FACULTY OF TECHNOLOGY
UNIVERSITY OF COLOMBO
SRI LANKA

IT 1020 Introduction to Computer Systems and Operating Systems

Lecture 4: Number Systems

Department of Information and Communication Technology

Outline of the syllabus

- Week 1 – Introduction and History of the computers
- Week 2 – Data Representation in Computer System
- Week 3 – Computer Architecture
- Week 4 – Number Systems
- Week 5 – Computer Arithmetic
- Week 6 – Logic & Control – Part I
- Week 7 – Logic & Control – Part II

Outline of the syllabus

- Week 8 – Mid Semester Exam
- Week 9 – SOP and POS
- Week 10 – Logic Circuits & Simplification
- Week 11 – Combinational and Sequential Circuits Part - I
- Week 12 - Combinational and Sequential Circuits Part - II
- Week 13– Introduction to Operating Systems Part - I
- Week 14 – Operating Systems Part - II

Learning Outcomes

After completing this lecture you will be able to:

- Understand the numerical system.
- Explain why computer designers chose to use the binary system for representing information in computers.
- Explain different number systems
- Translate numbers between number systems
- Appraise binary number system

Outline of the Lecture

- Number bases used with computers
- Why binary?
- Number Base Conversion
- Conversion of Fractions

Number Bases

We are used to dealing with numbers in the decimal system, where we use a **base of 10**, counting up from **0 to 9** and then resetting our number to 0 and carrying 1 into another column.

This is probably a result of having ten fingers.

The alien shown here has only eight fingers, so it would most probably work in **base 8**, counting from **0 up to 7** and then resetting to 0 and carrying 1.

Why Binary is used in Computers?

The numeric values may be represented by two different voltages that can be represented by binary;

- The original computers were designed to be high-speed calculators.
- The designers needed to use the electronic components available at the time.
- The designers realized they could use a simple coding system--the binary system to represent their numbers

Why Binary is used in Computers?

Computers work using electronic circuits which can only be switched to be **on or off**, with no shades of meaning in between.

When a key is pressed the keyboard characters and numbers have to be converted into a sequence of 1's and 0's so that the computer can open or close its electronic switches in order to process the data.

Because only two possible symbols can be used this is called a Binary system. This system works to a base of 2.

Data Representation

How is a character sent from the keyboard to the computer?

Step 1:

The user presses the letter T key on the keyboard

Step 2:

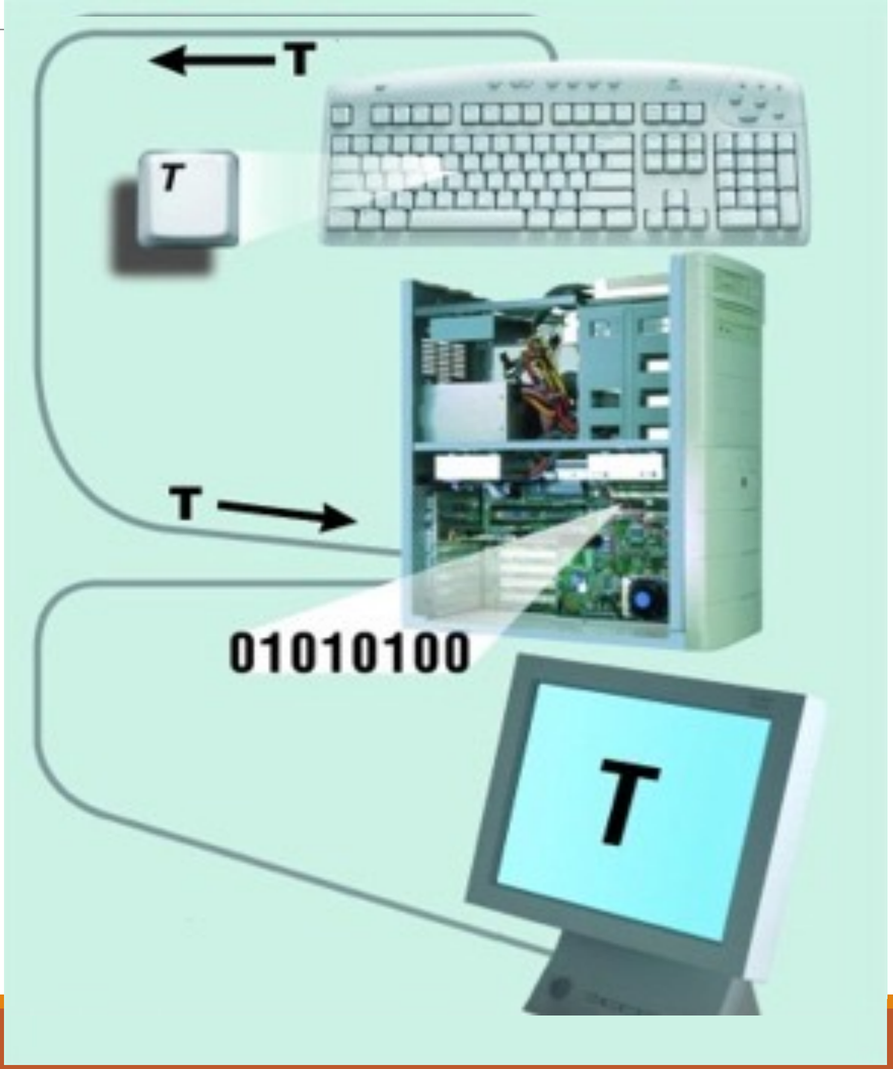
An electronic signal for the letter T is sent to the system unit

Step 3:

The signal for the letter T is converted to its ASCII binary code (01010100) and is stored in memory for processing

Step 4:

After processing, the binary code for the letter T is converted to an image on the output device



Representing Information in Computers

All the different types of information in computers can be represented using binary code.

- Numbers
- Letters of the alphabet and punctuation marks
- Microprocessor instruction
- Graphics/Video
- Sound

Computer Number Systems

Decimal Numbers : **Decimal, b=10**
a={0,1,2,3,4,5,6,7,8,9}

Binary Numbers : **Binary, b=2**
a={0,1}

Octal Numbers : **Octal, b=8**
a={0,1,2,3,4,5,6,7}

Hexadecimal Numbers : **Hexadecimal, b=16**
a={0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

A Number can be Represented

$$(a_5a_4a_3a_2a_1a_0.a_{-1}a_{-2}a_{-3}\ )_r$$

Where r is the base of the number and a_r must be less than r

$(10011)_2$ is a valid number but $(211.01)_2$ is not.

Eg : 365 = $3x10^2$ + $6x10^1$ + $5x10^0$

Decimal Number System

The prefix "**deci**-" stands for 10 and the base is 10

- There are 10 symbols that represent quantities:

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

- Each place value in a decimal number is a power of 10.

Background Information

Any number to the 0 (zero) power is 1.

$4^0 = 1$, $16^0 = 1$ $1{,}482^0 = 1$.

Any number to the 1st power is the number itself.

$10^1 = 10$ $49^1 = 49$ $827^1 = 827$

Decimal Number System

10^3 10^2 10^1 10^0

1000 100 10 1

Example : 1492 $_{10}$

$$
\begin{aligned}
1 \times 1000 &= 1000 \\
4 \times 100 &= 400 \\
9 \times 10 &= 90 \\
2 \times 1 &= +2 \\
\hline
& \ 1492
\end{aligned}
$$

Binary Numbers

The prefix "bi-" stands for 2 and the base is 2

- There are 2 symbols that represent quantities:

  0, 1

- Each place value in a binary number is a power of 2.

$$\begin{array}{cccc} 8 & 4 & 2 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 2^3 & 2^2 & 2^1 & 2^0 \end{array}$$

Binary Numbers

$$(\underbrace{a_5a_4a_3a_2a_1a_0}_{\text{integer}}.\underbrace{a_{-1}a_{-2}a_{-3}}_{\text{fraction}})_2$$

Converting Decimal Numbers to Binary

There are two methods that can be used to convert decimal numbers to binary:

- **Repeated division method**
- **Repeated subtraction method**

Both methods produce the same result and you should use whichever one you are most comfortable with.

Converting Decimal (Contd.)

65_{10}

198_{10}

957_{10}

Repeated Division Method

- Divide the number successively by 2,
- After each division record the remainder which is either 1 or 0.

Example : 123_{10} becomes

123/2 =61 r=1

61/2 = 30 r=1

30/2 = 15 r=0

15/2 = 7 r=1

7/2 =3 r=1

3/2 =1 r=1

1/2 =0 r=1

The result is read from the last remainder upwards

$123_{10} = 1111011_2$

Converting Decimal Numbers to Binary

The Repeated Subtraction method

- Convert the Decimal number **853** to Binary.

- **Step 1:**

  Starting with the 1s place, write down all of the binary place values in order until you get to the first binary place value that is GREATER THAN the decimal number you are trying to convert.

1024 512 256 128 64 32 16 8 4 2 1

Converting Decimal (Contd.)

The Repeated Subtraction Method (Contd.)

- **Step 2:**

  Mark out the largest place value (it just tells us how many place values we need).

853

~~1024~~ 512 256 128 64 32 16 8 4 2 1

Converting Decimal (Contd.)

The Repeated Subtraction Method (Contd.)

- **Step 3:**

  Subtract the largest place value from the decimal number. Place a "1" under that place value.

853-512 = 341

512	256	128	64	32	16	8	4	2	1
1									

Converting Decimal (Contd.)

The Repeated Subtraction Method (Contd.)

- **Step 4:**

  For the rest of the place values, try to subtract each one from the previous result.

  – If you can, place a "1" under that place value.

  – Unless place a "0" under that place value.

Converting Decimal (Contd.)

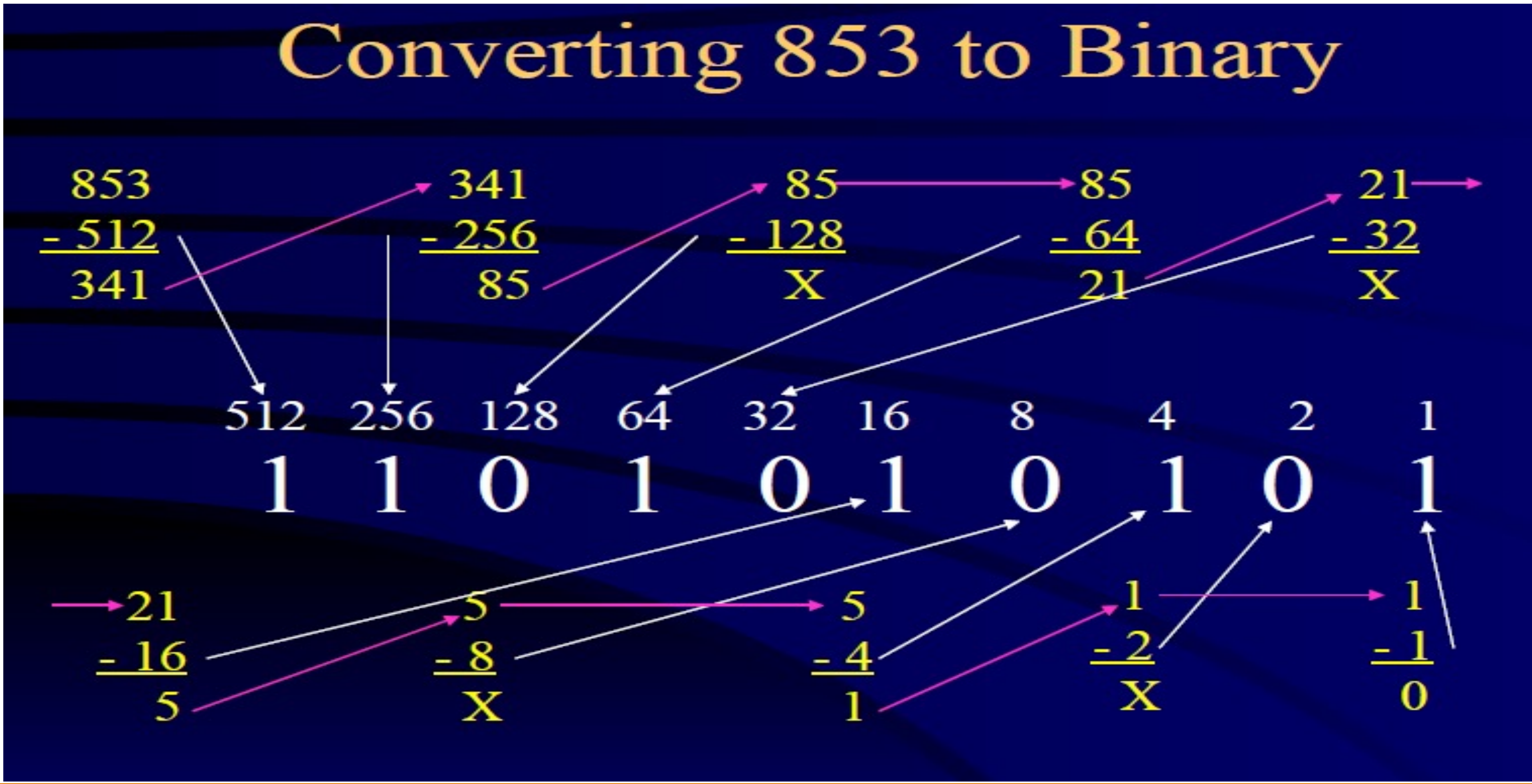
The Repeated Subtraction Method (Contd.)

- **Step 5:**

Repeat Step 4 until all of the place values have been processed.

- ✓ The resulting set of 1s and 0s is the binary equivalent of the decimal number you started with.

The Repeated Subtraction Method



Converting Decimal Fraction to Binary

A decimal number representation of $(0.XY)_{10}$ can be converted into base of 2 and represented by $(0.a_{-1}, a_{-2}, a_{-3}, \text{etc.})_2$.

The fraction number is multiplied by 2, the result of integer part is a_{-1} and fraction part multiply by 2, and then separate integer part from fraction, the integer part represents a_{-2}; this process continues until the fraction becomes 0.

$(0.35)_{10} = (\qquad)_2$

0.35*2	=	0.7	=	0	+	0.7	$a_{-1} = 0$
0.7*2	=	1.4	=	1	+	0.4	$a_{-2} = 1$
0.4*2	=	0.8	=	0	+	0.8	$a_{-3} = 0$
0.8*2	=	1.6	=	1	+	0.6	$a_{-4} = 1$
0.6*2	=	1.2	=	1	+	0.2	$a_{-5} = 1$

Sometimes, the fraction does not reach 0 and the number of bits use for the fraction depends on the accuracy that the user defines, therefore the 0.35 = 0.010011 in binary

Converting from Binary to decimal

$$\underbrace{(a_5a_4a_3a_2a_1a_0}_{\text{Integer}}.\underbrace{a_{-1}a_{-2}a_{-3}}_{\text{Fraction}})_2 = a_0 \times 2^0 + a_1 \times 2^1 + a_2 \times 2^2 + a_3 \times 2^3 + \ldots$$

$$+a_{-1} \times 2^{-1} + a_{-2} \times 2^{-2} + \ldots$$

- Convert $(110111.101)_2$ to decimal.

$$\begin{aligned}(110111.101)_2 &= 1*2^0 + 1*2^1 + 1*2^2 + 0*2^3 + 1*2^4 + 1*2^{*5} + 1*2^{-1} \\ &\quad + 0*2^{-2} + 1*2^{-3} \\ &= 55.625\end{aligned}$$

Number Base Conversion

Binary to Decimal

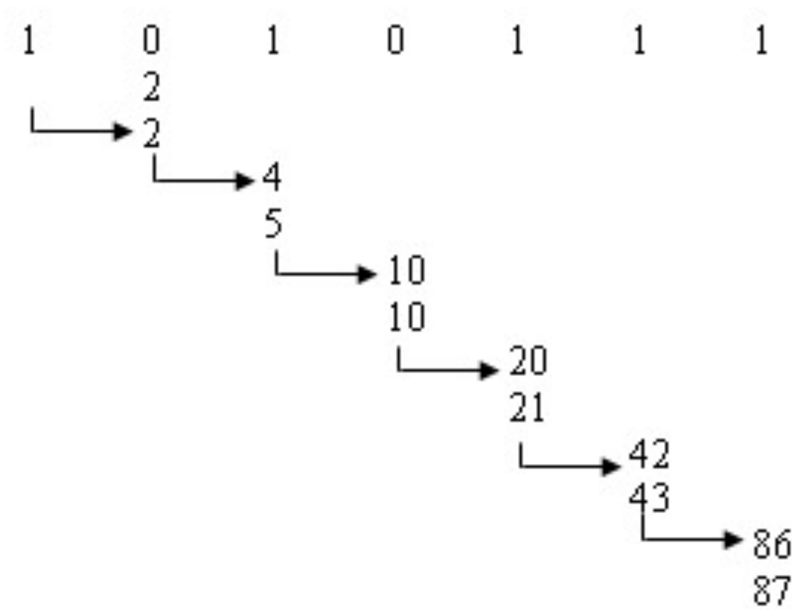
- Take the left most none zero bit,
- Double it and add it to the bit on its right.
- Now take this result, double it and add it to the next bit on the right.
- Continue in this way until the least significant bit has been added in.

Number Base Conversion

110010110_2
111010100_2

Binary to Decimal (contd)

For example, 1010111_2 becomes



Therefore, $1010111_2 = 87_{10}$

65_{10}
198_{10}
957_{10}

Number Base Conversion

Decimal to Octal

- Divide the number successively by 8
- After each division record the remainder which is a number in the range 0 to 7.

Example : 4629_{10} becomes

$4629/8 = 578 \quad r=5$

$578/8 = 72 \quad r=2$

$72/8 = 9 \quad r=0$

$9/8 = 1 \quad r=1$

$1/8 = 0 \quad r=1$

<u>The result is read from the last remainder upwards</u>

$4629_{10} = 11025_8$

65_{10}
198_{10}
957_{10}

Number Base Conversion

Decimal to Hexadecimal

- Divide the number successively by 16
- After each division record the remainder which lies in the decimal range 0 to 15, corresponding to the hexadecimal range 0 to F.

Example : 53241_{10} becomes

53241/16	= 3327	r=9
3327/16	= 207	r=15 = F
207/16	=12	r=15 = F
12/16	=0	r=12 = C

The result is read from the last remainder upwards

53241_{10}=$CFF9_{16}$

110010110_2

111010100_2

Number Base Conversion

Binary to Octal

- Form the bits into groups of three starting at the binary point and move leftwards.
- Replace each group of three bits with the corresponding octal digit (0 to 7).

For example:

11001011101_2 becomes 11 001 011 101

	Octal	3	1	3	5

Therefore $11001011101_2 = 3135_8$

Number Base Conversion

110010110_2
111010100_2

Binary to Hexadecimal

- Form the bits into groups of four bits starting at the decimal point and move leftwards.
- Replace each group of four bits with the corresponding hexadecimal digit from 0 to 9, A, B, C, D, E, and F.

For example:

11001011101_2 becomes 110 0101 1101
6 5 D

Therefore, $11001011101_2 = 65D_{16}$

1755_8

Number Base Conversion

Octal to Decimal

Take the left-most digit, Multiply it by eight and add it to the digit on its right.

Then, multiply this subtotal by eight and add it to the next digit on its right.

The process ends when the left-most digit has been added to the subtotal.

For example, 6437_8 becomes

6 4 3 7

48

52

416

419

3352

3359

Therefore, $6437_8 = 3359_{10}$

1755_8

Number Base Conversion

Octal to Binary

Each octal digit is simply replaced by its 3-bit binary equivalent.

It is important to remember that (say) 3 must be replaced by 011 and not 11.

For example,

41357_8 becomes	4	1	3	5	7
	100	001	011	101	111

Therefore, $41357_8 = 100001011101111_2$.

$1EB9_{16}$

Number Base Conversion

Hexadecimal to Decimal

The method is identical to the procedures for binary and octal except that 16 is used as a multiplier.

For example, 1AC becomes 1 A C

16

26

416

428

Therefore, $1AC_{16} = 428_{10}$

$1EB9_{16}$

Number Base Conversion

Hexadecimal to Binary

Each hexadecimal digit is replaced by its 4-bit binary equivalent.

For example

$AB4C_{16}$ becomes

A	B	4	C
1010	1011	0100	1100

Therefore, $AB4C_{16} = 1010101101001100_2$

Conversion of Fractions

Converting Decimal Fractions to Binary Fractions

For example,

0.6875_{10} becomes

$$
\begin{aligned}
0.6875 \times 2 &= 1.3750 \\
0.3750 \times 2 &= 0.7500 \\
0.7500 \times 2 &= 1.5000 \\
0.5000 \times 2 &= 1.0000
\end{aligned}
$$

0.0000 x 2 ends the process

Therefore, $0.6875_{10} = 0.1011_2$

Conversion of Fractions

Converting Binary Fractions to Decimal Fractions

For example, consider the conversion of 0.01101_2 into decimal form.

0.	0	1	1	0	1
				1/2	
				1/2	
			1/4		
			5/4		
		5/8			
		13/8			
	13/16				
	13/16				
13/32					

Therefore, $0.01101_2 = 13/32 = 0.40625_{10}$

Summary

How to remember

- All of these conversions follow certain patterns that you need to remember.
- When converting **from decimal** you always use **divide**
- When converting **to decimal** you always **multiply**
- Converting between hexadecimal and binary as well as octal and binary is a bit easier to remember.
  - Just remember that **hexadecimal is 8421** and **octal is 421**.
  - The only thing you need to know about converting between **hexadecimal and octal** is that you must always **convert to binary first**.

Thank You