PDS0101

Introduction to Digital Systems

Number system operations and codes

MMU
MULTIMEDIA UNIVERSITY

Lecture outcome

- By the end of today's lecture you should know
  - Decimal and Binary Number Systems – Representation and Counting, Conversions – Binary to Decimal, Decimal to Binary Arithmetic in binary-Addition, Subtraction, Multiplication, and Division
  - Octal Number System – Representation, conversions from decimal to octal, octal to decimal, binary to octal, octal to binary, Arithmetic-Addition, Subtraction
  - Hexadecimal Number System - Representation, conversions from decimal to Hexadecimal, Hexadecimal to decimal, binary to Hexadecimal, Hexadecimal to binary, Arithmetic-Addition, Subtraction

Decimal Number System

- The decimal number system has 10 digits 0 through 9
- The decimal numbering system has a base of 10 with each position weighted by a factor of 10
  - ....$10^5\ 10^4\ 10^3\ 10^2\ 10^1\ 10^0.\ 10^{-1}\ 10^{-2}\ 10^{-3}\ 10^{-4}\ 10^{-5}$...
  - $14.2 = 1\times10^1 + 4\times10^0 + 2\times10^{-1}$

- Express decimal 656 as a sum of the values of each digit.

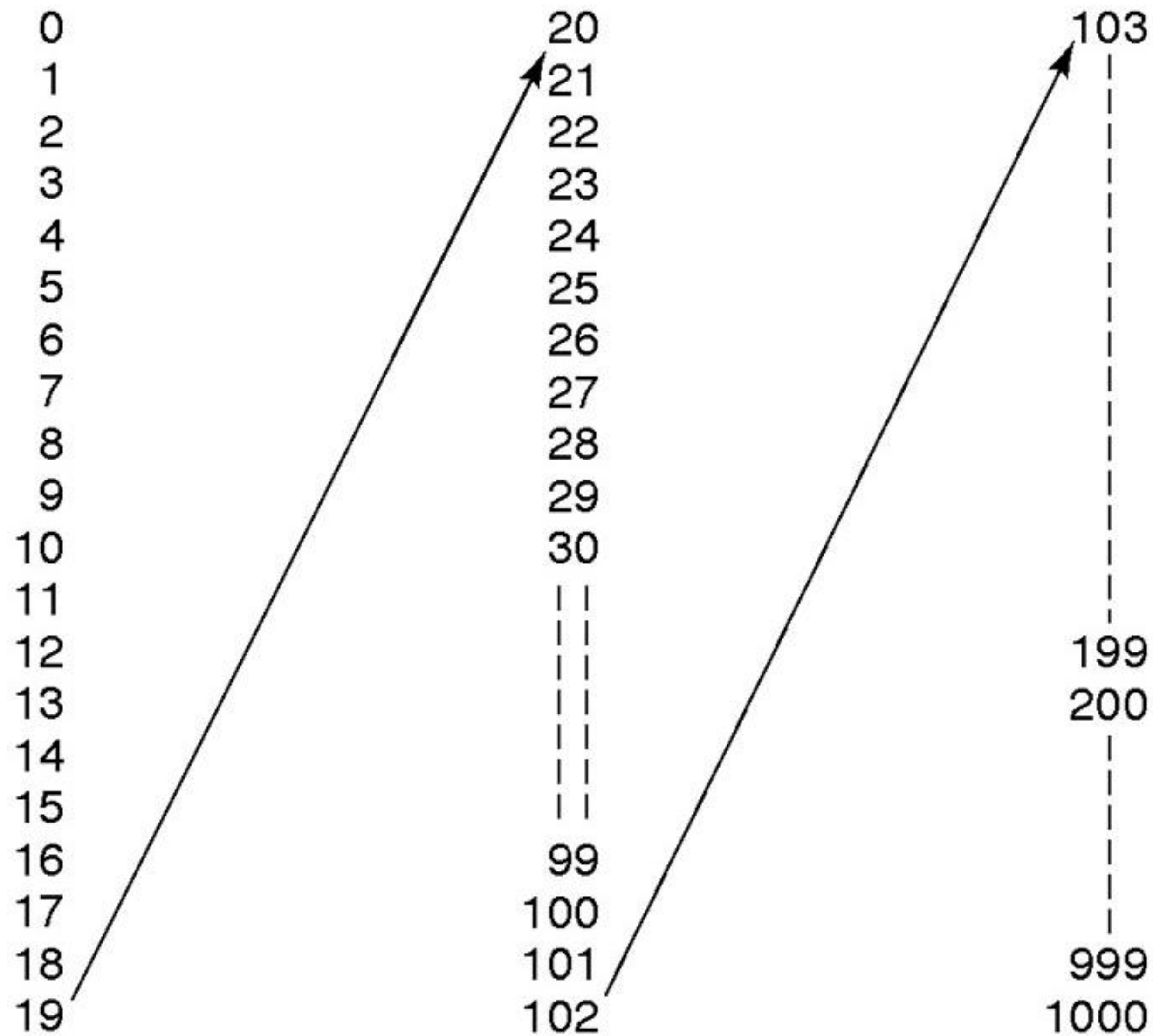
  Solution

$$6\ \ 5\ \ 6$$

$$6\times100 + 5\times10 + 6\times1$$

$$6\times10^2 + 5\times10^1 + 6\times10^0$$

Decimal Counting



Decimal Number System

- Express the decimal number 75.68 as a sum of the values of each digit.

$$7 \quad 5 \,.\, 6 \quad 8$$

$$7 \times 10 \; + \; 5 \times 1 + 6 \times 0.1 \; + \; 8 \times 0.01$$

$$7 \times 10^{1} + 5 \times 10^{0} + 6 \times 10^{-1} + 8 \times 10^{-2}$$

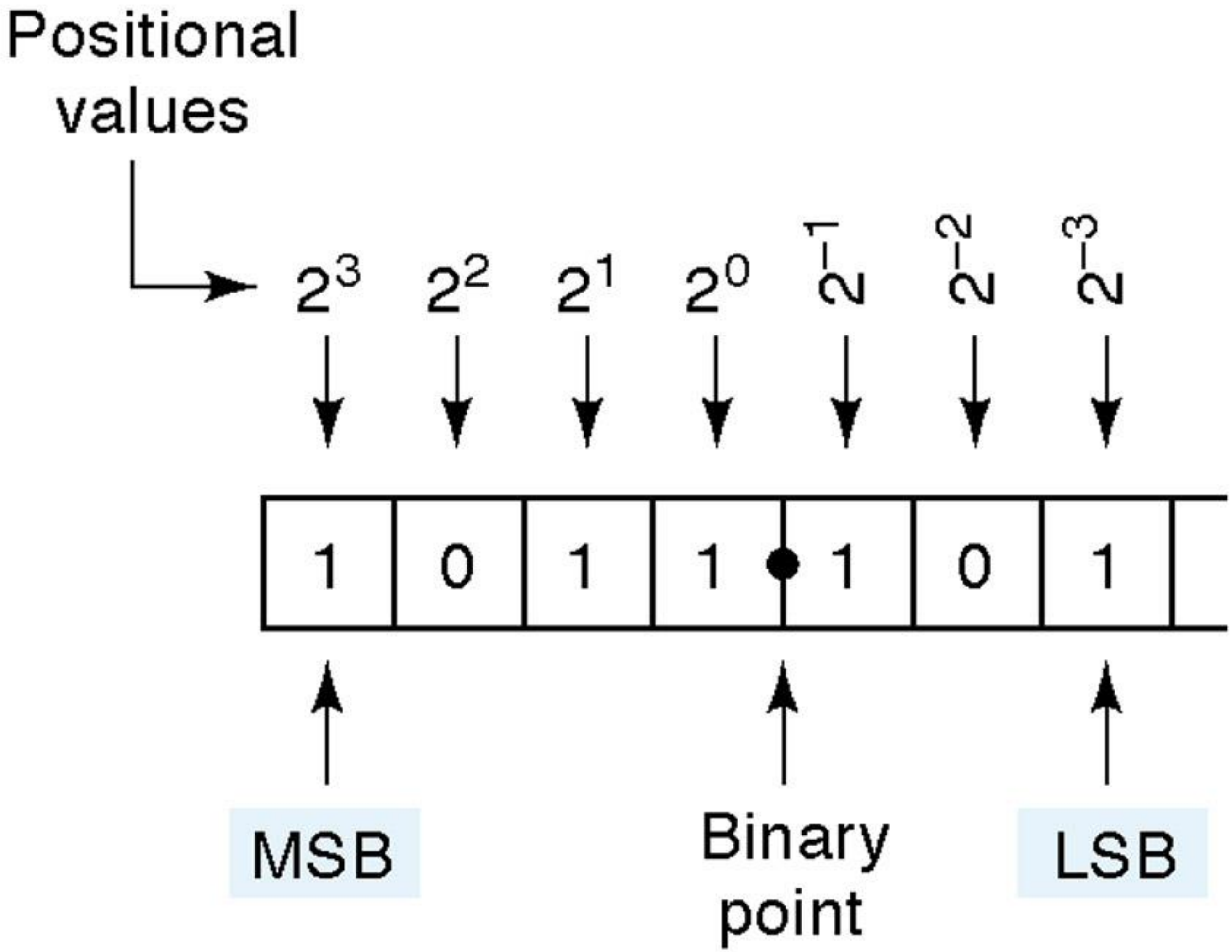
Binary number system

- The binary numbering system has 2 digits 0 and 1
- The binary numbering system has a base of 2 with each position weighted by a factor of 2

$$\ldots.2^5\ 2^4\ 2^3\ 2^2\ 2^1\ 2^0\ .\ 2^{-1}\ 2^{-2}\ 2^{-3}\ 2^{-4}\ 2^{-5}\ \ldots$$

- $10111_2 = (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$
  - LSB (Least Significant Bit) – right-most bit with a weight of 2^0
  - MSB (Most Significant Bit) – left-most bit with a weight of 2^n-1
- Largest decimal number that can be represented by a given number of bits = 2^n-1
  - Ex: If the number of bits = 6, largest decimal number = 2^6-1=63

Binary number system



Counting in binary

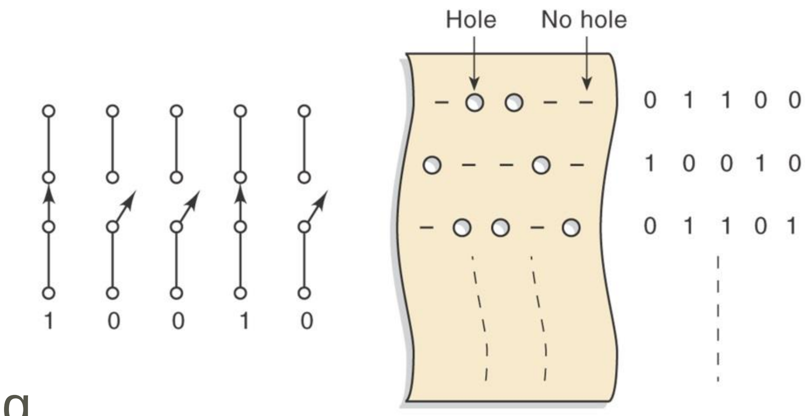
DECIMAL NUMBER	BINARY NUMBER			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Counting in binary

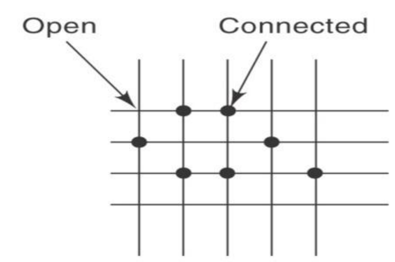
- Each time the unit bit changes from a 1 to 0, the two (2 power of 1) position will toggle (change states).
- Each time the twos position changes from 1 to 0, the four (2 power of 2) position will toggle (change states).
  - For e.g.: with two bits we can go through 2 power of 2=4 counts (00 through 11).
  - With four bits can go through 16 counts (0000 through 1111).
- The last count will always be all 1s and is equal to 2 power of N minus 1 in the decimal system.
  - For eg: 4 bits, the last count is 1111_2=(2 power of 4) –1=15 (in decimal)

Why use binary?

- Most of the devices associated with computers or components which are inside computers are bi-stable devices.
  - Wires and rows form a matrix. This forms the foundation for programmable logic devices.
  - Open and closed switches
  - Paper Tape
- Other two state devices:
  - Light bulb (off or on)
  - Diode (conducting or not conducting
  - Relay (energized or not energized)
  - Transistor (cutoff or saturation)
  - Photocell (illuminated or dark)



(a) (b)



(c)

Binary to decimal conversion

- Any binary number can be converted to decimal by multiplying the weight of each position with the binary digit and adding together
- Example: Convert the binary number 11011_2 into decimal.

Solution

Binary number	1		1		0		1		1
Power of 2 position	2^4	+	2^3	+	2^2	+	2^1	+	2^0

$$(2^4 \times 1) + (2^3 \times 1) + (2^2 \times 0) + (2^1 \times 1) + (2^0 \times 1)$$

Decimal value $16 + 8 + 0 + 2 + 1 = 27_{10}$

Binary to decimal conversion

- Example: Convert the binary number 110.11_2 to its decimal equivalent.

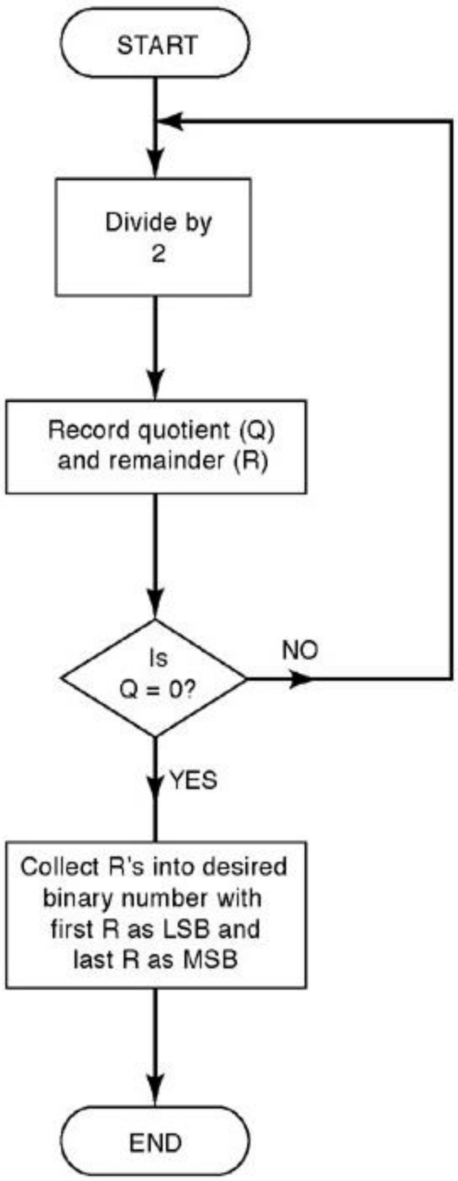
Solution :

Binary number	1		1		0	.	1		1
Power of 2 position	2^2	+	2^1	+	2^0	.	2^{-1}	+	2^{-2}
	$(2^2\times1)$	+	$(2^1\times1)$	+	$(2^0\times0)$	.	$(2^{-1}\times1)$	+	$(2^{-2}\times1)$
Decimal value	4	+	2	+	0	.	0.5	+	0.25

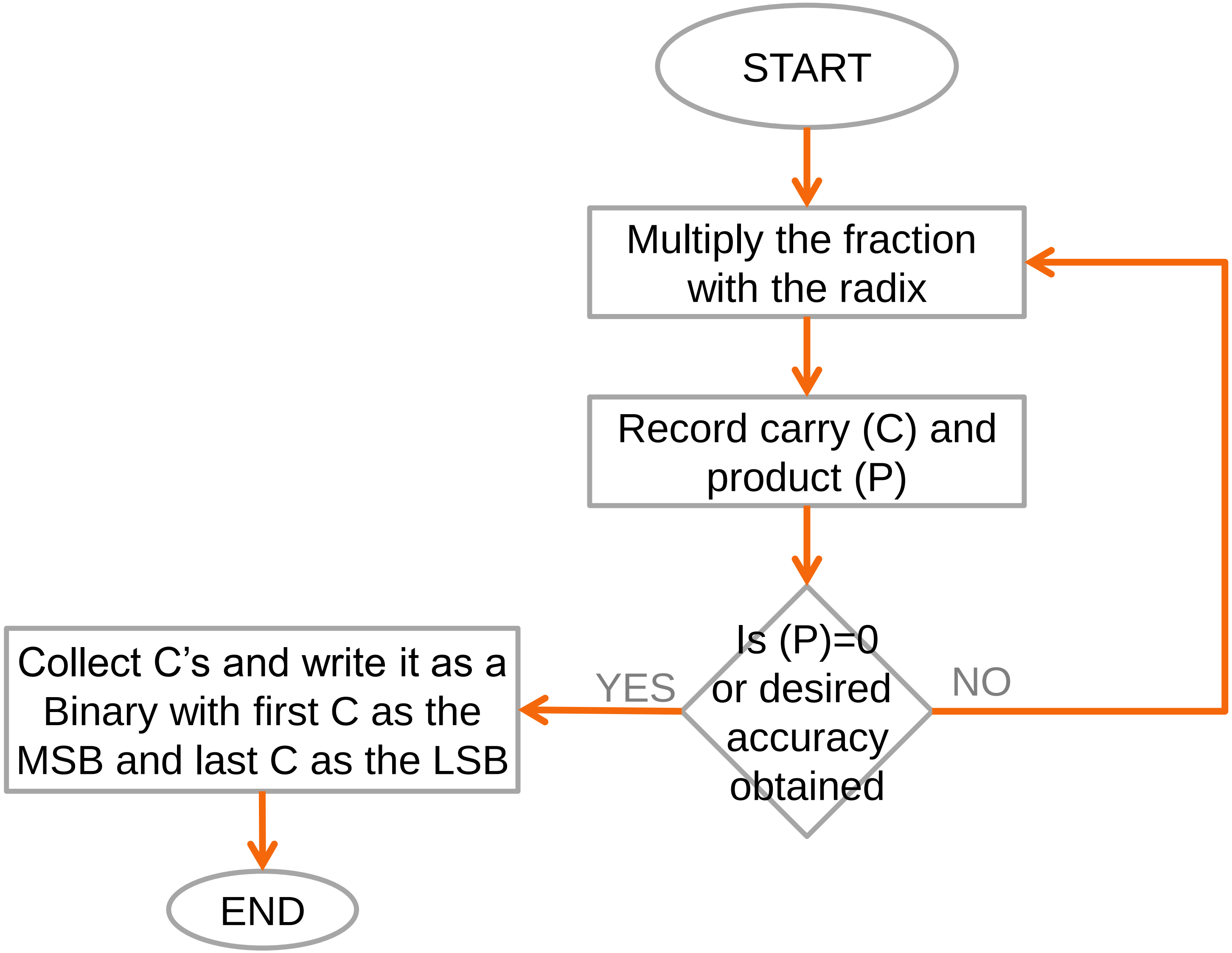
$= (6.75)_{10}$

Converting decimal integers

- For decimal integer to binary, we have to divide by 2 - the radix of binary number system.
- In general, for converting from decimal integer to any number system, we have to divide by the corresponding radix



Converting decimal fractions



Decimal-to-Binary Conversion

- Sum-of-weight method
- Binary weights

256	128	64	32	16	8	4	2	1
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

- $357 = 256 + 64 + 32 + 4 + 1 \rightarrow 101100101_2$

- Binary weights

1024	512	256	128	64	32	16	8	4	2	1
2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

- $1937 = 1024 + 512 + 256 + 128 + 16 + 1 \rightarrow 11110010001_2$

Decimal-to-Binary Conversion

- Repeated *division* by 2 method
- Steps:
  - Divide the decimal number by 2
  - Write the remainder after each division until a quotient of zero is obtained.
  - The first remainder is the LSB and the last is the MSB

Decimal-to-Binary Conversion

Convert the following decimal numbers to binary:

(a) 19 **(b)** 45

Solution

(a)

	Remainder
$\frac{19}{2} = 9$	1
$\frac{9}{2} = 4$	1
$\frac{4}{2} = 2$	0
$\frac{2}{2} = 1$	0
$\frac{1}{2} = 0$	1

1 0 0 1 1

MSB LSB

(b)

	Remainder
$\frac{45}{2} = 22$	1
$\frac{22}{2} = 11$	0
$\frac{11}{2} = 5$	1
$\frac{5}{2} = 2$	1
$\frac{2}{2} = 1$	0
$\frac{1}{2} = 0$	1

1 0 1 1 0 1

MSB LSB

Related Problem Convert decimal number 39 to binary.

Decimal fractions to binary

- Using Sum-of-weights

Binary weights

64	32	16	8	4	2	1	.5	.25	.125	.0625
2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}

95.6875 = 64 + 16 + 8 + 4 + 2 + 1 + .5 + .125 + .0625

1011111.1011_2

Decimal fractions to binary

- Using repeated *multiplication* method
- When converting a decimal fractional number to its binary, the decimal fractional part will be multiplied by 2 till the fractional part gets 0 or till the required number of decimal places accuracy is reached.

- Example of Convert Decimal 0.825_{10} to its binary equivalent.

Steps:

1. 0.825 will be multiplied by 2 ($0.825 \times 2 = 1.650$)
2. The integer part will be the MSB in the binary result
3. The fractional part of the earlier result will be multiplied again. ($0.650 \times 2 = 1.200$)
4. Each time after the multiplication the integer part of the result will be written as the binary number.
5. The procedure should continue till the fractional part gets 0 or until the desired accuracy is reached.

Decimal fractions to binary

- Equivalent binary number for the decimal fraction 0.825 = $(0.11011)_2$ (for 5 bit accuracy)

$$
\begin{array}{r}
0.825 \\
\times 2 \\
\hline
\color{red}{1}.650 \\
\times 2 \\
\hline
\color{red}{1}.300 \\
\times 2 \\
\hline
\color{red}{0}.600 \\
\times 2 \\
\hline
\color{red}{1}.200 \\
\times 2 \\
\hline
\color{red}{1}.400 \\
\times 2 \\
\hline
\text{Etc…}
\end{array}
$$

Binary Addition

- Rules

Augend	Addend	Sum	Carryout
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Example: Perform the binary additions of 1101 0110 and 111 1011

  Solution:

$$\begin{array}{rr} & 1101\ 0110 \\ + & 0111\ 1011 \\ \hline & 1\ 0101\ 0001 \\ \hline \end{array}$$

Binary Subtraction

- Rules

Minuend	Subtrahend	Difference	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

- Example: Perform binary subtraction: $1100_2 – 0101_2$

Solution

$$\begin{array}{r} 1100_2 \\ -\,0101_2 \\ \hline 111_2 \\ \hline \end{array}$$

Binary Multiplication

- Rules

Multiplicand	Multiplier	Product
0	0	0
0	1	0
1	0	0
1	1	1

- Example: multiply 100110_2 with 101_2

$$
\begin{array}{r}
100110_2 \\
\times \quad 101_2 \\
\hline
100110 \\
000000 \\
100110 \\
\hline
10111110_2
\end{array}
$$

Binary Division

- Rules
  Dividend=Top Number, Divisor=Bottom Number, Quotient and Remainder = Result
- You cannot divide by 0 or 1!!

Perform the following binary divisions:

(a) $110 \div 11$ **(b)** $110 \div 10$

Solution

(a)
$$\begin{array}{r} \mathbf{10} \\ 11\overline{)110} \\ \underline{11} \\ 000 \end{array} \qquad \begin{array}{r} 2 \\ 3\overline{)6} \\ \underline{6} \\ 0 \end{array}$$

(b)
$$\begin{array}{r} \mathbf{11} \\ 10\overline{)110} \\ \underline{10} \\ 10 \\ \underline{10} \\ 00 \end{array} \qquad \begin{array}{r} 3 \\ 2\overline{)6} \\ \underline{6} \\ 0 \end{array}$$

Related Problem Divide 1100 by 100.

Octal Number System

- The octal numbering system uses the symbols 0, 1, 2, 3, 4, 5, 6 and 7
- 8 different digits – Radix / Base is 8
- Octals *may* be identified with a starting number zero OR subscript letter 'O' instead of the subscript base 8
- Similarly to decimal and binary, the octal numbering is a weighted system as follows

$$8^{n-1}\ 8^{n-2} \ldots\ldots 8^{3}\ 8^{2}\ 8^{1}\ 8^{0}.\ 8^{-1}\ 8^{-2}\ 8^{-3}\ \ldots\ldots\ldots\ 8^{-(n+1)}$$

Octal to Decimal Conversion

- Similar to binary to decimal conversion, multiply the weight of each position with the octal number and add together.
- Example

Convert the octal number $(23754)_8$ into its decimal equivalent.

$$(23754)_8 = 2 \times 8^4 + 3 \times 8^3 + 7 \times 8^2 + 5 \times 8^1 + 4 \times 8^0$$
$$= 10{,}220_{10}$$

Decimal to Octal Conversion

- Divide the decimal number repeatedly by 8 until the remainder becomes zero
- The first remainder is the LSD and the last is the MSD
- Example: Convert 12,34510 to its equivalent octal.

8	12345	
8	1543	remainder 1 (LSD)
8	192	remainder 7
8	24	remainder 0
8	3	remainder 0
	0	remainder 3 (MSD)

$$12{,}345_{10} = 30071_8$$

Octal to binary conversion

- Convert each octal digit to its equivalent 3 digit binary number

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

- Example: Convert 345621_8 to its binary equivalent

3	4	5	6	2	1
011	100	101	110	010	001

Result:

11100101110010001_2

Binary to Octal Conversion

- Convert from binary to octal by grouping bits in threes starting with the LSB
- Each group is then converted to the octal equivalent

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

- Leading zeros can be added to the left of the MSB to fill out the last group.
- Example:

  Convert 110001100101001_2 to octal equivalent

  110 001 100 101 001

  6 1 4 5 1

  Result = 61451_8

Octal addition and subtraction

- The largest single digit in octal is 7. If the result of A + B is greater than 7, then must subtract 8 (the base) and carry 1 to the next digit.
- Example: Perform the following octal addition → $7456_8 + 537_8$

$$\begin{array}{r} 7456_8 \\ +\ 0537_8 \\ \hline 10215_8 \\ \hline \end{array}$$

- For subtraction, if the subtrahend is bigger then the minuend we borrow a 1 from the digit on left and its weight will be 8
- ple: Perform the following octal subtraction → $3533_8 - 175_8$

$$\begin{array}{r} 3533_8 \\ -0175_8 \\ \hline 3336_8 \\ \hline \end{array}$$

Hexadecimal numbering system

- The hexadecimal numbering system uses the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F
- 16 different symbols – Radix/ Base of 16
- Hexadecimals can be written with the 16 subscript or more commonly with the prefix 0x or suffix h (e.g. 0x123 and 123h are equivalent)
- It is a weighted system as follows

$$16^{n-1}\ 16^{n-2} \ldots\ldots 16^{3}\ 16^{2}\ 16^{1}\ 16^{0}.\ 16^{-1}\ 16^{-2}\ 16^{-3} \ldots 16^{-(n+1)}$$

Hexadecimal to Decimal Conversion

- Hexadecimal number can be converted to decimal by multiplying the weight of each position of the hexadecimal number (power of 16) and adding together.
- Example: Convert $23ABC_{16}$ into its decimal equivalent.

$23ABC_{16}$

$= 2\times16^4 + 3\times16^3 + 10\times16^2 + 11\times16^1 + 12\times16^0$

$= 146108_{10}$

Hexadecimal to Decimal Conversion

- Repeated division of a decimal number by 16 will produce the equivalent hexadecimal number, formed by the remainder of the divisions
- The first remainder produced is the LSD
- Each successive division by 16 yields a remainder that becomes a digit in the equivalent hexadecimal number.
- Example: Perform the conversion of a decimal 234567 to hexadecimal.

16	234567	
16	14660	remainder 7 (LSD)
16	916	remainder 4
16	57	remainder 4
16	3	remainder 9
	0	remainder 3 (MSD)

$$234567_{10} = 39447_{16}$$

Hexadecimal to Binary Conversion

- To convert hex to binary, change each hex digit to the equivalent four bit binary number

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

- Example: Convert $A12BF_{16}$ into its equivalent binary value

A 1 2 B F

1010 0001 0010 1011 1111

$A12BF_{16} = 10100001001010111111_2$

Binary to Hexadecimal conversion

- Procedure
  - Group bits in four starting with the LSB
  - Leading zeros can be added to the left of the MSB to fill out the last group
  - Each group is then converted to the hex equivalent
- Example:
  Convert 1110100110_2 into its equivalent hexadecimal number

1110100110_2 = 0011 1010 0110

(Note the addition of leading zeroes)

= 3 A 6

= $3A6_{16}$

Hex addition and subtraction

- In any given column of an addition problem, think of the two hexadecimals in terms of their decimal values.
  - If the sum of these two digits is 15_{10} or less, the corresponding hexadecimal digit is written
  - If the sum is greater than 15_{10} subtract 16 (base) from the sum and carry a 1 to the next column.
- For subtraction, if the subtrahend is bigger then the minuend, we borrow a 1 from the digit on left and its weight will be 16.
- Example:

Addition of $AB6_{16}$ + $C5_{16}$

$$\begin{array}{r} A\ B\ 6 \\ +\ 0\ C\ 5 \\ \hline B\ 7\ B_{16} \\ \hline \end{array}$$

Subtraction of 4AB.C5H - 3C.DH

$$\begin{array}{r} 4AB.C5H \\ -03C.D0H \\ \hline 46E.F5H \\ \hline \end{array}$$

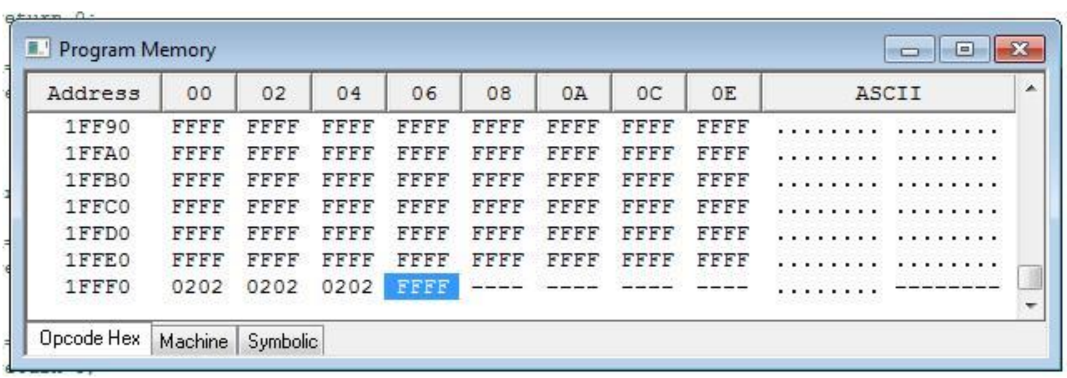
- *Optional* → Now how about multiplication and division for octals and hexadecimals?

Decimal, binary, octal and hex equivalents

Decimal	Binary	Octal	Hexadeicmal
0	0000	00	0
1	0001	01	1
2	0010	02	2
3	0011	03	3
4	0100	04	4
5	0101	05	5
6	0110	06	6
7	0111	07	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

Usefulness of Octal and Hex Numbers

- Hex and octal are often used in digital system as a “shorthand” way to represent strings of bits → it is more convenient and less error–prone to write the binary numbers in hex or octal.
  - E.g.: to print out the contents of 50 memory locations, each of which was a 16-bit numbers like $1111\ 1111\ 1111\ 1111_2$, it is easier to use hexadecimal notation like this $FFFF_{16}$
- Many systems in computing make use of hex numbers
  - memory addresses use hex addressing to indicate locations in memory banks
  - network addressing will one day migrate to IPv6 which uses hex notation to represent network device addresses

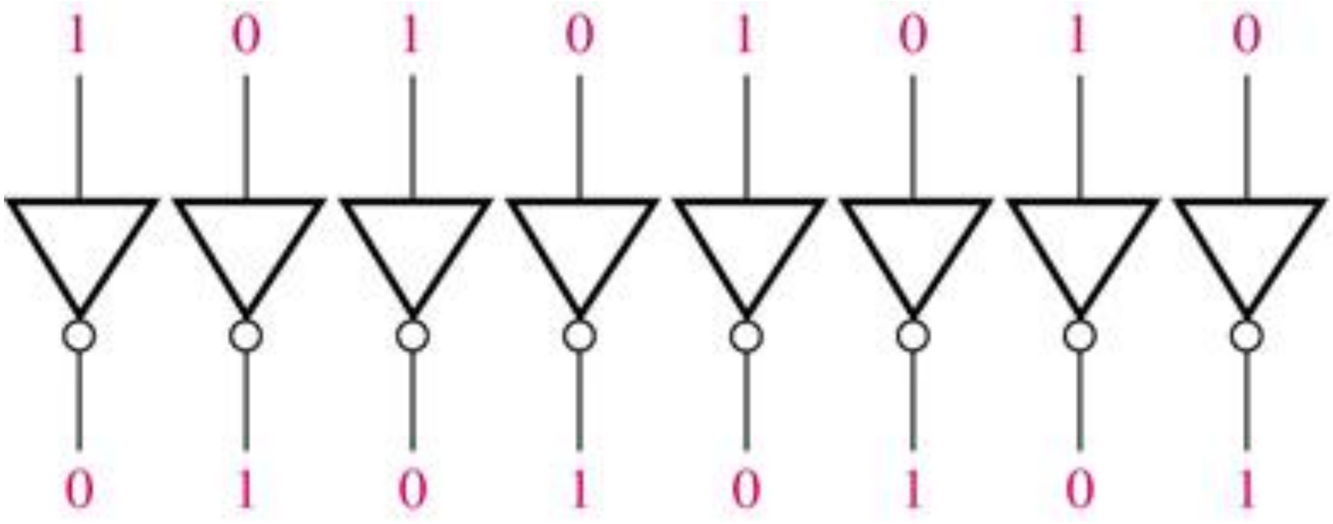


One's Complement Numbers

- The one's complement of a binary number is defined as the value obtained by inverting all the bits in the binary representation of the number (swapping 0s for 1s and vice-versa).
- The one's complement of the number then behaves like the negative of the original number in some arithmetic operations.

Binary number	0001100100101
1's complement	1110011011010

- Example of inverters used to obtain the 1's



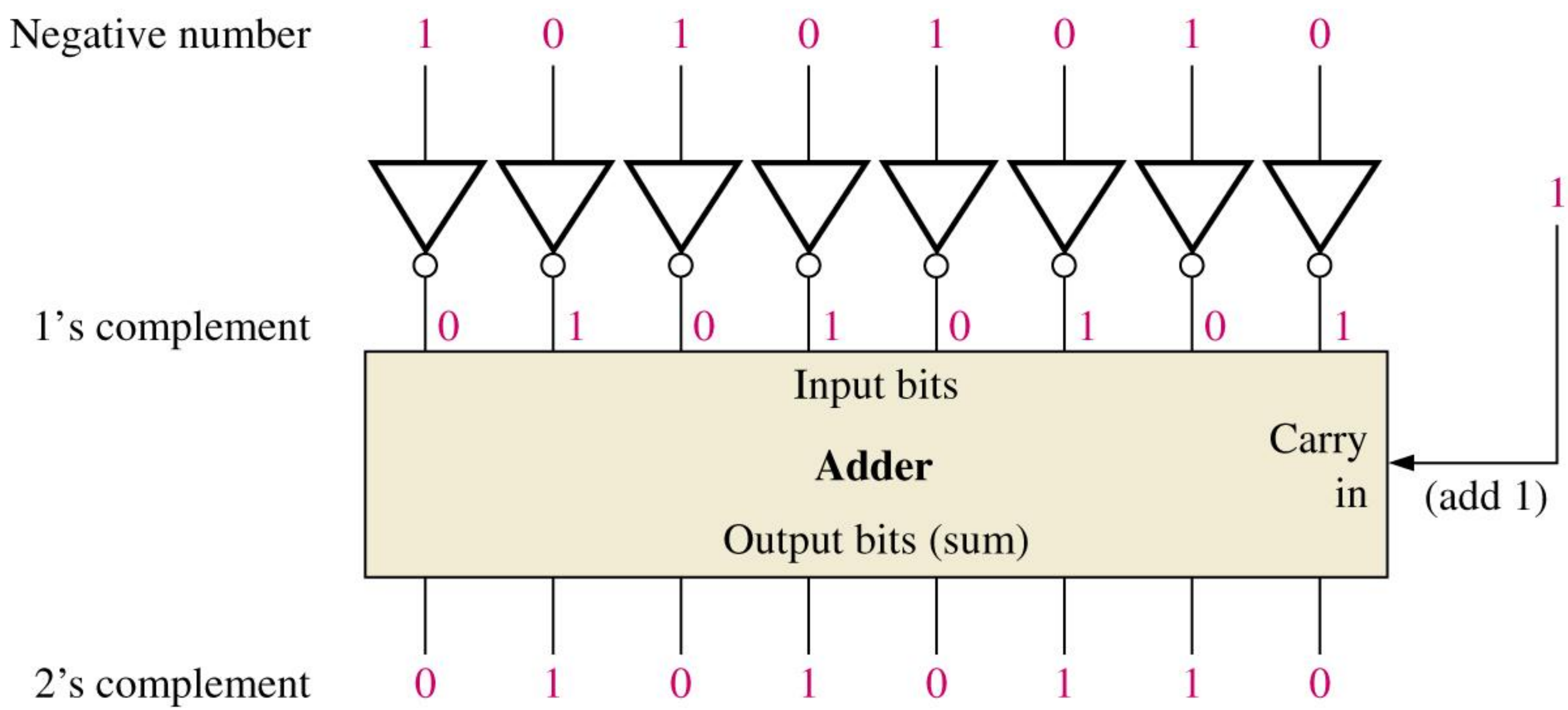
Finding 2's complement

2's complement = 1's complement + 1

Binary number	111000101010001_2
1's complement	000111010101110_2
Add	1_2
2's complement	000111010101111_2

Implementation of twos comp

- Example of obtaining the 2's complement of a negative binary number.



Why two’s complement?

- 2’s complement was devised as an easy way to represent negative binary numbers without the need for an additional bit to represent a number’s sign
  - If a bit was assigned to indicate a number is +ve/-ve then zero would have double assignment (called the signed zero problem)
- Mathematical operations with 2’s comp numbers can be simplified thus reducing the work required to manipulate numbers

Representation of Signed Integers in Binary – Three forms

Sign bit – 0 is for positive and 1 is for negative

Sign-Magnitude form – a negative number has the same magnitude as the corresponding positive number but the sign bit is a 1 rather than a zero.

Example: **+43** in 8-bit S-M form = **0 0101011**

-43 in 8-bit S-M form = **1 0101011**

1's complement form – a negative number is the 1's complement of the corresponding positive number.

Example: **+43** in 8-bit 1's complement form = **00101011**

-43 in 8-bit 1's complement form = **11010100**

2's complement form – a negative number is the 2's complement of the corresponding positive number

Example: **+43** in 8-bit 2's complement form = **00101011**

-43 in 8-bit 2's complement form = **11010101**

Determination of decimal value of signed numbers (1) ...

Sign-magnitude form

- Convert all but the MSB to decimal and call the decimal value negative if the MSB is 1

Example: 01010011 = 64+16+2+1 = +83 since MSB is 0

11010011 = 64+16+2+1 = -83 since MSB is 1

1's complement form

- If the big sign (MSB) is 0, sum the weighted values of the binary number
- If the bit sign (MSB) is 1, sum the weighted values of the binary number, assigning the MSB a negative weight and add 1.

Example: 01010011 = 64+16+2+1 = +83 since MSB is 0

10101100 = -128+32+8+4 = -84 + 1 since MSB is 1

Determination of decimal value of signed numbers (2)

2's complement form

- If the bit sign (MSB) is 0, sum the weighted values of the binary number.
- If the bit sign (MSB) is 1, sum the weighted values of the binary number assigning MSB, a negative weight.
- Example:

  01010011 = 64+16+2+1 = +83 since MSB is 0

  10101101 = -128+32+8+4+1 = -83 since MSB is 1

Arithmetic operations with 2's complement numbers - Addition

- Add the two numbers and discard any final carry bit.
- Example:

Add (34) + (-46) in 8-bit 2's complement arithmetic

34 =	00100010
-46 =	11010010

-12=	11110100

Arithmetic operations with 2's complement numbers – Subtraction (1)

A – B = A + (-B)

Add the two numbers and discard any final carry bit.

- Example: (34 -46) in 8-bit 2's complement arithmetic

Solution: 34 – 46 = 34 + (-46)

```
 34 =  00100010
-46 =  11010010
      ---------------
-12 =  11110100
      ---------------
```

Arithmetic operations with 2's complement numbers – Subtraction (2)

A – (-B) = A + B)

Add the two numbers and discard any final carry bit.

- Example: (34 – (-46)) in 8-bit 2's complement arithmetic

Solution 34 – (-46) = 34 + 46

$$\begin{array}{r} 34 = 00100010_2 \\ \underline{+46 = 00101110_2} \\ 80 = 01010000_2 \end{array}$$

Problem

- Perform the following conversions:

$$(101101.101)_2 = (?)_{10} = (?)_H = (?)_8$$

- Solution:

Problem (2)

- Suppose your microcomputer memory address is 16 bits wide, how many locations that it can have, assuming each memory location is 8 bits wide. Express the memory capacity in bytes, megabytes, gigabytes and terabytes.
- Solution:

Summary

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