

■目次 (ほげが今回の範囲)

第1部 0と1だけでどうやって計算するの? ⇒	★1 p 進法による数の表現
第2部 コンピュータの気持ち	★2 論理回路と加算器
第3部 情報をどのように表現するか	★3 負の数の表現と減算の仕組み
第4部 コンピュータシステム	

「コンピュータはなんでも0と1で表して計算している」ということは多くの人が知っているだろう。でも実際のところ、内部では何が起きているのだろうか？第1部では、コンピュータと情報処理の世界への第1歩として、「整数を0と1で表して加算・減算する仕組み」を理解することを目標とする。この第2回の授業では、高校数学で学んだ2進法の話の復習し、0か1を入れると0か1が出てくる「論理回路」というものについて学ぶ。コンピュータの中核は、この論理回路によって構成されている。

★1 p 進法による数の表現

★1.1 ビット列 — 0と1のならば

コンピュータの内部では、情報を表現するのに2種類の値のみを用いる(☆1)。前回の授業では○と●を使っていたが、人間の見慣れた数の表現と対応させた方が分かりやすいので、一般的には0と1を用いる。このとき、0と1を並べたものを**ビット列**または**ビットパターン**という。

101010001

はビット列の例である。

0または1を合わせて n 個並べてできたビット列があったとき、この n のことを**ビット長**といい、単位**ビット (bit)** (☆2)をつけて数える。上の例のビット列のビット長は9ビットである。

☆1) それでどうやって(10進数の)数を表現したり計算したりするのかを学ぶのがこの授業の当面の目標。「ほな文字や音声、画像のようなんは？」ちう話はそのうち。

☆2) ビットという単位は、情報の量を測るために用いられる。いずれまた登場します。

★1.2 p 進法

ここでは、0以上の整数に限定して話をすすめる。負の数についてはいずれまた。

★1.2.1 10進法と2進法

■10進法 (☆3) 人間の世界で一般的なやり方では、10進法で位取り表示を用いる。これは、0から9までの10種類の記号を一行に並べるものである。例えば、

0	2	0	3	4
---	---	---	---	---

 という表記(☆4)では、左端が最上位の桁、右端が最下位の桁であり、

$$\boxed{0} \boxed{2} \boxed{0} \boxed{3} \boxed{4} = 0 \times 10^4 + 2 \times 10^3 + 0 \times 10^2 + 3 \times 10^1 + 4 \times 10^0$$

となる。つまり、

0	2	0	3	4
---	---	---	---	---

 と表される数は、 $10^4 = 10000$ が0個、 $10^3 = 1000$ が2つ、 $10^2 = 100$ が0個、 $10^1 = 10$ が3つ、 $10^0 = 1$ が4つでできている。

Q1. 箱の数が n 個、すなわち n 桁の場合、10進法で表せる最大の数はいくつか。また、全部で何通りの数が表せるか。

☆3) 英語では decimal numeral system または単に decimal という。10進数は decimal number.

☆4) 箱は分かりやすくするために付け加えたものである。また、一般的には最上位の0は省略して書かないことが多いが、ここでは省略せずに書いている。

■2進法 (☆5) 上記で10が出てきたところを2に置き換えると、2進法での数の表記を考えることができる。2進法では、0と1の2種類の記号を並べる。例えば (☆6)

$$\begin{array}{|c|c|c|c|c|} \hline 0 & 1 & 0 & 1 & 1 \\ \hline \end{array}_{(2)} = (0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0)_{(10)} \\ = 11_{(10)}$$

であるから、この数は10進法の $2^4 = 16$ が0個、 $2^3 = 8$ が1つ、 $2^2 = 4$ が0個、 $2^1 = 2$ が1つ、 $2^0 = 1$ が1つでできており、10進表記の11に等しい。

Q2. 箱の数が n 個、すなわち n ビット(n 桁)の場合、2進法で表せる最大の数はいくつか。また、全部で何通りの数が表せるか。

★1.2.2 p 進法

上記の話を一一般化すると、 p を2以上の整数として、 p 通りの記号を並べて数を表現する「 p 進法」を考えることができる。 p 進法は、

「 p を底(☆7)としてそのべき乗の数を基準に数を表す方法」

である(☆8)。5桁の p 進表記の数は

$$\begin{array}{|c|c|c|c|c|} \hline \square & \square & \square & \square & \square \\ \hline \end{array}_{(p)} = (\square \times p^4 + \square \times p^3 + \square \times p^2 + \square \times p^1 + \square \times p^0)_{(10)}$$

の右辺の10進表記の数に等しい。ここで、左辺の箱一つには p 通りの記号が入り、右辺の丸一つにはその記号が表す数を10進表記したものが入る。 $2 \leq p \leq 10$ の場合は、箱の中に10進数と同じ数記号(0, 1, ..., 9)を使えばよいが、 $p \geq 11$ の場合は記号が足りないので、工夫が必要となる(★1.2.4参照)。

★1.2.3 10進表記と2進表記の間の変換

■10進から2進へ 10進法で表された数を10で割って余りを記録することを繰り返すと、10進表記の各桁の数を取り出すことができる。では、2で割って余りを記録していくとどうなるだろう？

Q3. 10進法で表された数337を10で割って余りを求める計算を繰り返し、得られる余りを順に答えなさい。

Q4. 10進法で表された数337を2で割って余りを求める計算を繰り返し、得られる余りを順に答えなさい。

☆5) 英語では binary numeral system または単に binary.

☆6) 「 $101_{(p)}$ 」のような表記は、この101が p 進法で表された数であることを示している。文脈から明らかな場合は省略する。時給 $1000_{(2)}$ 円のバイトと $1000_{(8)}$ 円のバイトならどっちがいい？

☆7) この「底」は対数の話で出てくるのと同じもの。

☆8) 60秒で1分、60分で1時間とする時間の表現は、60進法表記そのものではない(60種類の記号を用いるわけではない)が、その考え方に基づくものといえる。

■2進から10進へ (☆9)

Q5. 2進法で表された数 101010001 に対して、最上位の桁から最下位の桁に向かって順に、 $x = 0$ から始めて

「現在の x の値を 2 倍して、現在の桁の数を加える」

という計算を繰り返すと、最終的な x はいくつになるか。

Q6. 2進法で表された数 101010001 に対して、最下位桁の値の 1 倍、その上位桁の 2 倍、その上位桁の 4 倍、その上位桁の 8 倍、…、の値を計算し、それらの和を求めなさい。

★1.2.4 16進法

2進法による数の表記は、人にあまりやさしくない。そこで、コンピュータに関わる世界では 16 進法を使うこともある (☆10)。ただし、0 から 9 までの記号では足りないので、10 進表記の 10 から 15 までを A から F までの英字で表すことが多い。例：

$$\boxed{\text{E}}\boxed{2}\boxed{9} = 14 \times 16^2 + 2 \times 16^1 + 9 \times 16^0 = 3625$$

Q7. ★1.2.3 で説明されたやり方を応用して 10 進表記の 3625 の 16 進表記を求め、上記に一致することを確かめなさい。

Q8. 10 進数の数を適当に考え、それを 16 進法で表しなさい。さらに、それを 10 進法に変換しなさい。正しい計算ができれば、当然元に戻るはずである。

☆9) このパラグラフでもそうですが、2進表記の数を扱うことの多いコンピュータの世界では、 $2^4 = 16, 2^8 = 256, 2^{10} = 1024, 2^{16} = 65536$ などの数が頻りに登場します。いちいち計算せずに済むように覚えておこう。

☆10) 英語では hexadecimal. $2^4 = 16$ だから、2進表記の 4 ビット毎に区切ったものをちょうど 1 桁で表すことになる。例えば 16 進数 E29 は、 $E = 1110, 2 = 0010, 9 = 1001$ だから、

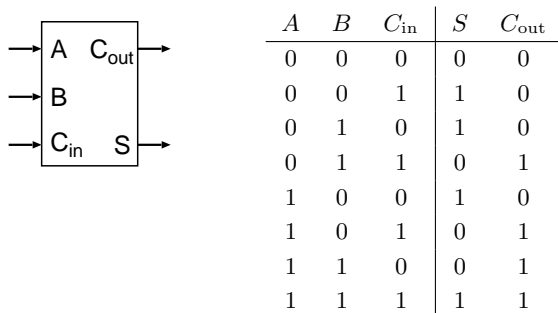
111000101001

となる。2進数より 16 進数の方がちょっとは人にやさしい…かな？

★2 論理回路と加算器

★2.1 論理回路と真理値表

右の図と表は、前回登場した、2つの2進数の加算を行う仕掛けに用いる部品と、その入力-出力の対応を表にまとめたものである。このように入出力が2種類の値で表される仕掛けのことを論理回路 (☆11) といい、その入出力をまとめた下のような表を真理値表 (☆12) という。



☆11) 何が「論理」なのかはいずれました。

☆12) 何が「真理」の値なのかはいずれました。

前回の授業で説明したように、上図の論理回路を複数組み合わせると、2進数で表された2つの数の和を求める論理回路を作ることができる。このことから想像できるように、コンピュータ（の中核をなしている部分）の正体は、非常に複雑な論理回路である。

上図の論理回路は2つの数の和を求める論理回路の部品となるが、この論理回路自身も、もっと単純な論理回路を部品として組み上げることができる。そのことを実際に示すのは次回にまわして、今回はもう少し簡単な例で感じをつかんでみよう。

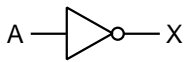
NOT ゲートの真理値表 AND ゲートの真理値表 OR ゲートの真理値表

A	X
0	1
1	0

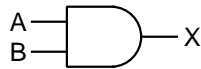
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

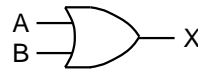
NOT ゲートの図記号



AND ゲートの図記号



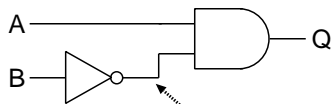
OR ゲートの図記号



上記の3つの表と3つの図は、論理回路を構成する部品となる**論理ゲート**のうち、最も単純な **NOT ゲート**、**AND ゲート**、**OR ゲート**の3つの真理値表と論理回路の図を描くための図記号を示している(☆13)。ここでは、 A や B が入力、 X が出力を表している。例えば、 A に1を、 B に0を入力されたANDゲートの出力 X は0となる。

☆13) NOT, AND, OR などの意味についてはいづれまた。

Q9. 下図は、 A, B にそれぞれ0または1の値が入力されると、0または1の値 Q を出力する論理回路の図である(矢印は説明のために描いたもので、回路図の一部ではない)。下の表を埋めることで、この回路の真理値表を求めなさい。ただし、表の P は、図の矢印の部分の値 (NOTゲートの出力) を表すものとする。



A	B	P	Q
0	0		
0	1		
1	0		
1	1		