

4コマ目 ビット演算



10進数, 2進数, 16進数について



10進数

- 0～9までの**10個の数字を使用して数字を扱う方法**
- 0～9まで数えた後，桁が上がりをもと表現する
 - 10進数では10，100，1000と桁が上がる
- 各桁を10のN乗と表現できる

$$\begin{aligned}
 3290 &= 3 \times 10^3 + 2 \times 10^2 + 9 \times 10^1 + 0 \times 10^0 \\
 &= 3 \times 1000 + 2 \times 100 + 9 \times 10 + 0 \times 1 \\
 &= 3000 + 200 + 90 + 0
 \end{aligned}$$

10進数
0
1
2
3
4
5
6
7
8
9
10 (桁上がり)
11
12

2進数


- 0, 1の**2個の数字を使用して数字を扱う方法**
 - **コンピュータで使用されている**
- 1まで数えた後、桁が上がり方を10と表現する
 - 2進数でも10, 100, 1000と桁が上がる
- 10進数の2は10, 3は11, 4は100と表現できる
 - 2進数の10を $10_{(2)}$ と表現することもある
 - N進数のMを $M_{(N)}$ と表現する
- 各桁を2のN乗と表現できる

$$\begin{aligned}
 1010_{(2)} &= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 \\
 &= 1 \times 8 + 0 \times 4 + 1 \times 2 + 0 \times 1 \\
 &= 10
 \end{aligned}$$

10進数	2進数
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100

10進数から2進数への変換

- 10進数の数を2で割って余りを出しながら，商が0になるまで割り，その余りを最後から順に並べると2進数に変換できる
- 例) $8_{(10)}$

$$\begin{array}{rcl}
 8 \div 2 = 4 & \text{余り} & 0 \\
 4 \div 2 = 2 & \text{余り} & 0 \\
 2 \div 2 = 1 & \text{余り} & 0 \\
 1 \div 2 = 0 & \text{余り} & 1
 \end{array}$$


- ゆえに， $8_{(10)} = 1000_{(2)}$

16進数

- 0～9, A, B, C, D, E, Fの
16個の文字を使用して数字を扱う方法
 - コンピュータで使用されている
 - $10_{(10)} = A_{(16)}$, $11_{(10)} = B_{(16)}$, $12_{(10)} = C_{(16)}$,
 $13_{(10)} = D_{(16)}$, $14_{(10)} = E_{(16)}$, $15_{(10)} = F_{(16)}$
- 各桁を16のN乗と表現できる

$$CDA_{(16)} = 12 \times 16^2 + 13 \times 16^1 + 10 \times 16^0$$

$$= 12 \times 256 + 13 \times 16 + 10 \times 1 = 3290_{(10)}$$

10進数	2進数	16進数
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
16	10000	10

10進数から16進数への変換

- 10進数の数を16で割って余りを出しながら，商が0になるまで割り，その余りを最後から順に並べると16進数に変換できる
- 例) $430_{(10)}$

$430 \div 16 = 26$	余り	14 $\rightarrow E_{(16)}$	
$26 \div 16 = 1$	余り	10 $\rightarrow A_{(16)}$	
$1 \div 16 = 0$	余り	1	

- ゆえに， $430_{(10)} = 1AE_{(16)}$

2進数と16進数の変換

- 2進数を16進数へ変換：下位4ビットごとを16進数に変換する
- 例) $106_{(10)} = 1101010_{(2)}$
 1101010 を0110と1010に分割する($110 \mid 1010 \rightarrow 0110 \mid 1010$)
 ※下位4ビットごとに分ける，桁が足りない場合は0を入れる
 $0110_{(2)} = 6_{(16)}$ ， $1010_{(2)} = A_{(16)}$ となるため， $1101010_{(2)} = 6A_{(16)}$
- 16進数を2進数へ変換：上記の逆順で変換する
 - 各桁を2進数に変数して繋げれば良い

ビット演算について



ビットとは

- コンピュータのデータの最小の単位
- **ON = 1 ・ OFF = 0とし，2進数を表現する**
 - フラグ管理，状態管理，I/Oなどに使用される
- **通常1桁だけでなく8bit(8桁)で表し，8bitを1byteと呼ぶ**
- 例)0b10101100 (先頭に0bを付けることで，2進数を表す)
 - 右から0bit目，1bit目，2bit目，・・・，6bit目，7bit目と呼ぶ

ビットとは

- 各桁ごとに意味を持たせて使用することが多い
- 例) モータ制御時：0b10101100
 - 3bit～8bit目(101011)：モータの回転数
 - 1bit, 2bit目 (00)：回転の向き
- 例) LED：0b00001101
 - 下位4bit(1101)を見ることで，
1個目，3個目，4個目のLEDが点灯していることがわかる

ビット演算について

- ビット演算：10進数ではなくビット（2進数）で計算
 - 利点：計算速度が速い，2進数のまま計算が出来る
- 演算子など
 - AND
 - OR
 - \gg （右シフト）
 - \ll （左シフト）

AND

- 2つのビットを比較して両方が1なら1を，それ以外は0を返す

評価式	演算結果
1 and 1	1
1 and 0	0
0 and 1	0
0 and 0	0

- **&**とも表記される

ANDの使用例

- 例) 0F & A8

00001111 (0F)

AND 10101000 (A8)

00001000 = 08

- 両方が1でない場合に0になることを利用して，特定のビットの値を知りたいときなどに使用する

- 例)

0b101011**01**

AND 0b000000**11**

0b000000**01**

Nビット目の値を知りたい場合，

Nビット目の値だけが1である（それ以外のビットは0）

2進数とANDを行えば良い

OR

- 2つのビットを比較して少なくとも一方が1なら1を，両方が0なら0を返す

評価式	演算結果
1 or 1	1
1 or 0	1
0 or 1	1
0 or 0	0

- |とも表記される

ORの使用例

- 例) 0F | A8

00001111 (0F)

OR 10101000 (A8)

10101111 = AF = 175₍₁₀₎

- 特定のビットの値を1にしたいときに使用する

- 例) 下位3bit目を1したい場合

0b11010011

OR 0b00000**1**00

0b11010**1**11

**Nビット目の値を1にしたい場合,
Nビット目の値だけが1である（それ以外のビットは0）
2進数とORを行えば良い**

<< (左シフト)

- ビットを左に数字分ずらす
- 例) `0b00010001 << 1`
 - 1つ左にずれるため, `0b00100010` となる
 - はみ出た0は消え, 空いた枠には0が入る
- 1を別の数字に変えるとその分だけずれる
- 例) `0b00010001 << 4`
 - 4つ左にずれるため, `0b100010000` となる

>> (右シフト)

- ビットを右に数字分ずらす
- 例) `0b00010001 >> 1`
 - 1つ右にずれるため, `0b00001000` となる
 - はみ出た部分は消え, 空いた枠には0が入る
- 1を別の数字に変えるとその分だけずれる
- 例) `0b00010001 >> 4`
 - 4つ右にずれるため, `0b00000001` となる