はじめに

- 今回から4回に渡って、命令セットアーキテクチャについて講義します。
- マシン命令について解説する前に、まず、今回は、そのマシン命令で処理するデータの表現方法についてみていくことにします。

- 教科書の対応範囲は、以下の通りです。
  - 2.2節始め～2.2.1項（p.35～38）
  - 3章始め～3.2.1項（p.73～87）
  - 6章始め～6.1.1項（p.164～171）

コンピュータ内部での数の表現形式

- *r*進数
  - 人間:　日常的には、10進数を用いています。
  - コンピュータ:　2進数で表現します。
    - 0と1の2値で動作制御しますので、1桁を0か1かで表現する2進数を用いるのが自然です。
- データの種類…以下の2(3)種のみ
  - 固定小数点数
  - 浮動小数点数
  - (文字コード)

これは個々の文字に対応する番号ですが、その番号を整数とみなすことにすると、固定小数点数として扱うことができます。

- データの単位
  - ビット（bit）……2進数1桁
  - バイト（byte）……1バイト＝8ビット
  - ワード（word）……コンピュータの内部表現における基本単位。コンピュータによって異なる。
    - 例えば、32ビットのプロセッサ（命令セット）の場合は、1ワード=32ビット（4B）。
    - しかし、古くからある命令セットの場合は、アセンブリ言語での混乱を避ける目的で、古いままの定義が用いられることもあります。
      - x86アーキテクチャでは、64ビットの現在でも、1ワードは16ビットのままです。

固定小数点数と浮動小数点数

- 固定小数点数
  - 2020や3.14のように表現。
    - 正確には、小数点の位置を固定（つまり、整数部と小数部の桁数を固定）して表現する方法です。
  - 今回の講義で詳しく扱います。
- 浮動小数点数
  - 6.02×10^{23}のように表現。
  - 限られたビット数で非常に大きな数や非常に小さな数を表現できます。
  - 今回の講義では扱いません。本科目の最後の方の授業で扱います。

プログラミング言語における基本データ型

- プログラミング言語では多くの基本データ型が定義されているのに、マシン語のレベルではなぜ2種類しかないのでしょうか？

- 実は、プログラミング言語における基本データ型は、データの種類とデータサイズの組み合わせで定義されています。
  - 例えばC言語の場合
    - char型: 8ビットの固定小数点数
    - short型: 16ビットの固定小数点数
    - int型: 32ビットの固定小数点数
    - float型: 32ビットの浮動小数点数
    - double型: 64ビットの浮動小数点数

    など

*r*進数表現

- 基数:
  - *r*進数の*r*のこと。
  - 1桁を表現する記号の数(種類)。

    - 2進数: 0と1の2種
    - 8進数: 0, 1, 2, 3, 4, 5, 6, 7 の8種
    - 10進数: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 の10種
    - 16進数: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
      a(10), b(11), c(12), d(13), e(14), f(15)
      の16種 (a～f は大文字でもよい)

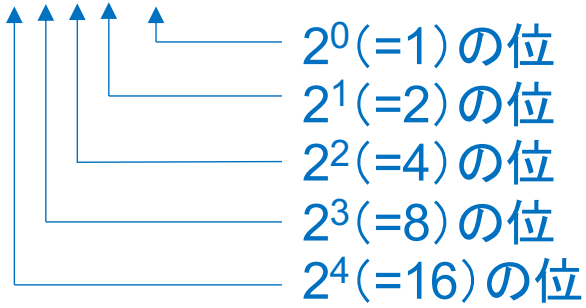
r進数から10進数への基数変換

- r進数表現の整数の各桁の記号を a_i とすると、

$$(a_{n-1}a_{n-2}\cdots a_1a_0)_r = (\sum_{i=0}^{n-1} a_i \times r^i)_{10}$$

- 2進数から10進数への変換例
  - 上記の式にしたがって計算すれば、

$(11101)_2=1\times2^4+1\times2^3+1\times2^2+0\times2^1+1\times2^0=(29)_{10}$



10進数からr進数への基数変換

- 下式より、r で割れば、最下位桁 a_0 が余りとなる。

$$\sum_{i=0}^{n-1} a_i \times r^i = r \times (\sum_{i=1}^{n-1} a_i \times r^{i-1}) + a_0$$

- 10進数から2進数への変換例

53÷2=26 … 1 ←最下位桁
26÷2=13 … 0
13÷2= 6 … 1
6÷2=3 … 0
3÷2=1 … 1
1÷2=0 … 1 ←最上位桁

よって、 $(53)_{10}=(110101)_2$

小数の基数変換

- *r*進数表現の数の各桁の記号を a_i とすると、

$$(a_{n-1}\cdots a_0.a_{-1}\cdots a_{-m})_r = (\sum_{i=-m}^{n-1} a_i \times r^i)_{10}$$

小数点

- 2進数から10進数への変換例
  - 上記の式にしたがって計算すれば、

$(1.101)_2=1\times2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}=(1.625)_{10}$

2^{-3}(=1/8)の位
2^{-2}(=1/4)の位
2^{-1}(=1/2)の位
2^0(=1)の位

- 10進数から*r*進数への基数変換
  - 整数部と小数部に分けて変換
  - 小数部に*r*をかけると、最上位桁（小数点第1位の桁）が整数部に上がってくる。

整数部

$$(0.a_{-1}\cdots a_{-m})_r \times r = (a_{-1}.a_{-2}\cdots a_{-m})_r$$

- 10進数から2進数への変換例

$(5.40625)_{10}=$
$=(5)_{10}+(0.40625)_{10}$
$=(101)_2+(0.01101)_2$
$=(101.01101)_2$

最上位桁

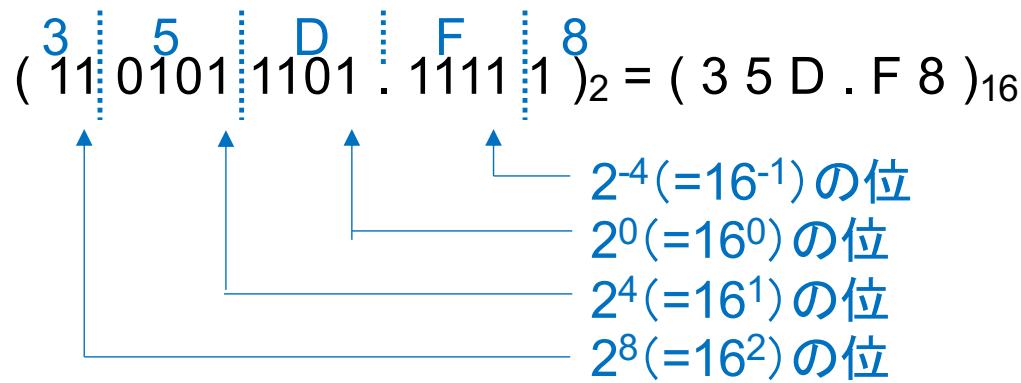
$0.40625\times2=0.8125$
$0.8125\times2=1.625$
$0.625\times2=1.25$
$0.25\times2=0.5$
$0.5\times2=1.0$
$0.0\times2=0$

最下位桁

8進数と16進数

- 2進表記では桁数が多くなって我々人間にとって扱うのが厄介です。そこで少ない桁数で表現するために8進数や16進数を使います。
  - あくまで、我々人間が簡略表記法として用いているだけです。
    - コンピュータは2進数（0と1からなる系列）しか扱いません。

  注） 数値以外の情報やアドレスにも使用します。
- 16進表記の場合は、2進数を4桁ずつ区切り、それを16進数の1桁と見なします（8進表記の場合は3桁ずつ）。
  - 例：（数値の場合は小数点が区切りの基準）



2進数の四則演算

- 10進数の場合と同じように計算します。

```
   11010
＋） 111001
-------
  1010011
```

```
  1010011
－） 111001
-------
   11010
```

```
    1101
×）  101
-------
    1101
   0000
  1101
-------
 1000001
```

```
        101
     -------
1101 ） 1000001
        1101
     -------
        1101
        1101
     -------
           0
```

実際のコンピュータ内部での表現は？

- まず、表現できる桁数は固定（有限）、つまり制限があります。
  - メモリ素子数や配線数を、コンピュータハードウェアの製造後に変更することはできません。
- 小数点をどう扱うか、決めておく必要があります。
- 負の数をどのように表現するのかも、決めておく必要があります。