コンピュータ工学 講義プリント(4 月 24 日)

・2 進法と 2 進数

2 進法とは、0 と 1 の 2 つの数字だけで数を表す方法である。2 進法で表した数を 2 進数と呼ぶ。コンピュータ内部では、2 進数で数値を表すことが多い。これはデジタル回路が低い電圧(L)と高い電圧(H)の 2 種類の電圧で動作するからである。通常 2 進数の 0 を低い電圧に、1 を高い電圧に対応させる事が多い。2 進数 1 桁の持つ情報量を 1 ビットという。

・10 進数、2 進数、8 進数、16 進数

人が使う数は、通常 10 進数であり、0～9 までの 10 種類の文字を使って表した数である。これは、人間の指が 10 本である事に由来している。

コンピュータ内部では 2 進数が使われる事は話したが、2 進数表記だと、数字の桁数が大きくなるため、人が扱うには不便な事がある。そこで、代わりに 16 進数が使われる事が多い。16 進数は 0～9、A、B、C、D、E、F の 16 文字の数字で表した数である。16 は 2 の 4 乗なので、16 進数 1 桁の情報量は、2 進数 4 桁の情報量(4 ビット)に相当する。

また、16 進数よりは使用頻度が低いが、8 進数が使われる事もある。8 は 2 の 3 乗なので、8 進数 1 桁は 3 ビットに相当する。

次に 10 進数、2 進数、8 進数、16 進数の対応表を示す。

表 1、10 進数、2 進数、8 進数、16 進数の対応表

10 進数	2 進数	8 進数	16 進数	10 進数	2 進数	8 進数	16 進数
0	0	0	0	8	1000	10	8
1	1	1	1	9	1001	11	9
2	10	2	2	10	1010	12	A
3	11	3	3	11	1011	13	B
4	100	4	4	12	1100	14	C
5	101	5	5	13	1101	15	D
6	110	6	6	14	1110	16	E
7	111	7	7	15	1111	17	F

ただ単に 1010 と書いた場合、それが何進数か、区別が付かないので、2 進数の場合は末尾に B を付けることがある。(1010B) 16 進数の場合は末尾に H を付けて区別する。(1010H)

・2 進数・16 進数から 10 進数への変換

例えば 1010B は、$1\times2^3+0\times2^2+1\times2^1+0\times2^0=10$ という風に、10 進数に変換できる。また 1A6FH は、$1\times16^3+10\times16^2+6\times16^1+15\times16^0=6767$ という風に、10 進数に変換できる。(AH は 10、FH は 15 である事に注意)

・10 進数から 2 進数・16 進数への変換

10 進数から 2 進数に変換するには、変換したい 10 進数を 2 で割り、商と余りを求め、さらにその商を

2 で割って商と余りを求め…という具合に 2 で割る作業を繰り返し、商が 0 になるまで続ける。次に、計算の過程で出てきた余りを逆順に並べると 2 進数になる。

例えば 54 を 2 進数に変換すると、次の様な計算過程で 110110B になる。

		余り
2	54	
2	27	― 0
2	13	― 1
2	6	― 1
2	3	― 0
2	1	― 1
	0	― 1

$54 = 27 \times 2 + 0 = (1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2 + 1) \times 2 + 0$
$= 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2 + 0 = 110110B$

$27 = 13 \times 2 + 1 = (1 \times 2^3 + 1 \times 2^2 + 0 \times 2 + 1) \times 2 + 1$
$= 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2 + 1 = 1101B$

$13 = 6 \times 2 + 1 = (1 \times 2^2 + 1 \times 2 + 0) \times 2 + 1 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2 + 1$
$= 1101B$

$6 = 3 \times 2 + 0 = (1 \times 2 + 1) \times 2 + 0 = 1 \times 2^2 + 1 \times 2 + 0 = 110B$

$3 = 1 \times 2 + 1 = 11B$

$1 = 0 \times 2 + 1 = 01B = 1B$

余りを下から上に並べると 110110B となる

図 1、10 進数から 2 進数への変換

16 進数から 10 進数に変換するには、変換したい 10 進数を 16 で割り、商と余りを求め、さらにその商を 16 で割って商と余りを求め…という具合に 16 で割る作業を繰り返し、商が 0 になるまで続ける。次に、計算の過程で出てきた余りを逆順に並べると 16 進数になる。

例えば、3741 を 16 進数に変換すると、次の様な計算過程で E9DH になる。

		余り
16	3741	
16	233	― 13 = DH
16	14	― 9 = 9H
	0	― 14 = EH

$3741 = 233 \times 16 + 13 = (14 \times 16 + 9) \times 16 + 13$
$= 14 \times 16^2 + 9 \times 16 + 13 = E9DH$

$233 = 14 \times 16 + 9 = E9H$

$14 = 0 \times 16 + 14 = 0EH = EH$

余りを下から上に並べると E9DH となる

図 2、10 進数から 16 進数への変換

・2 進数から 16 進数への変換

教科書 53 ページを参照。

・16 進数から 2 進数への変換

教科書 54 ページを参照。

・2 進数の四則演算

教科書 49 ページを参照。

・2 進数での負数の表現

-5 を 2 進数で表現する事を考える。また、ここでの議論においては、2 進数は全て 4 桁で表現するものとする。5=0101B であるから、数学的には-5=-0101B と表現できる。しかしながら、”-“という、第 3 の文

字(0 でも 1 でもない文字)を導入すると、2 進数を高い電圧と低い電圧の組み合わせで表現できなくなる。

そこで、最上位桁を正の値の場合は 0、負の値の場合は 1 にする方法(サインビット方式)もあるが、この方式によると､計算回路が複雑になるので､余り用いられない｡ちなみに､サインビット方式なら 5=0101B、-5=1101B と表現できる。

今日では、負の数を表現するのに、次に説明する 2 の補数表現を使うのが一般的である。

・2 の補数

ある 2 進数の 2 の補数は、次の様に求まる。

(1) 1 の後に 4 桁の 0 を並べて 2 進数を作る。(10000B) 4 桁ではなく、n 桁の 2 進数を扱っている場合は 1 の後に n 桁の 0 を並べて 2 進数を作る。
(2) (2)の手順で求めた 2 進数から、2 の補数を求めたい 2 進数を引く。例えば 5=0101B の 2 の補数を求めたいなら、10000B-0101B=1011B が答えになる。

10000B から 0101B を引くというのは、考えようで 0000B から 0101B を引く際に、5 桁目から 4 桁目に強制的に繰り下がるのと同じである。0(0000B)から 5(0101B)を引くので、1011B が-5 を表すというわけである。この様に負の数を 2 の補数を使って表現する方法を 2 の補数表現という。

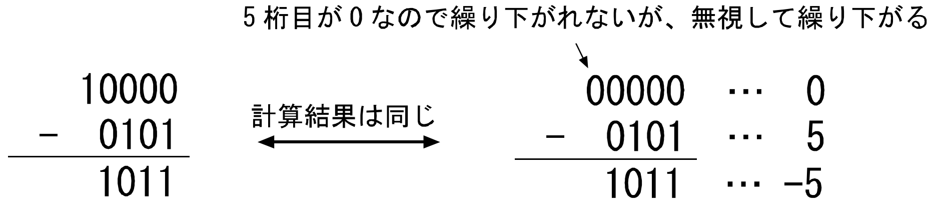

図 3、2 の補数の求め方

教科書の 50 ページでは、1 の補数に 1 を加えて 2 の補数を求める方法が紹介されているが、教科書の方法でもこのプリントの方法でも、同じ結果が得られる。

負の数を 2 の補数で表す事にすると、4 ビット(2 進数 4 桁)で-8 から 7 までの数が表現できる。(教科書 51 ページ表 3.2) 負の数の場合は、最上位ビットが 1 になる。0 または正の数の場合は最上位ビットが 0 となる。

・2 の補数表現と加算、減算

ある数の 2 の補数を求め、その結果の 2 の補数をさらに求めると、元の数に戻る。例えば、0101B の 2 の補数は 1011B であるが、1011B の 2 の補数は元の 0101B となる。0101B が 10 進数で 5、1011B が 10 進数で-5 を表す事を考えると、2 の補数を求めるという作業は、-1 を掛ける(符号を反転させる)のと同じ意味を持つ事が分かる。

2 の補数表現を使うと、0 または正の数を扱う通常の加算回路で、負の数も扱う事ができるようになる。次のページの図 4 に示すのは、0101B と 1101B を足した例である。

この性質を利用すると減算回路を用いずに、加算回路と 2 の補数を求める回路を組み合わせて引き算を行う事ができる。2 つの 2 進数 a と b の差 a-b を求めるには、a と、b の 2 の補数とを足せばよい。

$$
\begin{array}{r l r}
 & 0101 & \cdots \ 5 \\
+ & 1101 & \cdots \ 13 \\
\hline
 & 10010 & \cdots \ 18
\end{array}
\qquad\qquad
\begin{array}{r l r}
 & 0101 & \cdots \ 5 \\
+ & 1101 & \cdots \ -3 \\
\hline
 & 0010 & \cdots \ 2
\end{array}
$$

繰上げを無視する

(a)2 進数を 0 または正の数とみなした場合　　(b)2 進数を 2 の補数表現とみなした場合

図 4、4 ビットの加算回路による 2 進数の足し算

例えば、5-3 を求める場合は、5 は 0101B、3(0011B)の 2 の補数は 1101B であるから、図 4(b)と同じ計算で答えが 0010B(10 進数で表すと 2)と求める事ができる。