9.1 ASCII and Unicode

Bits: 0's and 1's

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Computers are built from connected switches that, like light switches, are either on or off. On is represented as 1, and off is 0. A single 0 or 1 is called a ***bit***. 1011 is four bits. Eight bits, like 11000101, are called a ***byte***.

Humans represent information using characters and numbers like Z or 42. To present information that people can understand, computers need a way to represent characters and numbers using 0's and 1's.

PARTICIPATION ACTIVITY 9.1.1: Bits.

1) A 0 or 1 is called a _____ .

Check Show answer

2) Eight bits are called a _____ .

Check Show answer

3) 101100 has _____ bits.

Check Show answer

Characters as bits: ASCII

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

A ***character*** is a letter (a, b, ..., z, A, B, ..., Z), symbol (!, @, #, ...), or single-digit number (0, 1, ..., 9). Basically, each item on a computer keyboard is a character (though more characters exist). Each character can be given a unique bit code.

ASCII is a popular code for characters. ASCII stands for American Standard Code for Information Interchange, and was developed in 1963. ASCII uses 7 bits per code, and has codes for 128 characters. Ex: Using ASCII, the letter Z would be stored in a computer as 1011010. This material

inserts a space for readability, as in: 101 1010. Each bit code is sometimes written as an equivalent decimal number (written as Dec below), discussed later.

Table 9.1.1: ASCII bit codes for common characters.

Bit code	Dec	Char
010 0000	32	space
010 0001	33	!
010 0010	34	"
010 0011	35	#
010 0100	36	$
010 0101	37	%
010 0110	38	&
010 0111	39	'
010 1000	40	(
010 1001	41	)
010 1010	42	*
010 1011	43	+
010 1100	44	,
010 1101	45	-
010 1110	46	.
010 1111	47	/
011 0000	48	0
011 0001	49	1
011 0010	50	2
011 0011	51	3

Bit code	Dec	Char
100 0000	64	@
100 0001	65	A
100 0010	66	B
100 0011	67	C
100 0100	68	D
100 0101	69	E
100 0110	70	F
100 0111	71	G
100 1000	72	H
100 1001	73	I
100 1010	74	J
100 1011	75	K
100 1100	76	L
100 1101	77	M
100 1110	78	N
100 1111	79	O
101 0000	80	P
101 0001	81	Q
101 0010	82	R
101 0011	83	S

Bit code	Dec	Char
110 0000	96	`
110 0001	97	a
110 0010	98	b
110 0011	99	c
110 0100	100	d
110 0101	101	e
110 0110	102	f
110 0111	103	g
110 1000	104	h
110 1001	105	i
110 1010	106	j
110 1011	107	k
110 1100	108	l
110 1101	109	m
110 1110	110	n
110 1111	111	o
111 0000	112	p
111 0001	113	q
111 0010	114	r
111 0011	115	s

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

011 0100	52	4
011 0101	53	5
011 0110	54	6
011 0111	55	7
011 1000	56	8
011 1001	57	9
011 1010	58	:
011 1011	59	;
011 1100	60	<
011 1101	61	=
011 1110	62	>
011 1111	63	?

101 0100	84	T
101 0101	85	U
101 0110	86	V
101 0111	87	W
101 1000	88	X
101 1001	89	Y
101 1010	90	Z
101 1011	91	[
101 1100	92	\
101 1101	93	]
101 1110	94	^
101 1111	95	_

111 0100	116	t
111 0101	117	u
111 0110	118	v
111 0111	119	w
111 1000	120	x
111 1001	121	y
111 1010	122	z
111 1011	123	{
111 1100	124	|
111 1101	125	}
111 1110	126	~

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.1.2: ASCII bit codes (and decimal number equivalents).

Type a character: A ASCII bit code: **1000001** ASCII number: **65**

PARTICIPATION ACTIVITY 9.1.3: ASCII.

1) What is the 7-bit code for a lower-case 'a'?

Check Show answer

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

2) What is the 7-bit code for a blank space?

Check Show answer

3) What two-letter word does this sequence of bits represents in ASCII? Pay attention to upper/lower case. Use the above ASCII table.
1001000 1101001

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

4) Suppose an email message has 500 characters. How many bits would a computer use to store that email, using ASCII code having 7 bits per character?

bits

Check Show answer

CHALLENGE ACTIVITY 9.1.1: ASCII code.

347136.1658324.qx3zqy7

Start

Convert the character to 7-bit ASCII code.
Ex: a is 1100001

E: Ex: 1100001

1 2 3 4

Check Next

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Text is a sequence of character codes

Computers commonly deal with text, consisting of a sequence of characters. The computer stores each character's ASCII code in successive locations in the computer's memory. Each location has at

least enough bits (often more) to store an ASCII code.

PARTICIPATION ACTIVITY 9.1.4: Characters are encoded and stored in the computer's memory.

Animation captions:

1. Each character has a unique bit code in ASCII. Uppercase S is 101 0011.
2. Text is stored as a sequence of bit codes in the computer's memory. S is 101 0011, e is 110 0101, a is 110 0001, and t is 111 0100.
3. Symbols and spaces are also characters, and stored in the memory as bit codes. A colon (:) is 011 1010. A space is 010 0000.
4. Here, 9 is just another character, and has code 011 1001. Character a appears a second time, again with code 110 0001.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.1.5: Characters in a computer's memory.

Refer to the above animation.

1) How many characters are in the text displayed on the screen on the left side of the animation?

Check Show answer

2) How many memory locations are used to store that text?

Check Show answer

3) The first letter in the text is S. What bits are stored in the first memory location?

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Check Show answer

4) What is the minimum number of bits that each memory location

must be able to store in the example above?

Check Show answer

5) How many total bits are needed to store the text shown?

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Check Show answer

Encoding more characters: Unicode

Unicode is another character encoding standard, published in 1991, whose codes can have more bits than ASCII and thus can represent over 100,000 items, such as symbols and non-English characters. Characters in Unicode are represented as a number, or ***code point***. Ex: In Unicode, the letter "H" is represented as U+0048. U+ means the character is encoded in Unicode, and 0048 is the corresponding code point. The code point is written in hexadecimal, which is discussed elsewhere.

Characters can range from U+0000 to U+10FFFF. The table below provides a very small subset of encodings.

Table 9.1.2: Unicode code points for control characters and basic Latin.

Code point	Char	Code point	Char	Code point	Char	Code point	Char	Code point	Char	Code point	C
0020	space	0030	0	0040	@	0050	P	0060	`	0070	
0021	!	0031	1	0041	A	0051	Q	0061	a	0071	
0022	"	0032	2	0042	B	0052	R	0062	b	0072	
0023	#	0033	3	0043	C	0053	S	0063	c	0073	
0024	$	0034	4	0044	D	0054	T	0064	d	0074	
0025	%	0035	5	0045	E	0055	U	0065	e	0075	
0026	&	0036	6	0046	F	0056	V	0066	f	0076	
0027	'	0037	7	0047	G	0057	W	0067	g	0077	

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

0028	(	0038	8	0048	H	0058	X	0068	h	0078	
0029	)	0039	9	0049	I	0059	Y	0069	i	0079	
002A	*	003A	:	004A	J	005A	Z	006A	j	007A	
002B	+	003B	;	004B	K	005B	[	006B	k	007B	
002C	,	003C	<	004C	L	005C	\	006C	l	007C	
002D	-	003D	=	004D	M	005D	]	006D	m	007D	
002E	.	003E	>	004E	N	005E	^	006E	n	007E	
002F	/	003F	?	004F	O	005F	_	006F	o		

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

UTF-8, UTF-16, and UTF-32 are encoding standards that indicate how the Unicode is stored. In the UTF-8 standard, characters are stored using variable widths and range from one to four bytes. Whereas in the UTF-32 encoding, all characters are stored as a single 32-bit value. An application that converts the encoding to the final characters viewed by the end user must know which standard is utilized. Emails, web pages, and other digital media frequently contain additional information, or metadata, to indicate how characters are stored. Ex: A webpage may contain the tag <meta charset='utf-8'> to indicate that the UTF-8 Unicode standard is used to encode text.

PARTICIPATION ACTIVITY 9.1.6: Unicode.

1) An uppercase letter P is represented as _____ in Unicode.
   - U+0050
   - U+0070

2) U+0020 represents _____ .
   - an uppercase A
   - a space

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

3) Which text is represented by the following unicode?
   U+0061 U+0020 U+0062 U+0020 U+0063
   - a b c

ABC

4) In Unicode, each character is stored as a 16-bit value.

True

False

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

CHALLENGE ACTIVITY 9.1.2: Unicode code point.

347136.1658324.qx3zqy7

Start

Convert the character to 4-digit hexadecimal code point.
Ex: a = U+0061

Z: U+ Ex: 0061

1 2 3 4

Check Next

This section provides a simple introduction to Unicode. We encourage the interested reader to Unicode Consortium for additional information on advanced topics and features.

Exploring further:

- Wikipedia: ASCII
- http://www.asciitable.com/
- Unicode 9.0 Character Code Charts

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

9.2 Unsigned binary numbers

Counting in binary

Humans have ten fingers so humans use a base ten number system. Ex: 452 means $4\times10^2 + 5\times10^1 + 2\times10^0$. Digital systems have two-valued signals (high, low) so digital systems use a base two number system. Ex: 1101 means $1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$. A number in base ten is called a ***decimal*** number (from Latin "decem" meaning ten), while a number in base two is called a ***binary*** number (from Latin "bini" meaning two together).

Base ten has ten symbols for a digit: 0, 1, ..., 9. When counting up and reaching 9, the digit resets to 0 and a 1 carries to the next digit. Ex: 008, 009, 010, 011, or 098, 099, 100, 101. Base two has only two symbols for a digit: 0 and 1. So counting up results in frequent carries. Ex: 000, 001, 010, 011, 100, 101, 110, 111. Each digit in a binary number is called a ***bit***, short for "binary digit".

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.2.1: Counting up in decimal and in binary.

Animation captions:

1. Various quantities.
2. In base ten, the first digit goes up to 9, then the digit is reset to 0 and a 1 is carried to the next digit.
3. If the carry is to a digit that is already 9, then that digit is also reset to 0 and a 1 is carried to the next digit.
4. In base two, the first digit goes up to 1, then the digit is reset to 0 and a 1 is carried to the next digit.
5. If the carry is to a digit that is already 1, then that digit is also reset to 0 and a 1 is carried to the next digit.
6. Resets and carries happen on every other count.

Note: This section only covers unsigned binary numbers. An ***unsigned binary*** number can only represent non-negative values, such as a 4-bit binary number being 0000 (0), 0001 (1), 0010 (2), ..., 1111 (15). In contrast, a signed binary number uses the leftmost bit to represent whether a number is positive or negative.

PARTICIPATION ACTIVITY 9.2.2: Counting up in binary.

Type the next higher 3-digit binary number.

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

1) 000

Check Show answer

2) 001

Check Show answer

3) 010

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

4) 011

Check Show answer

5) 111 (type a 4-bit answer)

Check Show answer

CHALLENGE ACTIVITY 9.2.1: Counting up with 3 bits.

Can you count from 000 to 111 in binary in 20 seconds?

347136.1658324.qx3zqy7

Start

Decimal	Binary (3 bits)
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

1 2 3 4 5 6 7 8

Check Next

Converting from binary to decimal

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Software and hardware developers benefit from being able to quickly convert between binary and decimal numbers.

Given a binary number, each digit's weight is summed to form a decimal number. Ex: 1101 = $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13$.

PARTICIPATION ACTIVITY 9.2.3: Binary to decimal tool.

Reset Each value ranges 0 to 1

0	0	0	0	0	0	0	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

0 · 128 + 0 · 64 + 0 · 32 + 0 · 16 + 0 · 8 + 0 · 4 + 0 · 2 + 0 · 1 = 0

(decimal value)

PARTICIPATION ACTIVITY 9.2.4: Converting from binary to decimal.

Convert from binary to decimal. Use the fewest decimal digits possible. Recall that four-bit binary digit weights are 8, 4, 2, and 1.

1) 0001

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

2) 0010

Check Show answer

3) 0111

Check Show answer

4) 1001

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

5) 1111

Check Show answer

Converting from decimal to binary

Given a decimal number, starting from the leftmost binary digit (greater than the decimal number), a 1 is placed in each digit as long as the resulting binary number doesn't exceed the decimal number.

PARTICIPATION ACTIVITY 9.2.5: Converting from decimal to four-bit binary.

Type a four-bit answer: 0101, not 101. Four-bit binary digit weights: 8, 4, 2, 1.

1) 3

Check Show answer

2) 4

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

3) 5

Check Show answer

4) 13

Check Show answer

PARTICIPATION ACTIVITY 9.2.6: Binary-to-decimal converter.

Binary		Decimal
0	⇄	0
Binary numbers have 0s and 1s		Value ranges from 0 to 65535

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

Bits required to represent a number

A computer has a limited amount of storage available to represent numbers as bits. The number of bits available to store a number defines the range of numbers that can be represented. Ex: If 2 bits are available, then the decimal numbers 0 (00_2), 1 (01_2), 2 (10_2), and 3 (11_2) can be represented, and the range is 0-3.

PARTICIPATION ACTIVITY 9.2.7: Number of bits required to represent a decimal number.

Determine the minimum number of bits required to represent each decimal number.

1) 1

Check Show answer

2) 5

Check Show answer

3) 15

Check Show answer

4) 16

Check Show answer

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

Adding binary numbers

For decimal numbers, adding by hand starts at the right and adds each digit, possibly carrying a 1 to the digit on the left. Adding binary numbers is identical.

PARTICIPATION ACTIVITY 9.2.8: Adding in binary.

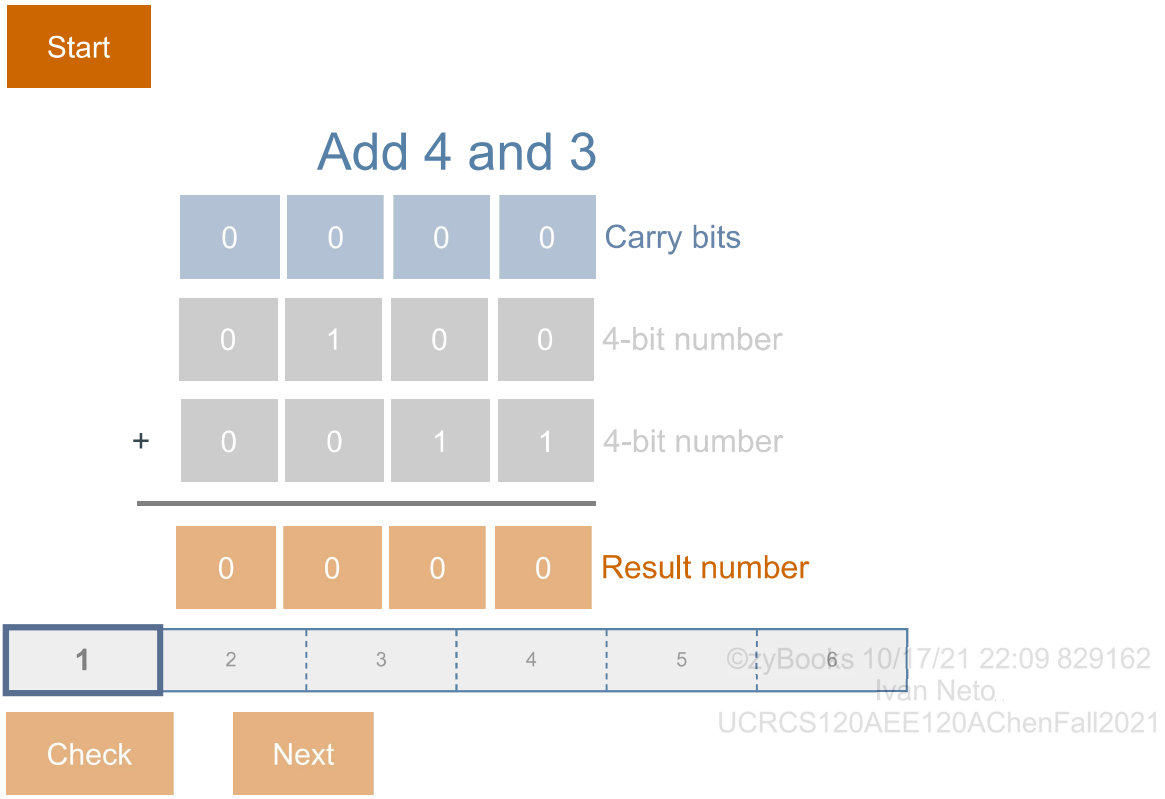
©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Animation captions:

1. Add rightmost digit: 1 + 0 = 1.
2. Add next digit: 1 + 1 = 10, so carry 1.
3. Continue for next digit: 1 + 1 + 1 = 11. Carry 1.
4. Continue for the last digit: 1 + 0 + 0 = 1.

CHALLENGE ACTIVITY 9.2.2: Binary addition.

347136.1658324.qx3zqy7



©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.2.9: Adding binary numbers.

Note: A carry bit may cause the result to have five bits.

1) 0010
+ 0010

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

2) 0110
+ 0010

Check Show answer

3) 0101
+ 0111

Check Show answer

4) 1111
+ 0001

Check Show answer

Overflow

Overflow occurs when the result of a binary operation is too large to fit in allowed number of bits. Ex: For four-bit numbers, 1111 + 0001 is 10000, which is too large for four bits. When adding two unsigned numbers, if the leftmost bit generates a carry bit, overflow has occurred.

zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.2.10: Overflow for unsigned numbers.

Indicate which operations yield overflow for four-bit binary numbers.

1) 0001 + 0010

- Overflow
- No overflow

2) 0111 + 0111

- Overflow
- No overflow

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

3) 1000 + 0111

- Overflow
- No overflow

4) 1000 + 1000

- Overflow
- No overflow

5) 1100 + 0111

- Overflow
- No overflow

6) 1111 + 1111

- Overflow
- No overflow

7) In base 10, for 2 digit numbers:
50 + 70

- Overflow
- No overflow

9.3 Signed binary numbers: Two's complement

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Two's complement

Unsigned numbers involve only non-negative numbers, like 0 and 3. ***Signed numbers*** involve both positive and negative numbers, like 3 and -3.

In binary, a ***signed-magnitude representation*** uses the left bit for the sign: 0 means positive, 1 means negative. Ex: For 4-bit numbers, 0011 is 3, and 1011 is -3. Signed-magnitude representation is rarely used, because calculations involving negative numbers, such as 5 - 3, would require special circuits beyond an adder.

A more clever negative number representation exists that can use an adder for both positive and negative numbers. A ***complement*** of an N-digit number is another number that yields a sum of 100...00 (with N 0's), and can be used to represent the negative of that number.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.3.1: Two's complement signed number representation.

Animation captions:

1. Intuition in base 10.
2. In base two, a complement is obtained by inverting each bit and adding 1.
3. Subtraction is performed by adding the complement.

The above is called the ***two's complement representation***, which inverts every bit and adds 1. One's complement also exists, but is rarely used, and so is not discussed further. This material uses "complement" to mean two's complement.

The left bit indicates the sign. 0011 is +3. 1101 is a negative; complementing yields the positive version: 0010 + 1 = 0011, which is 3. So 1101 is -3.

Given a negative number like 1110, the value can be obtained by complementing, so 0001 + 1 = 0010, and negating, so -0010. Thus 1110 is -2.

PARTICIPATION ACTIVITY 9.3.2: Two's complement signed number representation.

1) In base ten , what is the complement of 33 (two digits)?

Check Show answer

2) In base two , what is the complement of 0010 (four bits)?

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Check Show answer

3) What is -2 in four-bit two's complement representation?

Check Show answer

4) What is -7 in four-bit two's complement representation?

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

5) Assuming four-bit two's complement representation, is 1011 positive or negative?

Check Show answer

6) Assuming two's complement representation, what base ten number does 1111 represent?

Check Show answer

7) Assuming two's complement representation, what base ten number does 1001 represent?

Check Show answer

8) In base two, for four bits, what is the complement of 0000?

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

9) What is -3 in eight-bit two's complement representation?

Check Show answer

Note: This section uses 4-bit numbers for ease of example; wider numbers like 8 or 32 bits are more typical.

Subtracting by adding

Two's complement representation has the benefit of allowing an adder to be used even when dealing with negative numbers. Ex: 5 + -3 is just 0101(5) + 1101(-3) = 10010, or 0010(2) after ignoring the carry. No extensive special circuitry for negative numbers is needed.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.3.3: Two's complement arithmetic.

Assume four-bit two's complement representation.

1) 6 + 2 is 0110 + ?

Check Show answer

2) 6 + -2 is 0110 + ?

Check Show answer

3) 3 + -4 is 0011 + 1100 = ?

Check Show answer

4) 2 - 3 is 0010 + ?

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

5) -3 + 2 is ? + 0010

Check Show answer

Overflow

The largest positive four-bit two's complement number is 0111, or 7. The smallest negative is 1000, or -8 (0111 + 1 = 1000, so magnitude is 8). Adding two positives, or adding two negatives, may yield a value that can't be represented in the given number of bits, a situation known as ***overflow***. Ex: 0101 (5) + 0011 (3) incorrectly yields 1000, which is -8 in two's complement.

PARTICIPATION ACTIVITY 9.3.4: Overflow.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Animation content:

undefined

Animation captions:

1. Adding two positives may overflow.
2. Adding two negatives may overflow. Note: Ignore carry-out bit.
3. Adding a positive and negative cannot overflow.
4. Overflow occurs if numbers' sign bits were 0's but sum's is 1, OR numbers' sign bits were 1's but sum's is 0.

As seen above, overflow occurs if the numbers being added have the same sign bit but the sum's sign bit differs. In other words, overflow occurs if two positives sum to a negative (clearly wrong), or two negatives sum to a positive (clearly wrong).

Adding a positive number and negative number (or vice-versa) cannot result in overflow. The sum always has a smaller magnitude than one or both of the numbers, so clearly can fit in the same number of bits. Ex: 7 + -2 = 5, and 5's magnitude is smaller than 7.

PARTICIPATION ACTIVITY 9.3.5: Overflow.

All numbers are in four-bit two's complement representation.

1) 0011 + 0010 results in overflow.

- ○ True
- ○ False

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

2) 0111 + 0110 results in overflow.

- ○ True
- ○ False

3) 0001 + 1111 results in overflow.

- True
- False

4) 1011 + 1110 results in overflow.

- True
- False

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

5) Number A's sign bit is 0. Number B's sign bit is 0. The sum's sign bit is 1. The addition resulted in overflow.

- True
- False

6) Number A's sign bit is 1. Number B's sign bit is 0. The sum's sign bit is 0. The addition resulted in overflow.

- True
- False

CHALLENGE ACTIVITY 9.3.1: Two's complement.

347136.1658324.qx3zqy7

Start

Note: Answer with 1 digit.

Decimal complement of 6 is Ex: 9

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

1 2 3 4 5 6

Check Next

9.4 Binary, hexadecimal, and octal

Hexadecimal

While decimal means a base 10 number, ***hexadecimal*** (or ***hex***) means a base 16 number. Each digit is an increasing power of 16: 16^0, 16^1, 16^2, etc.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

16 symbols are needed for a digit. The symbols are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Hexadecimal is popular due to being a compact representation of a binary number. Four binary digits can be represented as one hex digit, since both have 16 possibilities. Thus 0000 is 0, 0001 is 1, 0010 is 2, ..., 1110 is E, and 1111 is F.

More binary digits use more hex digits. Ex: 00101111 is 2F (0010 is 2, and 1111 is F).

PARTICIPATION ACTIVITY 9.4.1: Hex is popular due to compactly representing binary.

Animation captions:

1. Four bits have 16 possible combinations.
2. One hex digit has 16 possible values. The first 10 use the usual numeric symbols. The next 6 use letters.
3. Given a binary number, every four bits can be represented with one hex digit. (In decimal, each hex digit represents 0 to 15).

PARTICIPATION ACTIVITY 9.4.2: Converting binary to hex.

1) 0011

- B
- 3

2) 1010

- A
- 10

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

3) 1111

- F
- Not possible

4) 11110000

- FF
- F0

5) 10100101

- A5
- 5A

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

6) 10011

- 91
- 13

7) 0001111100001111

- 1F0F
- F0F1

PARTICIPATION ACTIVITY 9.4.3: Converting hex to binary.

1) AA

Check Show answer

2) 0F

Check Show answer

3) 00

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

4) FF0077

Check Show answer

PARTICIPATION ACTIVITY 9.4.4: Hex example: Colors on the web.

Colors on web pages are specified using 24 bits: 8 bits for red, 8 green, and 8 blue. Hex is a convenient way to specify those 24 bits. Thus FF0000 is bright red, 990000 is a darker red, FF00FF is a bright purple, and 000000 is black. Modify some colors below and press "Render HTML".

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Reset

Type HTML below

```
<!DOCTYPE html>
<html>
<body>

<p style="background-color:#FF0000">
Line1 </p>
<p style="background-color:#990000">
Line2 </p>
<p style="background-color:#FF00FF">
Line3 </p>
<p style="background-color:#00FF00">
Line4 </p>
<p style="background-color:#777777">
Line5 </p>
<p style="background-color:#000000">
Line6 </p>
</body>
```

Rendered HTML

Line1

Line2

Line3

Line4

Line5

Line6

Render HTML

Hex to decimal

A hex number is converted to decimal simply by multiplying each digit's decimal value by that digit's weight and summing. Ex: A7F = $10 \times 16^2 + 7 \times 16^1 + 15 \times 16^0 = 2560 + 112 + 15 = 2687$.

PARTICIPATION ACTIVITY 9.4.5: Hex to decimal.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

1) A

- 1
- 10

2) A7

- 17
- 107
- 167

3) FFF

- 45
- 4095
- 151515

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Decimal to hex

Decimal is converted to hex by finding the highest hex digit where a 1 doesn't exceed the decimal value, incrementing as much as possible without exceeding the decimal value, and repeating for lower digits.

PARTICIPATION ACTIVITY 9.4.6: Decimal to hex.

1) 20 decimal

Check Show answer

2) 40 decimal

Check Show answer

3) 258 decimal

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Octal

Octal means a base 8 number. Octal is sometimes used as a compact binary representation because three bits can be represented as one octal digit, though hex is more common. The eight symbols for an octal digit are 0, 1, 2, 3, 4, 5, 6, 7.

PARTICIPATION ACTIVITY 9.4.7: Binary to octal, octal to binary.

1) 101010 binary is ____ octal.

- 52
- 42

2) 1010 binary is ____ octal.

- 22
- 12

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

3) 14 octal is ____ binary.

- 001100
- 1110

4) 07 octal is ____ binary.

- 111
- Invalid

5) 80 octal is ____ binary.

- 1000
- Invalid

CHALLENGE ACTIVITY 9.4.1: Binary, hexadecimal, and octal conversions.

347136.1658324.qx3zqy7

Start

Convert from hex to decimal

37 = Ex: 73

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

1 2 3 4 5

Check Next

9.5 General number bases

Bases

Numbers can use nearly any base, such as base 5 or base 17. Popular bases are:

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

- 2: Binary, due to computers using 0's and 1's
- 8: Octal, which can compactly represent binary (each octal digit represents three bits)
- 16: Hexadecimal (hex), which can compactly represent binary (each hex digit represents four bits).
- 10: Decimal, due to humans having ten fingers.

When a number's base is unclear, the base is written as a subscript, as in 930_{15}.

Each digit in a base B is weighted by a power of B, as below. Each digit can be 0 to B − 1.

Figure 9.5.1: Each digit in a base B is weighted by a power of B.

$$\underline{\quad}_{B^3}\ \underline{\quad}_{B^2}\ \underline{\quad}_{B^1}\ \underline{\quad}_{B^0} \qquad \frac{v}{B^3}\ \frac{w}{B^2}\ \frac{y}{B^1}\ \frac{z}{B^0}$$

$$v \times B^3 + w \times B^2 + y \times B^1 + z \times B^0$$

PARTICIPATION ACTIVITY 9.5.1: General bases.

1) For base 5, what are the possible values for each digit?
   - 0 to 4
   - 0 to 5

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

2) 104 base 5 is ____ in decimal.
   - 29
   - 135

3) 128 is a valid base 8 number.
   - True

False

From any base to decimal

Converting from any base to decimal is straightforward: Each digit's decimal value is multiplied by each digit's decimal weight, and summed.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.5.2: Various bases to decimal tool: Try selecting different bases.

Base-2 **Reset** Each value ranges 0 to 1

0	0	0	0	0	0	0	0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

0 · 128 + 0 · 64 + 0 · 32 + 0 · 16 + 0 · 8 + 0 · 4 + 0 · 2 + 0 · 1 = 0

(decimal value)

PARTICIPATION ACTIVITY 9.5.3: General bases to decimal.

1) What is 111 base 5 in decimal?

- 31
- 111

2) What is 200 base 30 in decimal?

- 60
- 1800

From decimal to any base

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Converting from decimal to any base can be done using a simple algorithm. The decimal is divided by the base, and remainder put in the rightmost digit. The process repeats with the quotient and the next digit, until the quotient is 0.

PARTICIPATION ACTIVITY 9.5.4: Converting from decimal to any base.

Animation captions:

1. To convert to decimal, one multiplies each digit's decimal value (2, 11, 3) by each digit's weight, then sums. (Note that in base 15, A's decimal value is 10, B's is 11, etc).
2. To convert decimal to any base like 5, one divides by the base, placing the remainder in rightmost digit.
3. One then repeats, starting with quotient, putting remainder in next digit.
4. One stops when quotient is 0.
5. The result can be converted back to base 10 to check one's work.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.5.5: Converting decimal to/from other bases.

1) What is 34 base 6 in decimal?

Check Show answer

2) What is 102 base 3 in decimal?

Check Show answer

3) When converting 29 decimal to base 3, what is the rightmost digit in base 3?

Check Show answer

4) When converting 29 decimal to base 3, 29 / 3 yields quotient 9. What is the second digit in base 3?

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

5) What is 29 decimal in base 3?

Check Show answer

Any base to any base

To convert a number in any base B directly to any other base C, a straightforward approach first converts the base B number to a decimal number, then converts that decimal number to base C. Ex: To convert 320_5 to base 8, one starts with $320_5 = 3{\times}25 + 2{\times}5 = 85_{10}$, followed by converting to base 8: 85/8 = 10 remainder 5, 10/8 = 1 remainder 2, 1/8 = 0 remainder 1, so concatenating yields 125_8.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.5.6: Any base to any base.

Order the steps for converting 200_3 to base 9.

18 / 9 = 2 rem 0 2 / 9 = 0 rem 2	18_{10}	20_9	200_3	2×9 + 0×3 + 0×1

1
2
3
4
5

Reset

CHALLENGE ACTIVITY 9.5.1: Converting between bases.

347136.1658324.qx3zqy7

Start

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Convert from base 8 to base 10

21_8 = Ex: 12 $_{10}$

1 2 3 4 5 6

Check Next

9.6 Floating-point numbers

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Floating-point numbers and normalized scientific notation

An ***integer*** is a whole number, like 42, 0, or -95. A ***floating-point number*** is a real number, like 98.6, 0.0001, or -666.667. The term "floating-point" refers to the decimal point being able to appear anywhere ("float") in the number.

- Integers are typically used for values that can be counted, like 42 cars, 0 pizzas, or -95 days.
- Floating-point numbers are typically used for values that are measured, like 98.6 degrees, 0.0001 meters, or -666.667 grams.

To improve readability and consistency, floating-point numbers are commonly written using ***normalized scientific notation***, such as 9.86×10^{1}, 1.0×10^{-4}, or -6.66667×10^{2}, where the number is written as a digit (+/- 1 to 9), decimal point, fractional part, times 10 to a power. The term "normalized" is in contrast to non-normalized where more than one digit, or a 0, may precede the decimal point, such as $-66.6667 \times 10^{-0.1}$ or 0.1×10^{-3}.

The parts of scientific notation are named ***significand*** for the part before × and ***exponent*** for the power of 10: significand × 10^{exponent}. If the exponent is 0, the power of ten part is sometimes omitted, as in 5.7.

In binary, normalized scientific notation consists of $1.f \times 2^{\text{exponent}}$, like 1.010×2^{5}. f is the fractional part.

PARTICIPATION ACTIVITY 9.6.1: Normalized scientific notation: Decimal.

Indicate which numbers are in decimal normalized scientific notation.

1) 2.05×10^{3}

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

- ○ Yes
- ○ No

2) 0.50×10^{3}

- ○ Yes
- ○ No

3) 27.8×10^3

- Yes
- No

4) 3.5

- Yes
- No

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

5) -5.77×10^3

- Yes
- No

6) 2.05×10^{-3}

- Yes
- No

7) 0.0

- Yes
- No

PARTICIPATION ACTIVITY 9.6.2: Normalized scientific notation: Binary.

Indicate which numbers are in binary normalized scientific notation.

1) 0.0

- Yes
- No

2) 1.01×2^3

- Yes
- No

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

3) 0.10×2^3

- Yes
- No

4) 11.10×2^3

- Yes
- No

5) -1.01×2^3

- Yes
- No

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

6) 1.01×2^{-3}

- Yes
- No

CHALLENGE ACTIVITY 9.6.1: Normalized scientific notation.

347136.1658324.qx3zqy7

Start

Write the number in normalized scientific notation.

Use ^ for exponents. Ex: 10^4 for 10^4.

121.157 = Ex: 1.23x10^3

1 2 3 4

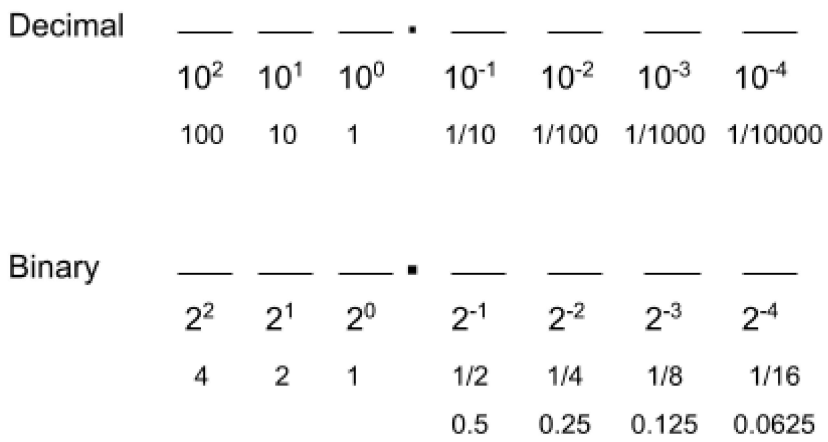
Check Next

Fractions in binary

Fractions in binary are similar to fractions in decimal. In decimal, each digit after the decimal point has weight $1/10^1$, $1/10^2$, $1/10^3$, etc. In binary, each digit after the binary point has weight $1/2^1$, $1/2^2$, $1/2^3$, $1/2^4$, etc. (so 1/2, 1/4, 1/8, 1/16, etc.). Ex: 1.1101 is 1 + 1/2 + 1/4 + 1/16. Note that for binary numbers, the "dot" is called a ***binary point*** (versus decimal point for decimal numbers) The general term is *radix point*.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Figure 9.6.1: Fractional digit weights for decimal and binary.



Ex: 1.01 = 1 + 1/4 = 1.25

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Converting decimal to binary or vice-versa when fractions are involved uses the same process as without fractions. However, when manually converting a decimal with a fraction into binary, converting the whole and fraction parts separately, then concatenating, may be easier. Ex: For 12.25, 12 is 1100_2, and 0.25 is 0.01_2, yielding 1100.01.

If a decimal fraction cannot be exactly represented as a binary fraction using limited bits, one gets as close as possible, over or under. Ex: To represent 0.8 *with only four* binary fraction bits:

- 0.1100 is 1/2 + 1/4 = 0.75, which is under by 0.05.
- 0.1110 is 0.875, which is over by 0.075.
- 0.1101 is 0.8125, also over but only by 0.0125, which is closest and so is the best representation.

Obviously, more bits means binary fraction values can be closer.

PARTICIPATION ACTIVITY 9.6.3: Fractions in binary.

1) 1.1 binary = _____ decimal. Type as: #.#

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

2) 0.001 binary = _____ decimal. Type as: 0.###

Check Show answer

3) 10.101 binary = _____ decimal. Type as: #.###

Check Show answer

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

4) 0.75 decimal = _____ binary. Type as: 0.##

Check Show answer

5) 0.625 decimal = _____ binary. Type as: 0.###

Check Show answer

6) 16.75 decimal = _____ binary. Type as: #####.##

Check Show answer

7) 0.6 decimal = _____ binary, using two fraction bits. Type as: 0.##

Check Show answer

Loss of precision

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

Using a fixed number of bits means that some numbers cannot be precisely represented in a computer. Ex:

- A fraction that requires more bits than the allocated space allows. Ex: $\frac{1}{8} = 0.125$ when only one fraction bit is available.
- A rational number with an infinitely-repeating fraction portion. Ex: $\frac{2}{3} = 0.66666\ldots$
- An irrational number. Ex: $\pi = 3.14159\ldots$, $\sqrt{2} = 1.41421\ldots$, and $e = 2.71828\ldots$

Increasing the number of bits improves the precision with which such numbers can be represented, but at the cost of more storage.

PARTICIPATION ACTIVITY 9.6.4: Precision using a fixed number of bits.

Determine[a][b] whether each value can be represented exactly using the indicated number of fraction bits.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

1) $\frac{1}{2}$ using one fraction bit
   - Yes
   - No

2) $\frac{1}{4}$ using one fraction bit
   - Yes
   - No

3) $\frac{1}{4}$ using two fraction bits.
   - Yes
   - No

4) $\frac{1}{10}$ using three fraction bits.
   - Yes
   - No

5) $\frac{1}{9}$ using three fraction bits.
   - Yes
   - No

6) $\sqrt{5}$ using three fraction bits.
   - Yes
   - No

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Binary floating-point representation

In a computer, a binary floating-point number must be represented using a fixed number of bits. Normalized scientific notation is used, commonly using 32 or 64 bits. A common 32-bit floating-point binary representation has these items:

- Sign: 1 bit. 0 means positive, 1 negative.

- Exponent: 8 bits. Instead of two's complement, the exponent is biased, meaning a fixed value is subtracted, in this case 127. Thus, an exponent of 00000000 means $2^{0-127} = 2^{-127}$, and of 11111111 means $2^{255-127} = 2^{128}$.
- Fraction: 23 bits. Because the significand's first digit is always 1, the 1 implicitly precedes the fraction and thus isn't explicitly represented. The significand in scientific notation is commonly called the *mantissa*.

PARTICIPATION ACTIVITY 9.6.5: A 32-bit floating-point representation.

©zyBooks 10/17/21 22:09 829102
Ivan Neto
UCRCS120AEE120AChenFall2021

Animation captions:

1. 32 bit floating-point representation: 1 bit for sign (0 is +, 1 is -), 8 for exponent (biased by 127), and 23 bits for fraction (significand's leading 1 is implicit).
2. Representing 8: 0 for sign bit, 1.0 for significand (so 0 for the fraction part; the leading 1 is implicit), and 130 for exponent (because 3 + 127 = 130).
3. For -20: 1 for sign, 1.25 for significand (so 0.25 for fraction), and 131 for exponent (because 4 + 127 is 131).

The above is known as the ***IEEE single-precision binary floating-point*** format, which uses 32 bits: 1 bit for sign, 8 bits for exponent (biased by 127), and 23 bits for significand (leading 1 before binary point is implicit). ***IEEE double-precision binary floating-point*** format uses 64 bits: 1 bit for sign, 11 bits for exponent, and 52 bits for significant (leading 1 before binary point is implicit).

Note: In C, C++, and Java, a variable like "float x" uses 32-bits (single precision), while "double x" uses 64 bits (double precision). Programmers usually use double to obtain more significand precision, unless memory is tightly constrained.

PARTICIPATION ACTIVITY 9.6.6: Representation of single-precision floating-point values.

Enter a decimal value: _____

Sign	Exponent								Significand															
0	0	0	0	0	0	0	0	0	1.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	30	29	28	27	26	25	24	23		22	21	20	19	18	17	16	15	14	13	12	11	10	9	8

©zyBooks 10/17/21 22:09 829102
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.6.7: Binary floating-point to decimal.

Given the following binary floating-point representation, determine the following.

0 10000001 00100000000000000000000

1) Sign bit (type 0 or 1)

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

2) Exponent bits

Check Show answer

3) Exponent in decimal

Check Show answer

4) Significand's fraction bits (consider copy-pasting)

Check Show answer

5) Normalized scientific notation

× 2^2

Check Show answer

6) Decimal number

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.6.8: Negative exponents.

A 32-bit binary floating-point number has an exponent of 01111101 (125). Assume the sign bit is 0 and the significand is 1.000...

1) The number is $+1 \times 2^{?}$

- 125
- 127
- 2
- -2

2) The number in base ten is _____ .

- 0.25
- 1.000...
- -2

3) In the 32-bit representation, the sign bit applies to _____ .

- the significand
- the exponent
- both the significand and the exponent

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Decimal to binary floating-point

Decimal is converted to 32-bit floating-point by first converting the decimal number to binary, then normalizing and adjusting the exponent, and finally filling in the appropriate fields.

PARTICIPATION ACTIVITY 9.6.9: Converting decimal with fraction to binary floating-point.

Animation captions:

1. To convert decimal to binary floating-point format, the first step is to convert to binary.
2. The next step is to normalize by moving the binary point until exactly one 1 exists to the left. Each move left divides by two, requiring multiplying by 2 to keep the same value.
3. The last step is to fill in the sign bit (0 for +), exponent bits (3 is 130 biased by 127), and significand bits (the leading 1 is implicit)
4. Converting a number with a fraction is done similarly. First, the decimal is converted to binary. Second, the binary point is moved.
5. And third, the fields are filled in.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.6.10: Binary floating-point.

Consider converting a binary number to the above 32-bit binary floating-point format.

1) For 1.001_2, the 23 fraction bits are _____ .
   - 1.001000...
   - 001000...

2) For 11.101_2, the 23 fraction bits are _____ .
   - 11101000...
   - 1101000...
   - 101000...

3) For 1111.01_2, normalizing yields 1.11101. Thus, the *unbiased* exponent should be _____ .
   - 0
   - 2
   - 3

4) For 10010.01, normalizing yields 1.001001. Thus, the *biased* exponent should be _____ .
   - 4
   - 127
   - 131

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.6.11: Decimal to binary floating-point.

Consider converting the decimal number -16.25 to the above 32-bit binary floating-point format. Note that -16.25 is -10000.01 in binary.

000001000... | 10000011 | 131 | 1 | 4 | 1.000001000...

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

	Sign bit
	Unbiased exponent in base ten

	Biased exponent in base ten
	Exponent bits
	Fraction bits
	Significand

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Reset

CHALLENGE ACTIVITY 9.6.2: Binary floating point.

347136.1658324.qx3zqy7

Start

Enter the decimal equivalent of the exponent of the following floating point binary number.

1 00000010 1.10100000000000000000000

Decimal exponent: Ex: 10

1 2 3 4 5 6

Check Next

Exploring further:

- IEEE single-precision binary floating-point format
- IEEE double-precision binary floating-point format

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

9.7 Floating-point arithmetic

Adding decimal floating-point numbers

Adding two decimal floating-point numbers in normalized scientific notation can be done in three steps:

1. Make the exponents the same
2. Add the significands, applying the same exponent
3. If necessary, adjust the result to normalized scientific notation

PARTICIPATION ACTIVITY 9.7.1: Adding decimal floating-point numbers.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Animation captions:

1. The exponents are the same, so step (1) is complete. (2) Add the significands, applying the same exponent.
2. (3) If necessary, adjust the result to normalized scientific notation.
3. If the exponents are different, (1) adjust one number to have the same exponent as the other number.
4. (2) Add the significands, applying the same exponent.

PARTICIPATION ACTIVITY 9.7.2: Decimal floating-point addition.

Provide answers in normalized scientific notation. Use the letter x to express multiplication and the ^ symbol to express exponents Ex. 3 x 10^2.

1) $4 \times 10^3 + 2 \times 10^3 =$ _____

Check Show answer

2) $5.4 \times 10^5 + 4.1 \times 10^5 =$ _____

Check Show answer

3) $6.2 \times 10^2 + 6.3 \times 10^2 =$ _____

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Check Show answer

4) $8.5 \times 10^4 + 1.7 \times 10^4 =$ _____

Check Show answer

5) $2\times10^2 + 5.3\times10^4 =$ ____

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

6) $4.7\times10^3 + 7.1\times10^6 =$ ____

Check Show answer

Adding binary floating-point numbers

To add together binary floating-point numbers represented using normalized scientific notation, the exponents must first be the same, just like in decimal notation. Then one can add together the significands. If the result is not in normalized scientific notation, the result's exponent and binary point location are adjusted.

If the exponents differ, the process is again similar to the decimal notation process. One value's exponent is adjusted to match the other exponent by moving the location of the value's binary point before the significands are added.

Figure 9.7.1: Adding binary floating-point numbers.

$$\begin{array}{r} 1.110 \times 2^3 \\ +\quad 1.000 \times 2^3 \\ \hline 10.110 \times 2^3 \end{array}$$

$$= 1.011 \times 2^4$$

1. The exponents are already the same.
2. Add the significands and apply the exponent.
3. Adjust to normalized scientific notation.

$$\begin{array}{r} 1.110 \times 2^2 \\ +\quad 1.000 \times 2^4 \\ \hline \end{array} \quad \rightarrow \quad \begin{array}{r} 0.0111 \times 2^4 \\ +\quad 1.000 \times 2^4 \\ \hline 1.0111 \times 2^4 \end{array}$$

1. Make the exponents the same.
2. Add the significands and apply the exponent.
3. Result is already adjusted to normalized scientific notation.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.7.3: Binary floating-point addition.

Use the letter x to express multiplication and the ^ symbol to express exponents. Ex: 1.0 x 2^2. Remove trailing 0s from the results.

1) $1.0 \times 2^2 + 1.1 \times 2^2 =$ _____

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

2) $1.11 \times 2^4 + 1.01 \times 2^4 =$ _____

Check Show answer

3) $1.101 \times 2^1 + 1.011 \times 2^3 =$ _____

Check Show answer

4) $1.0 \times 2^2 + 1.001 \times 2^5 =$ _____

Check Show answer

5) $1.1 \times 2^4 + 1.1 \times 2^5 =$ _____

Check Show answer

6) $1.11 \times 2^7 + 1.11 \times 2^8 =$ _____

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Multiplying binary floating-point numbers

Multiplying floating-point numbers can be done in three steps:

1. Multiply the significands.

2. Add the exponents .
3. Adjust to normalized scientific notation.

Note that the exponents need not be the same. Ex: For decimal, given $(3.1\times10^5) \times (4.0\times10^2)$, then

1. 3.1 × 4.0 is 12.4
2. 5 + 2 is 7
3. Adjusting 12.4×10^7 yields 1.24×10^8

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

The steps for binary are the same.

PARTICIPATION ACTIVITY 9.7.4: Multiplying binary floating-point numbers.

Animation captions:

1. (1) Multiply the significands 1.01 and 1.11.
2. (2) Add the exponents 2 and 1.
3. Adjust to normalized scientific notation.

PARTICIPATION ACTIVITY 9.7.5: Binary floating-point multiplication.

1) $(1.01\times2^4) \times (1.1\times2^4) = ? \times 2^8$

Check Show answer

2) $(1.01\times2^3) \times (1.01\times2^4) = 1.1001\times2^?$

Check Show answer

3) For the following questions, use the letter x to express multiplication and the ^ symbol to express exponents. Ex. 1.0x2^2. Remove trailing 0s from the significands.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

$(1.1\times2^6) \times (1.0\times2^3) = ?$

Check Show answer

4) $(1.101\times2^3) \times (1.1\times2^3) = 10.0111\times2^6$ = ?

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

5) $(1.1\times2^5) \times (1.1\times2^3)$ = ?

Check Show answer

CHALLENGE ACTIVITY 9.7.1: Floating-point arithmetic.

347136.1658324.qx3zqy7

Start

Enter the exponent of the normalized scientific notation result.

Use ^ for exponents. Ex: 2^3 for 2^3.

$$\begin{array}{rl} & 1.100 \times 2^3 \\ + & 1.001 \times 2^3 \\ \hline & 1.0101 \times 2 \end{array}$$ Ex: 2

1 2 3 4 5 6

Check Next

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

9.8 Arrays

Array concept

A typical variable stores one data item, like the number 59 or the character 'a'. Instead, sometimes a *list* of data items should be stored. Ex: A program recording points scored in each quarter of a basketball game needs a list of 4 numbers. Requiring a programmer to define 4 variables is annoying; 200 variables would be ridiculous. An ***array*** is a special variable having one name, but storing a list of data items, with each item directly accessible. Some languages use a construct similar to an array called a ***vector***. Each item in an array is known as an ***element***.

PARTICIPATION ACTIVITY 9.8.1: Sometimes a variable should store a list, or array, of data items.

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

Animation captions:

1. A variable usually stores just one data item.
2. Some variables should store a list of data items, like variable pointsPerQuarter that stores 4 items.
3. Each element is accessible, like the element numbered 3.

You might think of a normal variable as a truck, and an array variable as a train. A truck has just one car for carrying "data", but a train has many cars each of which can carry data.

Figure 9.8.1: A normal variable is like a truck, whereas an array variable is like a train.

(Source for above images: Truck, Train)

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

Array indexing

In an array, each element's location number is called the ***index***; myArray[2] has index 2. An array's key feature is that the index enables direct access to any element, as in myArray[2]; different languages may use different syntax, like myArray(3) or myVector.at(3). In many languages, indices start with 0 rather than 1, so an array with 4 elements has indices 0, 1, 2, and 3.

CHALLENGE ACTIVITY 9.8.1: Update the array's data values.

347136.1658324.qx3zqy7

Start

Update myItems with the given code.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

myItems

0	12
1	53
2	67
3	92
4	49
5	30
6	59

1	2	3	4	5	6

Check Next

PARTICIPATION ACTIVITY 9.8.2: Array basics.

Array peoplePerDay has 365 elements, one for each day of the year. Valid accesses are peoplePerDay[0], [1], ..., [364].

1) Which assigns element 0 with the value 250?

- `peoplePerDay[250] = 0`
- `peoplePerDay[0] = 250`
- `peoplePerDay = 250`

2) Which assigns element 1 with the value 99?

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

- `peoplePerDay[1] = 99`
- `peoplePerDay[99] = 1`

3) Given the following statements:

```
peoplePerDay[9] = 5;
peoplePerDay[8] =
```

```
peoplePerDay[9] - 3;
```

What is the value of peoplePerDay[8]?

- 8
- 5
- 2

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

4) Assume N is initially 1. Given the following:

```
peoplePerDay[N] = 15;
N = N + 1;
peoplePerDay[N] =
peoplePerDay[N - 1] * 3;
```

What is the value of peoplePerDay[2]?

- 15
- 2
- 45

PARTICIPATION ACTIVITY 9.8.3: Arrays with element numbering starting with 0.

Array scoresList has 10 elements with indices 0 to 9, accessed as scoresList[0] to scoresList[9].

1) Assign the first element in scoresList with 77.

Check Show answer

2) Assign the second element in scoresList with 77.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Check Show answer

3) Assign the last element with 77.

Check Show answer

4) If that array instead has 100 elements, what is the last element's index?

Check Show answer

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

5) If the array's last index was 499, how many elements does the array have?

Check Show answer

(*Note_language_neutral) This section is mostly language neutral

9.9 Records

Grouping data

In the physical world, we are surrounded by basic items made from wood, metal, plastic, etc. But to keep the world understandable, we think at a higher level, in terms of objects like an oven, car, or house. Similarly, keeping track of thousands of separate data values, such as thousands of variables in a program, is hard. A higher level approach is needed to organize data in a more understandable way.

Figure 9.9.1: Grouping data into higher level objects improves comprehension.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Source: Lumber yard (Olivier Colas / CC-BY-SA-3.0 via Wikimedia Commons), House (Ellin Beltz / Public domain via Wikimedia Commons)

A ***record*** (also called a ***struct***) declares a new type, which can be used to group multiple subitems. Ex: A patient's health data may include subitems such as height, weight, and age. Each record subitem is called a ***field***. Some programming languages refer to subitems as members. A subitem may be any type like int, char, or string.

PARTICIPATION ACTIVITY 9.9.1: A record groups multiple data items.

Animation captions:

1. A record defines a new type, which is typically composed of several subitems.
2. Each field is accessible, like the weight field containing 120.

PARTICIPATION ACTIVITY 9.9.2: Naturally grouped data.

1) Select the pair indicating a flight's travel time.
   - Hours and minutes
   - Hours and cost
   - Pounds and ounces

2) Select the group of items most likely to indicate the change provided to a person who pays for a meal.
   - Ounce, gill, pint, quart, and gallon
   - Mile, furlong, yard, feet, and inches
   - Dollars, quarters, dimes, nickels,

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

and pennies

Accessing fields

`record healthData { ... }` defines a new data type named healthData. A programmer creates a record of type healthData using a variable declaration, as in the statement `healthData myVar;`. Fields can be accessed using ".", known as a ***dot notation*** operator.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.9.3: Using records in a program.

Animation captions:

1. healthData is a record composed of three subitems.
2. Two healthRecords are created, named EmilyDickinson and RalphEmerson. Memory is allocated for each record's fields.
3. A fields is accessed with the "." notation.

PARTICIPATION ACTIVITY 9.9.4: Records and fields.

```
record houseStats {
   int bedrooms;
   float bathrooms;
   int garageSize;
};

houseStats houseA;
```

1) `houseStats` is a _____ .

- record
- field

2) The following statement assigns houseA's bedroom field to 5.

```
houseA.bedrooms = 5;
```

- True
- False

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

3) Fields within a record should be the same data type.

- True
- False

(*Note_language_neutral) This section is mostly language neutral

9.10 Graphics

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Basic graphics

A ***digital graphic*** is a visual depiction of a scene comprised of a grid of picture elements known as ***pixels***, each having a number representing the pixel's color. As such, a series of numbers can represent a digital graphic.

PARTICIPATION ACTIVITY 9.10.1: A digital graphic is comprised of colored pixels.

Animation captions:

1. A digital graphic is comprised of a grid of pixels (whose borders are invisible, but are shown above). Coloring pixels depicts a visual scene, like a smiley face.
2. Each pixel has a color formed as a combination of red, green, and blue. In hex, FFFFFF is white, 0000AA is blue, and 00CC00 is green. The first row's values are shown.
3. The shown graphic has 64 pixels (8 × 8) per square inch. A decent quality graphic typically has 10,000 pixels (100 × 100) per square inch or more.

PARTICIPATION ACTIVITY 9.10.2: Digital graphic.

Consider the above digital graphic of a smiley face.

1) How many total pixels are available for the graphic?
   - 8
   - 64

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

2) How many pixels are colored blue?
   - 2
   - 64

3) How many pixels are colored white?
   - none

- 54
- 64

4) In hex, the fourth row starts with: ____ ____ ____ ____ .

- FFFFFF 0000AA FFFFFF
- FFFFFF 00CC00 FFFFFF
- 000000 00CC00 000000

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

5) A value like 00FF00 is three bytes (00 is a byte, FF is a byte, 00 is a byte). How many bytes are needed to represent an entire 64-pixel graphic?

- 64 bytes
- 128 bytes
- 192 bytes
- Depends on the scene being depicted

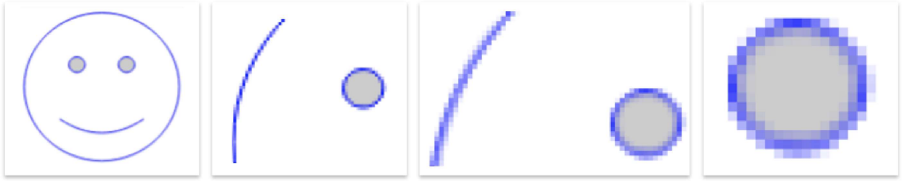
Smooth appearance

With enough pixel density, the human eye doesn't see individual pixels, instead seeing smooth lines. Pixel density is often indicated as ***pixels per inch***, being the number of rows (or columns) per inch. So 80 pixels per inch means 80 × 80 = 6400 pixels per square inch. Typically, about 100 pixels per inch are needed for humans to not see individual pixels. Higher quality graphics may have 200 or even 300 pixels per inch.

Digital graphics often use clever coloring and filling to make curved lines look smooth. The figure below shows how different shades of blue and grey pixels are used to make the curves look smooth. When zoomed in, the individual pixel colors are clearly seen, but at normal zoom, the curved lines look smooth.

Figure 9.10.1: With enough pixels per square inch, humans see smooth images, but zooming in shows individually-colored pixels.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021



A more complex scene is shown in the figure below. Again, zooming shows individually-colored pixels.

Figure 9.10.2: More complex scenes are similarly composed of individual pixels.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.10.3: Smooth images.

1) Pixel density is indicated as _____ .

- pixels per inch
- total number of pixels

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

2) If an image contains 500 pixels per inch, then each square inch contains _____ pixels .

- 500
- 250000

3) Pixels with differing shades of a given color can be used to make curved lines look smooth.

- True
- False

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

9.11 Image and video data

Images

An image is a depiction of a visual scene, typically composed of regions of varying colors. In the physical world, an image is an analog item, with continuous colors. Some forms of capturing an image use analog means, such as film. In contrast, a ***digital image*** is comprised of a grid of picture elements, ***pixels*** for short, with each pixel having one color. Each pixel's color is represented by a number, so a series of numbers represents an image. Each pixel's color number represents some combination of red, green, and blue, such as bright red (FF0000), a darker red (990000), or brown (663300 is a particular shade of brown).

PARTICIPATION ACTIVITY 9.11.1: Digitized images are represented by pixels. Each pixel's color is represented by a number. The more pixels, the better the image quality.

Animation captions:

1. An analog scene (say a tree) is converted to a digital image using a grid of colored pixels.
2. Each pixel is fully one color, like green, brown, or white. Each pixel's color is represented as a hex number. With too few pixels, the digitized image looks bad.
3. More pixels per inch means a more accurate digitization of an image. The example image above has about 8 pixels per inch. A real digitized image has 300.

For illustration purposes, the above image uses only about 8 pixels per inch (so 8 × 8 = 64 pixels per square inch). In reality, an image requires perhaps 50 pixels per inch to look decent to the human eye.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

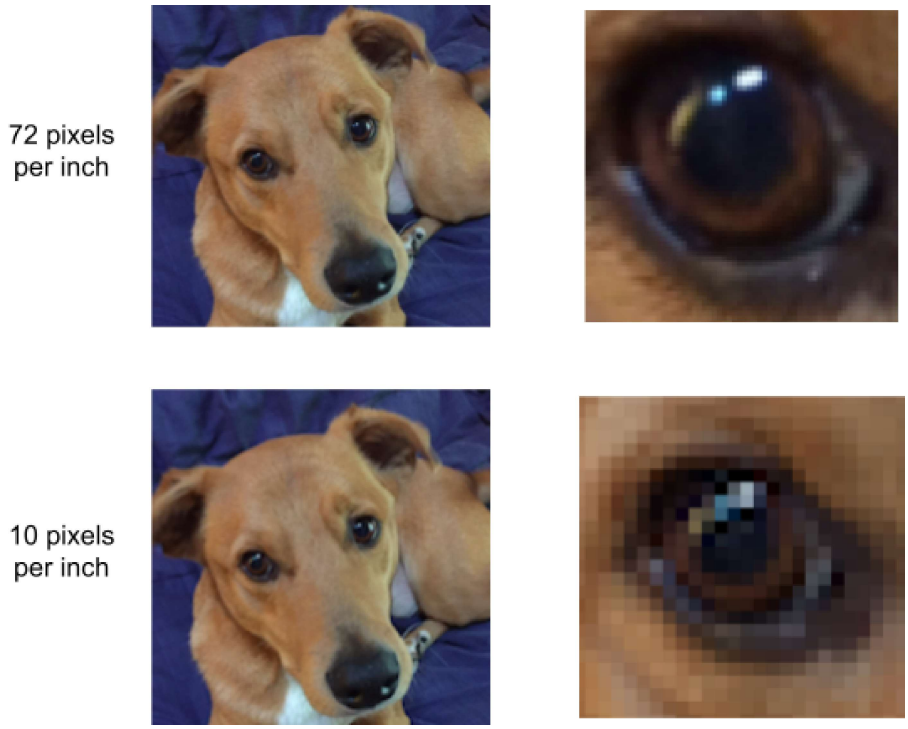
A typical camera digitizes images using more pixels per inch than needed for typical viewing sizes, just in case a user wishes to zoom in or blow up the image. Below, a photo taken with 72 pixels per inch can be zoomed into and still look OK, but a photo taken with 10 pixels per inch shows obvious poor quality when zoomed in.

A digital image file is a sequence of pixel color numbers. Due to the amount of pixels, a digital image may be large. For example, if an image has 800 x 600 = 480,000 pixels, and each pixel color number

requires 3 bytes, then the image may require 480,000 pixels × 3 bytes/pixel = 1,440,000 bytes or 1.4 megabytes. A megabyte is one million (1,000,000) bytes. Images are thus usually compressed to reduce size, losing some image quality. ***JPEG*** is a common image compression standard.

Figure 9.11.1: Cameras typically digitize images using more pixels per inch than needed for typical sizes, to support zooming or blowing up.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021



PARTICIPATION ACTIVITY 9.11.2: Digital images.

Consider the above animation.

1) For the image with 4 pixels per inch, the first row's pixel numbers would be FFFFFF, FFFFFF, FFFFFF, and FFFFFF.
   - True
   - False

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

2) For the image with 4 pixels per inch, the second row's pixel values are shown. The third row's pixel values would be FFFFFF, 00FF00, FFFFFF, and FFFFFF.

- True
- False

3) For the image with 4 pixels per inch, how many total pixels exist in one square inch?

- 8
- 16

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

4) For the image with 8 pixels per inch, how many total pixels exist in one square inch?

- 16
- 64

5) For an image with 200 pixels per inch, how many total pixels would exist in one square inch?

- 40,000
- 4 million

6) If an image has 300 pixels per inch, then a square inch has 300 × 300 = 90,000 pixels. If each image's color number uses 3 bytes, how many bytes are needed to represent a 2 inch by 3 inch image?

- 270,000
- 1.6 million

7) A photo is taken with a smart phone using 200 pixels per inch. The photo is "blown up" to be printed as a poster, such that each square inch of the poster will have 5 pixels. Can the poster be expected to be good quality?

- Yes
- No

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Video

Video is a series of slightly-differing images shown fast enough to appear continuous to humans. Each image in video is known as a ***frame***, and the number of frames per second (fps) is the ***frame rate***. Standard video uses about 24 frames per second, and each image has 525 pixel rows (known as lines). Compared to standard video, ***high-definition video*** uses more frames per second (like 60) and more lines (like 1080), as well as wider lines.

Video files can be quite large. If a single image required 1 MB, then at 30 frames per second, 100 minutes (a common movie duration) would require 100 min × 60 sec / min × 30 frame / sec × 1 MB / frame = 180,000 MB or 180 GB. Common video file formats, such as ***MP4***, ***H.264***, or ***MOV***, differ in how compression is used to reduce video file size. The key idea of video compression is to only store some frames completely, with most other frames stored as the difference from the previous frame (in a video, such differences between successive frames are typically tiny). Video compression usually loses some quality. After compression, a 100 minute movie may only require 5 to 10 GB. Most devices that record video do compression while recording, and devices that play video automatically decompress.

PARTICIPATION ACTIVITY 9.11.3: Video is just a series of slightly-differing images shown fast enough to appear continuous to humans.

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

Animation captions:

1. A video file is a series of images, each called a frame.
2. A video player app displays each frame one at a time.
3. Shown fast enough, the images appears as one continuous video.

PARTICIPATION ACTIVITY 9.11.4: Video basics.

1) Video consists of a series of images known as _____ .
   - photos
   - frames

2) For a human to not notice each frame in a video, the frame rate should be at least _____ frames per second.
   - 5
   - 24
   - 500

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

3) Because video files could be large, compression is used. The main idea of

compression is to only store the _____ a frame and the previous frame.

- difference between
- name of
- size of

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

9.12 Audio

Audio basics

Audio, or sound, is the vibration of air molecules that human ears can detect. Audio is naturally an ***analog*** signal, meaning the signal changes continuously over time, like a flowing river. Because computers can only store 0's and 1's, a computer records audio as a ***digital*** signal, meaning as a series of numbers—digital means countable like a hand's fingers (fingers are also known as "digits", hence the word "digital").

A microphone converts vibrating air into an analog electrical voltage on a wire. An audio recording app may then rapidly "sample" that voltage (44,100 samples / second is common for music), using hardware called an analog-to-digital converter to convert each sample to a number, and storing the numbers in a file. An ***audio file*** is basically a series of numbers of sampled audio voltages, along with the sampling rate. An ***audio player*** app plays an audio file such that the audio can be heard, by sending numbers one at a time to a digital-to-analog converter, at the specified rate.

PARTICIPATION ACTIVITY 9.12.1: How digital audio works.

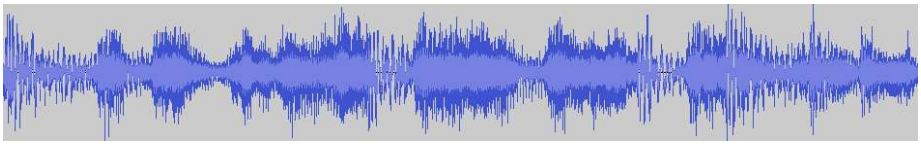
Animation captions:

1. A microphone's internal magnet vibrates due to sound (moving air molecules), causing varying voltage levels on a wire.
2. An audio recording app converts voltages to numbers (using an analog-to-digital converter), at a rate of about 44,000 times per second, and stores the numbers in a file.
3. An audio player app reads an audio file, and sends numbers to a digital-to-analog converter at the proper rate. The voltages cause a speaker to vibrate, creating sound.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Figure 9.12.1: Audio on a computer is viewed as varying voltage levels, as in this 3 second

excerpt "Welcome to the Hotel California" from the song Hotel California (viewed using Audacity).



©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.12.2: Microphone to digital audio signal.

Microphone | Song

Start microphone

Microphone is off.

Uses wavesurfer.js (katspaugh) / CC BY 3.0

PARTICIPATION ACTIVITY 9.12.3: Audio basics.

1) A microphone converts sound into an analog signal.
   - ○ True
   - ○ False

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

2) A computer records audio as an analog signal.
   - ○ True
   - ○

False

3) An audio recording app samples the voltage output of a microphone.
   - True
   - False

4) An audio file stores a song's notes.
   - True
   - False

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Storing audio

Each sampled audio value is stored using a fixed number of bits, known as ***bit depth***, commonly 16 or 24 bits. Thus, an audio file can simply be a series of 0's and 1's, such as 0000000000000001 0000000000000010 0000000000000011 0000000000000011 etc. (representing numbers 1, 2, 3, 3). Of course, values typically span the range of possible values, such as 0011110111011110 or 1111000000000011.

Audio files may be large. If each sample's number requires two bytes, the sampling rate is 44,100 samples / second, and a song is 3 minutes (180 sec), then a song's audio file would require about 180 sec × 44100 samples / sec × 2 bytes / sample = 16 MB (megabytes, or million bytes).

Audio files are usually stored in compressed form to reduce file size. As such, a typical song file might only be about 4 MB. Common compressed audio file formats include ***mp3***, ***AAC***, and ***M4A***. ***WAV*** is a common uncompressed audio format, primarily for Microsoft Windows PCs, while ***AIFF*** is a common uncompressed audio format for Apple computers.

PARTICIPATION ACTIVITY 9.12.4: Audio file storage.

1) Assume each audio sample is stored as one byte and the sampling rate is 48,000 samples / second. What is the size of a 2 minute audio recording?
   - 180 × 44,100 × 1 = 7.9 MB
   - 120 × 48,000 × 1 = 5.8 MB
   - 2 × 48,000 = 96,000 B

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

2) Which is NOT an audio file format?
   - mp3
   - WAV
   -

pdf

3) Humans can only hear frequencies up to about 20,000 Hz. To capture audio frequencies that humans can hear, an audio sample rate should be at least double that frequency. Thus, music audio frequency sampling rates should be at least _____.

- 20,000 Hz
- 40,000 Hz
- 80,000 Hz

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

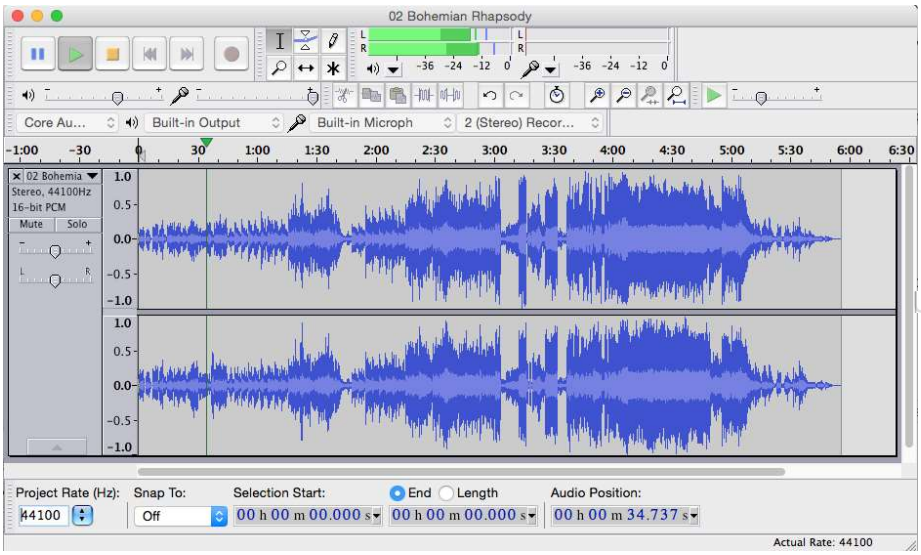
Audio apps

Popular computing devices come with audio player apps, such as ***iTunes*** (comes with Macs, iPhones, iPads, and other Apple products), the ***Music app*** or ***Windows Media Player*** (for Microsoft Windows products), and a built-in music player for Android devices.

Popular computing devices also come with apps for audio recording, such as ***Quicktime*** or ***Garage Band*** for Macs.

Apps for editing audio files such as getting excerpts or mixing music are also available, such as ***Garage Band*** for Macs, or the free ***Audacity*** app.

Figure 9.12.2: Users can create, edit, and save audio files with audio editor apps such as Audacity.



©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

PARTICIPATION ACTIVITY 9.12.5: Audio apps.

1) Most computers, including tablets and smartphones, come with an app for playing audio.
   - True
   - False

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

2) Most computers, including tablets and smartphones, come with an app for editing audio.
   - True
   - False

Exploring further:

- Digital audio (Wikipedia)
- Audio file formats (Wikipedia)
- Audio player apps (Wikipedia)
- Audio editing apps (Wikipedia)

9.13 Naming numerous bits

Modern computers may store large amounts of data, such as thousands, millions, billions, or even trillions of bits. Ex: A smartphone's memory (used by apps when running) may consist of about 1 billion bits, typically written as 1 gigabit or 1 Gb. In the metric system, the prefix "giga" means billion. Common metric prefixes are kilo (thousand), mega (million), giga (billion), and tera (trillion).

©zyBooks 10/17/21 22:09 829162
Ivan Neto

Just as humans commonly use powers of 10, computers commonly use powers of 2. A reason is because, just like neighborhoods use decimal addresses for house locations, storage devices use binary addresses for data locations. A binary address' range is a power of 2, such as 30 bits representing 2^{30} addresses. That number equals 1,073,741,824, which is very close to 1 billion, so for convenience people commonly refer to that number of bits as 1 gigabit.

But using metric prefixes for computers is not accurate. Alternative prefixes, known as ***IEC prefixes***, exist like kibi (2^{10} or 1024), mebi (2^{20} or 1,048,576), gibi (2^{30} or 1,073,741,824), and tebi (2^{40} or

1,099,511,627,776). In gibi, the gi refers to the metric prefix giga, and the bi to "binary". A gibi is abbreviated Gi, as in 1 Gib for 1 Gibibit. Likewise for other IEC prefixes.

When metric prefixes are used as in 1 Gigabit, computer folks understand those prefixes are informal and actually refer to the nearest power of 2.

Table 9.13.1: Powers of 2, and common metric and IEC prefixes used near a thousand, million, and billion. Powers of 2 to the 16 and 32 are common so are listed too.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Power of 2	Value	Metric prefix (informal)	IEC prefix
1	2		
2	4		
3	8		
4	16		
5	32		
6	64		
7	128		
8	256		
9	512		
10	1024	kilo (K): 1 kilobit or 1 Kb	kibi (Ki): 1 kibibit or 1 Kib
11	2048		
12	4096		
13	8192		
14	16384		
15	32768		
16	65536	64 Kb	64 Kib
17	131072		
18	262144		

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

19	524288		
20	1048576	mega (M): 1 megabit or 1 Mb	mebi (Mi): 1 mebibit or 1 Meb
21	2097152		
22	4194304		
23	8388608		
24	16777216		
25	33554432		
26	67108864		
27	134217728		
28	268435456		
29	536870912		
30	1073741824	giga (G): 1 gigabit or 1 Gb	gibi (Gi): 1 gibibit or 1 Gib
31	2147483648		
32	4294967296	4 Gb	4 Gib

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

Sizes often refer to bytes rather than bits, as in 128 gigabytes. The abbreviation uses a B rather b, as in 128 GB rather than 128 Gb.

To help remember the relation of powers of two and powers of ten, one may note that 2^{10} is near a thousand, 2^{20} is near a million, and 2^{30} is near a billion.

PARTICIPATION ACTIVITY 9.13.1: Metric and IEC prefixes.

Write all answers in abbreviated form. Ex: Write Kb rather than kilobit.

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

1) 1024 bits using the metric prefix is informally written 1 Kb. Write 4096 informally using the metric prefix.

Check Show answer

2) 2^{30} is close to one billion. Write 2^{30} bits informally using a metric prefix.

Check Show answer

3) 2 to the power of 32 is a common value in computers. Write 2^{32} bits using an informal metric prefix.

Check Show answer

4) 1024 bits using the IEC prefix is written 1 Kib. Write 4096 using the IEC prefix.

Check Show answer

5) 2^{30} is close to one billion. Write 2^{30} bits using the IEC prefix.

Check Show answer

6) 2 to the power of 32 is a common value in computers. Write 2^{32} bits using the IEC prefix.

Check Show answer

7) Sizes are often written using bytes rather than bits. Write 2^{16} bytes using an IEC prefix.

Check Show answer

©zyBooks 10/17/21 22:09 829162 Ivan Neto UCRCS120AEE120AChenFall2021

Exploring further:

- IEC prefixes

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021

©zyBooks 10/17/21 22:09 829162
Ivan Neto
UCRCS120AEE120AChenFall2021