complement notation.

Sol: 0101

↑ sign

↳ magnitude

Ex: Represent −5 using 0.C.N.

+5 −5

Sol: 0101 → (complement) → 1010

↑ sign ↑ magnitude

Ex: Represent +2 and −2 using 0.N.C

+2 −2

Sol: 0010 → (complement) → 1101

Ex: Represent 0.

Sol: 0000 +0

1111 −0

} still have two zeros

Ex: Add +5 and −5.

Sol:

$$\begin{array}{rr} +5 & 0\,1\,0\,1 \\ -5 & +\;1\,0\,1\,0 \\ \hline & 1\,1\,1\,1 \end{array} \longrightarrow -0_{10}$$

Try this: Add +5 and −2

Sol: +5 is 0101 +2 is 0010

−2 is 1101

$$\begin{array}{rr} & 1\;\;1 \\ +5 \longrightarrow & 0\,1\,0\,1 \\ +\;-2 \longrightarrow & +\;1\,1\,0\,1 \\ \hline & 1\,0\,0\,1\,0 \end{array}$$

drop ↑

1 0010 → sign mag } +2

Two's Complement Notation

Wednesday, April 7, 2021 2:35 PM

Positive values are the same as unsigned binary.

To negate values:

Step 1: Start with given value in binary

Step 2: Complement every bit.

Step 3: Add 1

Ex: Write −5 using two's complement

1.) +5 is 0101

2.) Complement 1010

3.) Add 1

$$\begin{array}{r} 1010 \\ +\ 0001 \\ \hline 1011 \end{array}$$

1011 is −5 using T.C.

Try this: Write −2 using two's complement

Sol: 1.) +2 is 0010

2.) Complement 1101

3.) Add 1 1110

$$\begin{array}{r} 1101 \\ +\ 0001 \end{array}$$

Try this: Convert −> (1001) to +7

Sol: 1.) -7 is 1001

2.) Complement 0110

3.) Add 1 0111 $\longrightarrow$ $+7$ is 0111

Ex: Find -0 from $+0$.

Sol: 1.) $+0$ is 0000

2.) complement 1111

3.) Add 1 0000 $\Leftarrow$

$$\begin{array}{r} {\scriptstyle 1\,1\,1} \\ 1111 \\ +\ 0001 \\ \hline 10000 \end{array}$$

drop

We only have 1 zero!!

Ex: Add $+5$ and -5

Sol: $+5$ is 0101

-5 is 1011 $\Rightarrow$

$$\begin{array}{r} {\scriptstyle 1\,1\,1} \\ 0101 \\ +\ 1011 \\ \hline 10000 \end{array}$$

drop 0_{10}

Ex: Add $+5$ and -2

Sol: $+5$ is 0101

-2 is 1110 $\Rightarrow$

$$\begin{array}{r} {\scriptstyle 1} \\ 0101 \\ +\ 1110 \\ \hline 10011 \end{array}$$

drop +3

Arithmetic works easily!!

We've worked with 4-bits of storage:

Min value is $-2^{4-1} = -2^3 = -8$

Max value is $2^{4-1} - 1 = 2^3 - 1 = 8 - 1 = 7$

In general with n-bits.

Min value is -2^{n-1}

Max value is $2^{n-1} - 1$

Overflow

Wednesday, April 7, 2021 3:00 PM

Def: Overflow – occurs when the value that is to be stored is outside the range of permissible values.

Handle by adding more bits or just detecting and reporting

Detecting Overflow:

Ex: Add +4 and +5

Sol:

```
      1
+5 is  0 1 0 1
+4 is +0 1 0 0
      1 0 0 1   → −7
```

Ex: Add −4 and −5

```
−4 is  1 1 0 0
−5 is +1 0 1 1
      1 0 1 1 1
        ↑ ~~~
      drop +7
```

sign is opposite the two original signs.