

Numbers and Logic

Common Numbering Systems:

- Decimal
- Binary
- Hexadecimal

Bits, Bytes, and Nibbles

- Are you hungry yet!
- Bit
- Byte
- Nybble (or Nibble)

Bit as a Representation of state

0 = The switch is off

1 = The switch is on

Numerical Representation

1 Byte = 8 Bits

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0

Numerical Representation

Binary	Decimal
0000 0000	0
0000 0001	1
0000 1000	8
0000 0011	3

0001 0000

16

0000 0011

3

1000 0000

128

1111 1111

255

There are 10 type of people in the world

- Those who know binary
 - Those who don't

Hexadecimal (0xHH)

- Binary is Base 2
- Hexadecimal is Base 16
- 0 - 9
- A - F
- 10 = A
 - or 0x0A
- 15 = F
 - 0x0F
- 16 = 10 ($16^1 * 1$) + 0
 - 0x10
- 17 = 11 ($16^1 * 1$) + 1
 - 0x11
- 255 = FF ($16^1 * 15$) + 15
 - 0xFF

ASCII







- American Standard Code for Information Interchange
- Since everything in the computer is represented as numbers, we need a way to represent letters
- An ASCII Table is used to convert the characters to the numerical representation used to store the characters in memory

ASCII Table

ASCII (1977/1986)

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_	NUL 0000 0	SOH 0001 1	STX 0002 2	ETX 0003 3	EOT 0004 4	ENQ 0005 5	ACK 0006 6	BEL 0007 7	BS 0008 8	HT 0009 9	LF 000A 10	VT 000B 11	FF 000C 12	CR 000D 13	SO 000E 14	SI 000F 15
1_	DLE 0010 16	DC1 0011 17	DC2 0012 18	DC3 0013 19	DC4 0014 20	NAK 0015 21	SYN 0016 22	ETB 0017 23	CAN 0018 24	EM 0019 25	SUB 001A 26	ESC 001B 27	FS 001C 28	GS 001D 29	RS 001E 30	US 001F 31
2_	SP 0020 32	! 0021 33	" 0022 34	# 0023 35	\$ 0024 36	% 0025 37	& 0026 38	' 0027 39	(0028 40) 0029 41	* 002A 42	+ 002B 43	, 002C 44	- 002D 45	. 002E 46	/ 002F 47
3_	0 0030 48	1 0031 49	2 0032 50	3 0033 51	4 0034 52	5 0035 53	6 0036 54	7 0037 55	8 0038 56	9 0039 57	: 003A 58	; 003B 59	< 003C 60	= 003D 61	> 003E 62	? 003F 63
4_	@ 0040 64	A 0041 65	B 0042 66	C 0043 67	D 0044 68	E 0045 69	F 0046 70	G 0047 71	H 0048 72	I 0049 73	J 004A 74	K 004B 75	L 004C 76	M 004D 77	N 004E 78	O 004F 79
5_	P 0050 80	Q 0051 81	R 0052 82	S 0053 83	T 0054 84	U 0055 85	V 0056 86	W 0057 87	X 0058 88	Y 0059 89	Z 005A 90	[005B 91	\ 005C 92] 005D 93	^ 005E 94	_ 005F 95
6_	` 0060 96	a 0061 97	b 0062 98	c 0063 99	d 0064 100	e 0065 101	f 0066 102	g 0067 103	h 0068 104	i 0069 105	j 006A 106	k 006B 107	l 006C 108	m 006D 109	n 006E 110	o 006F 111
7_	p 0070 112	q 0071 113	r 0072 114	s 0073 115	t 0074 116	u 0075 117	v 0076 118	w 0077 119	x 0078 120	y 0079 121	z 007A 122	{ 007B 123	 007C 124	}	~ 007E 126	DEL 007F 127

Basic Computer Logic

- AND 
- OR 
- NOT 
- NAND 
- NOR 
- XOR 

1 = True

0 = False

Truth Tables

- The computer is constantly evaluating values.
- We can use truth table to help visualize the outcome

And

&&



AND (&&)	TRUE	FALSE
TRUE	TRUE	FALSE
FALSE	FALSE	FALSE

And - &&



And &&	1	0
1		
0		

OR - ||




OR 	True	False
True	True	True
False	True	False

OR - ||



OR	1	0
1		
0		

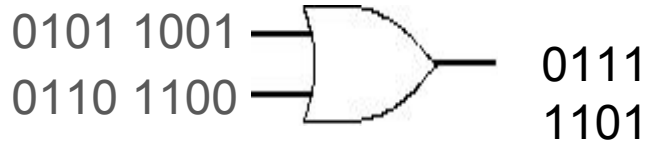
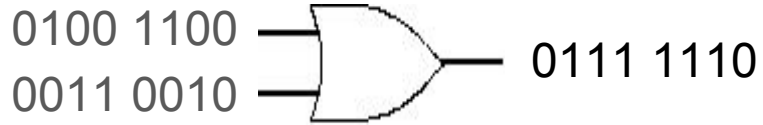
Not

- Not X
- Not 1 = 0
- Not 0 = 1
-  0111 0011 (Bitwise Not)
 - 1000 1100

Bitwise And



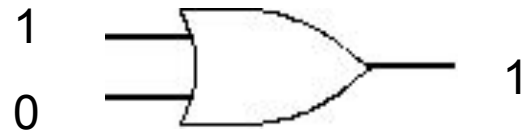
Bitwise OR



And



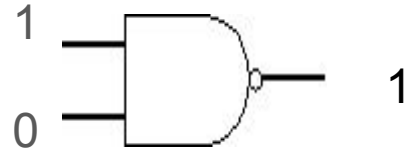
Or



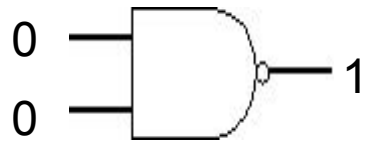
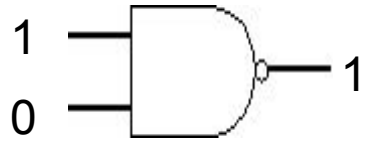
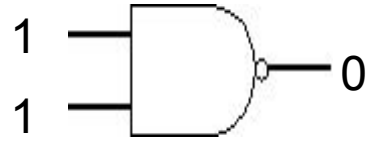
NAND



- NAND is Not And



NAND



NOR - Not OR



XOR - Exclusive Or

