Chapter 1
Binary Systems

Digital Systems

Binary Numbers

Number Base Conversion

Octal and Hexadecimal Number

Complements

Signed Binary Numbers

Binary Codes

Binary Storage and Registers

Binary Logic

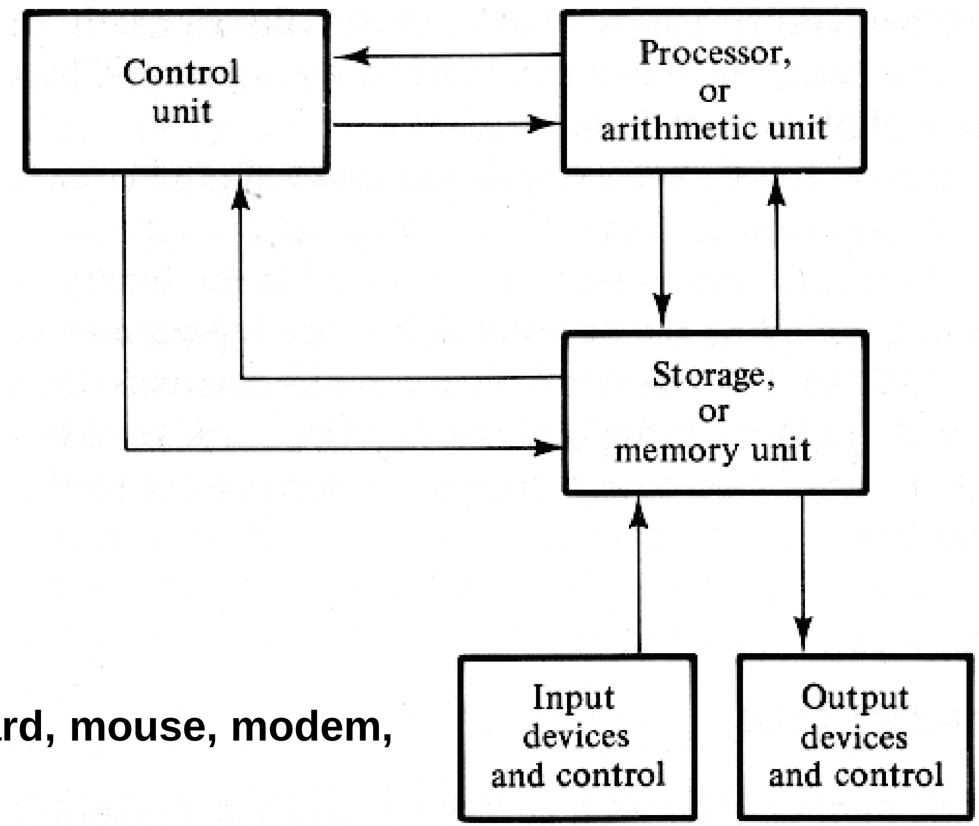
Binary Systems

- Digital Age
- Digital computers
  - Many scientific, industrial and commercial applications
  - Space program
- Digital systems
  - Telephone switching exchanges
  - Digital Camera, Mobile Phone
  - Electronic Calculators, PDA's, Tabs
  - Digital TV
- Discrete information-processing systems
- Why binary?
  - Reliability: A transistor circuit is either on or off (two stable states)

Digital Computer

- **The** digital computer **is one of the most well known digital systems.**
- **The digital computer consists of the following components:**
  - **Memory unit**
  - **Central processing unit**
  - **Input and output units**
- **The digital computer can perform both arithmetic and logical operations.**

A digital computer



Inputs: **Keyboard, mouse, modem, microphone**

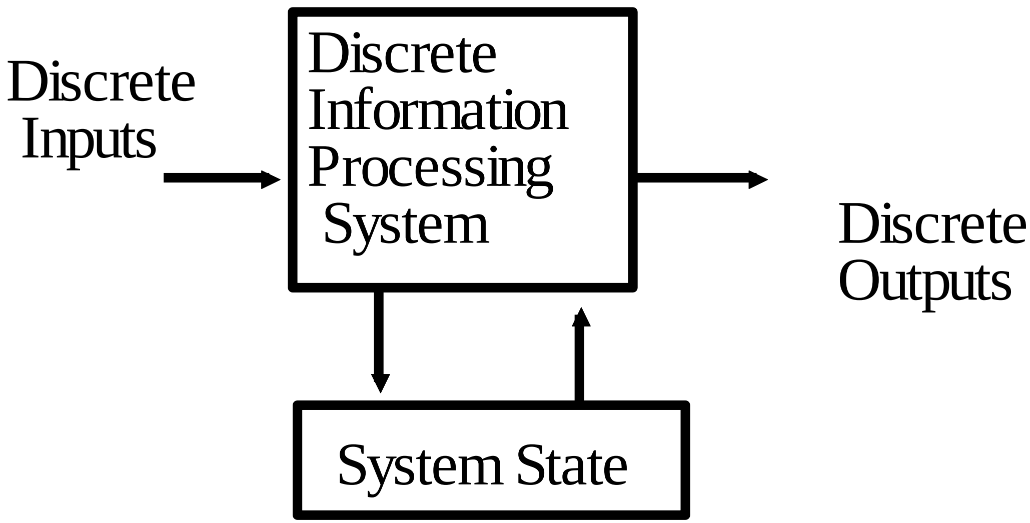
Outputs: **CRT, LCD, modem, speakers**

- **stored program**
- **control unit**
- **arithmetic computations and logical operations**

Digital Systems

- Digital Systems **represent systems that understand, represent and manipulate discrete elements.**
  - **A** discrete element **is any set that has a finite number of elements, for example 10 decimal digits, 26 letters of the alphabet, etc.**
- **Discrete elements are represented by** signals**, such as electrical signals (voltages and currents)**
- **The signals in most electronic digital systems use two discrete values, termed** binary**.**
- Digital System **takes a set of discrete information inputs and discrete internal information (system state) and generates a set of discrete information outputs.**

Digital System



Signals

- A collection of information variables mapped to some physical quantity
- For digital systems, the quantities take on discrete values
- Two level, or binary values are the most prevalent values in digital systems
- The binary values are represented abstractly by digits 0 and 1
- Other physical signals represented by 1 and 0?
  - CPU Voltage
  - Disk Magnetic Field Direction
  - CD Surface Pits/Light
  - Dynamic RAM Electrical Charge

Why Digital Components?

- **Why do we choose to use digital components?**
  - **The main reason for using digital components is that they can easily be programmed, allowing a single hardware unit to be used for many different purposes**
  - **Advances in circuit technology decrease the price of technology dramatically**
  - **Digital integrated circuits can perform at speeds of hundreds of millions of operations per second**
  - **Error-checking and correction can be used to ensure the reliability of the machine**

Binary Digits

- **A binary digit, called a bit, is represented by one of two values: 0 or 1.**
  - **Discrete elements can be represented by groups of bits called binary codes. For example the decimal digits 0 to 9 are represented as follows:**

Decimal	Binary Code
0	**0000**
1	**0001**
2	**0010**
3	**0011**
4	**0100**
5	**0101**
6	**0110**
7	**0111**
8	**1000**
9	**1001**

Differing Bases

- **In order to represent numbers of different bases, we surround a number in parenthesis and then place a subscript with the base of the number**
  - **A decimal number $(9233)_{10}$**
  - **A binary number $(11011)_2$**
  - **A base 5 number $(3024)_5$**
- **Decimal number digits are 0 through 9**
- **Binary number digits are 0 through 1**
- **Base (radix) *r* number digits are 0 through *r* - 1**

Commonly Occurring Bases

Name	Radix	Digits (0 through r-1)
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

Decimal Numbers

- **A decimal number such as 5723 represents a quantity equal to:**
  - **5 thousands**
  - **7 hundreds**
  - **2 tens**
  - **3 ones**

- **Or, it can be written as:**
  - $5 \times 10^3 + 7 \times 10^2 + 2 * 10^1 + 3 * 10^0$
- **The 5, 7, 2, and 3 represent** coefficients**.**
- **The decimal number system is said to be of base or radix 10 because it uses the 10 digits (0..9) and the coefficients are multiplied by powers of 10.**

Binary Numbers

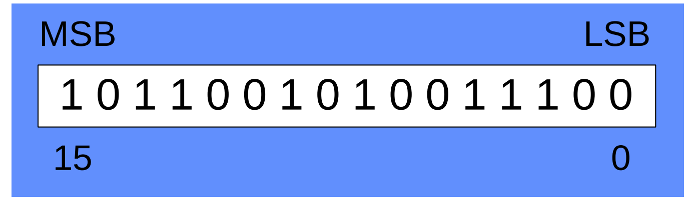
- The binary system contains only two values in the allowed coefficients (0 and 1).
- The binary system uses powers of 2 as the multipliers for the coefficients.
- For example, we can represent the binary number 10111.01 as:
  - $1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} = 23.25$

Understanding Binary Numbers

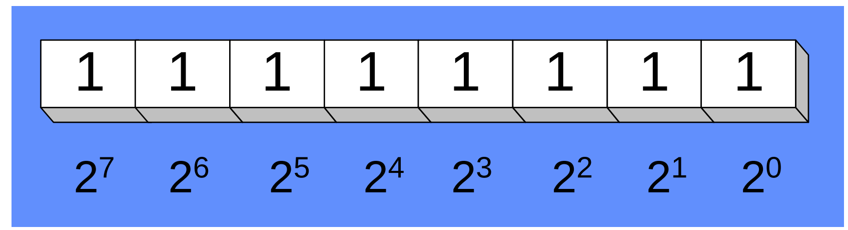
- **Binary numbers are made of binary digits (bits):**
  - **0 and 1**
- **How many items does a binary number represent?**
  - $(1011)_2 = 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = (11)_{10}$
- **What about fractions?**
  - $(110.10)_2 = 1x2^2 + 1x2^1 + 0x2^0 + 1x2^{-1} + 0x2^{-2}$
- **Group of eight bits is called a *byte***
  - $(11001001)_2$
- **Group of four bits is called a *nibble***
  - $(1101)_2$
- **Group of sixteen bits is called a *half word***
  - $(1011011010011001)_2$
- **Group of thirty two bits is called a *word***
- **Group of sixty four bits is called a *double word***

Understanding Binary Numbers (Contd…..)

- **MSB – most significant bit**
- **LSB – least significant bit**
- **Bit numbering**



- **Each digit (bit) is either 1 or 0**
- **Each bit represents a power of 2**



Why Use Binary Numbers?

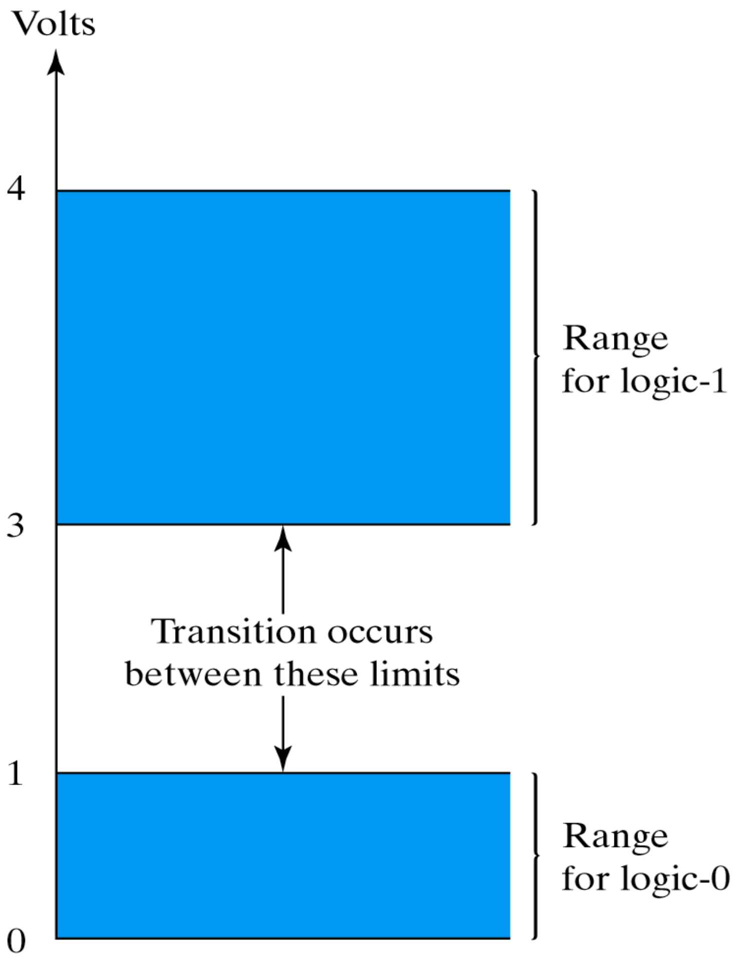
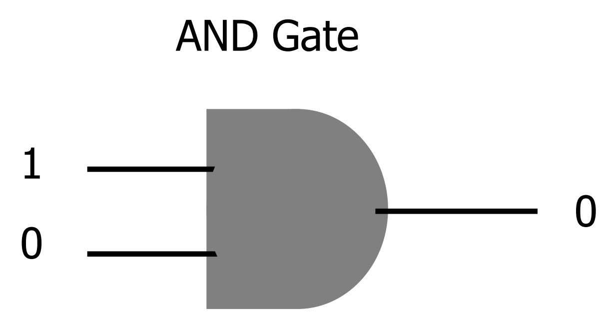


Fig. 1-3 Example of binary signals

- **Easy to represent 0 and 1 using electrical values.**
- **Possible to tolerate noise.**
- **Easy to transmit data**
- **Easy to build binary circuits.**



Powers of Two

n	2^n	n	2^n	n	2^n
0	1	8	256	16	65,536
1	2	9	512	17	131,072
2	4	10	1,024	18	262,144
3	8	11	2,048	19	524,288
4	16	12	4,096	20	1,048,576
5	32	13	8,192	21	2,097,152
6	64	14	16,384	22	4,194,304
7	128	15	32,768	23	8,388,608

n	2^n	n	2^n	n	2^n	n	2^n
0	1	16	65,536	32	4,294,967,296	48	281,474,976,710,656
1	2	17	131,072	33	8,589,934,592	49	562,949,953,421,312
2	4	18	262,144	34	17,179,869,184	**50**	**1,125,899,906,842,620**
3	8	19	524,288	35	34,359,738,368	51	2,251,799,813,685,250
4	16	**20**	**1,048,576**	36	68,719,476,736	52	4,503,599,627,370,500
5	32	21	2,097,152	37	137,438,953,472	53	9,007,199,254,740,990
6	64	22	4,194,304	38	274,877,906,944	54	18,014,398,509,482,000
7	128	23	8,388,608	39	549,755,813,888	55	36,028,797,018,964,000
8	256	24	16,777,216	**40**	**1,099,511,627,776**	56	72,057,594,037,927,900
9	512	25	33,554,432	41	2,199,023,255,552	57	144,115,188,075,856,000
10	**1,024**	26	67,108,864	42	4,398,046,511,104	58	288,230,376,151,712,000
11	2,048	27	134,217,728	43	8,796,093,022,208	59	576,460,752,303,423,000
12	4,096	28	268,435,456	44	17,592,186,044,416	**60**	**1,152,921,504,606,850,000**
13	8,192	29	536,870,912	45	35,184,372,088,832	61	2,305,843,009,213,690,000
14	16,384	**30**	**1,073,741,824**	46	70,368,744,177,664	62	4,611,686,018,427,390,000
15	32,768	31	2,147,483,648	47	140,737,488,355,328	63	9,223,372,036,854,780,000

n	**2^n**	**n**	**2^n**	**n**	**2^n**	**n**	**2^n**
0	1	16	65,536	32	4,294,967,296	48	281,474,976,710,656
1	2	17	131,072	33	8,589,934,592	49	562,949,953,421,312
2	4	18	262,144	34	17,179,869,184	**50**	**1,125,899,906,842,620**
3	8	19	524,288	35	34,359,738,368	51	2,251,799,813,685,250
4	16	**20**	**1,048,576**	36	68,719,476,736	52	4,503,599,627,370,500
5							
6							
7							
8							
9							
10							
11							
12	4,096	28	268,435,456	44	17,592,186,044,416	**60**	**1,152,921,504,606,850,000**
13	8,192	29	536,870,912	45	35,184,372,088,832	61	2,305,843,009,213,690,000
14	16,384	**30**	**1,073,741,824**	46	70,368,744,177,664	62	4,611,686,018,427,390,000
15	32,768	31	2,147,483,648	47	140,737,488,355,328	63	9,223,372,036,854,780,000

32 bits wide word can store an unsigned magnitude
4,294,967,295 = (4 GB -1)
64 bits wide word can store an unsigned magnitude
18,446,744,073,709,600,000 = (18 PB -1)

Important powers of 2

2^{10} is referred to as Kilo, called "K"

2^{20} is referred to as Mega, called "M"

2^{30} is referred to as Giga, called "G"

2^{40} is referred to as Tera, called “T”

2^{50} is referred to as Peta, called “P”

2^{60} is referred to as Exha, called “E”

Octal Numbers

- **The octal number system is a** base-8 **system that contains the coefficient values of** 0 **to** 7**.**
- **The octal system uses** powers of 8 **as the multipliers for the coefficients.**
- **For example, we can represent the octal number 72032 as:**

$$7 \text{ X } 8^4 + 2 \text{ X } 8^3 + 0 \text{ X } 8^2 + 3 \text{ X } 8^1 + 2 \text{ X } 8^0 = (29722)_{10}$$

Hexadecimal Numbers

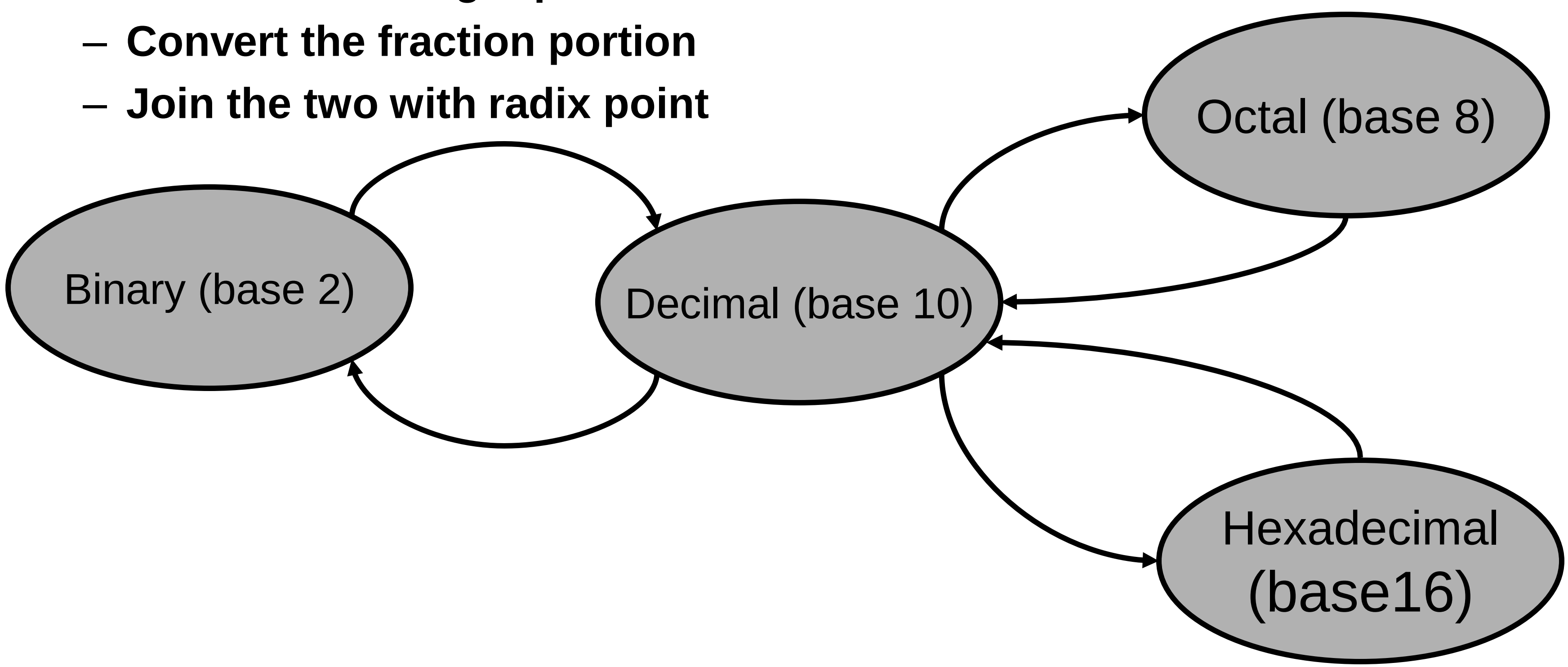
- **The hexadecimal number system is a** base-16 **system that contains the coefficient values of** 0 **to** 9 **and** A **to** F**. The letters A through F represent the coefficient values of 10, 11, 12, 13, 14, and 15, respectively.**
- **The hexadecimal system uses** powers of 16 **as the multipliers for the coefficients.**
- **For example, we can represent the hexadecimal number C34D as:**
  - $12 \times 16^3 + 3 \times 16^2 + 4 \times 16^1 + 13 \times 16^0 = (49997)_{10}$

Number Examples

Decimal (base 10)	Binary (base 2)	Octal (base 8)	Hexadecimal (base 16)
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	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

Conversion between bases

- **To convert from one base to other:**
  - **Convert the integer portion**
  - **Convert the fraction portion**
  - **Join the two with radix point**



r-Decimal Conversion

- **Conversion of a number in base r to decimal is done by expanding the number in a power series and adding all the terms.**
- **For example, $(C34D)_{16}$ is converted to decimal:**

  $12 \times 16^3 + 3 \times 16^2 + 4 \times 16^1 + 13 \times 16^0 = (49997)_{10}$
- **$(11010.11)_2$ is converted to decimal:**

  $1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 26.75$
- **In general $(Number)_r$ =** $\left(\sum_{i=0}^{i=n-1} a_i \bullet r^i\right) + \left(\sum_{j=-m}^{j=-1} a_j \bullet r^j\right)$

  (Integer Portion) + (Fraction Portion)

Decimal-r Conversion

- **If a decimal number has a radix point, it is necessary to separate the number into an integer part and a fraction part.**
- **The conversion of a decimal integer into a number in base-r is done by dividing the number and all successive quotients by r and accumulating the remainders in reverse order of computation.**
- **For example, to convert decimal 13 to binary:**

	Integer Quotient		Remainder	Coefficient
13/2 =	6	+	1	$a_0 = 1$
6/2 =	3	+	0	$a_1 = 0$
3/2 =	1	+	1	$a_2 = 1$
1/2 =	0	+	1	$a_3 = 1$

Answer $(13)_{10} = (a_3\, a_2\, a_1\, a_0)_2 = (1101)_2$

Example

- **Convert $(37)_{10}$ to binary**

Division	Quotient	Remainder
37 / 2	18	1
18 / 2	9	0
9 / 2	4	1
4 / 2	2	0
2 / 2	1	0
1 / 2	0	1

$$(37)_{10} = 100101$$

Decimal-r Conversion (converting fraction)

- To convert the fraction portion repeatedly multiply the fraction by the radix and save the integer digits that result. The process continued until the fraction becomes 0 or the number of digits have sufficient accuracy. The new radix fraction digits are the integer digits in computed order.
- For example convert fraction $(0.6875)_{10}$ to base 2

0.6875 * 2 = 1.3750	integer = 1
0.3750 * 2 = 0.7500	integer = 0
0.7500 * 2 = 1.5000	integer = 1
0.5000 * 2 = 1.0000	integer = 1

Answer = $(0.1011)_2$

Example

- **When converting fractions, we must use multiplication rather than division. The new radix fraction digits are the integer digits in *computed order*.**

	Integer		Fraction	Coefficent
0.8432 X 2 =	1	+	0.6864	$a_{-1} = 1$
0.6864 X 2 =	1	+	0.3728	$a_{-2} = 1$
0.3728 X 2 =	0	+	0.7456	$a_{-3} = 0$
0.7456 X 2 =	1	+	0.4912	$a_{-4} = 1$
0.4912 X 2 =	0	+	0.9824	$a_{-5} = 0$
0.9824 X 2 =	1	+	0.9648	$a_{-6} = 1$
0.9648 X 2 =	1	+	0.9296	$a_{-7} = 1$

Continue until fraction becomes 0 or until sufficient accuracy.

$$(0.8432)_{10} = (0.a_{-1}a_{-2}a_{-3}a_{-4}a_{-5}a_{-6}a_{-7})_2 = (0.1101011)_2$$

Example:

- **Convert 0.8125 decimal to binary.**
  - **To convert the decimal 0.8125 to binary, we multiply by the radix 2.**
  - **$(0.1101)_2$**

$$\begin{array}{r} .8125 \\ \times \quad 2 \\ \hline 1.6250 \end{array} \qquad \begin{array}{r} .6250 \\ \times \quad 2 \\ \hline 1.2500 \end{array} \qquad \begin{array}{r} .2500 \\ \times \quad 2 \\ \hline 0.5000 \end{array} \qquad \begin{array}{r} .5000 \\ \times \quad 2 \\ \hline 1.0000 \end{array}$$

Decimal to Octal Conversion

- **In converting decimal to octal we must divide by 8.**

	Integer Quotient		Remainder	Coefficent
$35 / 8 =$	4	+	3/8	$a_0 = 3$
$4 / 8 =$	0	+	4/8	$a_1 = 4$

$$(35)_{10} = (a_1 a_0)_8 = (43)_8$$

Converting Fractions (Decimal to Octal)

- **Decimal to Octal fraction conversion takes the same approach but it multiplies by the base 8.**

	Integer		Fraction	Coefficent
0.8432 X 8 =	6	+	0.7456	$a_{-1} = 6$
0.7456 X 8 =	5	+	0.9648	$a_{-2} = 5$
0.9648 X 8 =	7	+	0.7184	$a_{-3} = 7$
0.7184 X 8 =	5	+	0.7472	$a_{-4} = 5$
0.7472 X 8 =	5	+	0.9776	$a_{-5} = 5$
0.9776 X 8 =	7	+	0.8208	$a_{-6} = 7$
0.8208 X 8 =	6	+	0.5664	$a_{-7} = 6$

Continue until fraction becomes 0 or until sufficient accuracy.

$$(0.8432)_{10} = (0.a_{-1}a_{-2}a_{-3}a_{-4}a_{-5}a_{-6}a_{-7})_8 = (0.6575576)_8$$

Converting Decimal to Hexadecimal

- The conversion of a decimal integer into hexadecimal is done by dividing the number and all successive quotients by 16 and accumulating the remainders in reverse order of computation.

Division	Quotient	Remainder
422 / 16	26	6
26 / 16	1	A
1 / 16	0	1

$$(422)_{10} = (1A6)_{16}$$

Binary, Octal and Hexadecimal

- **Conversions between binary, octal and hexadecimal have an easier conversion method.**
  - **Each octal digit represents 3 binary digits.**
  - **Each hexadecimal digit represents 4 binary digits.**

$$(11\ 010\ 101\ 111\ 111\ .\ 101\ 110\ 01)_2 = (32577.561)_8$$

3 2 5 7 7 5 6 1

$$(11\ 0101\ 0111\ 1111\ .\ 1011\ 1001)_2 = (357F.B9)_{16}$$

3 5 7 F B 9

Binary to Octal and back

- **Binary to Octal:**
  - Group the binary digits into three bit groups starting at the radix point and going both ways, padding with zeros as needed (at the ends).
  - Convert each group of three bits to an equivalent octal digit.
- **Octal to Binary:**
  - It is done by reversing the preceding procedure
  - Restate the octal as three binary digits
  - Start at the radix point and go both ways, padding with zeros as needed.

Examples

- Convert $(10110001101011.11110000011)_2$ to Octal

  = 010 110 001 101 011 . 111 100 000 110

  = 2 6 1 5 3 . 7 4 0 6

  = $(26153.7406)_8$
- Convert $(673.124)_8$ to binary

  = 110 111 011 . 001 010 100

  = $(110111011.001010100)_2$
- Convert $(11010100011011)_2$ to Octal

011	010	100	011	011
3	2	4	3	3

Binary to Hexadecimal and back

- **Binary to Hexadecimal:**
  - Group the binary digits into four bit groups starting at the radix point and going both ways, padding with zeros as needed (at the ends)
  - Convert each group of four bits to an equivalent hexadecimal digit
- **Hexadecimal to Binary:**
  - It is done by reversing the preceding procedure
  - Restate the hexadecimal as four binary digits
  - Start at the radix point and go both ways, padding with zeros as needed

Examples

- Convert $(1011000110101 1.11110010)_2$ to hexadecimal

  = 0010 1100 0110 1011 . 1111 0010

  = 2 C 6 B . F 2

  = $(2C6B.F2)_{16}$
- Convert $(306.D)_{16}$ to binary

  = 0011 0000 0110. 1101

  = $(001100000110.1101)_2$
- Convert $(11010100011011)_2$ to hexadecimal

0011	0101	0001	1011
3	5	1	B

Base-r Arithmetic

- **Arithmetic operations with numbers in base r follow the same rules as for decimal numbers.**
- **When a base other than 10 is used, one must remember to use only the r-allowable digits.**
- **The following are some examples:**

```
augend:    110011     minuend:        110101    multiplicand:    1011
addend:  +100011      subtrahend:    -100111      multiplier:   X 101
          1010110                     001110                     1011
                                                                0000
                                                               1011
                                                               110111
```

Arithmetic Rules

- **The sum of two digits are calculated as expected but the digits of the sum can only be from the r-allowable coefficients.**
- **Any carry in a sum is passed to the next significant digits to be summed.**
- **In subtraction the rules are the same but a borrow adds r (where r is the base) to the minuend digit.**

Binary Addition

Given two binary digits (X,Y), a carry in (Z) we get the following sum (S) and carry (C):

Carry in (Z) of 0:

Z	0	0	0	0
X	0	0	1	1
+ Y	+ 0	+ 1	+ 0	+ 1
C S	0 0	0 1	0 1	1 0

Carry in (Z) of 1:

Z	1	1	1	1
X	0	0	1	1
+ Y	+ 0	+ 1	+ 0	+ 1
C S	0 1	1 0	1 0	1 1

Binary Addition Examples

carry: 1

	0	0	0	0	0	1	0	0	(4)
+	0	0	0	0	0	1	1	1	(7)
	0	0	0	0	1	0	1	1	(11)
bit position:	7	6	5	4	3	2	1	0	

```
   1  1  1  1  1  1  <---------  carries
      1  1  1  1  0  1
 +       1  0  1  1  1
 ---------------------
   1  0  1  0  1  0  0
```

Binary Subtraction

- **Subtraction Table**

0 - 0 = 0
0 - 1 = 1 and borrow 1
1 - 0 = 1
1 - 1 = 0

- **The borrow process works:**
- **Decimal subtraction:**
  - $1 \times 10^{n} = 10 \times 10^{n-1}$
- **Binary subtraction:**
  - $1 \times 2^{n} = 2 \times 2^{n-1}$

Binary Subtraction Example

$$
\begin{array}{cccccccccc}
 & & 1 & & & 10 & & & & \longleftarrow \texttt{borrows} \\
 & 0 & \cancel{10} & 10 & 0 & \cancel{0} & 10 & & & \\
 & \cancel{1} & \cancel{0} & \cancel{0} & \cancel{1} & \cancel{1} & \cancel{0} & 1 & & \\
- & & & 1 & 0 & 1 & 1 & 1 & & \\
\hline
 & & 1 & 1 & 0 & 1 & 1 & 0 & &
\end{array}
$$

Binary Multiplication and Division

- **Multiplication table**

  0 x 0 = 0
  0 x 1 = 0
  1 x 0 = 0
  1 x 1 = 1

```
                1   0   1   1   1
x                   1   0   1   0
---------------------------------
                0   0   0   0   0
            1   0   1   1   1
        0   0   0   0   0
    1   0   1   1   1
---------------------------------
    1   1   1   0   0   1   1   0
```

- **Binary division is similar to decimal division**

Complements

- Complements are used to simplify subtraction operations. We do subtraction by adding.

  $A - B = A + (-B)$

- There are two types:
  - The radix complement, called the r's complement.
  - The diminished radix complement, called the (r-1)'s complement.

Diminished Radix Complement (DRC)

- **Given a number *N* in base *r* having *n* digits, the (r-1)'s complement of N is defined as:**

  $(r^n - 1) - N$

- **Decimal numbers are in base-10.**

  (r-1) = (10-1) = 9.

- **The 9's complement would be defined as:**

  $(10^n - 1) - N$

- **So, to determine the 9's complement of 52:**

  $(10^2 - 1) - 52 = 47$

- **Another example is to determine the 9's complement of 3124:**

  $(10^4 - 1) - 3124 = 6875$

Finding Diminished Radix Complement (DRC)

- **The DRC or (r-1)'s complement of decimal number is obtained by subtracting each digit from 9**
- **The (r-1)'s complement of octal or hexadecimal number is obtained by subtracting each digit from 7 or F, respectively**
- **The DRC (1's complement) of a binary number is obtained by subtracting each digit from 1. It can also be formed by changing 1's to 0's and 0's to 1's**

DRC for Binary Numbers

- For binary numbers r = 2 and (r-1) = 1. So, the 1's complement would be defined as:

  $(2^n - 1) - N$

- To determine the 1's complement of 1000101:

  $(2^7 - 1) - 1000101 = 0111010$

- To determine the 1's complement of 11110111101:

  $(2^{11} - 1) - 11110111101 = 00001000010$

- Note that 1's complement can be done by switching all 0's to 1's and 1's to 0's.

Radix Complement

- **The r's complement of an n-digit number N in base-r is defined as:**

  $r^n - N$ - for $N \neq 0$

  0 - for $N = 0$

- **We may obtain r's complement by adding 1 to (r-1)'s complement. Since $r^n - N = [(r^n - 1) - N] + 1$**
- **10's complement of 3229 is:**

  $10^4 - 3229 = 6771$

- **2's complement of 101101 is:**

  $2^6 - 101101 = 010011$

- **Note that to determine 2's complement, leave the least significant 0's and the first 1 unchanged and then switch the remaining 1's to 0' and 0's to 1's.**

2's Complement

- **Another method to find 2's complement is**
  - **Complement (reverse) each bit**
  - **Add 1**
- **Example:**

Starting value	00000001
Step 1: reverse the bits	11111110
Step 2: add 1 to the value from Step 1	11111110 +00000001
Sum: two's complement representation	11111111

Note that 00000001 + 11111111 = 00000000

Signed Binary Numbers

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