



Digital Systems- IT131

Dr. Alaa Eldeen S. Mohamed

Lecture 01: Number Systems (Part1)

Contents

- Digital Vs Analog
- Numbering Systems [binary, octal and hexadecimal]
- Base Conversion
- Information Representation
- Binary Additions
- Complement

Number Systems (Part1)

Digital versus Analog signal

- We live in an *Analog* world.
- We use the word *Analog* to express phenomena that have smooth gradual and continuous change or movement.
- Everything in the physical world is an analog signal Sound, light, temperature, pressure
- Need to convert into electrical signals

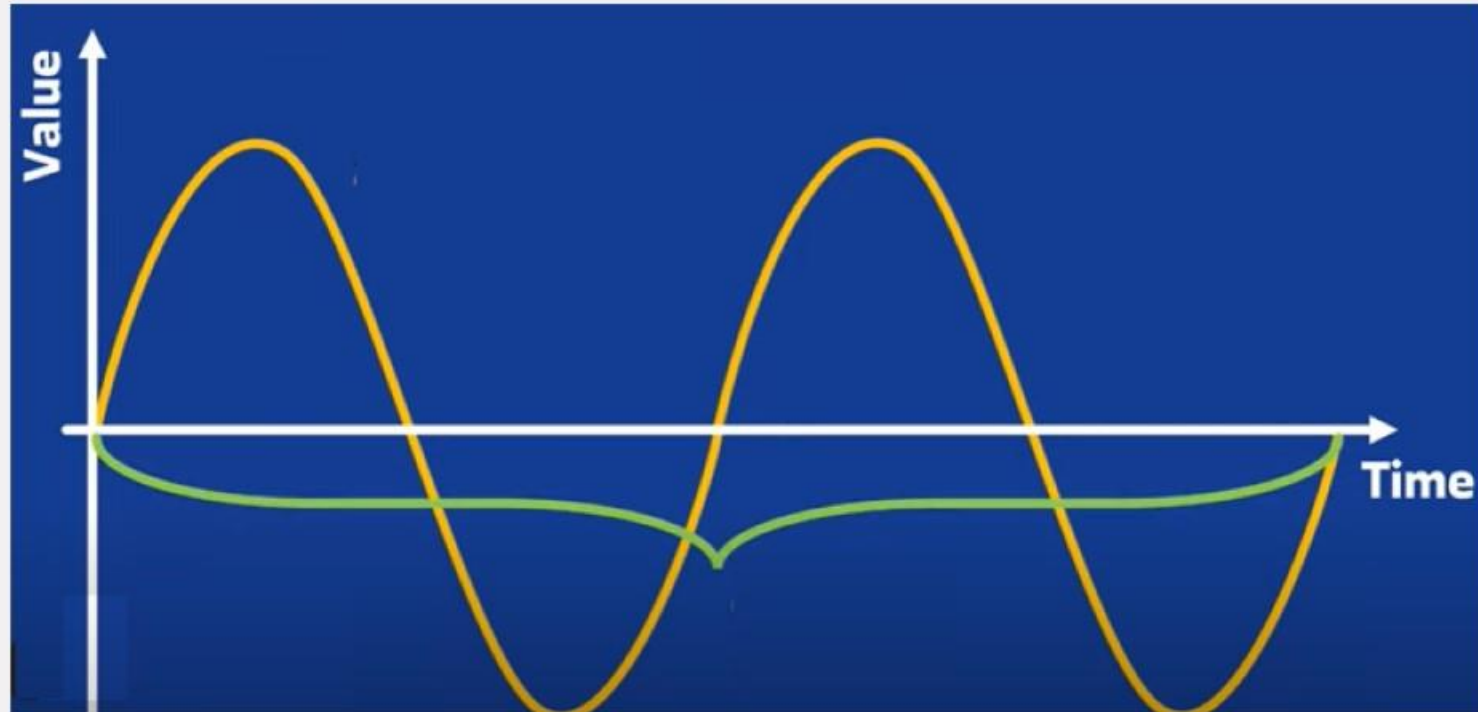
Transducers: converts one type of energy to another

Electro-mechanical, Photonic, Electrical, ...

Examples, Microphone
speaker

Thermocouples (measure temperature)

Analog



Number Systems (Part1)

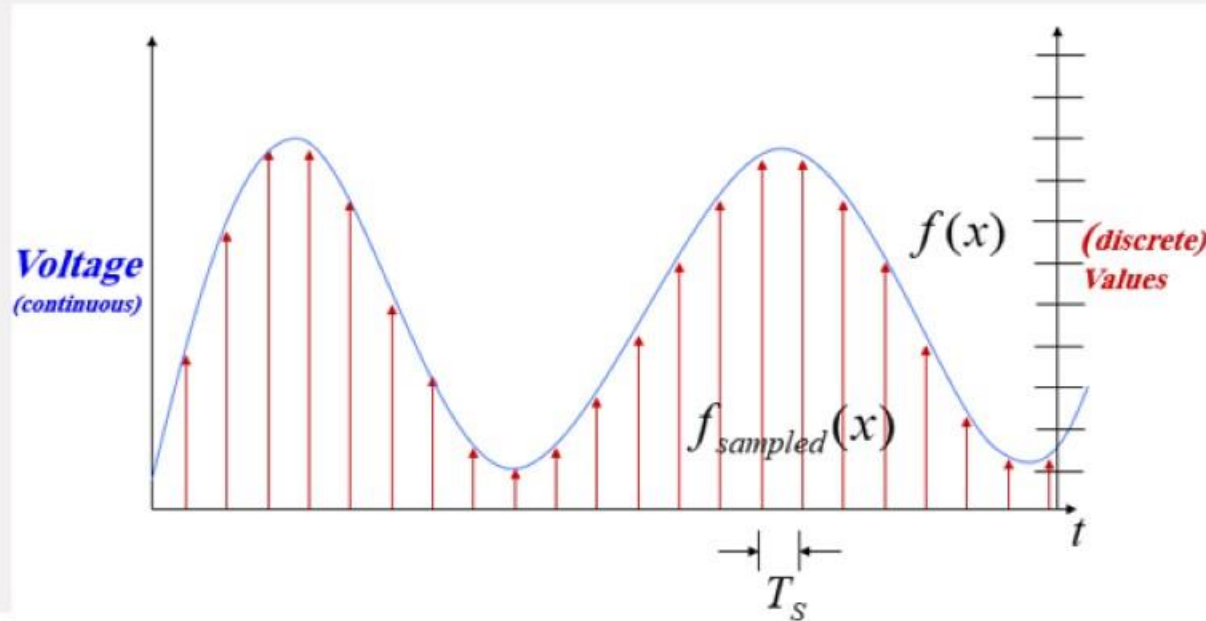
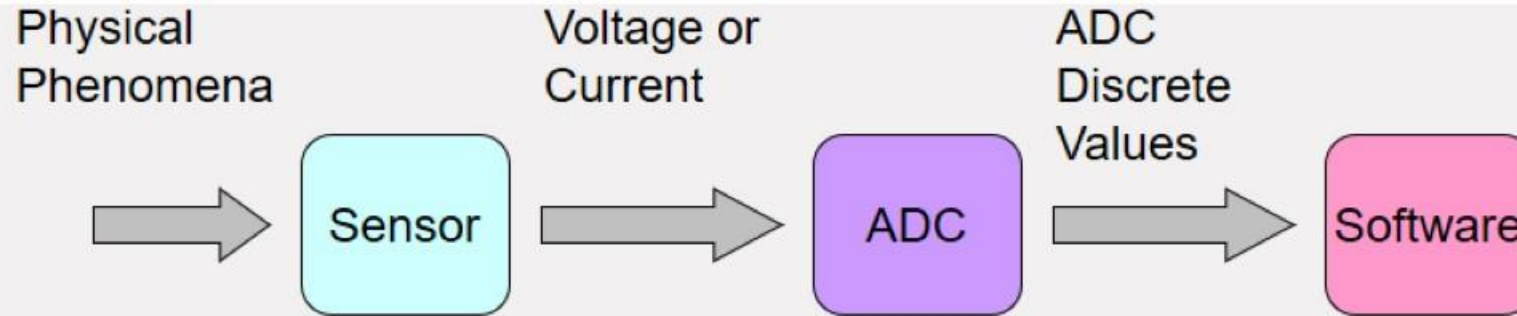
Digital versus Analog

- The word *Digital*, however, means just the opposite.
- In *Digital Systems*, parameters have a limited set of *Discrete Values* that they can assume.
- In Other words, digital parameters don't have a *Continuous range*.
- This means that, digital parameters change their values by *Jumping* from one allowed value to another.
- As an example, the day of the month is a parameter that may only assume one value out of a set of limited discrete values {1, 2, 3, ..., 31}.

Number Systems (Part1)

Digital versus Analog

- To convert an Analog signal into a digital one, some loss of accuracy is inevitable since digital systems can only represent a finite discrete set of values.
- The process of conversion is known as *Digitization or Quantization*.
- Analog-to-digital-converters (ADC) are used to produce a digitized version of analog signals.
- Digital-to-analog-converters (DAC) are used to regenerate analog signals from their digitized form.



Number Systems (Part1)

Digital versus Analog

- Why digital?
 - digital systems are easier to design *since dealing with a limited set of values rather than an infinite (or very large) continuous range of values is significantly simpler*
 - information storage is easy
 - accuracy and precision is better
 - operation can be programmed
 - digital circuits are less affected by noise

Number Systems (Part1)

Example of Digital Systems

- Digital Computer
 - Usually design to maximize performance (speed).
- Handheld Calculator
 - Usually designed to minimize cost.
- Digital Watch
 - Usually designed to minimize power.



Number Systems (Part1)

Numbering Systems

- Computers only deal with binary data (**0s and 1s**), hence all data manipulated by computers must be represented in binary format.
- Machine instructions manipulate many different forms of data:
 - Numbers:
 - Integers: 33, +128, -2827
 - Real numbers: 1.33, +9.55609, -6.76E12, +4.33E-03
 - Alphanumeric characters (letters, numbers, signs, control characters):
examples: A, a, c, 1, 3, ", +, Ctrl, Shift, etc.

Number Systems (Part1)

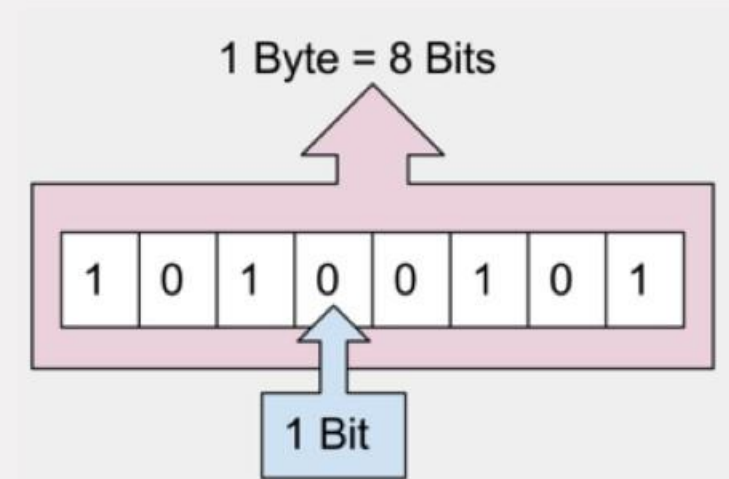
Numbering Systems

- Images (still or moving): Usually represented by numbers representing the Red, Green and Blue (RGB) colors of each pixel in an image,
- Sounds: Numbers representing sound amplitudes sampled at a certain rate.
- So in general we have two major data types that need to be represented in computers; numbers and characters.

Number Systems (Part1)

Binary Numbers

- In binary, we have bytes and bits which are major components in writing binary.
- **Bit**: a single 0 or 1 in a binary number
 - For Example: 10100000
- **Byte**: eight bits put together
 - For Example: 10100000



Number Systems (Part1)

Converting Binary to Decimal

- The Process : Weighted Multiplication
- a) Multiply each bit of the Binary Number by it corresponding bit-weighting factor (i.e. Bit-0 $\rightarrow 2^0=1$; Bit-1 $\rightarrow 2^1=2$; Bit-2 $\rightarrow 2^2=4$; etc).
 - b) Sum up all the products in step (a) to get the *Decimal Number*.

Example:

Convert the binary number 0111_2 into its decimal equivalent

$$(0111)_2 = 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 7_{10}$$

$$(10101001)_2 = (169)_{10}:$$

$$(1 \times 2^7) + (1 \times 2^5) + (1 \times 2^3) + (1 \times 2^0) = 128 + 32 + 8 + 1 = 169$$

Number Systems (Part1)

Binary Numbers

Table 1-3 Binary Bit Position Values.

2^n	Decimal Value	2^n	Decimal Value
2^0	1	2^8	256
2^1	2	2^9	512
2^2	4	2^{10}	1024
2^3	8	2^{11}	2048
2^4	16	2^{12}	4096
2^5	32	2^{13}	8192
2^6	64	2^{14}	16384
2^7	128	2^{15}	32768

Number Systems (Part1)

Convert Decimal to Binary

- Repeatedly divide the decimal integer by 2. Each remainder is a binary digit in the translated value:

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

← least significant bit

← most significant bit

stop when
quotient is zero

$$(37)_{10} = (100101)_2$$

Number Systems (Part1)

Another Procedure for Converting from Decimal to Binary

- Start with a binary representation of all 0's
- Determine the highest possible power of two that is less or equal to the number.
- Put a 1 in the bit position corresponding to the highest power of two found above.
- Subtract the highest power of two found above from the number.
- Repeat the process for the remaining number

Number Systems (Part1)

Another Procedure for Converting from Decimal to Binary

• Example: Converting $(76)_{10}$ to Binary

- The highest power of 2 less or equal to 76 is 64, hence the **seventh (MSB)** bit is 1

1	.	.	.	.	.	.
---	---	---	---	---	---	---

- Subtracting 64 from 76 we get 12.

- The highest power of 2 less or equal to 12 is 8, hence the **fourth** bit position is 1

1	0	0	1	.	.	.
---	---	---	---	---	---	---

- We subtract 8 from 12 and get 4.

- The highest power of 2 less or equal to 4 is 4, hence the **third** bit position is 1

1	0	0	1	1	.	.
---	---	---	---	---	---	---

- Subtracting 4 from 4 yield a zero, hence all the left bits are set to 0 to yield the final answer

1	0	0	1	1	0	0
---	---	---	---	---	---	---

- **Exercises:**

- Convert $(10110)_2 = ??$

Solution:

- $(10110)_2 = 16 + 4 + 2 = (22)_{10}$

What are the weights of the digits with value 1?

Solution,

the weights of the digits with value 1 is: 16 , 4 , 2

- What is the largest binary number that can be expressed with 5 bits?
- 5-bit binary: 11111

- Convert the decimal number 35 to binary

Results of repeated division by 2 (quotients are followed by remainders):

$$35_{10} = 17(\mathbf{1}); 8(\mathbf{1}); 4(\mathbf{0}); 2(\mathbf{0}); 1(\mathbf{0}); 0(\mathbf{1})$$

Answer: $(100011)_2$

- What is the value of the third quotient ?
- The value of the third quotient = 4

Number Systems (Part1)

Hexadecimal Integers

- Binary values are represented in hexadecimal.

Table 1-5 Binary, Decimal, and Hexadecimal Equivalents.

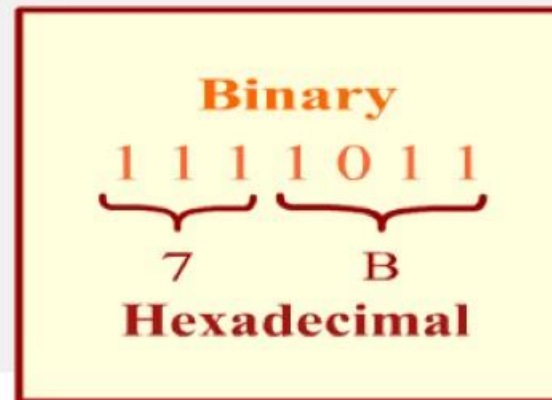
Binary	Decimal	Hexadecimal	Binary	Decimal	Hexadecimal
0000	0	0	1000	8	8
0001	1	1	1001	9	9
0010	2	2	1010	10	A
0011	3	3	1011	11	B
0100	4	4	1100	12	C
0101	5	5	1101	13	D
0110	6	6	1110	14	E
0111	7	7	1111	15	F

Number Systems (Part1)

Converting Binary to Hexadecimal

- Each hexadecimal digit corresponds to 4 binary bits.
- Example: Translate the binary integer **0001 0110 1010 0111 1001 0100** to hexadecimal

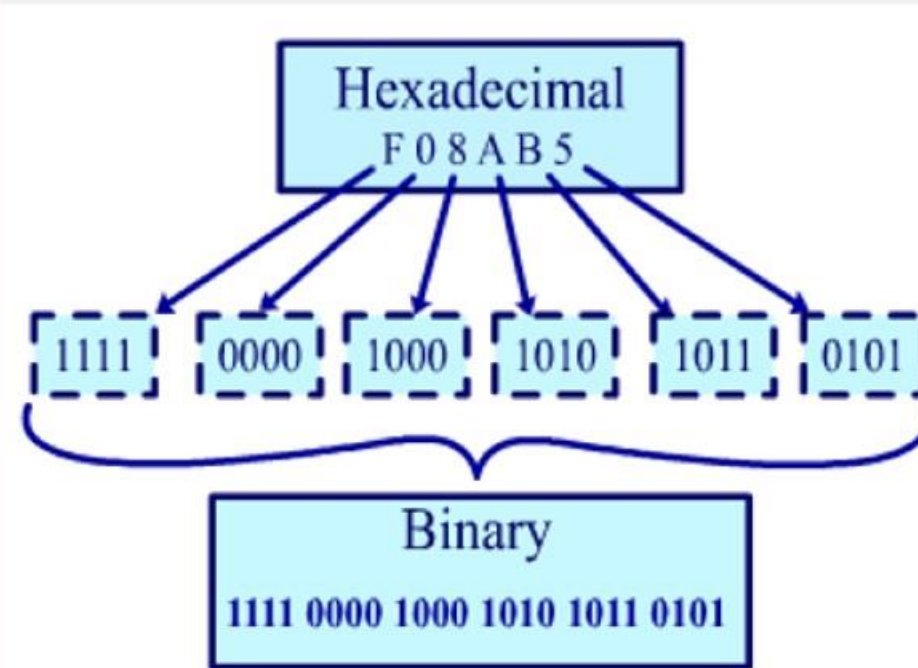
1	6	A	7	9	4
0001	0110	1010	0111	1001	0100



Number Systems (Part1)

Converting Hexadecimal to Binary

- Each Hexadecimal digit can be replaced by its 4-bit binary number to form the binary equivalent.



Number Systems (Part1)

Conversion from Binary to Octal

- Partition binary number into groups of 3 digits starting from the radix point
- When it's not a multiple of 3, you can add 0s to the left or right as you want depending on radix point
- Replace each group of 3 binary digits by the correspondent octal digit
- Example
 - $(11101010)_2 = (011 \ 101 \ 010)_2$
 - $= (352)_8$

Number Systems (Part1)

Conversion from Octal to Binary

- Replace each octal digit by the 3 corresponding binary digits. You can suppress Leading 0s
- Example
 - $(352)_8 = (011\ 101\ 010)_2 = (11101010)_2$
- For conversion from octal to hexadecimal, convert first to binary, then from binary to hexadecimal
- Example
 - $(352)_8 = (011\ 101\ 010)_2 = (11101010)_2$
 - $(11101010)_2 = (1110\ 1010)_2 = (EA)_{16}$
 - $(352)_8 = (EA)_{16}$

- Exercise

- Convert the number $(10110)_2$ to Octal?

- $(10110)_2 = 10\ 110 = (2\ 6)_8$

- Convert the number $(10110)_2$ to Hexadecimal?

$$(1001110)_2 = 100\ 1110 = (4\ E)_{16}$$

- **Exercise**

- Convert the number $(372)_8$ to Binary?

- $(372)_8 = (011\ 111\ 010)_2$

- Convert the number $(10110)_2$ to Hexadecimal?

- $(372)_{16} = (0011\ 0111\ 0010)_2$

Converting Hexadecimal to Decimal

- Multiply each digit by its corresponding power of 16:

$$\text{Decimal} = (d3 \times 16^3) + (d2 \times 16^2) + (d1 \times 16^1) + (d0 \times 16^0)$$

d = hexadecimal digit

- Examples:

$$(1234)_{16} = (1 \times 16^3) + (2 \times 16^2) + (3 \times 16^1) + (4 \times 16^0) = (4,660)_{10}$$

$$(3BA4)_{16} = (3 \times 16^3) + (11 \times 16^2) + (10 \times 16^1) + (4 \times 16^0) = (15,268)_{10}$$

Number Systems (Part1)

Converting Decimal to Hexadecimal

- ❖ Repeatedly divide the decimal integer by 16. Each remainder is a hex digit in the translated value:

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

← least significant digit

← most significant digit

stop when
quotient is zero

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

- Exercise:

- Convert the number $(735)_8$ to decimal number.

$$(735)_8 = 7 * 8^2 + 3 * 8^1 + 5 * 8^0 =$$

$$= 7 * 64 + 3 * 8 + 5 * 1 =$$

$$= 448 + 24 + 5 = 477_{10}$$

What is the value of the third term in conversion equation from right?

the value of the send term in conversion equation from right = 448

- Exercise:

- Convert the number $(FAFA)_{16}$ to decimal number.

- $$\begin{aligned} FAFA_{16} &= 15 \cdot 16^3 + 10 \cdot 16^2 + 15 \cdot 16^1 + 10 \cdot 16^0 = \\ &= 15 \cdot 4096 + 10 \cdot 256 + 15 \cdot 16 + 10 \cdot 1 = \\ &= 61,440 + 2560 + \mathbf{240} + 160 = 64,250 \end{aligned}$$

What is the value of the second term in conversion equation from right?

the value of the send term in conversion equation from right = 240

Convert the following numbers with the indicated bases to decimal:

a. $(4310)_5 = ??$

$$(4310)_5 = 4 * 5^3 + 3 * 5^2 + 1 * 5^1 = (580)_{10}$$

b. $(198)_{12} = ??$

$$(198)_{12} = 1 * 12^2 + 9 * 12^1 + 8 * 12^0 = 260_{10}$$

Number Systems (Part1)

Integer Storage Sizes

Standard sizes:

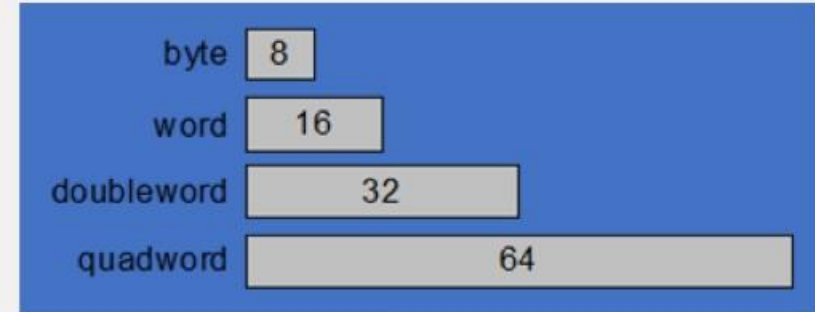


Table 1-4 Ranges of Unsigned Integers.

Storage Type	Range (low–high)	Powers of 2
Unsigned byte	0 to 255	0 to $(2^8 - 1)$
Unsigned word	0 to 65,535	0 to $(2^{16} - 1)$
Unsigned doubleword	0 to 4,294,967,295	0 to $(2^{32} - 1)$
Unsigned quadword	0 to 18,446,744,073,709,551,615	0 to $(2^{64} - 1)$

What is the largest unsigned integer that may be stored in 20 bits?

Number Systems (Part1)

Binary Addition

- Start with the least significant bit (rightmost bit)
- Add each pair of bits
- Include the carry in the addition, if present

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	

Sum Rules	
0+0 = sum 0 carry 0	
0+1 = sum 1 carry 0	
1+0 = sum 1 carry 0	
1+1 = sum 0 carry 1	

Number Systems (Part1)

Binary Addition Example

- Extending this to two multiple bit examples:

$$\begin{array}{r}
 0 \\
 01100 \\
 +10001 \\
 \hline
 \end{array}
 \qquad
 \begin{array}{r}
 0 \\
 10110 \\
 +10111 \\
 \hline
 \end{array}$$

- Note: The 0 is the default Carry-In to the least significant bit.

$$\begin{array}{r}
 00000 \\
 01100 \\
 +10001 \\
 \hline
 11101
 \end{array}
 \qquad
 \begin{array}{r}
 01100 \\
 10110 \\
 +10111 \\
 \hline
 101101
 \end{array}$$

Sum

- **Exercise:**
- **Add the following numbers without converting them to decimal.**

Binary numbers 1011 and 101.

1011

0101

10000

Number Systems (Part1)

Complements in binary systems

- Complements are used for simplifying the subtraction operation and for easy manipulation of certain logical rules and events.
- Two types of complements for each base- r system:
 - (1's complements) and 2's complements).
- **1's complement** is obtained by value obtained by inverting all the bits in the binary representation of the number (swapping 0s and 1s)
 - For example: 1's complement of 1011000 = 0100111

- **2's complement** is obtained by starting at the LSB, leaving all the 0s unchanged, look for the first occurrence of a 1. Leave this 1 unchanged and replace each 1 by 0 and each 0 by 1 for all the bits after it.
- Another method is to add 1 to the 1's complement of the bits in the binary representation of the number.
 - For example: 2's Complement of $11010010 = (00101110)_2$
 - 1's of $11010010 = 00101101$
Then add 1 to it, $00101101 + 1 = (00101110)_2$

- Exercise:

- Obtain the 1's and 2's complements of the following binary numbers:

00000000 $\rightarrow$ 1's complement = 11111111

2's complement = 00000000

11011010 $\rightarrow$ 1's complement = 00100101

2's complement = 00100110

10000101 $\rightarrow$ 1's complement = 01111010 $\rightarrow$ 2's complement = 01111011

- Exercise:

- Find the 2's complement of $(B2FA)_{16}$.

- $B2FA = (1011\ 0010\ 1111\ 1010)_2 \rightarrow 2's\ complement = (0100110100000110)_2$

- Find the 2's complement of $(23)_{10}$.

- $(23)_{10} = (10111)_2 \rightarrow 2's\ complement = (01001)_2$

END OF Lecture 01