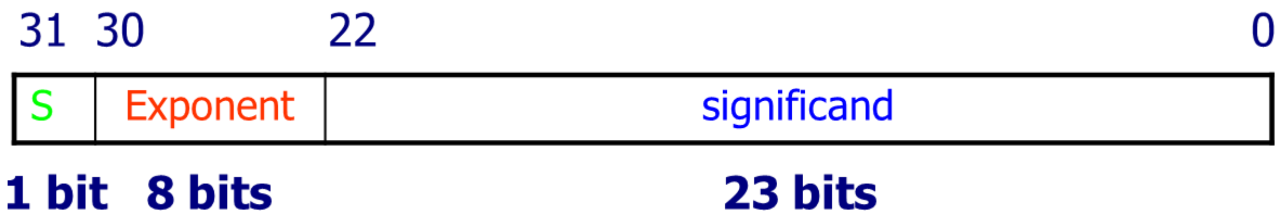
TUTORIAL 4 FLOATING POINT NUMBER REPRESENTATION AND CHARACTER

Overview

- **We will review the following concept in this tutorial:**
- **Representation of real numbers, very large/small numbers**
  - IEEE 754 Single precision floating point
  - IEEE 754 Double precision floating point
- **Representation of English letters**
  - ASCII encoding

The IEEE 754 single precision floating point

- The IEEE 754 standard uses 32 bits to represent single precision floating point numbers.
- S: sign bit (0 positive, 1 negative)
- Exponent: 8-bit field, bias = 127
- Significand: 23-bit field, implicit 1
- No 1's nor 2's complement for negative numbers.
  - No 1's nor 2's complement in Exponent and Significand.
  - The only difference between positive and negative numbers of the same magnitude is the sign-bit.



Decimal to IEEE754 single precision

- Convert -1541.625$_{(10)}$ to the single precision floating point format

Solution:

Scientific Notation:

$-1541.625_{(10)} = -\ 11000000101.101_{(2)} \times 2^{0}$

Normalized scientific notation:

$-1541.625_{(10)} = -\ 1.1000000101101_{(2)} \times 2^{10}$

Sign = 1 (negative), exponent = $10_{(10)}$

Single precision floating point format:

S = 1, Significand = 10000001011010...00 (23 bits)

Biased exponent = 10 + 127 = 10001001

=> 1 10001001 10000001011010000000000

IEEE754 double precision

- The IEEE 754 standard uses 64 bits to represent double precision floating point numbers.
- S: sign bit (0 positive, 1 negative)
- Exponent: 11-bit field, bias = 1023
- Significand: 52-bit field, implicit 1

63	62	51 … 0
S	Exponent	significand
1 bit	**11 bits**	**52 bits**

Decimal to IEEE754 double precision

- Convert $-1541.625_{(10)}$ to the double precision floating point format

Solution:

Scientific Notation:

$-1541.625_{(10)} = -\ 11000000101.101_{(2)} \times 2^0$

Normalized scientific notation:

$-1541.625_{(10)} = -\ 1.1000000101101_{(2)} \times 2^{10}$

Sign = 1 (negative), exponent = $10_{(10)}$

Double precision floating point format:

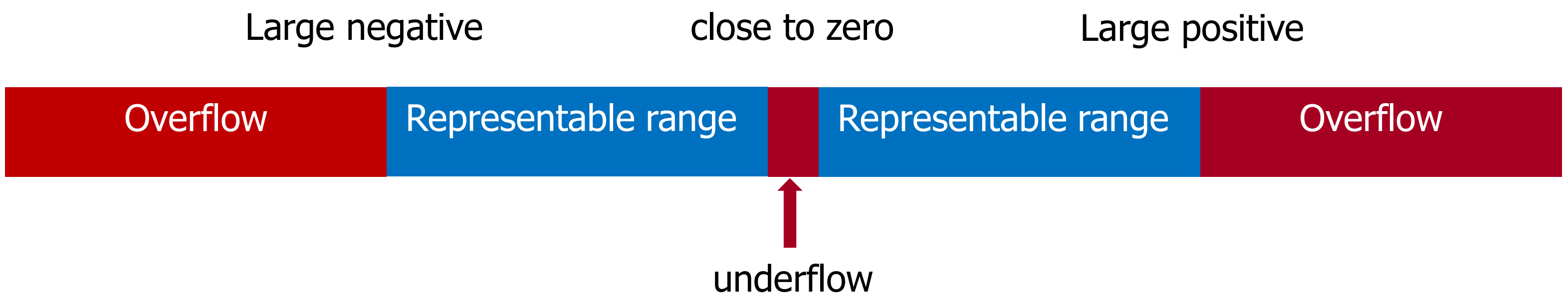
S = 1, Significand = 1000000101101000...00 (52 bits)

Biased exponent = 10 + 1023 = 10000001001

=> 1 10000001001 1000000101101000...00 (64 bits in total)

Overflow and underflow in floating point

- **Overflow (floating-point)**
  - A positive exponent becomes too large to fit in the exponent field
- **Underflow (floating-point)**
  - A negative exponent becomes too large to fit in the exponent field



IEEE754 special cases, denormalized cases

- **Single precision**

Significand \ Exponent	0	1 - 254	255
0	0	$(-1)^S \times (1.F) \times (2)^{E-127}$	$(-1)^S \times (\infty)$
$\neq 0$	$(-1)^S \times (0.F) \times (2)^{-126}$		non-numbers e.g. 0/0 , $\sqrt{-1}$

- **Double precision**

Significand \ Exponent	0	1 - 2046	2047
0	0	$(-1)^S \times (1.F) \times (2)^{E-1023}$	$(-1)^S \times (\infty)$
$\neq 0$	$(-1)^S \times (0.F) \times (2)^{-1022}$		non-numbers e.g. 0/0 , $\sqrt{-1}$

Examples

- **Example:**
- **0 00000000 00000000000000000000000 = 0**
- **1 00000000 00000000000000000000000 = -0**
- **0 11111111 00000000000000000000000 = + infinity**
- **1 11111111 00000000000000000000000 = - infinity**
- **0 11111111 01001100010001000001000 = NaN (Not a Number)**
- **1 11111111 01001100010001000001000 = NaN**

- **Question:**
- **0 00000000 00000000000000000000001 = ?**
- **0 00000001 00000000000000000000000 = ?**
- **0 00000000 10000000000000000000000 = ?**
- **0 10000000 00000000000000000000000 = ?**
- **1 00000010 10100000000000000000000 = ?**

Solution

Solution:

0 00000000 00000000000000000000001 = 2^{-149}

0 00000001 00000000000000000000000 = 2^{-126}

0 00000000 10000000000000000000000 = 2^{-127}

0 10000000 00000000000000000000000 = 2

1 00000010 10100000000000000000000 = -1.625 $\times 2^{-125}$

Representation of text with ASCII codes

- The American Standard Code for Information Interchange (ASCII)
- ASCII is a character encoding scheme for encoding text in 8 bits
- The list of the first 128 characters are shown below

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Exercise 1

- Given the bit pattern 1000 0000 0100 0110 0000 0000 0000 0000
  - What is the value if this is a 2's complement representation?
  - Solution: -2142896128_{10}
  - What if the pattern is an unsigned integer?
  - Solution: 2152071168_{10}
  - What if it is an IEEE single precision number?
  - Solution: $-0.100011_2 * 2^{-126}$
  - What if it represents 4 ASCII characters (assume bits 31-24, 23-16, 15-8, 7-0 store the characters, and ASCII value of 128 is the symbol '€').
  - Solution: € F NULL NULL

Exercise 2

- Assume the bit pattern 1001 1100 follows the IEEE-like floating point representation format

S	Exponent	significand
1 bit	**3 bits**	**4 bits**

- What is the bias of the exponent?
- Solution: $2^2 – 1 = 3$
- What value is the given pattern representing?
- Solution: $-1.11_2 * 2^{-2} = -0.4375_{10}$
- What is the range of numbers that this IEEE-like floating point representation system can represent?
- Solution: -15.5 to 15.5 (in decimal).
- Can 14.25 (decimal) be represented using this system?
- Solution: No, there is not enough bits for significand since 14.25 converted to binary is 1110.01 = $1.11001 * 2^3$