Implementation of flexible control systems

Pitesti, Romania
30 June 2016
Outline

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**

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

```vhdl
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 example;
```

<table>
<thead>
<tr>
<th>INPUTS</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 1</td>
<td>1</td>
</tr>
<tr>
<td>1 0</td>
<td>1</td>
</tr>
<tr>
<td>1 1</td>
<td>1</td>
</tr>
</tbody>
</table>
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:**

```
A → out
B → sel
```
4. Combinational systems (II)

- **Entity:**

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

![Diagram](image-url)
5. Sequential systems (II)

• Entity:

```vhdl
entity simple_machine is
port(clk, input, reset : in bit;
     output : out bit);
end simple_machine;
```
• 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;
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
Thank You!

Gracias

Jose Manuel Lopez-Guede
(jm.lopez@ehu.es)