



前回に続き、2進数がコンピューターにとって「都合のよい」理由の一つである「計算が簡単にできる」についてお話しします。

今回は1桁の加算を例に、0と1のみを扱う論理演算に置き換えられることを説明しました。実際の計算では、全ての桁で桁上りを考慮しなければならないので、これを踏まえたA+Bの2進数での各桁の計算は、表1のとおり行います。

一つ下(右)の桁で桁上がりがあった場合を考慮した「Cの値」を設け、これを含めて計算します。表1で確認できるとおり、「A+B+Cの値」は、次の論理演算により求められます。

「Aの値とBの値のXOR演算 *による結果」と「Cの値」のXOR演算の結果

また、「桁上りの有無」は、A、B、Cのうち、1であるものが2つ以上ある場合と定義できます。これを論理演算で表すと、次のようになります。

$(A=1 \text{ AND } B=1) \text{ OR } (A=1 \text{ AND } C=1) \text{ OR } (B=1 \text{ AND } C=1)$

これらの論理演算を桁数分(8ビットなら8桁分)繰り返し、計算が完了します。ここで得られた値は2進数のままなので、必要に応じて10進数に変換します。

表1 2進数の加算結果

Aの値	Bの値	Cの値 (一つ下(右)の桁の計算における桁上りの有無。ある場合1)	A+B+Cの値 (桁上がりがある場合は末尾1桁の値)	桁上りの有無 (ある場合に1。次(左)の桁の計算の際にCの値として使用する)
1	1	1	1	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	1	0	1
0	1	0	1	0
0	0	1	1	0
0	0	0	0	0

このように、2進数を扱うことで論理演算が可能となり、「計算が簡単にできる」ようになっています。

ここまで、加算を例に説明しましたが、減算の場合はNOT演算 *により求めた「補数」を用いて負の値を表し、これを加算します。また、乗算は加算の繰り返し、除算は減算できる回数を数えることで実現しています。

次回は、インターネットについてです。

* XOR演算、NOT演算については、前回(広報ひみ9月号22ページ)の表1を参照してください。