ここから始めるアナ・ディジ混載設計

デバイスの記事

第4回 A-Dコンバータの基本構成と仕様項目

湯川 彰

今回からA-Dコンバータの設計について解説する．D-Aコンバータの場合と同様，A-Dコンバータにも多くの実現方式があり，使用目的に合わせて選択する必要がある．ここでは，A-Dコンバータの基本構成と仕様項目について説明する．また，後半ではA-Dコンバータの精度の問題についても議論する．入力容量などの影響を考慮して誤差を正確に見積もる必要がある．（編集部）

A-Dコンバータは，身近なところではテレビの映像処理，携帯電話の音声入力部と受信復調器など，アナログ信号を受ける部分にかならずと言っていいほど使われています．制御用途では，例えば最近の自動車には数十個のマイコンが使われていますが，そのほとんどがA-Dコンバータを搭載しています．しかし，ひと口にA-Dコンバータと言っても，変換する信号の性質や使う用途は異なり，要求される特性もさまざまです．D-Aコンバータの場合と同じく，A-Dコンバータにも万能のものは存在しません．そこで，使用目的にあったものを選び，選んだものの弱点を把握したうえで設計を進める必要があります．

MOS（metal oxide semiconductor）回路，とくにCMOS（complementary metal oxide semiconductor）回路は昔からの一般的な電子回路のイメージとはかなり異なったふるまいをします．論理回路でも，真空管やバイポーラ回路は定常的な電流が流れ，スイッチングに伴う電流分は定常電流より小さいというイメージがありました．ですから，消費電流の仕様を見たとき，この電流が流れ続けるものだという感覚になりがちです．これに対してCMOS論理回路はスイッチングするときだけドカンと電流が流れ，状態が変化しないときはほとんど電流が流れません．この過渡電流の見積もりが最近のLSIでは大きな問題になっています．

アナログ回路，とくにA-Dコンバータの入力部分でも似たような問題があります．スイッチト・キャパシタで信号をサンプリングするとき，瞬間的に電流が流れ，平衡状態に達すると流れなくなってしまうのです．このことは変換精度に大きな影響を与えるため，A-Dコンバータを使用する周辺回路を設計する際には注意が必要です．

そこで今回は，A-Dコンバータの出力形式と基本的な回路構成，仕様項目，A-Dコンバータを使用するうえで精度を維持するために配慮しなければならない入力インピーダンスの問題などについて解説します．

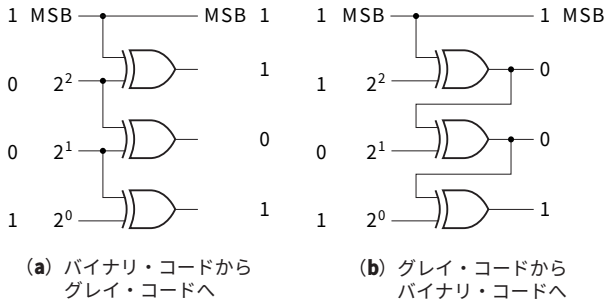
● 3種類のディジタル・コードを使い分ける

2進コードは，コンピュータの高級言語では（正負の）符号なし整数（unsigned integer），符号付き整数（signed integer）に用いる2の補数（2's complement）表現，そして浮動小数点（floating point）の3種類で，あとは語長の違いがあるだけです．一方，A-Dコンバータや D-Aコンバータでは，ほかにいくつかの特徴的な表現方法があります．これを理解していないと，接続したときに「データがおかしい．接続は正しいはずなのに…」ということになってしまいます．また，A-D/D-Aコンバータの設計では，特殊なコードの利点を利用して動作の安定性を向上させることもあります．そこで，主要なコードとその特徴について説明します．

まず，正の値だけを対象にしたコードでもっともよく使われているのが，符号なしの整数表現である（ストレート）バイナリ・コード（straight binary code）です．通常はもっとも低い電圧を‘0’とし，2進数で表現します．積分型や逐次比較型のA-Dコンバータによる変換では，結果が直接このコードになるようにすると，すっきりした設計にな

表1
グレイ・コードとバイナリ・コードの比較
グレイ・コードでは，隣り合うコードの間の違いはどれか1ビットだけである．このコード間の演算は実行できない．

数	グレイ・コード	バイナリ・コード
0	0000	0000
1	0001	0001
2	0011	0010
3	0010	0011
4	0110	0100
5	0111	0101
6	0101	0110
7	0100	0111
8	1100	1000
9	1101	1001
10	1111	1010
11	1110	1011
12	1010	1100
13	1011	1101
14	1001	1110
15	1000	1111



（**a**）バイナリ・コードからグレイ・コードへ　（**b**）グレイ・コードからバイナリ・コードへ

図1　コード変換回路
バイナリ・コードからグレイ・コードへの変換の際の遅延は排他的論理和1段ですむ．グレイ・コードからバイナリ・コードへの変換では排他的論理和によってキャリが伝播することに注意する必要がある．

ります．制御用などの中速および高速のA-Dコンバータではこれをインターフェース・コードに用いるのが普通です．

使う用途は限られますが，グレイ・コード（Gray code）も重要です．**表1**に示す4ビットの対照表にあるように，グレイ・コードも‘0’から始まり，2進数と同じような形をとりますが，隣り合うコードがどれか1ビットだけ異なるというコード体系を持っています．ですから，ビットの位取りがかならずしも重みに対応していません．

バスを介してデータを高速に送る場合，受信点の波形ひずみやビットごとの伝播遅延の差が原因で受信タイミングがずれ，データ誤りを生じることがあります．とくに，MSB（most significant bit）だけが‘0’でほかが‘1’のコードからMSBだけが‘1’のコードに変化したとき，あるいはその逆のとき，伝送路の状態変化が最大になります．駆動側でも負荷条件が最大となるので，駆動電源電圧が低下してデータ誤りを生じやすくなります．このときエラーを起こすと，本来1 LSB（least significant bit）しか違わないのに大きな誤差となり，信号対ノイズ比（S/N）が大きく劣化します．

グレイ・コードを用いれば，隣り合うコードは1ビットだけしか変化しないので，微小変化のときに大きな誤りが発生することを防げます．また，詳細は並列型A-Dコンバータのところで述べますが，グレイ・コードは並列型A-Dコンバータのコード変換にも用いられます．バイナリ・コードからグレイ・コードへの変換回路は**図1（a）**のようになります．また，グレイ・コードからバイナリ・コードへの変換は**図1（b）**のようになります．とくに，**図1（b）**の場合には，EXOR（排他的論理和）回路の信号伝播を伴います．用途がもともと高速なA-D変換なので，コード変換に際して論理遅延を考慮したタイミング設計が必要です．

もう一つ，最近はほとんど使われませんが，1けた当たり4ビットを用いて10進で出力するBCD（binary coded decimal）コードがあります．発光ダイオードなどを使った表示デバイスでは，10進1けたが1素子として市販されています．これを使って表示する装置では，10進出力があると直結できるので便利です．BCDコードはこの要求にこたえるために設定された符号です．しかし，このコード体系では，4ビットで表現できる値が本来16値であるのに対して10値しか使いません．効率が悪く，配線数が増えるため，最近ではほとんど使われていません．

正負両方の値を表現する場合，符号の扱いによっていくつかの表現があります．まず，負の最小値を‘0’としてバイナリ表現する方法があります．これはオフセット・バイナリ（offset binary code）と呼ばれます．ちょうどMSBだけが‘1’のとき，入力を‘0’とするのが普通です．変換範囲の基準となる電圧を1個のD-A変換回路を用いてA-Dコンバータを構成したとき，もっとも簡単に作れるのがこのコードです．

2の補数表現は加減算を加算器だけで実現できるため，ディジタル信号処理を行うときの標準的な表記法になっています．最上位ビットが符号ビットです．正の値は符号ビットが‘0’で，ストレート・バイナリの場合と同じです．負の値は符号ビットが‘1’で，絶対値が等しい正の値を足し合わせたとき，最上位ビットから上位に対してキャリが発生し，全ビットがちょうど‘0’になる数字になります．正の値と負