Number Systems

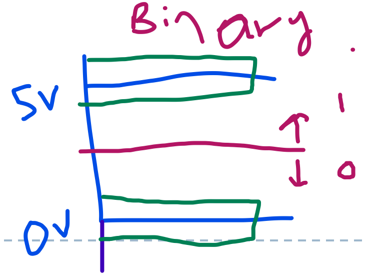
Dr. Sapumal Ahangama
Department of Computer Science and Engineering

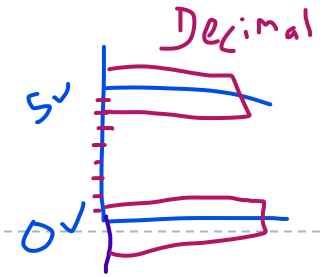
COUNTING AND ARITHMETIC

- Decimal or base 10 number system
  - Origin: counting on the fingers
  - "Digit" from the Latin word digitus meaning "finger"

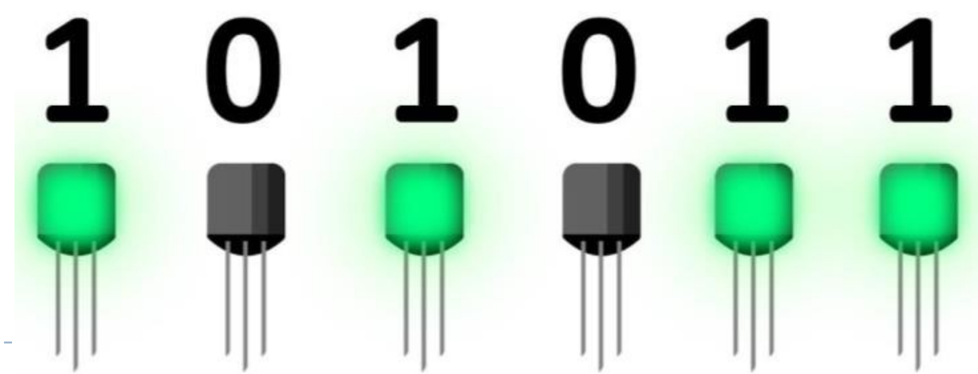
- Base: The number of different digits including zero in the number system
  - Decimal (Base10) -10 digits, 0 through 9
  - Binary (Base2) - 2 digits, 0 and 1
  - Octal (Base8) – 8 digits, 0 through 7
  - Hexadecimal (Base16)-16 digits, 0 through F
    - Examples: 1010=A; 1110=D

WHY BINARY?





- Early computer design was decimal
  - Mark I and ENIAC
- John von Neumann proposed binary data processing (1945)
  - Simplified computer design
  - Used for both instructions and data
- Natural relationship between on/off switches and calculation using Boolean logic



KEEPING TRACK OF THE BITS

- Bits commonly stored and manipulated in groups
  - 8 bits = 1 byte
  - 4 bytes = 1 word (in many systems –32 bit word size)

- Number of bits used in calculations
  - Affects accuracy of results
  - Limits size of numbers manipulated by the computer

$$
\begin{array}{r}
1\ 1\ 0\ 1 \\
+\ 1\ 1\ 1\ 1 \\
\hline
\end{array}
$$

NUMBERS: PHYSICAL REPRESENTATION

- Consider you want to represent value “5”.
  - Different numerals and number systems
  - Roman: V
  - Arabic: 5
- Different bases to represent the same number
  - 5_{10}
  - 101_2
  - 12_3

NUMBERS

- Example 527_{10}

$$527 = 5 \times 10^2 + 2 \times 10^1 + 7 \times 10^0$$

100's place 10's place 1's place

Place	10^2	10^1	10^0
Value	100	10	1
Evaluate	5 x 100	2 x 10	7 x1
Sum	500	20	7

NUMBERS

- Example 624_8

$624_8 = 404_{10}$

64's place 8's place 1's place

Place	8^2	8^1	8^0
Value	64	8	1
Evaluate	6 x 64	2 x 8	4 x 1
Sum for Base 10	384	16	4

NUMBERS

- Example 101111_2

1

2

4

8

32

47

CONVERSION BETWEEN DIFFERENT BASES

- Convert from Base 10 to Other Bases:
  - Base 10 to Base 2
  - Base 10 to Base 8
  - Base 10 to Base 16

$42_{10} = 101010_2$

Power / Base	6	5	4	3	2	1	0
2	64	32	16	8	4	2	1
		1	0	1	0	1	0
Integer		42/32 = 1	10/16 = 0	10/8 = 1	2/4 = 0	2/2 = 1	0/1 = 0
Remainder		10	10	2	2	0	0

CONVERSION BETWEEN DIFFERENT BASES

- Convert Between Different Bases
  - If not base 10
    - Convert to decimal
    - Convert back to needed base
  - If base 10
    - Conversion in previous step
  - But, Conversion between base 2 and 8 and 16?
    - 11010111011000 to base 16 → 4 godawal walata kadagena hex walin liyanna
    - 275331_8 to base 2

BASE OR RADIX

- Base: The number of different symbols required to represent any given number

- The larger the base, the more numerals are required
  - Base 10: 0,1,2,3,4,5,6,7,8,9
  - Base 2: 0,1
  - Base 8: 0,1,2,3,4,5,6,7
  - Base 16: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

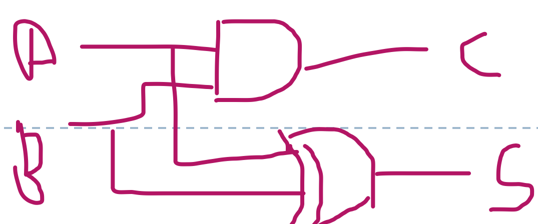
- For a given number, the larger the base the more symbols required, but the fewer digits needed
  - 65_{16} to base 10 $= 101_{10}$
  - 65_{16} to base 8 $= 145_8$
  - 65_{16} to base 2 $= 1100101_2$

BINARY ADDITION AND BOOLEAN LOGIC

- Adds two one-bit binary numbers (A and B)

- The output is the sum of the two bits (S) and the carry (C)

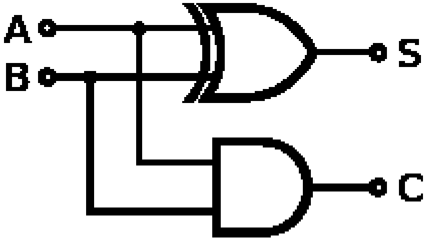
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



BINARY ADDITION AND BOOLEAN LOGIC

- Half Adder

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



BINARY ADDITION AND BOOLEAN LOGIC

$$
\begin{array}{r}
\color{red}{1\;1\;1\;1} \\
1\;1\;0\;1\;1\;1 \\
+\quad 0\;1\;1\;1\;0\;0 \\
\hline
1\;0\;1\;0\;0\;1\;1
\end{array}
$$

- Explain the Full Adder

Exam

home work widihata balanna

BINARY MULTIPLICATION

A	B	M
0	0	0
0	1	0
1	0	0
1	1	1

AND gate

BINARY MULTIPLICATION

- Example

```
       1010
   ×   1011
-----------
       1010
      1010
     0000
    1010
-----------
    1101110
-----------
```

BINARY MULTIPLICATION

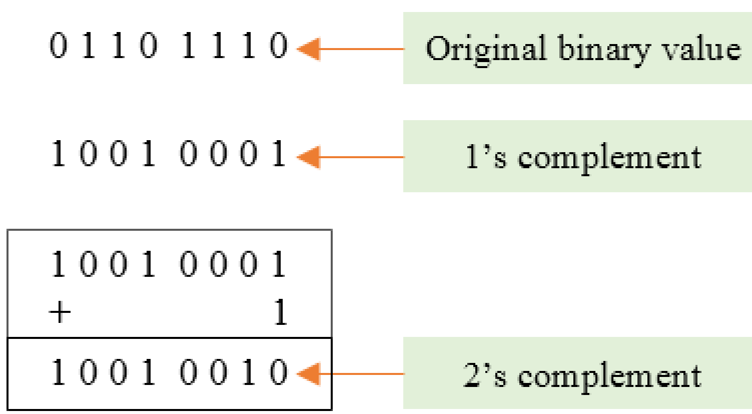
- Multiply the multiplicand by one bit of the multiplier at a time
- Result of the partial product for each bit is placed in such a manner that the LSB is under the corresponding multiplier bit.
- Then add partial products are added to get the complete product

$$
\begin{array}{r}
1010 \\
\times\ 1011 \\
\hline
1010 \\
1010 \\
0000 \\
1010 \\
\hline
1101110 \\
\hline
\end{array}
$$

1s & 2S COMPLEMENT

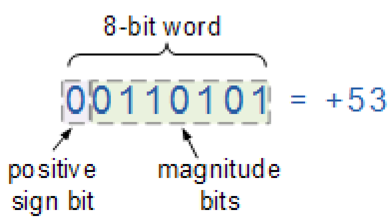
- 1's complement:
  - Another binary number obtained by toggling all bits in it.

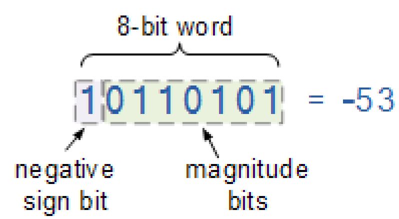
- 2's complement:
  - 1 added to the 1's complement of the binary number.



Negative Numbers

- Use a sign bit
  - 0 for positive, 1 for negative





- Using 2's complement:
  - +5 = 00000101
  - -5 = 11111011
  - 11111011 = -128 + 64 + 32 + 16 + 8 + 0 + 2 + 1 = -5

Fractions

- Example: 0.2589_{10}

$.2589_{10}$

Place	10^{-1}	10^{-2}	10^{-3}	10^{-4}
Value	1/10	1/100	1/1000	1/10000
Evaluate	2 x 1/10	5 x 1/100	8 x 1/1000	9 x1/1000
Sum	.2	.05	.008	.0009

$.101011_2 = 0.671875_{10}$

Place	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
Value	1/2	1/4	1/8	1/16	1/32	1/64
Evaluate	1 x 1/2	0 x 1/4	1x 1/8	0 x 1/16	1 x 1/32	1 x 1/64
Sum	.5		0.125		0.03125	0.015625

Fractions

- A fractional number that can be represented exactly in one number base may be impossible to be exactly represented in another base

- 0.1_{10} in binary?
  - $=0.000110011001 1_2 \ldots$

- $0.33333_{10} \ldots$ in base 3?
  - $=0.1_3$

THANK YOU