

## グレイコード

グレイコード (Gray code) とは、2進数の表現で、隣り合う値でビットの変化が1ビットだけになるようにしたコード (符号) をいう。

### 2 ビット

10 進数	通常の 2 進数		グレイコード	
0	0	0	0	0
1	0	1	0	1
2	1	0	1	1
3	1	1	1	0

### 4 ビット

10 進数	通常の 2 進数				グレイコード			
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

### 3 ビット

10 進数	通常の 2 進数			グレイコード		
0	0	0	0	0	0	0
1	0	0	1	0	0	1
2	0	1	0	0	1	1
3	0	1	1	0	1	0
4	1	0	0	1	1	0
5	1	0	1	1	1	1
6	1	1	0	1	0	1
7	1	1	1	1	0	0

### <長所>

グレイコードは、ある値から隣の値に変化する際に、常に1ビットしか変化しないので、誤差を減らすことができる。

例として4桁の2進数で考える。10進数の7から8に変化する場合に通常の2進数では「0111」から「1000」となる。つまり4桁すべてのビットが0から1または、1から0へと変化する。

一方、電子回路で0から1に変化するのに要する時間 (遅延時間) と、1から0へと変化するのに要する時間は異なる。したがって10進数の7から8に変化する場合、通常の2進数表示では入力回路の遅延によって

0111⇒1111⇒1000 となるケース1 (0→1の変化が速い場合) と

0111⇒0000⇒1000 となるケース2 (1→0の変化が速い場合) があり得る。

ケース1を10進数で表すと7⇒15⇒8

ケース2を10進数で表すと7⇒0⇒8

となり、どちらも7から8に変化する間に大きく値が変わってしまうことになる。

一方、グレイコードの場合は、隣り合う数値で、ビットの変化が1ビットしかないので、回路で遅延時間があっても大きな誤差を生じない。

#### <通常の2進数表示からグレイコードへの変換方法>

- ① 2進数表示のMSBを、そのままグレイコードのMSBとする。
- ② 2進数表示のMSBと隣接するビットを加算してその和を求め、グレイコードのビットとする。このとき、キャリー（桁上げ）は無視する。
- ③ 同様の手順を2進数表示のLSBまで続ける。

#### <グレイコードから通常の2進数表示への変換方法>

- ① グレイコードのMSBを、そのまま2進数表示のMSBとする。
- ② 2進数表示のMSBとグレイコードMSBの隣接するビットを加算してその和を求め、2進数表示のビットとする。このとき、キャリー（桁上げ）は無視する。
- ③ 同様の手順をグレイコードのLSBまで続ける。

### 2 out of 5 コード (5者択2符号 ごしゃたくにふごう)

#### <特徴>

5ビットで0から9の10進数を表す。5ビットで構成されるうち2ビットを「1」、残りの3ビットを「0」とする。「1」の個数が常に2個か否かを検査することによって誤りを検出することができる。つまり「1」の個数が2個でないときは誤りが生じたとわかる。

以下の表で左から0-1-2-3-6の重みを付けている。ただし0は01100としてコード化している。

#### 重みづけ

0 1 2 3 6

10 進数	5 者 択 2				
0	0	1	1	0	0
1	1	1	0	0	0
2	1	0	1	0	0
3	1	0	0	1	0
4	0	1	0	1	0
5	0	0	1	1	0
6	1	0	0	0	1
7	0	1	0	0	1
8	0	0	1	0	1
9	0	0	0	1	1