ME 133 lecture 11 2/10/2022

Digital circuits Ch-6

→ Input signal processing

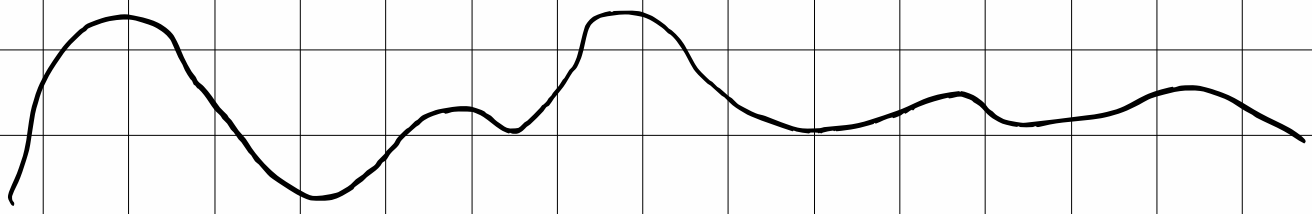
→ digital controller

→ Display.

Today:

- digital representations
- Combinational logic
- Timing Diagrams

What is an analog signal?



- changes continously
- infinite values
- 'analog' → signal represents some other physical quantity

Digital Signals

- Can only take discrete values

→ Binary signal

[on / off
true / false
0, 1

→ Boolean logic

→ binary numbers

Digital Representations

What does base of a number system mean?

Base : # of possible symbols to represent a digit

$10 \rightarrow 0, 1, 2, 3, 4, 5, 6, 7, 8, 9$

Base 10:

$$d_{n-1} \cdots d_3 d_2 d_1 d_0 = d_{n-1} \cdot 10^{n-1}$$
$$+ \vdots$$
$$+ d_3 \cdot 10^3$$
$$+ d_2 \cdot 10^2$$
$$+ d_1 \cdot 10^1$$
$$+ d_0 \cdot 10^0$$

Ex. $123 = 1\times10^{2} + 2\cdot10^{1} + 3\times10^{0}$

$100 + 20 + 3$

$= 123$

→ you can include fractions!

$(d_{-1}, d_{-2}, \ldots)$

Binary Representation (Base 2)

Why? → fundamental operation of digital device can only be on or off (we are using transistors)

$d_{n-1} \cdots d_3 d_2 d_1 d_0 = d_{n-1}\cdot 2^{n-1}$

$+ \vdots$

$+ d_3 2^3 + d_2 2^2$

$+ d_1 2^1 + d_0 2^0$

$\boxed{d_i = 1 \text{ or } 0}$

Ex.

$$1101 = 1\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0$$

$d_3\ d_2\ d_1\ d_0$

$$= 8 + 4 + 0 + 1$$

$$= 13$$

Digit in binary called Bits.

MSB: most significant bit
furthest left
highest power of 2.

LSB: least significant bit.

Byte: 8 bits

→ typically smallest unit used in hardware
e.g. for addresses

From Wiki:

Byte – basic characters (A, B, C ...)

256

Kilobyte – 1024 (2^{10}) Bytes

"Jabberwockey"

megabyte – (2^{20}) Bytes

Harry Potter

gigabyte – (2^{30}) Bytes

half hour video

terabyte – (2^{40}) Bytes

→ largest consumer harddrive in 2001

petabyte – (20^{50})

2000 years of MP3 music.

Binary Arithmetic

$$\begin{array}{r} 9 \\ +3 \\ \hline 12 \end{array}$$

$9 = 1\cdot2^3 + 0\cdot2^2 + 0\cdot2^1 + 1\cdot2^0$

1001

$3 = 0011$

$$\begin{array}{r} {\scriptstyle 1\ 1} \\ 1001 \\ +\ 0011 \\ \hline 1100 \end{array} \rightarrow$$

$1\cdot2^3 + 1\cdot2^2 = 12$

$$\begin{array}{r} 9 \\ \times\ 3 \\ \hline 27 \end{array}$$

$$\begin{array}{r} 1001 \\ \times\ 0011 \\ \hline 1001 \\ 1001 \\ 0000 \\ 0000 \\ \hline 11011 \end{array} \rightarrow 27$$

Binary is annoying to write & read

Hexadecimal is extensively used for programming devices

Base 16 $\{0, \dots, 9,$

A, B, C, D, E, F
$10\ 11\ 12\ 13\ 14\ 15$

$9A = 9 \cdot 16^1 + 10 \cdot 16^0$

$123_{10} = \underbrace{0111}_{7}\ \underbrace{1011}_{B}{}_2 = \underbrace{0x7B}$

ASCII: American Standard Code for Information Interchange

→ for example:

```
char myvar = 'a'
myvar == 56.
```

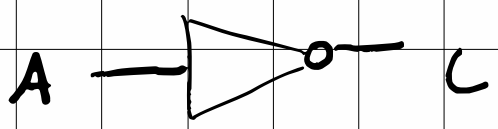
Combinational logic

"Gates" → control flow of signals

Truth Table

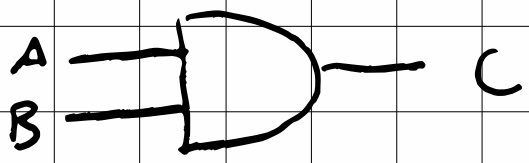
- a way to display all possible combos of input/output

Inverter



A	C
0	1
1	0

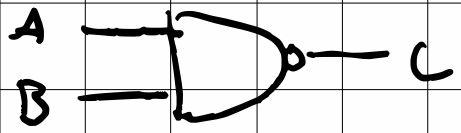
AND



$C = A \cdot B$

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

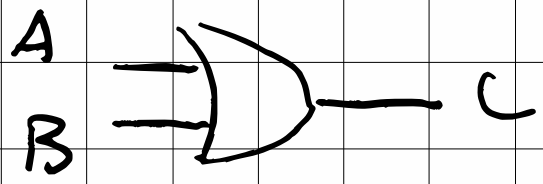
NAND



$C = \overline{A \cdot B}$

A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

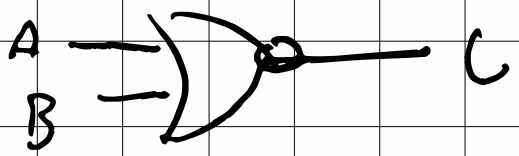
OR



$C = A + B$

A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

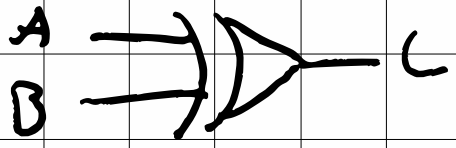
NOR



$C = \overline{A + B}$

A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

XOR

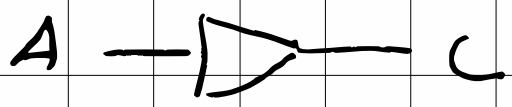


$C = A \oplus B$

$= A \cdot \overline{B} + \overline{A} \cdot B$

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

Buffer



$C = A$

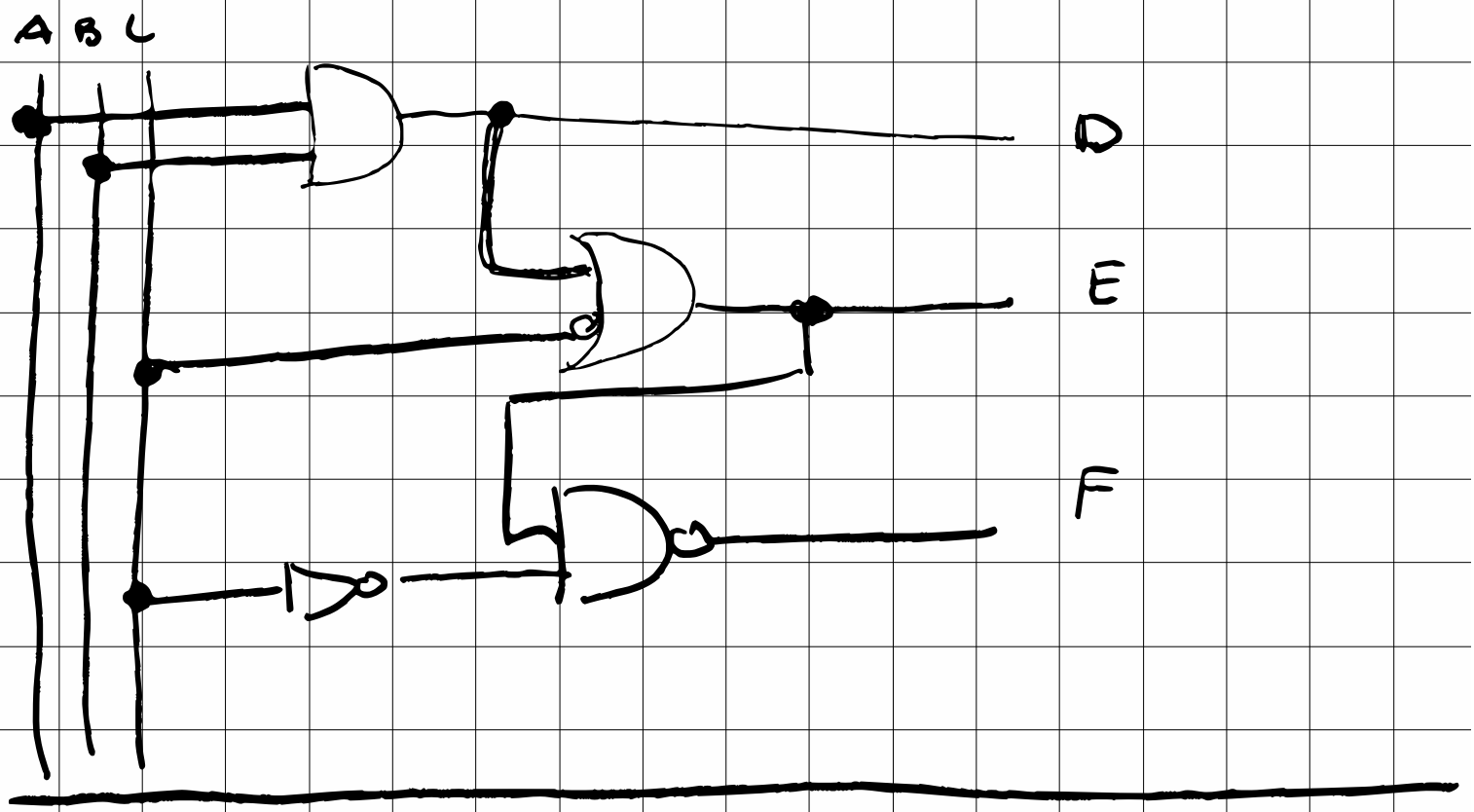
A	C
0	0
1	1

- used to increase current
- important to drive multiple input with single output

  'fanout'

  ⟶ max # of inputs driven by single output
- all gates are manufactured as integrated circuits

Example



$$D = A \cdot B$$

$$E = D + \overline{C} = (A \cdot B) + \overline{C}$$

$$F = \overline{E \cdot \overline{C}}$$

$$F = \overline{[(A \cdot B) + \overline{C}] \cdot \overline{C}}$$

$$D = A \cdot B$$

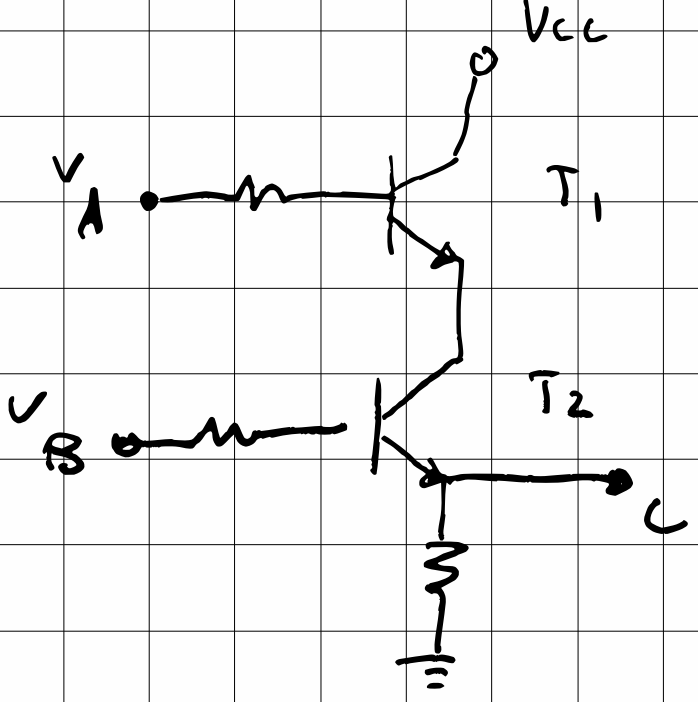
$$E = D + \overline{C} = (A \cdot B) + \overline{C}$$

$$F = \overline{E \cdot \overline{C}}$$

$$F = \overline{[(A \cdot B) + \overline{C}] \cdot \overline{C}}$$

A	B	C	D	E	F
0	0	0	0	1	0
0	0	1	0	0	1
0	1	0	0	1	0
0	1	1	0	0	1
1	0	0	0	1	0
1	0	1	0	0	1
1	1	0	1	1	0
1	1	1	1	1	1

AND gate w/ transistor



V_A = Low $\Rightarrow$ T_1 open

V_C = Low

V_B = Low $\Rightarrow$ T_2 open

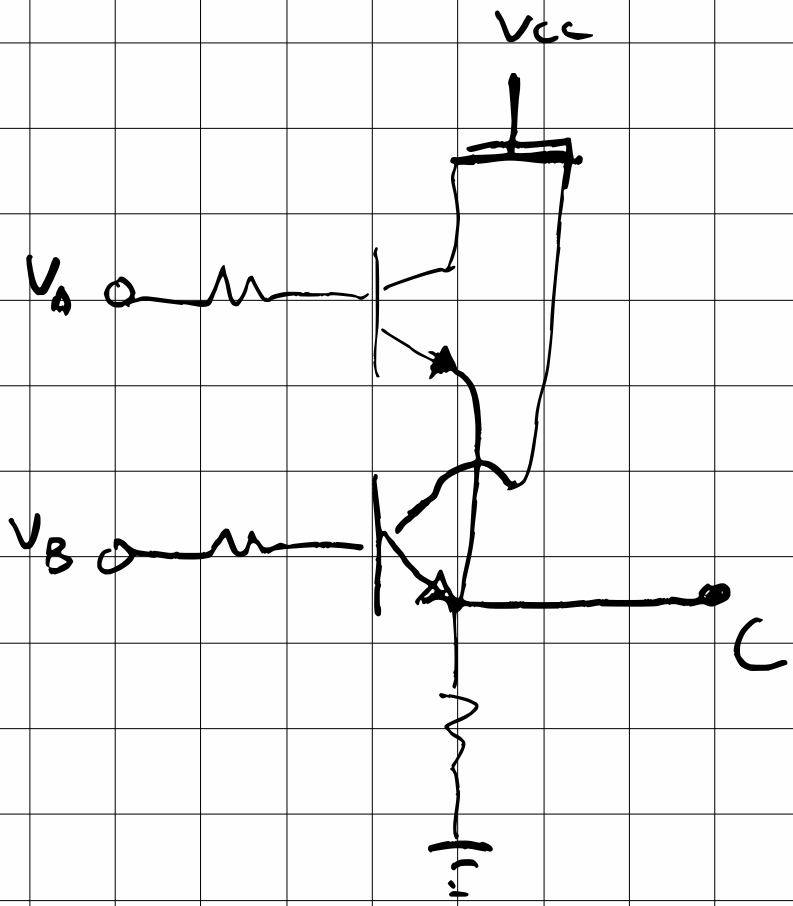
V_C = Low

iif $V_A = V_B$ = High

V_C = High

V_A	V_B	V_C
0	0	0
1	0	0
0	1	0
1	1	1

OR gate



V_A = high

V_C = high

V_B = high

V_C = high

V_A	V_B	V_C
0	0	0
0	1	1
1	0	1
1	1	1

Timing Diagram