HDL記述による設計法をマスタする

実験で学ぶロジック回路設計

木村 真也
Shinya Kimura

第2回　ロジック設計の基礎の基礎

今回は，前回に引き続き，ロジック回路を設計するときに知っておくべき基礎的な事柄の解説です．

▶ロジック回路界の数値を表すのに使う「2進数」

ロジック回路では，基本的には‘0’と‘1’しか扱えないのですが，このままでは数値を表すことができません．そこで，‘0’と‘1’を並べた数字の列を使って数値を表すことにします．これが2進数です．

▶値を一時的に保持する回路「フリップフロップ」

単純なゲート回路の組み合わせでは，入力が変わると出力も変わってしまいます．前の値も欲しい，という場合も多いでしょう．それを実現するのが，フリップフロップと呼ばれる回路です．タイミングを合わせる用途にも使われ，現在のロジック回路はフリップフロップなしでは考えられません．　(編集部)

ロジック回路での数値表現…2進数

ブール代数では，結果が‘0’と‘1’のどちらかでした．答えがYesかNoだけの回路を扱うならこれでもよいのですが，実際のロジック回路では，数値や量を表すデータを扱わなければいけないことも多いはずです．そこで，‘0’と‘1’を並べて扱う方法，つまり2進数の出番です．

● どうやって1と0だけで大きな数を表現する？

2進数の1桁(1ビット)では0と1しか表現できませんが，桁数(ビット数)を増やすことで，どんなに大きな数でも表現できます．

多数桁の2進数は，**図2-1**に示すような重み付けが各桁にあります．2進数によって表された値は‘1’になっている桁の重みを合計して求められます．

重み付けの意味は，10進数の場合を考えるとわか

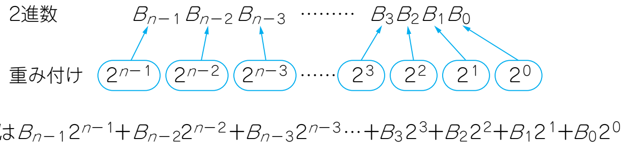


図2-1　2進数を使えば0と1以外の値を表現できる
値が大きくなれば多くの桁が必要になる

Keyword 1　ビット

2進数の1桁のことをビット(bit：binary digit)と呼んでいます．

8桁(8ビット)の2進数を1バイト(byte)と呼んでおり，ほとんどのコンピュータにおいて，メモリのアドレス付けの単位となっています．

また，複数桁の2進数において，左端の桁の1ビットをMSB(Most Significant Bit)，右端の桁の1ビットをLSB(Least Significant Bit)と呼んでいます(**図2-A**)．

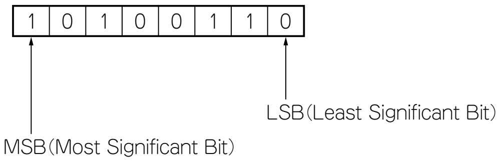


図2-A　MSBとLSB

りやすいでしょう．10進数では，各桁の重み付けが下の桁から一，十，百，千…となっているわけです．

一般に，n桁の2進数は0から2^n-1までの値を表現することができます．4桁(4ビット)の2進数と，10進数の対応を**表2-1**に示します．

● 2進数どうしの加算

2進数で表された値どうしの加算は，基本的には10進数の場合と同じです．

最下位の桁から1桁ずつ計算をしていきます．足した結果が“10”(10進数では2)になれば，その桁の和は0で，繰り上がり(キャリー)が発生します．

次の桁はキャリーも含めて計算することになります．つまり，最下位の桁以外では，その桁の値二つと下の桁からのキャリーを含めた，三つの値による加算(3ビットの加算)を行うことになります(**図2-2**)．

表2-1　4ビットの2進数と10進数の対応

2進数	10進数	2進数	10進数
0000	0	1000	8
0001	1	1001	9
0010	2	1010	10
0011	3	1011	11
0100	4	1100	12
0101	5	1101	13
0110	6	1110	14
0111	7	1111	15

2進	10進
0110	6
−0001	−1
0101	5

図2-3　2進数の減算の例

'0'−'1'は繰り下がりになるので一つ上の桁から1引く

2進	10進
0111	7
−1000	−8
11111 →	31…?

図2-4　負の数がないので減算の答えがおかしい

元の値が4ビットなので，5ビット目を考えるのは本来おかしい

● 2進数どうしの減算

2進数で表現された値どうしの引き算も，10進数の場合と同様，最下位の桁から各桁ごとに計算をします．

引けない場合は，上の桁からの借り(ボロー)が発生します．借りる値は“10”(10進数での2)になります．‘0’ − ‘1’ は ‘1’ で，ボローが発生します．

次の桁は最下位からのボローも含めて計算することになります．最下位の桁以外では，その桁の値二つと下の桁からのボローも含めて，三つの値による引き算(3ビットの引き算)を行うことになります(**図2-3**)．

引かれる値が引く値より小さい場合，答えは負の値になりますが，そのまま計算すると，**図2-4**のように正しい結果が得られません．

● ＋や－の符号を使えない2進数で負の値をどうやって表現するの？

この**図2-4**の結果は－1のはずです．そこで，このような手順で得られた2進数を－1として扱えないか，と考えることができます．これが，負の値を表現する方法として一般的な「2の補数」による表示です．

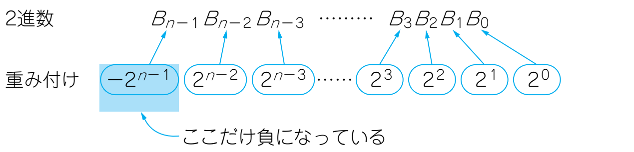
先の引き算の例でいうと，演算結果のうち，下位4ビットを－1と見なす方法です．4ビットの数(4桁の

2進	10進
0111	7
＋0001	＋1
1000	8

2進	10進
1100	12
＋0111	＋ 7
10011	19

図2-2　2進数の加算の例

'1'＋'1'は繰り上がりになるので一つ上の桁に1加える



値は$-B_{n-1}2^{n-1}+B_{n-2}2^{n-2}+B_{n-3}2^{n-3}\cdots+B_3 2^3+B_2 2^2+B_1 2^1+B_0 2^0$

図2-5　負の数も表現する「2の補数」表示の2進数

Keyword 2　補数

通常，ある数xの負の数は$-x$と表記し，両者を加えると0になります．しかし，符号を使えない場合もあります．

いま，2桁の10進数を考えます．ある数xを25とし，75を加えると100になります．扱う桁数を2桁としたので，下位2桁に注目すると，加算結果は00となります．

ここで「足して0になる」ということから，75を25に対する負の数とみなすことができます．

このような考え方で負の数を表現する方法を「補数表示」と呼んでいます．この例の場合，正確には10の補数表示です．

10進2桁の場合，0～49が正の数，50(－50)から99(－1)が負の数となります．

一般に，t桁のr進数の場合，xに対する補数表示の負の数yは次式で求まります．

$$y=r^t-x$$

4ビットの2進数の場合，xに対する2の補数yは，

$$y=2^4-x$$
$$=“10000”-x$$
$$=“1111”-x+1$$ ←1111－xはxの反転

となり，xを反転して＋1で補数を求められます．