

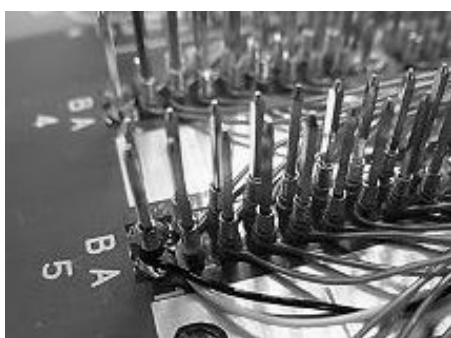
物理屋のための電子回路論 第 13 回

勝本信吾

東京大学理学部・理学系研究科 (物性研究所)

2017 年 1 月 8 日

7.3 論理ゲートの実装



以上簡単な例で見たように、IC へのカプセル化によって回路図や配線は劇的に簡単になり論理 IC のピン間を回路図通りにつなげば良い、と考えられていた時期があった。このため、自作のデジタル回路では、左の写真のように長いピンを持つ IC ソケットに硬めの配線を巻き付けて結線するワイラッピング配線法が盛んに使用されていたこともあり、現在でもシミュレータで十分に拾いきれない問題を調べるため、テスト回路などでは使われている。しかし、すでに述べたように、高いクロック周波数での動作、また前回述べた EMC の観点などから、このような配線法はあまり行われなくなり、自作やテストも FPGA を用いることが多くなった。

以上は余談であり、本節の目的はこのように「カプセル化」されているデジタル IC 論理ゲートのアナログの自身について簡単に見ておくことである。これは、カプセルブロック化した IC の様々な物理的規定がどのような原因で生じているかを理解するためである。図 7.9 に論理ゲートの回路構成法の代表である、TTL (transistor-transistor logic) と CMOS (complimentary metal-oxide-semiconductor) による nand ゲートの構成例 (等価回路) を示した。いずれも 4 つのトランジスタまたは FET で構成されているが、その仕様のされ方はかなり異なっている。

(a) の TTL 構成では、入力にダブルエミッタトランジスタを用いている。これは挿入図にも示したように 2 つのトランジスタが並列になっているものとして動作を理解することでができる。従って、 A_{in} または B_{in} が 0 (L, GND) であれば、トランジスタ Q_1 が ON になって電流が流れ、 Q_2 を ON にする。 Q_3 と Q_4 はプッシュプル回路を構成しており、 Q_2 が ON の時は Q_3 が ON で Q_4 が OFF、 Q_2 が OFF の時は Q_3 が OFF で Q_4 が ON になるように回路定数が決められている。以上から、入力が両方 H になった時のみ Q_1 が OFF になって出力が GND 側に振れ、nand 動作をすることがわかる。この回路から、TTL では入力レベルとして初段のトランジスタを ON/OFF することが求められること、出力は V_{CC} から GND まで振れるわけではなく、出力段のプッシュプルトランジスタによる電圧ドロップを考慮しなければならず、また、出力の負荷抵抗によって出力電圧が変化することも理解される。

これに対して、(b) の CMOS 構成は非常に簡潔である。complimentary という名前の通り、必ず p チャネル、n チャネル両方の MOS-FET を使用する。これは、全く同じ入力電圧に対して ON と OFF の逆の動作をするスイッチが使えることを意味し、このような入力に対して自在に V_{CC} 側と GND 側をつなぎ替えることが可能である。(b) では、 A_{in} 、 B_{in} の H 信号に対して OFF になる p-MOSFET を V_{CC} に並列につなぎ、ON になる n-MOSFET を GND に直列接続している。このため、両方が H になった時のみ出力が GND に接続され、それ以外は V_{CC} に接続されて nand 動作をすることがわかる。MOS 回路は次段の入力抵抗が非常に高いのが普通であり、これに対して FET の ON 抵抗は無視でき、ON/OFF 電圧の V_{CC}/GND からのずれは、FET の自己バイアス分で TTL に比べて大きな出力の振幅が取れることがわかる。もちろん、この素朴な回路では、電源側に接続した並列回路と接地側の直

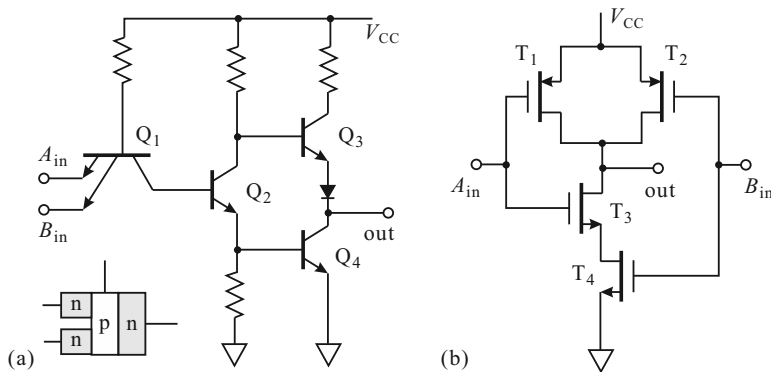


図 7.9 (a) TTL 構成による nand ゲート内部回路の等価回路例. 挿入図はダブルエミッタトランジスタの層構成模式図. (b) CMOS 構成による nand ゲート内部回路の等価回路例. 内向き矢印 MOS-FET は p チャネル, 外向き矢印は n チャネル. (エンハンスモード)

列回路とが遷移領域で同時に ON となって筒抜けになる危険性があるなど, この周辺に様々な安全回路を設ける必要がある.

以上から, 論理ゲートを動作させるための特徴的な電圧が, V_{CC} を 5V に設定した場合, 図 7.10 のようになることが理解される. ただし, 特に TTL の場合, 適当な負荷や電源の周囲配線を行うことが前提となっている. また, V_{CC} を 5V から大きく変更することは, TTL の場合, 通常の半導体物質パラメタから困難であることも理解されるであろう.

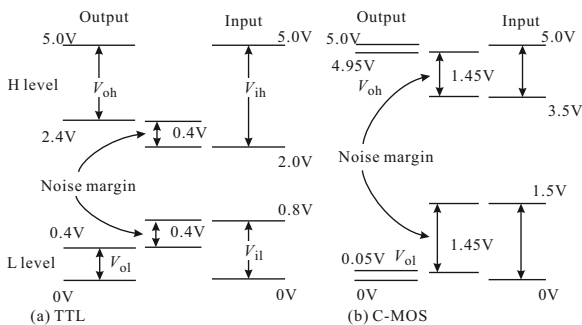


図 7.10 標準論理ゲート動作の電圧ダイアグラム. (a) TTL, (b) C-MOS

歴史的にみると, IC 化され論理ゲートして盛んに使用され始めたのは TTL が先であり, 圧倒的な利用範囲を持っていたため論理信号の H, L が 5V, 0V というのが標準となった. 初期の頃の CMOS IC は, 動作のためにむしろ高い電源電圧を必要とし, TTL 回路と接続のためにはレベル変換回路を用意したりした. また, やはり初期は CMOS の動作速度は TTL を改良した LS 型, ALS 型に及ばず, PC の周辺回路等, ほとんど TTL で組まれていた. しかし, クロック周波数が高くなり LSI の集積度が高くなるにつれて, 素子間の距離をできるだけ短くすることが非常に重要になり, ゲートあたりの素子数が少なく, 電流がゲートの帯電/放電時のもの程度で消費電力が極めて小さい CMOS が次第に有利となった. 素子特性も単ゲート化によってむしろ改善され 5V より低い電圧でも十分動作するようになり速度も向上した. 3.3V や 2.5V を H レベルとする回路も増え, 最近では京などのスーパーコンピュータの CPU も CMOS となり, 現在のところ論理ゲートの世界は CMOS に完全支配されてしまったように見える.

ただし, CMOS も含めて論理演算そのものが現在様々な物理的壁に当たっており, 並列計算などのソフトウェア手法で全体としての行き詰まりを回避しているのが現状であるから, ハードウェア側にも何か大きなブレークスルーがある可能性はある.

7.4 論理演算の回路化と簡単化

アナログ回路が力学系など, 物理系の表現と見ることができたのに対し, デジタル演算回路は, 論理式の具体的表現と見ることが出来る. 従って, 論理式が演算を代数的に行うことで簡単化されれば, 回路を簡単化することになる. その代表的方法を説明する. また, 順序回路は「状態」概念が入ってくるため, これに対応する論理式側の取り扱い概念が必要となる. その 1 つ (ツール) であるダイアグラムを紹介する.

7.4.1 カルノー図

A, B を論理値として論理式

$$Y = A \cdot B + A \cdot \bar{B} + \bar{A} \cdot B \quad (7.1)$$

を考える. \cdot は and, $+$ は or を表す. 論理値においては, $A + A = A$ であったことから, 右辺に $A \cdot B$ を加えて変形すると

$$\begin{aligned} Y &= A \cdot B + A \cdot B + A \cdot \bar{B} + \bar{A} \cdot B \\ &= A \cdot (B + \bar{B}) + B \cdot (A + \bar{A}) = A + B \end{aligned} \quad (7.2)$$

となり, 単なる or 演算であったことがわかる. $A \cdot B$ を加えるような論理演算は式だけでは思いつきにくく, 真理値表を眺むなどの非機械的手法がむしろ有効である. これを比較的機械的に行う手法がカルノー図 (Karnaugh map) の手法である.

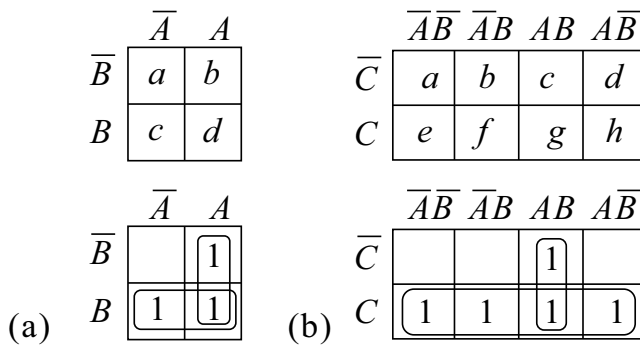


図 7.11 カルノー図の例. (a) 2 論理変数に対するカルノー図. 下は or 論理演算の場合. (b) 3 論理変数に対するカルノー図. 下は, $\bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C + A \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C}$ の場合.

カルノー図とは, ある論理式に対し図 7.11 のようにそれを構成する論理変数のすべての場合を 2 次元に並べて書いたものである. マスの中には出力が 1 (T) となる場合のところに 1 を入れるようにする. 3 変数の場合, 2 変数ずつをまとめて書くと図 7.11(b) のように 2 次元にしておくことができる. この時, 隣の列を作る際に 2 変数同時に否定を取ることはせず, 1 つずつ取って場合を潰すことが必要である. このように論理変数の数が増加すると次第に図が描きにくくなる.

式 (7.1) の場合をカルノー図に描くと図 7.11(a) の下の図になる. 隣接する「1」を で括ると, 2 つの括りができることがわかる. すなわち, このカルノー図が表す論理式は, この 2 つの「括り」が表す論理式の or を取ったものである.

$$Y = A \cdot \bar{B} + A \cdot B + \bar{A} \cdot B + A \cdot B = A + B \quad (7.3)$$

と, 当然 (7.2) の結果が得られる. 図 7.11(b) の下の図は

$$Y = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C + A \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} + A \cdot B \cdot C = A \cdot B + C \quad (7.4)$$

である.

以上見てわかるように, カルノー図による簡単化の原理は

$$A \cdot B + A \cdot \bar{B} = A \quad (7.5)$$

という簡単なものであるが, 図 7.11 のように 2 次元に並べて隣接する 1 を括ることでこのようにして消せるペアを組織的に見つけられる, というところが利点である. なお, 「隣接する 1」を探す際, 行や列の端同士は隣接している, と見なければならぬので注意が必要である. 例えば, 図 7.11(b) の上の図で a と d や e と h は隣接している.

7.4.2 クワイン・マクラスキー法

この簡単化を更に機械的手続きにすることを考える。このためには、最初に与えられた論理式の形は、簡単化を標準化するためにはむしろ障害になることが多く、カルノー図のように一旦真理値表で表してしまい、そこから一定の標準的手続きにより論理式に戻すことにする。

論理式 $Y = f(A_1, A_2, \dots, A_n)$ を考える。真理値 0, 1 に対して、各論理変数を与える関数 g_i を

$$g_i(0) = \bar{A}_i, \quad g_i(1) = A_i \quad (7.6)$$

と定義する。各変数またはその否定をすべて1つずつ含む論理式(項)を**標準項**(canonical term)という。うち、論理和 + を区切りとして分解される「項」の数が最も少ないものを最小項と呼ぶが、最小項は当然項数1個であるから、結局最小項とは、 $\prod_{i=1}^n g_i(a_i)$ で表される。 $a_i = 0$ または 1 で、 \prod は論理積を表す。 Y を真理値表で表す時、ある1行の入力項は 0 または 1 の n 個の真理値列であるから、これを $\{a_i\}$ と見ると、真理値表の各行に1個の最小項が対応することがわかる。そこで、 $Y = 1$ を与える $\{a_i\}$ をすべてリストアップして指数 j を付け、行列の形で $\{a_{ij}\}$ と表すと、 Y の論理式としてこれらで指定される最小項をすべて論理加算したもの

$$Y = \sum_j \prod_{i=1}^n g_i(a_{ij}) \quad (7.7)$$

を考えることができる。このように最小項の論理和での論理値の表現を**主加法標準展開**(principal disjunctive canonical expansion)と呼ぶ。

主加法標準展開から(7.5)を使い冗長項を圧縮していく。この時、最小項を作るために考えた $\{a_i\}$ を並べたものを2進数として取り扱う方法を特にクワイン・マクラスキー法(Quine-McCluskey algorithm)と呼んでいる。これら最小項を表す2進数を、その中に出てくる「1」の個数で分類する。2進数表示での(7.5)は、0と1とを取るある桁以外は同じ2進数の間でこの桁を消去する式であるから、圧縮はこの分類で1つだけ異なるグループ間で行われる。消去した桁は(アンダースコア)として並べることで第1次圧縮リストができる。再度1の個数で分類、圧縮を可能な限り繰り返す。

こうして簡単化された論理式について、各項(「主項」と呼ぶ)を行指数とし、元の主加法標準展開に含まれる最小項を列指数とする(反対でも良い)表(主項図)を作る。最小項の中に何らかの形で主項が含まれている欄に印をつける(○とする)。例えば、主項が $_111$ であれば、0011, 0111, 1011, 1111 の4つがそれに当たる。論理的には「これらの最小項を主項が包含する」ことになる。この表をまず縦に見て、○が1回だけ現れる最小項の○を◎に変更する。次に横に見て、◎がついた主項(必須項)の行に◎になっていない○があればこれをすべて◎に変更する。最後に○しか付いていない主項が省略候補である。すべての最小項をカバーするように必要な主項があれば残し、後を省略することで簡単化が終了する。

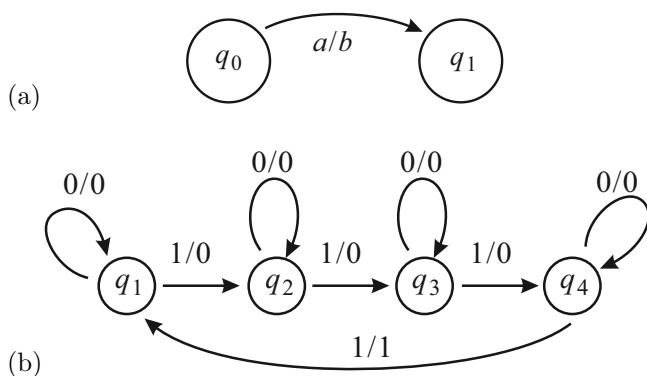


図 7.12 状態遷移図。(a) 概念図。(b) 4進カウンタ。

7.4.3 状態遷移図

順序回路を設計する際に、ダイアグラムとして助けになるのが**状態遷移図**(state transition diagram)である。これは図 7.12(a)のように、入力 a に対して回路の状態が q_0 から q_1 へと遷移し、 b が出力されることを表している。

これを使って、7.2.4 節で見たカウンタ回路の設計手順を追ってみる。簡単のため、2ビット(4進)カウンタとする。カウント入力が入る度に状態が変化し、初期状態から4つ目のカウント入力で1を出力して初期状態に戻るから、状態は全部で q_1, \dots, q_4 の4つで

ある。図 7.12(b) のようにこの 4 つの状態を並べて書き、入力 $x = 0$ の場合は状態が変化せず出力も 0 なので、0/0 でループを描いて元の状態に戻り、1 に対しては状態が 1 シフトし、 $q_4 \rightarrow q_1$ の初期値に戻るときのみ 1 が出力されるので 1/1, それ以外は 1/0 となる、ということで、図 7.12(b) のような状態遷移図が描かれる。

2 ビットであるから T-FF を 2 個使用して実現することにする。これら 2 個の出力を、状態 q_n に対してそれぞれ、 $Q_n^{(1)}$, $Q_n^{(2)}$ と書く。 n は 4 進で回る、すなわち $4 + 1 = 1$ とする。状態遷移表をカルノー図の形にするため、 $Q_{n+1}^{(1)}$ と $Q_{n+1}^{(2)}$ とに分けて描くと次のようになる。ただし、最初にビットが上がる方を (2) の T-FF としている。 $(Q_{n+1}^{(i)})$ を決めるための論理変数が、 x , $Q_n^{(1)}$, $Q_n^{(2)}$ の 3 つ存在するため、図 7.11(b) を縦にした図になっている。

		$Q_{n+1}^{(1)}$				$Q_{n+1}^{(2)}$	
		x				x	
$Q_n^{(1)}$	$Q_n^{(2)}$	0	1	$Q_n^{(1)}$	$Q_n^{(2)}$	0	1
0	0			0	0		1
0	1		1	0	1	1	
1	1	1		1	1	1	
1	0	1	1	1	0		1

カルノー図であるから、 で隣接する 1 を囲い、単純化により次のような漸化式を得る。

$$Q_{n+1}^{(1)} = \bar{x} \cdot Q_n^{(1)} + Q_n^{(1)} \cdot \overline{Q_n^{(2)}} + x \cdot \overline{Q_n^{(1)}} Q_n^{(2)}, \quad (7.8a)$$

$$Q_{n+1}^{(2)} = \bar{x} \cdot Q_n^{(2)} + x \cdot \overline{Q_n^{(2)}}. \quad (7.8b)$$

また、一般に FF については、入力 x で表される論理式 α , β を用いて

$$Q_{n+1} = \alpha Q_n + \beta \overline{Q_n} \quad (7.9)$$

という、特性方程式 (characteristic equation) と呼ばれる式が成立する。実際、図 6.5(b) の T-FF の真理値表から、 $\alpha = \bar{T}$, $\beta = T$ として、(7.9) が成り立っていることがわかる。T-FF の (1), (2) に対応して $\alpha_{1,2}$, $\beta_{1,2}$ と添字をつけると、(7.8a), (7.8b) より、

$$\alpha_1 = \bar{x} + \overline{Q_n^{(2)}} = \overline{x \cdot Q_n^{(2)}}, \quad \beta_1 = x \cdot Q_n^{(2)}, \quad (7.10a)$$

$$\alpha_2 = \bar{x}, \quad \beta_2 = x \quad (7.10b)$$

である。以上から、図 7.13 の回路図がほぼ自動的に描かれる。

7.5 A-D/D-A コンバータ

アナログ信号をデジタル信号に変換する、あるいはその逆の動作をする回路は、物理実験では極めて重要である。D-A/A-D 変換回路技術は、音響・映像技術にデジタル技術が入り込んで以来飛躍的に発展した。素子性能の向上による性能向上ばかりでなく、信号形式、回路方式の改革によって古い電子回路の教科書はみな書き換えなければならないほどの進歩が積み重ねられている。ただし、A-D/D-A 技術に関しては、CCD 技術、画素式ディスプレイの発達によりまず映像系からアナログ回路がほとんどなくなり、音響においても D 級アンプやスピーカーのデジタ

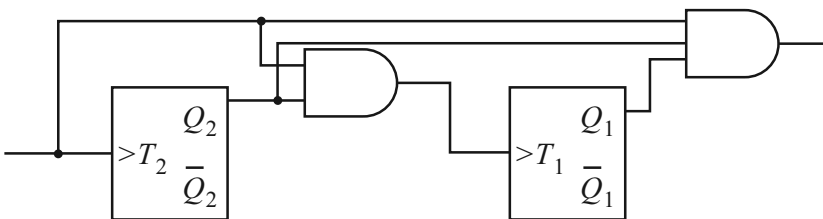


図 7.13 設計した 4 進 (2 ビット) カウンタ

ル駆動によりアナログ回路はデジタルのカプセルの向うの存在となりつつあることから、今後はこれまでのような急速な進展はないかもしれない。

「変換回路」と言っても、一方(アナログ)が物理量であるのに対して他方(デジタル)は、人間がこれを記述するために考えた記号であるから、「変換」のためには極めて様々なアイデアが存在しうる。これらは、変換形式に従い、外部クロックが必要になるなどの若干の使用法の違いはあるものの、A-Dコンバータ(ADC)、D-Aコンバータ(DAC)としてLSI化されてカプセル的に使用することが可能で、内部形式を知らなくても使えてしまう。それでも、どのようにして変換しているのか知っておくことは、電子回路の応用例としても悪くはない。代表的な回路形式を紹介しておこう。

7.5.1 デジタル-アナログ変換

まず、デジタルからアナログへの変換を考えよう。

抵抗ストリング型、ラダー型

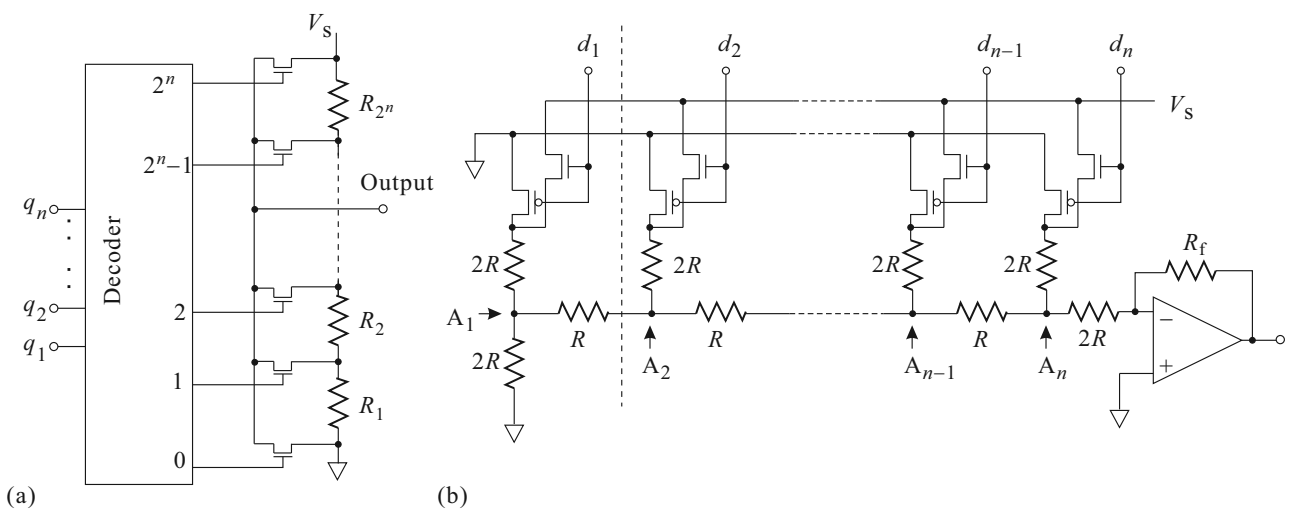


図 7.14 抵抗ネットワークを用いた D-A コンバータの等価回路図。(a) 抵抗ストリング型。(b) 抵抗ラダー型。FET スイッチは、ゲートに○がついていないものが n チャネルで H 入力に対して ON, 反転記号である○がついているものが p チャネルで H 入力に対して OFF となる。

抵抗ネットワークを使用するものの内、図 7.14(a) に示した抵抗ストリング型は、余り現実性はないが、コンセプトとしては最も単純なものである。n ビットのデジタルデータは 2^n の分解能を持っている。そこで、回路は標準電圧 V_S を持っているとし、これを、 2^n 個の同一抵抗値 R を持つ抵抗を直列につないだ回路に印可する(他端はグラウンド)。グラウンドから順番に抵抗に $1, \dots, 2^n$ の番号を付けると、 i 番目と $i+1$ 番目の抵抗の間の配線部の電位は、 $i \times V_S / 2^n$ となり、出力へは図のように、MOS スイッチを用いて接続する。受電部の入力インピーダンスを十分高くし、バイアス電流を抵抗ストリングに流す電流に比べて十分に低く取ることで $i \times V_S / 2^n$ をそのまま出力できる。

この方法は現実問題として、8 ビットでも 256 個の高精度抵抗と MOS スイッチを結線する必要があり、デコーダーも同じだけの数の出力を要するなど、使いにくいものである。

これに対して実際にもよく使用されるのが図 7.14(b) の抵抗ラダー型 DA 変換である。n-1 個の抵抗値 R の抵抗と n+2 個の値 $2R$ の抵抗を用意し、n ビットの入力ラインを図のように相反性 MOS スイッチのゲートに接続し、 $2R$ の抵抗を介して R の抵抗を直列にした各端点に接続する。最低ビットラインの端点は $2R$ で接地し、他端はやはり $2R$ でこちらは OP アンプの反転増幅器の - 入力への入力抵抗として使用する。

OP アンプの項で学んだように、- 入力端子は仮想接地されているので、ここでは右端の $2R$ もやはり接地されているものとして A_n 点の電位を考える。今、k ビット目が 1(H) で他はすべて 0(L) が入力されたとする。 A_n 点から右および上方の接地側を見ると $2R$ の抵抗が 2 個並列になっており、合成抵抗 R で接地されている。従ってその左の

A_{n-1} 点から右方向の A_n 点を通る経路の接地までの合成抵抗はやはり $2R$ で A_n 点と状況が全く同じであることがわかる。これが、 A_{k+1} 点まで続く。 A_{k-1} 点より左側を見ても、全く同じ状況である。従って A_k 点は、左右に $2R$ で接地され、 V_S より $2R$ を通して接続されていることがわかる。結局流れる全電流は $V_S/3R$ 、片方の分岐に流れるのはその $1/2$ である。以下、分岐ごとに $1/2$ になるので、最終的に OP アンプ方向へ流れ出す電流は

$$J_{\text{out}} \begin{pmatrix} 0 \cdots 0 & 1 & 0 \cdots 0 \\ n & k & 1 \end{pmatrix} = \frac{V_S}{3R} \left(\frac{1}{2}\right)^{n-k+1} = \frac{V_S}{6 \cdot 2^n R} 2^k$$

となる。 J_{out} の引数は、入力を 2 進数で表したものになっている。従って、入力 $\{d_i\}$ ($d_i = 0$ or $1, i = 1, \dots, n$) があつた時の OP アンプ出力は、重ね合わせの定理 (2.5 節) を用いて、

$$V_{\text{out}}(\{d_i\}) = -\frac{1}{3 \cdot 2^n} \frac{R_f}{2R} V_S \sum_{k=1}^n 2^k d_k \quad (7.11)$$

となつて、AD 変換がされている事がわかる。ストリング型に比べて必要抵抗は圧倒的に少なく、デコーダなども必要としない。実際多用される形式である。

この他、受動素子ネットワークを用いるものとしては、容量アレイ型などがある。

電流加算型

原理的にはたいへん簡単な変換法であり、レポート問題として考えてもらう予定。

容量アレイ型

これも簡単なので、レポート問題として考えてもらう予定である。

パルス幅変調型

パルス幅変調 (pulse width modulation, PWM) は、どちらかと言うとアナログ信号を PWM 信号にしてパワートランジスタをスイッチすることで電力増幅し、ローパスフィルターでアナログ信号に戻す、アナログ電力増幅に用いることが多い。が、無論、速くて正確なクロック信号とカウンタを用いてデジタル的に PWM 信号を作り出すことができ、これをローパスフィルターに通すことで DA 変換ができる。

原理は以上で終了、という位簡単明瞭である。逆に PWM 信号を作り出す方法の一例を図 7.15 に示した。クロックに同期したのこぎり三角波を作り出し、原信号でベースラインをシフトさせる。これをコンパレータに入れてこの変調三角波が閾値電圧を超えた領域だけ H 信号を出すようにすることで、PWM 信号が得られる。

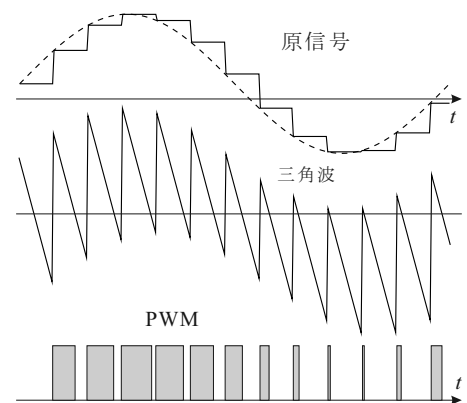
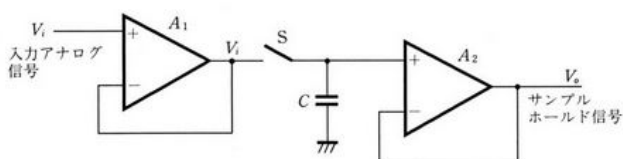


図 7.15 パルス幅変調法の一例。

7.5.2 アナログ-デジタル変換

逐次比較型

この方式では標本回路 (sample and hold circuit) により入力電圧を固定しておき (これは、他の多くの方式でも用いられる)、DA 変換によって既知電圧を作り出し、これとホールドした電圧とが一致するかどうかを調べることでデジタルデータに変換する。様々な意味での「既知電圧」を次々とホールド電圧と比較することから逐次比較 (successive approximation) 型と呼ばれている。



サンプルアンドホールド回路 (AD 変換回路ではトラックアンドホールドと呼ばれることもある) にも多くの形式があるが、左図に挙げるものが最も簡単である。信号によってキャパシタにチャージし、これを OP アンプにより孤立状態にして維持する。新しくサ

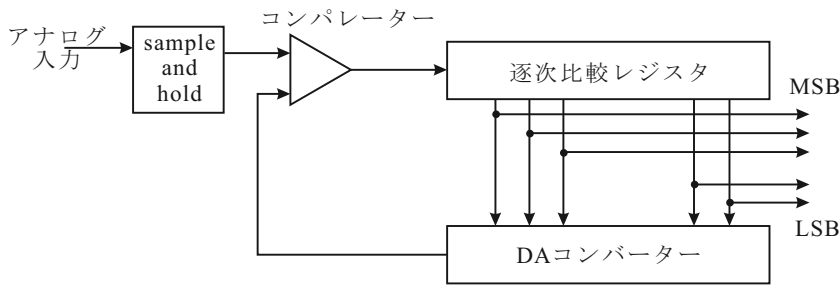
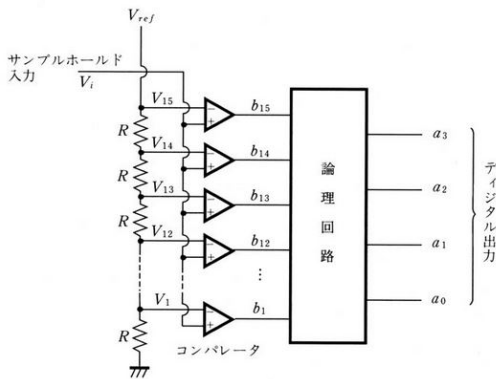


図 7.16 逐次比較型 AD コンバーターのブロックダイアグラム

サンプルするには、放電するのが一般的であるが、信号時間変化が緩やかでかつ低出力抵抗である場合は、帯電状態で接続することも考えられる。

2進数に変換することから、逐次比較は高いビットから、数値計算でいう2分法によって行うのが一般的である。最初に最高ビットのみを立てた参照電圧をDACにより用意し、コンパレータで比較する。コンパレータもまた、OPアンプを開ループ(可能な限り大きな開ループゲインを持たせる)で使用するのが精度的に有利である。参照電圧がサンプルより大きければ0、小さければ1が立つ。このデジタル信号の次のビットを立てたものをDACに入れて次の参照電圧を作る。以下最終ビットまで繰り返すことでデジタル変換が完了する。

フラッシュ型



逐次比較型は1ビットごとに変換していくので変換回路は1回路で済むが、多ビットのAD変換を行う場合にはビット分だけのサイクルを繰り返す必要があり時間がかかるようになる。これに対して、多数のコンパレータを並べて一気にAD変換を行おうというのがフラッシュ型ADCである。左図のように、抵抗ストリング型DACのちょうど逆のような形式である。

パイプライン型

フラッシュ型は逐次比較型に比べて圧倒的な速度を持っているが、当然ながら分解能分の分割抵抗とコンパレータという消費リソースも巨大となる。また、精度限界はコンパレータの開ループゲインAで決まることが多い。例えば、Aが100dBだと

すると、入力 $10\mu\text{V}$ で出力は1Vになり、デジタル回路がTTLレベルで動作しているとすると、しきい値電圧 V_{th} には到達しない。さりとて、アナログアンプで例えば10倍に増幅して入力しようとする、アンプの出力、あるいはコンパレータの入力の飽和値によってアナログ電圧の大きさが制限されてしまう。

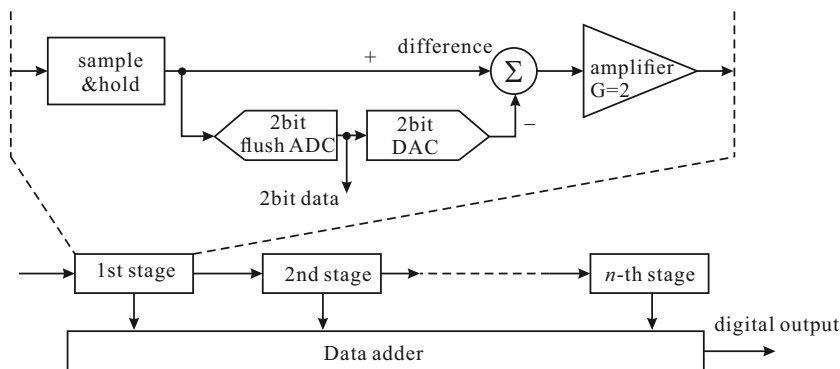


図 7.17 2ビット単位パイプライン型ADCのブロック図

このような欠点を、速度を若干犠牲にしながらカバーする方式が幾つか使われており、パイプライン型はその1つである。図7.17はそのブロック図(後述)で、数ビットをまとめてフラッシュ型でAD変換し、残った信号を増幅器で増幅しながら次のサイクルを行うようになっている。フラッシュ型と逐次比較型の間の子のような形式である。

積分型

逆に時間がかかるができるだけ少ない部品リソースでAD変換を行おうというのが積分型ADCで、積分回路に定電流を流して時間に対してリニアに増加する電圧を作り出し、入力電圧とコンパレータで比較する。増加電圧が下端からスタートする際にデジタル

ルカウンタをスタートし、増加電圧が入力電圧をクロスしてコンパレータが ON になった時点でストップする。カウンタ出力により AD 変換ができる。積分回路とコンパレータの OP アンプと積分用のキャパシタで精度がほぼ決まり (カウンタの桁数を出すのは比較的容易)、安価に AD 変換を構成できるため、パネルメータ (機器の表面に沢山はめ込んで、様々なパラメータをモニターできるようにするもの) などに良く使用されている。

デルタシグマ型

オーバーサンプリング型 と呼ばれる ADC の代表的な形式である。1 ビット (従ってコンパレータ 1 個のみ) の超低分解能 ACD を用いフィードバックを使ったデルタシグマ変換により高精度デジタル信号に変換する。詳細は後述。

7.5.3 情報の符号化

「デジタル信号序論」で述べたように、パルス振幅変調 PAM でパルス高を離散化すれば、デジタル信号になる。それをそのまま変調の形にしたのが **パルス符号変調** (pulse code modulation, PCM) である。PCM データは、クロックに合わせて何ビットかのデータを伝送するもので、少ない伝送チャンネルで送る場合は、時系列信号にしてシリアル伝送する。復調する必要がある場合は、パラレルデータに変えて DAC によりアナログ信号にする。

この直接的な PCM 法に対して、様々な変形がある。デジタル信号は符号化されたものであるため、急激な信号値変化もデータ上は表現できるが、アナログ信号は、特にそれに含まれている周波数成分の上限で表現できるよりも大きな変化は生じない。そこで、値そのものではなく前の信号値との差分を抽出して伝送することも考えられる。これを差分 PCM (differential PCM, DPCM) 方式と呼び、予測符号化方式の一種である。DPCM は一般に PCM より小さなビットレートで同精度の信号を伝送することができる。「予測」という文字がついているのは、実際には単なる差分ではなく、最も単純な前値予測方式であっても予測係数がついて、受信データ $\{x_i\}$ に対する予測値は $x'_i = ax_{i-1}$ で与えられ、送付データは差分 $\epsilon_i = x_i - ax'_i$ である。

また、PCM でない符号化形式 (時間領域で変換してから符号化するもの) など、符号化にも沢山の形式があるが、とても踏み込む余地がないので本講義ではここまでとする。

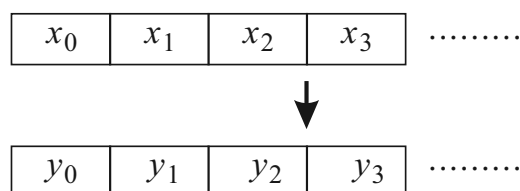
7.6 デジタルフィルタ

一旦デジタルに落とした信号は、回路中で数値として扱うことができるので非常に自由度が高く、安定な処理が可能である。このような「処理」の内、デジタル信号からデジタル信号への変換処理を **フィルタ** (filter) 作用と呼ぶ。デジタル信号処理 (digital signal processing, DSP) も極めて巨大な分野であり、講義の 1 回程度ではほとんど「舐めてみる」程度のことしかできないが、デジタルフィルタの初歩について考え、入り口を経験することにしよう。なお、この **DSP** という略語は「デジタル信号処理」という意味、および「デジタル信号プロセッサ」 (digital signal processor) の両方に用いられ、混用されている。

デジタルフィルタは物理的に行っていることはアナログフィルタとは全くと言って良い程異なっているが、応答関数の形に書くと、類似の舞台の上で考えることができる対象であることがわかる。デジタルフィルタは、アナログフィルタがスマートに滑らかに行っていたことをゴツゴツと各段階計算で押し切っているので必要な回路リソースは桁違い以上に大きい、特性が部品特性に制約されず設計に対する忠実度が高く、自由な設計が可能であり、何より、制御信号などによって容易に特性を変化させることができる。

7.6.1 ブロック・ダイアグラム

デジタルフィルタの働きを簡単にまとめると、右図のように見るのが 1 つの考え方であろう。すなわち、時系列で並んだ数列 $\{x_i\} = (x_0, x_1, \dots)$ に対して、別の数列 $\{y_i\} = (y_0, y_1, \dots)$ を得るプロセスである。ここで、数列値 y_n は、 x_n およびこれよ



りも以前の k 個の x_i の値によって決定されるとすると, $k + 1$ 変数の関数 F を使って

$$y_n = F(x_{n-k}, x_{n-k+1}, \dots, x_n) \quad (7.12)$$

と書けることになる.

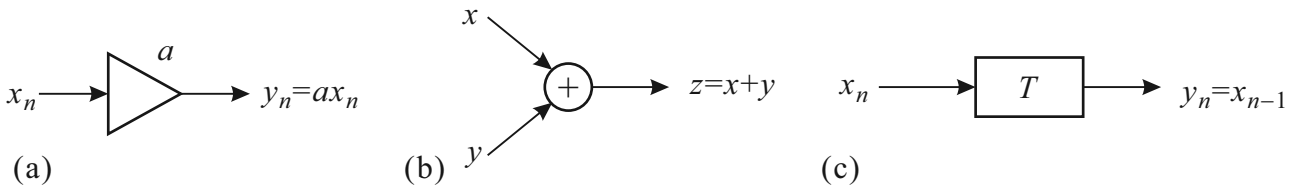


図 7.18 デジタルフィルタ用ブロックダイアグラムに使われる, 「基本ブロック」. (a) 定数倍回路. (b) 加算回路. (c) 遅延回路 (シフトレジスタ).

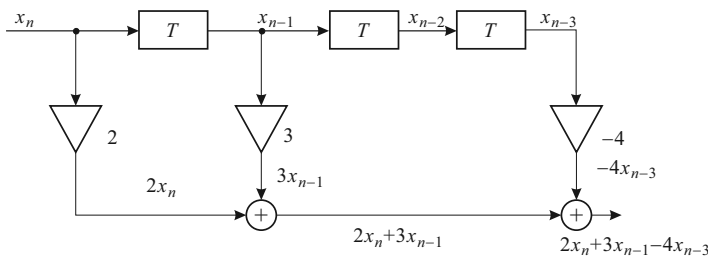


図 7.19 式 (7.13) のダイアグラム化. タル回路のバッファの形をしているが, 記入されている数倍だけする, **定数倍回路**, である. (b) は信号の和を取る **加算回路**である. (c) は, 信号を時間領域で 1 単位だけ遅らせる **遅延回路**である.

例として

$$y_n = 2x_n + 3x_{n-1} - 4x_{n-3} \quad (7.13)$$

をダイアグラム化すると前頁の図のようになる.

7.6.2 z 変換空間表示

デジタルの線形応答回路においてもアナログ回路同様, z 変換空間において伝達関数を使って応答関係を表すことができることを示した. そこで, デジタルフィルタも (片側) z 変換空間で表すことを考える.

ここではやはり例として (7.13) を考える. 今の場合, フィルタの入出力の z 変換は定義 (6.83) より,

$$X(z) = \sum_{n=0}^{\infty} x_n z^{-n}, \quad Y(z) = \sum_{n=0}^{\infty} y_n z^{-n} \quad (7.14)$$

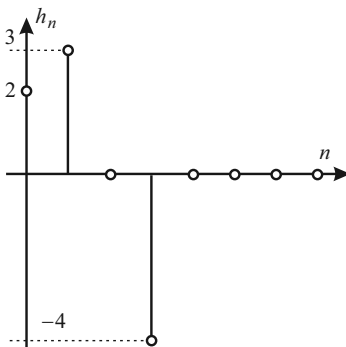
である. ここで, 表 5.2 に示した片側 z 変換の性質を思い出すと, 線形性より加算や定数倍はそのまま記述することができ, 周期 τ の k 回文の遅延は時間移動で z^{-k} をかければ良いことがわかる. 以上より, z 変換空間では

$$Y(z) = 2X(z) + 3z^{-1}X(z) - 4z^{-3}X(z) = (2 + 3z^{-1} - 4z^{-3})X(z) \quad (7.15)$$

で表されることがわかる. これより, 伝達関数を $H(z)$ とすると, (6.89) より

$$H(z) = 2 + 3z^{-1} - 4z^{-3} \quad (7.16)$$

となることがわかる.



伝達関数は $t = 0$ でインパルス入力があった時の応答の z 変換であるから、これより、このデジタルフィルタにインパルス入力があると、前ページの図のような応答があることになる。一般にフィルタの関数 F (7.12) が、

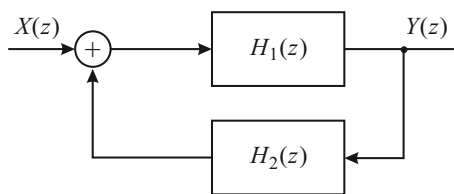
$$F(x_{n-k}, x_{n-k+1}, \dots, x_n) = \sum_{j=0}^k a_j x_{n-j} \quad (7.17)$$

のようにかける場合、伝達関数は

$$H(z) = \sum_{j=0}^k a_j z^{-j} \quad (7.18)$$

となる。

7.6.3 ブロックダイアグラムと伝達関数



z 変換空間内でもブロックダイアグラムを考えることができる。この際、伝達関数をブロックで表してしまうのが簡単である。左図のように、2つの伝達関数 $H_1(z)$ と $H_2(z)$ とを使って、フィードバック構造をつくるブロックダイアグラムを考えてみる。 $X(z)$ から + 演算を経て H_1 へ入力される関数を $W(z)$ と置くと、

$$Y(z) = H_1(z)W(z) = H_1(z)(X(z) + H_2(z)Y(z)),$$

$$\therefore Y(z) = \frac{H_1(z)}{1 - H_1(z)H_2(z)}X(z).$$

すなわち、全体の伝達関数は

$$H(z) = \frac{H_1(z)}{1 - H_1(z)H_2(z)} \quad (7.19)$$

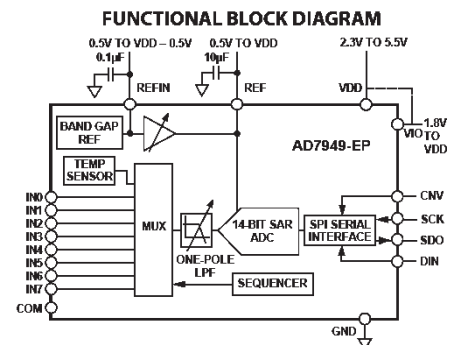
これは、むしろ、アナログ回路の場合の (4.3) 式に対応するもので、フィードバック回路の場合、一般に

$$(\text{全体の伝達関数}) = \frac{(\text{直接部分の伝達ゲイン})}{1 - (\text{フィードバック部分の伝達ゲイン})} \quad (7.20)$$

となることを示している。ここで、フィードバックをかけると、式 (7.12) の引数が形式上無限に続くことになることに注意しよう。

付録 K : A-D/D-A コンバータ選択・使用上の注意

現在の物理実験家が自ら A-D/D-A 回路を設計製作することはまず考えられないが、集積回路の形で用いる可能性はある。デジタルオーディオの発展に伴って驚異的な多ビットの DAC などが二束三文で売られるようになり、これらを上手に使用することで高速/高精度の実験測定系が安価・低労力で得られる可能性がある一方、絶対精度等には注意が必要である。



現在オーディオ用に 24 ビットの DA 変換 IC は極めて安価に手に入る。AD 変換も同様であり、中には 32 ビットのような高いビット数を持つものもある。これらはほとんど Δ - Σ 型の AD 変換器を用いており、相対精度が非常に高くして便利である。が、絶対精度はあまり当てにできないので、計測に用いる場合は精密機器による較正が必要である。

実際に ADC/DAC を選択する場合、データシートをきちんと見る必要がある。まず、どのような回路が使用されているのかを把握し、トラブルが生じた際に備える (回路形式が原因の可能性ある)。これには、ブロック図を見ておけば十分で、データシートには大抵簡単なブロック

図が掲載してある。左図は AD7949 の場合であり、中に MUX と書かれた 8 チャンネルのマルチプレクサ (入力チャンネル切替器) が入っており、ボード線図で「極」が 1 つだけある低域通過フィルター (LPF) が入れられていることがわかる。その先は「14bit SAR ADC」につながっていて、この IC が 14 ビットの抵抗逐次比較型 (successive approximation resistor) の ADC であることがわかる。

次に見なければならないのが specification table で、面倒でも定格電圧やノイズ、不確定性の最悪値などをチェックし、実験の用途に耐えるものかどうかを調べておく必要がある。

AD7949-EP					
SPECIFICATIONS					
VDD = 2.3 V to 5.5 V, VIO = 1.8 V to VDD, VREF = VDD, all specifications -55 °C to +125 °C, unless otherwise noted.					
Table 2.					
Parameter	Conditions/Comments	Min	Typ	Max	Unit
RESOLUTION		14			Bits
ANALOG INPUT					
Voltage Range	Unipolar mode	0		+VREF	V
	Bipolar mode	-VREF/2		+VREF/2	V
Absolute Input Voltage	Positive input, unipolar and bipolar modes	-0.1		VREF + 0.1	V
	Negative or COM input, unipolar mode	-0.1		+0.1	V
	Negative or COM input, bipolar mode	VREF/2 - 0.1	VREF/2	VREF/2 + 0.1	V
Analog Input CMRR	fIN = 250 kHz		68		dB
Leakage Current at 25°C	Acquisition phase		1		nA
Input Impedance ¹					
THROUGHPUT					
Conversion Rate					
Full Bandwidth ²	VDD = 4.5 V to 5.5 V	0		250	kSPS
	VDD = 2.3 V to 4.5 V	0		200	kSPS
¼ Bandwidth ²	VDD = 4.5 V to 5.5 V	0		62.5	kSPS
	VDD = 2.3 V to 4.5 V	0		50	kSPS
Transient Response	Full-scale step, full bandwidth			1.8	µs
	Full-scale step, ¼ bandwidth			14.5	µs
ACCURACY					
No Missing Codes		14			Bits
Integral Linearity Error		-1	±0.5	+1	LSB ³
Differential Linearity Error		-1	±0.25	+1	LSB

図 K.1 AD7949 の specification table の一部

最後に、A-D 変換方式と分解能、変換速度の「分布図」をあげておく。

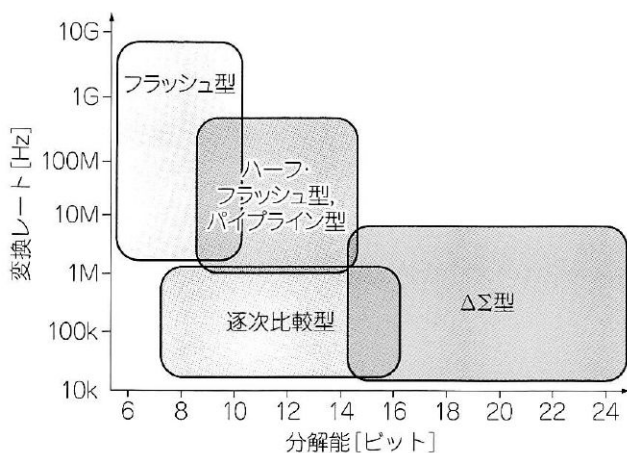


図 K.2 A-D 変換器の回路方式による、分解能、変換速度に対する分布図。