演習 1

R = $\dot{1}\ \dot{0}\ \dot{0}\ \dot{0}\ \dot{0}\ \dot{1}\ \dot{0}\ \dot{1}$

7 6 5 4 3 2 1 0

位取り記数法によって

$V((R)_2)=1*2^7+1*2^2+1*2^0=133$

$V((R)_4) = 1*4^7+1*4^2+1*4^0 = 16401$

$V((R)_{10}) = 1*10^7 + 1*10^2 + 1*10^0 = 10000101$

$V((R)_B)=2^7+2^2+2^0=133$

$V((R)_{SM}) = 2^2 + 2^0 = 5$

$V((R)_{1C}) = -(2^6+2^5+2^4+2^3+2^1) = -122$

$V((R)_{2C})=-(2^2+2^1)=-6$

演習 2

2進数の倍数原則:

2進数が1ビット左にシフトされるたびに、元の数の2倍に拡張されます。

(以下は2進数を x とする)
1. ある2進数を3倍する演算:xの2進数の末尾に0を追加してから + x

例: 12(10) = 1100(2).　1100の3倍:

11000 + 1100 =100100.　　100100(2)

検証: 100100(2) = 36(10).　12*3=36

2. ある2進数を 4 倍する演算:x を左に 2 ビットシフトする

3. ある2進数を 5 倍する演算:x を左に 2 ビットシフトしてから +x

4. ある2進数を 7 倍する演算:x を左に 3 ビットシフトしてから-x

5. ある2進数を 8 倍する演算:x を左に 3 ビットシフトする

6. ある2進数を 10 倍する演算:x を左に 3 ビットシフトしてから +x+x

7. ある2進数を 20 倍する演算:x を左に 4 ビットシフトした数字 +x を左に 2 ビットシフ トした数字

演習 3

10進数を2進数、8進数、16進数に変換します

10進整数からN進整数への変換は、「Nで割り、余りを取り、逆の順序で並べる」方法を採用しています。具体的なアプローチは次のとおりです。

Nを除数として使用し、10進整数をNで除算して、商と剰余を取得します。

余りを保持し、商でNで除算し続け、新しい商と余りを取得します。

余りは引き続き保持され、商は引き続きNで除算され、新しい商と余りが取得されます。

..。

これが繰り返され、余りが保持されるたびに、商がゼロになるまで商がNで除算されます。

最初に得られた余りをN基数の下位桁とし、後で得られた余りをN基数の上位桁とし、順番に並べてN基数を求めます。

演習 4

簡単に言えば、符号拡張は下位桁から上位桁への変換です。そして、桁数が要件を満たすまで、下位桁の左側に下位桁の符号ビットを追加するだけで済みます。

1、Iをkビットの正の2進数、m> k、aiとします。0または1です。

1.

$$
\begin{aligned}
I &= \sum_0^k a_i 2^i \\
&= 0 + \sum_0^k a_i 2^i \\
&= \sum_(k+1)^m 0 * 2^i + \sum_0^k a_i 2^i
\end{aligned}
$$

Iが正の数の場合、必要な桁数に達するまで、その左側に0を追加するだけであることがわかります。

$$
\begin{aligned}
&-(\sum_(k+1)^m 0 * 2^i + I) \\
&= -(\sum_(k+1)^m 0 * 2^i + \sum_0^k a_i 2^i) \\
&= \sum_(k+1)^m (1-0) * 2^i + \sum_0^k (1-a_i) 2^i + 1 \\
&= \sum_(k+1)^m 1 * 2^i + (-\sum_0^k a_i 2^i) \\
&= \sum_(k+1)^m 1 * 2^i + (-I)
\end{aligned}
$$

2.便宜上、 ここに高
い位置から低い位置への証明があります。

逆は次のとおりです。

$$
\sum_(k+1)^m 1 * 2^i + (-I) = -(\sum_(k+1)^m 0 * 2^i + I)
$$

明らかに、下位ビットが負の数の場合、桁数が要件を満たすまで、左側に0を追加するだけで済みます。

以上です。

演習 5

例:

2の補数で+10を表現するのに必要な最小のビット数は4
1010
これを6ビットの+10にするには
00 0111
と最上位の0を2個上側につければいいです。
これが6ビットでの+10です。