

*University of Basque Country (Spain)*

# *Implementation of flexible control systems*

*J.M. López-Guede*

*Pitesti, Romania  
30 June 2016*



- 1. Motivation**
- 2. Basic concepts**
- 3. Working cycle with VHDL**
- 4. Combinational systems**
- 5. Sequential systems**
- 6. Challenge**



# 1. Motivation



**Reference: Florin's presentation...**



# 2. Basic concepts (I)



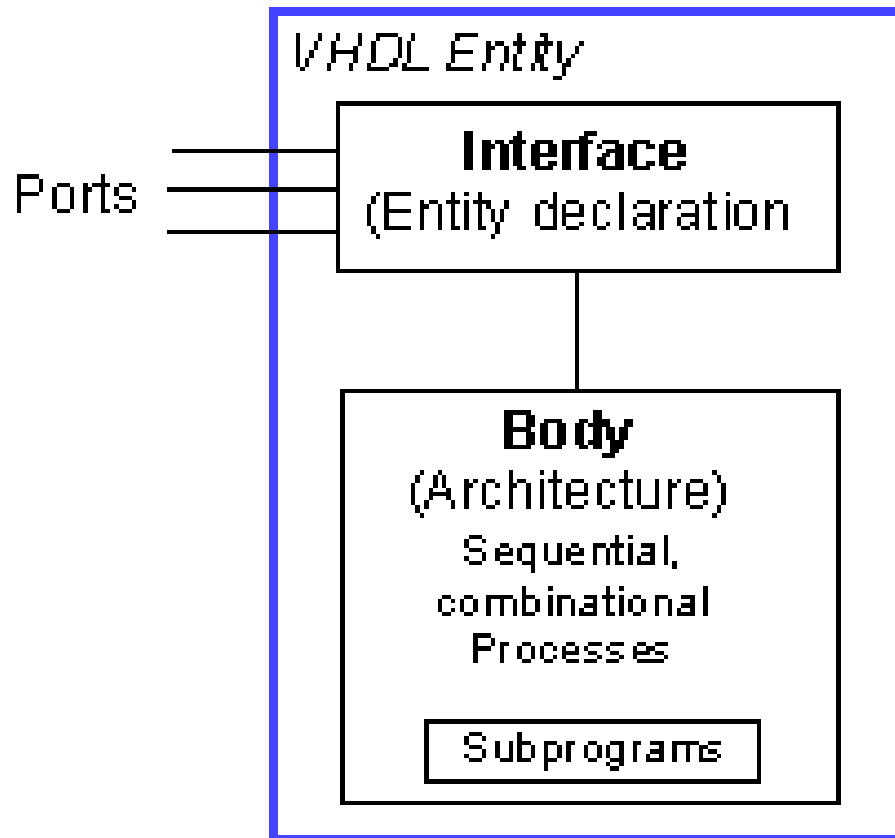
- FPGAs are flexible devices
- We can use Graphic HDL, VHDL or Verilog.
- VHDL: VHSIC Hardware Description Language



## 2. Basic concepts (II)



- Structure of a VHDL description



## 2. Basic concepts (III)



- **Entity**

```
entity NAME_OF_ENTITY is
    port (signal_names: mode type;
          signal_names: mode type;
          :
          signal_names: mode type);
end [NAME_OF_ENTITY];
```



## 2. Basic concepts (IV)



- **Architecture**

architecture architecture\_name of NAME\_OF\_ENTITY is

-- Declarations

:

begin

-- Statements

:

end architecture\_name;



## 2. Basic concepts (V)



- Process

process (sensitivity\_list)  
begin

list of **sequential statements**

end process;





# 2. Basic concepts (VI)

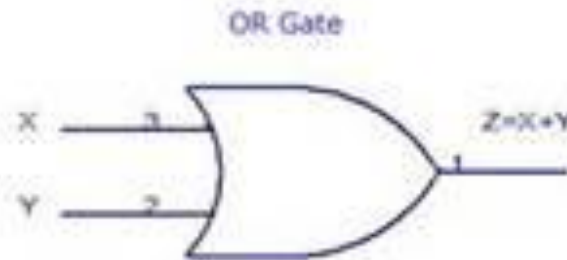


## • Example: OR gate

```
ENTITY my_or IS
  PORT (op1, op2      : IN  bit;
        result        : OUT bit);
END my_or;
```

```
ARCHITECTURE example OF my_or IS
BEGIN
  result <= op1 or op2;
END maxpld;
```

2 Input OR Gate



TRUTH TABLE

INPUTS		OUTPUT
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

# 3. Working cycle with VHDL



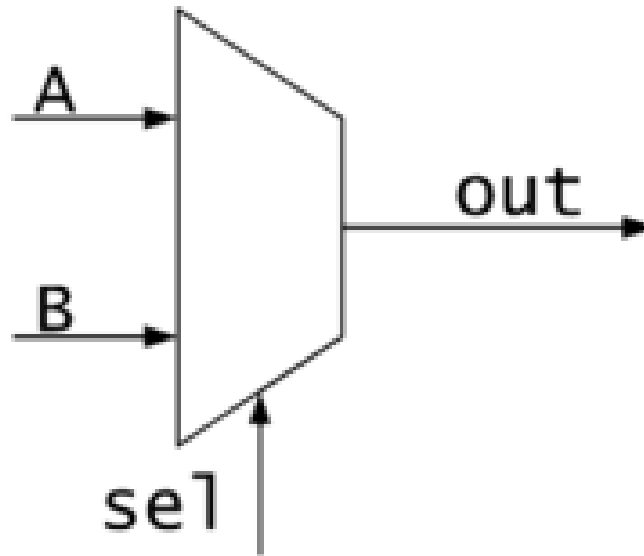
1. Describe the behavior using VHDL
2. Compile the description
3. Define tests
4. Simulations
5. Assign nodes
6. Download the behavior into the FPGA device



# 4. Combinational systems (I)



- Description of the behavior:



# 4. Combinational systems (II)



- **Entity:**

```
entity Mux21 is port (a: in bit;  
                    b: in bit;  
                    sel: in bit;  
                    output: out bit);  
end Mux21;
```



# 4. Combinational systems (III)



- **Architecture:**

architecture example of Mux21 is

```
begin
```

```
    process (a, b, sel)
```

```
    begin
```

```
        if (sel= '0') then
```

```
            out<= a;
```

```
        else
```

```
            out<= b;
```

```
        end if;
```

```
    end process;
```

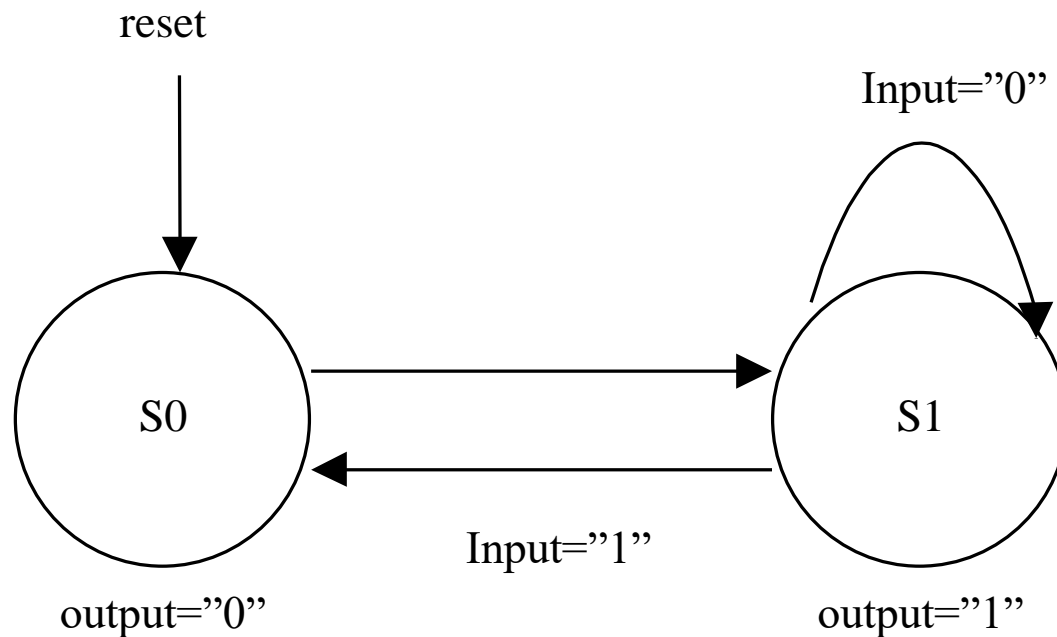
```
end example;
```



# 5. Sequential systems (I)



- **Description of the behavior:**



# 5. Sequential systems (II)



- **Entity:**

```
entity simple_machine is
port (clk, input, reset : in bit;
      output : out bit);
end simple_machine ;
```



# 5. Sequential systems (III)



- **Architecture:**

```
architecture simple of simple_machine is
    type state_type is (s0, s1);
    signal state: state_type;
begin
    process (clk, reset)
    begin
        :
        :
    end process;
    output <= '1' when state = s1 else '0';
end simple;
```





# 5. Sequential systems (IV)



```
if reset = '1' then
    state <= s0;
elsif (clk'event and clk = '1') then
    case state is
        when s0 =>
            state <= s1;
        when s1 =>
            if input = '1' then
                state <= s0;
            else
                state <= s1;
            end if;
        end case;
    end if;
```



# 6. Challenge (I)

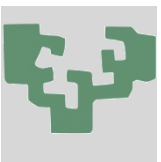
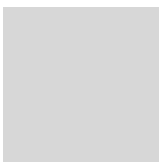
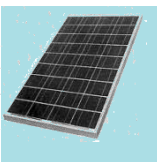
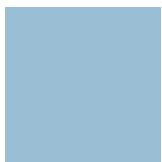
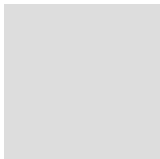
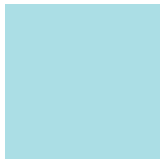


- **Controlling the gate of a garage:**
  - **Actuators:**
    - Reset
    - M: activation push button
    - MA: activate motor to lift the gate
    - MB: activate motor to lower the gate
    - F: brake
  - **Sensors:**
    - FC1: door totally open
    - FC2: door totally closed



*University of Basque Country (Spain)*

**Thank You!**  
**Gracias**



**Jose Manuel Lopez-Guede**  
([jm.lopez@ehu.es](mailto:jm.lopez@ehu.es))