

2進数の加算

- 2進数の加算は10進数の加算と変わらない
- 加算の結果が2になると桁上げ(繰り上がり、carry)が起こり、その桁は0になる

$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array} \quad \begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ + 0 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ + 1 \\ \hline 10 \end{array}$$

2進数	10進数
00010101 ...	21
+ 00000101 ...	5
<hr/>	
00011010 ...	26

2進数	10進数
00010111 ...	23
+ 00001001 ...	9
<hr/>	
? ... ?	

← Question

2進数の減算・負の数の表現

- これまで、正の数しか考えてこなかった。では、負の数はどう表現するか
- また、手計算であれば、2進数の減算も10進数の減算と同じやり方も考えられるが、コンピュータの世界では、減算問題を加算問題に変えて行うのが一般的
- つまり、10進数でも減算問題の $5-3$ を加算問題の $5+(-3)$ に見方を変えると同様、2進数の減算 $(X)_2 - (Y)_2$ も $(X)_2 + \text{負数の}(Y)_2$ に変えることが可能
- したがって、ポイントは、この**負数の $(Y)_2$** ... つまり、2進数の負数をどう表現するかである

2進数の減算・負の数の表現

- 一般的に、正数Aに対して、 $-A$ とは $A+B=0$ を満たす数Bである(これはまさに負数の定義)
- 2進数 $(A)_2$ に対しても、 $(A)_2+(B)_2=(0)_2$ を満たす2進数 $(B)_2$ は、負の $(A)_2$ を表現していることになる
- 実際、この $(0)_2$ は桁数が固定(決まった桁数のみ使用)の場合の0である
 - たとえば固定の桁数が4であれば、桁あふれが生じて下位四ビットが $0(10000)_2$ のことを指す
- 注意: コンピュータの世界では手計算と違って、必ず有効桁数というものがあるので、上記考えで負数表現が実現できる

負の数の表現

- ではこのような $(B)_2$ はどうやって、 $(A)_2$ からもとめるか？
- 固定の桁数が8で $(A)_2=00000101$ とする

A: 00000101
 ↓ 各ビットを反転
C: 11111010
 ↓ A+C
A+C: 11111111
 ↓ 1を足す
A+C+1: 100000000
 ↓
 B



$(A)_2$ の各ビットを反転させ、それに1を足すことで $(B)_2$ が得られる。

この $(B)_2$ を $(A)_2$ の2の補数という。

ちなみに、 $(C)_2$ を $(A)_2$ の1の補数ともいう

演習02-1

2進数の負数を求める問題

manaba: 3分

負の数の表現

- 以上をまとめると、
 - ① 負の数は「2の補数」でもとめることができる
 - ② 最上位ビットが0か1で正数・負数を表現できそう

8ビットの場合の正数・負数の2進数表現(8ビット符号付き2進数)

10進数	2進数(2の補数による負数の表現)
127	01111111
126	01111110
⋮	⋮
2	0000010
1	0000001
0	0000000
-1	11111111
-2	11111110
⋮	⋮
-127	10000001
-128	10000000

$2^8=256$ 通り

符号ビット↑

演習02-2

- 前の表の通りで数を8ビットの符号付き2進数で表現するとした場合、正負の2進数の10進数をもとめる問題
- manaba:5分

補足

- 0は $(00000000)_2$ なので、これの2の補数(補数+1)も同じく $(00000000)_2$ である
- つまり、このような負数の求め方は、0についても不備はない

8ビットで符号なし2進数表現

10進数	2進数
255	11111111
254	11111110
⋮	⋮
129	10000001
128	10000000
127	01111111
126	01111110
⋮	⋮
1	00000001
0	00000000

$2^8=256$ 通り

2進数の減算

- 2の補数への加算に変換
- $42-17=42+(-17)$

$$\begin{array}{r} 00101010 \dots 42 \\ + 11101111 \dots (-17) \\ \hline 100011001 \dots 25 \end{array}$$

←17の2の補数で求める

あふれが出た。これを無視

- $12-17=12+(-17)$

$$\begin{array}{r} 00001100 \dots 12 \\ + 11101111 \dots (-17) \\ \hline 11111011 \dots -5 \end{array}$$

演習02-3

- 2進数の減算問題
- manaba: 8分

演習 (オーバーフローについて)

1. 120を8ビット符号付きまたは符号なしの2進数に変換すると01111000となり、30は00011110となる。
 - a. これらを符号付き2進数とみなしその和を8ビット固定で計算して2進数と10進数で答えなさい
 - b. これらを符号なし2進数とみなしその和を8ビット固定で計算して2進数と10進数で答えなさい
2. -120と-30を8ビット符号付き2進数に変換し、その和を8ビット固定で計算して2進数と10進数で答えなさい

オーバーフロー

- 固定の桁数(今の場合8)で表した数同士の演算の結果が、その桁数で表せる範囲を外れてしまうことを、オーバーフローという
- 8ビット符号付きの値の範囲:-128~127
- 8ビット符号なしの値の範囲:0~255