AS Systems & Control Study Notes

Part 7: Logic

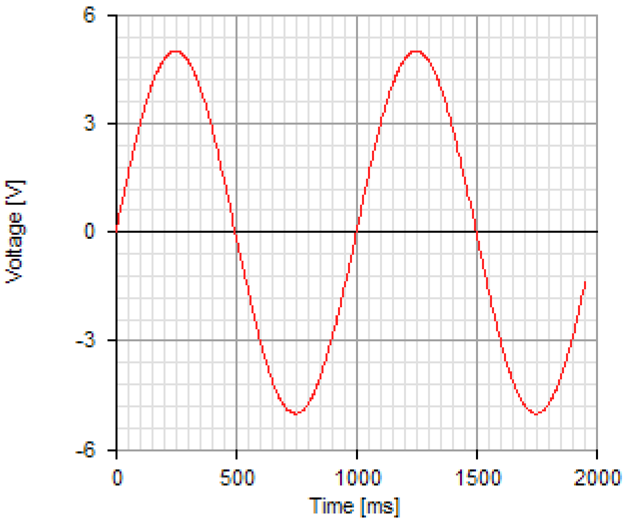
Part 1 Learning Objectives

- Know the difference between analogue and digital signals
- Understand the function of logic gates:
- AND
- OR
- NAND
- NOR
- NOT
- XOR

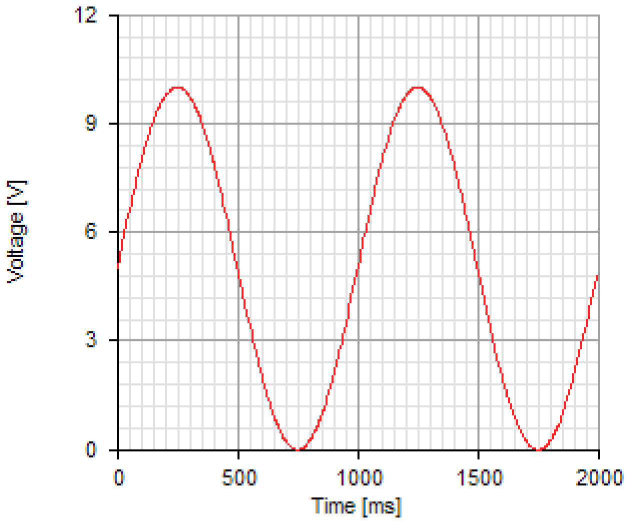
Analogue signals

An analogue signal is a continuous signal where the amplitude varies over time.

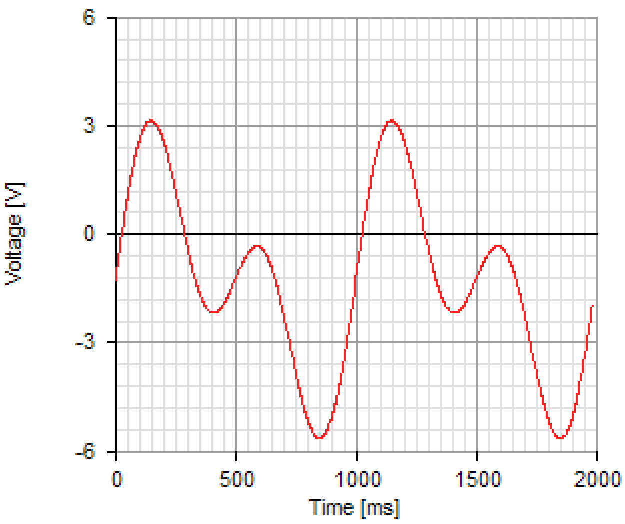
The graphs below show various analogue signals:



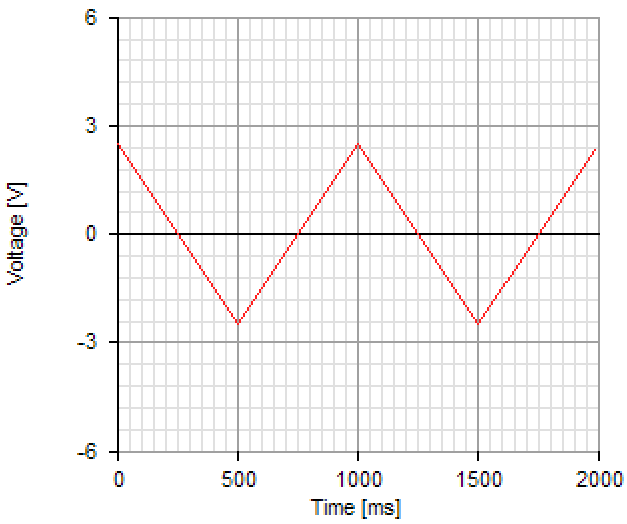
Sinusoidal waveform



DC Sinusoidal waveform (AC with DC offset)



A complex waveform



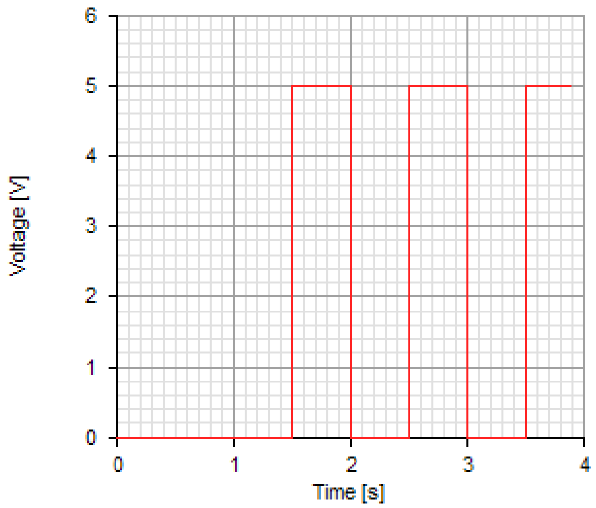
A triangle waveform

E.Clarvis 2009

Digital signals

A digital signal is a signal whose amplitude varies between two discrete levels only.

The graphs below shows a digital signal:



A digital signal

Digital systems

A digital logic system only has two states: On and Off. These on and off states are know by a few different names. These names are interchangeable:

On
High
Logic 1

Off
Low
Logic 0

Logic Functions

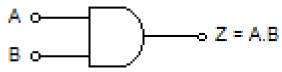
Logic systems usually consist of a range of logic functions. A logical function takes an input or inputs and produces an output. The output is dependent on the particular logic function.

These logical functions are known as Gates. The gates are usually constructed with integrated transistor based circuits (i.e. in chips.)

AND gate

The output of an AND gate is only high when input A AND input B are high:

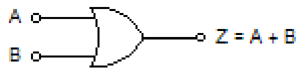
Input B	Input A	Output Z=A.B
0	0	0
0	1	0
1	0	0
1	1	1



OR gate

The output of an OR gate is high when input A OR input B are high OR if both are high:

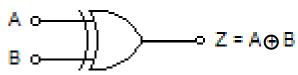
Input B	Input A	Output Z=A+B
0	0	0
0	1	1
1	0	1
1	1	1



XOR gate

The output of an XOR gate is only high when input A OR input B are high. Unlike the OR the output is low if both inputs are high:

Input B	Input A	Output Z=A⊕B
0	0	0
0	1	1
1	0	1
1	1	1

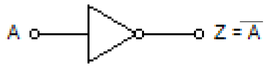


E.Clarvis 2009

NOT gate (Inverter)

A NOT gate produces the opposite logic level to that on its input:

Input A	Output $Z = \overline{A}$
0	0
1	0

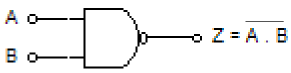


A NOT Gate

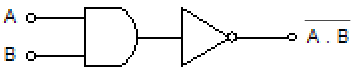
NAND gate

A NAND gate is like an AND gate with a NOT gate on the end. The output is therefore low when both inputs are high:

Input B	Input A	Output $Z=\overline{A.B}$
0	0	1
0	1	1
1	0	1
1	1	0



A NAND Gate

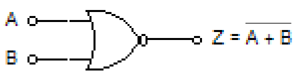


NAND Equivalent Circuit

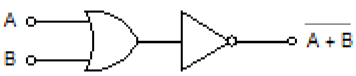
NOR gate

A NOR gate is like an OR gate with a NOT gate on the end. The output is therefore only high when both inputs are low:

Input B	Input A	Output $Z=\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0



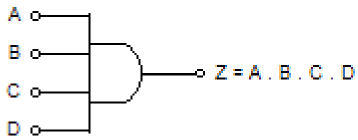
A NOR Gate



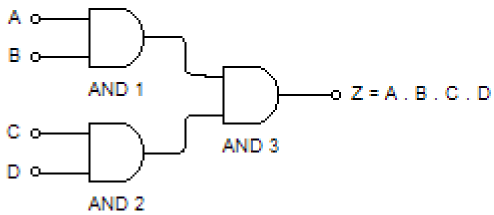
NOR Equivalent Circuit

Combinational logic

A number of logic gates can be connected to create more complicated systems. For example, the circuit below shows how three AND gates can be used to create a four-input AND gate:



A four-input AND gate



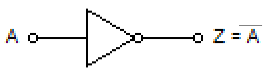
A four-input AND gate using three AND gates

Input D	Input C	Input B	Input A	AND1 (A.B)	AND2 (C.D)	AND3 / Z A.B . C.D
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	1	0	0
1	1	0	0	0	1	0
1	1	0	1	0	1	0
1	1	1	0	0	1	0
1	1	1	1	1	1	1

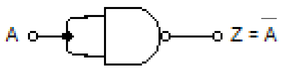
Four-input AND gate truth table

Making NOT gates

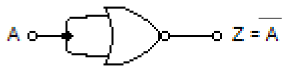
NOT gates can be made from NOR and NAND gates by simply connecting their two inputs together. The three circuits shown below all perform the same function:



A NOT Gate



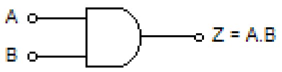
A NOT Gate (Using a NAND)



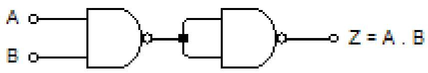
A NOT Gate (Using a NOR)

Making an AND gate using NANDs

An AND gate can be made by connecting a NOT gate to its output:



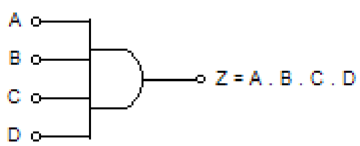
An AND gate)



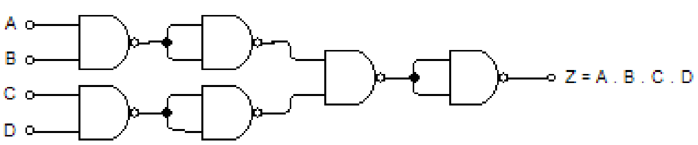
An AND gate (using NANDs)

Making a 4-input AND gate using NANDs

An AND gate can be made by connecting a NOT gate to its output. It isn't important that you know this but it illustrates how logic gates can be combined:



A 4-input AND gate

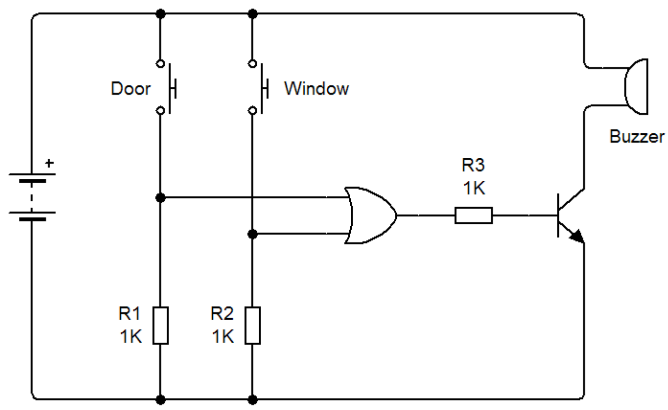


A 4-input AND gate made from NAND gates

A practical logic circuit

This is an example of how an OR gate can be used as part of a simple alarm. The two switches are fitted on the door and window of a house. When a door or window is opened the buzzer will sound.

Both inputs are tied low through resistors R1 and R2. Therefore when either door is opened a logic 1 is applied to the OR gate which then switches the siren on:



Simple house alarm using an OR gate