コンピュータアーキテクチャ レポート課題

17ec084 平田智剛

1．講義資料 No.2：~~P.30～P.31~~(→P.25～P.26) 固定小数点の計算

(a)	Unsigned	Binary	$Result_{(2)}$	$Result_{(10)}$	CA	OV	isCorrect
Ex.	0.25-0.5	0001-0010	1111	3.75	1	0	False
(1)	3-2.5	1100-1010	0010	0.5	0	0	True
(2)	0.5-2.5	0010-1010	1000	2	1	1	False
(3)	3.75-3.25	1111-1101	0010	0.5	0	0	True
(4)	1-3.5	0100-1110	0110	1.5	1	0	False
(5)	1.25+3.25	0101+1101	0010	0.5	1	0	False
(6)	3.5-3.75	1110-1111	1111	3.75	1	0	False
(7)	1.75+0.75	0111+0011	1010	2.5	0	1	True
(8)	3+3.75	1100+1111	1011	2.75	1	0	False
(9)	0.75+2.75	0011+1011	1110	3.5	0	0	True
(10)	1.5+3.75	0110+1111	0101	1.25	1	0	False

(b)	Signed	Binary	$Result_{(2)}$	$Result_{(10)}$	CA	OV	isCorrect
Ex.	0.25-0.5	0001-0010	1111	-0.25	1	0	True
(1)	1.5+(-0.25)	0110+1111	0101	1.25	1	0	True
(2)	0.75+(-1.25)	0011+1011	1110	-0.5	0	0	True
(3)	-0.5+(-0.75)	1110+1101	1011	-1.25	1	0	True
(4)	1.75+0.75	0111+0011	1010	-1.5	0	1	False
(5)	-0.5-(-0.25)	1110-1111	1111	-0.25	1	0	True
(6)	-1.75+(-0.75)	1001+1101	0110	1.5	1	1	False
(7)	1-(-0.5)	0100-1110	0110	1.5	1	0	True
(8)	-0.25-(-0.75)	1111-1101	0010	0.5	0	0	True
(9)	0.5-(-1.5)	0010-1010	1000	-2	1	1	False
(10)	-1-(-1.5)	1100-1010	0010	0.5	0	0	True

２．１の問題を，配布した電卓で動作確認すること．
(詳細が分かり次第再提出させてください)

３．講義資料 No.2：P.40 浮動小数点数値の範囲を 16 進数と 10 進数(概数)で示すこと．

答え

・単精度浮動小数点数値の(特殊表現を除く)範囲は
-2.56×10^{38}～2.56×10^{38}となる。
(0xFF7FFFFF～0x7F7FFFFF)

より厳密にいうならば-3.40282×10^{38}～3.40282×10^{38}で、精度は 6 桁である。

・倍精度浮動小数点数値の(特殊表現を除く)範囲は
-1.6×10^{307}～1.6×10^{307}となる。
(0xFFEFFFFFFFFFFFFF～0x7FEFFFFFFFFFFFFF)

より厳密にいうならば$-1.79769313486232 \times 10^{307}$～$1.79769313486232 \times 10^{307}$で、精度は

15 桁である。
・半精度浮動小数点数値の(特殊表現を除く)範囲は
-65536～65536 となる。
(0x7BFF～0xFBFF)
より厳密にいうならば-6.55×10^4～6.55×10^4 で、精度は 3 桁である。

導出

単精度:符号 1bit、指数部 8bit、仮数部 23bit

指数部 11111111 は「特殊表現」となり、±∞または NaN を意味する。

したがって、最大となる指数部は 11111110 であり、127 を意味する。(11111110=254 であるが、ここから 127 を引き算する)

最大となる仮数部は 11111111111111111111111 である。

したがって、単精度浮動小数点数値の最大値を表現するものは

0 11111110 11111111111111111111111

つまり

0111 1111 0111 1111 1111 1111 1111 1111

であり、16 進数で表現すると

0x7F7FFFFF

となる。この数値はおよそ2.56×10^{38}を意味する。

(証明)

$$+1.11111111111111111111111_{(2)} \times 10_{(2)}^{11111110-01111111_{(2)}}$$

$$\fallingdotseq +10_{(2)} \times 10_{(2)}^{11111110-01111111_{(2)}}$$

$$=2 \times 2^{127}$$

$$=2^{128}$$

$$\fallingdotseq (2^{10})^{12} \times 2^{8}$$

$$=2.56 \times 10^{38}$$

(証明終わり)

最小となるものについては、絶対値が同じまま(＝指数部と仮数部をそのまま※1)符号だけ入れ替えればよい。

したがって、単精度浮動小数点数値の最小値を表現するものは

1 11111110 11111111111111111111111

であり、16 進数で表現すると

0xFF7FFFFF

となる。この数値はおよそ-2.56×10^{38}を意味する。

したがって、単精度浮動小数点数値の(特殊表現を除く)範囲は

-2.56×10^{38}～2.56×10^{38}となる。

(0xFF7FFFFF～0x7F7FFFFF)

より厳密にいうならば-3.40282×10^{38}～3.40282×10^{38}で、精度は 6 桁である※2

倍精度:符号 1bit、指数部 11bit、仮数部 52bit

最大値

0 11111111110 1111111111111111111111111111111111111111111111111111

=0111 1111 1110 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111

=0x7FEFFFFFFFFFFFFF

$\fallingdotseq +10_{(2)} \times 10_{(2)}^{11111111110-01111111111_{(2)}}$

$=2 \times 2^{1023}$

$=2^{1024}$

$=(2^{10})^{102} \times 2^4$

$\fallingdotseq (10^3)^{102} \times 2^4$

$=1.6 \times 10^{307}$

最小値

0xFFEFFFFFFFFFFFFF

$=-1.6 \times 10^{307}$

したがって、倍精度浮動小数点数値の(特殊表現を除く)範囲は

-1.6×10^{307}～1.6×10^{307}となる。

(0xFFEFFFFFFFFFFFFF～0x7FEFFFFFFFFFFFFF)

より厳密にいうならば$-1.79769313486232 \times 10^{307}$～$1.79769313486232 \times 10^{307}$で、精度は 15 桁である※2

半精度:符号 1bit、指数部 5bit、仮数部 10bit

最大値

0 11110 1111111111

=0111 1011 1111 1111

=0x7BFF

$\fallingdotseq +10_{(2)} \times 10_{(2)}^{11110-01111_{(2)}}$

$=2 \times 2^{15}$

$=2^{16}$

$=2^{10} \times 2^{6}$

=65536

最小値

0xFBFF

$\fallingdotseq -65536$

したがって、半精度浮動小数点数値の(特殊表現を除く)範囲は

$-65536 \sim 65536$となる。

(0x7BFF～0xFBFF)

より厳密にいうならば$-6.55 \times 10^4 \sim 6.55 \times 10^4$で、精度は 3 桁である※2

※1:浮動小数点数値に関しては、符号ビットは「-1 を掛け算するか否か」の意味しか持たず、整数の場合のように基数の補数を考える必要がない。
※2:このことは、単精度については次のように求めることができる。
精度・・・
精度は有効桁数で決まるので、仮数部の精度そのものとなる。

Δ仮数部

$= 1.00000000000000000000001_{(2)} - 1.00000000000000000000000_{(2)}$
$= 0.00000000000000000000001_{(2)}$
$= 2^{-23}$
ここで、2^{-23}における「23」の意味は「2 進数で小数第 23 位まで有効」ということなので、
$2^{-23} = 10^{-n}$とするなら、「10 進数で小数第 n 桁まで有効」とわかる。
両辺の常用対数をとると、
$\log_{10} 2^{-23} = -n$
$n = 23 \log_{10} 2$
ここで$\log_{10} 2 = 0.30$より、
$n = 6.9$
したがって、精度は 6 ケタとなる。
仮数・・・
$1.11111111111111111111111_{(2)}$
$= 10_{(2)} - 2^{-23}$
なので、$\left(10_{(2)} - 2^{-23}\right) \times 2^{127}$を 6 ケタまで求めればよい。
しかし、$2^{-23} (\fallingdotseq 10^{-6.9}) < 10^{-6}$なので、
2^{128}を 6 ケタまで求めるのでよいだろう。
まず$\log_{10} 2^{128} = 128 \times 0.30 = 38(.531839445)$の小数部分は(0.531839445)なので、
（ここは電卓必要）

$10^{0.531839445} \fallingdotseq$ 仮数部 といえる※3

よって、仮数部は 3.40282
倍精度や半精度でも同様に求められる。

倍精度の場合

精度

$2^{-52} = 10^{-n}$

$n = 52\log_{10} 2 = 15.6$

よって、精度 15 ケタ

仮数

$10^{(\log_{10} 2^{1024})\text{の小数部分}(=0.2547155599167439)} \fallingdotseq$ 仮数部

よって、仮数部は 1.79769313486232

半精度の場合

精度

$2^{-10} = 10^{-n}$

$n = 10\log_{10} 2 = 3.01$

よって、精度 3 ケタ

仮数

$10^{\left(\log_{10}\left((2-2^{-10})\times 2^{15}\right)\right)\text{の小数部分}(=0.81626782098)} \fallingdotseq$ 仮数部

よって、仮数部は 6.55

※3:

$\text{仮数部} \times 10^{38} \leq 2^{128} < \left(\text{仮数部} + 1\right) \times 10^{38}$

$\left(\log_{10} \text{仮数部}\right) + 38 \leq 38.531839445 < \left(\log_{10}\left(\text{仮数部} + 1\right)\right) + 38$

$\left(\log_{10} \text{仮数部}\right) \fallingdotseq 0.531839445$

４．講義資料 No.2：~~P.42～P.44~~(→P.37～P.39)を解くこと．

問題[1]

(6bit).(6bit)という 12bit の固定小数点数値を考える。MSB は符号 bit とし、この bit が 1 の時は負数とし、その絶対値は 2 の補数により求められるものとする。このような固定小数点で表記できる最大値、最小値、最小絶対値を 2 進数及び 10 進数で求めよ。特殊表現除く。

答え

	2 進表記	10 進表記
最大値	011111111111	31.984375
最小値	100000000000	-32
最小絶対値	000000000001	0.015625

導出

最大値は符号 bit が 0 で他が全て 1 であるものだから、$011111111111_{(2)}$である。
これを 10 進数で表現しよう。
011111.111111
=100000.000000(unsigned)-000000.000001
$=2^5-2^{-6}$
$=32-(2^{-3})^2$
$=32-(0.125)^2$
$=32-0.015625$
$=31.984375$
最小値は負数 $100000000000_{(2)}$である。基数 2 の補数をとれば絶対値が求められる。
筆算で表すと、
1000000000000
-100000000000
==========
100000000000

つまり 64-32=32
負号をつけて、-32 となる。
最小絶対値については説明するまでもない。

問題[2]と答え

[1]と同じ固定小数点数値で、以下の 2 進数を 10 進数に、10 進数を 2 進数に変換せよ。ただし、正確に当てはまらない場合は、一番近い数値を用いること。

(1) 010101001000
010101.001000
=16+4+1+0.125
=21.125

(2) 110100110000
まず、unsigned で考える。
110100.110000
=32+16+4+0.5+0.25
=52.75
実際は signed で、しかも負号 bit が 1 なので、基数 2 の補数をとって、負号を付ける。
-(64-52.75)
=-11.25
(3) 37.375
=32+4+1+0.25+0.125
=100101.011000
よって 100101011000
但し、今回は signed なので、MSB は符号 bit である。したがって、
37.375 は-(64-37.375)=-26.625 とみなされる。
(4) 0.3
=0.25+0.05
=000000.010000+0.05
=000000.010000+.00001+0.01875
=000000.010000+.00001+.000001+0.003125
≒000000.010011
よって 000000010011

(5) -1.41421356
これは unsigned で考えたときの
64-1.41421356=62.58578644 に等しい
=111110.100000+0.08578644
=111110.100000+.000100+0.02328644
=111110.100000+.000100+.000001+0.00766144
≒111110.100101
よって 111110100101

問題[3]

符号 1bit、指数部は 4bit、仮数部は 7bit とし、その他は IEEE 754 の言うとおりにする 12bit 浮動小数点数値を考える。また非正規化数値は扱わないものとする。
この浮動小数点において、0 の値、正の無限大、負の無限小はどのように表されるかを 2 進数で示せ。

答え

0 は X00000000000
∞は 011110000000
-∞は 111110000000
但し X は 0 でも 1 でもよいという意味

問題[4]

[3]の浮動小数点で表すことのできる数値の範囲を 10 進数で、理由とともに示せ。

答え

-255～255

導出

最大値

0 1110 1111111

=0111 0111 1111

=0x77F

$=+1.1111111_{(2)} \times 10_{(2)}^{1110-0111_{(2)}}$

$=10-0.0000001_{(2)} \times 2^{7}$

$=1.9921875 \times 128$（または分配法則を適用して$2^{8}-2^{0}$）

$=255$

最小値

符号を入れ替えて-255

問題[5]

[3]の浮動小数点で表すことのできる数値の 0 を除く絶対値の範囲を 2 進数で、理由とともに示せ。

答え

0000 1000 0000～0111 0111 1111

導出

最大値
0 1110 1111111
=0111 0111 1111
(絶対値は結局正で表すので符号 bit は 0、最大値なので指数部は最大である 1110、仮数部も 1111111)
最小絶対値
0 0001 0000000
=0000 1000 0000
(符号 bit は 0、最小値なので指数部は最小である 0001、仮数部も 0000000)

問題[6]と答え

[3]の固定小数点数値で、以下の 2 進数を 10 進数に、10 進数を 2 進数に変換せよ。ただし、正確に当てはまらない場合は、 0 捨 1 入を用いること。

(1) 010101001000

$=+1.1001000_{(2)} \times 10_{(2)}^{1010-0111_{(2)}}$

$=1.1001000_{(2)} \times 10_{(2)}^{3}$

$=1100.1000_{(2)}$

$=12.5$

(2) 110100110000

$=-1.0110000_{(2)} \times 10_{(2)}^{1010-0111_{(2)}}$

$=-1.0110000_{(2)} \times 10_{(2)}^{3}$

$=-1011.0000_{(2)}$

$=-11$

(3) 37.375

まず普通に 2 進数の小数点で表すと([2](3)より)

$100101.011000_{(2)}$

これを変形すればいい。

$= 1.00101011000_{(2)} \times 10_{(2)}^{5}$

$= +1.00101011000_{(2)} \times 10_{(2)}^{1100-0111_{(2)}}$

0 捨 1 入すると

= 011000010110

(4) 0.3

[2](4)より

$000000.010011\ldots_{(2)}$

$= 1.0011\ldots_{(2)} \times 10_{(2)}^{-2}$

$= 1.0011\ldots_{(2)} \times 10_{(2)}^{0101-0111_{(2)}}$

$= 1.0011\ldots_{(2)} \times 10_{(2)}^{0101-0111_{(2)}}$

$= 001010011XXX$（X は不明な bit）

つまり、後 4 ケタ必要。(0 捨 1 入するため 1bit 余分に)

=000000.010011+0.003125

=000000.010011+.000000001+0.001171875

=000000.010011+.000000001+.0000000001+0.0001953125

=000000.0100110011+0.0001953125

0 捨 1 入すると

=000000.010011010

よって

001010011010

(5) -1.41421356

浮動小数点では、負数の場合も基数の補数をとったりしない。

したがって、まず、1.41421356 を普通の 2 進数で表す。

1.41421356

$= 1 + 0.41421356$

$= 1 + .01 + 0.16421356$

$= 1 + .01 + .001 + 0.03921356$

$= 1 + .01 + .001 + .00001 + 0.00796356$

$= 1 + .01 + .001 + .00001 + .0000001 + 0.00015106$

$= 1 + .01 + .001 + .00001 + .0000001 + .00000000 + 0.00015106$

$= 1.01101010 + 0.00015106$

0 捨 1 入すると

$= 1.0110101$

よって仮数部は 0110101

符号部は負数だから 1

指数部は $2^{eeee-0111}$ が 2^0 になるように、0111 とする。

したがって 101110110101

５．π＝3.14159 を半精度浮動小数点で示し，誤差を求めよ

答え

0100001001001000 と表される。
誤差 0.000965(浮動小数点に変換することで 0.000965 だけ小さくなった)

導出

半精度浮動小数点の精度は、3. で求めた通り、3 ケタ以上 4 ケタ未満。
だからまず 3.141 を普通の 2 進数の小数点で表す。仮数部は 10bit であり、先頭の 1 は決まっているので、実質的に 11bit を表現する。
整数部分 3 は 2bit であるから、小数部分を 9bit で表現すればよい(つまり小数第 10bit 目を 0 捨 1 入)。
3.141
$=11_{(2)}+0.141$
$=11+.001+0.016$
$=11+.001+.000001+0.000375$
$=11+.001+.000001+.000000000001+0.00013085937$
0 捨 1 入すると、
$=11.001001000(+0.000375)$
$=1.1001001000\times10_{(2)}{}^{1}(+0.000375)$
よって、仮数部は 1001001000 であり、指数部は $2^{eeeee-01111}=2^{1}$ となるように 10000、符号部は正なので 0
よって、0100001001001000 となる。
これは 3.141 より 0.000375 だけ小さいので
3.14159 より 0.000375+0.00059=0.000965 だけ小さい。