UES103

Programming for Problem Solving

Saif Nalband, PhD

1A27-31/1A42-46 cslab768

Topics covered

Introduction to Computer Fundamentals- Computer Memory Hierarchy, Types of Software Binary number system, Algorithm, Flowchart, Formulate simple algorithms for logical and arithmetic problems.

About Myself

- 90’s Kid From Mumbai
- Completed my PhD BITS-Pilani, Goa Campus (YUP, I studied in GOA!)
- Taught at IIIT Pune
- Went for Post-Doctoral Researcher at Cork, Ireland for a year.
- Pandemic→DYPIU→IIIT Pune→TIET
- Social Acitve on Linkdin, twitter
- Hobbies ; Football, MUFC, binge watch some series, courses

I can be found on (Contact)

Preference on EMAILS:

1. Email: **saif.nalband@thapar.edu**
2. **Mob: 9322317570**
3. **Avoid WhatsApp** unless its emergency!!!
4. Chamber No: 710, CSED Building /
5. **Venue: Activity Space -1**

OutCome of Course

1. Programming paradigm → First Step
2. Initial Steps for Software development.

How to pass this course

1. Will keep you update on this. Regarding the weightage of marking scheme
2. Attendance is MUST!!!
3. DO NOT MUG THE PROGRAMS.
4. PRACTICE CODING .. It’s FUN.

Weightage Scheme : Theory : 100 marks

Modules	Weightage
MST	40*
EST	40*
Lab Sessional(x2)	(10+10) 20

Lecture (1A42 -- 46)

- Tuesday :-- 9:40 AM – 10:30 AM → LP103
- Wednesday :-- 12:10 PM – 1:00 PM → LP102
- Thursday :-- 10:30 AM – 11:20 AM → LP104

- Office Hours :
  - Monday – 3:30 PM → 5:10 PM
  - Tuesday – Tuesday 2:40 PM → 5:10 PM

Lecture (1A27-- 31)

- Monday :-- 2:40 PM – 3:30 PM → LP102
- Wednesday :-- 4:20 PM – 5:10 PM → LP102
- Friday :-- 8:00 AM – 8:50 AM → LP102

- Office Hours :
  - Monday – 3:30 PM → 5:10 PM
  - Tuesday – 2:40 PM → 5:10 PM

Lecture

- Keeping the class interesting
- Humor breaks
  - Actually helps with attention span!
  - Not surprisingly, most of it will be computer humor!
- Lecture etiquette
  - Please don’t use electronic devices during lectures
  - Plus no earbuds

Books

Text Books

1. C Programming Language, Brian W. Kernighan Dennis M. Ritchie, 2nd ed, 2012.
2. **Programming in ANSI C, Balagurusamy G., 8th ed., 2019**

Reference Books

1. **Let Us C, Kanetkar Y., 16th ed., 2017**
2. Programming with C, Byron S Gottfried, McGraw Hill Education, Forth edition, 2018

Let Cook some codes

Programming ? why

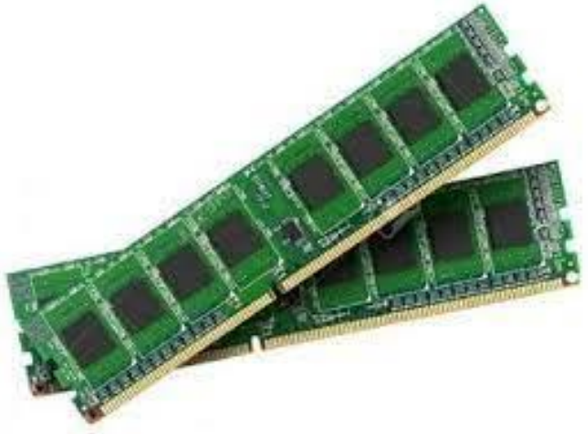
- Why Do we Write codes ?

Computer Memory

Memory is the electronic holding place for the instructions and data a computer needs to reach quickly.

Two types of memory - primary and secondary

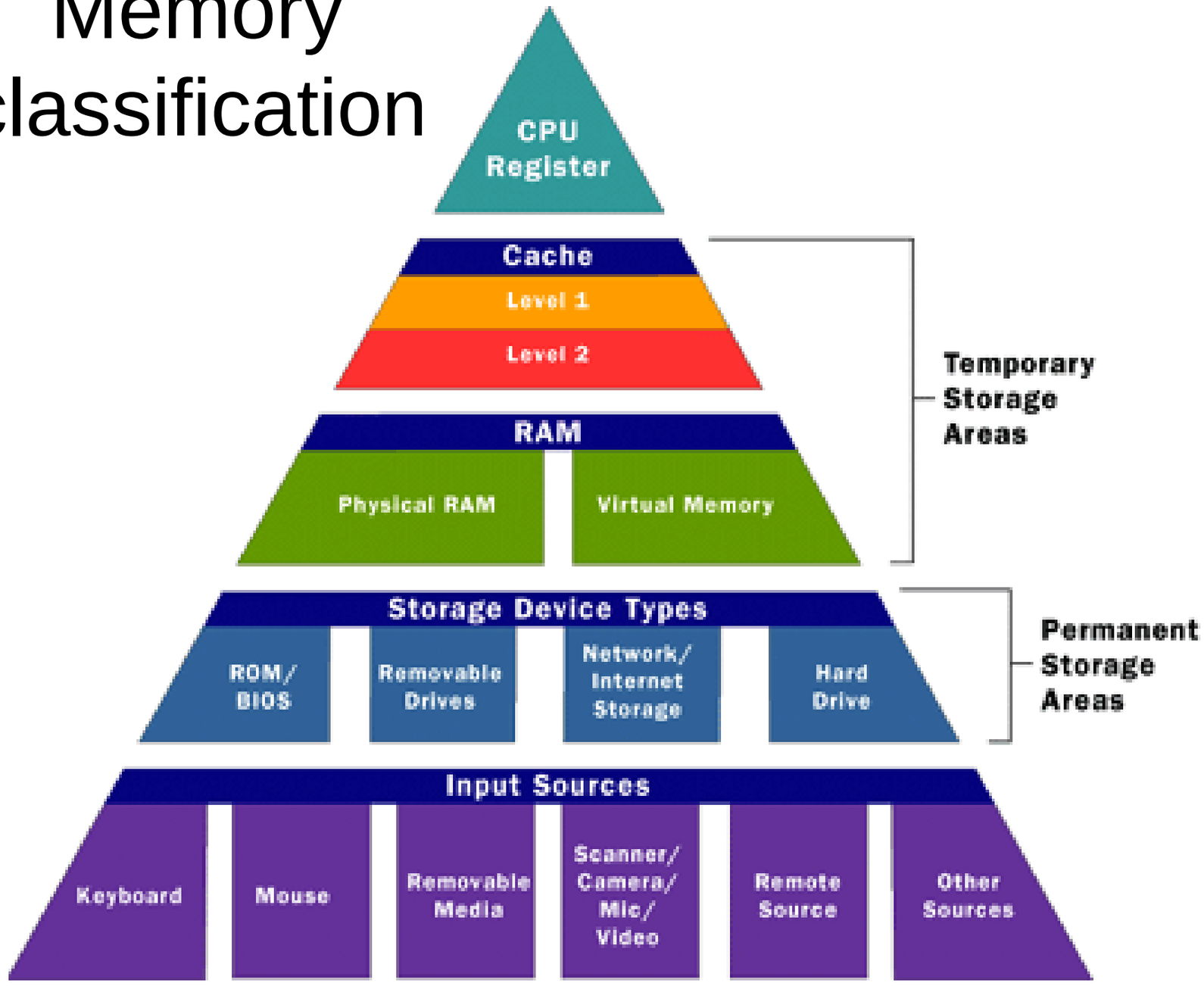
Need for memory hierarchy – To minimize the memory access time



Memory hierarchy

Register – Memory that is built directly into the CPU
Cache – Chip based memory closest to CPU
RAM – Random access memory (main memory for computer calculations or temporary storage
ROM – Read only memory; permanent memory; stores booting instructions for smooth computer operations
Removable memory - pen drive or external hard drive)

Memory classification



The C Programming Language

Who? Dennis Ritchie
When? ~1972
Where? Bell Labs
Why? Develop the Unix OS

C: Details

C

hello.c:

```
#include <stdio.h>

int main(void)
{   printf("hello, world\n");
    return 0;
}
```

```
$ gcc217 hello.c -o hello
```

```
$ hello
hello, world
$
```

Types of Errors

There are five different types of errors in C.

1. Syntax Error
2. Run Time Error
3. Logical Error
4. Semantic Error
5. Linker Error

Computer Software

The instructions that control what a computer does; computer programs. For example:
Operating system (System software)
C compiler (application software)
MS office – (which type of software)
Python (Anaconda, Pycharm, Jupiter) - ?
WhatsApp - ?
MAC - ?

Why !!

A power programmer must know number systems and data representation to fully understand C’s primitive data types

Number Systems

Number Systems

Finite representation of unsigned integers
Finite representation of signed integers
Finite representation of rational numbers (if time)

The Decimal Number System

Name

“decem” (Latin) => ten

Characteristics •

Ten symbols • 0 1 2 3 4 5 6 7 8 9

Positional

2945 ≠ 2495

2945 = (2*103) + (9*102) + (4*101) + (5*100)

(Most) people use the decimal number system

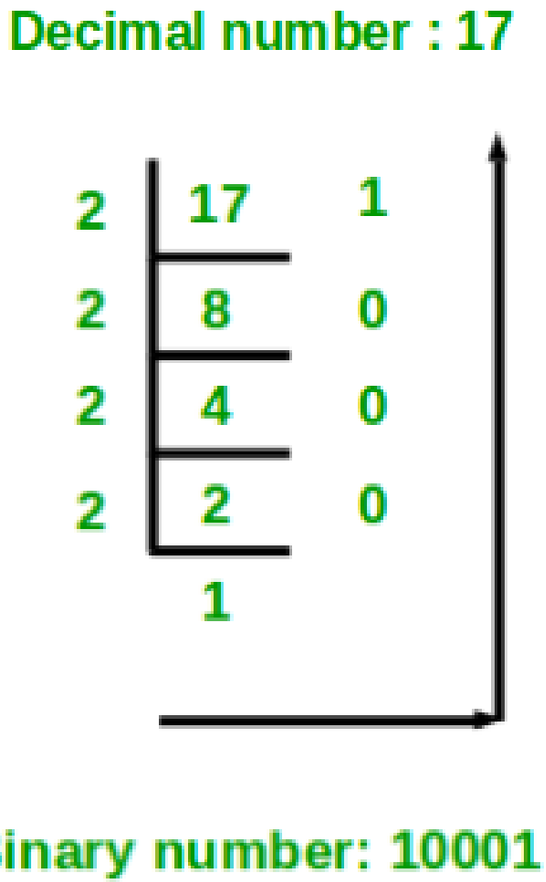
The Binary Number System

Name • “binarius” (Latin) => two
Characteristics
Two symbols
---0 1 ---
Positional
1010B ≠ 1100B
Most (digital) computers use the binary number system
Terminology
Bit: a binary digit
Byte: (typically) 8 bits

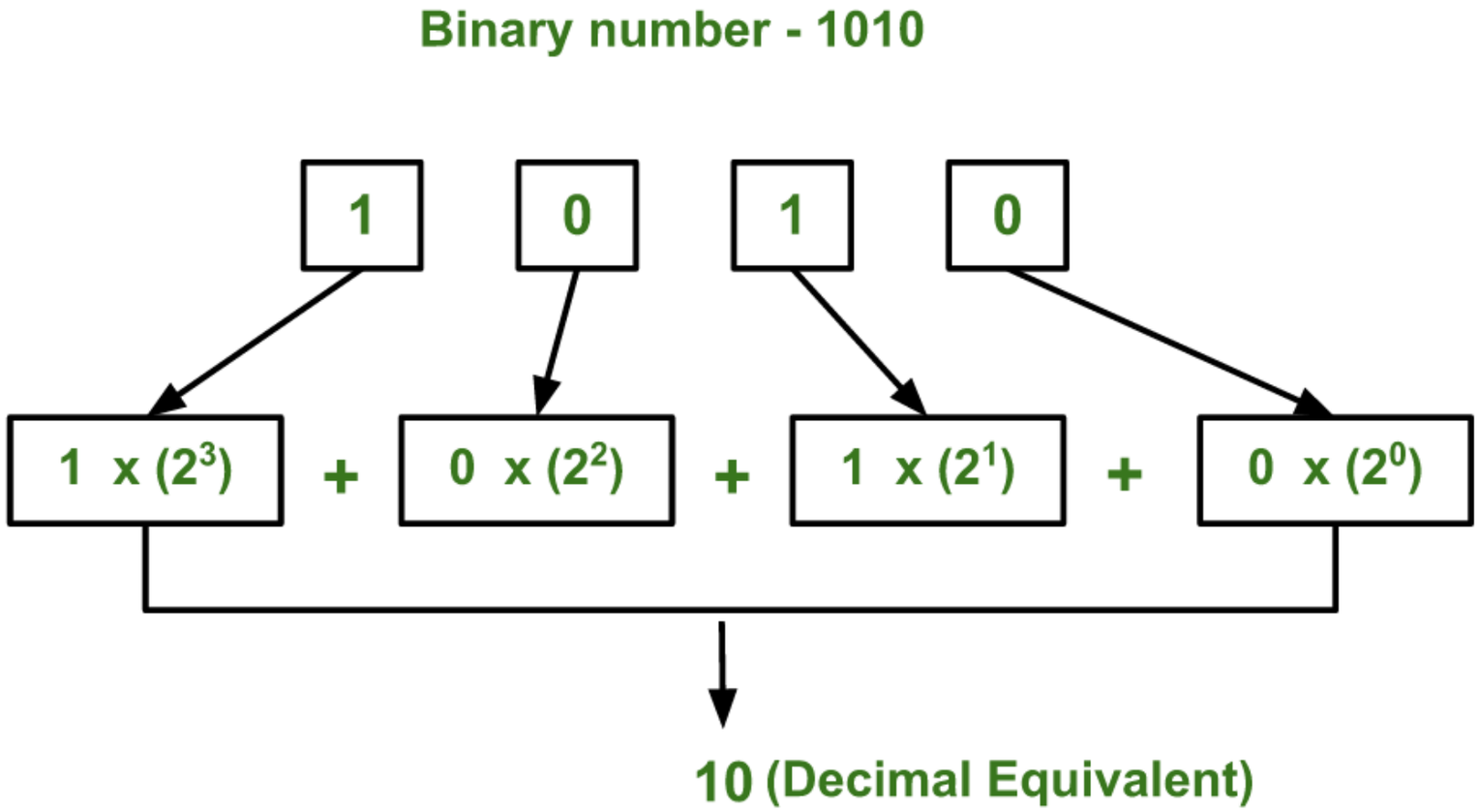
Binary number system

A system in which every number is represented in two digits (either 0 or 1).
For example, 1100 0111 is an 8-bit binary number or binary string.
Digital computer only understands only two signals (on and off) and thus only two alphabets (0 and 1)

Decimal to binary



Binary to decimal



Notation

$(1010)_2$ or 1010 (base 2) means that 1010 is in binary

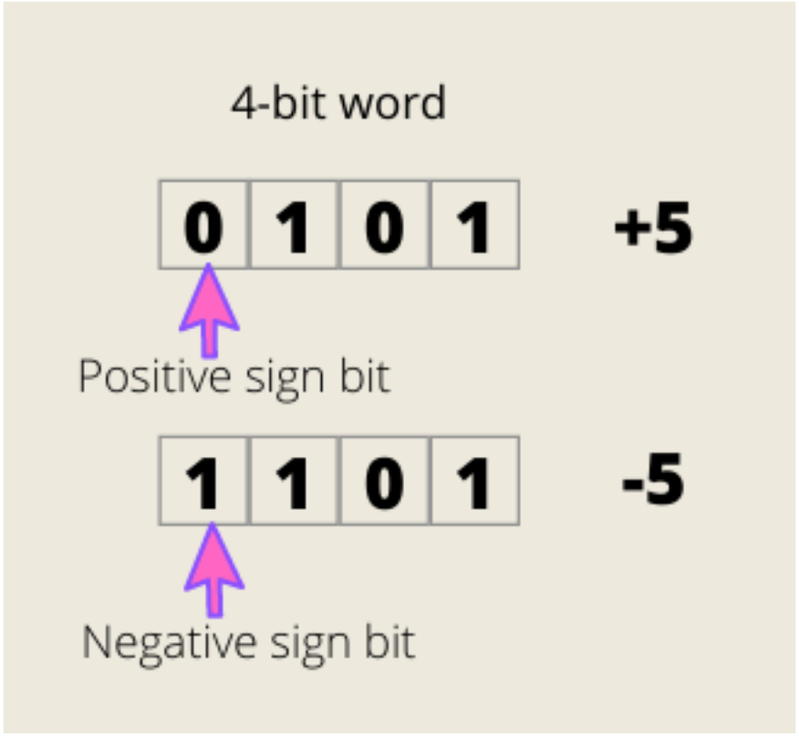
COUNTING IN BINARY NUMBERS

Decimal	Binary	Decimal	Binary
0	0	12	1100
1	1	13	1101
2	10	14	1110
3	11	15	1111
4	100	16	10000
5	101	17	10001
6	110	18	10010
7	111	19	10011
8	1000	20	10100
9	1001	21	10101
10	1010	22	10110
11	1011	23	10111

Signed binary numbers

In case of 4 bits we have 2^4 permutations (why?)
So we have total 16 representations {0000,…,1111} (why?)
If all positives then we have {0, 1, …, 15} (why?)
If we have negatives then {-8,-7,…,-1, 0,…7} (why)

First type of notation - *sign bit*



Q: Use sign bit to find the range of 4-bit binary numbers

All possibilities of 4-bit numbers

0000, 0001, 0010, 0011
0100, 0101, 0110, 0111
1000, 1001, 1010, 1011
1100, 1101, 1110, 1111

Use sign bit to find the range of 4-bit binary numbers

ANS: Problem in representing 0

1111 = -7
1000 = -0
0000 = +0
0111 = 7

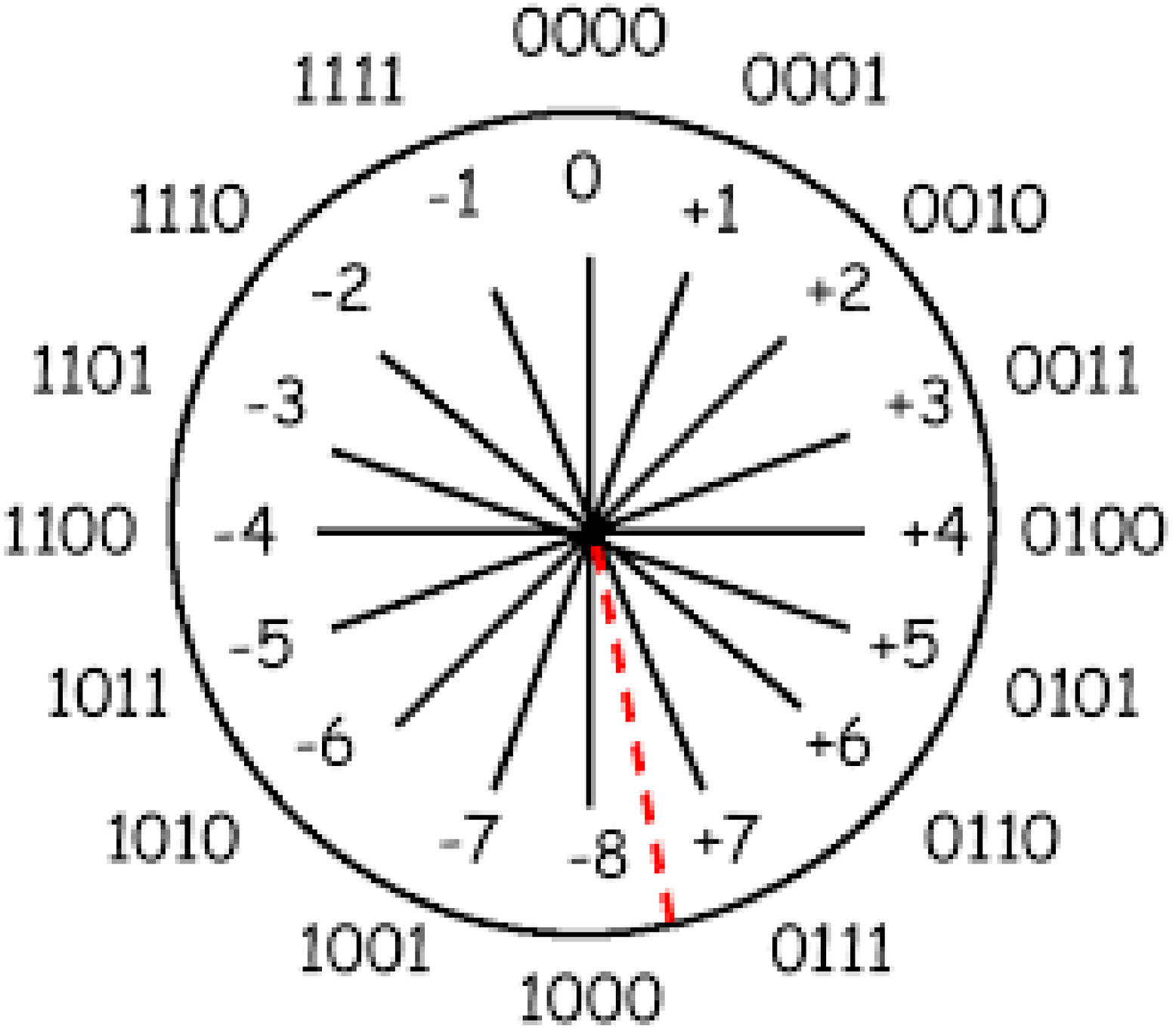
Second type of notation – 2’s complement

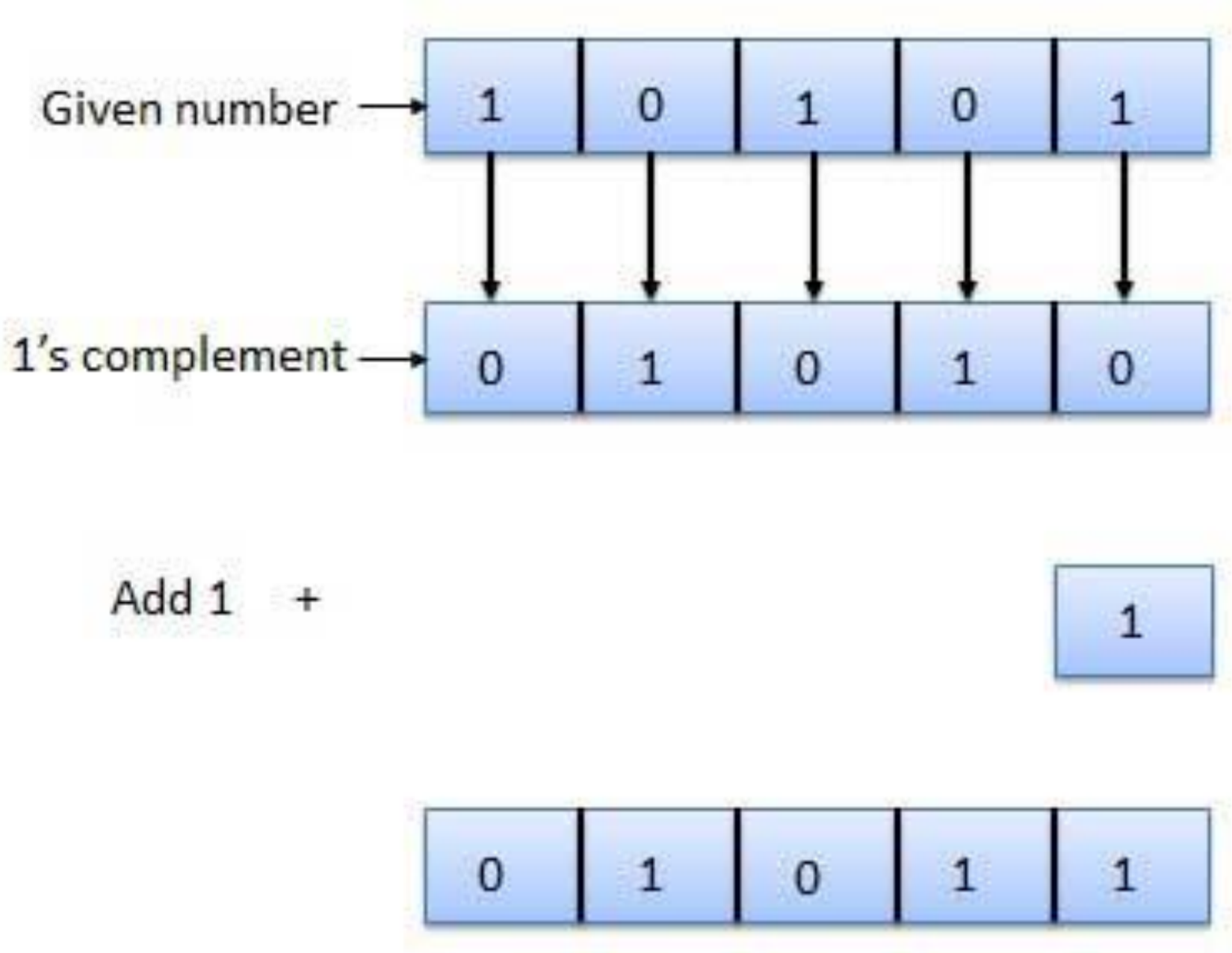
It resolves the the limitation of the signed bit representation.

Negative binary numbers: 2's complement

Flip zeros and ones and add 1

decade number	binary II K
-8	1000
-7	1001
-6	1010
-5	1011
-4	1100
-3	1101
-2	1110
-1	1111
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111


0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
0
+1
+2
+3
+4
+5
+6
+7
-8
-7
-6
-5
-4
-3
-2
-1


Given number
1
0
1
0
1
1's complement
0
1
0
1
0
Add 1
+
1
0
1
0
1
1

1101 into unsigned and signed

Unsigned (+): So 1101 = 13 in decimal

Signed (+/-): If the most significant bit (MSB) is 1 then number is negative. So 1101 means a negative number. Then find its 2's complement to find its value which is 0011. So 1101 = -3.

Find *signed* decimal values for 10100101 and 01111111.

Algorithm

Finite sequence of explicit and unambiguous instructions, which when provided with a set of input values produces an output and then terminates.

Pseudo code

(1) Pseudo code is a generic way of describing an algorithm without using any specific programming language-related notations.

(2) It is an outline of a program, written in a form, which can easily be converted into real programming statements.

Examples Of Algorithms

Write an algorithm to add two numbers entered by user.

```
Step 1: Start
Step 2: Declare variables num1, num2 and sum.
Step 3: Read values num1 and num2.
Step 4: Add num1 and num2 and assign the result to sum.
          sum←num1+num2
Step 5: Display sum
Step 6: Stop
```

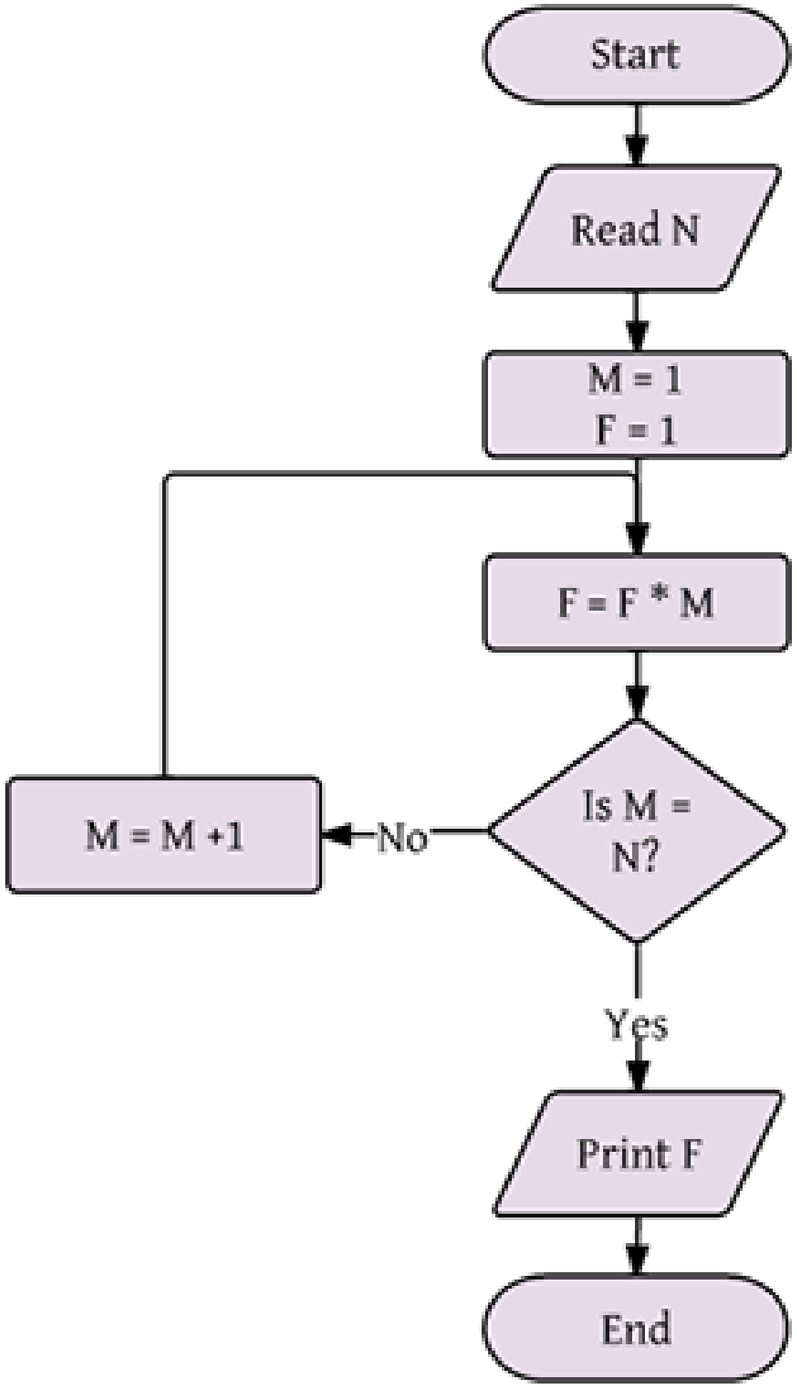
Flowchart

It is a pictorial form of an algo

Boxes represent operations and arrows represent sequence in which the operations are executed

Flowchart – types of boxes

	Start/end	An oval represents a start or end point
	Arrows	A line is a connector that shows relationships between the representative shapes
	Input/Output	A parallelogram represents input or output
	Process	A rectangle represents a process
	Decision	A diamond indicates a decision


Start
Read N
M = 1
F = 1
F = F * M
Is M = N?
No
M = M +1
Yes
Print F
End

Practice

Write and algorithm to input a positive integer and print its multiplication table from 1 to 10. For example if input n = 4 then the table should print 4 x 1 = 4 … 4 x 10 = 40.

Draw the flow chart for the above algorithm