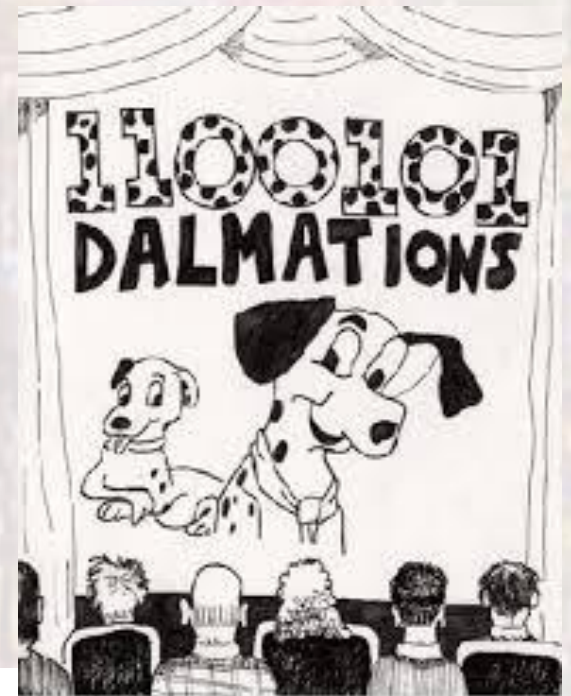


Real programmers code in binary.



Film Night at the Binary Society



www.clipartof.com - 443435

Number Systems Intro

Last updated 8/20/20

Number Systems Intro

- Base 10 (decimal)
 - The most familiar base for most people
 - ones, tens, hundreds, thousands
 - tenths, hundredths, thousandths
 - Base 10 → 10 individual digits
 - Range of individual digit: 0 → 9
 - Each position to the left of the decimal point is 10X the previous position
 - Each position to the right of the decimal point is $1/10^{\text{th}}$ the previous position

1	2	3	4	.	5	6	7
Thousands	Hundreds	Tens	Ones	decimal point	tenths	hundredths	thousandths

1	2	3	4	.	5	6	7
digit x 10^3	digit x 10^2	digit x 10^1	digit x 10^0	decimal point	digit x 10^{-1}	digit x 10^{-2}	digit x 10^{-3}

Number Systems Intro

- Base 2 (binary)
 - The most common base for digital electronics
 - ones, twos, fours, eights
 - halves, quarters, eighths
 - Base 2 → 2 individual digits
 - Range of individual digit: 0 → 1
 - Each position to the left of the binary point is 2X the previous position
 - Each position to the right of the binary point is 1/2 the previous position

1	1	0	1	.	1	0	1
Eights	Fours	Twos	Ones	binary point	Halves	Quarters	Eighths

1	1	0	1	.	1	0	1
digit x 2^3	digit x 2^2	digit x 2^1	digit x 2^0	binary point	digit x 2^{-1}	digit x 2^{-2}	digit x 2^{-3}

Number Systems Intro

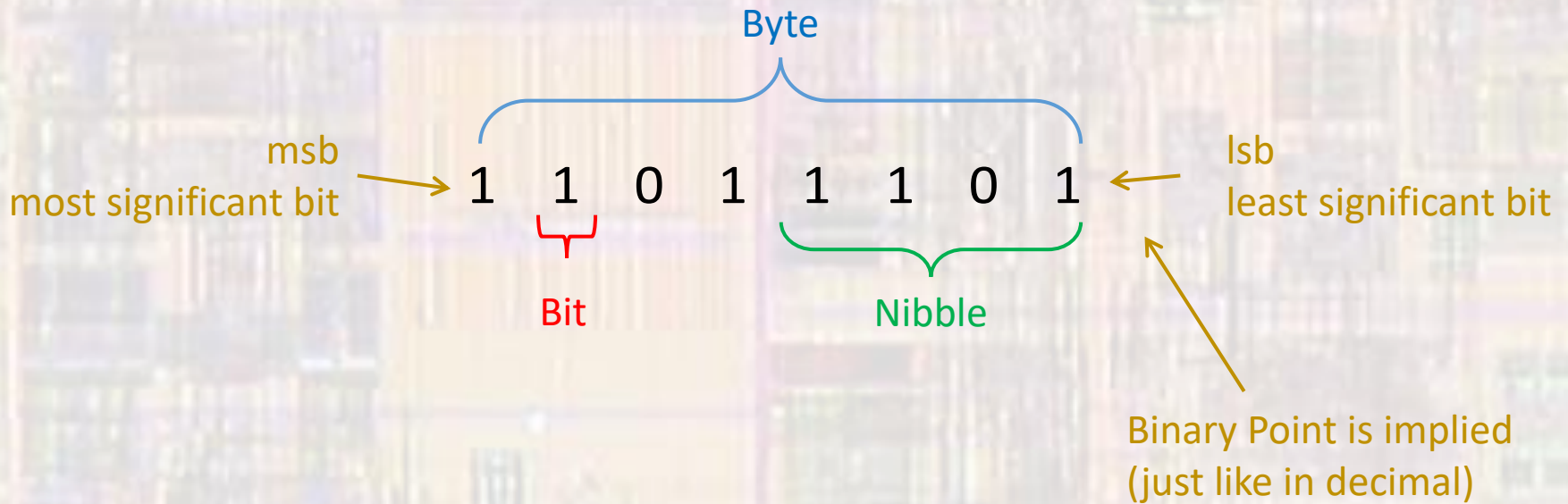
- Base 16 (hexadecimal)
 - Used as a short hand for binary
 - ones, 16s, 256s, 4096s
 - 16ths, 256ths
 - Base 16 → 16 individual digits
 - Range of individual digit: 0 → 9, A → F
 - 10=A, 11=B, 12=C, 13=D, 14=E, 15=F
 - Each position to the left of the hexadecimal point is 16X the previous position
 - Each position to the right of the hexadecimal point is 1/16 the previous position

2	B	0	E	.	3	A	2
4096s	256s	16s	Ones	hexadecimal point	16ths	256ths	4096ths

2	B	0	E	.	3	A	2
digit x 16 ³	digit x 16 ²	digit x 16 ¹	digit x 16 ⁰	hexadecimal point	digit x 16 ⁻¹	digit x 16 ⁻²	digit x 16 ⁻³

Number Systems Intro

- Binary Terminology



Number Systems Intro

- Binary Terminology

16 Bit Word

1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1

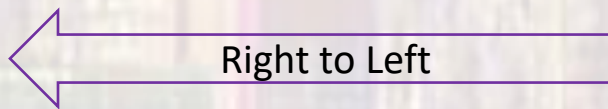
32 Bit Word

1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1

64, 128, 256, 512, 1024 Bit Words

Number Systems Intro

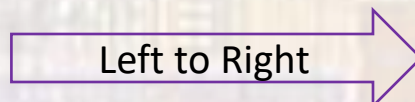
- Bit Values



Bit #	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Value	2,147,483,648	1,073,741,824	536,870,912	268,435,456	134,217,728	67,108,864	33,554,432	16,777,216	8,388,608	4,194,304	2,097,152	1,048,576	524,288	262,144	131,072	65,536	32,768	16,384	8,192	4,096	2,048	1,024	512	256	128	64	32	16	8	4	2	1

1K
1M
1G

Bit #	8	7	6	5	4	3	2	1
Value	0.00390625	0.0078125	0.015625	0.03125	0.0625	0.125	0.25	0.5



Number Systems Intro

- Exponential shorthand
 - Use of K, M, G, T is situationally dependent
 - In science and math:
 - $K = x10^3$
 - $M = x10^6$
 - $G = x10^9$
 - $T = x10^{12}$
 - In computer and digital systems:
 - $K = x2^{10} = x1,024$
 - $M = x2^{20} = x1,048,576$
 - $G = x2^{30} = x1,073,741,824$
 - $T = x2^{40} = x1,099,511,627,776$

Number Systems Intro

- More Terminology

- Assume S is an 8 bit binary number

S = 10010110

- S[7:0] = 10010110 10010110
- S[3:0] = 0110 10010110
- S[7:6] = 10 10010110
- S[5] = 0 10010110
- S[6,3] = 00 10010110
- S[1] = 1 10010110
- S[0] = 0 10010110

Number Systems Intro

- Special note on binary numbers in C programming
 - Some **but not all** compilers allow binary numbers to be represented in C code directly

95 → **0b01011101**

- To be safe and ensure our code is portables we will **NOT** use this notation.
- Binary numbers can be represented with:
 - Their decimal equivalents **95**
 - Their hexadecimal equivalents **0x5D**