Lecture 3a
Digital Logic - Number System

Chintan Kr Mandal

যাদবপুর বিশ্ববিদ্যালয়
JADAVPUR UNIVERSITY

Outline [1] [2]

- This discussion deals with the representation of numerical data, with emphasis on those representations that use only two symbols, 0 and 1.
- We discuss special methods of representing numerical data that afford protection against various transmission errors and component failures.

Why the Number System Again ??

We already know the DECIMAL NUMBER SYSTEM ....
IS IT NOT ENOUGH ????????

NO ...

- We have discussed where the Boolean logic system is closed under $\{0, 1\}$.
- All binary and unary operations will thus only computed on $\{0, 1\}$.
- All numbers thus will be represented by $\{0,1\}$, i.e. only *two numbers*.
- However, we know a number system, known widely as the **Decimal System**
- We thus require some relations which can connect the Decimal system and the system in which numbers are represented using $\{0, 1\}$

Decimal System

- We are familiar with the decimal number system because we use the decimal system every day.
- Although decimal numbers are commonplace, their weighted structure is often not understood.
- In this section, we review the structure of the decimal numbers, *which will be useful in understanding the binary number system*

Decimal System

- The **decimal number system** has ten digits, **0** through **9**
- Each of the ten digits represents a certain quantity.
- Since there are ten distinct digits used in the decimal number system, therefore, the decimal number system has a base (or *radix*) of 10.
- The ten symbols (**digits**) do not limit oneself to express only ten different quantities because *one can use the various digits in appropriate positions within a number to indicate the magnitude of the quantity.*

E.g If we want to express the quantity twenty-three, we use the **digit 2** to represent the quantity **twenty** and the **digit 3** to represent the quantity **three**

The weighted structure of Decimal Numbers I

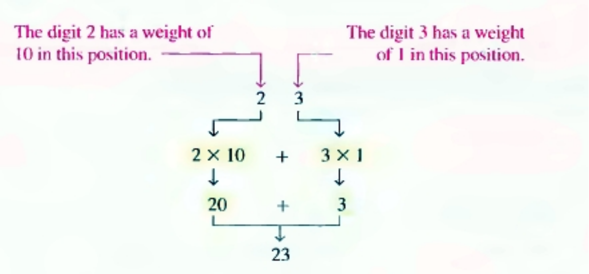


Figure: The weighted structure of Decimal Numbers

The weighted structure of Decimal Numbers II

- The position of each digit in a decimal number indicates the magnitude of the quantity represented and can be assigned a **weight**.
- The weights for whole numbers are positive powers of ten that increase from right to left, beginning with 10^1

$$\ldots\ 10^5\ 10^4\ 10^3\ 10^2\ 10^1\ 10^0$$

- For fractional numbers, the weights are negative powers of ten that decrease from left to right beginning with 10^{-1}.

$$10^2\ 10^1\ 10^0.10^{-1}\ 10^{-2}\ 10^{-3}$$

Counting in Decimal

Counting in Decimal System !!!!!! I

Figure: Counting Sheep ..... Whoops !!

Counting in Decimal System !!!!!! II

Let us review how we count in the Decimal System

1. We start at **zero** and count upto **nine** before we run out of digits.
2. We then start another digit position (to the left) and continue counting 10 through 99
3. At this point we have exhausted all two digit combinations, so a third digit position is needed to count from 100 to 999.

Counting in Decimal System !!!!!! III

0	1_	2_	3_	4_	5_	6_	7_	8_	9_	10_
0	10	20	30	40	50	60	70	80	90	100
1	11	21	31	41	51	61	71	81	91	
2	12	22	32	42	52	62	72	82	92	
3	13	23	33	43	53	63	73	83	93	
4	14	24	34	44	54	64	74	84	94	
5	15	25	35	45	55	65	75	85	95	
6	16	26	36	46	56	66	76	86	96	
7	17	27	37	47	57	67	77	87	97	
8	18	28	38	48	58	68	78	88	98	
9	19	29	39	49	59	69	79	89	99	

Figure: Counting in Decimal

Binary Number System

Binary Numbers

- The binary number system is simply another way to represent quantities.
- The binary system is less complicated than the decimal system because it has only two digits.

Note The decimal system with its ten digits is a base-ten system; the binary system with its two digits is a base-two system.

- The two binary digits (**bits**) are **1** and **0**

The weighted structure of Binary Number System

- Similar to the Decimal System, we can also figure out the weighted structure of the Binary Number System
- The position of a 1 or 0 in a binary number indicates its weight, or value within the number, *just as the position of a decimal digit determines the value of that digit.*
- The weights in a binary number are based on powers of two.
- The weighted structure of a binary number is

$$2^{n-1} \ \ldots \ 2^3 \ 2^2 \ 2^1 \ 2^0 \ . \ 2^{-1} \ 2^{-2} \ \ldots \ 2^{-n}$$

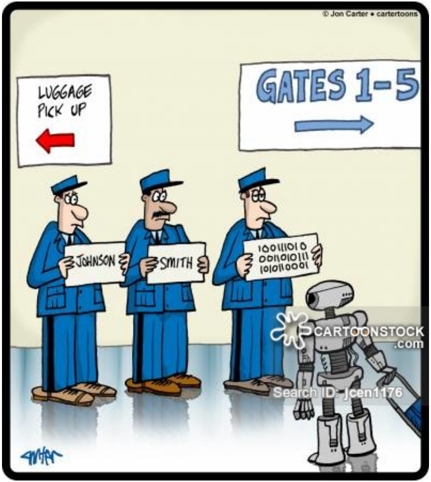
where n is the number of bits from the binary point.

Note The right-most bit is the **LSB** (*least significant bit*) in a binary whole number and has a weight of 2^0

Note The left-most bit is the **MSB** (*most significant bit*); its weight depends on the size of the binary number

Counting in Binary

Counting in Binary I



Counting in Binary II

A comparable situation occurs when we count in binary, except that we have only twp digits, called **bits**.

1. Begin counting; 0,1
2. At this point, we have exhausted both digits, so include another digit postion and continue; 10, 11
3. We have again exhausted all combinations of two digits, so a third position is required.
4. Continue to count; 100, 101, 110 and 111
5. and so on . . .

Counting in Binary III

DECIMAL NUMBER	BINARY NUMBER			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Figure: Notice
[a] the patterns with which the 1s and 0s alternate in each column
[b] four (4) bits are required to count from 0 to 15

Counting in Binary IV

- The value of a bit is determined by its position in the number
- In general, with n bits one can count up to a number equal to $2^n - 1$

E.g. With five bits ($n = 5$), we can count from **zero to thirty-one**

$$2^5 - 1 = 32 - 1 = 31$$

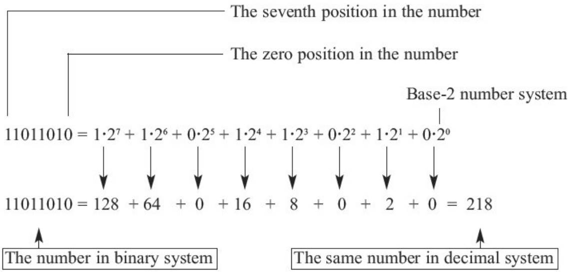
Conversions !!

Binary-to-Decimal Conversion

Binary-to-Decimal Conversion I

The decimal value of any binary number can be found by adding the weights of all bits that are 1 and discarding the weights of all bits that are 0

Example 1



Binary-to-Decimal Conversion II

Example 2

Convert the fractional binary number 0.1011 to decimal

Weight		2^{-1}	2^{-2}	2^{-3}	2^{-4}
Binary Number	0 .	1	0	1	1

$$\begin{aligned} 0.1011 &= 2^{-1} + 2^{-3} + 2^{-4} \\ &= 0.5 + 0.125 + 0.0625 \\ &= \mathbf{0.6875} \end{aligned}$$

Decimal Integer-to-Binary Conversion

Sum-of-Weights Method

- Determine the set of binary weight(s) whose sum is equal to the decimal number
- An easy way to remember binary weights is that the lowest is 1, which is 2^0.
- By doubling any weight, one can get the next higher weight;

Thus 64, 32, 16, 8, 4, 2, 1

E.g. $9 = 8 + 1$ or $9 = 2^3 + 2^0$

Placing 1s in the appropriate weight positions, 2^3 and 2^0, and 0s in the 2^2 and 2^1 positions determines the binary number for decimal 9

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

1 0 0 1

Repeated Division-by-2 Method I

- This is a popular method for converting a Decimal number to a number equivalent in the binary system.
- This is a *systematic* method of converting whole numbers from decimal to binary by *repeated division-by-2* process.
- To convert a Decimal Integer number to its binary equivalent
  1. Begin by dividing the *given number* by 2
  2. Divide each resulting quotient by 2 until there is a 0 whole quotient.
  3. The **remainders** generated by each division form the **binary number**
  4. The *first remainder* to be produced is the **LSB (least significant bit)** in the binary number
  5. The *last remainder* to be produced is the **MSB (most significant bit)** in the binary number

Repeated Division-by-2 Method II

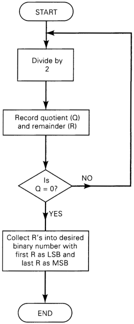
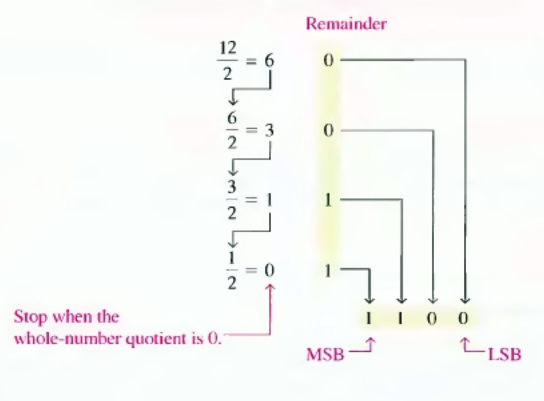


Figure: Flowchart for repeated-division method of decimal-to-binary conversion of integers.

The same process can be used to **convert** a *decimal integer* to any *other number system*

Repeated Division-by-2 Method III

E.g Convert $(12)_{10}$ to the Base-2 equivalent.



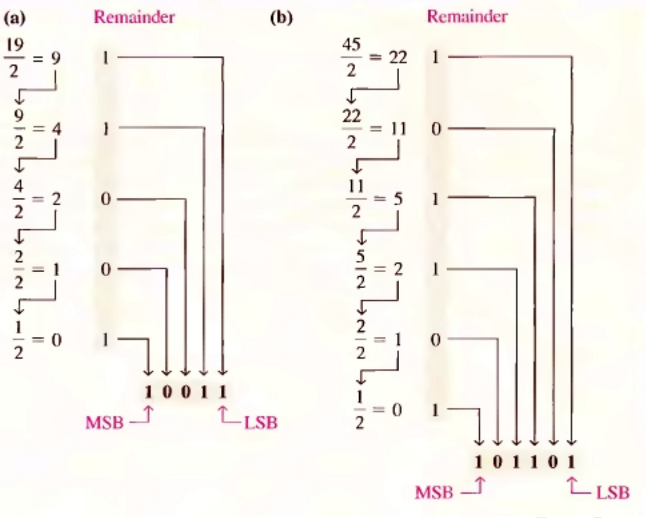
Exercise !!

Convert the following decimal numbers to binary

(a) 19

(b) 45

Answer


(a)
Remainder
19/2 = 9
1
9/2 = 4
1
4/2 = 2
0
2/2 = 1
0
1/2 = 0
1
1 0 0 1 1
MSB
LSB
(b)
Remainder
45/2 = 22
1
22/2 = 11
0
11/2 = 5
1
5/2 = 2
1
2/2 = 1
0
1/2 = 0
1
1 0 1 1 0 1
MSB
LSB

Decimal Fractions-to-Binary Conversion

Sum-of-Weights Method

The Sum-of-Weights method applied for converting a Decimal Integer to its equivalent Binary number can also be applied to convert Decimal Fractions to its Binary equivalent

- An easy way to remember fractional binary weights is that the most significant weight is 0.5, which is 2^{-1}
- By halving any weight, we get the nest lower weight; thus a list of binary weights would be

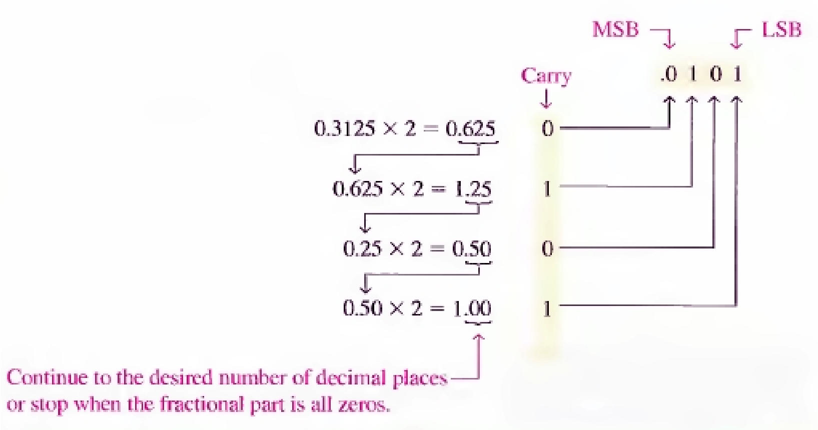
$$0.5,\ 0.25,\ 0.125,\ 0.0625,\ \ldots \equiv 2^{-1},\ 2^{-2},\ 2^{-3},\ 2^{-4},\ \ldots$$

E.g. $0.625 = 0.5 + 0.125 = 2^{-1} + 2^{-3} = 0.101$

Repeated Multiplication-by-2

- Decimal fractions can be converted to binary by repeated multiplication by 2
- To convert a decimal fraction to its binary equivalent
  1. Begin by multiplying the Decimal fraction by 2
  2. Thereafter, multiply each resulting fractional part of the product by 2 until the fractional product is zero or until the desired number of decimal places is reached.
  3. The carried digits, or **carries**, generated by the multiplications produce the binary number.
  4. The *first carry* produced is the **MSB**
  5. The *last carry* is the **LSB**

Example


MSB
LSB
Carry
.0 1 0 1
0.3125 × 2 = 0.625
0
0.625 × 2 = 1.25
1
0.25 × 2 = 0.50
0
0.50 × 2 = 1.00
1
Continue to the desired number of decimal places
or stop when the fractional part is all zeros.

Exercise

Convert the following Decimal numbers to its binary equivalent

1. 12
2. 0.235
3. 10.625

Other Number Systems

Octal System

- The octal number system is often used in digital computer work.
- The **octal number system** has a base / radix of *eight*.
- It has eight possible digits: 0, 1, 2, 3, 4, 5, 6, 7
- The digit positions in an octal number have weights as follows:

8^4	8^3	8^2	8^1	8^0	8^{-1}	8^{-2}	8^{-3}	8^{-4}	8^{-5}

•
octal point

Counting in Octal

- The largest octal digit is 7.
- A digit *position* is incremented upward from 0 to 7, once it reaches 7.

E.g 1 65, 66, 67, 70, 71

E.g. 2 275, 276, 277, 300

Octal-to-Decimal Conversion

- An octal number can be easily converted to its decimal equivalent by multiplying each octal digit by its positional weight

E.g. 1

$$\begin{aligned}(372)_8 &= 3 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 \\ &= 3 \times 64 + 7 \times 8 + 2 \times 1 \\ &= (250)_{10}\end{aligned}$$

E.g. 2

$$\begin{aligned}(24.6)_8 &= 2 \times 8^1 + 4 \times 8^0 + 6 \times 8^{-1} \\ &= (20.75)_{10}\end{aligned}$$

Decimal-to-Octal Conversion

A decimal integer can be converted to octal by using the same repeated-division method that was used in the Decimal-to-binary conversion but with a division factor of 8 instead of 2

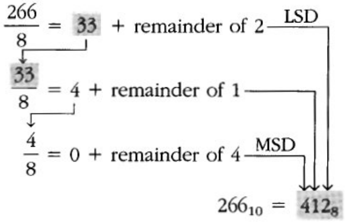


Figure: (a) The first remainder becomes the **least significant digit (LSD)**
(b) The last remainder becomes the **most significant digit (MSD)**

Octal-to-Binary Conversion

- The primary advantage of the octal number system is the easr with which conversion can be made between binary and octal numbers
- The conversion from octal to binary is performed by converting *each* octal digit to its three-bit binary equivalent.
- The eight possible digits are converted as follows

Octal Digit	0	1	2	3	4	5	6	7
Binary Equivalent	000	001	010	011	100	101	110	111

- Using these conversions, one can convert any octal number to binary by individually converting each digit

Example: Octal-to-Binary Conversion

E.g. 1 Convert $(472)_8$ to its equivalent binary

$$\begin{array}{ccc} 4 & 7 & 2 \\ \downarrow & \downarrow & \downarrow \\ 100 & 111 & 010 \end{array}$$

$(472)_8 \equiv (100111010)_2$

E.g. 2 Convert $(5431)_8$ to its equivalent binary

$$\begin{array}{cccc} 5 & 4 & 3 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 101 & 100 & 011 & 001 \end{array}$$

$(5431)_8 \equiv (101100011001)_2$

Hexadecimal Number System I

- The **hexadecimal number system** uses base 16.
- It has **16 possible digit symbols**: *0 through 9* and *letters A, B, C, D, E, F*

Note Each hexadecimal digit represents a group of four binary digits.

Hexadecimal Number System II

Hexadecimal	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Counting in Hexadecimal

- When counting in hex, each digit position can be incremented (increased by 1) from 0 to F
- Once a digit position reaches the value F, it is reset to 0, and the next digit position is incremented.

E.g. 1 38, 39, 3A, 3B, 3C, 3D, 3E, 3F, 40, 41, 42

E.g. 2 6F8, 6F9, 6FA, 6FB, 6FC, 6FD, 6FE, 6FF, 700

Hex-to-Decimal Conversion

- A hex number can be converted to its decimal equivalent by using the fact that each hex digit position has a weight that is a power of 16
- The LSD has a weight of $16^0 = 1$; the next higher digit positions has a weight of $16^1 = 16, 16^2 = 256, \ldots$
- The conversion process is as

$$\begin{aligned}(2AF)_{16} &= 2 \times 16^2 + 10 \times 16^1 + 15 \times 16^0 \\ &= 512 + 160 + 15 \\ &= (687)_{10}\end{aligned}$$

Decimal-to-Hex Conversion

We repeat the repeated-division method using 8

E.g. Convert $(423)_{10}$ to Hex

$$\frac{423}{16} = \boxed{26} + \text{remainder of } 7$$

$$\frac{\boxed{26}}{16} = 1 + \text{remainder of } 10$$

$$\frac{1}{16} = 0 + \text{remainder of } 1$$

$$423_{10} = \boxed{1A7_{16}}$$

Hex-to-Binary Conversion

- Like the octal number system, the hexadecimal number system is used primarily as a "shorthand" method for representing binary numbers.
- *Each* hex digit is converted to its four bit binary equivalent.

E.g. Convert $(9F2)_{16}$ to its binary equivalent

$$
\begin{array}{lcccccccccccccc}
9F2_{16} = & & & 9 & & & & F & & & & & 2 & & \\
 & & & \downarrow & & & & \downarrow & & & & & \downarrow & & \\
 & = 1 & 0 & 0 & 1 & & 1 & 1 & 1 & 1 & & 0 & 0 & 1 & 0
\end{array}
$$

$$= 100111110010_2$$

Binary-to-hex Conversion

- Conversion from binary to hex is just the reverse process of Hexadecimal-to-binary conversion
- The binary number is grouped into groups of *four* bits, and each group is converted to its equivalent hex digit
- The grouping should be done *from right to left* for binary digits before the fractional point **and** *from left to right* after the fractional point

E.g.

$$1110100110_2 = \underbrace{0\,0\,1\,1}_{3}\,\underbrace{1\,0\,1\,0}_{A}\,\underbrace{0\,1\,1\,0}_{6} = 3A6_{16}$$

Summary of Conversions

1. When converting from from binary [or octal or hex] to Decimal, use the method of taking the **weighted sum of each digit position**
2. When converting from Decimal to binary [or octal or hex], use the method of **repeatedly dividing by 2 [or 8 or 16] and collecting the remainders**
3. When converting from binary to octal [or hex], **group the bits in groups of three [or four], and convert each group into the correct octal [or hex] digit.**
4. When converting from octal [or hex] to binary, **convert each digit into its three-bit [or four-bit] equivalent.**
5. When converting from octal to hex [or vice-versa], **first convert to binary; then convert the binary into the desired number system.**

Check the following !!

Check also the following Number system

- Binary-Coded-Decimal (BCD)
- Error Detection Codes
- Gray Coded
- Alphanumeric Codes - ASCII Codes
- Parity Method for Error Detection
- Conversion between various other number systems.

References

[1] Thomas L. Floyd.
Digital Fundamentals, 8th edition.
Pearson Education Inc., 2003.

[2] Ronald J. Tocci and Neal S. Widmer.
Digital systems: principles and applications.
Prentice Hall, 2001.

QUESTIONS !!!