Digital Laboratory

| | | | |
|---|---|---|---|
| **Nama** | Edgrant Henderson Suryajaya | **Kode Asisten** | …… |
| **NPM** | 2206025016 | **Jenis Tugas** | TP |

**Jawaban**

1. Truth table input output

| | Input | | | Output | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **I** | **C** | **B** | **A** | **g** | **f** | **e** | **d** | **c** | **b** | **a** |
| **0** | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |
| **1** | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| **2** | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| **3** | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| **4** | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |
| **5** | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| **6** | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| **7** | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |

2. Kmap dan persamaan setiap output

| G | BC | | | |
|---|---|---|---|---|
| | 00 | 01 | 11 | 10 |
| **A** **0** | 1 | 1 | 1 | 1 |
| **A** **1** | 1 | 1 | 1 | 1 |

G = 1

| F | BC | | | |
|---|---|---|---|---|
| | 00 | 01 | 11 | 10 |
| **A** **0** | 1 | 1 | 1 | 1 |
| **A** **1** | 0 | 0 | 0 | 1 |

F = A' + BC'

| E | BC | | | |
|---|---|---|---|---|
| | 00 | 01 | 11 | 10 |
| **A** **0** | 1 | 0 | 1 | 1 |
| **A** **1** | 1 | 1 | 1 | 1 |

E = A + B + C'

| D | BC | | | |
|---|---|---|---|---|
| | 00 | 01 | 11 | 10 |
| **A** **0** | 0 | 1 | 1 | 0 |
| **A** **1** | 1 | 1 | 1 | 1 |

D = A + C

| C | BC | | | |
|---|---|---|---|---|
| | 00 | 01 | 11 | 10 |
| **A** **0** | 1 | 1 | 0 | 0 |
| **A** **1** | 1 | 0 | 1 | 1 |

C = (A XNOR B) + AC'

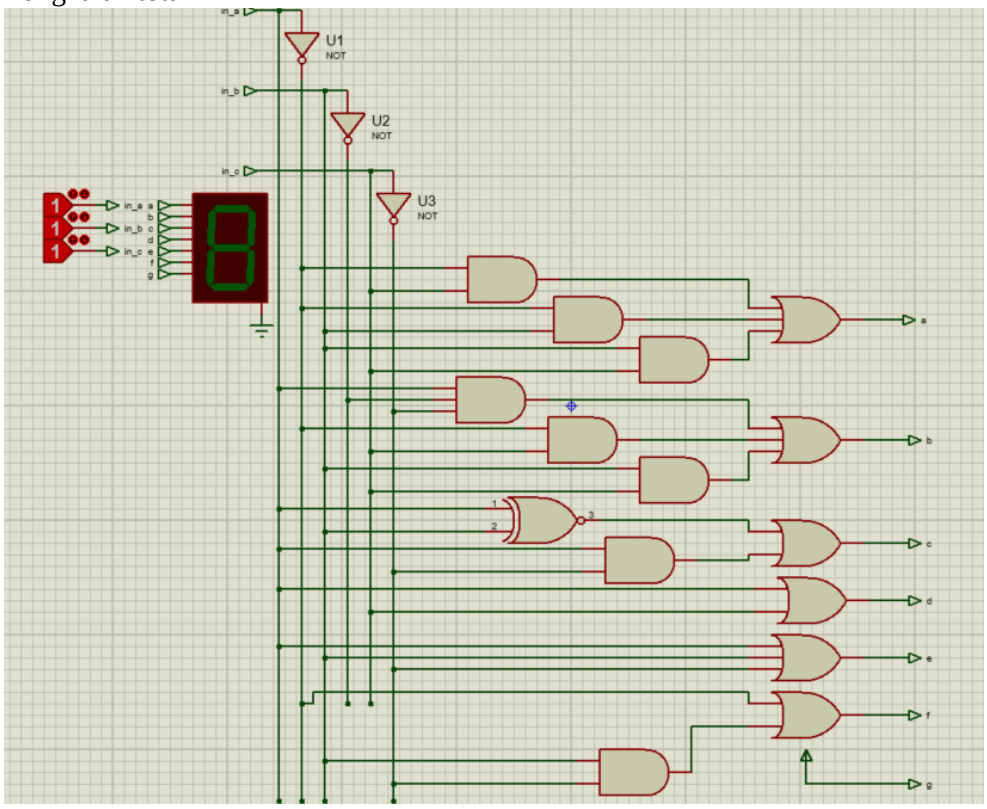
| B | BC | | | |
|---|---|---|---|---|
| | 00 | 01 | 11 | 10 |
| **A** **0** | 0 | 1 | 1 | 0 |
| **A** **1** | 1 | 0 | 1 | 0 |

B = BC + A'C +AB'C'

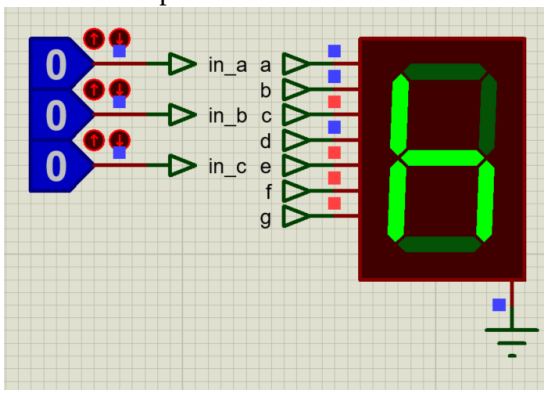
| A | BC | | | |
|---|---|---|---|---|
| | 00 | 01 | 11 | 10 |
| **A** **0** | 0 | 1 | 1 | 1 |
| **A** **1** | 0 | 0 | 1 | 0 |

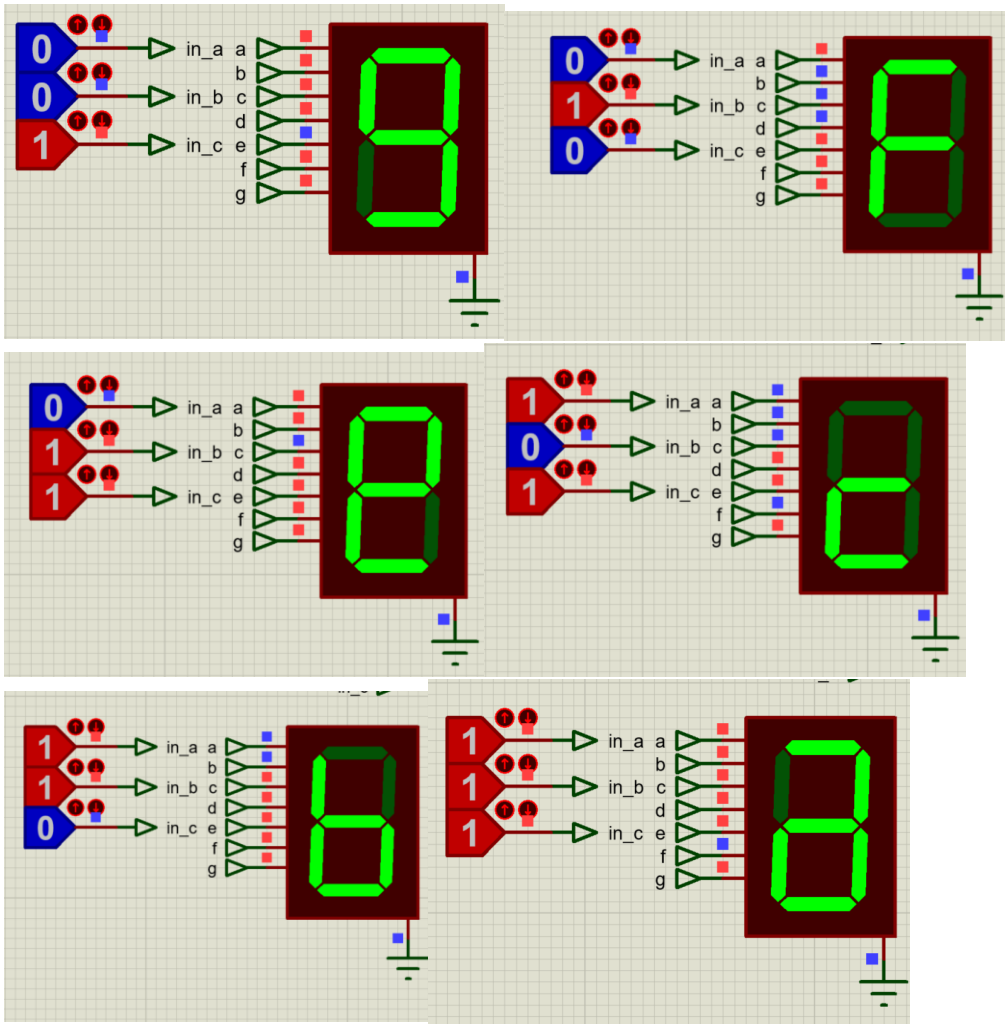
A = A'C + A'B + BC

3. Rangkaian total




Gambar setiap kombinasi




0
0
1
in_a
in_b
in_c
a
b
c
d
e
f
g
0
1
0
in_a
in_b
in_c
a
b
c
d
e
f
g
0
1
1
in_a
in_b
in_c
a
b
c
d
e
f
g
1
0
1
in_a
in_b
in_c
a
b
c
d
e
f
g
1
1
0
in_a
in_b
in_c
a
b
c
d
e
f
g
1
1
1
in_a
in_b
in_c
a
b
c
d
e
f
g

4. Flow style dan behavioral style adalah paradigma pemograman di vhdl
   **Dataflow style** adalah sistem pemograman yang menunjukkan bagaimana data mengalir di sistem, penulisan data sama seperti implementasi di hardware, yakni menggunakan gerbang logika. Dataflow style bersifat manyak concurrent signal yang jalan sekaligus
   Contohnya:

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

entity half_adder is
  port (a, b: in std_logic;
    sum, carry_out: out std_logic);
  end half_adder;

architecture dataflow of half_adder is
  begin
  sum <= a xor b;
  carry_out <= a and b;
end dataflow;
```

   **Behavioral style** adalah sistem pemograman di vhdl secara fungsional dan algoritmik, gaya ini lebih abstrak dibanding dataflow dan tidak langsung menulis di level gerbang logika. Style ini biasanya menggunakan process statement yang setiapnya merupakan 1 statement concurrent yang sekuensial
   Contohnya:

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

entity half_adder is
  port (a, b: in std_logic;
    sum, carry_out: out std_logic);
  end half_adder;

architecture behavior of half_adder is
  begin
    ha: process (a, b)
    begin
      if a = '1' then
        sum <= not b;
        carry_out <= b;
      else
        sum <= b;
        carry_out <= '0';
      end if;
    end process ha;

end behavior;
```

   Referensi:


- "VHDL Modelling Styles: Behavioral, Dataflow, Structural," *Buzztech*, Apr. 24, 2018. https://buzztech.in/vhdl-modelling-styles-behavioral-dataflow-structural/
- Efy, "Behavioral Style of Modelling," *Electronics for You*, Mar. 30, 2023. https://electronicsforyou.in/behavioral-style-of-modelling/

5. Dalam blok proses, sensitivity list adalah parameter ke proses yang memberi tahu semua signal yang sensitif terhadap proses. Kalau satu pun signal yang berubah, proses akan berjalan dan kode di dalamnya akan tereksekusi.
   Syntax dari sebuah proses adalah

```
process(<signal1>, <signal2>, ..) is
 begin
    <main logic here>
 end process;
```

   signal 1 dan signal 2 adalah dalam sensitivity list, jika salah satunya berubah, proses akan berjalan.
   **Referensi:**
   - J. J. Jensen, "How to create a process with a sensitivity list in VHDL," *VHDLwhiz*, Aug. 2023, [Online]. Available: https://vhdlwhiz.com/sensitivity-list/
   - Efy, "Behavioral Style of Modelling," *Electronics for You*, Mar. 30, 2023. https://electronicsforyou.in/behavioral-style-of-modelling/

6. Wait statement secara umum berfungsi untuk menunggu dalam sebuah proses sekuensial. Wait dapat diatur untuk menunggu waktu tertentu (contohnya 10ns (wait for 10 ns)), tetapi juga bisa menunggu hal spesifik seperti sensitivity list menggunakan syntax (wait on *sensitivity list*). Terakhir wait juga dapat menunggu sampai sebuah kondisi true (wait until *condition*). Catatan, proses yang ada sensitivity list di paramaternya gak bisa ada wait statement.
   **Referensi:**
   - J. J. Jensen, "How to delay time in VHDL: Wait For," *VHDLwhiz*, Aug. 2023, [Online]. Available: https://vhdlwhiz.com/wait-for/
   - Russell, "VHDL Example Code of wait Statement," *Nandland*, Jun. 30, 2022. https://nandland.com/wait-statement-wait-until-wait-on-wait-for/

7. **Kode**

```
library IEEE;
use IEEE.std_logic_1164.all;

entity mux is
    port (
        I : IN STD_LOGIC_VECTOR (3 downto 0);
        S : IN STD_LOGIC_VECTOR (1 downto 0);
        O : OUT STD_LOGIC
    );
end entity mux;

architecture rtl of mux is

begin
    process(I, S)
    begin
        if S(0) = '0' and S(1) = '0' then
            O <= I(0);
        elsif S(0) = '0' and S(1) = '1' then
            O <= I(1);
        elsif S(0) = '1' and S(1) = '0' then
            O <= I(2);
        else
            O <= I(3);
        end if;
    end process;
end architecture rtl;
```