# Sistemas Digitais I

LESI - 2º ano

Unit 1 - Introduction

**João Miguel Fernandes**
www.di.uminho.pt/~jmf

DEP. DE INFORMÁTICA
ESCOLA DE ENGENHARIA
UNIVERSIDADE DO MINHO

# 1. Introduction

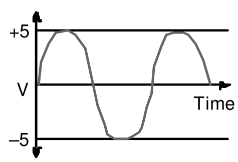
*- Summary -*

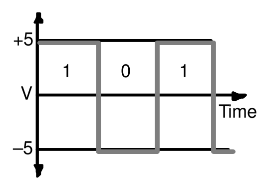
- Digital vs. Analog Systems
- Digital Abstraction
- Synchronous vs. Asynchronous Systems
- Gates
- Flip-flops
- Software tools
- Integrated Circuits
- PLDs
- Digital Design Levels

# 1. Introduction

*- Digital vs. Analog Systems (1) -*

- A system is a set of related parts that actuate as a whole to achieve a given goal.
- A system has inputs and outputs and exhibits behaviour characterised by functions that translate inputs into outputs.
- Analog systems process time-varying signals that can take on any value across a continuous range of voltage, current, or ...
- The same applies to digital systems: the difference is that we pretend that they don't.

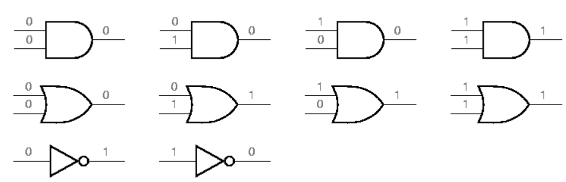



# 1. Introduction

*- Gates -*

- Gates are the most basic digital devices.
- A gate has one or more inputs and produces an output that is a function of the current input values.
- A gate is a combinational circuit, because its output depends only on the current input combination.



# 1. Introduction

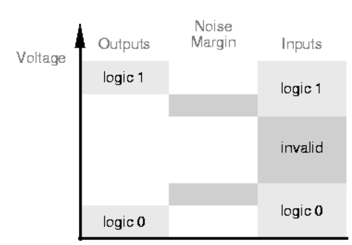
*- Digital vs. Analog Systems (2) -*

- The critical advantage of digital systems is their ability to deal with electrical signals that have been degraded.
- Due to the discrete nature of the outputs, a slight variation in an input is still interpreted correctly.
- In analog circuits, a slight error at the input generates an error at the output.
- The simplest form of digital system is binary.
- A binary signal is modelled as taking on only two discrete values (0 or 1, LOW or HIGH, False or True).

# 1. Introduction

*- Digital Abstraction -*

- Digital circuits deal with analog voltages and currents.
- The digital abstraction allows analog behaviour to be ignored in most cases, so circuits can be modelled as if they really process 0s and 1s.
- Association of a range of analog values with each logic value (0 and 1).
- The difference between the range boundaries is called noise margin.



# 1. Introduction
*- Synchronous vs. Asynchronous Systems -*

- A <u>synchronous system</u> is one whose elements change their values at certain specified times.
- An <u>asynchronous system</u> has outputs that can change at any time.
- As an example, consider a digital alarm clock, which has the alarm set for 13:59.
- In a synchronous system, the outputs all change at the same time: (12:59 $\rightarrow$ 13:00 $\rightarrow$ 13:01 $\rightarrow$ ...).
- In an asynchronous system, outputs are not constrained to change simultaneously: (12:59 $\rightarrow$ 13:59 $\rightarrow$ 13:00 $\rightarrow$ ...).

# 1. Introduction
*- Flip-flops -*

- A <u>flip-flop</u> is a device that store either a 0 or a 1.
- The <u>state</u> of a flip-flop is the value currently stored.
- The stored value can only be changed at certain times, regulated by a "<u>clock</u>" input.
- A digital circuit that contains flip-flops is called a <u>sequential circuit</u>.
- The output of a sequential circuit depend, at any time, not only in its current input but also on the past sequence of inputs that have been applied to it.
- A sequential circuit has <u>memory</u> of past events.

# 1. Introduction
*- Software Tools -*

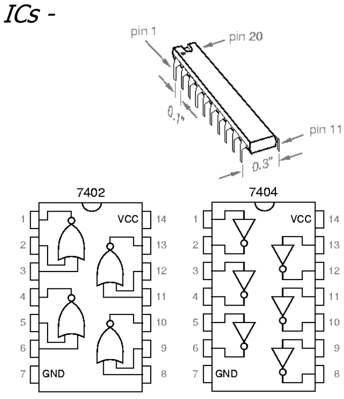
- Digital design need not involve any software tools.
- Software tools are nowadays an essential part of digital design.
- HDLs (<u>Hardware Description Languages</u>) and the corresponding simulation and synthesis tools are widely used.
- In a CAD (Computer-Aided Design) environment, the tools improve the productivity and help to correct errors and to predict behaviour.
  - **Schematic entry;**
  - **HDLs compilers, simulators and synthesis tools;**
  - **Timing analysers;**
  - **Simulators**
  - **Test benches.**

# 1. Introduction
*- Integrated Circuits (ICs) -*

- An <u>IC</u> is a collection of gates fabricated on a single silicon chip.
- ICs are classified by their size:
  - **SSI (small scale integration): 1 to 20 gates<br>- ANDs, ORs, NOTs.**
  - **MSI (medium scale integration): 20 to 200 gates<br>- decoder, register, counter.**
  - **LSI (large scale integration): 200 to 200.000 gates<br>- small memories, PLDs.**
  - **VLSI (very large scale integration): > 1.000.000 transistors<br>- microprocessors, memories.**
- The Pentium 4 has 42.000.000 transistors!!!

# 1. Introduction
*- SSI ICs -*

- Dual in-line pin (DIP) packages.
- A <u>pin diagram</u> shows the assignment of device signals to package pins.
- Nowadays, SSI ICs are used as "glue" to tie together larger components in complex systems.
- SSI ICs have been largely supplanted by PLDs (Programmable Logic Devices).

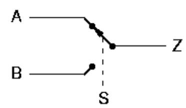



# 1. Introduction
*- Programmable Logic Devices (PLDs) -*

- Some ICs can have their logic function "programmed" into them after they are manufactured.
- Most of them can even be **<u>re</u>**programmed, which allows bugs to be corrected without replacing or rewiring the device.
- PLD (PLA or PAL): two-level structure of AND and OR gates with user-programmable connections.
- CPLDs (Complex PLDs) and FPGAs (Field Programmable Gate Arrays) were devised to accommodate larger systems.
- HDLs and the respective tools allow a design to be compiled, synthesised, and downloaded into a device in a short time.
- This permits <u>rapid prototyping</u> to be a reality.

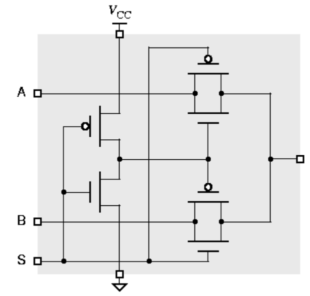
# 1. Introduction
*- Digital Design Levels (1) -*

- Digital design can be carried out at several different levels of representation and abstraction.
- Although one may practice design at a particular level, sometimes he needs to go up and down to understand what is going on.
- The lowest level is device physics and IC manufacturing processes. [not covered by this course]
- The transistor level is the next one. [not covered by this course]
- To explain this and the next levels, consider a multiplexer, with 2 input bits (A, B), 1 control bit (S), and 1 output bit (Z).




# 1. Introduction
*- Digital Design Levels (2) -*

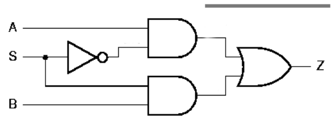
- For some functions it is advantageous to optimise them by designing at the transistor level.
- The multiplexer can be designed in CMOS technology using transistor circuit structures.
- Using this approach, the multiplexer can be built with just 6 transistors.




# 1. Introduction
*- Digital Design Levels (3) -*

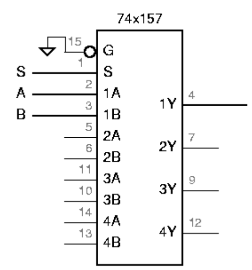
- In traditional logic design, a truth table is used to describe the multiplexer's logic function.
- A truth table lists all possible combinations of input values and the corresponding output values.
- Once a truth table exists, Boolean algebra and minimization algorithms are used to derive an "optimal" 2-level AND-OR equation.
- For the multiplexer: Z = S'.A + S.B
- This equation can be converted into a gate level logic diagram.

| S | A | B | Z |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |




# 1. Introduction
*- Digital Design Levels (4) -*

- For commonly used functions, most digital technologies provide predefined building blocks.
- The 74x157 is an MSI chip that performs multiplexing on 2 4-bit inputs.
- The figure shows the block level diagram for the 74x157 chip.
- The numbers in colour are pin numbers of a 16-pin DIP package containing the device.




# 1. Introduction
*- Digital Design Levels (5) -*

- HDLs, like VHDL, can be used to specify the multiplexer's function at the algorithmic level.
- The entity specifies the inputs and outputs of the circuit.
- The architecture defines the multiplexer behaviour.
- A synthesis tool can process this behavioural description and produce a circuit in a given target technology.

```
library IEEE;
use IEEE.std_logic_1164.all;

entity Vchap1mux is
    port ( A, B, S: in  STD_LOGIC;
           Z:       out STD_LOGIC );
end Vchap1mux;

architecture Vchap1mux_arch of Vchap1mux is
begin
  Z <= A when S = '0' else B;
end Vchap1mux_arch;
```