

Lesson 21

Variables

Mohd Saufy Rohmad

Technical Trainer and Consultant

Variables

- Variables can be very useful for using integers in VHDL coding.
- Applying integers in VHDL makes life easier for a digital designer.
- A very useful example of applying variables can be found in the counter designs shown in the Intel's site.
- The link is given below:

Variables

- There is a difference between signal and variable in a process.
- Signals in a variable are updated instantly at the time of the trigger, while variables in a process are computed sequentially one after another. An example will clarify things.
- Lets assume, we have, $x1 = 1$, $x2 = 2$ and $x3 = 3$.
- We want to compute,
 - $x1 = x2$;
 - $x2 = x1 + x3$;
 - $x3 = x2$;
- We will first use $x1$, $x2$ and $x3$ as signals and then as variables inside a process.
- The first example of $x1$, $x2$ and $x3$ being signal is given below.

----- Signal in a process -----

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use IEEE.STD_LOGIC_ARITH.ALL;
```

```
entity SIG_test is  
  Port ( Trig : in  STD_LOGIC;  
         Res : out STD_LOGIC_VECTOR (31 downto 0));  
end SIG_test;
```

architecture SIG of SIG_test is

```
  signal signal1: STD_LOGIC_VECTOR(31 downto 0) := "000000000000000000000000000001";  
  signal signal2: STD_LOGIC_VECTOR(31 downto 0) := "0000000000000000000000000000010";  
  signal signal3: STD_LOGIC_VECTOR(31 downto 0) := "0000000000000000000000000000011";  
begin  
  process(Trig)  
  begin  
    signal1 <= signal2;  
    signal2 <= conv_std_logic_vector(signed(signal1) + signed(signal3),signal2'length);  
    signal3 <= signal2;  
    Res <= conv_std_logic_vector(signed(signal1) + signed(signal2) + signed(signal3),Res'length);  
  end process;  
end SIG;
```

example

----- Variables in a process -----

```
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use IEEE.STD_LOGIC_ARITH.ALL;
```

```
entity VAR_test is  
  Port ( Trig : in  STD_LOGIC;  
         Res : out STD_LOGIC_VECTOR (31 downto 0));  
end VAR_test;
```

architecture VAR of VAR_test is

```
begin  
  process(Trig)  
    variable variable1: integer :=1;  
    variable variable2: integer :=2;  
    variable variable3: integer :=3;  
    begin  
      variable1 := variable2;  
      variable2 := variable1 + variable3;  
      variable3 := variable2;  
      Res <= conv_std_logic_vector(variable1 + variable2 + variable3,Res'length);  
    end process;  
end VAR;
```

2nd example

Elaboration

- In the first example,
- $\text{Res} = 8$, while using x_1 , x_2 and x_3 as signals
- So, in the first one when $x_1 \leq 1$, $x_2 \leq 2$, $x_3 \leq 3$,
- the computations inside the process works like,
- $x_1 \leq x_2$; --- x_1 becomes 2
- $x_2 \leq x_1 + x_3$; --- x_2 becomes $1+3 = 4$, note that, the result of x_2 does not use the updated value of x_1
- $x_3 \leq x_2$; ---- x_3 becomes 2, note that, the result of x_3 does not use the updated value of x_2
- So, when triggered, all the computations use the present values of x .
- In the second example,
- $\text{Res} = 12$, because x_1 , x_2 and x_3 are updated in each line, in other words, work sequentially.

Thank You