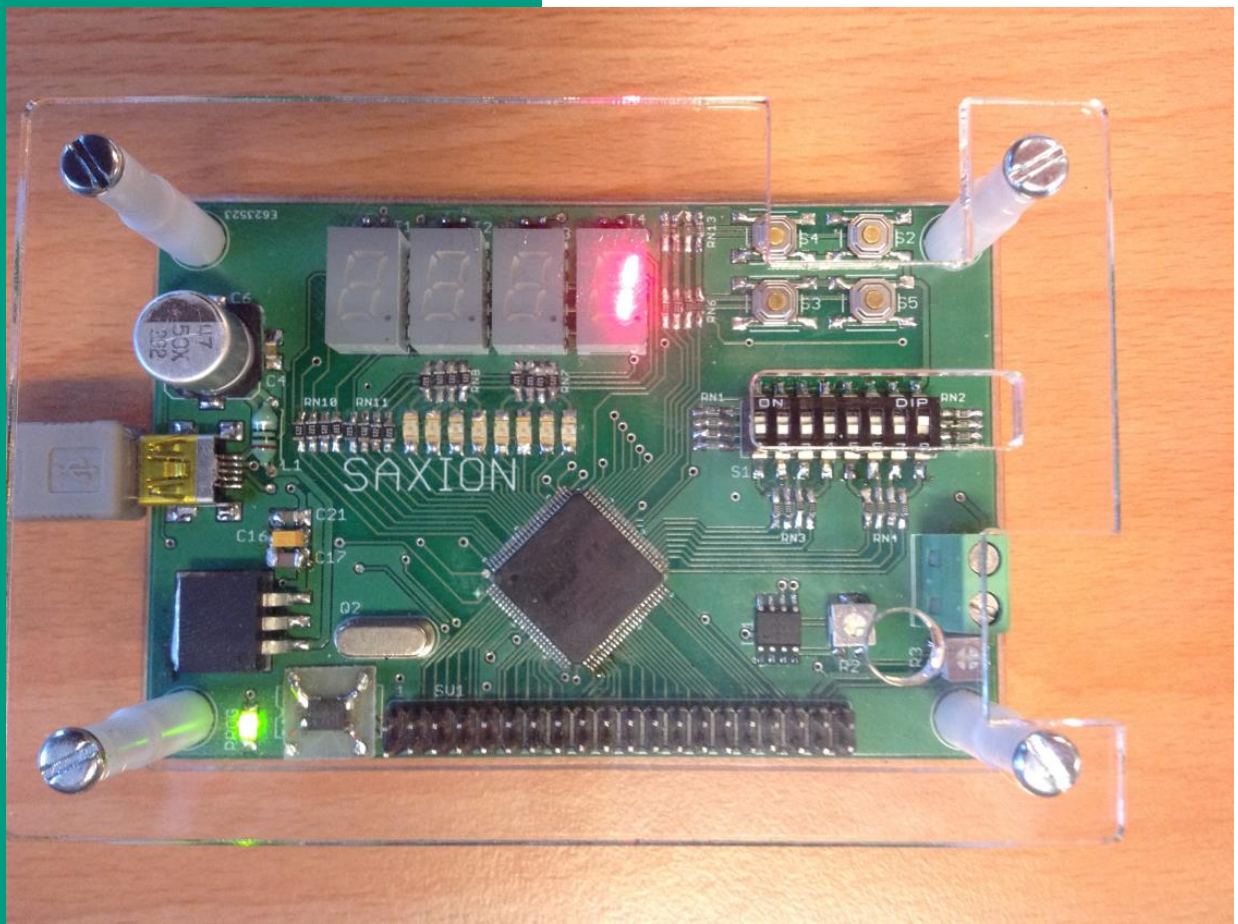
Very High Speed Integrated Circuits Hardware Description Language

Lesson 2
VHDL
Synthesis



SAXION
UNIVERSITY OF APPLIED SCIENCES

SAXION UNIVERSITY OF APPLIED SCIENCES

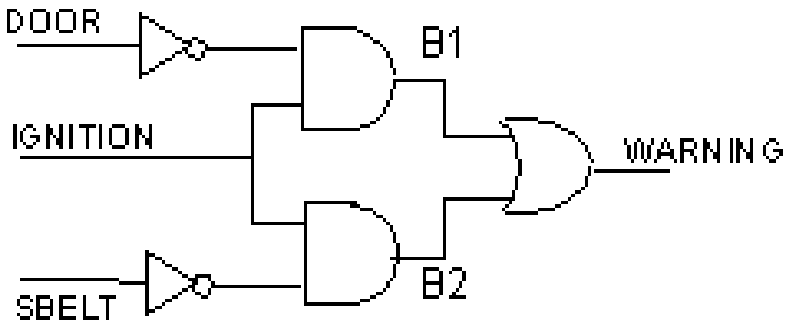
Lesson 2 outline

- Levels of design (synthesis)
- Behavioural
- Structural
- Register Level
- Timing (chapter4)
- Seven segment decoder (chapter 5)
- Questions.

Behavioral level

```
behavior OF ilatch IS
    BEGIN
  PROCESS (A, B)
      BEGIN
    IF A = '0' THEN
          Output1 <= '0';
          Output2 <= '0';
         ELSE
         IF B = '1' THEN
               Output1 <= '1';
               Output2 <= '1';
              ELSE
          Output1 <= '0';
      END IF;
        END IF;
  END PROCESS;
END behavior;
```

Structural level




```
architecture structural of BUZZER is
        component AND2
                port (in1, in2: in std_logic;
                        out1: out std_logic);
        end component;
        component OR2
                port (in1, in2: in std_logic;
                        out1: out std_logic);
        end component;
        component NOT1
                port (in1: in std_logic;
                        out1: out std_logic);
        end component;

signal DOOR_NOT, SBELT_NOT, B1, B2: std_logic;
        begin
U0: NOT1 port map (DOOR, DOOR_NOT);
U1: NOT1 port map (SBELT, SBELT_NOT);
U2: AND2 port map (IGNITION, DOOR_NOT, B1);
U3: AND2 port map (IGNITION, SBELT_NOT, B2);
U4: OR2  port map (B1, B2, WARNING)
    end structural;
```

RTL level design

```
                BEHAV_DFF of DFF_CLEAR is
               Begin
DFF_PROCESS:   process (CLK, CLEAR)
                begin
                if (CLEAR = '1') then
                                Q <= '0';
                elsif (CLK'event and CLK = '1') then
                                Q <= D;
                end if;
                    end process;
                end BEHAV_DFF;
```

level descriptions

```
ARCHITECTURE behavior OF gate_network IS
BEGIN
        X <= A AND NOT(B OR C) AND (D(1) XOR D(2)) after 10 ns;
PROCESS (A,B,C,D)
        BEGIN
         Y <= A AND NOT(B OR C) AND (D(1) XOR D(2))after 20 ns;
        END PROCESS;
END behavior;
```

Explicit processes, The Sensitivity list

```
compute_xor: process (b,c)
 begin
 a<=b xor c;
 end process
```

```
compute_xor: process (c)
 begin
 a<=b xor c;
 end process;
```

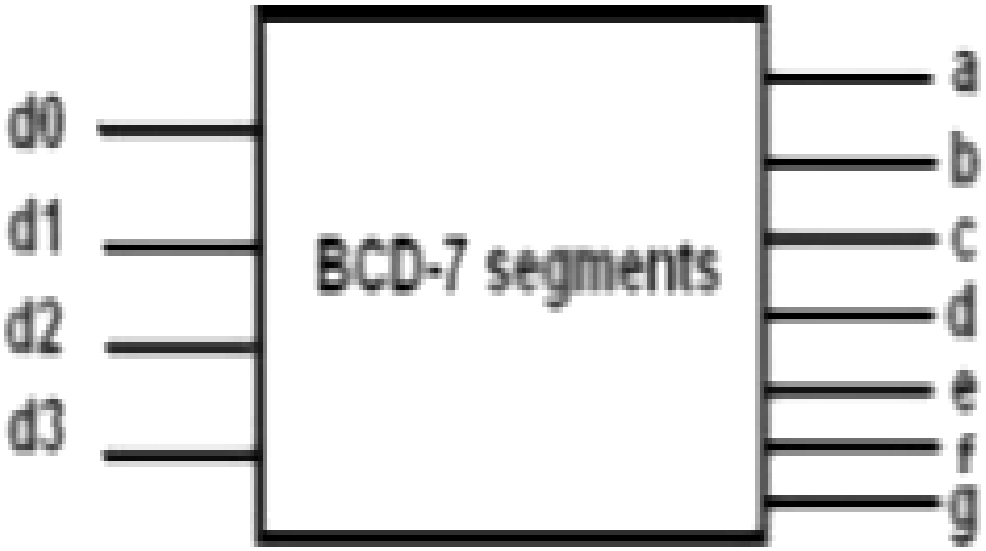
Implicit processes

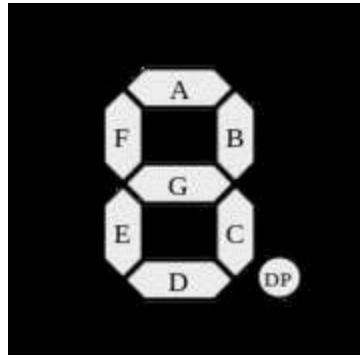
In implicit processes you do not see the word "process". Each assignment is a process on itself

Example:

```
begin
 a<=b xor c;
 d<=b and c;
 end archx;
```

7 segments decoder





7 segments decoder

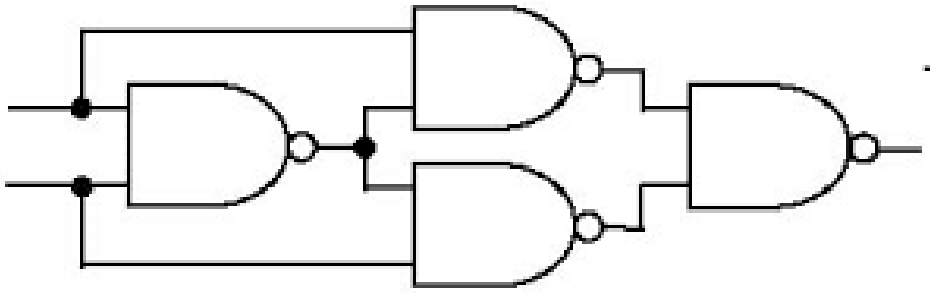
Architecture with case statement

```
ARCHITECTURE behaviour OF sevensegm
 PROCESS(msd)
    BEGIN
     CASE msd IS
        WHEN “0000” =>
            msd_7seg <= “1111110”;
         WHEN “0000” =>
              msd_7seg <=“0110000”;
              etc
```

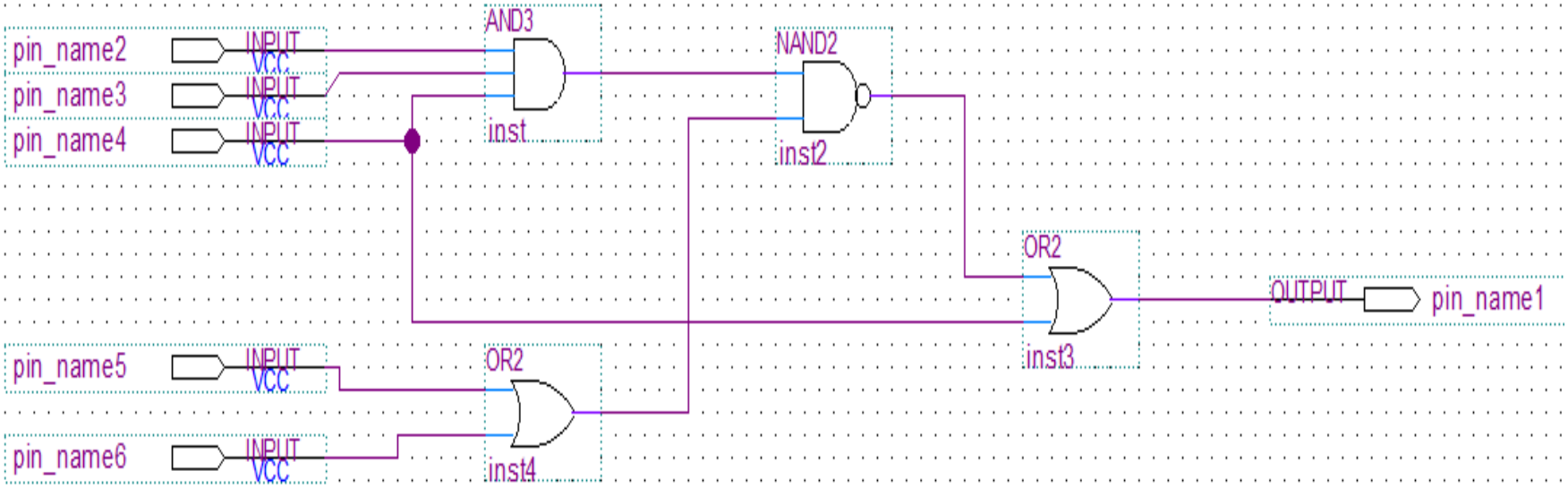
Assignment 1, 2

Write a structural architecture of the next schematics:

Exclusive OR (XOR)



A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0



Assignment 3

Draw the schematic of the vhdl description: See also next page

```
architecture structural_2 of multiplexer_4_to_1_st is
component
 NOT1 port( in1 : in std_logic; out1 : out std_logic);
end component;

component
  AND3 port( in1, in2, in3 : in std_logic; out1 : out std_logic);
end component;

component
 OR4 port( in1, in2, in3, in4 : in std_logic; out1 : out std_logic);
end component;
```

Assignment 4

Write an alternative in behavioural VHDL

```
signal S_n : std_logic_vector(0 to 1);
signal N : std_logic_vector(0 to 3);

begin
 g0: NOT1 port map (S(0), S_n(0));
 g1: NOT1 port map (S(1), S_n(1));
 g2: AND3 port map (S_n(1), S_n(0), D(0), N(0));
 g3: AND3 port map (S_n(1), S(0), D(1), N(1));
 g4: AND3 port map (S(1), S_n(0), D(2), N(2));
 g5: AND3 port map (S(1), S(0), D(3), N(3));
 g6: OR4 port map (N(0), N(1), N(2), N(3), Y);
end structural_2;
```

Exercises (week 2)

- Do the self test of chapter 3, 4, 5
- Read chapter 3, 4, 5
- Prepare the practical work of chapter 3 ,4 and 5

Submit before week 4 in Black Board:
practical work and the self tests in one file.
.pdf or word