Lecture 3

Number Systems

TOPICS

- Decimal Number Systems
- Binary number Systems
- Octal
- Hexadecimal
- Number conversion

Decimal Number System

- The word decimal is a derivative of decem, which is the Latin word for ten.
- Numbers today are almost universally written in a form of notation known as positional number representation.
- A number is represented as a string of digits.

Decimal Number Systems

- In a decimal number there are ten such digits that may be used, ranging in value from zero to nine
- Decimal numbers are positional numbers that have a *base* or *radix* of ten.
- Why ten ?

Decimal Number Systems

- Exactly ten distinct numerals in order to represent all possible values for each position in the number, and hence to enable us to represent all possible integer numbers in decimal notation.
- **0, 1, 2, 3, 4, 5, 6, 7, 8,** and **9** values
- For example, a number like "0123456789" has ten positions and each position can contain the digits 0-9.

Decimal Number Systems

- Each digit position has a *weight* associated with it.
- Each digit corresponds to a power of 10 based on its position in the number
- Number’s value = a *weighted sum* of the digits

Decimal Number Systems

- Number's value = a *weighted sum* of the digits
- Number' value = digit * 10^x + digit * 10^x where x = (position number - 1).

It sounds complicated ?

Decimal Number Systems

- The powers of 10 increment from 0, 1, 2, etc. as you move right to left

Hundreds	Tens	Units

Decimal Number Systems

- The BASE is 10 or Dec or simply D
- BASE = 10

$10^3=1000$

$10^2=100$

$10^1=10$

$10^0=1$

10^{3} Power or Exponent

Base 10

Decimal Number Systems

- Example

$$1234_{10} = 1\text{x } 10^3 + 2\text{x } 10^2 + 3\text{x}10^1 + 4\text{x}10^0$$

$$= 1000 \quad + 200 \quad + 30 \quad + 4$$

$$= 1234_{10}$$

Decimal Number Systems

- Example

$$9876_D = 9x\ 10^3 + 8x\ 10^2 + 7x10^1 + 6x10^0$$

$$= 9000 \quad + 800 \quad + 70 \quad + 6$$

$$= 9876_D$$

Decimal Number Systems

- Summary
- In the decimal system, there are 10 digits (0 through 9) which combine to form numbers as follows:

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 ...

Binary Number Systems

- The term computer numbering formats refers to the schemes implemented in digital computer and calculator hardware and software to represent numbers
- Digital Computers and Calculator use a binary systems
  - Why binary?

Binary Number Systems

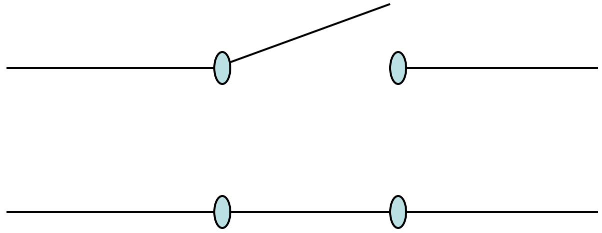
- Easy to represent binary values electrically
- Voltages and currents.
- Can be implemented using circuits
- Create the building blocks of modern computers

Binary Number Systems

- Reliability
  - With only 2 values, can be widely separated, therefore clearly differentiated
  - Binary numbers are made of binary digits

Binary Number Systems

- Binary DigiTs (BITs) can be represented electronically:



… 0 (no signal)

… 1 (signal)

Binary Number Systems

Base is 2 or ‘b’ or ‘B’ or ‘Bin’

Two symbols: 0 and 1

Each place is weighted by the power of 2

Binary Number Systems

- All the information in the digital computer is represented as bit patterns
- What is a bit pattern?

This is one bit

01010101

This is called as the bit pattern

Binary Number Systems

- Look at this bit pattern

  0101 0101

- How many bits are present ?
- Count the number of ones and the zeros in the above pattern

Answer = Total 8 bits

Binary Number Systems

0101 0101

This pattern is represented as follows in the digital computer

Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	1	0	1	0	1

Binary Number Systems

- A single bit can represent two states:0 1
- Therefore, if you take two bits, you can use them to represent four unique states:

  00, 01, 10, & 11
- And, if you have three bits, then you can use them to represent eight unique states:

  000, 001, 010, 011, 100, 101, 110, & 111

Binary Number Systems

•With every bit you add, you double the number of states you can represent. Therefore, the expression for the number of states with *n* bits is 2^n. Most computers operate on information in groups of 8 bits,

Binary Number Systems

Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	1	0	1	0	1

- There are 8 bits in the above table
- Group of 4 bits = 1 Nibble
- Group of 8 bits = 1 Byte
- Group of 16 bits = 1 Word   2 Bytes = 1 Word

Binary Number Systems

Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	1	0	1	0	1	0	1

- There are 8 bits in the above table
- Bit 0 is called the Least Significant Bit LSB
- Bit 1 is called the Most Significant Bit MSB

Binary Number Systems

Bit positions and their values

Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

Convert Decimal to Binary

1. Divide decimal number by the base (e.g. 2), remainder is 0 or 1.
2. The *remainder* is the Lowest Significant Bit (LSB).
3. Repeat first two steps until no *divisor* remains. means divide quotient by 2, marking down the remainder each time. Last remainder is Most Significant Bit (MSB).
4. Write the remainder in reverse order

Convert Decimal to Binary

Example: Convert Decimal 13 (13_{10}) to Binary :

Repeated division by 2 (till quotient is zero)

Divide-by -2	Quotient	Remainder	Binary Bits
13/2	6	1	Bit 0 = 1
6 / 2	3	0	Bit 1 = 0
3 / 2	1	1	Bit 3 = 1
1 / 2	0	1	Bit 4 = 1

Answer = 1101_2

Convert Binary to Decimal

Example: Convert 1101_2

Multiply each 1 bit by the appropriate power of 2 and add them together.

1 0 1 1

Bit 3 Bit 2 Bit 1 Bit 0

Convert Binary to Decimal

Example: Convert 1101_2

Bit7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
				1	1	0	1

Multiply with these values

8 x 1 + 4 x 1 + 2 x 0 + 1x 1 = 8 + 4 + 0 + 1

= 13

Binary Number Systems

Example:

1011_2 *or* 1011_B

$= 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

$= 8 + 0 + 2 + 1$

$= 11_{10}$

Binary Number Systems

Example:

10110_2 *or* 10110_B

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

$$= 16 + 0 + 4 + 2 + 0$$

$$= 22_{10}$$

Other Number Systems

- Octal and hex are a convenient way to represent binary numbers, as used by computers.
- Computer mechanics often need to write out binary quantities, but in practice writing out a binary number such as

Other Number Systems

- 1001001101010001

  is tedious, and prone to errors.
- Therefore, binary quantities are written in a base-8 ("octal") or, much more commonly, a base-16 ("hexadecimal" or "hex") number format.

Octal Number Systems

- Base = 8 or ‘o’ or ‘Oct’
- 8 symbols: { 0, 1, 2, 3, 4, 5, 6, 7}
- Example 123, 567, 7654 etc

  ~~987~~ This is incorrect why?
- How to represent a Decimal Number using a Octal Number System ?

Octal Number Systems

- Repeated Division by 8
- Example

$213_{10} = (\ )_8$?

Divide-by -8	Quotient	Remainder	Octal digit
213 / 8	26	5	Lower digit = 5
26 / 8	3	2	Second digit =2
3 / 8	0	3	Third digit =3

Answer = 325_8

Octal Number Systems

- How to convert 325_8 back to Decimal ?
- Use this table and multiply the digits with the position values

Digit 8	Digit 7	Digit 6	Digit 5	Digit 4	Digit 3	Digit 2	Digit 1
8^7	8^6	8^5	8^4	8^3	8^2	8^1	8^0
……	……	32768	4096	512	64	8	1

Octal Number Systems

- How to convert 325_8 back to Decimal ?
- Consider the above number

$$3\ 2\ 5_{(8)}$$

Digit 1

Digit 3

Digit 2

$3 \times 8^2 + 2 \times 8^1 + 5 \times 8^0 = 3 \times 64 + 2 \times 8 + 5 \times 1$

$= 192 + 16 + 5$

$= 213$

Octal Number Systems

- Example Convert 611_8
- Consider the above number

$611_{(8)}$ → Digit 1

Digit 3

Digit 2

$$6 \times 8^2 + 1 \times 8^1 + 1 \times 8^0 = 6 \times 64 + 1 \times 8 + 1 \times 1$$
$$= 384 + 8 + 1$$
$$= 393$$

Octal Number Systems

- Convert 393 to octal

Divide-by -8	Quotient	Remainder	Octal digit
393 / 8	49	1	Lower digit = 1
49 / 8	6	1	Second digit =1
6 / 8	0	6	Third digit =6

Answer = 611_8

Hexadecimal Number Systems

- Base = 16 or ‘H’ or ‘Hex’

  16 symbols: { 0, 1, 2, 3, 4, 5, 6, 7,8,9 }

  { 10=A, 11=B, 12=C, 13=D, 14=E, 15= F}

Hexadecimal Number Systems

- {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F} It uses 6 Letters !
- Example AB12, 876F, FFFF etc
- How to represent a Decimal Number using a Hexadecimal Number System ?

Hex Number Systems

- Repeated Division by 16
- Example

$213_{10} = (\ )_{16}$?

Divide-by -16	Quotient	Remainder	Hex digit
213 / 16	13	5	Lower digit = 5
13 / 16	0	13	Second digit =D

Answer = $D5_{16}$

Hex Number Systems

- How to convert $D5_{16}$ back to Decimal ?
- Use this table and multiply the digits with the position values

Digit 8	Digit 7	Digit 6	Digit 5	Digit 4	Digit 3	Digit 2	Digit 1
16^7	16^6	16^5	16^4	16^3	16^2	16^1	16^0
……	……	…..	……	4096	256	16	1

Hex Number Systems

- How to convert $D5_{16}$ back to Decimal ?
- Consider the above number

$D\ 5_{(16)}$ → Digit 1

Digit 2

$$D \times 16^1 + 5 \times 16^0 = 13 \times 16 + 5 \times 1$$
$$= 208 + 5$$
$$= 213$$

Number System

- Each number system has a number of different digits which is called the **`radix`** or the **`base`** of the number system.
  - Decimal Base = 10
  - Binary Base = 2
  - Octal Base = 8
  - Hexadecimal (Hex) Base = 16

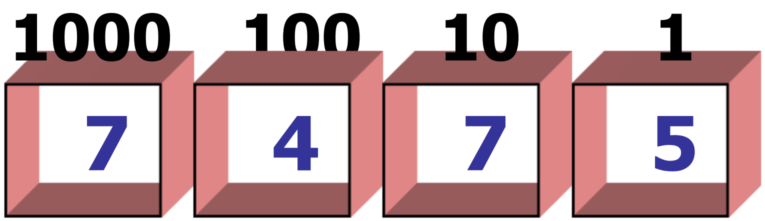
Decimal Number System

Base (Radix) **10**

Digits **0, 1, 2, 3, 4, 5, 6, 7, 8, 9**

e.g. **7475_{10}**

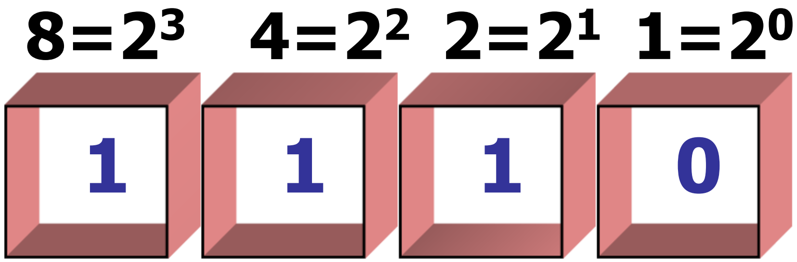
The magnitude represented by a digit is decided by the position of the digit within the number.



For example the digit 7 in the left-most position of 7475 counts for 7000 and the digit 7 in the second position from the right counts for 70.

Binary Number System

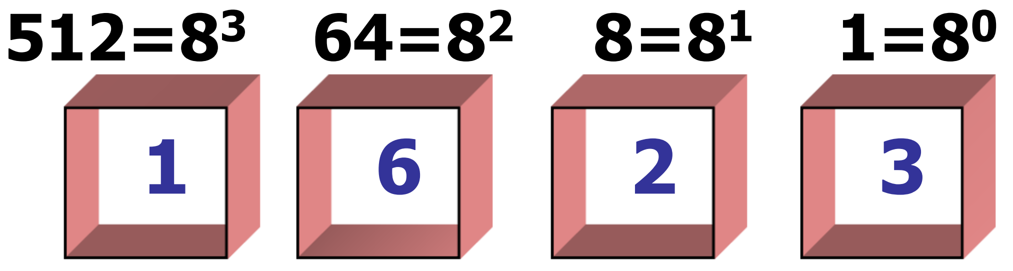
Base (Radix)	**2**
Digits	**0, 1**
e.g.	$\mathbf{1110_2}$



The digit 1 in the third position from the right represents the value 4 and the digit 1 in the fourth position from the right represents the value 8.

Octal Number System

Base (Radix)	8
Digits	0, 1, 2, 3, 4, 5, 6, 7
e.g.	1623_8

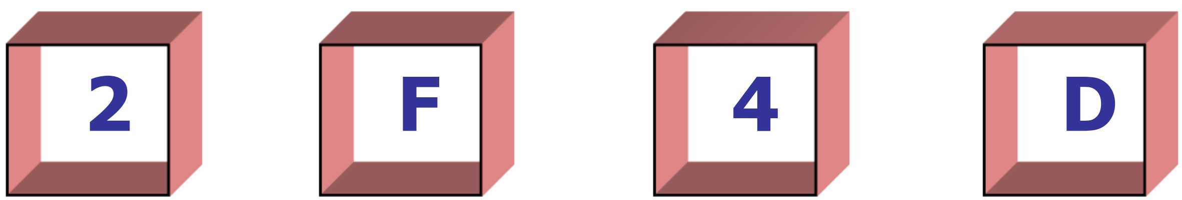


The digit 2 in the second position from the right represents the value 16 and the digit 1 in the fourth position from the right represents the value 512.

Hexadecimal Number System

Base (Radix)	16
Digits	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
e.g.	$2F4D_{16}$

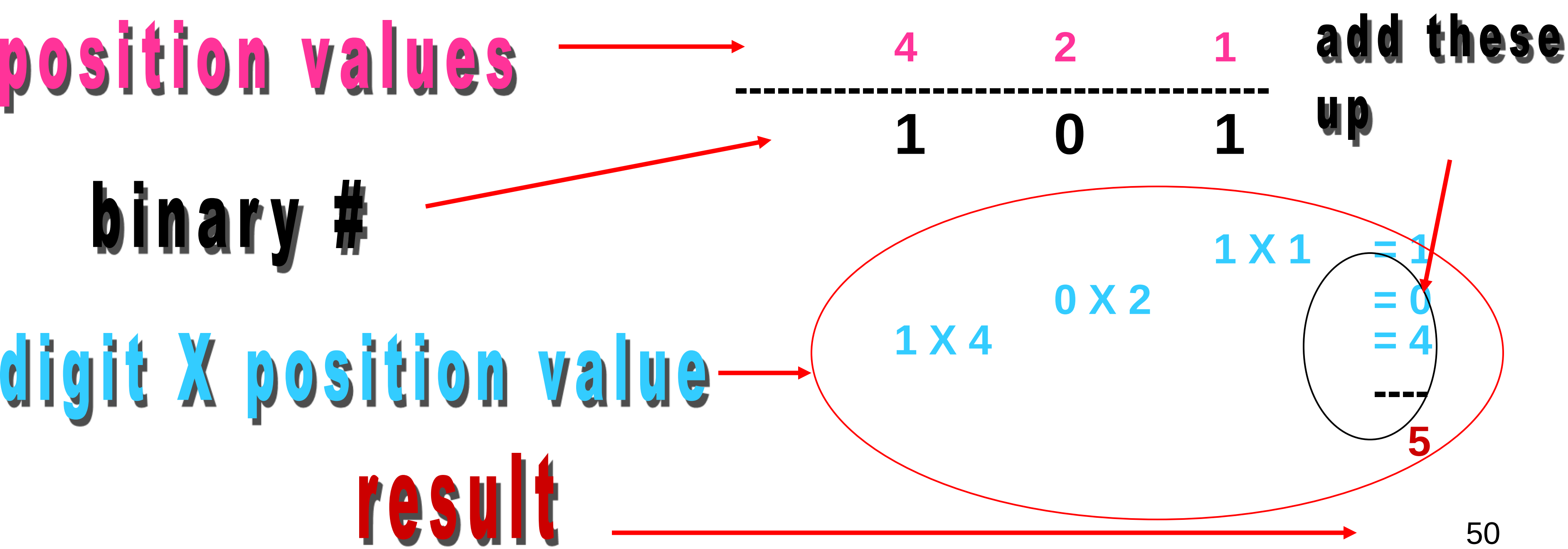
$4096=16^3$ $256=16^2$ $16=16^1$ $1=16^0$



The digit F in the third position from the right represents the value 3840 and the digit D in the first position from the right represents the value 1.

Converting from binary to decimal

- Each position for a binary number has a value.
- For each digit, multiply the digit by its position value
- Add up all of the products to get the final result
- The decimal value of binary 101 is computed below:



What about a longer number?

- In general, the "position values" in a binary number are the powers of two.
  - The first position value is 2^0 , i.e. one
  - The 2nd position value is 2^1 , i.e. two
  - The 2nd position value is 2^2 , i.e. four
  - The 2nd position value is 2^3 , i.e. eight
  - The 2nd position value is 2^4 , i.e. sixteen
  - etc.
- Example on next slide

Example

- **The value of binary 01100001 is decimal 105. This is worked out below:**

128	64	32	16	8	4	2	1	
0	**1**	**1**	**0**	**1**	**0**	**0**	**1**	
							1 X 1	= 1
						0 X 2		= 0
					0 X 4			= 0
				1 X 8				= 8
			0 X 16					= 0
		1 X 32						= 32
	1 X 64							= 64
0 X 128								= 0

							Answer:	

105

Another example

- **The value of binary 10011100 is decimal 156. This is worked out below:**

128	64	32	16	8	4	2	1	
1	**0**	**0**	**1**	**1**	**1**	**0**	**0**	
							0 X 1	= 0
						0 X 2		= 0
					1 X 4			= 4
				1 X 8				= 8
			1 X 16					= 16
		0 X 32						= 0
	0 X 64							= 0
1 X 128								= 128

							Answer:	

156

Some Terminology

- The following are some terms that are used in the computer field
  - Each digit of a binary number is called a **bit**.
  - A binary number with eight bits (i.e. digits) is called a **byte**.

How many different numbers?

- There are two different binary numbers with one bit:
  - 0
  - 1

- There are four different binary numbers with two bits:
  - 00 (i.e. decimal 0)
  - 01 (i.e. decimal 1)
  - 10 (i.e. decimal 2)
  - 11 (i.e. decimal 3)

How many different numbers?

- There are eight different binary numbers with three bits:
  - 000 (i.e. decimal 0)
  - 001 (i.e. decimal 1)
  - 010 (i.e. decimal 2)
  - 011 (i.e. decimal 3)
  - 100 (i.e. decimal 4)
  - 101 (i.e. decimal 5)
  - 110 (i.e. decimal 6)
  - 111 (i.e. decimal 7)

different numbers - General Rule

- **For n bits there are 2^n different binary numbers:**

# of bits	# of different binary numbers
1 bit:	$2^1 = 2$
2 bits:	$2^2 = 4$
3 bits:	$2^3 = 8$
4 bits:	$2^4 = 16$
5 bits:	$2^5 = 32$
6 bits:	$2^6 = 64$
7 bits:	$2^7 = 128$
8 bits:	$2^8 = 256$
9 bits:	$2^9 = 512$
10 bits:	$2^{10} = 1024$
etc.	

Smallest value for a binary

- The smallest value for a binary number of any number of bits is zero.

- This is the case when all bits are zero:

Smallest value for a binary

- **The smallest value for a binary number with any number of bits is zero (i.e. when all the bits are zeros)**

# of bits	smallest binary #	decimal value
1 bit:	0	0
2 bits:	00	0
3 bits:	000	0
4 bits:	0000	0
5 bits:	00000	0
6 bits:	000000	0
7 bits:	0000000	0
8 bits:	00000000	0
etc.		

Largest value for a binary

- The largest value for a binary number with a specific number of bits (i.e. digits) is when all of the bits are one.

- General rule: for a binary number with n bits, the largest possible value is : $2^n - 1$

Largest numbers

- The following are the largest values for binary numbers with a specific number of bits:

# of bits	largest binary #	decimal value
1 bit:	1	1
2 bits:	11	3
3 bits:	111	7
4 bits:	1111	15
5 bits:	11111	31
6 bits:	111111	63
7 bits:	1111111	127
8 bits:	11111111	255
etc.		

Converting a decimal# to a binary# (1)

- Step 1: figure out how many bits you will need (see the chart on the previous slide).

  – **Example 1: To convert the decimal number 16 to binary, you will need at least 5 bits.**
  (With 4 bits you can only store numbers up to 15 but with 5 bits you can store numbers up to 31)

  – **Example 2: To convert the decimal number 106 to binary, you will need at least 7 bits.**
  (With 6 bits you can only store numbers up to 63 but with 7 bits you can store numbers up to 127)

Converting a decimal# to a binary# (2)

- Step 2: Keep a chart of the position values and the digits for your binary number. At first you will not know what any of the digits will be.

- Example: convert decimal 106 to binary
  - step 1 : figure out that you need 7 bits (see earlier slides)
  - step 2 : keep track of position values and bits for binary #

position values

64	32	16	8	4	2	1
?	?	?	?	?	?	?

binary #

Converting a decimal # to binary (3)

- Step 3: starting with the leftmost digit, see if the position value is greater than, less than or equal to the number you are trying to convert.
  - if the position value is greater than the number then
    - make the binary digit in that position zero
  - if the position value is less than the number then
    - make the binary digit in that position one
    - subtract the position value from the decimal # you are trying to convert
  - if the position value is equal to the number then
    - make the binary digit in that position one
    - make the rest of the binary digits zero
    - you are done

Converting a decimal # to binary (4)

- Step 4: do step 3 again with the next digit. Keep doing this until you've figured out all of the digits.

Example

- Example: convert decimal 106 to binary
  - Step 1: You need 7 bits *(see earlier slides for explination)*.
  - Step 2: keep track of position values for bits

64	32	16	8	4	2	1
?	**?**	**?**	**?**	**?**	**?**	**?**

  - Step 3: Check leftmost position value (i.e. 64)
    - 64 is less than 105, therefore
    - the first binary digit is 1

64	32	16	8	4	2	1
1	**?**	**?**	**?**	**?**	**?**	**?**

- subtract : **106 - 64 = 42**

Example (continued-2)

– Step 4: Check next position value (i.e. 32)
  - 32 is less than **42**, therefore
  - the next binary digit is 1

64	32	16	8	4	2	1
	1	**1**	**?**	**?**	**?**	**?**

?

  - subtract : **42 - 32 = 10**

– Step 4(continued): Check next position value (i.e. 16)
  - 16 is greater than **10**, therefore
  - the next binary digit is 0

64	32	16	8	4	2	1
	1	**1**	**0**	**?**	**?**	**?**

?

Example (continued-3)

– Step 4(continued): Check next position value (i.e. 8)
  - 8 is less than **10**, therefore
  - the next binary digit is 1

64	32	16	8	4	2	1
	1	**1**	**0**	**1**	**?**	**?**

?

  - subtract : **10 - 8 = 2**

– Step 4(continued): Check next position value (i.e. 4)
  - 4 is greater than **2**, therefore
  - the next binary digit is 0

64	32	16	8	4	2	1
	1	**1**	**0**	**1**	**0**	**?**

?

Example (continued-4)

– Step 4(continued): Check next position value (i.e. 2)
  - 2 is equal to **2**, therefore
  - the next binary digit is 1

64	32	16	8	4	2	1
	1	**1**	**0**	**1**	**0**	**1**

?

  - Since the position value was equal to the number (i.e. 2) the rest of the binary digits are all zeros

64	32	16	8	4	2	1
1	**1**	**0**	**1**	**0**	**1**	**0**

– **Answer: 1101010**

Dec → Binary : Another Method

Example:

Convert the decimal number 26_{10} into its binary equivalent.

Solution:

$$2\overline{)26}^{\,13} \quad r = 0 \leftarrow \text{LSB}$$

$$2\overline{)13}^{\,6} \quad r = 1$$

$$2\overline{)6}^{\,3} \quad r = 0$$

$$2\overline{)3}^{\,1} \quad r = 1$$

$$2\overline{)1}^{\,0} \quad r = 1 \leftarrow \text{MSB}$$

$\therefore\ 26_{10} = 11010_2$

Contd…

Example:

Convert the decimal number 41_{10} into its binary equivalent.

Solution:

$$2\overline{\smash{)}41}^{\,20} \quad r = 1 \leftarrow \text{LSB}$$

$$2\overline{\smash{)}20}^{\,10} \quad r = 0$$

$$2\overline{\smash{)}10}^{\,5} \quad r = 0$$

$$2\overline{\smash{)}5}^{\,2} \quad r = 1$$

$$2\overline{\smash{)}2}^{\,1} \quad r = 0$$

$$2\overline{\smash{)}1}^{\,0} \quad r = 1 \leftarrow \text{MSB}$$

$$\boxed{\therefore\ 41_{10} = 101001_2}$$

Dec $\rightarrow$ Binary : More Examples

a) $13_{10} = ?$ $1\,1\,0\,1_{\,2}$

b) $22_{10} = ?$ $1\,0\,1\,1\,0_{\,2}$

c) $43_{10} = ?$ $1\,0\,1\,0\,1\,1_{\,2}$

d) $158_{10} = ?$ $1\,0\,0\,1\,1\,1\,1\,0_{\,2}$

Binary → Decimal Conversion

Example:

Convert the binary number 10010_2 into its decimal equivalent.

Solution:

1		0		0		1		0		
2^4		2^3		2^2		2^1		2^0		
16		8		4		2		1		
16	+	0	+	0	+	2	+	0	=	18_{10}

$$\therefore 10010_2 = 18_{10}$$

Binary → Dec : Example 2

Example:

Convert the binary number 0110101_2 into its decimal equivalent.

Solution:

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

$$0 + 32 + 16 + 0 + 4 + 0 + 1 = 53_{10}$$

$$\therefore 0110101_2 = 53_{10}$$

Binary → Dec : More Examples

a) $0110_2 = ?$

b) $11010_2 = ?$

c) $0110101_2 = ?$

d) $11010011_2 = ?$

Summary & Review

a) Divide the *Decimal Number* by 2; the remainder is the LSB of *Binary Number* .

b) If the Quotient Zero, the conversion is complete; else repeat step (a) using the Quotient as the Decimal Number. The new remainder is the next most significant bit of the *Binary Number.*

a) Multiply each bit of the *Binary Number* by it corresponding bit-weighting factor (i.e. Bit-0→2^0=1; Bit-1→2^1=2; Bit-2→2^2=4; etc).

b) Sum up all the products in step (a) to get the *Decimal Number*.

Why is it called "binary" (or base-2)?

- The prefix "bi" means "**two**" in Latin

- Binary derives its name from the fact that the digits in a "**Bi**nary" number can only have **two** possible values, 0 or 1

- It is also called "base-**2**" based on the fact that the column values are the powers of **2.**
(i.e. $\mathbf{2}^0$ $\mathbf{2}^1$ $\mathbf{2}^2$ $\mathbf{2}^3$ $\mathbf{2}^4$ $\mathbf{2}^5$ etc.)

Hexadecimal (AKA “Hex”) numbers

What is a hexadecimal or base-16 number.

- A “hexadecimal” number is a number where each digit may be one of sixteen possible values.
- The possible values for a hexadecimal digit are:
  0 1 2 3 4 5 6 7 8 9 A B C D E F
- A digit of
  “A” stands for the number 10
  “B” stands for the number 11
  “C” stands for the number 12
  “D” stands for the number 13
  “E” stands for the number 14
  “F” stands for the number 15

- Keep reading …

Hexadecimal numbers

- The following are all valid hexadecimal nubmers
  - A
  - 9(yes, a hexadecimal number does not HAVE TO contain letters)
  - 1001 (yes, a hexadecimal number does not HAVE TO contain letters)
  - 9C5
  - BFE
  - Etc.
- To understand what a specific hexadecimal number means, you can convert it into an equivalent decimal number. (see next slide)

Converting a Hexadecimal number to Decimal

- **The value of hexadecimal A12F is decimal 41,263. See below:**

4096 (i.e 16^3)	**256 (i.e 16^2)**	**16 (i.e 16^1)**	**1 (i.e 16^0)**	
A	**1**	**2**	**F**	
			15 X 1	**= 15**
		2 X 16		**= 32**
	1 X 256			**= 256**
10 X 4096				**= 40,960**

			Answer:	
41,263				

Hex numbers are a “shorthand” for binary nubmers

- It is very easy to convert between Hex and binary numbers.
- Each Hex number is 1/4th the length of its equivalent binary number.
- Therefore Hex is often used as a “shorthand” for writing an equivalent binary number.
- Keep reading …