

# ビットシフトと回転シフト

# ビットシフト

## ① ビットシフト

ビット列を右または左に「ずらす」ことである。

② 右に1ビットずらすことを  
“右に1ビットシフトする”という。

③ 左に1ビットずらすことを  
“左に1ビットシフトする”という。

④ 右や左にシフトアウトしたビットは捨てる。

# 論理シフト

- ① 符号を含まないビット列のシフトを論理シフトと言う。
- ② 右にシフトして空欄になった左の桁を0で補う
- ③ 左にシフトして空欄になった右の桁を0で補う。
- ④ シフトアウトしたビットは捨てる。

1ビット右にシフトする

0	1	0	1	1	1	0	1	
↘	↘	↘	↘	↘	↘	↘	↘	
0	0	1	0	1	1	1	0	1

空いたところには0を埋める

# 具体例1

- ① 「右へ3ビット論理シフトする」場合、次のようになる。

01101010 → 00001101  
0で補う

- ② 空欄になった左の3桁を0で補う。
- ③ シフトの結果 010 の3ビットが右にシフトアウトする。
- ④ シフトアウトした010の3ビットは捨てる。

# 具体例2

- ① 「左へ3ビット論理シフトする」場合、次のようになる。

01101010 → 01010000  
0で補う

- ② 空欄になった右側の3桁を0で補う。
- ③ シフトの結果 011 の3ビットが左にシフトアウトする。
- ④ シフトアウトした011の3ビットは捨てる。

# 算術シフト

- ① 算術シフトは  
最上位のビット(MSB)を符号と考える場合である。
- ② 符号ビットはシフトの対象外とする。
- ③ 右にシフトする場合も、左にシフトする場合も、  
符号を除いたビットがシフトの対象になる。

#### ④ 右に算術シフトする場合

- ① 符号の後の空になった所には、  
符号と同じビットが入る。
- ② 正の場合は0、負の場合は1 が入る。

#### ⑤ 左に算術シフトする場合

- ① 符号はシフトの対象外である。  
符号の位置は変わらない。
- ② 下位の空いた桁には0が入る。

# 具体例1④の場合

8ビットの2進数を「3ビット右へ算術シフトする」場合

01101010 → 00001101

符号はシフトの対象外

0で補う

10101100 → 11110101

符号はシフトの対象外

1で補う

# 具体例2⑤の場合

8ビットの2進数を「3ビット左へ算術シフトする」場合

01101010 → 01010000



0で補う

11111100 → 11100000



0で補う



# 回転シフト

① 回転シフトはシフトアウトされたビットが  
反対側から入ってくる場合である。

① 「右に4ビット回転シフトする」場合  
01101010 → 10100110

② 「左に4ビット回転シフトする」場合  
01101010 → 10100110

② この場合、右回転シフトも左回転シフトも  
結果は同じになる。

# 具体例1

「16進数の6A93を右に4ビット回転シフトする」場合

6A93 → 0110101010010011

0011011010101001 → 36A9

# 具体例2

「16進数の6A93を右に6ビット回転シフトする」場合

6A93 → 0110101010010011

0100110110101010 → 4DAA

# 具体例3

「16進数の6A93を左に6ビット回転シフトする」場合

6A93 → 0110101010010011

1010010011011010 → A4DA

# 16進数の値の回転シフト

① 16進数を2進数に変換する。

② 2進数の状態で回転シフトする。

(16進数で回転シフトすると誤りを起こしやすい)

③ 回転シフトした結果を16進数に変換する。