Section 4

Summer 2020

Lecture 1

TnF

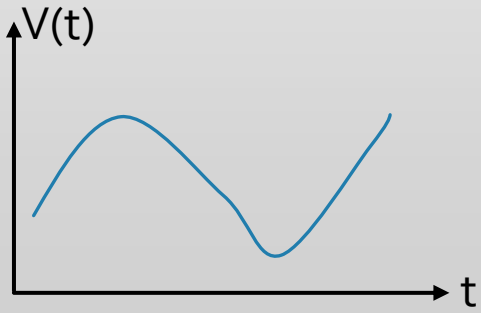
CSE 231: Digital Logic Design

Digital

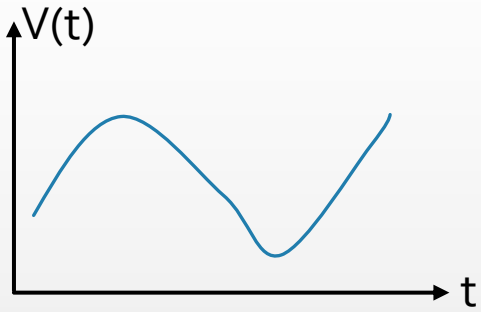
- The term digital is associated with Digits. Most commonly with Binary digits.
- Common use of the term “Digital” includes, digital camera clock, digital clock, digital media and many more.
- It means, these products or services use Digital Technology.
- Digital technology is more efficient in terms of performance and quality with low cost.
- What gave Digital technology power over Analog technology is its:
  - data storing and its preciseness (ability to cancel noise and interference).
  - data transferring capacity. Digital data could be compressed and decompressed for transmission purpose. There are options for the error correction and the encryption. It also provides various transmission options such as serial and parallel.

Signal

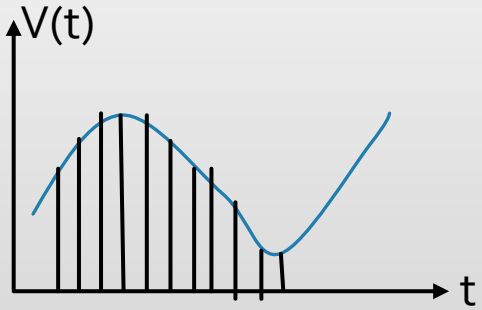
- Signal is a variation in physical parameter. Such as: Speech variation & Temperature variation.
- Signal is a function that represents the variation of physical quantity with respect to any parameter.
- Signals need to be processed so that they could be interpreted and maybe stored. For example, a microphone take the speech signal of the speaker and amplifies it for the listeners.
- There are various types of signals.
- Now in this course we will be focusing on electrical signals.
- Electrical signal is a variation of an electrical quantity (voltage or current) as a function of time.
- A signal may vary continuously.
- In a continuous signal, at any instance of time, we will get a value of the processed parameter.



Signal Cont.



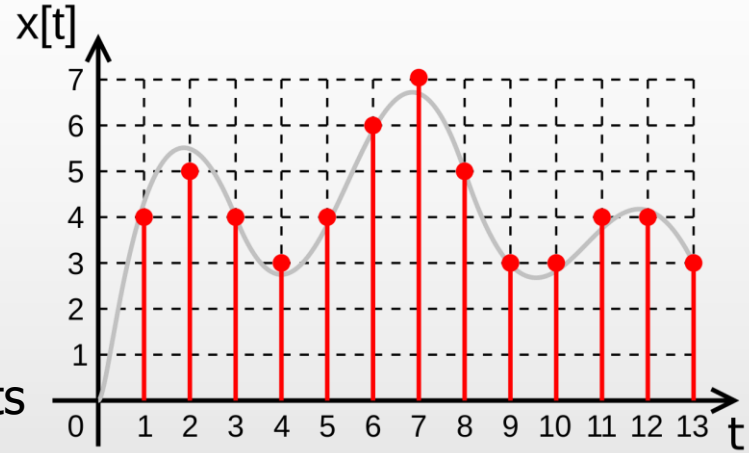
- The signal that takes any value at any & all instance of time is call analog signal within the specified range. Let's say we want to measure the voltage flowing through any system for next 5 minutes.
- But how can I record a signal all the time…
- So the solution is measuring the parameter at discrete time. These signals are known as analog signal.



- Whenever we want to record and retransmit or reproduce a signal, we discretize it.
- The accuracy of re-transmission or reproduction depends on the level of discretization.

Digital Signal

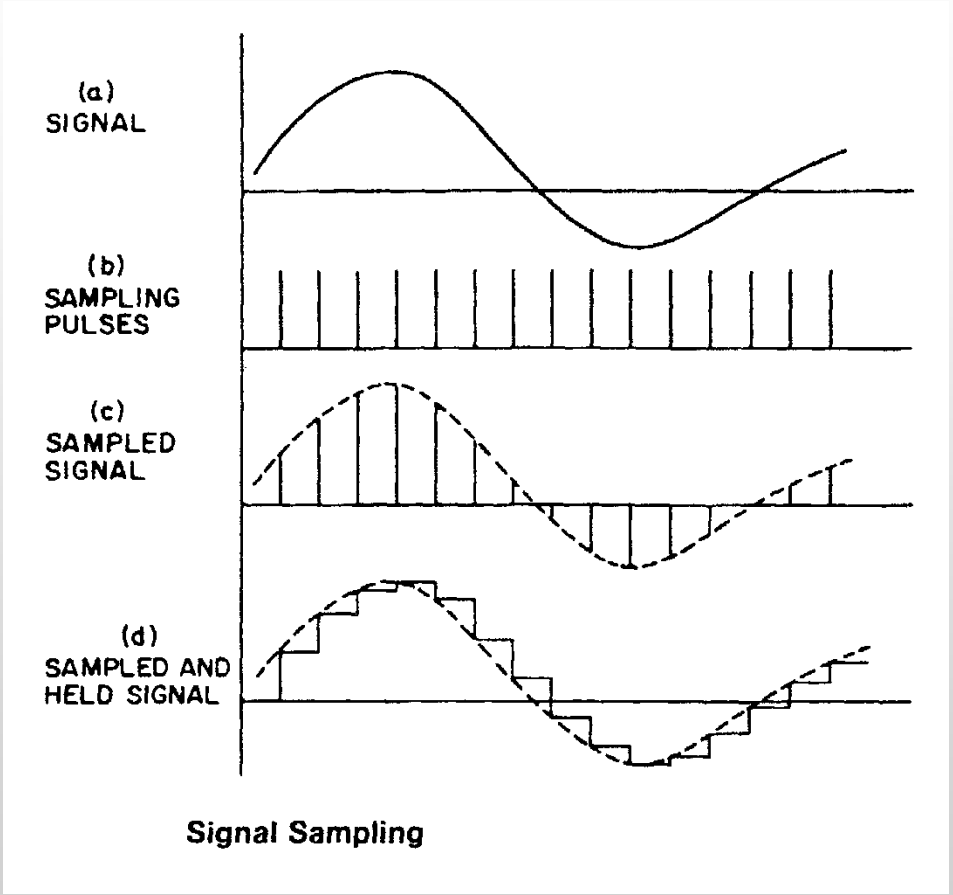
- Digital sigla discretize both time and magnitude.
- Digital signals represent only finite number of discrete values.
- The signals in present day digital system uses Binary.
- Discrete elements of information are represented with group of binary codes in digital system.
- A digital system is a system that manipulates discrete elements of information represented internally in binary form.



- The discrete quantities of information either emerges from nature od the data being processed or may be quantized from a continuous process.
- The fundamental reason for commercial products being made with digital circuits is,, most digital devices are programmable.
- Digital system is an interconnection of digital modules. To understand the operation of each module, it is necessary to have the basic knowledge of digital circuits and logic functions.

Sampling

- **Quantization**
  - Conversion from analog to discrete values
- Quantizing a signal
  - We sample it



Signal Sampling

Datel Data Acquisition and Conversion Handbook

Number Systems

- We will look into four different number systems:
  - Binary: Base 2
  - Decimal : Base 10
  - Octal: Base 8
  - Hexadecimal : Base 16
- Positional number system
  - $101_2 = 1\times2^2 + 0\times2^1 + 1\times2^0$
  - $63_8 = 6\times8^1 + 3\times8^0$
  - $A1_{16} = 10\times16^1 + 1\times16^0$

Binary

- Binary is base 2 and has 2, so we use only 2
- symbols: 0,1
- Binary numbers are built by concatenating a string of bits together. Example: 10101010
- Low Voltage = 0 High Voltage = 1
- Positional Notation: Binary Numbers
- $1011_{bin} = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0$
- $1011_{bin} = (1 * 8) + (0 * 4) + (1 * 2) + (1 * 1)$
- $1011_{bin} = 8 + 0 + 2 + 1 = 11_{dec}$

Binary → hex/decimal/octal conversion

- Conversion from binary to octal/hex
  - Binary: 10011110001
  - Octal: 0010 | 011 | 110 | 001=2361_8
  - Hex: 0100 | 1111 | 0001=$4F1_{16}$
- Conversion from binary to decimal
  - $101_2 = 1\times2^2 + 0\times2^1 + 1\times2^0 = 5_{10}$

Decimal→ binary/octal/hex conversion

Binary

	Quotient	Remainder
56÷2=	28	0
28÷2=	14	0
14÷2=	7	0
7÷2=	3	1
3÷2=	1	1
1÷2=	0	1

Octal

	Quotient	Remainder
56÷8=	7	0
7÷8=	0	7

$56_{10}=111000_2$

$56_{10}=70_8$

Octal & Hexadecimal to Decimal conversion:

$63.4_8 = 6\times8^1 + 3\times8^0 + 4\times8^{-1} = 51.5_{10}$

$A1_{16} = 10\times16^1 + 1\times16^0 = 161_{10}$

Fraction conversion

Convert $(0.6875)_{10}$ to binary.

	Integer		Fraction	Coefficient
$0.6875 \times 2 =$	1	+	0.3750	$a_{-1} = 1$
$0.3750 \times 2 =$	0	+	0.7500	$a_{-2} = 0$
$0.7500 \times 2 =$	1	+	0.5000	$a_{-3} = 1$
$0.5000 \times 2 =$	1	+	0.0000	$a_{-4} = 1$

Convert $(0.513)_{10}$ to octal.

$$0.513 \times 8 = 4.104$$
$$0.104 \times 8 = 0.832$$
$$0.832 \times 8 = 6.656$$
$$0.656 \times 8 = 5.248$$
$$0.248 \times 8 = 1.984$$
$$0.984 \times 8 = 7.872$$

The answer, to seven significant figures, is obtained from the integer part of the products:

$$(0.513)_{10} = (0.406517\ldots)_8$$

Numbers with different bases

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

Sign Number

- How do we write negative binary numbers?
- Historically: 3 approaches
  - Sign-and-magnitude
  - Ones-complement
  - Twos-complement
- For all 3, the most-significant bit (MSB) is the sign digit
  - 0 ≡ positive
  - 1 ≡ negative
- twos-complement is the important one
  - Simplifies arithmetic
  - Used almost universally

Sign number (1's & 2's Complement)

- 1's Complement: 1's complement of a binary number is formed by changing 1's to 0's and 0's to 1's. The following are some numerical examples: The 1's complement of **1011000** is **0100111**.

- Negative number: Bitwise complement positive number
  - $0011 \equiv 3_{10}$
  - $1100 \equiv -3_{10}$

- Solves the arithmetic problem

Add		Invert, add, add carry		Invert and add	
4	0100	4	0100	− 4	1011
+ 3	+ 0011	− 3	+ 1100	+ 3	+ 0011
= 7	= 0111	= 1	1 0000	− 1	1110
		add carry:	+1		
			= 0001		

- 2's complement: 2's complement of binary is obtained by adding 1 to the 1's-complement value.

- Negative number: Bitwise complement plus one
  - $0011 \equiv 3_{10}$
  - $1101 \equiv -3_{10}$

Binary addition & subtraction

- Binary Addition:

				1 ← (carry)
0	0	1	1	1
+0	1	0	1	1
0	1	1	10	11

Binary addition of 13 & 6:

```
00000110
00001101
--------
00010011
```

- Binary Subtraction: (carry)

0	0	1	1
0	1	0	1
0	(1)1	1	0

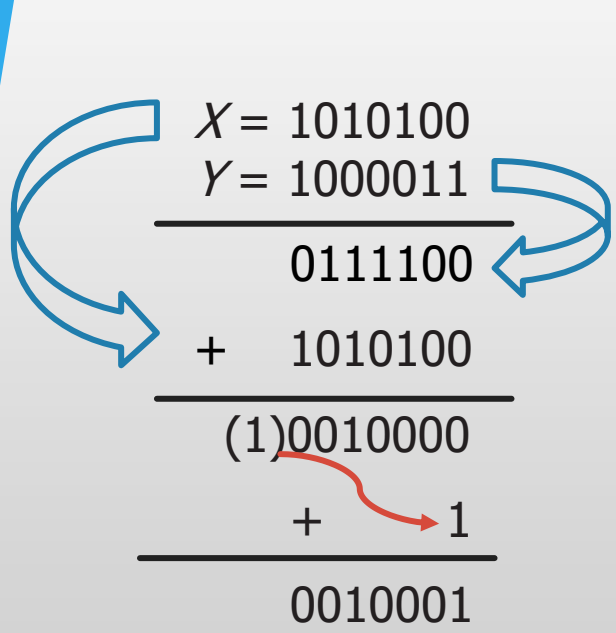
(Borrow)

Binary subtraction of 10 & 5:

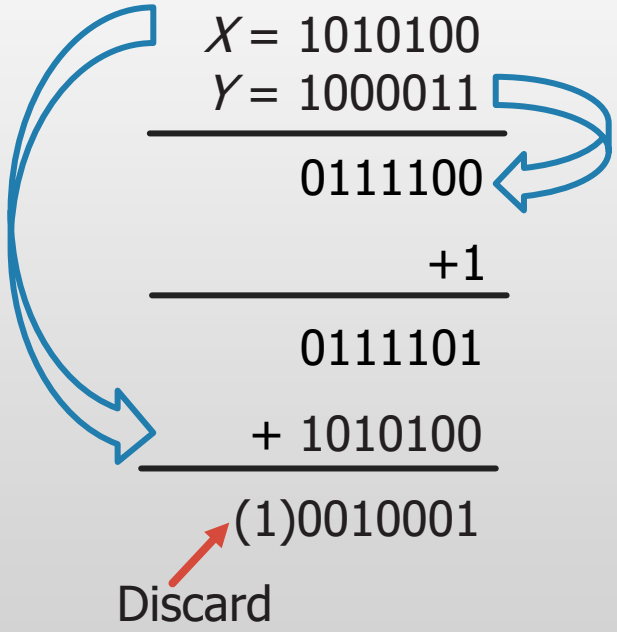
```
 1 0 1 0
(-) 1 0 1
---------
 0 1 0 1
```

Binary Subtraction: 1's & 2's

- 1's Complement Subtraction/ Diminished Radix Complement:



- 2's Complement Subtraction / Radix Complement:



BINARY CODES

- Any discrete element of information that is distinct among a group of quantities can be represented with a binary code (i.e., a pattern of 0's and 1's).
- An n-bit binary code is a group of n bits that assumes up to 2^n distinct combinations of 1's and 0's, with each combination representing one element of the set that is being coded.

Binary-Coded Decimal Code (BCD)

- Decimal values 0-9 is represented in 4-bit binary value as shown in the following table:
- Value higher then 9 is represented as 4-bit binary value of each digit:
- $$(185)_{10} = (0001\ 1000\ 0101)_{BCD}$$
-

Binary-Coded Decimal (BCD)

Decimal Symbol	BCD Digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

BCD Addition

- In BCD, the range is from 0000 to 1001.
- When the binary sum is greater than or equal to 1010, the result is an invalid BCD digit. The addition of 6 = (0110)2 to the binary sum converts it to the correct digit and also produces a carry as required.

4	0100	4	0100	8	1000
+5	+0101	+8	+1000	+9	1001
9	1001	12	1100	17	10001
			+0110		+0110
			10010		10111

Other Decimal Codes

Four Different Binary Codes for the Decimal Digits

Decimal Digit	BCD 8421	2421	Excess-3	8, 4, −2, −1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
	1010	0101	0000	0001
Unused	1011	0110	0001	0010
bit	1100	0111	0010	0011
combi-	1101	1000	1101	1100
nations	1110	1001	1110	1101
	1111	1010	1111	1110

Gray and BCD codes

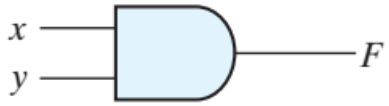
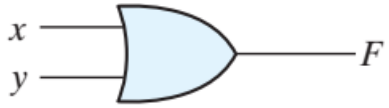
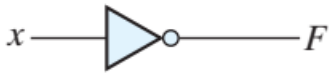
- The Gray code is used in applications in which the normal sequence of binary numbers generated by the hardware may produce an error or ambiguity during the transition from one number to the next.
- A typical application of the Gray code is the representation of analog data by a continuous change in the angular position of a shaft.
- Keeping the left bit fixed, perform and operation in each pair of bits from left most bit.

Decimal Symbols	Gray Code
0	0000
1	0001
2	0011
3	0010
4	0110
5	0111
6	0101
7	0100
8	1100
9	1101

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

Digital Logic Gates

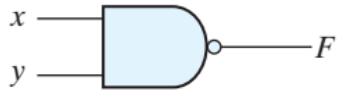
- Three divisions:
- Basic, Advance, Universal
- Basic:

AND		$F = x \cdot y$	x	y	F
			0	0	0
			0	1	0
			1	0	0
			1	1	1
OR		$F = x + y$	x	y	F
			0	0	0
			0	1	1
			1	0	1
			1	1	1
Inverter		$F = x'$	x		F
			0		1
			1		0

Digital Logic Gates

- Universal

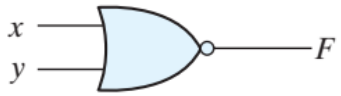
NAND



$F = (xy)'$

x	y	F
0	0	1
0	1	1
1	0	1
1	1	0

NOR

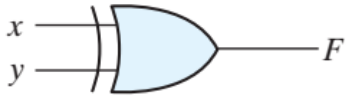


$F = (x + y)'$

x	y	F
0	0	1
0	1	0
1	0	0
1	1	0

- Advance

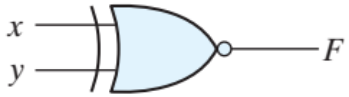
Exclusive-OR (XOR)



$F = xy' + x'y$
$= x \oplus y$

x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

Exclusive-NOR or equivalence



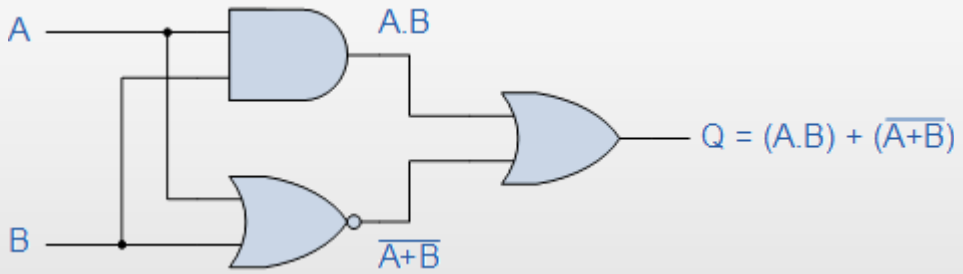
$F = xy + x'y'$
$= (x \oplus y)'$

x	y	F
0	0	1
0	1	0
1	0	0
1	1	1

Combinational circuit example:

- Combinational circuit for the following Boolean function:

$$Q = (A.B)+(A+B)'$$



- Truth Table:

Inputs		Intermediates		Output
B	A	A.B	$\overline{A+B}$	Q
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1