CMP 101
BINARY NUMBERS AND OTHER NUMBERING SYSTEMS

Dr. S.A. Arekete

Inside Computers - Bits and Pieces

“Your computer successfully creates the illusion that it contains photographs, letters, songs, and movies. All it really contains is bits, lots of them, patterned in ways you can't see. Your computer was designed to store just bits - all the files and folders and different kinds of data are illusions created by computer programmers.”

(Hal Abelson, Ken Ledeen, Harry Lewis, in "Blown to Bits")

Inside Computers - Bits and Pieces..

- Basically, computer instructions perform operations on groups of bits.
- A bit is either on or off, like a light bulb.

Inside Computers - Bits and Pieces..

- Figure 1.2_a shows an open switch and a light bulb that is off - just like a transistor in a computer represents a bit with the value: zero.
- Figure 1.2_b shows the switch in the closed position and the light bulb is on, again just like a transistor in a computer representing a bit with the value: one.

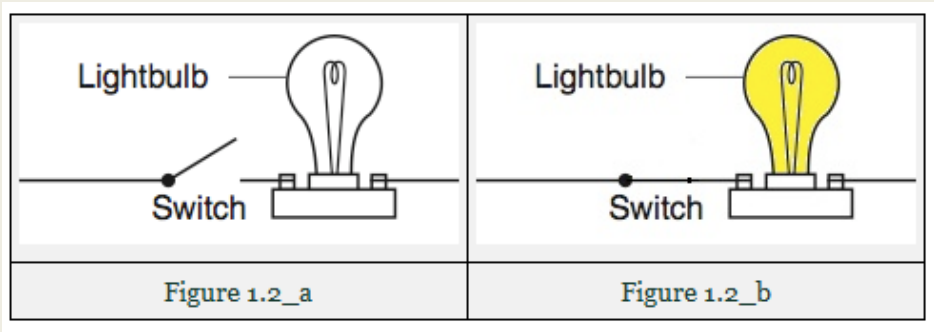

Figure 1.2_a Figure 1.2_b

Microprocessors
Inside Computers - Bits and Pieces..

- A microprocessor, which is the heart of a computer, is very primitive but very fast.
  - *It takes groups of bits and moves around their contents, adds pairs of groups of bits together, subtracts one group of bits from another, compares a pair of groups, etc... - that sort of stuff.*
- Inside a microprocessor, at a very low level, everything is simply a bunch of switches, also known as bits - things that are either on or off!
- Let's explore how groups of bits can be used to form numbers.

Microprocessors
Inside Computers - Bits and Pieces..

Binary

There are only 10 (2) different kinds of people in the world:

those who know binary and those who don't.

- Anonymous

Binary..

- Bits are coined from ***Bi***nary Digi***ts***
- Computers are full of millions of bits that are either on or off.
- The way we talk about the value of a bit in the electrical engineering and computer science communities is first as a logical value (true if on, false if off) and secondly as a binary number (1 if the bit is on and 0 if it's off).
- Most bits in a computer are manipulated in groups, so we humans need a way to describe groups of bits, things/objects a computer manipulates.
- Today, bits are most often grouped in quantities of 8, 16, 32, and 64.

Binary and Decimal

- Think about how you write down sequential numbers starting with zero: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, etc…
- Our decimal number system has ten symbols.
  - *In this sequential series, when we ran out of symbols, we combined them.*
- You learned how to do this so long ago, in grade school, that today you just naturally think in terms of single digit numbers, then tens, hundreds, thousands, etc…
  - *The decimal number 1234 is one thousand, two hundreds, three tens, and four units.*
  - *Note that 1234 actually means* $(1 \times 10^3)+(2 \times 10^2)+ (3 \times 10^1)+ (4 \times 10^0)= (1 \times 1000)+(2 \times 100) + (3 \times 10)+(4 \times 1)$

Binary System and Other Numbering Systems

- So, how does the binary number system used inside computers work?
- Well, with only two symbols, we would write the same sequential numbers as above: 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011.
- The decimal number 1234 in binary is 10011010010.
- How do we arrive at this?

1024	512	256	128	64	32	16	8	4	2	1
1	0	0	1	1	0	1	0	0	1	0

Binary System and Other Numbering Systems

- Since even reasonable numbers that we use all the time make for very long binary numbers, the bits are grouped in 3s and 4s which are simple to convert into numbers in the octal and hexadecimal number systems. For octal, we group three bits together.
- Take the binary equivalent of decimal 1234, 10011010010, and put spaces in between each group of three bits - starting at the right and going left.
- 10011010010 = 10 011 010 010 = 2322 in Octal

Inside Computers - Bits and Pieces..

- When you group four bits together and use sixteen symbols (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F) as their abbreviations, you have a hexadecimal representation.
- 10011010010 = 100 1101 0010 = 4D2
- As you continue to explore how computers work, you'll hear more about numbers expressed in octal and hex; these are just more manageable representations of binary information - *the digital world*.

Decimal Number	in Binary	in Octal	in Hex
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

Table 1.2

Grouping of Bits

- A group of eight bits has binary values 00000000 through 11111111, or expressed in decimal 0 through 255.
- A group of sixteen bits has binary values 0000000000000000 through 1111111111111111, or decimal 0 through 65535.
- The range of decimal values for a group of 32 bits is 0 through 4,294,967,295.
- The range of decimal values for a group of 64 bits is 0 through 18,446,744,073,709,551,615 - *or almost eighteen and a half quintillion*.

Grouping of Bits

Number of Bits	Unsigned Maximum Value	Signed Minimum Value	Signed Maximum Value
8	255	-128	127
16	65535	-32768	32767
32	4,294,967,295	-2,147,483,648	2,147,483,647
64	18,446,744,073,709,551,615	-9,223,372,036,854,775,808	9,223,372,036,854,775,807

Table 1.3

Symbols as Bits - ASCII Characters

- Ok, so numbers are simply groups of bits. What other objects will the computer's instructions manipulate? How about the symbols that make up an alphabet?
- It should come as no surprise that symbols that make up alphabets are just numbers, groups of bits, too. But how do we know which numbers are used to represent which symbols, or characters as I'm going to call them from this point on?
- It's all about standards. In these lessons, we will use the American Standard Code for Information Interchange (ASCII) standard. It is so ubiquitous (general or universal) that it even has its own web page, www.asciitable.com.

Symbols as Bits - ASCII Characters..

- Let's walk through a couple of examples, entries in the table. Here are some characters, their decimal value and their binary value which is then transformed into an octal number.
- Uppercase 'A' = decimal 65 = binary 01000001 = 01 000 001 = octal 101
- Uppercase 'Z' = decimal 90 = binary 01011010 = 01 011 010 = octal 132
- The digit '1' = decimal 49 = binary 00110001 = 00 110 001 = octal 061

ASCII Character	in Binary	in Octal	in Decimal	in Hex
space	00100000	040	32	20
(	00101000	050	40	28
)	00101001	051	41	29
*	00101010	052	42	2A
0	00110000	060	48	30
1	00110001	061	49	31
2	00110010	062	50	32
9	00111001	071	57	39
A	01000001	101	65	41
B	01000010	102	66	42
C	01000011	103	67	43
Z	01011010	132	90	5A
a	01100001	141	97	61
b	01100010	142	98	62
c	01100011	143	99	63
z	01111010	172	122	7A

Table 1.4