

Nama	Edgrant Henderson Suryajaya
NPM	2206025016

Kode Asisten
Jenis Tugas	TP/CS.....

Teori

1. Testbench di VHDL adalah sebuah alat untuk membantu mengecek sebuah desain menggunakan simulasi wave. Test bench memberi stimulus untuk Unit Under Test (UUT) untuk mengecek hasil output. Maksudnya adalah test bench akan memberi input pada component dan kita dapat melihat output dari input yang berbeda-beda.

Test bench dalam VHDL dapat memberi stimulus input yang terdokumentasi dan dapat diulang dan dapat digunakan pada simulator-simulator yang beragam. Pada file vhd test bench, terdapat entity tanpa isi port IO, sebuah desain yang terinstansiasi sebagai component, dan berbagai port input

Contoh syntaxnya adalah seperti ini:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;

ENTITY tb_up_down IS
END tb_up_down; --entity kosong

ARCHITECTURE behavior OF tb_up_down IS

COMPONENT up_down_counter
PORT(
clock : IN std_logic;
reset : IN std_logic;
up_down : IN std_logic;
counter : OUT std_logic_vector(3 downto 0)
);
END COMPONENT;
```

Referensi:

- S. Team, “vhdl testbench Tutorial,” Invent Logics, Jan. 10, 2018. Available: <https://allaboutfpga.com/vhdl-testbench-tutorial/>
 - FTUI, Digital Laboratory “MODUL 4: TESTBENCH”, 2023. [online]. [Accessed: 19-Sep-2023].
2. Sebuah modul dalam VHDL adalah unit yang *self-contained*, artinya tidak berkomunikasi ke luar atau dari luar. Komunikasinya menggunakan entity, port map adalah bagian dari instansiasi modul yang memapping signal lokal pada *architecture* ke input dan outpunya modul. Pada sebuah testbench, port map berguna untuk menghubungkan signal-signal test bench ke entity yang ingin diuji.

Syntax dari testbench adalah sebagai berikut:

```
UUT: entity entityname port map (entitySignalName => localSignalName, entitySignalName2 => localSignalName2, ...);
```

Referensi:

- J. J. Jensen, “How to use Port Map instantiation in VHDL,” VHDLwhiz, Jul. 2023, Available: <https://vhdlwhiz.com/port-map/>
 - FTUI, Digital Laboratory “MODUL 4: TESTBENCH”, 2023. [online]. [Accessed: 19-Sep-2023].
3. Beberapa cara merubah nilai input testbench di VHDL:

Testbench Sederhana

Mengeset setiap input secara concurrent setelah waktu yang kumulatif. Cara ini tidak menggunakan proses, dan menggunakan assignment “<=” dengan keyword “after”. Contohnya adalah

```
a <= '0', '1' after 20 ns, '0' after 40 ns, '1' after 60 ns  
b <= '0', '0' after 20 ns, '1' after 40 ns, '1' after 60 ns
```



Tesbench Proses

Mengeset setiap input secara sekuensial menggunakan proses dengan jeda waktu diantara setiap assignment:

Syntaxnya adalah sebagai berikut

```
tb1: process()  
constant period: time := 20 ns  
begin  
    a <= '0';  
    b <= '0';  
    wait for period;  
    a <= '0';  
    b <= '1';  
    wait for period;  
    a <= '1';  
    b <= '0';  
    wait for period;  
    a <= '1';  
    b <= '1';  
    wait for period;  
    wait; -- pause proses  
end process tb1;
```

Testbench menggunakan look-up table

Membuat setiap input secara sekuensial dengan look-up table atau sebuah vector/array angka yang sudah dibuat. Metode ini juga menggunakan proses dan bersifat sekuensial.

Contoh syntaxnya:

```
architecture rtl of tes is
    constant stream_a : std_logic_vector (3 downto 0) := ('0', '1', '0', '1');
    constant stream_b : std_logic_vector (3 downto 0) := ('0', '0', '1', '1');
    constant period: time := 20 ns
begin
    tb1: process()
    begin
        for i in (0 to 3) loop
            a <= stream_a(i);
            b <= stream_b(i);
            wait for period
        end loop;
        wait; -- pause proses
    end process tb1;
```

Referensi:

- FTUI, Digital Laboratory “MODUL 4: TESTBENCH”, 2023. [online]. [Accessed: 19-Sep-2023].
- Digital Laboratory DTE FTUI, “Tutorial Praktikum PSD Modul 6: Testbench,” YouTube. Mar. 30, 2021. Available: <https://www.youtube.com/watch?v=3Gld-kKh41o>
- 4. Assert merupakan statement yang mengecek apakah sebuah kondisi benar, jika salah maka akan direport sebuah error. Sebuah kondisi yang ditulis dalam assertion statement harus bisa dievaluasi menjadi nilai boolean (true atau false). Report adalah message yang dikeluarkan saat terjadi error di assert, hal yang ditulis di report harus bertipe data string.

Terdapat 4 tingkat severitas assert:

- NOTE digunakan untuk memberi informasi saat simulasi.
- WARNING digunakan untuk hal yang unusual di simulasi, tetapi simulasi tetap berjalan walaupun output dapat agak tidak dapat diprediksi.
- ERROR digunakan saat kelanjutan simulasi tidak mungkin.
- FAILURE digunakan saat sebuah error terjadi dan simulasi harus dihentikan segera.
- can be used when the assertion violation is a fatal error and the simulation must be stopped at once (example 4).



Contoh syntaxnya adalah:

```
assert not (S= '1' and R= '1')  
report "Both values of signals S and R are equal to '1'"  
severity ERROR;
```

Referensi:

- “VHDL - Assertion Statement.” Available:
https://peterfab.com/ref/vhdl/vhdl_renerta/mobile/source/vhd00007.htm
- “VHDL Assert and report,” Sigasi, Feb. 02, 2015. Available:
<https://insights.sigasi.com/tech/vhdl-assert-and-report/>

