ශ්‍රී ලංකා විවෘත විශ්වවිද්‍යාලය
இலங்கை திறந்த பல்கலைக்கழகம்
The Open University of Sri Lanka

EEX3373 Communication and Computer Technology

Day School #1

Welcome to Day School #1

- Rules for online Day Schools
  - Mute the mic, unmute only when you are speaking
  - Answer the questions as soon as possible
  - Need to be interactive

Announcements

- Check the LMS Regularly.
  - Check the announcements section always (We may not be able to message you about all the activities all the time).
  - Use LMS messaging, LMS Discussion forums, etc.. to communicate with academics and others.
- Before coming to the Day School.
  - You have to read relevant sessions in the course materials.
  - Refer to the other materials (Web pages, videos, books, etc..) given in the LMS under the particular session.
  - Answer the questionnaires, if available.
- Deadlines are very important.
  - Always complete activities on or before the deadline (We may not be able to extend deadlines).

The Course Aim

To provide knowledge on fundamentals of computer technology, networking and communication.

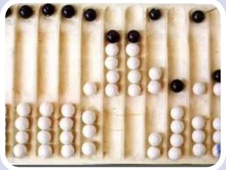
Session Outcomes

- Evolution of computers.
- Technologies in computer generations and computer models.
- Number systems.
  - Number conversions among different systems.
  - Simple arithmetic calculations.
  - Concept of Boolean algebra.
  - Fixed point and floating-point representation of numbers.

Sessions to be covered in this Day School (course material)

1. Session 01: Evolution of computers
2. Session 02: Arithmetic in computers: Whole numbers
3. Session 03: Arithmetic in computers: Fractional numbers

Evolution of Computers



3000 BC
Babylonians
Abacus

1200 AD
Unknown
Chinese abacus

1600
Unknown
Japanese abacus

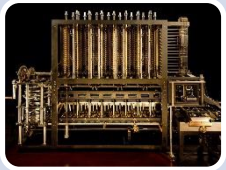
1600
John Napier
Napier's Bones

1621
William Oughtred
Slide Rule

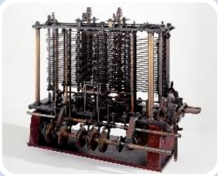
1623
Wilhelm Schickard
Schickard's Calculator

1801-1804
Joseph Jacquard
Jacquard Loom

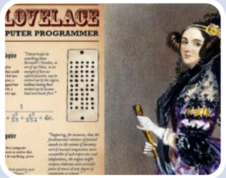
1820
Thomas de Colmar
De Colmar's Arithometer



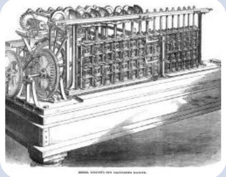
1822-1833
Charles Babbage
Difference Engine



1834
Charles Babbage
Analytical Engine



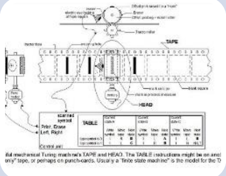
1848
Ada Lovelace
world's first computer program



1853
Per Georg Scheutz and Edvard
world's first printing calculator

1890
Herman Hollerith
IBM100 punch-card system to U.S. Census

1931
Vannevar Bush
Differential Analyzer



1936
Alan Turing
Turing machine

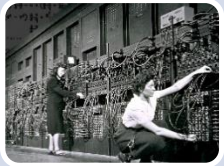
1937
John Vincent Atanasoff
first electric-only computer

1939
David Packard and Bill Hewlett
Hewlett Packard (HP) company

1941
Konrad Zuse
Z3 machine

Evolution of Computers Cont..

1941
Atanasoff and Clifford Berry
Atanasoff-Berry



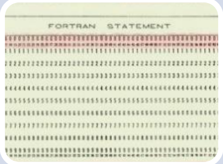
1945
John Mauchly and J. Presper Eckert
ENIAC

1946
Mauchly and Presper
UNIVAC

1947
Shockley, Bardeen and Brattain
Transistor

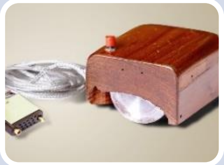
1949
A team at the University of Cambridge
EDSAC

1953
Grace Hopper
first computer language



1954
John Backus and his team
FORTRAN programmin g language

1958
Jack Kilby and Robert Noyce
Integrated Circuit



1968
Douglas Engelbart
Prototype of the modern computer

1969
Ken Thompson, Dennis Ritchie
UNIX

1970
Intel
Intel 1103, the first DRAM chip

1971
Alan Shugart and team
Floppy Disk

1972
Ralph Baer
Magnavox Odyssey

1973
Robert Metcalfe
Ethernet

1977
Commodore International
PET(MOS 8-bit 6502 microproces

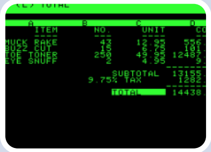
1975
MITS
Altair 8080 (world's first minicomput er)

1976
Steve Jobs and Steve Wozniak
Apple I

1977
Radio Shack
Trash 80

Evolution of Computers Cont..

1977
Jobs and Wozniak
Apple II



1978
Dan Bricklin and Bob Frankston
VisiCalc (first

1979
MicroPro International
WordStar

1981
IBM
Acorn

1983
Apple
Apple Lisa

1984
Apple
Apple Macintosh

1985
Microsoft
Windows

1989
Tim Berners-Lee
World Wide Web

1993
Intel
Pentium microprocessor advances

1996
Sergey Brin and Larry Page
Google search

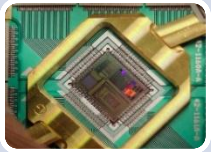
1997
Vic Hayes
Wi-Fi

2001
Apple
Mac OS X

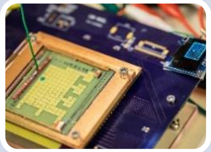
2003
AMD
Athlon 64, the first 64-bit processor

2010
Apple
iPad

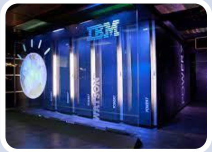
2011
Google
Chromebook



2016
first reprogrammable quantum computer



2017
DARPA
"Molecular Informatics" program

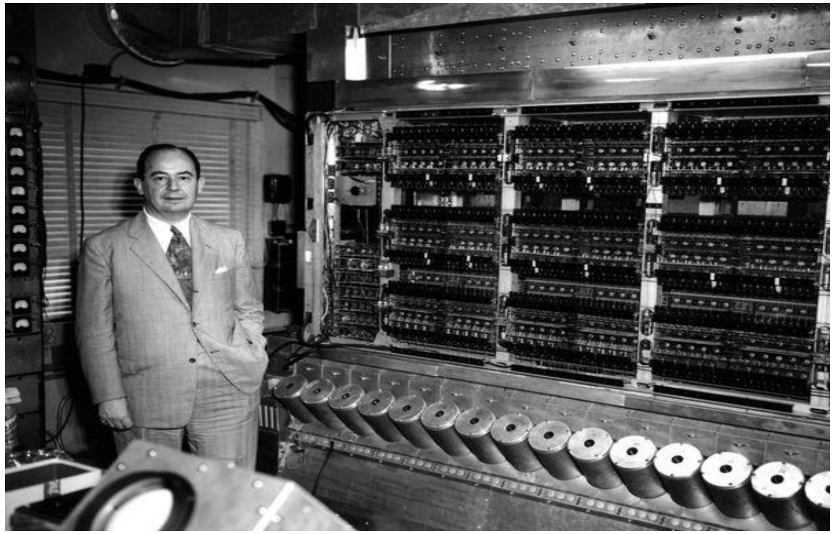


To Present

Computer Generations

-First Generation

When	Inventor	Name of device	Technical Details
1937-1942	John V. Atanasoff and Clifford E. Berry	Atanasoff-Berry Computer (ABC)	First computer prototype to use **vacuum tubes** instead of mechanical switches, Used the **binary number system** like a modern computer.
1939	Howard Aiken (IBM sponsored)	Harvard Mark 1	**Electromechanical relay-**based computer with many moving parts, used **decimal number system**. [known as Harvard Model]
1943-1945	John W. Mauchly and J. Presper Eckert	ENIAC (Electronic Numerical Integrator and Computer)	**World's first general purpose electronic digital computer** using vacuum tubes, designed to calculate trajectory tables for the US Army, but wasn't finished until shortly after the war, **programmed via switches and patch cables**. 30 tons, 1500 square feet, 18000 vacuum tubes, University of Pennsylvania. [Designer: John Von Neumann]
1951	Eckert-Mauchly Computer Corp.	UNIVAC	**First commercially successful digital computer**, a big pocket calculator, vacuum tubes, also used magnetic tape for storage, took punched cards too, marketed by **Remington-Rand** sold UNIVAC, very expensive. http://www.letsfindout.com/subjects/space/univac.html



Computer Generations Cont..

-Second Generation

When	Inventor	Name of device	Technical Details
1951	U.S. Air Force's	Whirlwind	Computer-controlled real-time defence system
1952	John Von Neumann	EDVAC	Electronic Discrete Variable Computer
1960	Digital Equipment	PDP-1	An interactive computer with CRT and keyboard. Its big screen inspires MIT students to write the world's first computer game.
Some other examples are IBM 1620, IBM 7094, CDC 1604, CDC 3600, UNIVAC 1108			

Then in the late 1950s, the vacuum tubes were replaced with **transistors**

- higher processing speeds.
- reliable
- smaller in size
- generated less heat
- consumed less electrical power

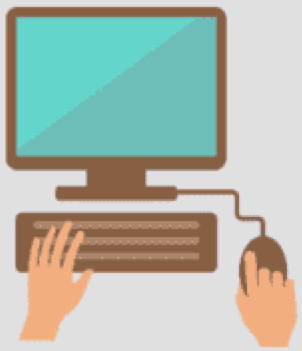
when compared to first generation computers

Computer Generations Cont..

-Third Generation

THIRD GENERATION FEATURES

- IC used
- More reliable in comparison to previous two generations
- Smaller size
- Generated less heat
- Faster
- Lesser maintenance
- Costly
- AC required
- Consumed lesser electricity
- Supported high-level language

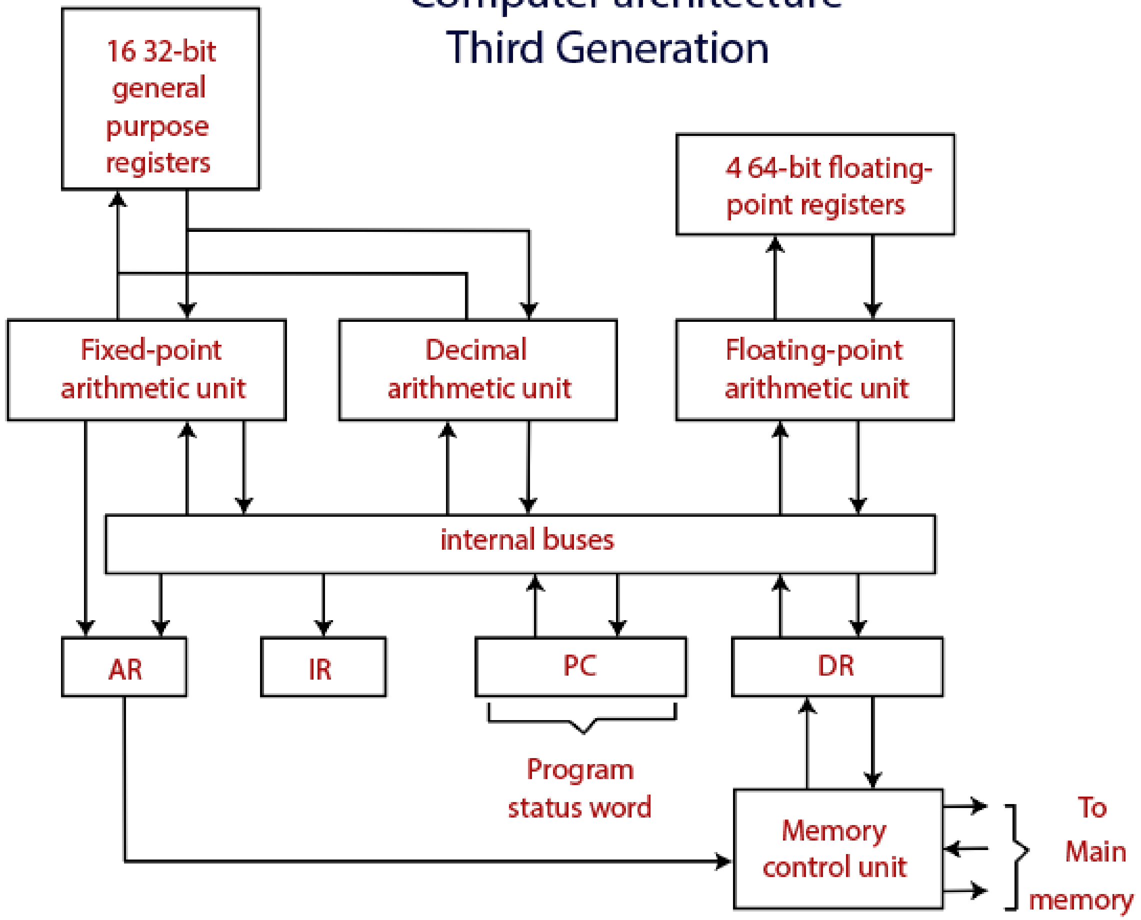


Period: 1965 - 1971

Technology Used: Integrated Circuits

Examples: IBM 360 series, Honeywell 6000 series, PDP, etc.

Computer architecture
Third Generation



Computer Generations Cont..

-Fourth Generation

Microcomputers were invented in the mid of 1970s based on - very large-scale integrated circuits (VLSI).
Perform calculations in Picoseconds

- Less expensive
- Portable
- Smaller size
- Internet connectivity
- Less heat generation
- Used all types of high-level languages
- Less maintenance cost
- Computers were more easily available to everyone; the PCs
- Computer, mouse and GUI made computers more user friendly

When	Inventor	Name of device	Technical Details
1971	Ted Hoff (INTEL)	Intel 4004 	First "Fourth Generation Computer", i.e. first microprocessor. That is, the first complete processor on a single chip built using integrated circuit technology. Followed by 8008, 8085, 8080, 8086, 8088, 80286, 80386, 80486, 80586 (a.k.a Pentium), Pentium II, Pentium III, Pentium IV.
1978	Steve Wozniak and Steve Jobs	Apple II	highly successful, mass produced home personal computer with colour graphics and useful software (VisiCalc spreadsheet)
1981	IBM	IBM PC (DOS)	Predecessor of the most popular personal computer platform in use today.
1984	Apple Computer	Macintosh	First commercially successful computer with a GUI, specifically in desktop publishing, based on Motorola 68000 microprocessor

Computer Generations Cont..

-Software Generations

Operating Systems

Operating systems in 1960s	
1961	The dawn of minicomputers
1962	Compatible Time-Sharing System (CTSS) from MIT
1964	IBM System/360
1969	The UNIX Time-Sharing System from Bell Telephone Laboratories.
Supported OS Features by 1970s	
	Multi User and Multi-tasking was introduced.
Accomplishments after 1970	
1971	Intel announced the microprocessor
1972	IBM came out with VM (Virtual Machine)
1973	UNIX 4th Edition was published
1974 The Personal Computer Age begins	
1974	**Bill Gates** and Allen wrote BASIC for the Altair
1976	Apple II
1981	IBM introduced the IBM PC
1983	Microsoft began to develop MS-Windows
1984	Apple Macintosh came out
1991	GNU/Linux was introduced
1993	Arrival of Windows NT
2008	Arrival of Android OS

Programming languages

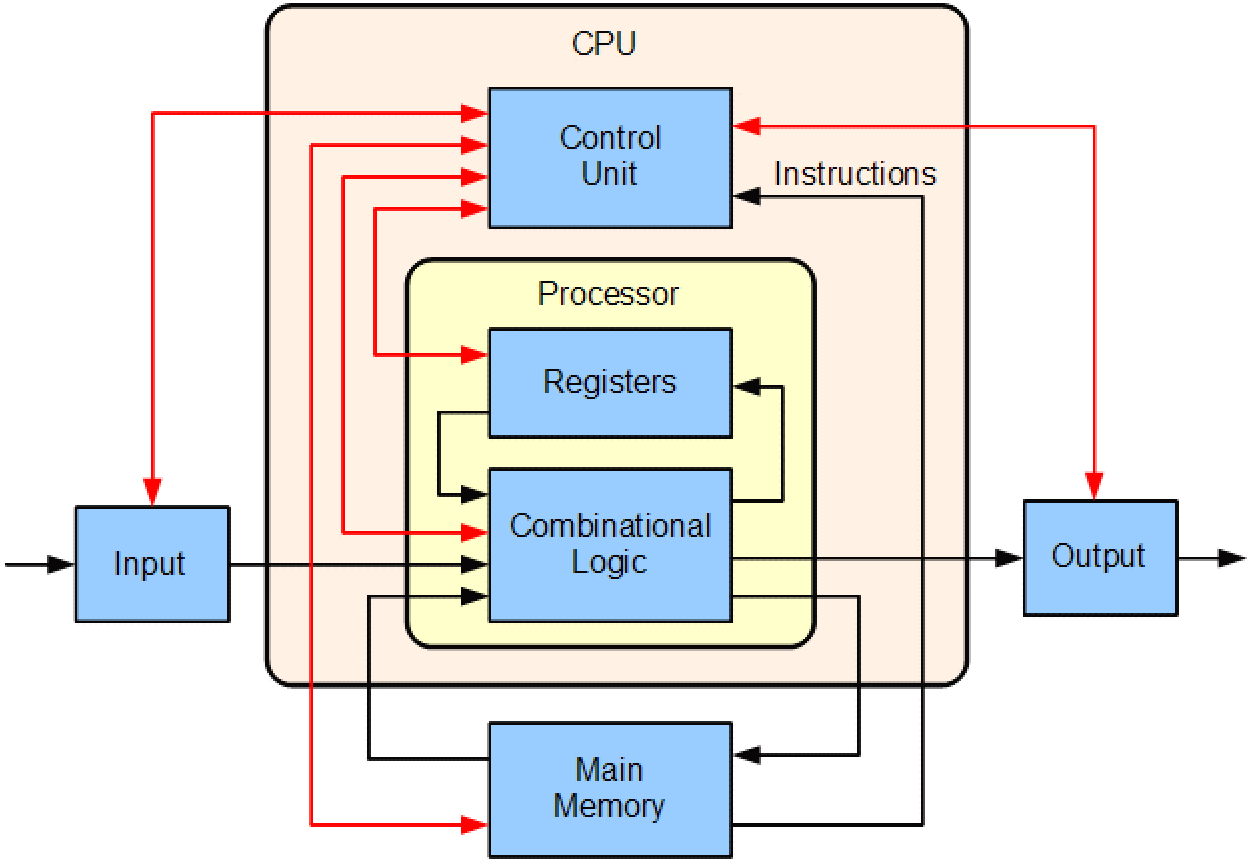
- **First generation:** Machine language.
- **Second generation:** Low-level programming languages such as Assembly language.
- **Third generation:** Structured high-level programming languages such as C, COBOL, FORTRAN, BASIC and PASCAL.
- **Fourth generation:** The concept of 4th generation languages (4GL) was developed

Computer Models

-What is Computer Architecture

Computer architecture, **structure of a digital computer, encompassing the design and layout of its instruction set and storage registers**

-Britannica



Computer Models Cont..

-Von Neumann vs Harvard

HARVARD ARCHITECTURE	VON-NEUMANN/PRINCETON ARCHITECTURE
This architecture consists of separate program and code memory.	It consists of common program and code memory.
Execution of instructions are faster, because fetching of code and data are separate.	Execution of instructions is relatively slow ,because not possible to fetch code and data simultaneously.
Execution of instruction takes less instruction (machine) cycle.	Execution of instruction takes more instruction (machine) cycle.
Use RISC processor	Use CISC processor
This is greater amount of parallelism	This is no need to have parallelism
Chip design is complex because of separate memory.	The largest advantages is that it simplifies the chip design because only one memory is accessed.

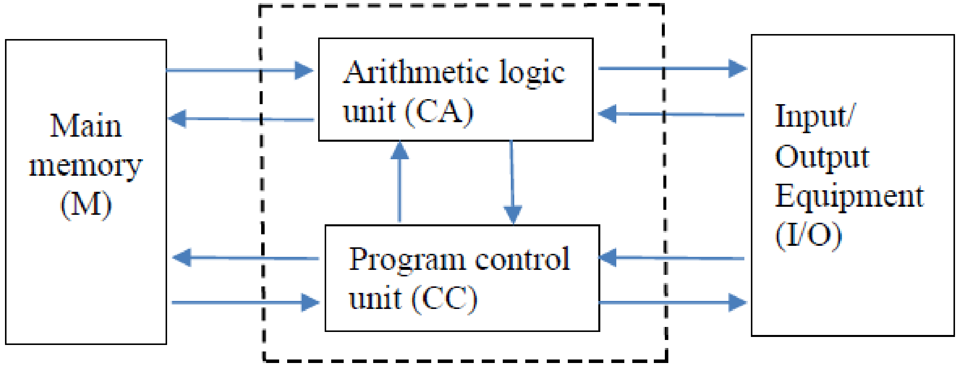


Figure 1.2: Von Neumann architecture

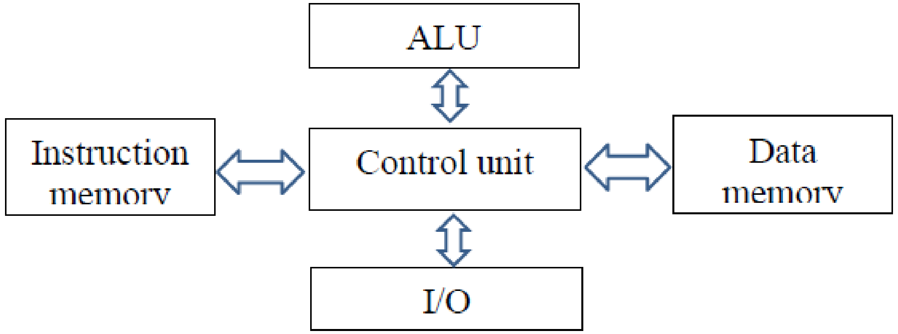


Figure 1.3: Harvard architecture

Development of Computers

-Graphical User Interface (GUI)

Graphical user interface (GUI), a computer program that enables a person to communicate with a computer through the use of symbols, visual metaphors, and pointing devices.

-Britannica

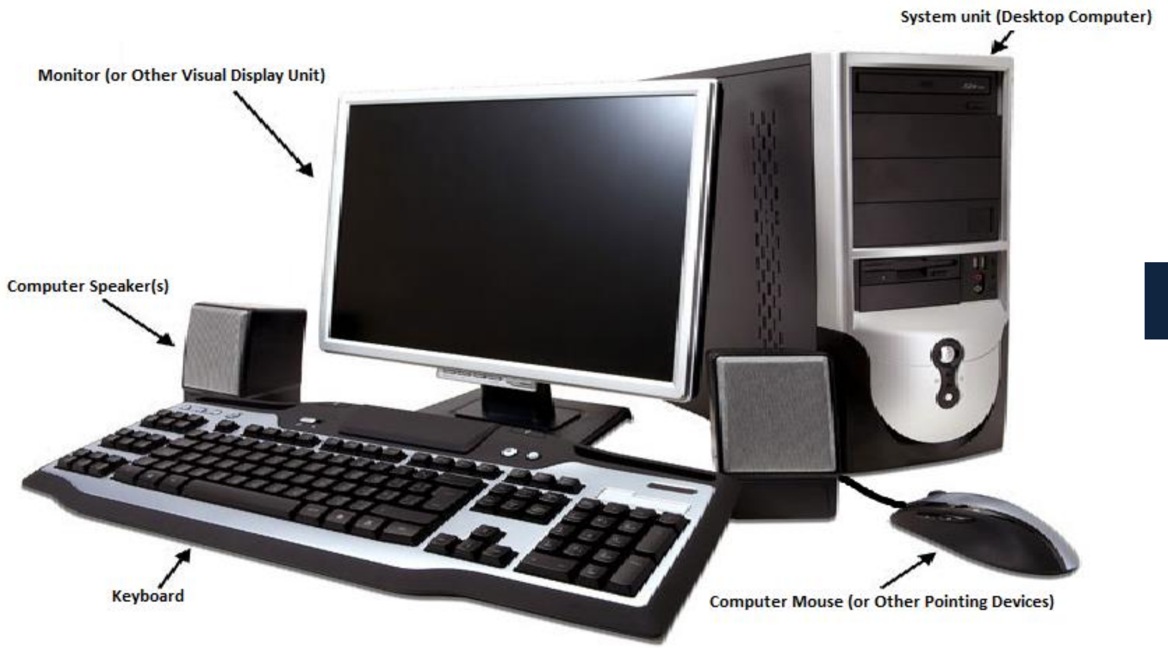
- GUI technology has made a revolution by replacing the command line operating systems in early computers.

- First introduced in 1970s - became popular with Apple Macintosh in 1980. desktop,
  - windows,
  - pointer,
  - pointing device,
  - icons,
  - menus.

Development of Computers Cont..

-Laptop Computers

- Laptop computers are the portable computers
- Size of the computer and the peripherals connected.



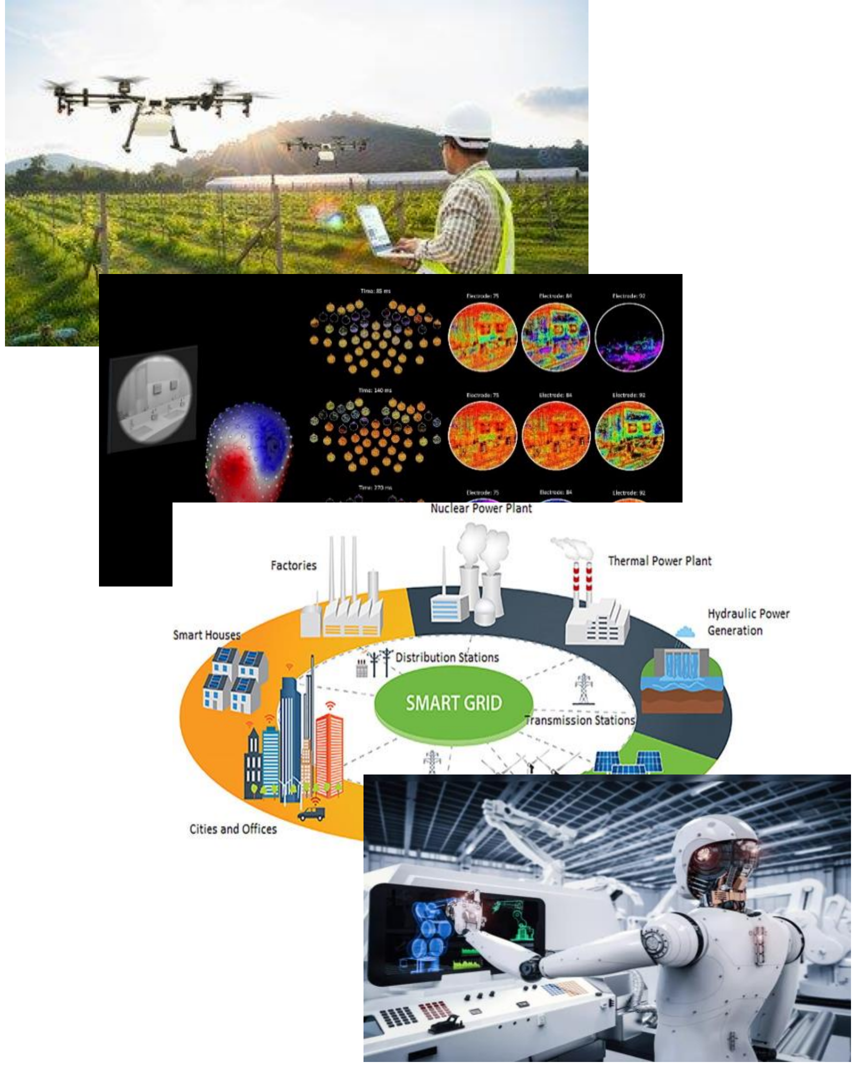
Future Development

-Fifth Generation Computer Systems (FGCS)

- Initiative by the Japanese Ministry of International Trade and Industry in 1980s.
- Computers with massive power of parallel computing/Processing.
- Supercomputers with Artificial Intelligence (AI) technology
- Develop devices that respond to natural language input and are capable of learning and self-organization
- Quantum computation, molecular and nanotechnology is another developing area

Future Development Cont..

-Trending Technologies

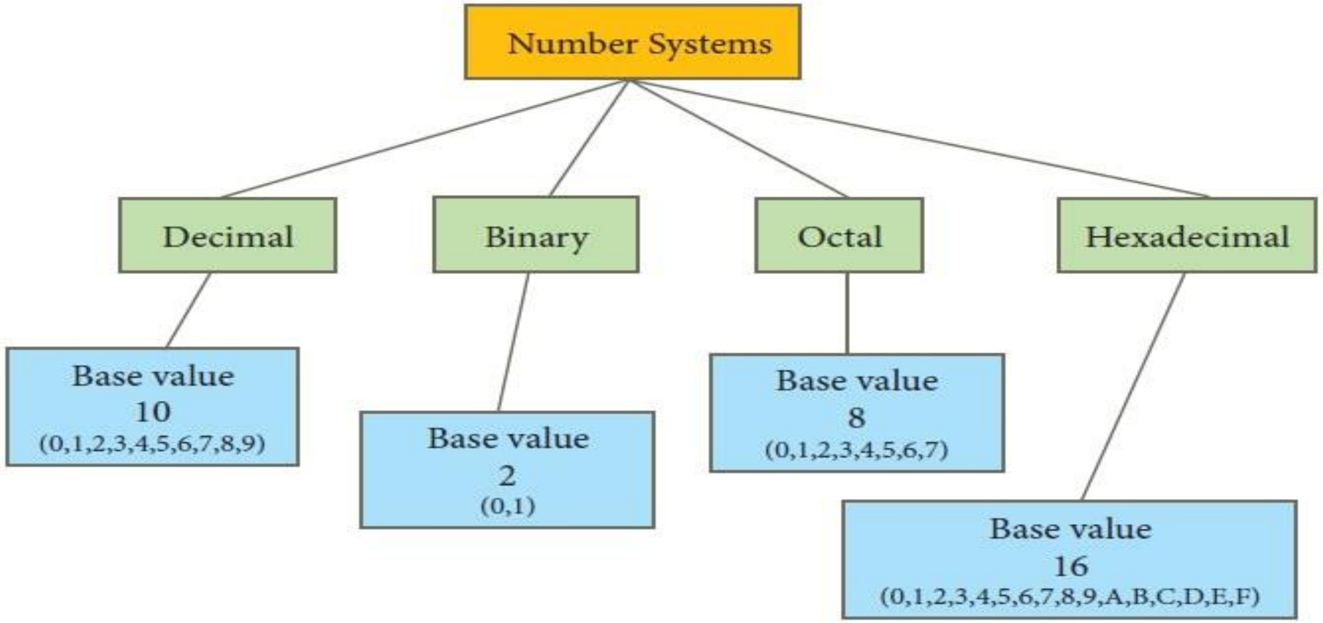


- **Agricultural Drones:** These are some cheap drones with advanced sensors and imaging. These allow farmers new ways to increase yields and reduce crop damage.
- **Brain Mapping:** Mapping structure of the brain in far greater detail than ever before, providing research avenues for neuroscientists.
- **Neuromorphic Chips:** Microprocessors configured more like brains.
- **Agile Robots:** Machines having more balance and agility to walk and run across rough terrains.
- **Smart Wind and Solar Power:** AI and Big Data are producing ultra-accurate forecasts that will make it feasible to integrate much more renewable energy into the present power grids.

Arithmetic in computers

Arithmetic in computers

- How a computer works to handle your data and produce results you need?
- Data in computer - Manipulation of numbers
- Total deal is a course of holding, processing and storing numbers.
- The design of a computer – capability of to represent numbers.



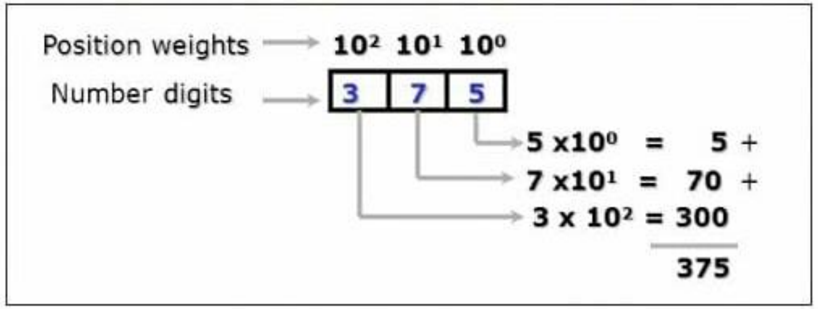
Decimal Number System

What are decimal numbers?

Decimal number system has ten-digits represented by 0,1,2,3,4,5,6,7,8,9. Any decimal number can be represented by these digits and since there are ten-digits, therefore the base or radix of this number system is 10.

- How is a **positive integer** represented in decimal?
- Let's analyze the decimal number **375**:

$$375 = (3 \times 100) + (7 \times 10) + (5 \times 1)$$
$$= (3 \times 10^2) + (7 \times 10^1) + (5 \times 10^0)$$



- The decimal numbering system is a positional number system.
- Example:

5 6 2 1

10^3 10^2 10^1 10^0

$1 \times 10^0 = 1$

$2 \times 10^1 = 20$

$6 \times 10^2 = 600$

$5 \times 10^3 = 5000$

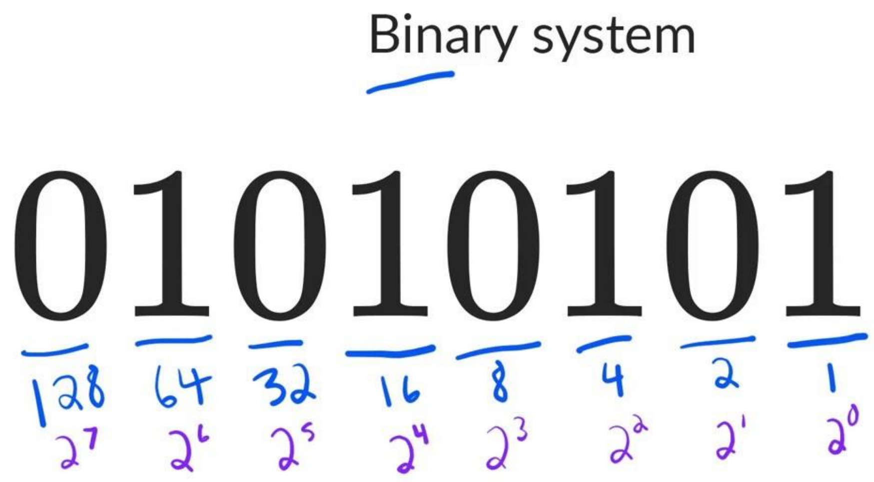
Binary Number System

-What are binary numbers?

Why Binary System?

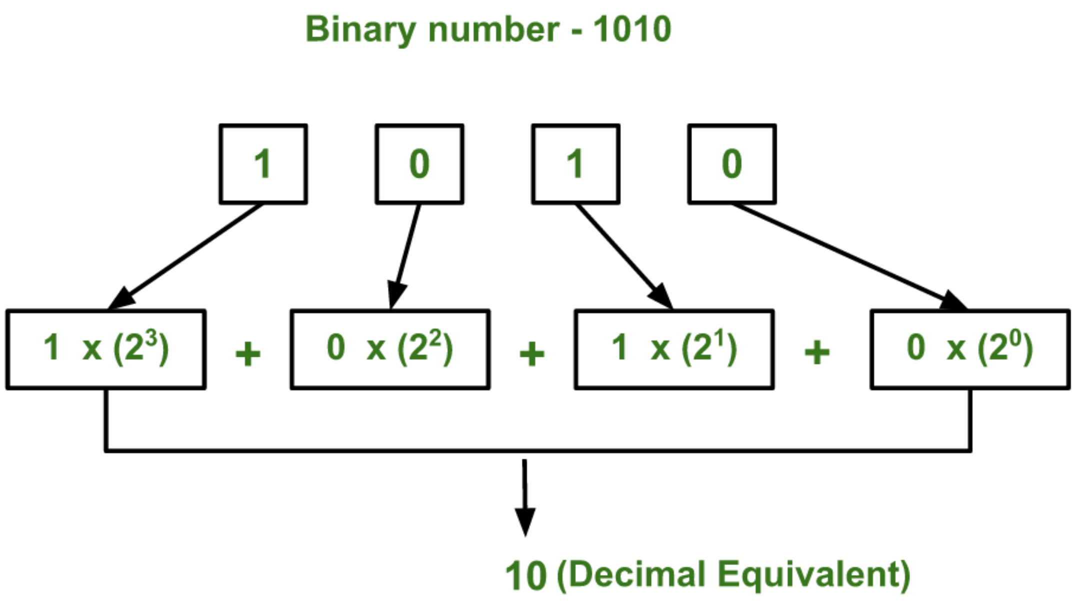
- Computers are made of a series of switches
- Each switch has two states: ON or OFF
- Each state can be represented by a number – 1 for "ON" and 0 for "OFF"
- Basically speaking, binary system simplifies information representation and information processing in electronic world.
- Binary number system is the easiest one to implement from the hardware point of view.
- The binary number system suits a computer extremely well, because it allows simple CPU and memory designs.
- So computers use binary numbers.

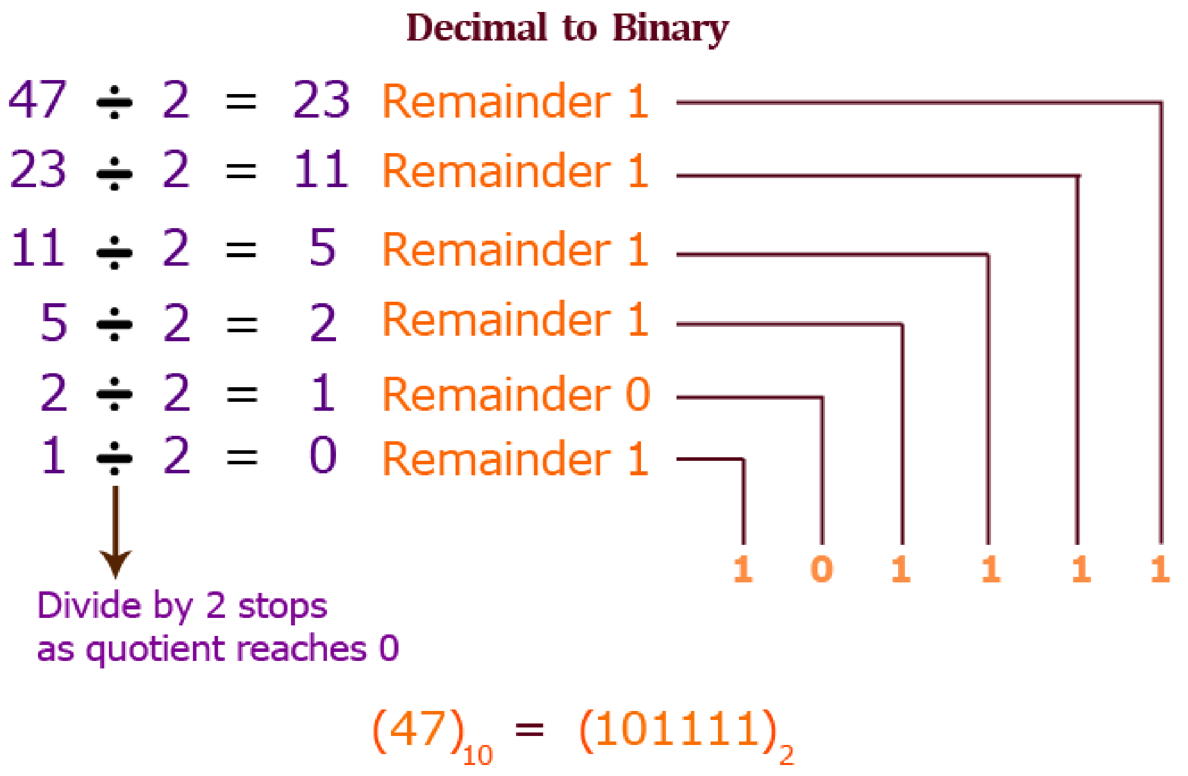
In binary number system we have two digits 0 and 1. Computer represents all kinds of data and information in binary numbers. It includes audio, graphics, video, text and numbers. The base of binary number system is 2.



Binary Number System Cont..

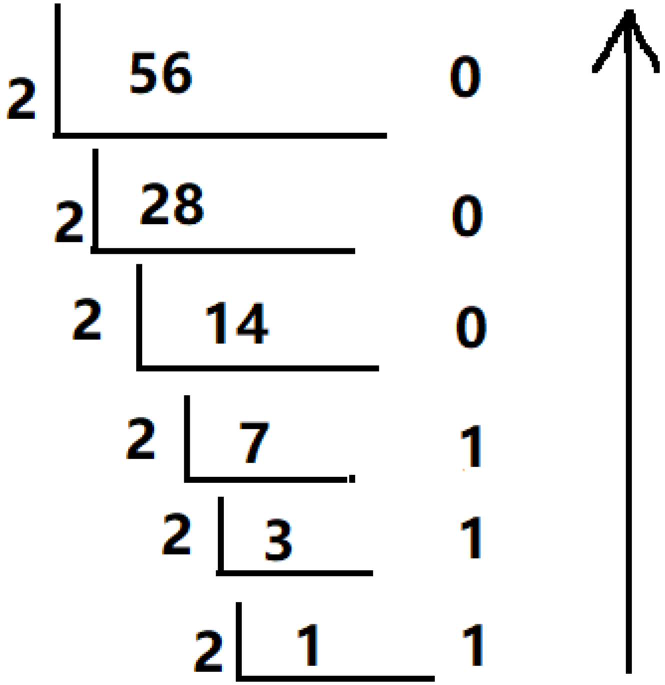
Decimal to Binary Conversion





Binary Number System Cont..

Activity 1.1



$$56_{10} = 111000_2$$

Binary Number System Cont..

Binary to Decimal Conversion

1×2^{3}	1×2^{2}	0×2^{1}	1×2^{0}		1×2^{-1}	0×2^{-2}	1×2^{-3}	1×2^{-4}
1	1	0	1	.	1	0	1	1
8	4	0	1		0.5	0	0.125	0.0625

Binary point

8 + 4 + 0 + 1 + 0.5 + 0 + 0.125 + 0.0625 = 13.6875 (Base 10)

- 1011.1_2

$$= 1\times2^3 + 0\times2^2 + 1\times2^1 + 1\times2^0 + 1\times2^{-1}$$

$$= 8 + 0 + 2 + 1 + .5$$

$$= 11.5_{10}$$

Note that positional notation can be used to convert from binary to its equivalent decimal value

Fraction Conversion Between Binary and Decimal Cont..

Decimal to binary

$(458.692)_{10} = (?)_2$

$.692 \times 2 = 1.384$	1	MSB
$.384 \times 2 = 0.768$	0	
$.768 \times 2 = 1.536$	1	
$.536 \times 2 = 1.072$	1	LSB

Binary Number System Cont..

Activity 1.2

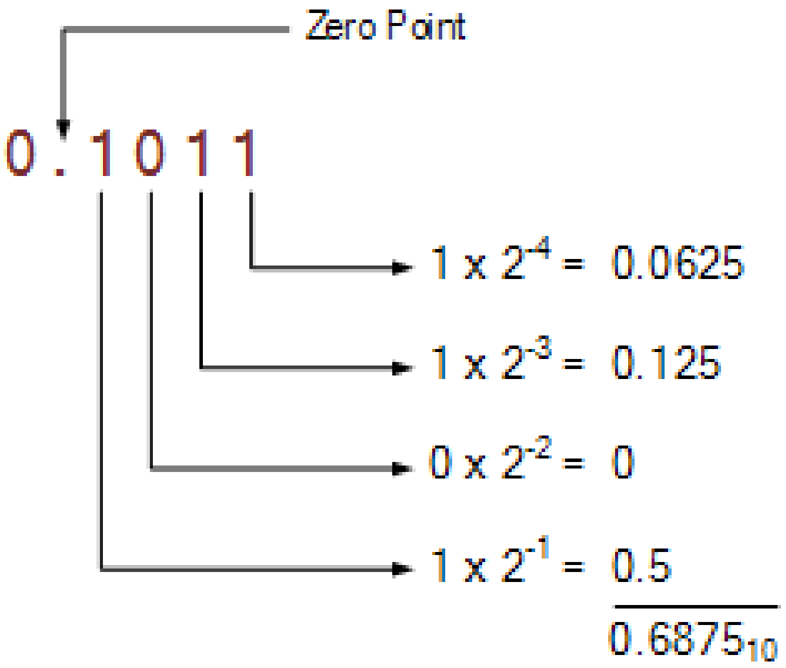
$1101\ 0110_2$ →

Place	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Value	128	64	32	16	8	4	2	1
Evaluate	1 x 128	1 x 64	0 x 32	1 x16	0 x 8	1 x 4	1 x 2	0 x 1
Sum for Base 10	128	64	0	16	0	4	2	0

$128 + 64 + 16 + 4 + 2 = 214_{10}$

Fraction Conversion Between Binary and Decimal

Binary to decimal



Octal and Hexadecimal Number System

Octal number system

An octal number system has eight-digits represented as 0,1,2,3,4,5,6,7. The base of octal number system is 8.

Decimal	Binary	Octal
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7
8	001 000	10
9	001 001	11
10	001 010	12
11	001 011	13
12	001 100	14

Octal Numbering System (base 8)

Characters = 0,1,2,3,4,5,6,7

4	3	7
64's place	8's place	1's place

= 4x64+ 3x8 + 7x1

written 437_o or 437_8

Octal and Hexadecimal Number System Cont..

Hexadecimal number system

The hexadecimal system has 16-digits, which are represented as 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. The base of hexadecimal number system is 16.

It uses base 16-system .It uses 0-9 and A,B,C,D,E,F

It is also a positional value system .

Four bits is called Nibble.

A nibble is one hexadecimal digit.

Two nibble is a byte(8 bits).

Most computer uses bytes or multiples of byte.

Hexadecimal Numbering System (base 16)

Characters = 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

E	4	C
256's place	16's place	1's place

= 14x256 + 4x16 + 12x1

written $E4C_h$ or $E4C_{16}$

Numbering System		
System	Base	Digits
Binary	2	0, 1
Octal	8	0,1,2,3,4,5,6,7
Decimal	10	0,1,2,3,4,5,6,7,8,9
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Binary Arithmetic

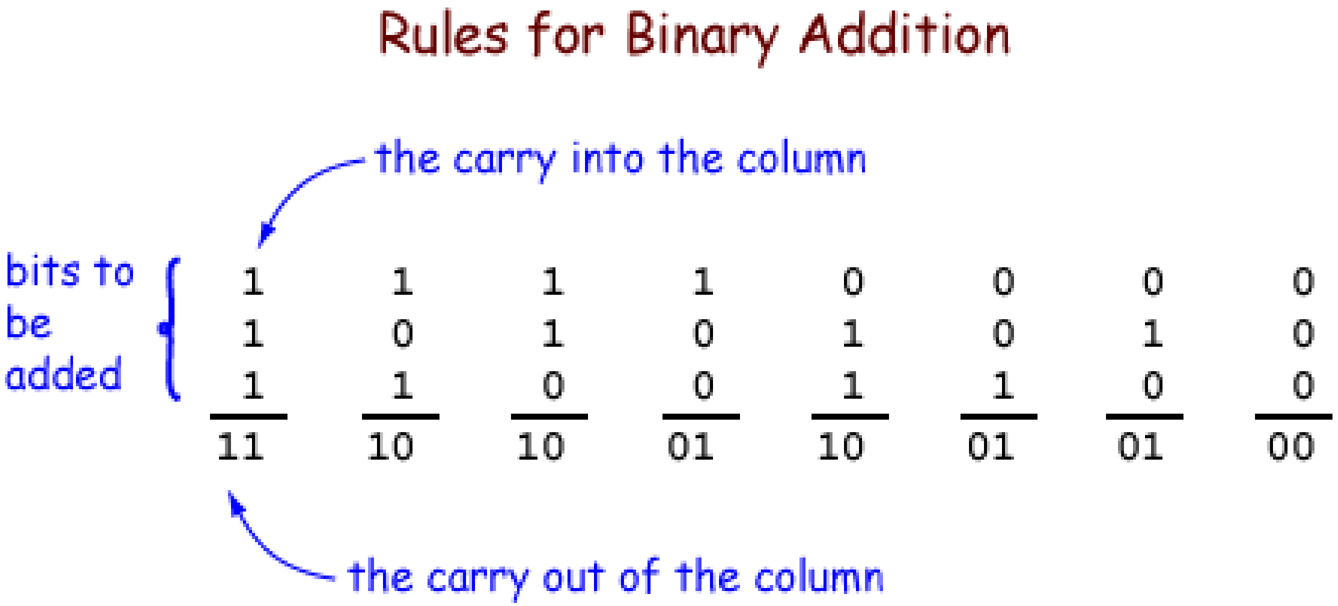
What is binary arithmetic?

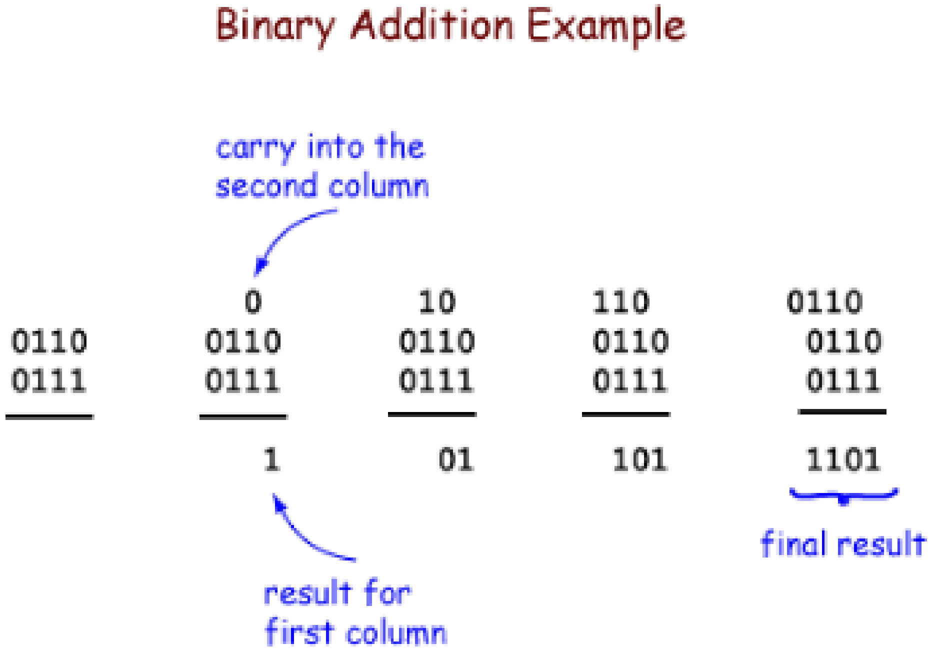
- Binary arithmetic is essential part of all the digital computers and many other digital system.
- **Binary arithmetic** is used in digital systems mainly because the numbers (decimal and floating-point numbers) are stored in **binary** format in most computer systems.
- All **arithmetic** operations such as addition, subtraction, multiplication, and division are done in **binary** representation of numbers.

	Addition	Subtraction	Multiplication	Division
i)	0 + 0 = 0	0 – 0 = 0	0 x 0 = 0	0 / 1 = 0
ii)	0 + 1 = 1	1 – 0 = 1	0 x 1 = 0	1 / 1 = 1
iii)	1 + 0 = 1	1 – 1 = 0	1 x 0 = 0	0 / 0 = not allowed (not valid)
iv)	1 + 1 =10	1-0 = 10 – 1 (with borrow 1) = 1	1 x 1 = 1	1 / 0 = not allowed (not valid)

Binary Arithmetic Cont..

Binary addition





		1	1	1	
		1	1	0	1
+		1	1	1	1
	1	1	1	0	0

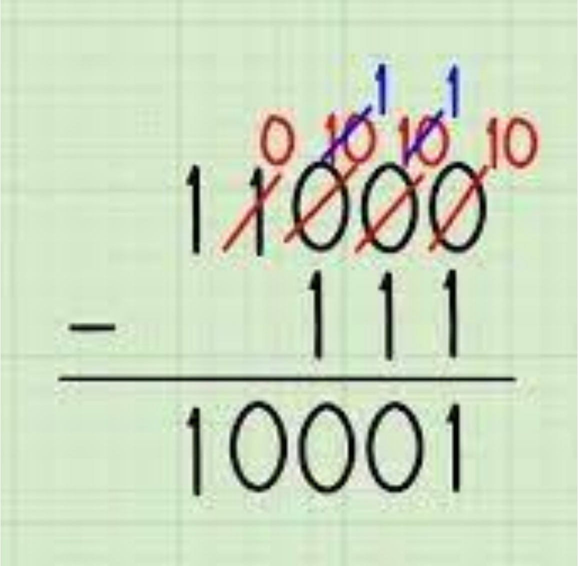
A + B	SUM	CARRY
0 + 0	0	0
0 + 1	1	0
1 + 0	1	0
1 + 1	0	1

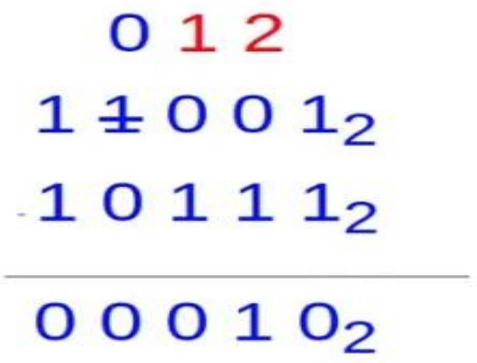
Binary Arithmetic Cont..

Binary subtraction

- Two binary numbers are subtracted by subtracting each pair of bits together with borrowing, where needed.
- Subtraction Example:

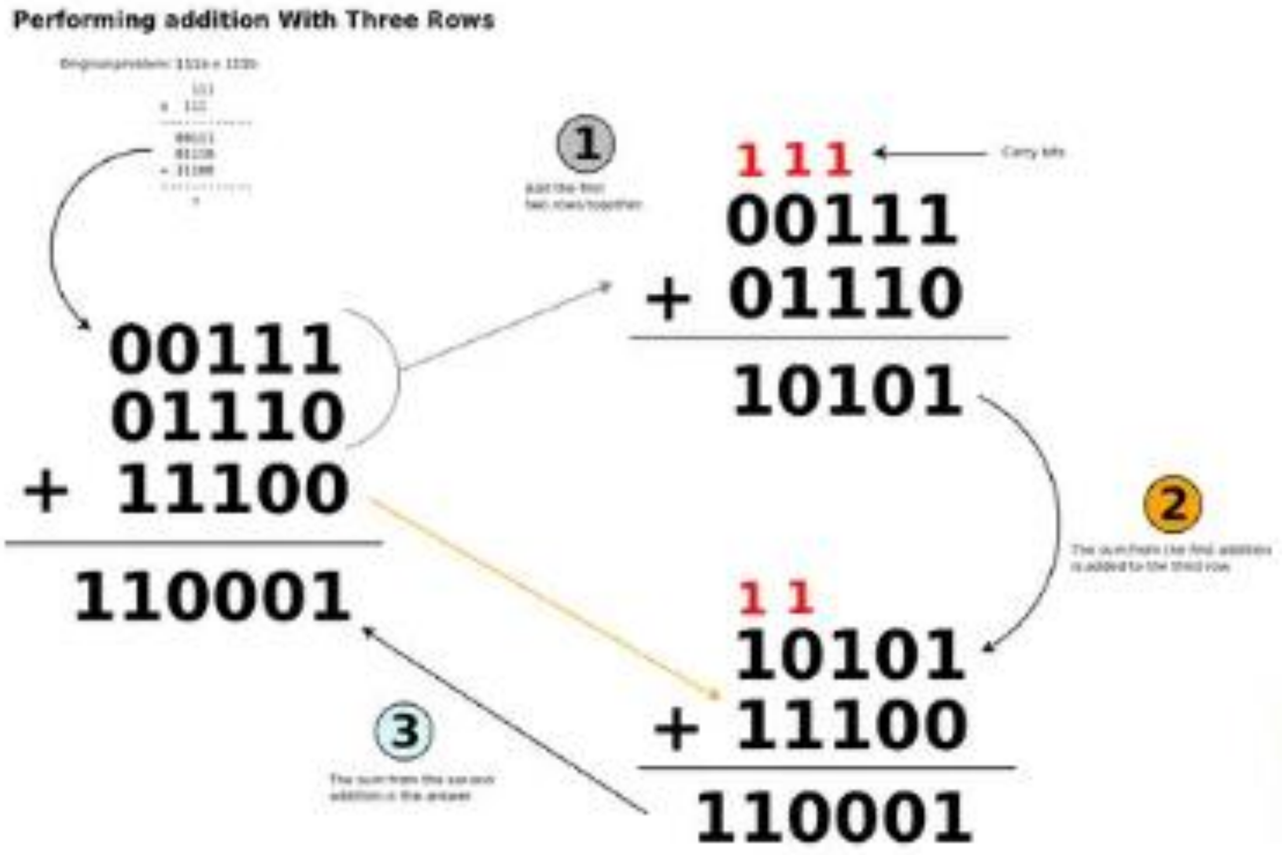
			0	0	1	1	1	1	1	0	0	Borrow
X		229		1	1	1	0	0	1	0	1	
Y	-	46	-	0	0	1	0	1	1	1	0	
		183		1	0	1	1	0	1	1	1	

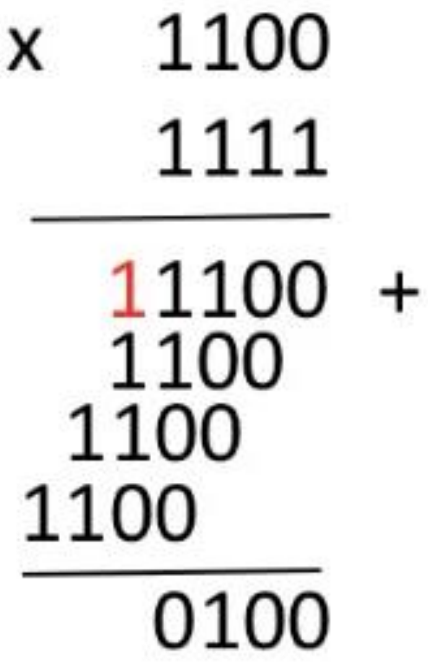




Binary Arithmetic Cont..

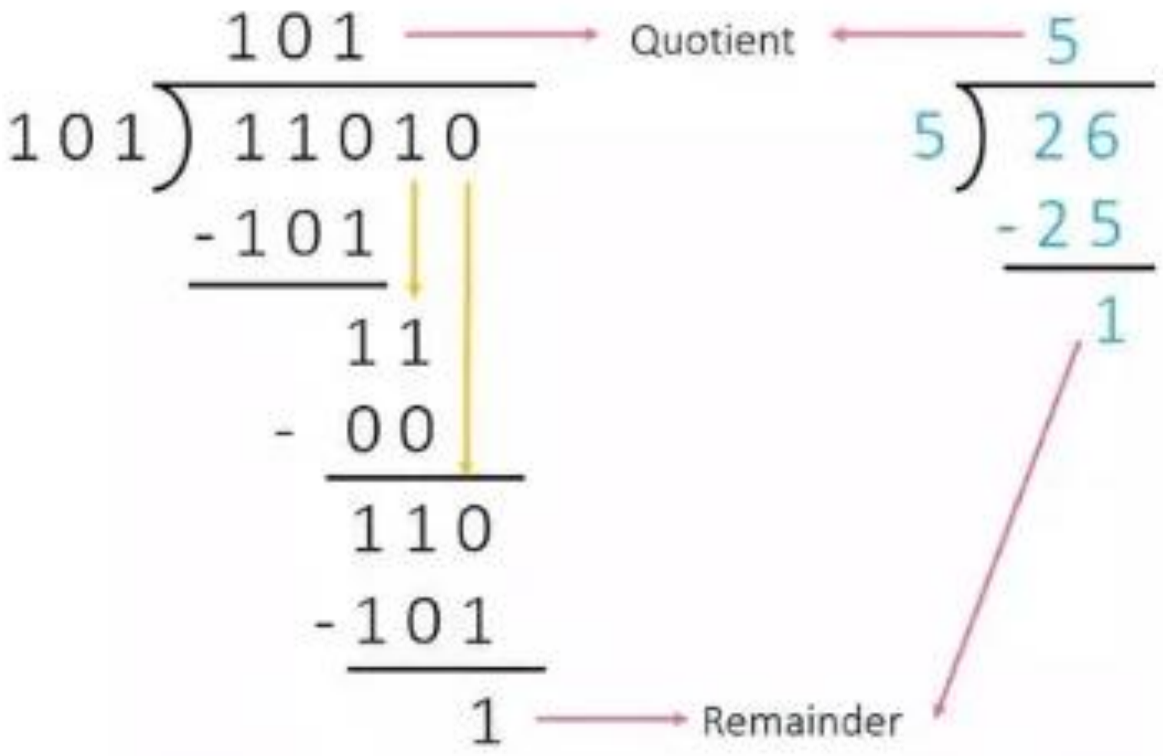
Binary multiplication

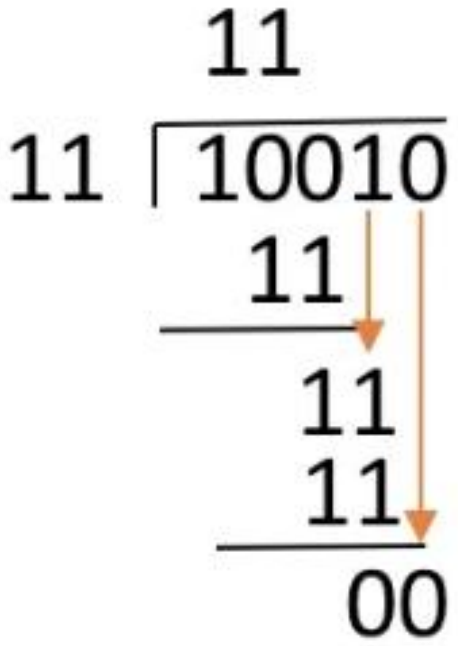




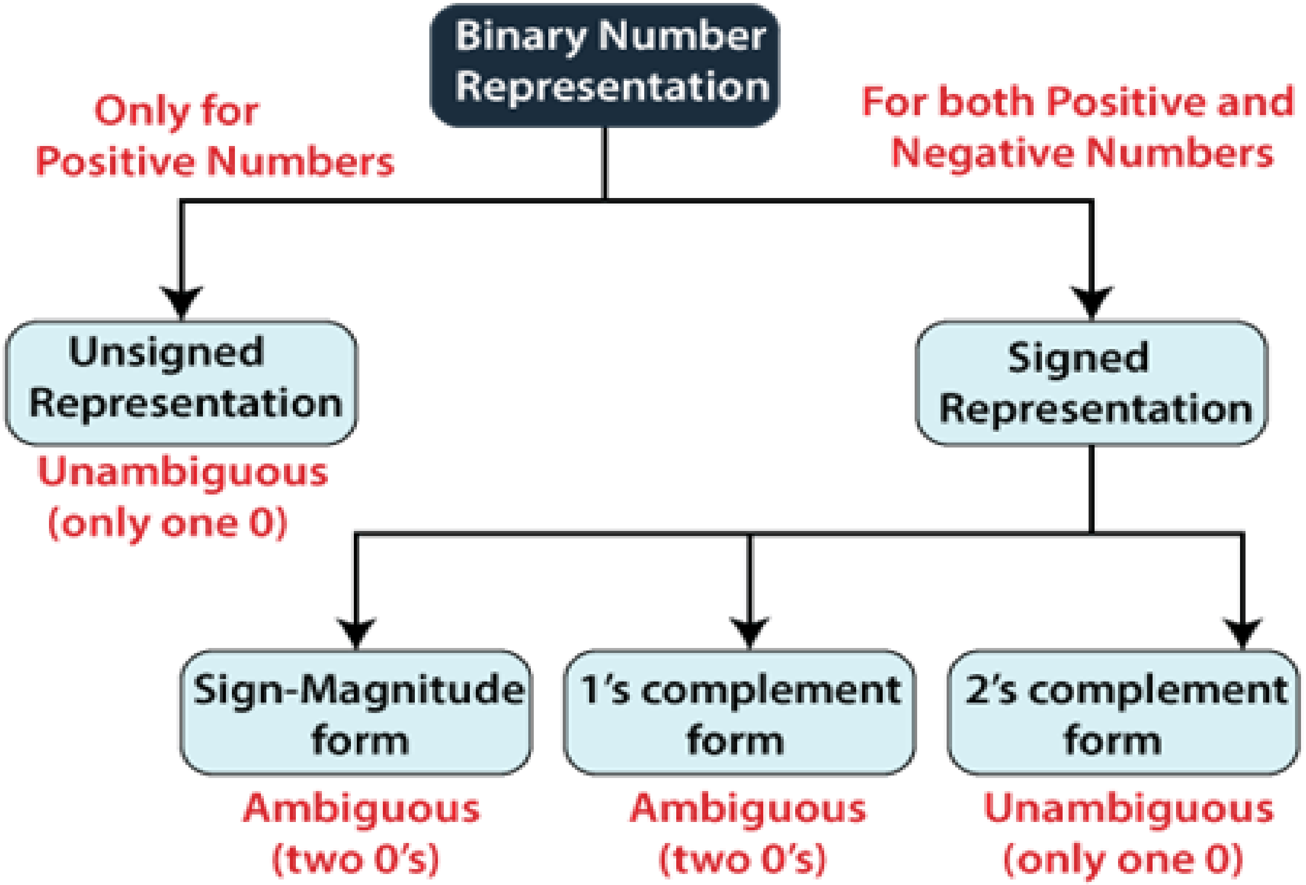
Binary Arithmetic Cont..

Binary division



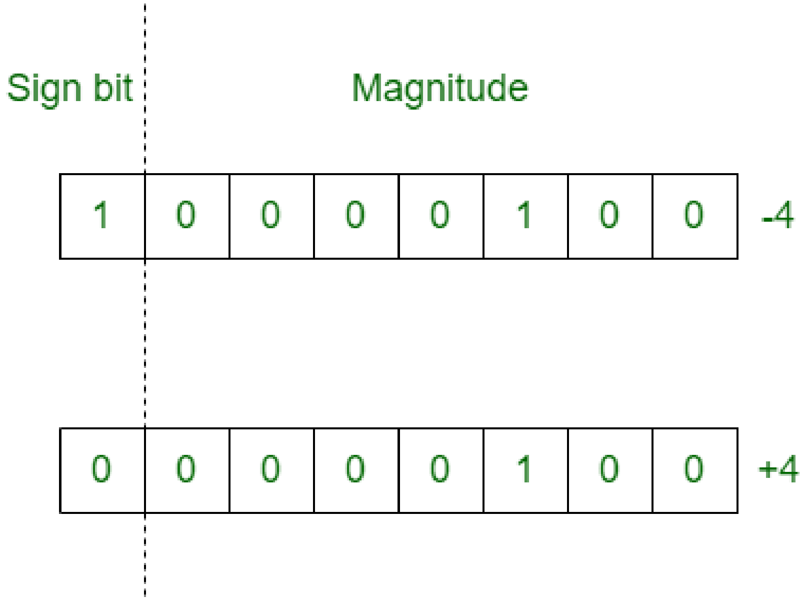


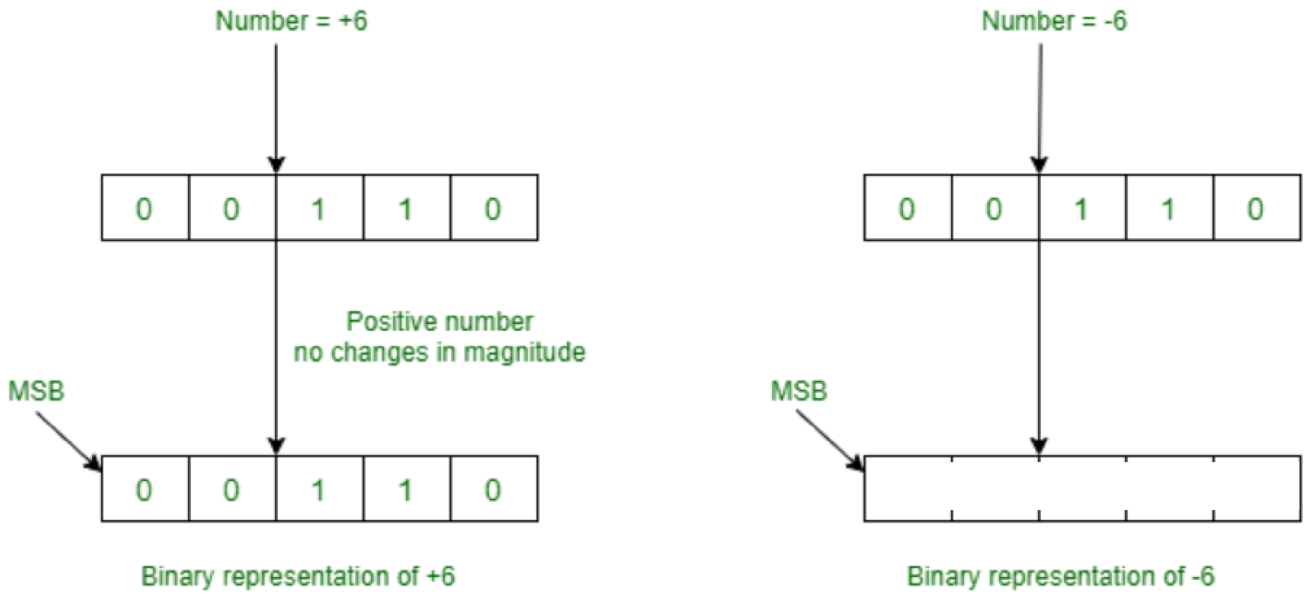
Negative Numbers

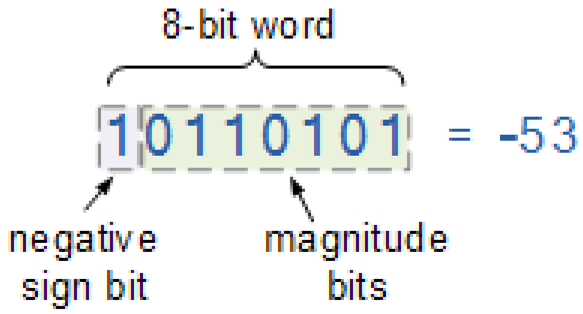


Negative Numbers Cont..

Sign and magnitude







Negative Numbers Cont..

One's complement

Invert all bits. Each 1 becomes a 0, and each 0 becomes a 1.

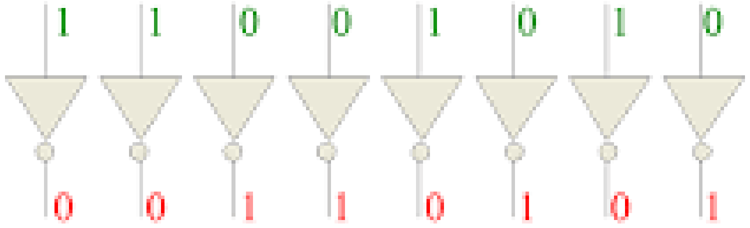
Original Value	One's Complement
0 1	1 0
1010 1111	0101 0000
11110000 10100011	00001111 01011100
11110000 10100101	00001111 01011010

-7 =	1	0	0	0
-6 =	1	0	0	1
-5 =	1	0	1	0
-4 =	1	0	1	1
-3 =	1	1	0	0
-2 =	1	1	0	1
-1 =	1	1	1	0
-0 =	1	1	1	1

The 1's complement of a binary number is just the inverse of the digits. To form the 1's complement, change all 0's to 1's and all 1's to 0's.

For example, the 1's complement of 11001010 is 00110101

In digital circuits, the 1's complement is formed by using inverters:



Negative Numbers Cont..

Two's complement

- 2's Complement
- (2-1)'s Complement / 1's Complement

- 1's Complement – It is obtained by subtracting each digit of the number from 1.

  Ex:-

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

11010

00101

- 2's Complement – It is obtained by adding 1 in the 1's complement.

  Ex: -

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

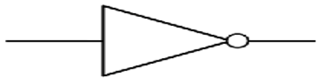
Introduction to Boolean Algebra

Basic operations in Boolean algebra

YES gate

INPUT	OUTPUT
0	0
1	1

NOT gate



INPUT	OUTPUT
0	1
1	0

AND gate

INPUT		OUTPUT
A	B	
0	0	0
0	1	0
1	0	0
1	1	1

NAND gate

INPUT		OUTPUT
A	B	
0	0	1
0	1	1
1	0	1
1	1	0

OR gate

INPUT		OUTPUT
A	B	
0	0	0
0	1	1
1	0	1
1	1	1

NOR gate

INPUT		OUTPUT
A	B	
0	0	1
0	1	0
1	0	0
1	1	0

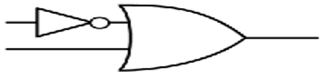
XOR gate

INPUT		OUTPUT
A	B	
0	0	0
0	1	1
1	0	1
1	1	0

XNOR gate

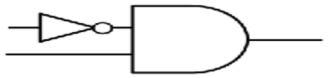
INPUT		OUTPUT
A	B	
0	0	1
0	1	0
1	0	0
1	1	1

IMPLY
(A IMPLY B)



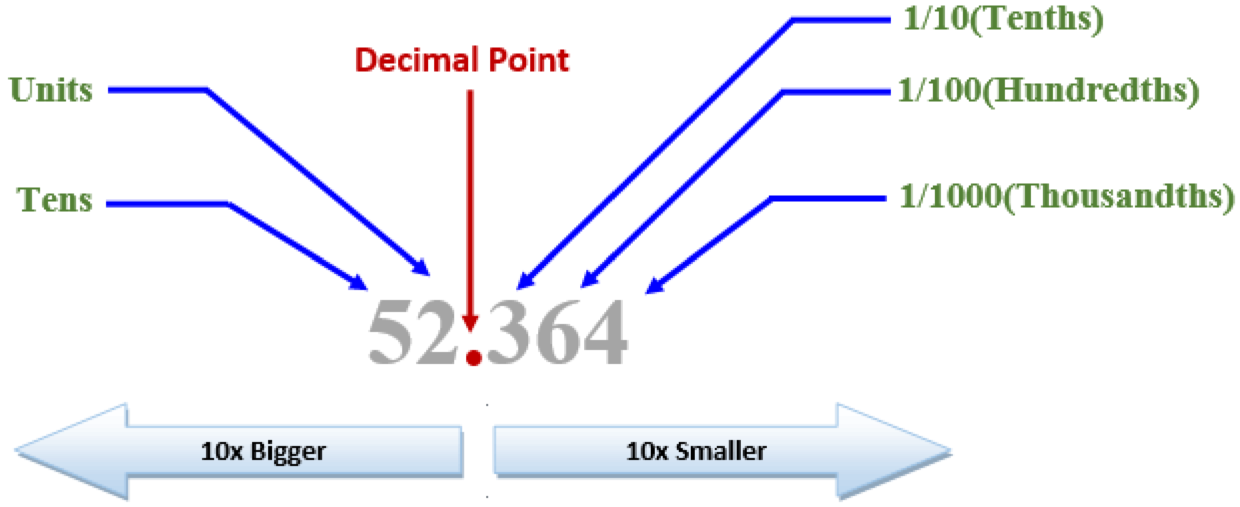
INPUT		OUTPUT
A	B	
0	0	1
0	1	1
1	0	0
1	1	1

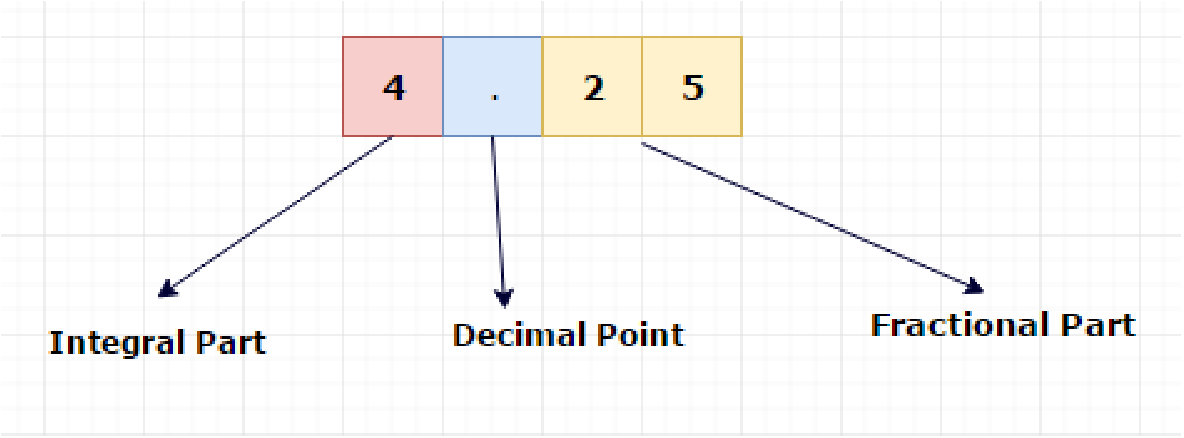
N-IMPLY
(B N-IMPLY A)

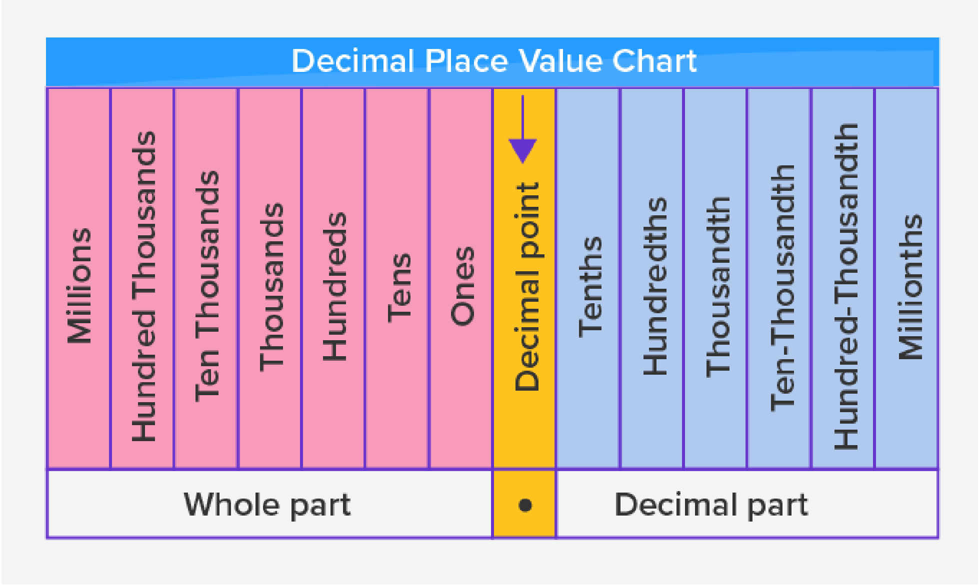


INPUT		OUTPUT
A	B	
0	0	0
0	1	1
1	0	0
1	1	0

Fractional Numbers







Octal and Hexadecimal Fractions

Example: Convert 71.5_8 to Hexadecimal:

Part 1: Convert the octal number into binary. Each octal symbol is represented with 3 bits.

$$7\ 1\ .\ 5_8$$

$$(1\ 1\ 1)\ (0\ 0\ 1)\ .\ (1\ 0\ 1)_2$$

$$111001\ .\ 101_2$$

Part 2: Convert the binary number into hexadecimal. Form groups of 4 bits representing hex symbols.

Step 1: $(0\ 0\ 1\ 1)\ (1\ 0\ 0\ 1)\ .\ (1\ 0\ 1\ 0)_2$

Whole number groupings start at the radix point and work left. Leading 0's are added as necessary.

Fractional number groupings start at the radix point and work right. Trailing 0's are added as necessary.

Step 2: $(0\ 0\ 1\ 1)\ (1\ 0\ 0\ 1)\ .\ (1\ 0\ 1\ 0)_2$

$$3\ 9\ .\ A_{16}$$

Binary Codes Cont..

Special binary codes

- Computers and other digital systems "work" with binary numbers.
- I/P & O/P is usually done using decimal numbers, alphabetics, special symbols.
- Some way of representing alphanumerics with binary numbers is required.
- These representations are called *codes*.
- Many codes are possible, and a few *standard codes* are used, such as:

ASCII - American Standard Code for Information Interchange — 7-bit System

EBCDIC - Extended Binary Coded Decimal Interchange Code — Used in Big Mainframe Systems

BCD - Binary Coded Decimal. For numbers only.

Hardware and/or software is required to convert coded numbers into binary numbers before any arithmetic operations can take place.

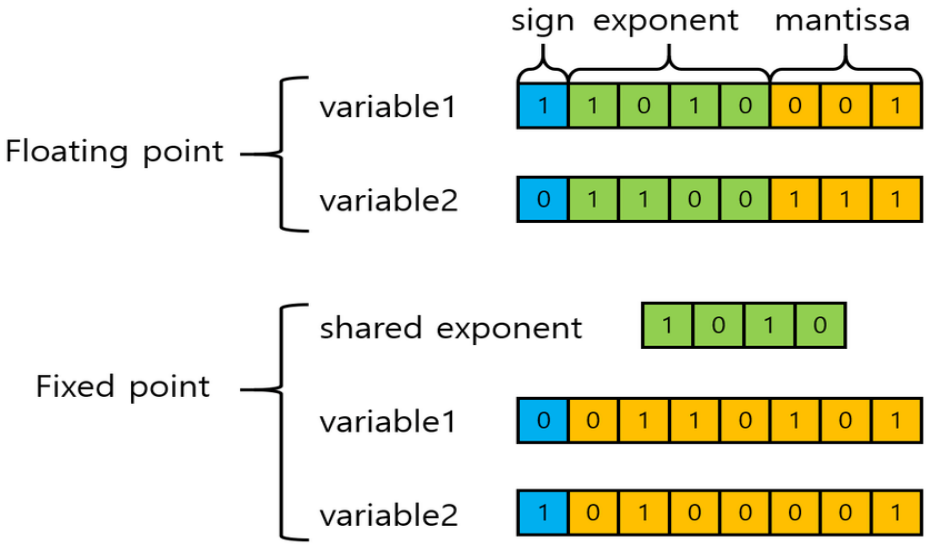
DACIMAL	8421 BCD	GRAY	BINARY
0	0000	0000	0000
1	0001	0001	0001
2	0010	0011	0010
3	0011	0010	0011
4	0100	0110	0100
5	0101	0111	0101
6	0110	0101	0110
7	0111	0100	0111
8	1000	1100	1000
9	1001	1101	1001
10	0001 0000	1111	1010
11	0001 0001	1110	1011
12	0001 0010	1010	1100
13	0001 0011	1011	1101
14	0001 0100	1001	1110
15	0001 0101	1000	1111

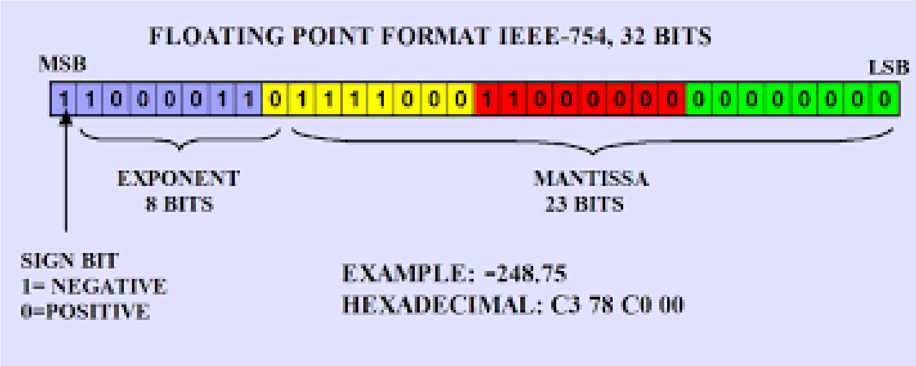
Fixed Point and Floating Point Numbers

Floating point numbers

Floating vs. fixed point

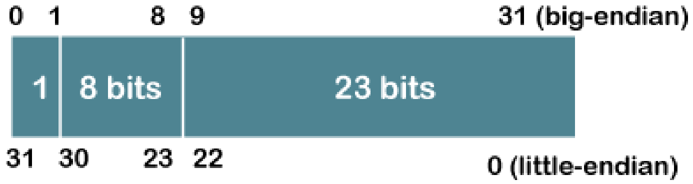
- *Floating point* refers to a binary decimal representation where there is not a fixed number of digits before and after the decimal place (contrast "fixed point").
- The decimal is said to be allowed to "float"
- Virtually all general purpose modern processors are based on ieee-754 floating point standard
- Allows representation of much greater range of numbers at cost of greater overhead



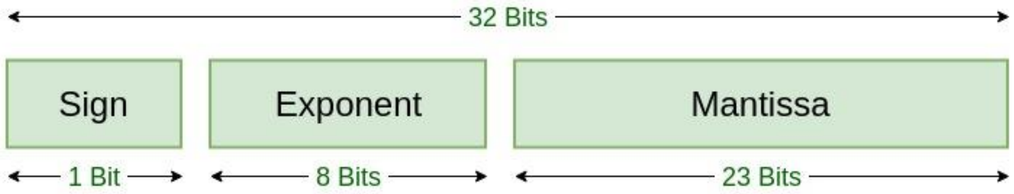


Fixed Point and Floating Point Numbers Cont..

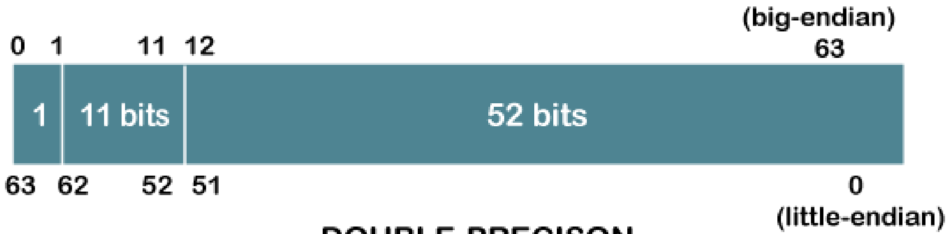
Floating point representation



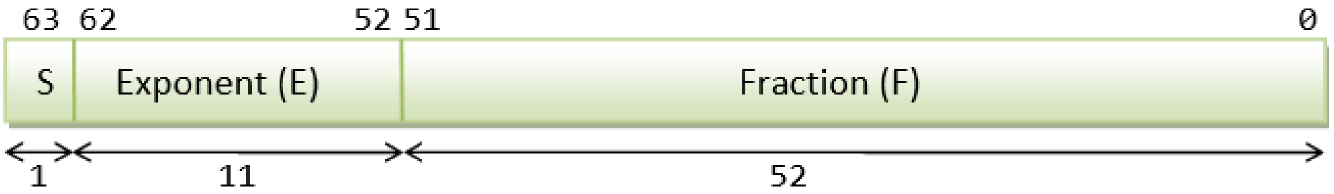
SINGLE-PRECISION



Single Precision
IEEE 754 Floating-Point Standard



DOUBLE-PRECISON

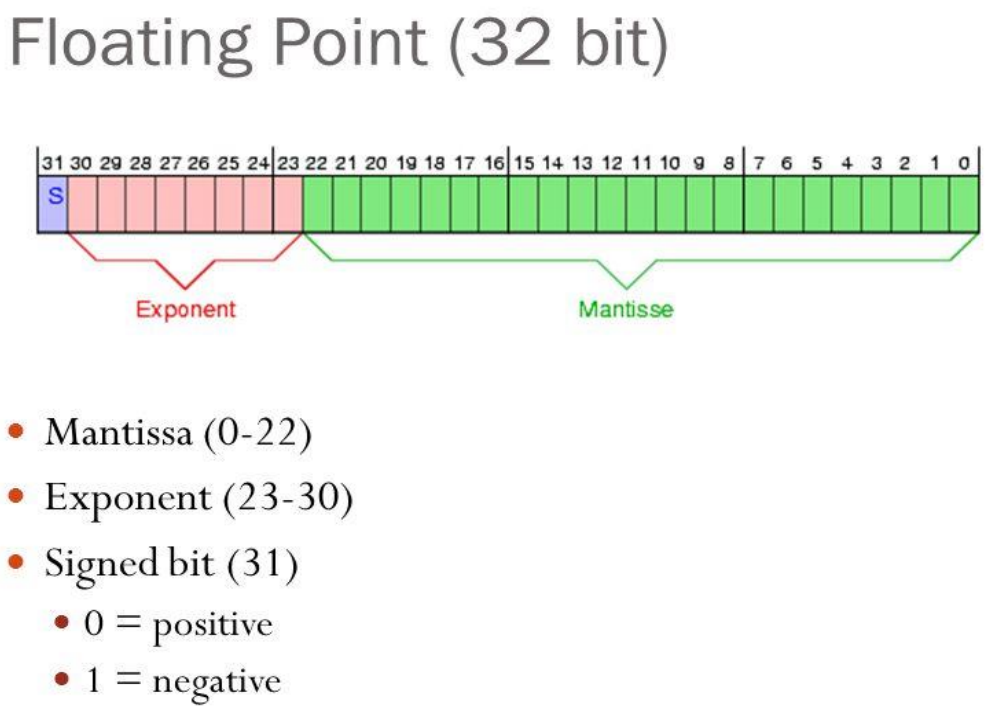


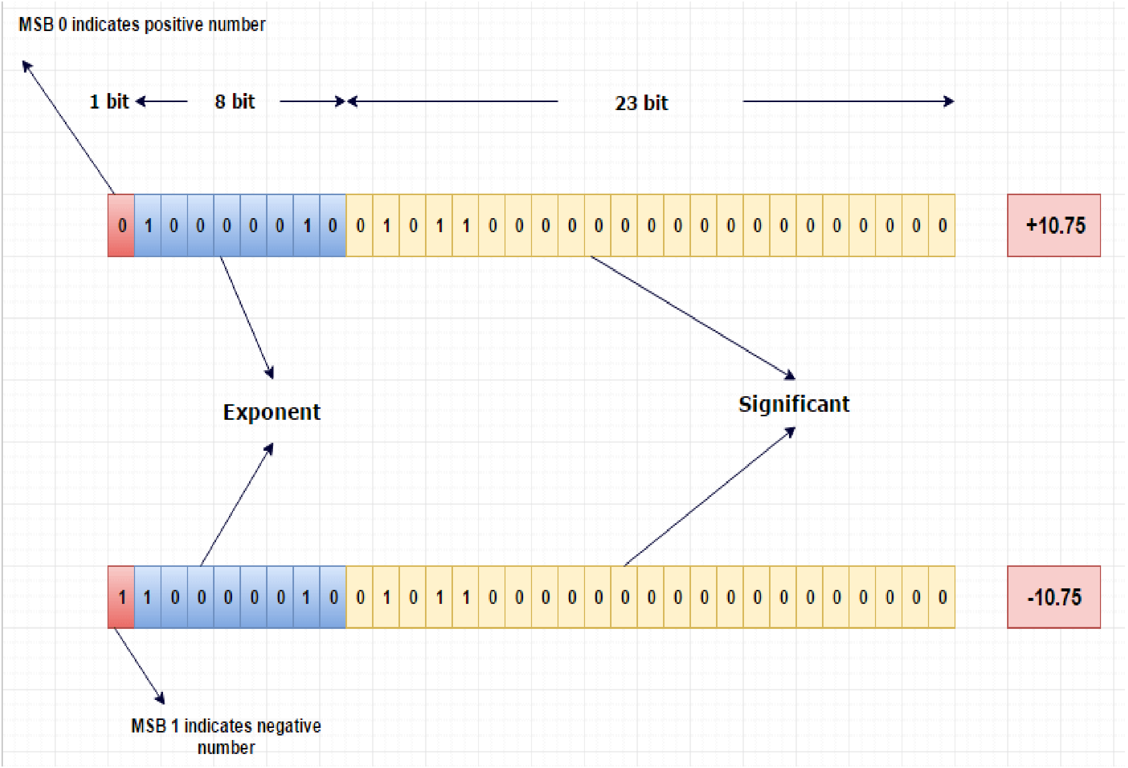
64-bit Double-Precision Floating-point Number

Fixed Point and Floating Point Numbers Cont..

Sign bit

This concept is very simple. **0 denotes a positive number, and 1 denotes a negative number.**

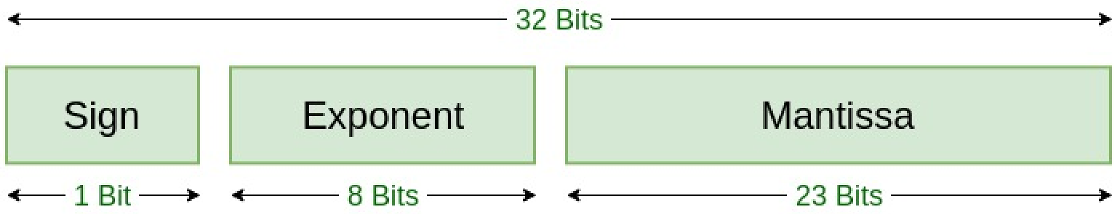




Fixed Point and Floating Point Numbers Cont..

Exponent

- The exponent is an integer that determines the scale of the number.
- Indicates how many positions the decimal point (or binary point) should be moved to the left or right within the mantissa/ significant.
    - A positive exponent means the decimal point is moved to the right, resulting in a larger magnitude.
    - A negative exponent means the decimal point is moved to the left, leading to a smaller magnitude.
- In single precision exponent is 8 bits, and the bias is 127
- In double precision, the exponent field is 11 bits, and has a bias of 1023



Fixed Point and Floating Point Numbers Cont..

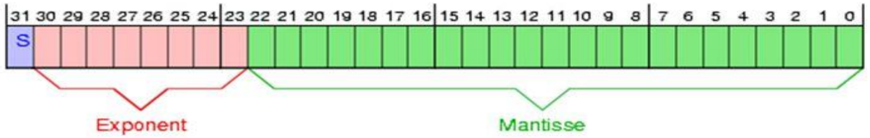
Mantissa

- Mantissa is the biggest part in the format and is **representing the precision bits**.

- This is the fractional part of the number. It contains the significant digits of the number, determining its precision.

- Mantissa is having **two parts**;
  - implicit leading bit (left of the radix point)
  - fraction bits (to the right of the radix point).

Floating Point (32 bit)



- Mantissa (0-22)
- Exponent (23-30)
- Signed bit (31)
  - 0 = positive
  - 1 = negative

Table 3.2 Typical convention of storing floating-point numbers

	Denormalized	Normalized	Approximate Decimal
Single Precision	$\pm 2^{-149}$ to $(1-2^{-23})\times 2^{-126}$	$\pm 2^{-126}$ to $(2-2^{-23})\times 2^{127}$	$\pm \approx 10^{-44.85}$ to $\approx 10^{38.53}$
Double Precision	$\pm 2^{-1074}$ to $(1-2^{-52})\times 2^{-1022}$	$\pm 2-10^{22}$ to $(2-2^{-52})\times 2^{1023}$	$\pm \approx 10^{-323.3}$ to $\approx 10^{308.3}$

Floating Point Operations

Floating point addition

Decimal addition

Step 1: Shift the decimal point of the smaller number to the left until the exponents are equal. Thus, the first number becomes $.0225\times10^2$.

Step 2: Add the numbers with decimal points aligned.

$$\begin{array}{rl} & 0.0225 \times 10^2 \\ + & 1.340625 \times 10^2 \\ \hline & 1.363125 \times 10^2 \end{array}$$

Step 3: Normalize the result.

(Once the decimal points are aligned, the addition can be performed by ignoring the decimal point and using integer addition).

Binary addition

Step 1: Shift the binary point of the smaller number to the left until the exponents are equal. Thus, the first number becomes 0.000001001×2^7

Step 2: Add the mantissas with binary points aligned.

	S	E	M
	0	1000 0110	(0) 000 0010 0100 0000 0000 0000
+	0	1000 0110	(1) 000 0110 0001 0000 0000 0000

	0	1000 0110	(1) 000 1000 0101 0000 0000 0000

Step 3: Normalize the result.

(The result is already in normalized form. If the sum overflows the position of hidden bit, the mantissa must be shifted one bit to the right and the exponent is incremented. Since the mantissa is always less than 2, the hidden bits can sum to no more than 3 (11).

Floating Point Operations Cont..

Floating point multiplication

When compared to floating point addition, floating point multiplication is much simpler. If we consider the multiplication of the same two numbers in above example.

2.25×10^0 is $10.01 = 1.001 \times 2^1$ in binary
1.340625×10^2 is $10000110.0001 = 1.00001100001 \times 2^7$ in binary

	S	E	M
	0	1000 0000	(1) 001 0000 0000 0000 0000 0000
+	0	1000 0110	(1) 000 0110 0001 0000 0000 0000

Step 1: Find the sign bit by XORing the sign bits two numbers.
i.e. Sign bit → 0 XOR 0 → 0

Step 2: Multiply the mantissa values including the "hidden one". The Resultant product of the 24 bits mantissas (M1 and M2) is 48bits (2 bits are to the left of binary point)

```
                         100001100001000000000000
                         100100000000000000000000
-------------------------------------------------
    100001100001000000000000000000000000000000000
  100001100001000000000000
-------------------------------------------------
 01.0010110110100100000000000000000000000000000000
```

Step 3: If 48th bit of resultant (mantissa) is 1, then
- normalise it by left shifting binary point by 1 place
- add 1 to exponent

If not it is already normalised and no need adjust exponent

Step 4: Truncate the result to 24 bits
1.00101101101001000000000 (normalised binary)

Step 5: Find the exponent of the result.
E1 + E2 - bias + (normalised exponent from step 2, if any)
= 10000000 + 10000110 – bias + 0
= 128+134-127
= 135

Then the answer is

S	E	M
0	1000 0111	(1) 001 0110 1101 0010 0000 0000

If we convert this to decimal, we get 301.640625, the correct answer.

THANK YOU

Any Questions?

ශ්‍රී ලංකා විවෘත විශ්වවිද්‍යාලය
இலங்கை திறந்த பல்கலைக்கழகம்
The Open University of Sri Lanka

PO Box 21, Nawala, Nugegoda, Sri Lanka
Phone: +94 11 288 1000
https://ou.ac.lk