

**1 D4 - TEKKOM B**



**PRAKTIKUM 2**

**PENDALAMAN SIMULASI ISIM**



Nama : Septian Bagus Jumantoro

Kelas : 1 – D4 Teknik Komputer B

NRP : 3221600039

Dosen : Reni Soelistijorini B.Eng., MT.

Mata Kuliah : Praktikum Rangkaian Logika 2

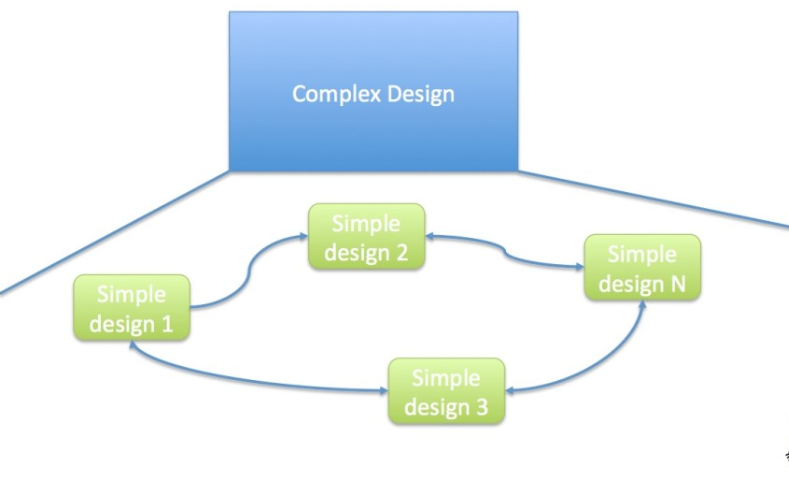
Hari/Tgl. Praktikum : Jumat, 18 Maret 2022

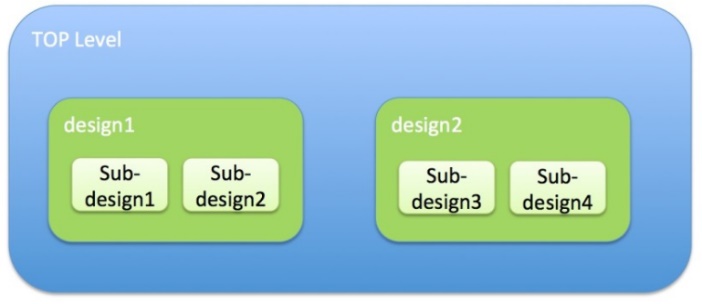
**PRAKTIKUM 2**

**PENDALAMAN SIMULASI ISIM**

1. TUJUAN
2. Mahasiswa dapat memahami konsep structural design.
3. Mahasiswa dapat membuat program dengan model structural design
4. TEORI
5. **DEFINISI**

* Sistem digital biasanya direpresentasikan sebagai kumpulan hirarki dari komponen. Setiap komponen memiliki satu set port yang berkomunikasi dengan komponen lain. Dalam deskripsi VHDL, hirarki desain diperkenalkan melalui deklarasi komponen dan pernyataan instantiasi komponen.
* Sementara unit dasar dari deskripsi behaviour adalah pernyataan proses, unit dasar dari deskripsi struktural adalah pernyataan instantiasi komponen.
* Pernyataan proses dan pernyataan instantiasi komponen harus dilampirkan dalam badan arsitektur.
* Istilah pemodelan struktural adalah terminologi yang digunakan VHDL untuk desain modular:
* proyek kompleks 🡪 bagi dalam dua atau lebih desain sederhana agar mudah menangani kerumitannya.
* Manfaat desain modular dalam VHDL : sebagaimana desain modular atau desain berorientasi objek untuk bahasa komputer tingkat tinggi.
* Desain modular memungkinkan Anda untuk mengemas fungsionalitas tingkat rendah ke dalam modul.
* Pendekatan ini memungkinkan penggunaan kembali desain tanpa perlu menemukan kembali dan menguji kembali setiap saat





**Structural Modeling: Hierarchical approach**

1. ALAT DAN BAHAN
2. Software ISE Design Suite Xilinx
3. LANGKAH PERCOBAAN
4. Module Komponen AND\_logic

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity AND\_logic is

port( a,b : IN STD\_LOGIC;

c : OUT STD\_LOGIC);

end AND\_logic;

architecture Behavioral of AND\_logic is

begin

c <= a and b;

end Behavioral;

1. Module Komponen OR\_logic

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity OR\_logic is

port( a,b : IN STD\_LOGIC;

c : OUT STD\_LOGIC);

end OR\_logic;

architecture Behavioral of OR\_logic is

begin

c <= a or b;

end Behavioral;

1. Module and\_or

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity and\_or is

port( a,b,d,e : IN STD\_LOGIC;

g : OUT STD\_LOGIC);

end and\_or;

architecture Behavioral of and\_or is

component AND\_logic

port( a,b : IN STD\_LOGIC;

c : OUT STD\_LOGIC);

end component;

component OR\_logic

port( a,b : IN STD\_LOGIC;

c : OUT STD\_LOGIC);

end component;

signal c : STD\_LOGIC;

signal f : STD\_LOGIC;

begin

u1: AND\_logic

port map( a=>a,

b=>b,

c=>c);

u2: AND\_logic

port map( a=>d,

b=>e,

c=>f);

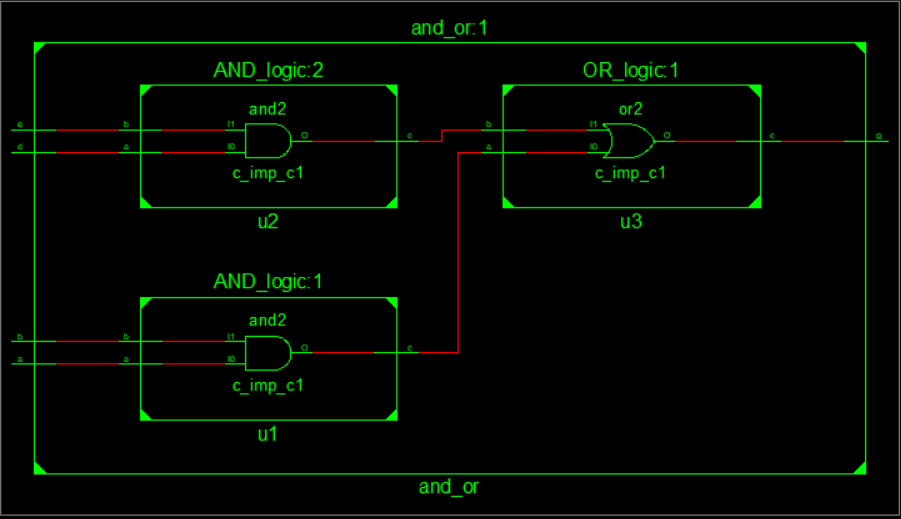
u3: OR\_logic

port map( a=>c,

b=>f,

c=>g);

end Behavioral;

1. RTL Schematic
2. TestBench and\_or

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

ENTITY tband\_or IS

END tband\_or;

ARCHITECTURE behavior OF tband\_or IS

COMPONENT and\_or

PORT(

a : IN std\_logic;

b : IN std\_logic;

d : IN std\_logic;

e : IN std\_logic;

g : OUT std\_logic

);

END COMPONENT;

signal a : std\_logic := '0';

signal b : std\_logic := '0';

signal d : std\_logic := '0';

signal e : std\_logic := '0';

signal g : std\_logic;

BEGIN

uut: and\_or PORT MAP (

a => a,

b => b,

d => d,

e => e,

g => g

);

a\_proc: process

begin

-- hold reset state for 100 ns.

wait for 25 ns;

a <= not a;

end process;

b\_proc: process

begin

-- hold reset state for 100 ns.

wait for 50 ns;

b <= not b;

end process;

d\_proc: process

begin

-- hold reset state for 100 ns.

wait for 100 ns;

d <= not d;

end process;

e\_proc: process

begin

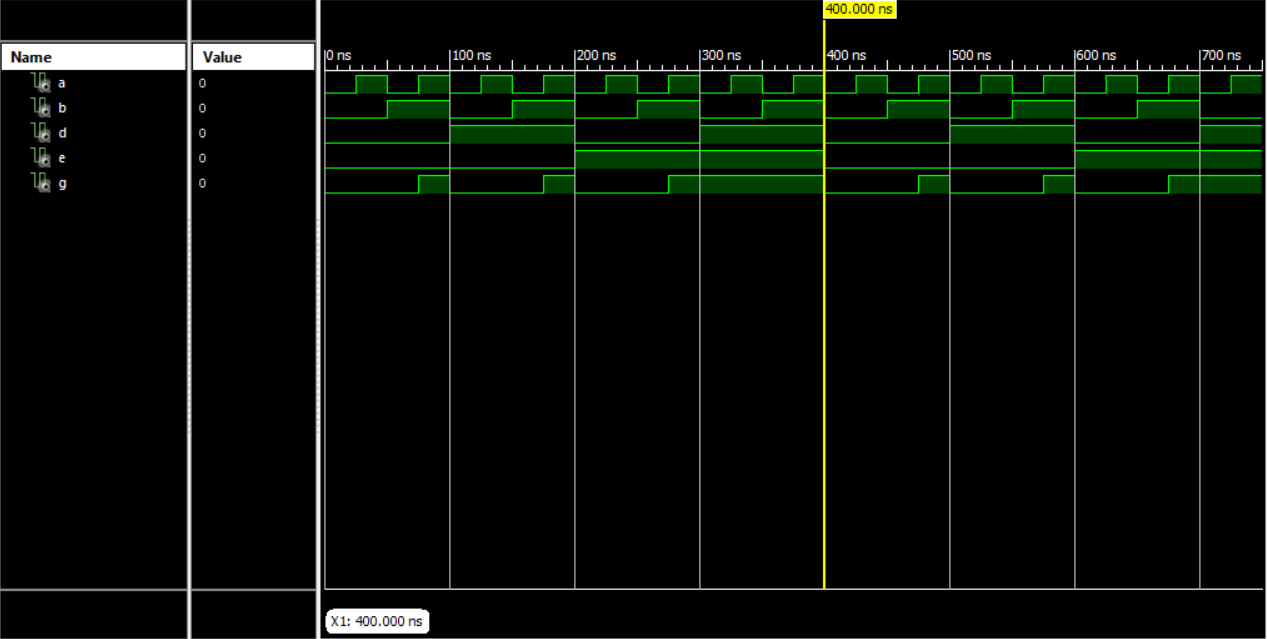
-- hold reset state for 100 ns.

wait for 200 ns;

e <= not e;

end process;

END;

1. Timing Diagram