- **Digital Computers and Digital Systems;** ○

Binary Numbers;

- **Number Base Conversions;**
- **Octal and Hexadecimal Numbers;**
- **Complements;**
- **Binary Codes**

- Digital logic is the underlying logic system drives electronic circuit that

board design. ○ Digital logic

is the manipulation of binary values

through printed circuit board

technology that uses circuits and logic

gates to construct the

implementation of computer operations. Digital Logic by

Prashant

- Digital logic is typically embedded into most electronic computers, more.
devices, including
calculators, video games,
watches, and many

- The main semiconductor used in electronic chips is silicon. Silicon is the most abundant element in the Earth's crust after oxygen.

- A lot of the internet information is sent optics. Using fiber optics, light is used

electricity to send the information. over fiber instead of

- The first computer chip was invented by Jack Kilby while working for the company Texas Instruments.
- In 2011 Apple computer chips iPhone became the largest buyer of in the world because of the

- Printed circuit boards are almost always

green

ause they are made from a glass- epoxy, bec which is naturally green.

Digital Computers and Digital

Systems

- AnalogSignal
  - The physical quantities or signals may vary continuously over a specified range.
  - Less Accurate
- Digital Signal

- The physical quantities or signals that assume only discrete value
- Greater accuracy

Analog Digital

Converts analog waveforms into set of numbers and records them. The numbers are converted into voltage stream for representation.

Technology: Analog technology records waveforms as they are.

Can be used in various computing

Uses: Signal:

OSand Windows. Analog signal is a continuous signal which transmits information as a response to changes in physical phenomenon.

Computing and electronics

Digital signals are discrete time signals generated by digital modulation.

platforms and under operating systems like Linux, Unix, Mac

to represent information.
Uses discrete or discontinuous values to represent information.

Representation: Uses continuous range of values
Memory unit: not required required
applications: Thermometer PCs, PDAs
Data

transmissions:not of highquality high quality
Result: not very accurate accurate
Storage capacity: limited high

Process: processed using OPAMP which

uses electronic circuitsusing microprocessor which uses logic circuits
Resposeto Noise: reducing accuracy response are analogin nature
More likely to get affected Less affected since noise
Waves: Denoted by sine waves Denoted by square waves Example: human voice in air
electronic devices

Digital Computer

• An information variable represented by physical quantity. • For digital systems, the variable takes on discretevalues. – Two level, or binary values are the most prevalent values. • Binary values are represented abstractly by:

– Digits 0 and 1

– Words (symbols) False (F) and True (T) – Words (symbols) Low (L) and High (H)

– And words On and Off

• Binary values are represented by values or ranges of values of physicalquantities.

undefine

Logic 0

t

Binary digital signal

$V(t)$

Logic 1

- ***A number system defines how a number can be represented using distinct symbols.***

- A number can be represented differently in different systems.

- For example, the two numbers $(2A)_{16}$ and $(52)_8$ both refer to the same quantity, $(42)_{10}$, but their representations are different.

Number system can be categorized as 1.

Decimal number system

2. Binary number system

3. Octal number system

4. Hexadecimal Number System

- Each number system is associated with a base or radix
- The decimal number system is said to be of base or radix10
- A number in *base r* contains r digits 0,1,2,...,r-1
- Decimal (Base 10): 0,1,2,3,4,5,6,7,8,9

- The word **decimal** is derived from the Latin root **decem**(ten). In this system the **base b = 10** and we use ten symbols.

S = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.

- The word binary is derived from the Latin root **bini** (or two by two).

- In this system the **base b = 2** and we use only two symbols,

$S = \{0, 1\}$

- The symbols in this system are often referred to as **binary digits** or **bits.**
- The word **hexadecimal** is derived from the Greek root hex (six) and the Latin root **decem** (ten).
- In this system the **base b = 16** and we use sixteen symbols to represent a number.

- The set of symbols is

S = **{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}**

- The symbols A, B, C, D, E, F are equivalent to 10, 11, 12, 13, 14, and 15 respectively.
- The symbols in this system are often referred to as **hexadecimal digits.**
- The word octal is derived from the Latin root octo (eight).
- In this system the base b = 8 and we use eight symbols to represent a number.
- The set of symbols is:

S = {0, 1, 2, 3, 4, 5, 6, 7}

System Base Symbols Used by Humans? Used in Computers?

Decimal 10 0, 1, … 9 *Yes No Binary 2 0, 1 No Yes*

Octal 8 0, 1, … 7 *No No*

Hexa decimal *16* 0, 1, … 9, A, B, … F *No No*

Decimal Binary Octal Hexa decimal

0 0 0 0 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

Decimal Binary Octal Hexa

- *Possibilities*

Decimal Octal

- *Example*

Hexadecimal

Binary

$$25_{10} = 11001_2 = 31_8 = 19_{16}$$ Base

Decimal Binary

- *Technique*
  - *Divide by two, keep track of the remainder*

First remainder is bit 0 (LSB, least-significant bit) ○ *Second remainder is bit 1 and so on*

$125_{10} = ?_2$ *2 125 1*

2 62 0

2 31 1

2 15 1

2 7 1

2 3 1

2 1 1

0

$125_{10} = 1111101_2$

$0.6875_{10} = ?_2$

integer fraction

0.6875 x 2 = 1.3750 1 + 0.3750 0.3750 x 2
= 0.7500 0 + 0.7500 0.7500 x 2 = 1.5000 1
+ 0.5000 0.5000 x 2 = 1.0000 1 + 0.0000

$0.6875_{10} = 0.1011_2$

Binary Decimal

- *Technique*

- *Multiply each bit by 2^n, where n* is the "weight" of the *bit*
- *The weight is the position of the bit, starting from 0 on the right*
- *Add the results*

1 0 1 0 1 1

1×2^0 1×2^1 0×2^2 1×2^3 0×2^4 1×2^5 + + + + + 32 + 0 + + + +

8

0 2 1

43_{10}

$1_2 =$

43_{10}

10101

1 1 . 1 1

$+$

1×2^{-2} 1×2^{-1}

2 1

$+ 0.5 + 0.25$

$+$ $+$

1×2^0 1×2^1 $+$

3.75_{10}

3.75_{10}

11.11_2

=

Decimal Octal

- *Technique*
  - *Divide by eight*
  - *Keep track of the remainder*

$125_{10} = ?_8$ *8 125 5*

8 15 7

8 1 1

0

$125_{10} = 175_8$

$0.6875_{10} = ?_8$

integer fraction

0.6875 x 8 = 5.5000 5 + 0.5000 0.5000 x 8
= 4.0000 4 + 0.0000

$0.6875_{10} = 0.54_8$

Octal Decimal

- *Technique*
  - *Multiply each bit by 8^n, where n* is the “weight” of the *bit*
  - *The weight is the position of the bit, starting from 0 on the right*
  - *Add the results*

7 2 4

4×8^0 2×8^1 7×8^2 + + 448 +

16 + 4

468_{10}

468_{10}

$724_8 =$

4 3 . 2 5

+

32 3

+ +

3×8^0 4×8^1 +

5×8^{-2} 2×8^{-1}

+ 0.25 +

0.0781

35.3281_{10}

0

$43.25_8 =$

35.3281 $_1$

Decimal Hexa-Decimal

- *Technique*
  - *Divide by 16*
  - *Keep track of the remainder*

$1234_{10} = ?_{16}$ *16* *1234* *2*

16 77 13=D

16 4 4

0

$1234_{10} = 4D2_{16}$

Hexa-Decimal Decimal

- *Technique*
  - *Multiply each bit by 16^n, where n* is the "weight" of the *bit*
  - *The weight is the position of the bit, starting from 0 on the right*
  - *Add the results*

A B C

$C \times 16^0$ $B \times 16^1$ $A \times 16^2$ + + 12

x 16^0 11×16^1 10×16^2 + + 2560 +

176 + 12 2748_{10}

$ABC_{16} = 2748_{10}$

Octal Binary

- *Technique*
  - *Convert each octal digit to a 3-bit equivalent binary representation*

0 000

1 001 2 010 3 011

4 100 5 101 6 110

7 111

$705_8 = ?_2$

7 0 5

111 000 101

$705_8 = 111000101_2$

Binary Octal

- *Technique*
  - *Group bits in threes, starting on right*
  - *Convert to octal digits*

$1011010111_2 = ?_8$

001 011 010 111

3 2 7

1

$$1011010111_2 = 1327_8$$

Hexa-Decimal Binary

- *Technique*
  - *Convert each hexadecimal digit to a 4-bit equivalent binary representation*

0 0000 8 1000 1 0001 9 1001 2 0010
A 1010 3 0011 B 1011 4 0100 C 1100
5 0101 D 1101 6 0110 E 1110 7 0111
F 1111

$10AF_{16} = ?_2$

1 0 A F

0000 1010 1111

0001

$10AF_{16} =$ $_2$

1000010101111

Binary Hexa-Decimal

- *Technique*
  - *Group bits in fours, starting on*

right ○ *Convert to hexadecimal digits*

$1011010111_2 = ?_{16}$

0010 1101 0111

2 D 7

$1011010111_2 = 2D7_{16}$

Octal Hexa-Decimal

- *Technique*
  - *Convert Octal to Binary*
  - *Regroup bits in fours from right*
  - *Convert Binary to Hexa-Decimal*

$1076_8 = ?_{16}$

1 0 7 6

000 111 110

001

0010 0011 1110 2 E 3 1076_8

$=23E_{16}$

Hexa-Decimal Octal

- *Technique*
  - *Convert Hexa-Decimal to Binary*
  - *Regroup bits in three from right*
  - *Convert Binary to Octal*

$1F0C_{16} = ?_8$

1 F 0 C

1111 0000 1100

0001

001 1

111 7

100 4 4 1

000 0

100 001

$$1F0C_{16} = 17414_8$$

- *Rules for binary addition*

0 + 0 = 0 0 + 1 = 1

1 + 0 = 1

1 1 1 1

1 1

1 + 1 = 10 i.e. 0

with a carry

of 1 1 1 0 1 . 1 0 1

+ 1 0 1 0 1 0 0 0

0 1 1 1 . 0 1 1 .

- *Rules for binary subtraction*

0 – 0 = 0 1 – 1 = 0

1 – 0 = 1

0 1 0 1 1 1 0 1

1 0 1 0 0 1 1 1 0 0 1 0 . 0 1 0 1

-

. 1 1 1 . 0 1 1

0 – 1 = 1, with a borrow 1

1 0 1 1 1

x 1 0 0 1 1 1

1 1 0 1

1 1 1 0 1

0 0 0 0 0

0 0 0 0 0

1 1 1 0 1

```
   1 0 1 0 1 1 0 1 1
 110 0 1 0 1 1 0 1
                      1 1 1 . 1
      0 0 0
      1 0 1 1
        1 1 0
         1 0 1 0
           1 1 0
           1 0 0 1
             1 1 0
        1   1
```

0

1 1 0

0 0 0

- *Two ways of representing signed numbers:*
- *1) Sign-magnitude form, 2) Complement form.*
- *Most of computers use complement form for negative number notation.*
- *1's complement and 2's complement are two different methods in this type.*

- 1's complement of a binary number is obtained by subtracting each digit of that binary number from 1.
- Example

1 1 1 1 01. 0110

1 1 0 1 - 0.

\- 111. 1

0 0 1 0 1 1 0 1

(1's complement of 1101) (1's complement of 101.01)

- 2's complement of a binary number is obtained by adding 1 to its 1's complement.

- Example

1 1 1 1 1 1 111.11

0 0 - 101.01

\-

0 0 1 1 0 1 0 . 1 0

+ + 1 1

0 1 0 0 complement of 101.01)

(2’s complement of 1100)

0 1 0 . 1 1 (2’s

- 9’s complement of a decimal number is obtained by *subtracting each digit of that decimal number from 9.*

- Example

9 9 9 9 82. 2 1 4 5

3 4 6 5 - 7.

\- 999. 7

6 5 3 4 9 9 5 4

(9's complement of 3465) (9's complement of 782.54)

- 10's complement of a decimal number is obtained by adding 1 to its 9's complement.

- Example

9 9 9 9 3 4 9 9 9 . 9 9

6 5 - 7 8 2 . 5 4

\-

6 5 3 4 2 1 7 . 4 5

+ + 1 1

6 5 3 5 complement of 782.54)

(10's complement of 3465)

2 1 7 . 4 6 (10's

- *Obtain 9's complement of subtrahend*
- *Add the result to minuend and call it intermediate result*
- *If carry is generated then answer is positive and add the carry to Least Significant Digit*

(LSD)

- *If there is no carry then answer is negative and take 9's complement of intermediate result and place negative sign to the result.*

1) 745.81 – 436.62

9's complement

7 4 5 . 4 3 6 . -

8 1 6 2

7 4 5 . 8 1 5 6 3 . + 3 7

3 0 9 . 1 9

3 0 9 . 1 1 8 + 1

3 0 9 . 1 9

2) 436.62 - 745.81

- 3 0 9 . 1 9

2 5 4 . + 1 8 6 9

4 3 6 .

9’s complement

6 2

0 . 8 0

4 3 6 . 6 2

- 8 1 7 4 5 .

9’s complement -

3 0 9 . 1 9

As carry is not generated, so take 9’s complement of the intermediate result and add ‘ – ‘ sign to the result

- *Obtain 10’s complement of subtrahend*
- *Add the result to minuend*
- *If carry is generated then ignore it and result itself is answer*

- *If there is no carry then answer is negative and take 10's complement of result and place negative sign to the result.*

1) 745.81 – 436.62

8 1

7 4 5 .

4 3 6 . -   6 2

7 4 5 . 8 1

10's complement 5 6 3 . +   3 8

3 0 9 . 1 1 9 3 0 9 . 1 9 Ignore the

carry

2) 436.62 - 745.81

- 8 1 7 4 5 .

2 5 4 . + 1 9 6 9

4 3 6 .

0 . 8 1

6 2

- 3 0 9 . 1 9

10’s complement

4 3 6 . 6 2

10’s complement

intermediate result and add ‘ – ‘ sign to the result

-

3 0 9 . 1 9

As carry is not generated, so take 10’s complement of the

- *Obtain 1’s complement of subtrahend*
- *Add the result to minuend and call it intermediate result*
- *If carry is generated, then answer is positive*

and add the carry to Least Significant Bit (LSB) ○
If there is no carry, then answer is negative and take 1's complement of intermediate result and place negative sign to the result.

1) 68.75 – 27.50 6 8 . 0 + 1 0 0 . 1 0 0 1 1 0

7 5 1 1 1

2 7 . -

0 0 0 . 1 1 1 0 0 0 1 0

5 0

1's complement

+ 4 1 . 2 5

0 1 0 0 1 1 1

0 1 0 . 0 1 0

\+ 1

0 0 1 . 1 0 0

0 1 0 1 0 0

2) 43.25 - 89.75 4 3 .

0 + 1 0 0 . 1 1 0 1 0 0

2 5

0 1 1

8 9 . -

1 0 1 . 0 1 0 1 1 0 0 0

7 5

1's complement 1's complement

\- 4 6 . 5 0

101111 110000

001.001   10.1100

As carry is not generated, so take 1's complement of the intermediate result and add ' – ' sign to the result

- *Obtain 2's complement of subtrahend*
- *Add the result to minuend*
- *If carry is generated, then ignore it and result itself is answer*
- *If there is no carry, then answer is negative and take 2's complement of result and place*

negative sign to the result.

1) 68.75 – 27.50 6 8 .   0 + 1 0 0 . 1 0 0 1 1 1

7 5   0 0 0

2 7 . -

0 0 0 . 1 1 1 0 0 0 1 0

5 0   1 0 1 0 0 0 0

+ 4 1 . 2 5   1 0 0 1 . 1 0

2's complement

0 0 0

Ignore Carry bit

0 0 1 . 1 0 1 0 1 0

2) 43.25 - 89.75 4 3 .

0 + 1 0 0 . 1 1 0 1 0 0

2 5

1 0 0

8 9 . -

1 0 1 . 0 1 0 1 1 0 0 0

7 5

- 4 6 . 5 0

2’s complement 2’s

1 0 . 1 1 0 0

0 0 1 . 0 1 0

1 0 0 0 0

complement

1 0 1 1 0 0 1

As carry is not generated, so take 2’s complement of the intermediate result and add ‘ – ‘ sign to the result

Digital data is represented, stored and transmitted as groups of binary digits also known as binary code.

Weighted codes: In weighted codes, each digit is assigned a specific weight according to its position. **Non-weighted codes:** In non-weighted codes are not appositionally weighted.

Reflective codes: A code is reflective when the code is self complementing. In other words, when the code for 9 is thecomplement the code for 0, 8 for 1, 7 for 2, 6 for 3 and 5 for 4.

Sequential codes: In sequential codes, each succeeding 'code is one binary number greater than its preceding code. **Alphanumeric codes:** Codes used to represent numbers, alphabetic characters, symbols

Error detecting and correcting codes: Codes which allow error detection and correction are called error detecting and' correcting codes.

Decimal Codes

- Each decimal digit, 0 through 9, is coded by 4-bit binary number
- 8, 4, 2 and 1 weights are attached to each bit
- BCD code is weighted code
- 1010, 1011, 1100, 1101, 1110 and 1111 are illegal codes
- Less efficient than pure binary
- Arithmetic operations are more complex than in pure binary
- Example

1 4 0001

Decimal

0100

BCD

Binary 1110