Exercices sur le VHDL

PROBLEME #1. Identifiez l’erreur dans les codes VHDL suivants en examinant le code et en regardant le message d’erreur fourni par le logiciel Quartus.

1. Code #1

```
PROCESS (clk)
BEGIN
    a <= bouton OR restart;

    IF clk'EVENT AND clk = '1' THEN
        IF s_count = 12 THEN
            s_count <= (OTHERS => '0');
            a <= '1';
        ELSE
            s_count <= s_count + 1;
        END IF;
    END IF;
END PROCESS;
```

```
Error (10818): Can't infer register for "a" at multiplexeur.vhd(25) because it does not hold its value outside the clock edge
Error (10822): HDL error at multiplexeur.vhd(25): couldn't implement registers for assignments on this clock edge
```

2. Code #2

```
PROCESS (clk)
BEGIN
    IF clk'EVENT AND clk = '1' THEN
        IF s_count = 49999999 THEN
            s_count <= (OTHERS => '0')
        ELSE
            s_count <= s_count + 1;
        END IF;
    END IF;
END PROCESS;
```

```
Error (10500): VHDL syntax error at multiplexeur.vhd(27) near text "ELSE";  expecting ";"
```

3. Code #3

```
ENTITY multiplexeur IS
          PORT (
                  clk    : IN STD_LOGIC;
                  rst_an : IN STD_LOGIC;
                  count  : OUT STD_LOGIC_VECTOR(3 DOWNTO 0)
            );
END multiplexeur;

ARCHITECTURE rtl OF multiplexeur IS

BEGIN

PROCESS (clk, rst_an)
BEGIN
    IF rst_an = '0' THEN
        count <= (OTHERS => '0');
    ELSIF clk'EVENT AND clk = '1' THEN
        count <= count + 1;
    END IF;
END PROCESS;

END;
```

Error (10309): VHDL Interface Declaration error in multiplexeur.vhd(22): interface object "count" of mode out cannot be read. Change object mode to buffer.

4. Code #4

```
PROCESS (clk, rst_an)
BEGIN
    IF rst_an = '0' THEN
        s_count <= (OTHERS => '0');
    ELSIF clk'EVENT AND clk = '1' THEN
        s_count <= s_count + 1;
    ELSIF a = '1' THEN
        s_count <= (OTHERS => '1');
    END IF;
END PROCESS;
```

Error (10818): Can't infer register for "s_count[0]" at multiplexeur.vhd(22) because it does not hold its value outside the clock edge
Info (10041): Inferred latch for "s_count[0]" at multiplexeur.vhd(20)
Error (10818): Can't infer register for "s_count[1]" at multiplexeur.vhd(22) because it does not hold its value outside the clock edge
Info (10041): Inferred latch for "s_count[1]" at multiplexeur.vhd(20)
Error (10818): Can't infer register for "s_count[2]" at multiplexeur.vhd(22) because it does not hold its value outside the clock edge
Info (10041): Inferred latch for "s_count[2]" at multiplexeur.vhd(20)
Error (10818): Can't infer register for "s_count[3]" at multiplexeur.vhd(22) because it does not hold its value outside the clock edge
Info (10041): Inferred latch for "s_count[3]" at multiplexeur.vhd(20)

5. Code #5

```
PROCESS (sel)
BEGIN
    CASE sel IS
        WHEN "00" =>
            c <= "01";
        WHEN "10" =>
            c <= "11";
        WHEN "11" =>
            c <= "00";
    END CASE;
END PROCESS;
```

```
Error (10313): VHDL Case Statement error at multiplexeur.vhd(17): Case Statement choices must cover all possible values of expression
```

6. Code #6

```
PROCESS (a, b)
BEGIN
      IF sel='0' THEN
            c <= a;
      ELSE
          c <= b;
      END IF;
END PROCESS;
```

```
Warning (10492): VHDL Process Statement warning at multiplexeur.vhd(19): signal "sel" is read inside the Process Statement but isn't in the Process Statement's sensitivity list
```

7. Code #7

```
PROCESS (a, b, sel)
BEGIN
      IF sel='0' THEN
            c <= a;
      END IF;
END PROCESS;
```

```
Warning (10631): VHDL Process Statement warning at multiplexeur.vhd(17): inferring latch(es) for signal or variable "c", which holds its previous value in one or more paths through the process
```

PROBLEME #2. Identifiez ce que fait chacun des blocs VHDL suivants.

1. Code #1

```
PROCESS (a)
BEGIN
      IF a > "001001" THEN
            b <= a + "000110";
      ELSE
            b <= a;
      END IF;
END PROCESS;
```

2. Code #2

```
PROCESS (s_clk_1hz)
BEGIN
    IF s_clk_1hz'EVENT AND s_clk_1hz = '1' THEN
        IF bouton = '1' THEN
             s_count <= "0001";
        ELSE
             s_count <= s_count(0) & s_count(3 DOWNTO 1);
        END IF;
    END IF;
END PROCESS;
```

3. Code #3

```
PROCESS (clk)
BEGIN
    IF clk'EVENT AND clk = '1' THEN
        datain1 <= datain;
        datain2 <= datain1;
        datain3 <= datain2;
        dataout <= datain3;
    END IF;
END PROCESS;
```

4. Code #4

```
PROCESS (clk)
BEGIN
    IF clk'EVENT AND clk = '1' THEN
        s_datain1 <= datain;
        s_datain2 <= s_datain1;
        s_datain3 <= s_datain2;
        s_datain4 <= s_datain3;
    END IF;
END PROCESS;

s_somme <= ("00" & s_datain1) + ("00" & s_datain2) + ("00" & s_datain3) +
("00" & s_datain4);
dataout <= s_somme(5 DOWNTO 2);
```