Sequential Statement

In a process, the execution of sequential statements will be initiated when there is a change in the signals listed in the **sensitivity list**. In general, the execution of statements in the process body will be carried out **continuously** until the end of the process **sequentially**. There are **two types** of sequential statements that will be discussed in this module:

● If statement

The if statement is used to create a branch in the execution flow of sequential statements. In VHDL, the if statement can only be used inside the process body. Example of a NAND gate with if statement:

```
library ieee;
use ieee.std_logic_1164.all;

entity nand_gate is
    port(
        A, B : IN STD_LOGIC;
        Y    : OUT STD_LOGIC
    );
end nand_gate;

architecture behavioral of nand_gate is
begin
    nand_proc : process (A, B) is
    begin
        if (A = '1' AND B = '1') then
            Y <= '0';
        else
            Y <= '1';
        end if;
    end process nand_proc;
end behavioral;
```

● Case statement

The case statement works in a way similar to the previous if statement. The difference is that the case statement will be more efficient to use when there are many variations of values. Example of a NAND gate using case statement:

```
library ieee;
use ieee.std_logic_1164.all;

entity nand_gate is
    port(
        A, B : IN STD_LOGIC;
        Y    : OUT STD_LOGIC
    );
end nand_gate;

architecture behavioral of nand_gate is
begin
    AB <= A & B;  -- combining signals A and B

    nand_proc : process (A, B) is
    begin
        case (AB) is
            when "11"    => Y <= '0';
            when others  => Y <= '1';
        end case;
    end process nand_proc;
end behavioral;
```

When using behavioral style, nested sequential statements are common and often used. This is also what makes behavioral style more **powerful** than data-flow style. However, even though behavioral style statements are used like programming in general, it should be noted that VHDL is a hardware description language, not a programming language. Also, try to always keep the process statement in the description you create as **simple** as possible to make hardware design easier when the circuit becomes more complex.

Revision #1
Created 2025-09-09 12:03:56 UTC by BH
Updated 2025-09-09 12:09:16 UTC by BH