Universitat Autònoma de Barcelona

1.1 DIGITAL SYSTEMS

Jean-Pierre Deschamps

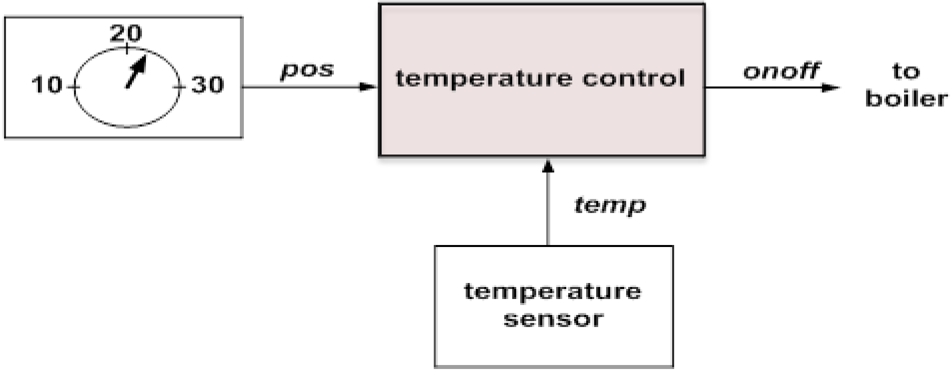
1 PHYSICAL SYSTEM

Universitat Autònoma de Barcelona

Set of interconnected objects or elements that realize some function. Characterized by

- a set of input signals,
- a set of output signals,
- TYPE (voltage, force, temperature, position of a switch, etc.),
- RANGE.
- a relation between input and output signals.

1 PHYSICAL SYSTEM: control of a boiler



Input and output signals

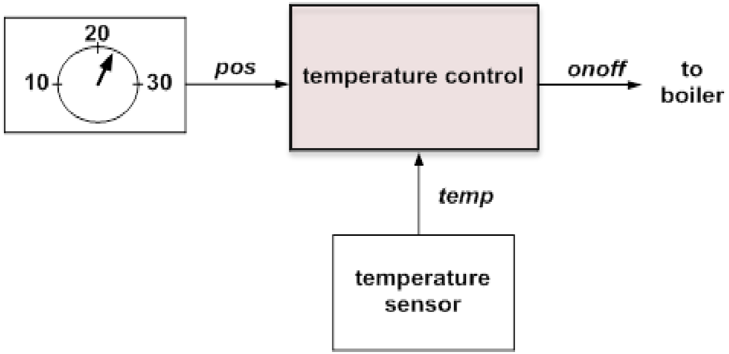
pos: selector position;

temp: temperature measured by the sensor;

onoff: binary signal (two values: ON or OFF).

1 PHYSICAL SYSTEM: control of a boiler

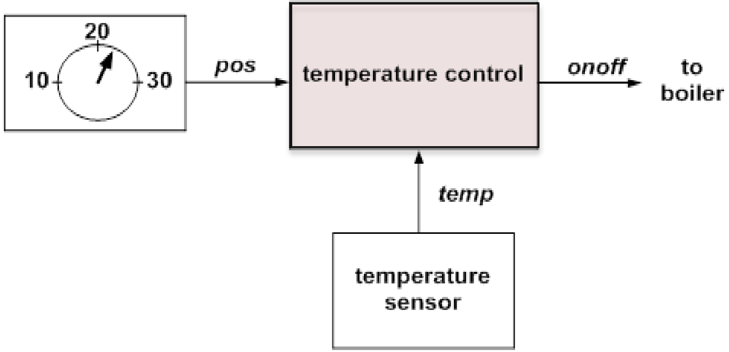
UAB
Universitat Autònoma de Barcelona



Relation between inputs and outputs

```
loop
  if temp < pos – half_degree then onoff <= on;
  elsif temp > pos + half_degree then onoff <= off;
  end if;
  wait for 10 s;
end loop;
```

1 PHYSICAL SYSTEM: control of a boiler

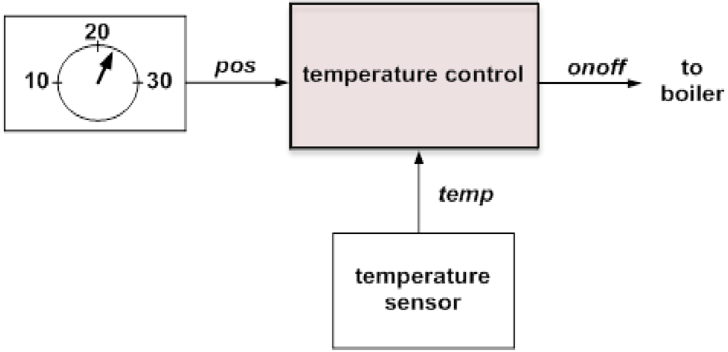


Type and range of input and output signals

- ***pos*** represents the position of a selector between two extreme positions (10 and 30); its value might be any position in

$$pos_{10} \leq pos \leq pos_{30} \; ;$$

1 PHYSICAL SYSTEM: control of a boiler



Type and range of input and output signals

- ***temp*** represents the ambient temperature; assuming that the sensor measures temperatures between 0 and 50 degrees, its value might be any temperature in

$$temp_0 \leq temp \leq temp_{50} \ ;$$

- ***onoff*** has only two possible values: ON and OFF.

Signals such as ***pos*** and ***temp*** that can take any value within a continuous (and thus infinite) set of values, are called …

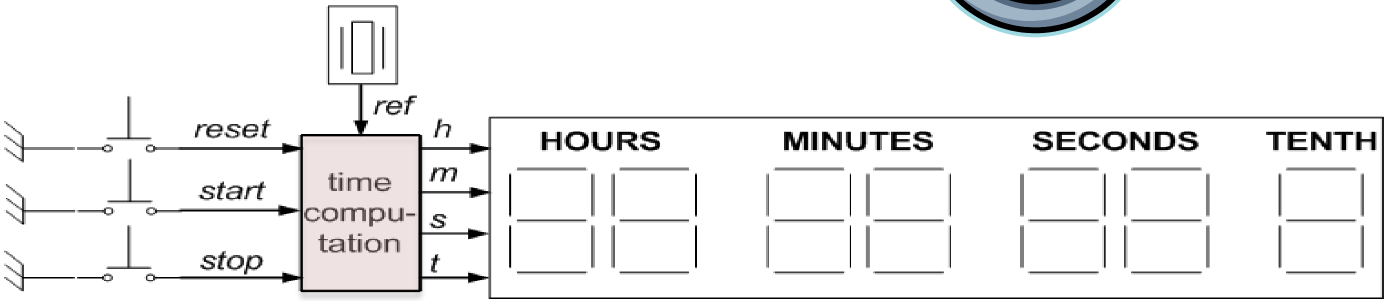
ANALOG SIGNALS

Signals such as ***onoff*** whose values belong to a finite set (in this case a 2-element set), are called …

DIGITAL (or DISCRETE) SIGNALS

1 PHYSICAL SYSTEM: chronometer

Universitat Autònoma de Barcelona



Input and output signals

reset, *start*, *stop*: position of three push-buttons;

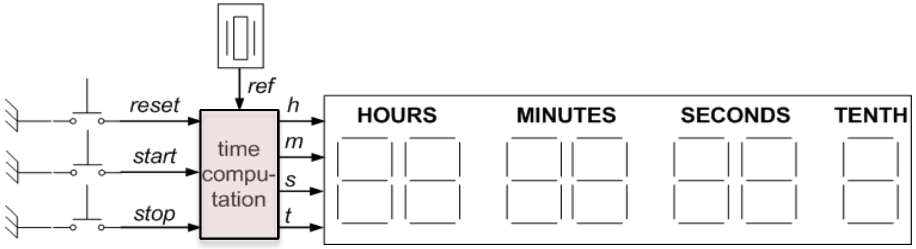
ref: 10 Hz square wave, V_L = 0V, V_H = 1V;

h: integer between 0 and 23;

m and *s*: integers between 0 and 59;

t: integer between 0 and 9.

1 PHYSICAL SYSTEM: chronometer



Relation between inputs and outputs
(natural language)

- when ***reset*** is pushed down, $h = m = s = t = 0$;
- when ***start*** is pushed down, the chronometer starts counting; *h*, *m*, *s* and *t* represent the elapsed time in tenth of seconds;
- when ***stop*** is pushed down, the chronometer stops counting; *h*, *m*, *s* and *t* represent the latest elapsed time.

Question

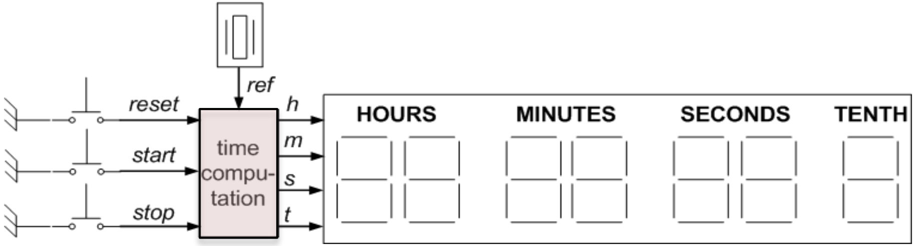
Assume that the current state of the chronometer is

17 hours, 22 minutes, 59 seconds, 9 tenth;

What will be the chronometer state after the next clock pulse?

1. 17 hours, 23 minutes, 60 seconds, 10 tenth;
2. 18 hours, 22 minutes, 59 seconds, 9 tenth;
3. 17 hours, 23 minutes, 0 seconds, 0 tenth;
4. 17 hours, 22 minutes, 59 seconds, 8 tenth;

1 PHYSICAL SYSTEM: chronometer



Type and range of input and output signals

All input and output signals are digital (or discrete)

- *reset*, *start*, *stop*: two values (ON, OFF);
- *ref*: two values (0 V, 1 V);
- *h* : 24 values (0, 1, 2, ··· , 23);
- *m* and *s* : 60 values (0, 1, 2, ··· , 59);
- *t* :10 values (0, 1, 2, ··· , 9)

Systems whose all inputs and outputs are digital signals are called …

DIGITAL SYSTEMS.

(Exercise)

Formally describe (pseudo-code) the relation between input and output signals of the chronometer.

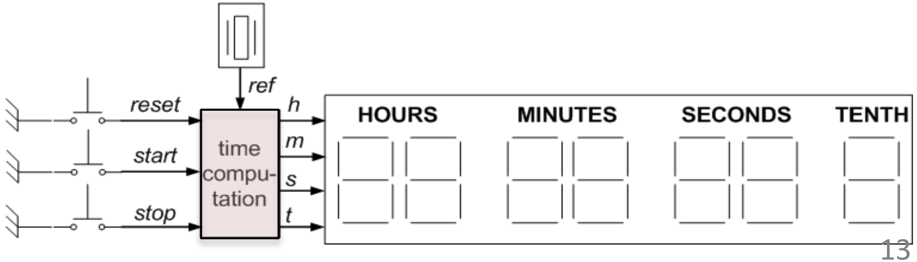
- Use a variable *ref_positive_edge* equal to TRUE when there is a positive edge (0 to 1 transition) on signal *ref*, and equal to FALSE in the contrary case.
- Assume that a procedure *update*(*h*, *m*, *s*, *t*), that adds a tenth of second to the elapsed time, has been previously defined.

Use instructions such as:

If … then … else …

While … loop …

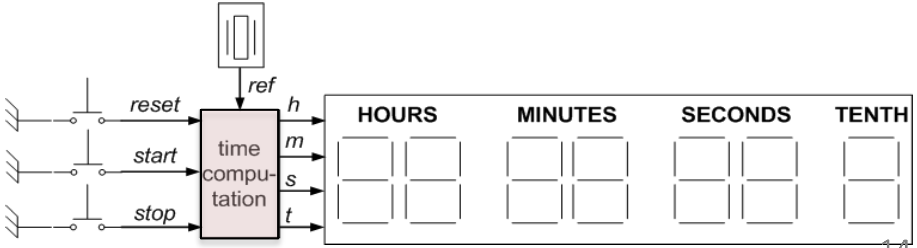
Loop …



UAB Universitat Autònoma de Barcelona

(Solution)

```
loop
    if reset = ON then h <= 0; m <= 0; s <= 0; t <= 0;
    elsif start = ON then
        while stop = OFF loop
            if ref_positive_edge = TRUE then

                update(h, m, s, t);

            end if;
        end loop;
    end if;
end loop;
```



SUMMARY

- Definition of digital signals and of digital systems.
- System considered as a “black box” with inputs and outputs , and a relation between inputs and outputs that defines its behavior.
- Examples of input –output specification using pseudo-instructions.

1.2 DIGITAL SYSTEM DESCRIPTION

Jean-Pierre Deschamps

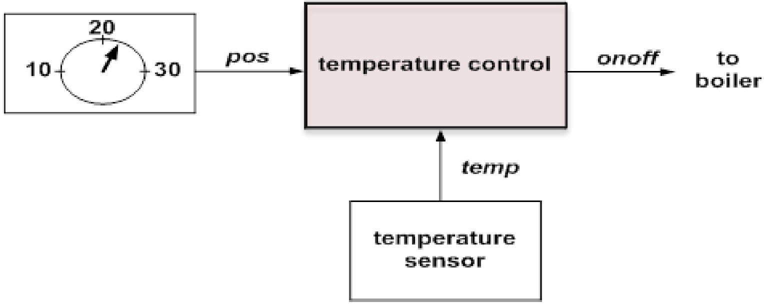
Universitat Autònoma de Barcelona

1. Functional description

Description of the relation between Inputs and Outputs, without information about the internal structure.

1.a Explicit functional description
1.b Implicit functional description: Algorithmic description

1.a Explicit functional description

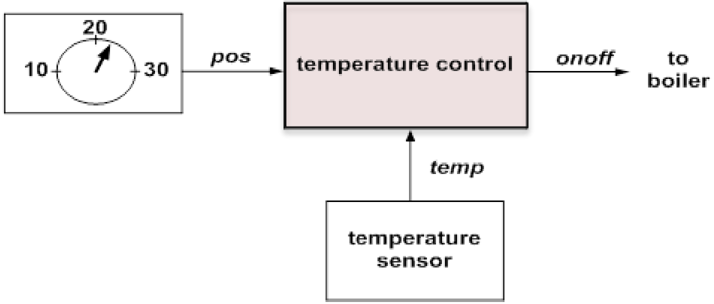


Ejemplo: temperature controller (simplified)

Assume that

- desired temperature (*pos*) = 20 degrees;
- measured temperature (*temp*) has been discretized and belongs to {0, 1, 2, ··· , 49, 50}.

1.a Explicit functional description



temp	*onoff*
0	ON
1	ON
…	…
18	ON
19	ON
20	DON'T CHANGE
21	OFF
22	OFF
…	…
49	OFF
50	OFF

1.b Implicit functional description (algorithm):

The preceding example might be described by the following algorithm:

temp	*onoff*
0	ON
1	ON
…	…
18	ON
19	ON
20	DON'T CHANGE
21	OFF
22	OFF
…	…
49	OFF
50	OFF

```
if temp < 20 then onoff <= ON;
elsif temp > 20 then onoff <= OFF;
end if;
```

1.b Implicit functional description (algorithm):

A second example: A 2-digit adder.

$X = x_1 x_0$ and $Y = y_1 y_0$ are 2-digit decimal numbers; their sum $X+Y$ is a 3-digit number $Z = z_2 z_1 z_0$.

“Pencil and paper algorithm:

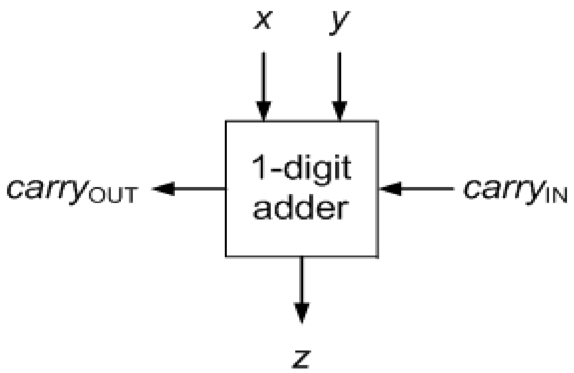

```
carry <= 0;
s0 <= x0 + y0 + carry;
if s0 > 9 then z0 <= s0 - 10; carry <= 1;
        else z0 <= s0; carry <= 0;
end if;

s1 <= x1 + y1 + carry;
if s1 > 9 then z1 <= s1 - 10; carry <= 1;
        else z1 <= s1; carry <= 0;
end if;

z2 <= carry;
```

2. Structural description

- Description of the internal system structure.
- Based on the use of previously defined digital subsytems, that is COMPONENTS.

2. Structural description: 4-digit adder

Assume that a 1-digit adder has been previously defined:



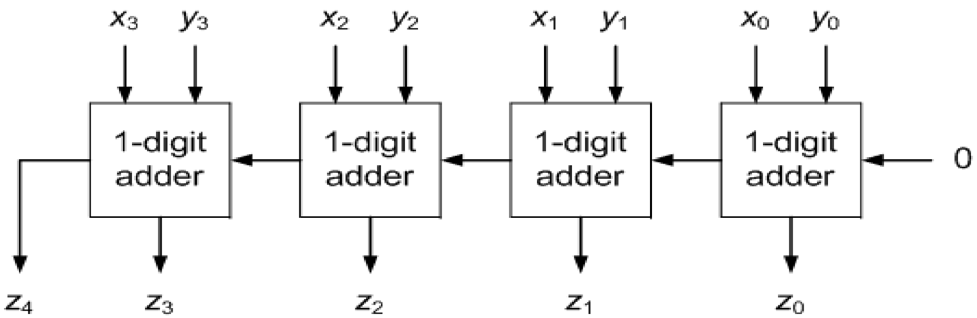
$carry_{IN}$, $carry_{OUT}$: carries $\in \{0, 1\}$,

x, y, z: decimal digits $\in \{0, 1, 2, \cdots, 9\}$,

function: $x + y + carry_{IN} = carry_{OUT}.10 + z$.

2. Structural description: 4-digit adder

The following system is a 4-digit adder made up of 1-digit adders.



It computes: $\boldsymbol{Z = X + Y}$

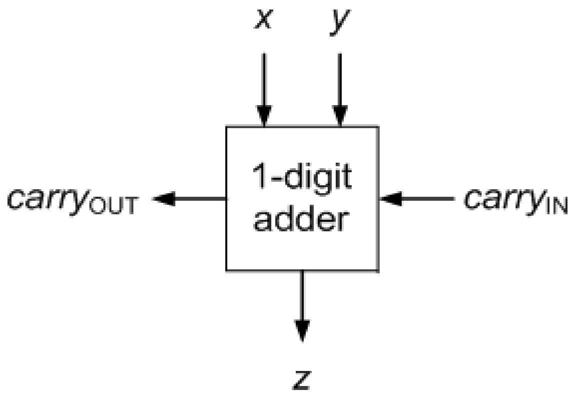
where $X = x_3\,x_2\,x_1\,x_0$ and $Y = y_3\,y_2\,y_1\,y_0$ are 4-digit numbers and $Z = z_4\,z_3\,z_2\,z_1\,z_0$ is a 5-digit number.

3. Hierarchical description

The previous example (4-digit adder) uses four available components, namely 1-digit adders. Every 1-digit adder might be defined by its function or by its structure.

Example (function):

$s <= x + y + carry_{IN}$;
if $s_1 > 9$ then $z <= s - 10$; $carry_{OUT} <= 1$;
else $z <= s$; $carry_{OUT} <= 0$;
end if;



This is a **2-level** hierarchical description.

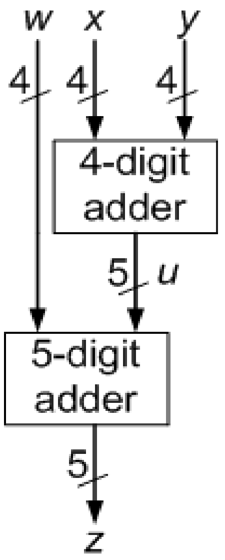
3. Hierarchical description

Example of **3-level** hierarchical description.

The following system (structural description) computes

$$z = w + x + y$$

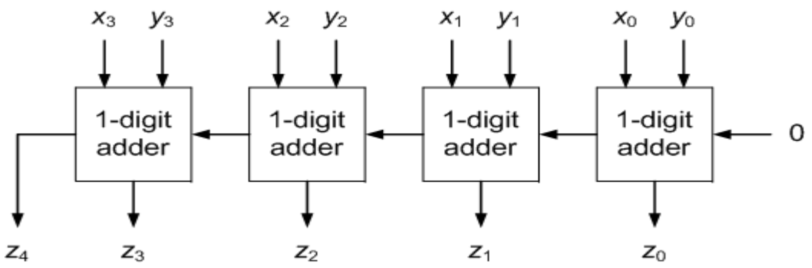
where w, x and y 4-digit numbers and
z is a 5-digit number (9999 + 9999 + 9999 = 29,997).



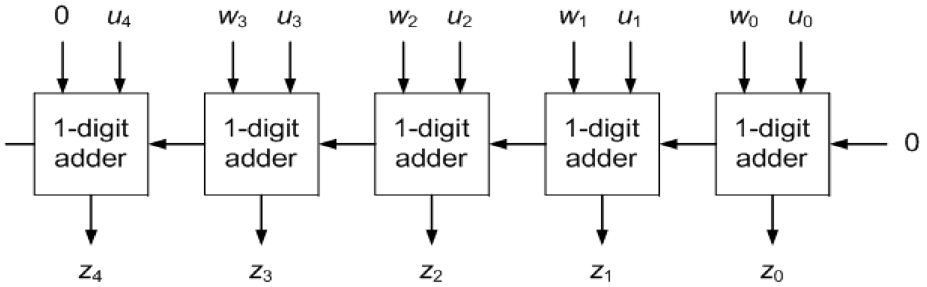
1st HIERARCHY LEVEL

3. Hierarchical description

4-digit decimal adder



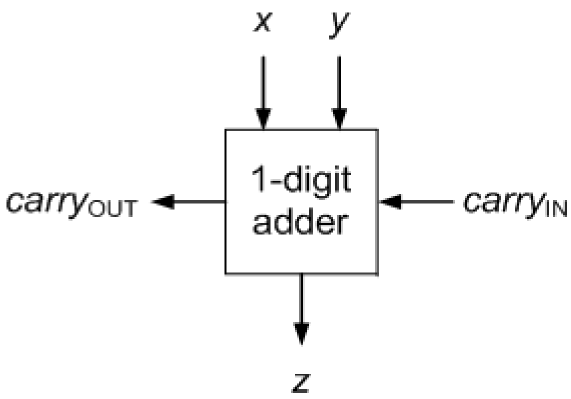
5-digit decimal adder



2nd HIERARCHY LEVEL

3. Hierarchical description

Both 4-digit and 5-digit adders are made up of 1-digit adders that might be defined by their function:



s <= x + y + $carry_{IN}$;
if s_1 > 9 then z <= s - 10; $carry_{OUT}$ <= 1;
else z <= s; $carry_{OUT}$ <= 0;
end if;

3rd HIERARCHY LEVEL

3. Hierarchical description

To summarize, an hierarchical description:

- Is a set of interconnected blocks.
- Every block, in turn, is described by its function or by a set of interconnected blocks, and so on.
- The final blocks correspond to available components defined by their function.

Question

Consider an intermediate level (= different from the last one) of a hierarchical description. Check the correct assertion(s):

1. All blocks MUST be described by their structure.
2. Some blocks CAN be described by their structure.
3. Some blocks CAN be described by their function.
4. All blocks MUST be described by their function.

SUMMARY

- Functional description.
- Structural description.
- Hierarchical description.

1.3 DIGITAL ELECTRONIC SYSTEMS

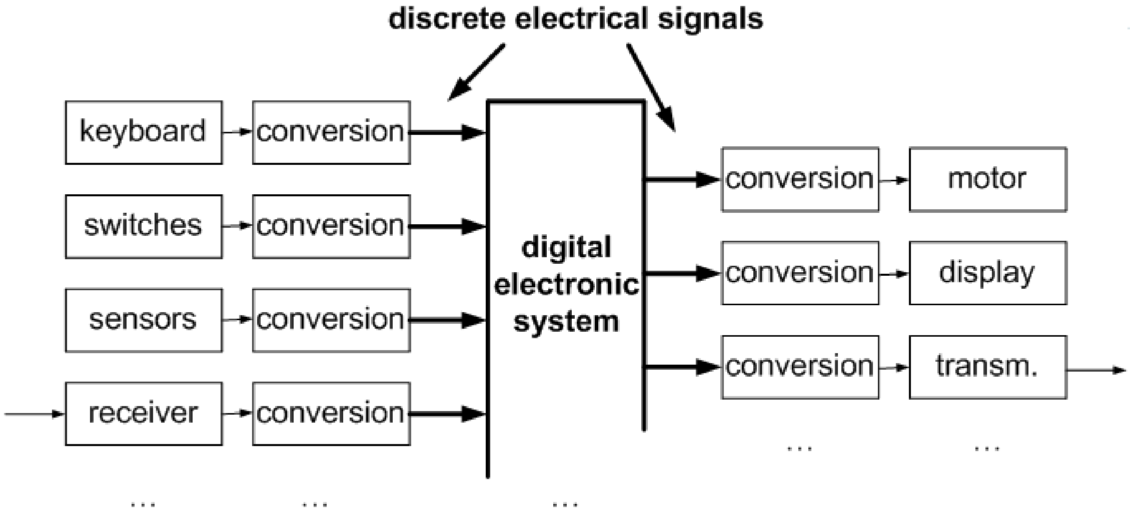
Jean-Pierre Deschamps

Universitat Autònoma de Barcelona

1. Previous observation

Real digital systems include

- input devices (sensors, keyboards, microphones, cameras, ···),
- output devices (actuators, displays, loudspeakers, motors, ···),
- input converters that translate the input device information to discrete electrical signals,
- output converters that translate discrete electrical data to signals able to control the output devices,
- a digital electronic circuit (the kernel of the system) that generates output data in function of input data.



This is a course about

Digital Electronic Systems

Inputs and outputs of a Digital Electronic System are binary encoded data.

Examples:

- numbers (binary code),
- alphanumeric data (ASCII codes),
- others.

2. Digital components

2.1 Binary codification

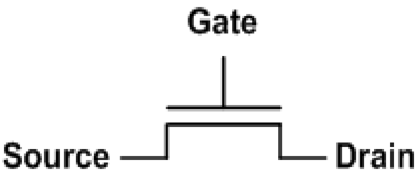
- Define two voltages V_L y V_H .
- 0 is represented by V_L, 1 is represented by V_H.
  - ✓ Example: $V_L = 0$ volt, $V_H = 1$ volt.

2.2 MOS transistors

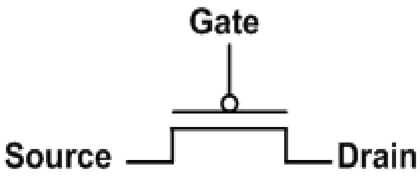
Most circuits are made up of MOS transistors.

Mos transistor: 3-terminal device (source, gate, drain).

Two types:

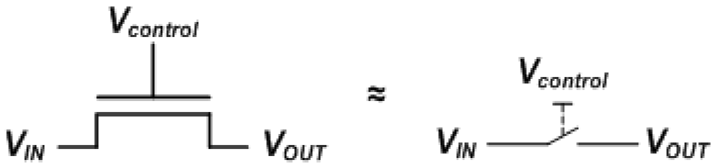


*n*MOS



*p*MOS

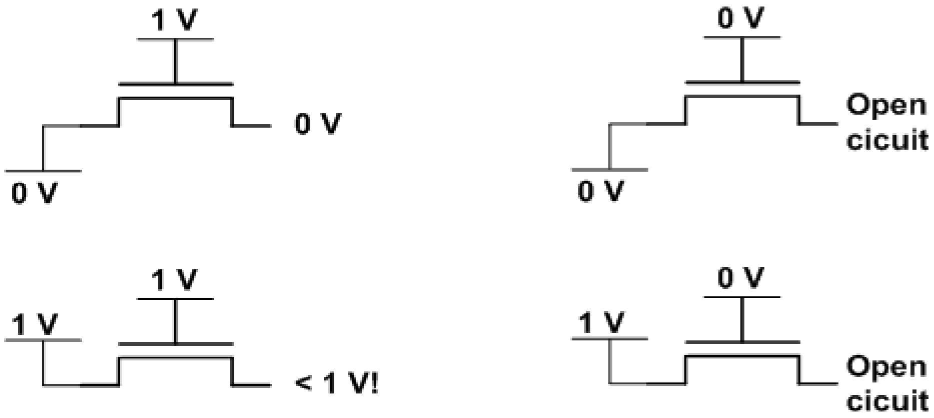
2.3 MOS transistors used as switches



… but:

not a good switch for any value of V_{IN}.

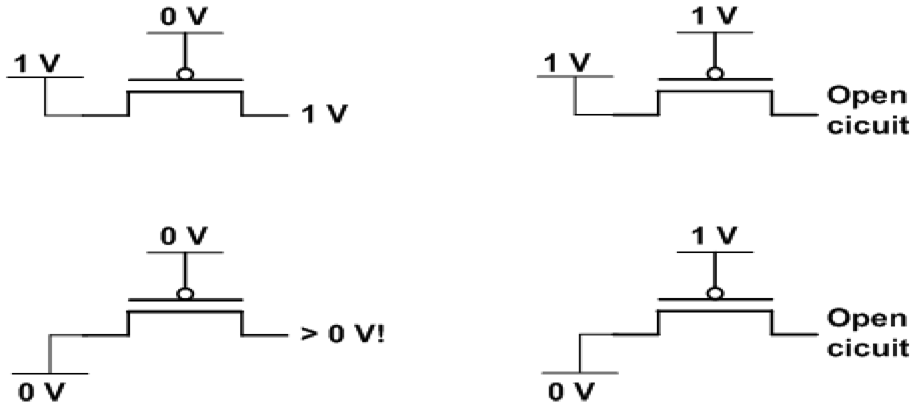
2.3.1 *n*MOS switch.



Conclusion: *n*MOS transistor is

- a "good" switch for transmitting V_L (0 V), but …
- a "not so good" switch for transmitting V_H (1 V).

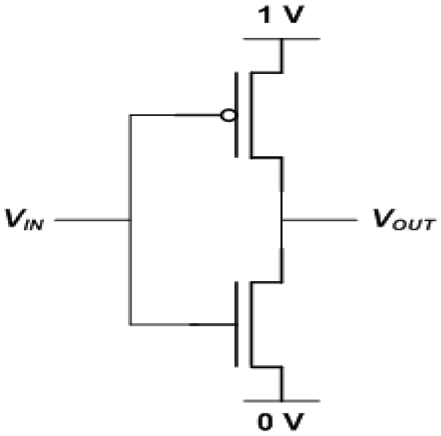
2.3.2 *p*MOS switch.

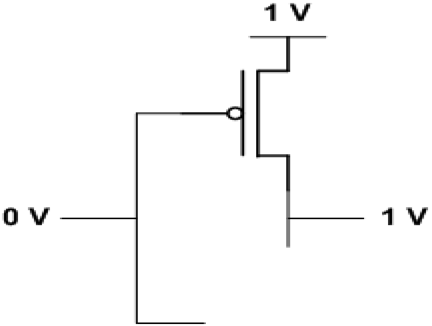


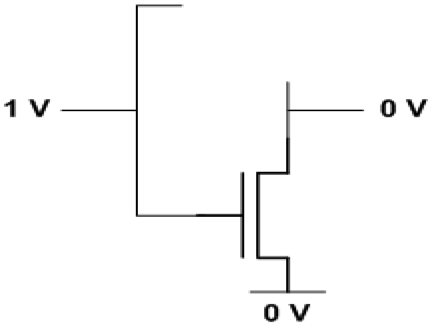
Conclusion: *p*MOS transistor is

- a “good” switch for transmitting V_H (1 V), but …
- a “not so good” switch for transmitting V_L (0 V).

2.4 CMOS inverter



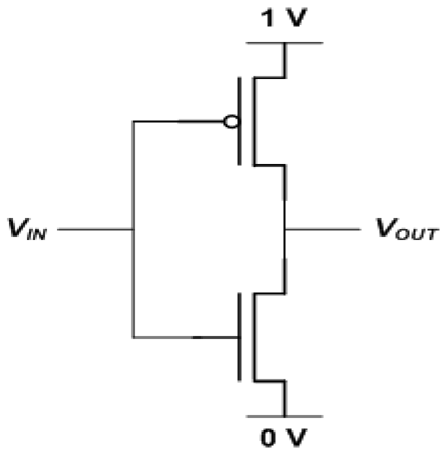




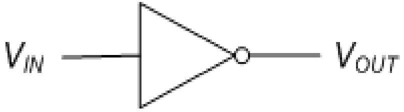
V_{IN} = 0 V : *p*MOS transmits 1 V

V_{IN} = 1 V : *n*MOS transmits 0 V

2.4 CMOS inverter



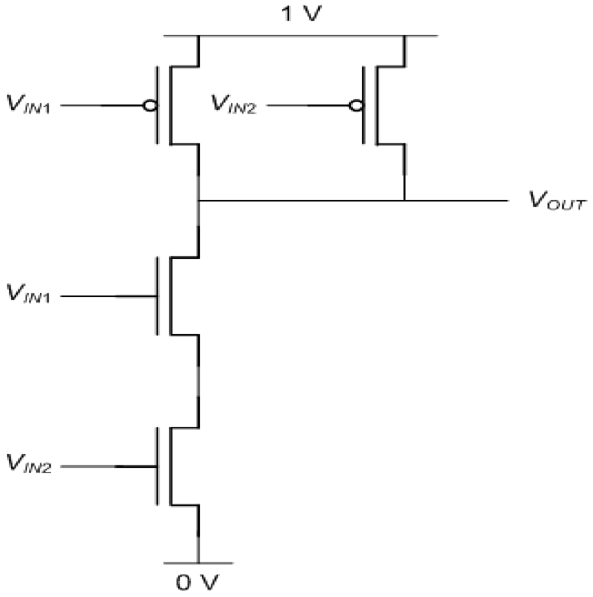
V_{IN}	V_{OUT}
0 v.	1 v.
1 v.	0 v.



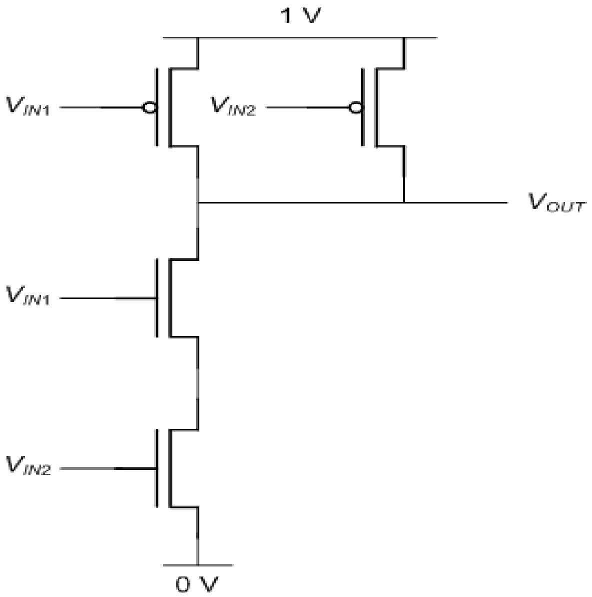
INV

2.5 NAND gate

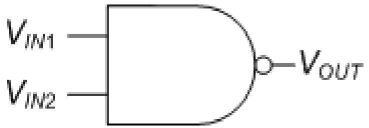
- (V_{IN1} = 1V) AND (V_{IN2} = 1V): V_{OUT} = 0V (both serially connected *n*MOS switches transmit 0V);
- (V_{IN1} = 0V) OR (V_{IN2} = 0V): V_{OUT} = 1V (one or both in-parallel connected *p*MOS switches transmit 1V).



2.5 NAND gate



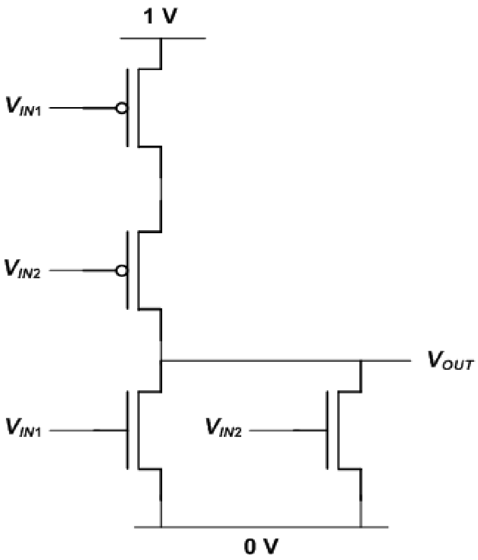
V_{IN1}	V_{IN2}	V_{OUT}
0 v.	0 v.	1 v.
0 v.	1 v.	1 v.
1 v.	0 v.	1 v.
1 v.	1 v.	0 v.



NAND2

UAB Universitat Autònoma de Barcelona

Question 1



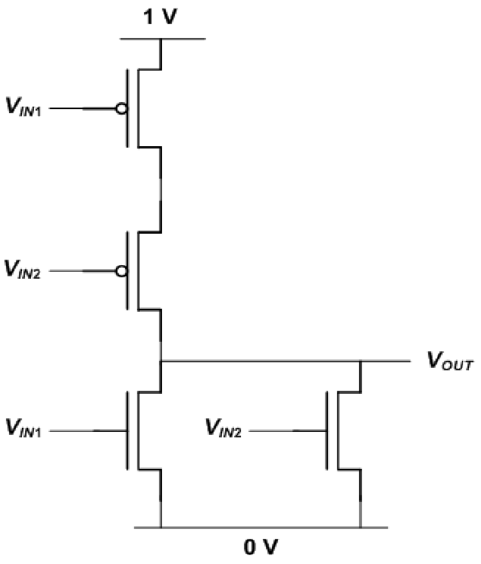
For each combination of values of V_{IN1} and V_{IN2} check the corresponding box if $V_{OUT} = 1$:

1. $V_{IN1} = 0$, $V_{IN2} = 0$,
2. $V_{IN1} = 0$, $V_{IN2} = 1$,
3. $V_{IN1} = 1$, $V_{IN2} = 0$,
4. $V_{IN1} = 1$, $V_{IN2} = 1$.

Universitat Autònoma de Barcelona

Question 2

Check the box that reflects the operation of the circuit



1.

V_{IN1}	V_{IN2}	V_{OUT}
0	0	1
0	1	0
1	0	0
1	1	1

2.

V_{IN1}	V_{IN2}	V_{OUT}
0	0	0
0	1	1
1	0	1
1	1	0

3.

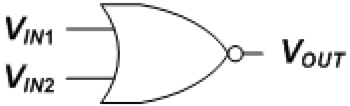
V_{IN1}	V_{IN2}	V_{OUT}
0	0	0
0	1	1
1	0	1
1	1	0

4.

V_{IN1}	V_{IN2}	V_{OUT}
0	0	1
0	1	0
1	0	0
1	1	0

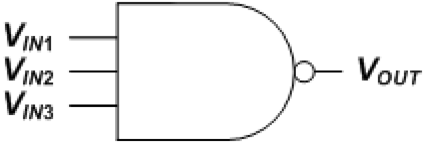
2.6 Other components

NOR2



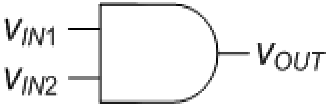
V_{OUT} = 0 iff (if and only if) V_{IN1} = 1 OR V_{IN2} = 1

NAND3

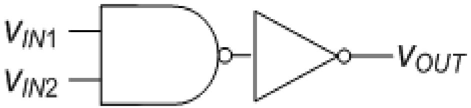


V_{OUT} = 0 iff $V_{IN1} = V_{IN2} = V_{IN3} = 1$

AND2

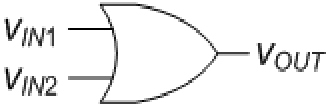


V_{OUT} = 1 iff $V_{IN1} = V_{IN2} = 1$

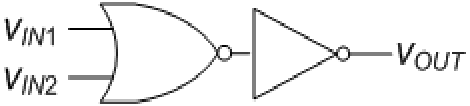


2.6 Other components

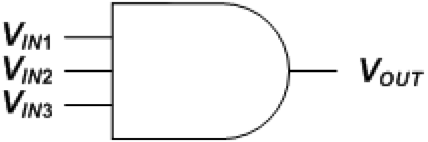
OR2



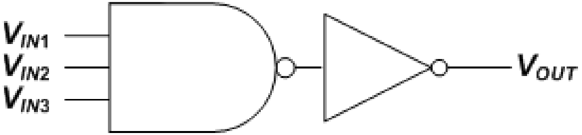
V_{OUT} = 1 iff V_{IN1} = 1 OR V_{IN2} = 1.



AND3



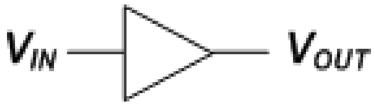
V_{OUT} = 1 iff V_{IN1} = V_{IN2} = V_{IN3} = 1.



2.6 Other components

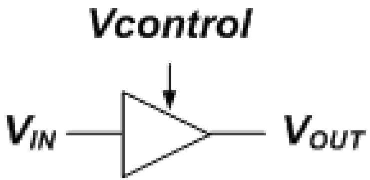
UAB Universitat Autònoma de Barcelona

BUFFER



$V_{OUT} = V_{IN}$

3-STATE BUFFER



$V_{control}$ = 1 V: $V_{OUT} = V_{IN}$;

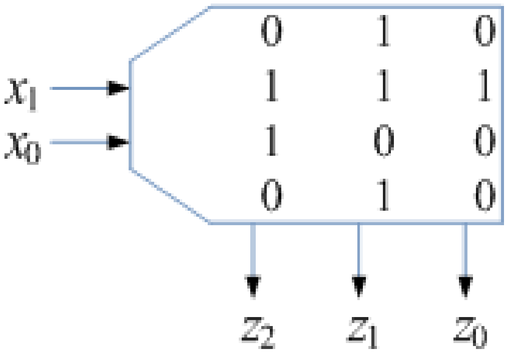
$V_{control}$ = 0 V: V_{OUT} : open circuit

… other components such as multiplexers, encoders, decoders, latches, flip flops, etc. , will be defined later on.

UAB Universitat Autònoma de Barcelona

ROM (Read Only Memory)

Example: 12-bit ROM
(4 words, 3 bits per word,
address: 2 bits)



$x_1\ x_0$	$z_2\ z_1\ z_0$
0 0	0 1 0
0 1	1 1 1
1 0	1 0 0
1 1	0 1 0

General case: ($m \cdot 2^n$)-bit ROM
(2^n words, m bits per word,
address: n bits)

3. Synthesis of Electronic Digital Systems

CENTRAL TOPIC OF THIS COURSE

3. Synthesis of Electronic Digital Systems

Digital electronic system specification

E.g.: Design a circuit that controls the working of a boiler according to ….

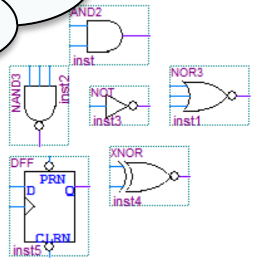
CENTRAL TOPIC OF THIS COURSE

3. Synthesis of Electronic Digital Systems

Digital electronic system specification

E.g.: Design a circuit that controls the working of a boiler according to ….

Catalog of components



CENTRAL TOPIC OF THIS COURSE

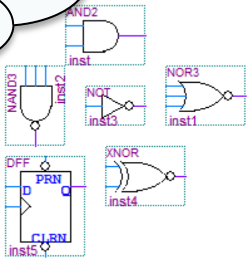
Universitat Autònoma de Barcelona

3. Synthesis of Electronic Digital Systems

Digital electronic system specification

E.g.: Design a circuit that controls the working of a boiler according to ….

Catalog of components



Generate an hierarchical description whose final blocks are components belonging to the catalog.

CENTRAL TOPIC OF THIS COURSE

SUMMARY

- What do we understand by “digital electronic system”?
- Binary encoding (1s and 0s represented by high and low voltage values).
- Catalog of components.
- Goal of the synthesis of digital electronic systems.