

1 D4 - TEKKOM B

STRUCTURAL DESIGN



Nama	:	Septian Bagus Jumentoro
Kelas	:	1 – D4 Teknik Komputer B
NRP	:	3221600039
Dosen	:	Reni Soelistijorini B.Eng., MT.
Mata Kuliah	:	Praktikum Rangkaian Logika 2
Hari/Tgl. Praktikum	:	Jumat, 13 Mei 2022



Percobaan 6 – Structural Design

1. Tujuan

- Dapat membuat program dengan model disain structural.
- Dapat mengimplementasikan algoritma multiplexer pada disain structural.

2. Teori

Sebuah pemodelan terstruktur untuk menjelaskan jalur komponen dan koneksinya. Setiap komponen diasumsikan telah didefinisikan sebelumnya dan dapat dijelaskan secara terstruktur, sebuah behavior atau model dataflow. Pada hirarki terendah setiap komponen dideskripsikan sebagai pemodelan behavior menggunakan operator logika dasar di VHDL. Secara umum pemodelan terstruktur sangat bagus untuk mendeskripsikan sistem digital yang kompleks, meskipun seperangkat komponen dalam model hirarki. Sebuah penjelasan struktur dapat dibandingkan secara terbaik terhadap diagram blok skematik yang dapat dijelaskan dengan komponen dan interkoneksi. VHDL memberikan cara sebagai berikut :

- Deklarasikan kompoen yang digunakan
- Deklarasikan sinyal yang menentukan nets yang menghubungkan dengan komponen
- Melabelkan kompoenen yang sama berkali – kali sehingga setiap contoh didefinisikan secara unik.

Komponen dan sinyal dideklarasikan di dalam arsitektur body.

```
architecture architecture_name of NAME_OF_ENTITY is
    -- Decl arations
    component decl arations
    signal decl arations
begin
    -- Stat ements
    component instantiation and connect ions ;
end architecture_name;
```

Deklarasi Komponen

Sebelum komponen dapat dipakai mereka butuh di deklarasikan di dalam sesi deklarasi arsitektur atau dalam paket deklarasi. Deklarasi komponen terdiri dari nama komponen dan interface nya (ports). Sintak nya sebagai berikut:

```

component component_name
    port (
        port_signal_names : mode type;
        port_signal_names : mode type;

        :
        port_signal_names : mode type);
end component ;

```

Nama komponen merujuk pada nama dari entity yang didefinisikan di library juga sebuah entity yang secara jelas didefinisikan di dalam file VHDL.

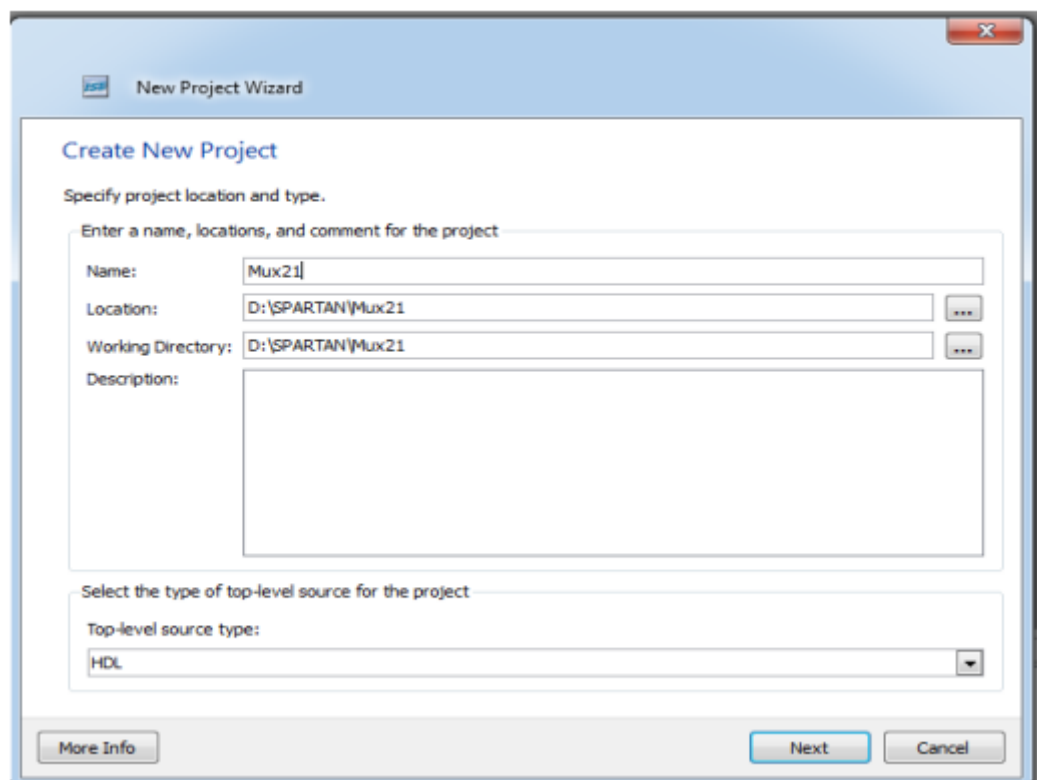
3. Alat dan Bahan

1. PC yang sudah terinstall ISE 13.1
2. Xilinx Spartan 3
3. Downloader JTAG USB
4. Power Supply 5 volt

4. Langkah Percobaan

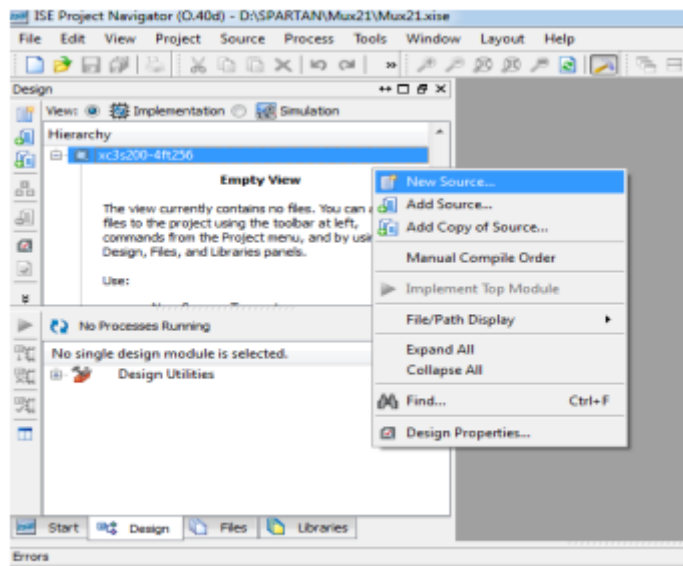
A. Membuat File Top Modul

1. Membuat project baru dengan nama folder Mux21, top level source HDL.



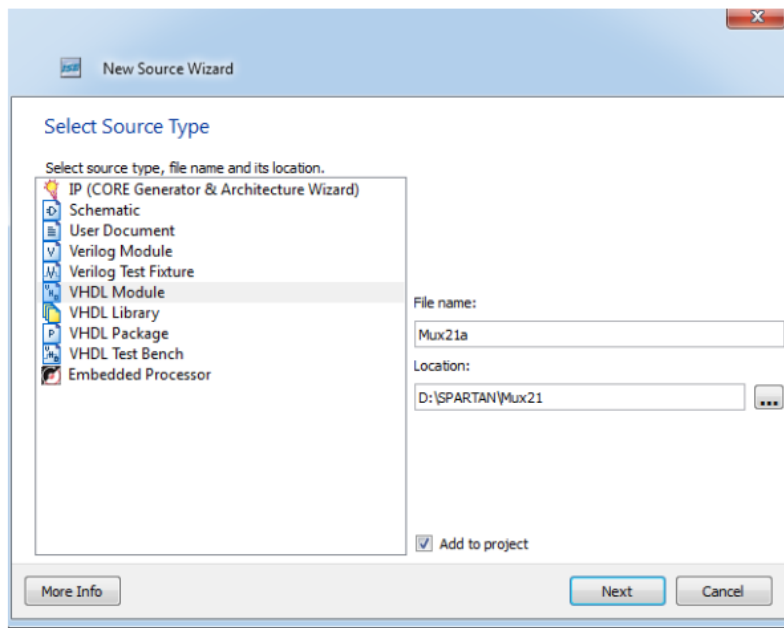
Gambar 6. 1 Membuat project baru dengan nama folder

- Setelah selesai membuat folder, klik kanan pada type IC kemudian pilih New Source untuk membuat file .vhd



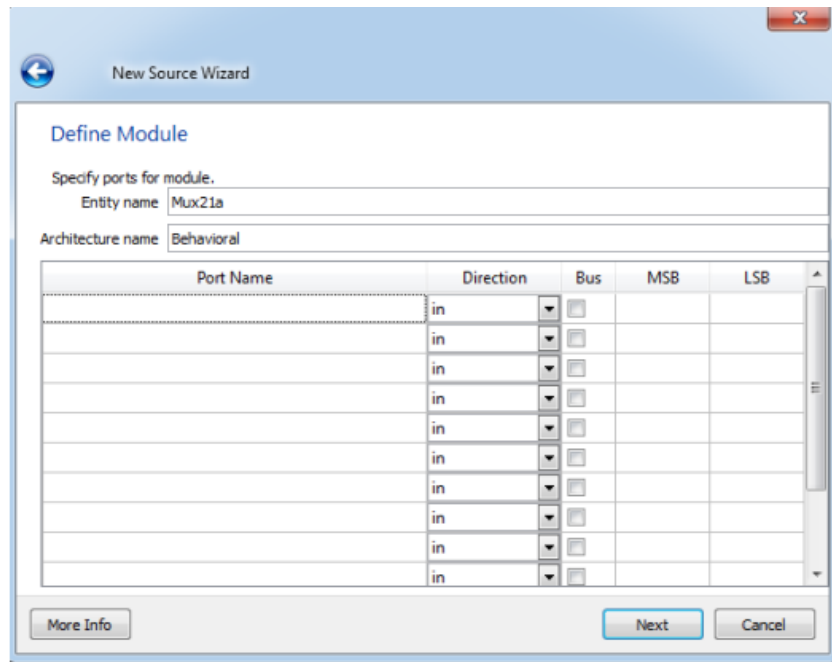
Gambar 6. 2 New Source

- Pada **New Source Wizard** - Pilih **VHDL Module** dan beri nama file misalnya Mux21a kemudian klik **next**.



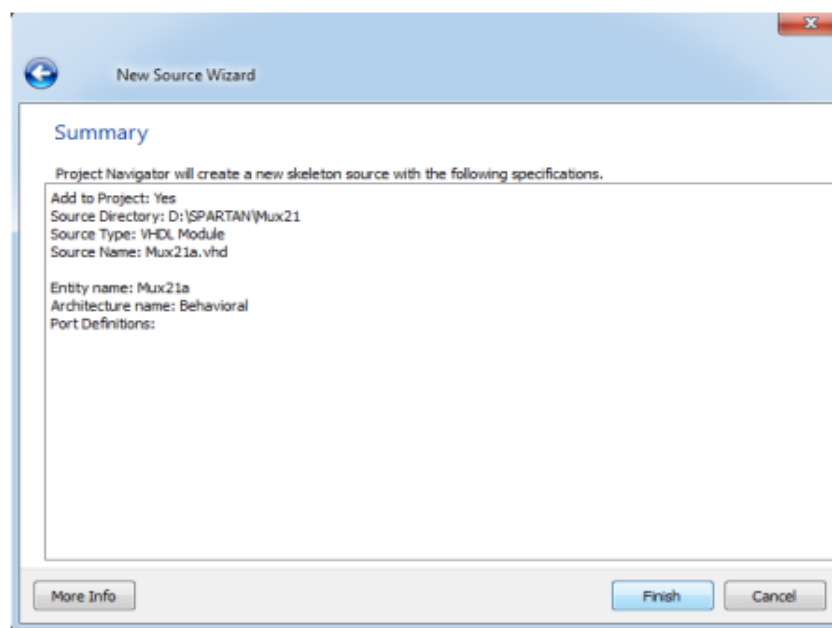
Gambar 6. 3. Memberi nama file

- Pada sesi Define Module biarkan kosong dan lewati saja, kemudian klik next.



Gambar 6. 4. Define module

5. Pilih Finish



Gambar 6. 5 New Source Wizard finish

6. Lengkapi program pada file Mux21a seperti pada skrip 6.1

```

Library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Mux21a is
    Port (
        a : in STD_LOGIC;
        lxx
    );
end entity Mux21a;

```

```

        b :inSTD_LOGIC;
        s :inSTD_LOGIC;
        y :outSTD_LOGIC

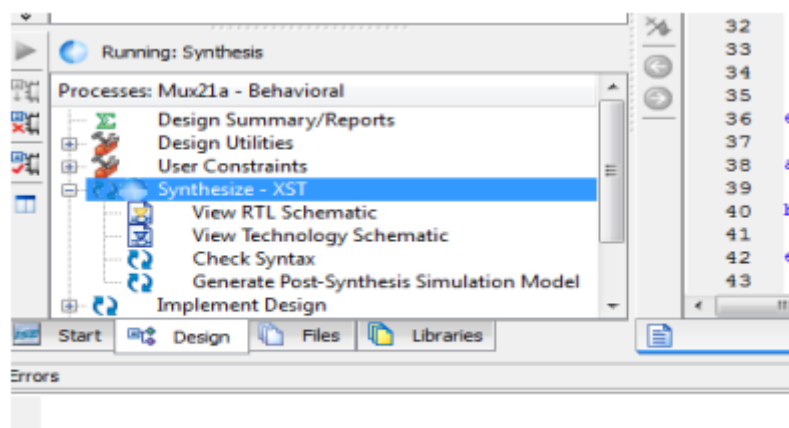
    );
end Mux21a;

architecture Behavioral of Mux21a is
begin
    y <= (not s and a) or (s and b);
end Behavioral;

```

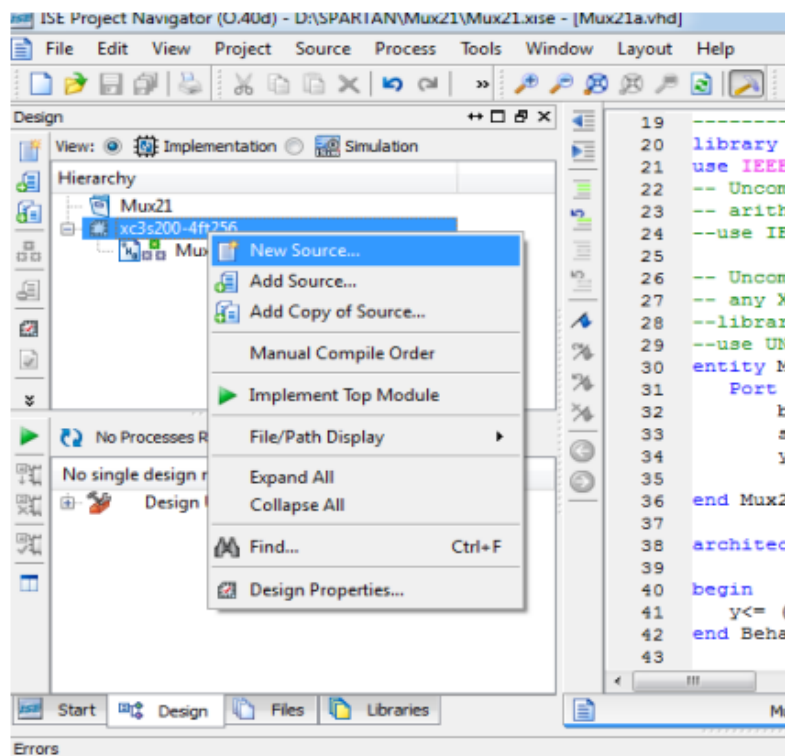
Skrip 6. 1 Program Mux21a

7. Lakukan double klik pada Synthesize-xst untuk check Syntax.

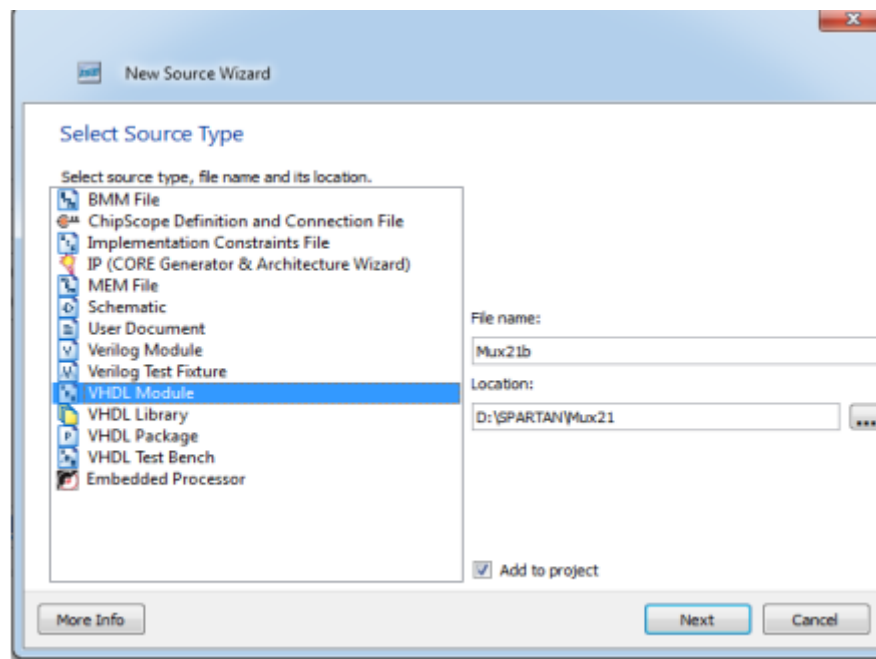


Gambar 6. 6 Double klik Synthesize -XST

8. Membuat file kedua dengan cara yang sama, klik kanan pada nama IC kemudian pilih New-Source

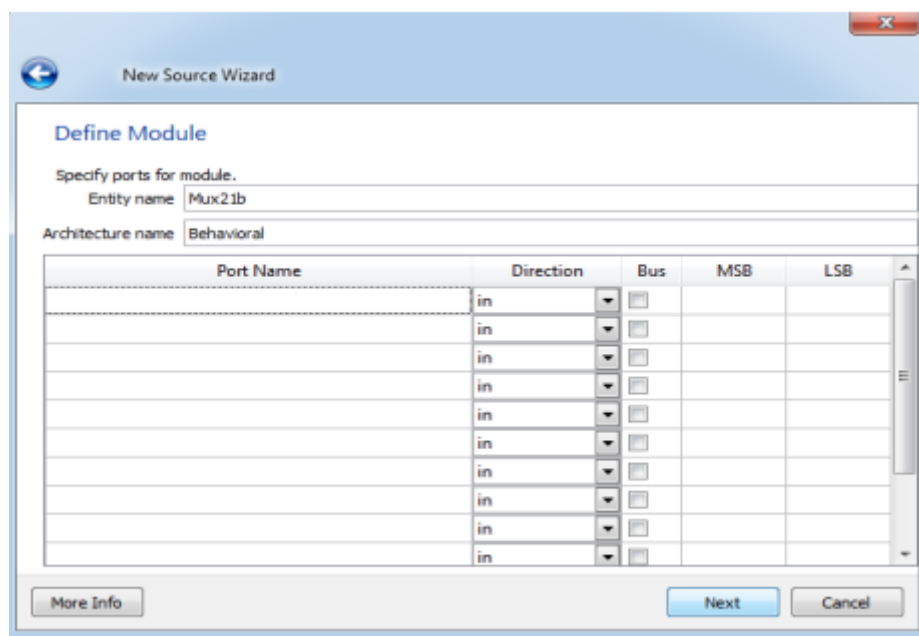


9. Pada New Source Wizard bagian Select Source Type, pilih VHDL Module dan beri nama file misalnya Mux21b, kemudian klik next



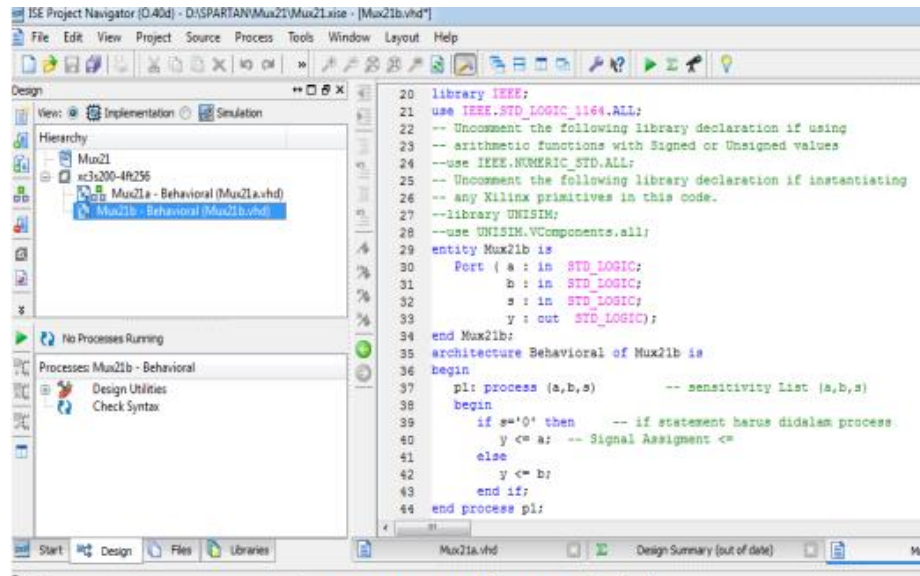
Gambar 6. 8 Membuat file baru Mux21b.vhdl

10. Pada New Source Wizard bagian Define Module, biarkan dan lewati saja bagian ini kemudian klik next. Dan selanjutnya klik Finish.



Gambar 6. 9 Define Module

11. Lengkapi program pada file Mux21b seperti gambar 6.10.



Gambar 6. 10 Program file Mux21b.vhdl

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Mux21b is
    Port(
        a:inSTD_LOGIC;
        b:inSTD_LOGIC;
        s:inSTD_LOGIC;
        y:outSTD_LOGIC;
    );
end Mux21b;

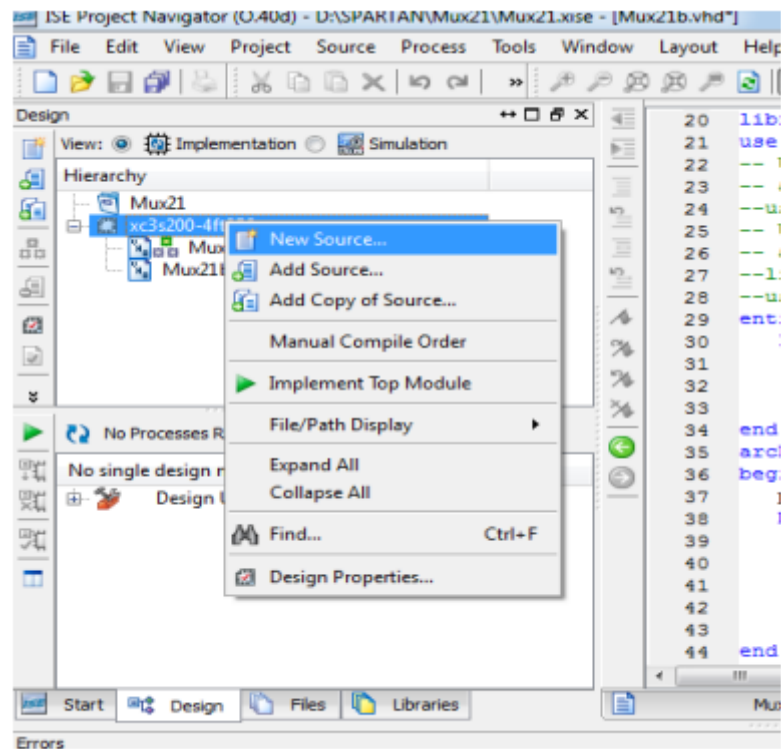
architecture Behavioral of Mux21b is
begin
p1 :process(a,b,s)
    begin
        if s='0' then
            y <= a;

        else
            y <= b;
        endif;
    end process p1;

```

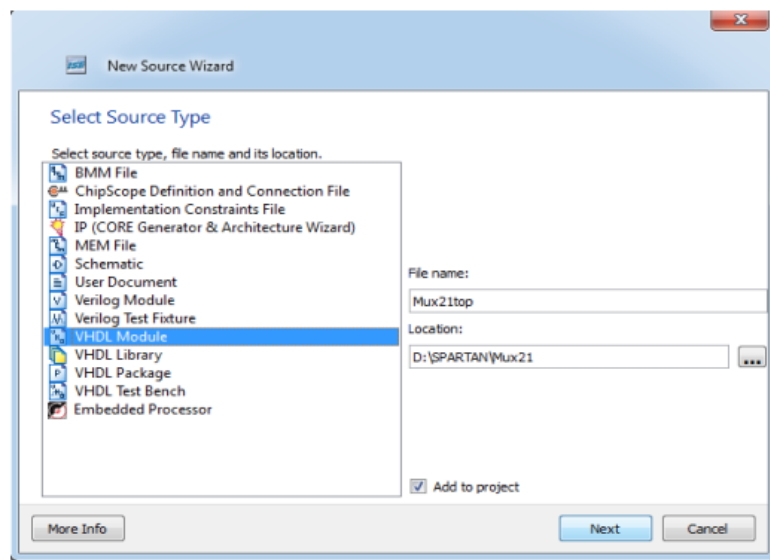
Skrip 6. 2 Program file Mux21b.vhdl

12. Kemudian buat file baru yang ketiga, dengan cara klik kanan pada nama IC- pilih New Source



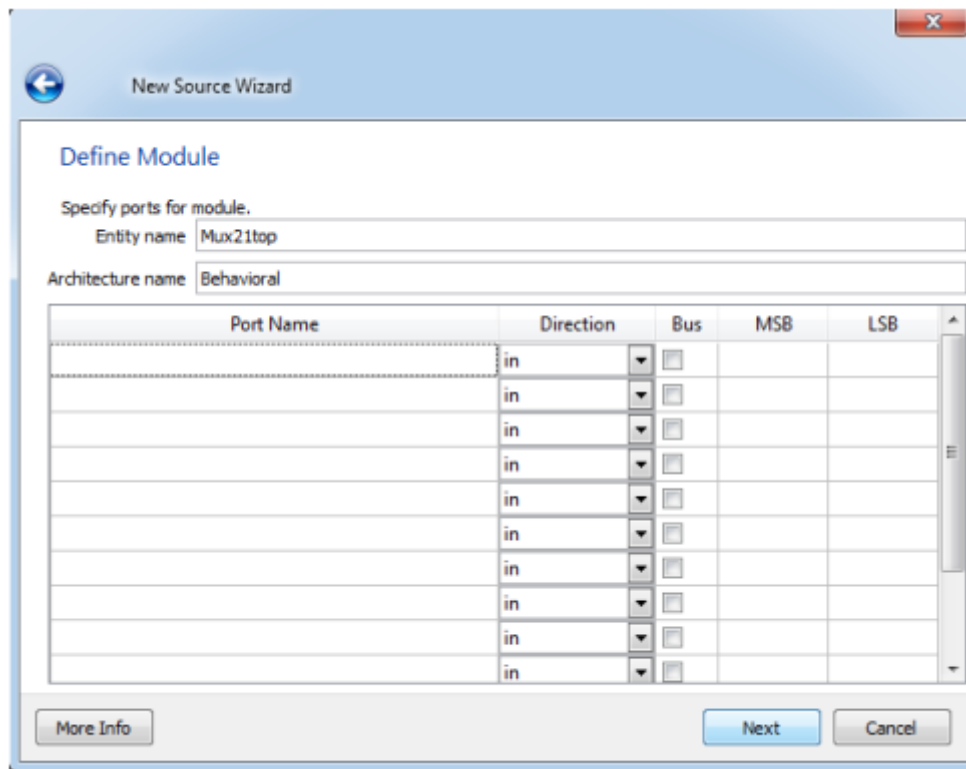
Gambar 6. 11 Membuat file baru

13. Pada New Source Wizard bagian Select Source Type pilih VHDL Module dan beri nama file Mux21top.



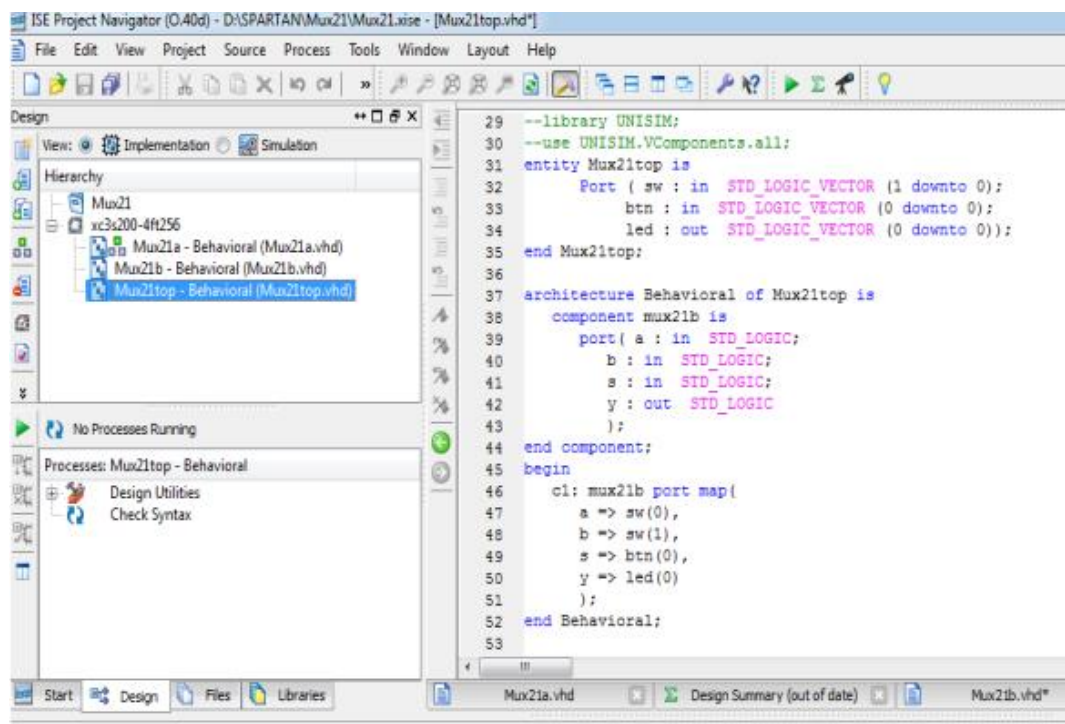
Gambar 6. 12 Membuat file baru .vhd

14. Pada bagian Define Module biarkan kosong dan lewati saja, kemudian klik next, dan selanjutnya klik Finish.



Gambar 6. 13 Define Module

15. Lengkapi kode program file Mux21top seperti pada gambar 6.13.



Gambar 6. 14. Program file Mux21top

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.all;

entity Mux21top is
    Port(
        sw : in STD_LOGIC_VECTOR(1 downto 0);
        btn : in STD_LOGIC_VECTOR(0 downto 0);

        led : out STD_LOGIC_VECTOR(0 downto 0)
    );
end Mux21top;

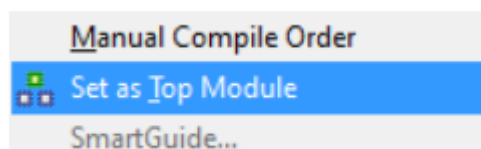
architecture Behavioral of Mux21top is
    component mux21b is
        port(
            a : in STD_LOGIC;
            b : in STD_LOGIC;
            s : in STD_LOGIC;
            y : out STD_LOGIC
        );
    end component;

begin
    c1: mux21b portmap(
        a => sw(0),
        b => sw(1),
        s => btn(0),
        y => led(0);
    );
end Behavioral;

```

Skrup 6. 3 Kode program file Mux21top.vhdl

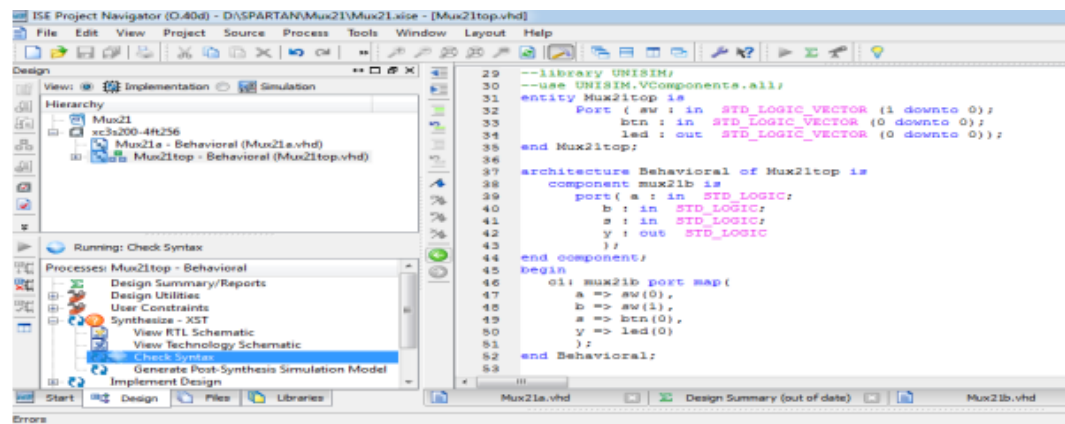
Saat kita membuat pertama kali sebuah file project .vhdl, maka secara otomatis ISE akan menempatkan file tersebut sebagai top modul. Sedangkan yang kita ingin jadikan sebagai top modul adalah file Mux21top. Sehingga untuk membuat file Mux21top menjadi top model caranya adalah klik kanan pada nama file, pilih Set as Top Module.



Gambar 6. 15 Kode program file Mux21top.vhdl

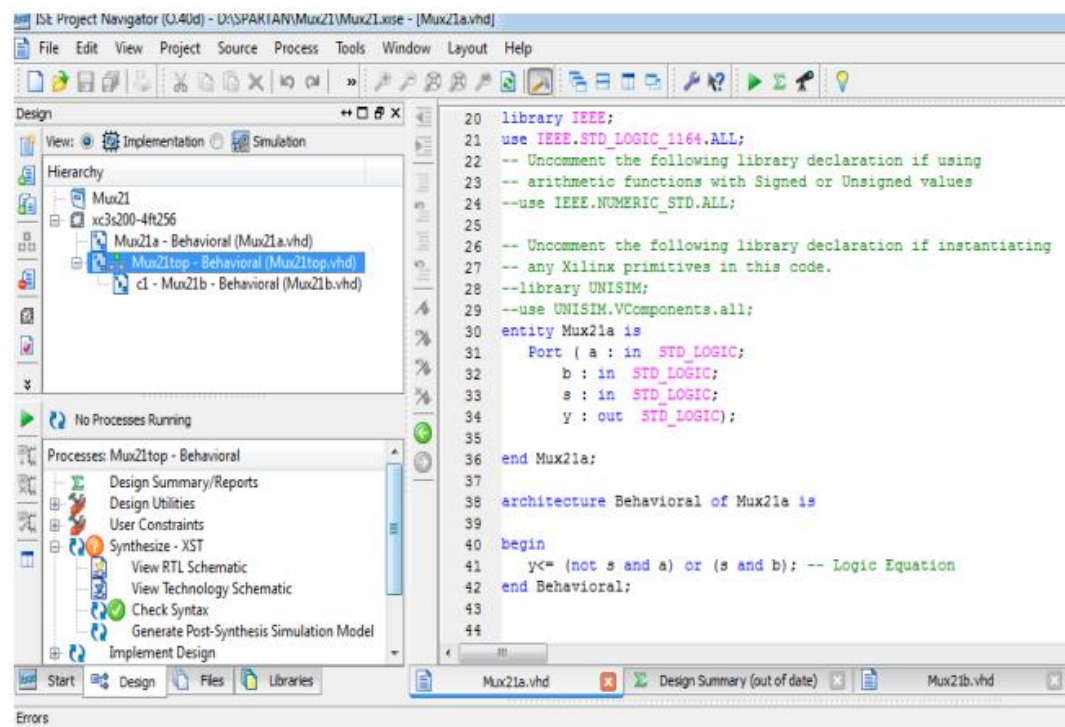
16. Lakukan Check Syntax pada file Mux21b, untuk cek kode program. (Secara otomatis file Mux21b akan pindah menjadi bagian dari file Mux21top, dan

secara otomatis pula file Mux21b menjadi top module. Hal ini karena program dari file Mux21b sudah di-call oleh program Mux21top)



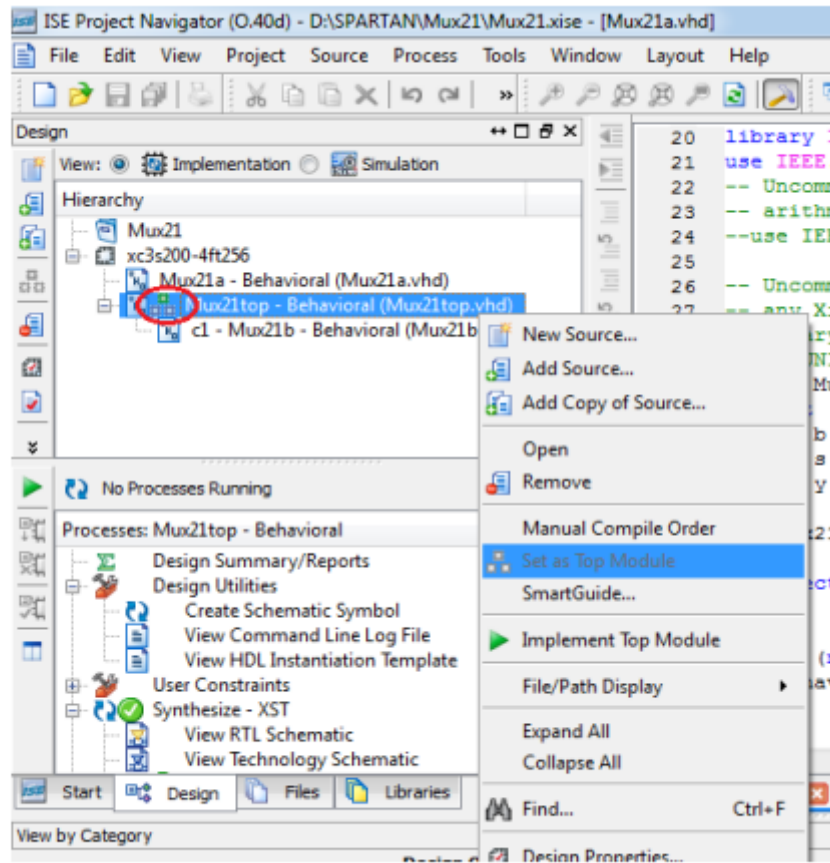
Gambar 6. 16 Check Syntax

17. Karena program Mux21a tidak di-call oleh file Mux21top, maka program Mux21a tidak berada di akar program Mux21top. Kemudian secara otomatis, Mux21top akan menjadi top module yang di dalamnya ada module Mux21b.



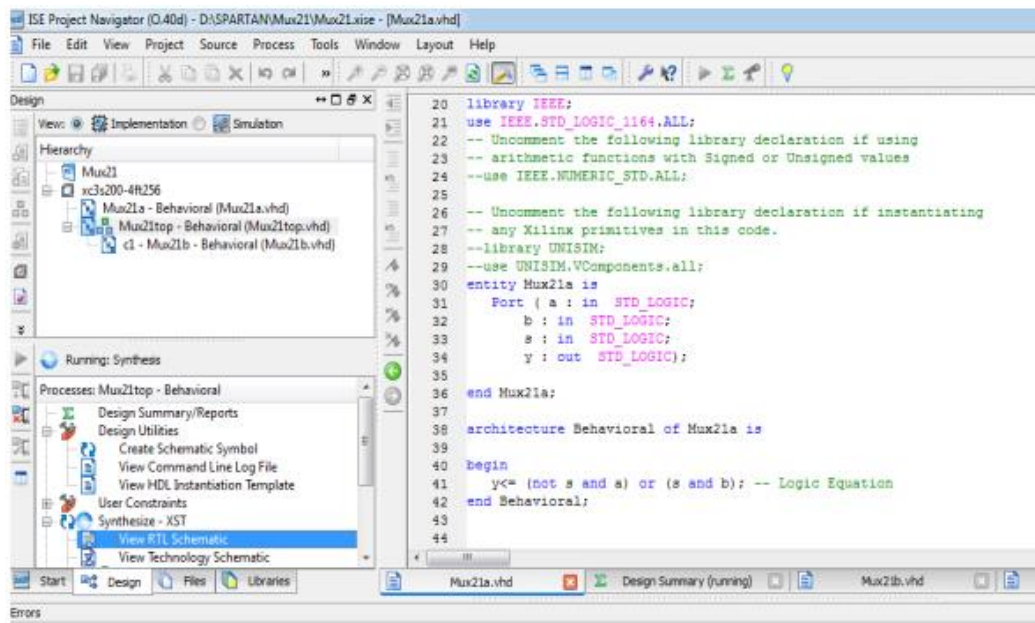
Gambar 6. 17 Top Module

Untuk membuat top module secara manual, caranya sebagai berikut :



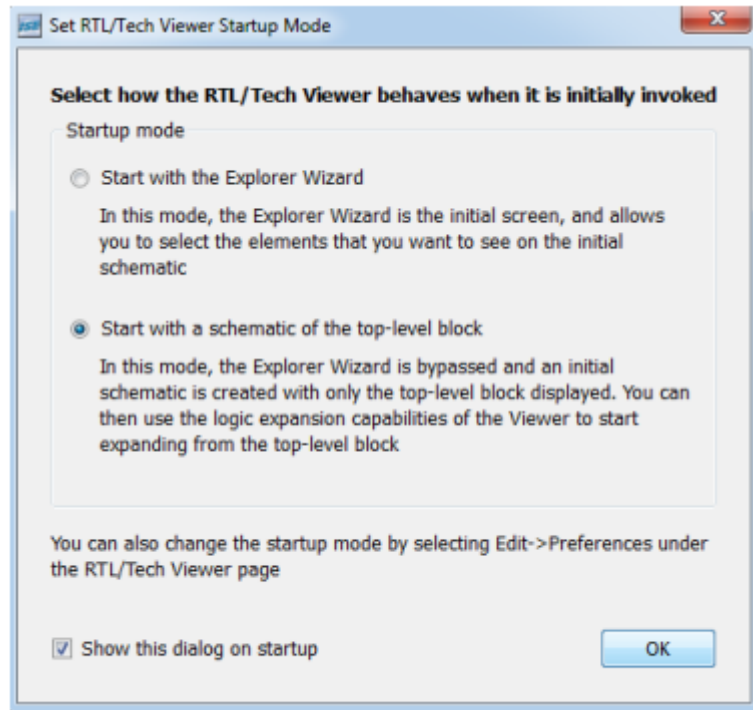
Gambar 6. 18 Tanda Top Module dan cara membuat file menjadi Top Module

18. View RTL Schematic dengan double klik View RTL Schematic pada Synthesize-XST



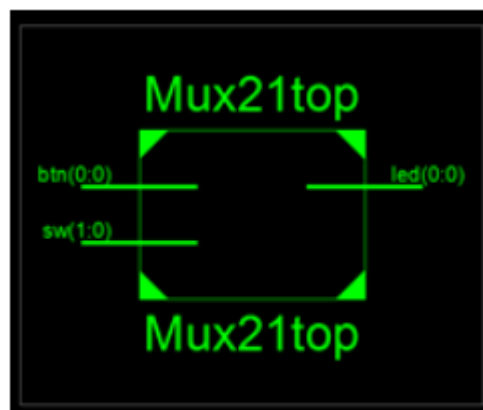
Gambar 6. 19 Double Klik View RTL Schematic

19. Pilih Start with a schematic of the top-level block pada dialog Set RTL



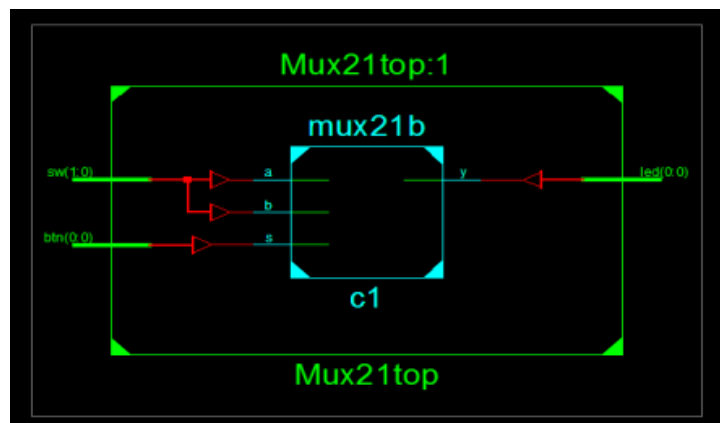
Gambar 6. 20 Set RTL Start with a schematic

20. Akan muncul diagram blok dari top module dengan IO, untuk melihat isi bagian dalam top module, klik 2x pada diagram blok top module.



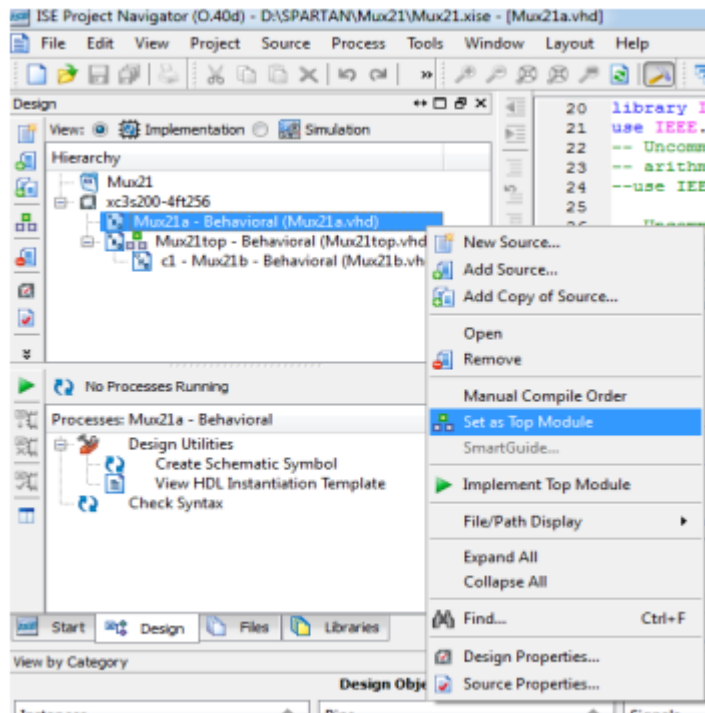
Gambar 6. 21 Diagram Blok top module

21. Akan diperlihatkan shcematic dari top module Mux21top.



Gambar 6. 22 Schematic Mux21top

22. Bila ingin menjadikan file lain sebagai top module, caranya adaah klik kanan pada nama file-pilih Set as top module



Gambar 6. 23 Membuat top module

5. Latihan

Membangun full adder 1 bit untuk membangun full adder 4 bit.

1. Buat program full adder 1 bit dan beri nama FA_1.vhd dan full adeder 4 bit dengan nama FA_4.

```
----- FA_1.vhd -----  
entity FA_1 is  
    Port ( A : in STD_LOGIC;  
          B : in STD_LOGIC;  
          Cin : in STD_LOGIC;  
          S : out STD_LOGIC;  
          Cout : out STD_LOGIC );  
end FA_1;  
  
architecture Behavioral of FA_1 is  
begin  
    S <= A xor B xor Cin;  
    Cout <= (A and B) or (A and Cin) or (B and Cin);  
end Behavioral;
```

```

----- FA_4.vhd -----
entity FA_4 is
    Port ( a : in  STD_LOGIC_VECTOR (3 downto 0);
          b : in  STD_LOGIC_VECTOR (3 downto 0);
          cin : in  STD_LOGIC;

```

```

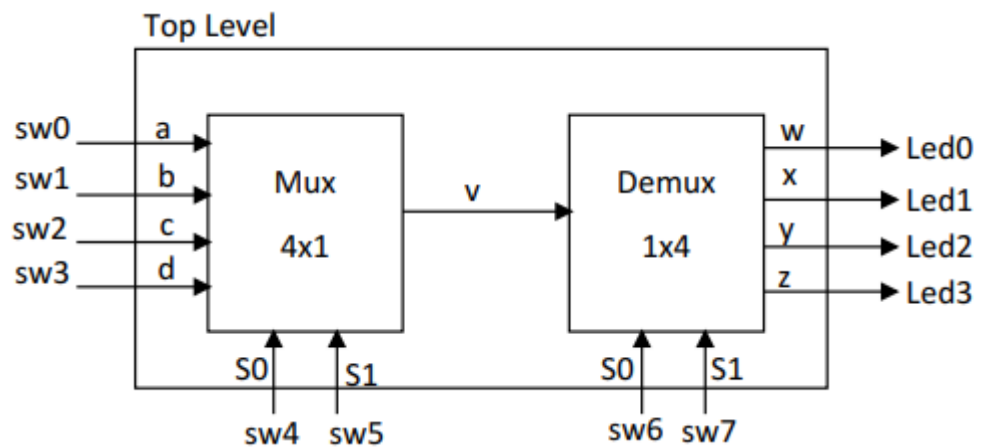
          sum : out  STD_LOGIC_VECTOR (3 downto 0);
          cout : out  STD_LOGIC );
end FA_4;

architecture Behavioral of FA_4 is
    component fA_1
        port(
            A, B, Cin : in std_logic;
            S, Cout : out std_logic
        );
    end component;
    signal ci : std_logic_vector(4 downto 0);

begin
    ci(0) <= cin;
    u0 : fA_1 port map(A => a(0), B => b(0), Cin => ci(0), S => sum
(0), Cout => ci(1));
    u1 : fA_1 port map(A => a(1), B => b(1), Cin => ci(1), S => sum
(1), Cout => ci(2));
    u2 : fA_1 port map(A => a(2), B => b(2), Cin => ci(2), S => sum
(2), Cout => ci(3));
    u3 : fA_1 port map(A => a(3), B => b(3), Cin => ci(3), S => sum
(3), Cout => ci(4));
    cout <= ci(4);
end Behavioral ;

```

2. Membangun multiplex dan demultiplex
3. Buatlah program Multiplexer 4x1 menggunakan if statement
4. Buatlah program Demultiplexer 1x4 menggunakan if statement.
5. Buatlah program top_level-nya seperti pada gambar dibawah ini



Hasil Praktikum

1. Multiplexer

Source Code

```

20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 |
23 entity mux21b is
24 port ( a,b,s : in std_logic;
25        y : out std_logic
26        );
27 end mux21b;
28
29 architecture Behavioral of mux21b is
30 begin
31   p1 : process (a,b,s)
32   begin
33     if s ='0' then
34       y <= a;
35     else
36       y <= b;
37     end if;
38   end process;
39 end Behavioral;
40

```

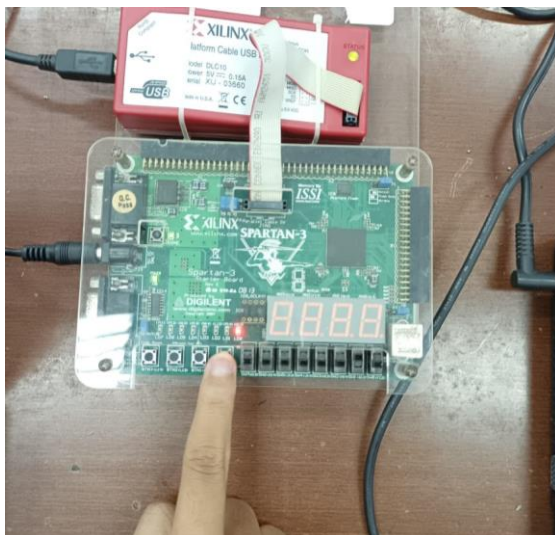
Top Level

```
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.all;
3  use IEEE.STD_LOGIC_unsigned.all;
4  entity mux2ltop is
5  Port(
6      sw :in STD_LOGIC_VECTOR(1 downto 0);
7      btn :in STD_LOGIC_VECTOR(0 downto 0);
8      led :out STD_LOGIC_VECTOR(0 downto 0)
9  );
10 end mux2ltop;
11 architecture Behavioral of mux2ltop is
12 component mux2lb is
13     port(
14         a :in STD_LOGIC;
15         b :in STD_LOGIC;
16         s :in STD_LOGIC;
17         y :out STD_LOGIC
18     );
19 end component;
20 begin
21     c1:mux2lb port map(
22         a => sw(0),
23         b => sw(1),
24         s => btn(0),
25         y => led(0)
26     );
27 end Behavioral ;
```

Pin yang digunakan

```
1  NET sw(0) loc = F12;
2  NET sw(1) loc = G12;
3  NET btn(0) loc = M13;
4  NET led(0) loc = K12;
```

Tampilan pada Board



Gambar 1. Mux dengan input 10 dan switch 1



Mux dengan input 01 dan switch 0

2. Demultiplexer

Source Code

```

20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 entity demuxb is
23 port ( D,S : in std_logic;
24       y : out std_logic_vector(1 downto 0)
25       );
26 end demuxb;
27
28 architecture Behavioral of demuxb is
29 begin
30     y(0) <= not (S) and D;
31     y(1) <= S and D;
32 end Behavioral;

```

Top Level

```
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22 use IEEE.STD_LOGIC_unsigned.all;
23
24 entity mux21top is
25 port ( sw : in std_logic_vector (1 downto 0);
26       btn : in std_logic_vector (0 downto 0);
27       led : out std_logic_vector (0 downto 0)
28 );
29
30 end mux21top;
31
32 architecture Behavioral of mux21top is
33
34 component mux21b is
35 port (
36     a,b,s : in std_logic;
37     y : out std_logic
38 );
39 end component;
40 begin
41     c1 : mux21b port map (
42         a => sw(0),
43         b => sw(1),
44         s => btn(0),
45         y => led(0)
46     );
47 end Behavioral;
```

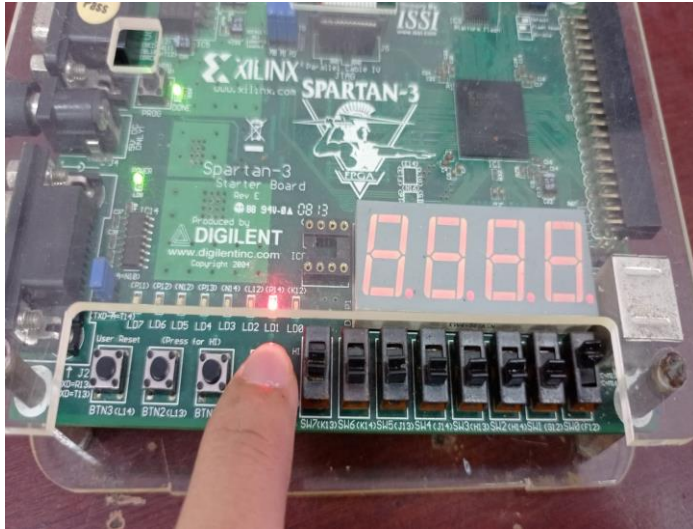
Pin yang digunakan

```
1 NET d loc = F12;
2 NET sw loc = M13;
3
4 NET y(0) loc = K12;
5 NET y(1) loc = P14;
```

Tampilan pada Board



Demux dengan input 1 dan switch 0



Demux dengan input 1 dan switch 0

3. Full Adder 4 Bit

Pin yang digunakan

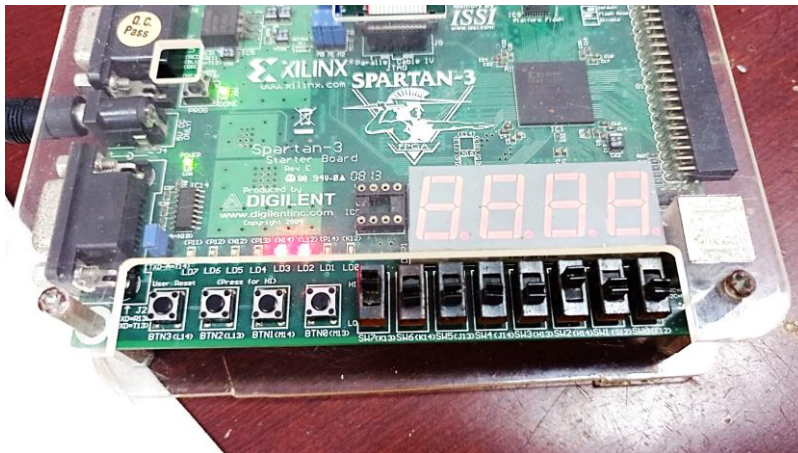
```

1  NET a(0) LOC=F12;
2  NET a(1) LOC=G12;
3  NET a(2) LOC=H14;
4  NET a(3) LOC=H13;
5
6  NET b(0) LOC=J14;
7  NET b(1) LOC=J13;
8  NET b(2) LOC=K14;
9  NET b(3) LOC=K13;
10
11
12 NET sum(0) LOC=K12;
13 NET sum(1) LOC=P14;
14 NET sum(2) LOC=L12;
15 NET sum(3) LOC=N14;
16
17 NET cout LOC=P13;
18 NET cin LOC=M13;

```

Tampilan pada Board





Full Adder dengan input 1000 dan 0100

4. Multiplexer If Statement

Source Code

```

20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 entity mux4bit is
24 port ( I : in std_logic_vector (3 downto 0);
25       sw : in std_logic_vector (1 downto 0);
26       y : out std_logic
27 );
28 end mux4bit;
29
30 architecture Behavioral of mux4bit is
31 begin
32   process (sw,I)
33   begin
34     if sw= "00" then
35       y <= I(0);
36     elsif sw = "01" then
37       y <= I(1);
38     elsif sw= "10" then
39       y <= I(2) ;
40     else
41       y <= I(3);
42     end if;
43   end process;
44 end Behavioral;
45

```

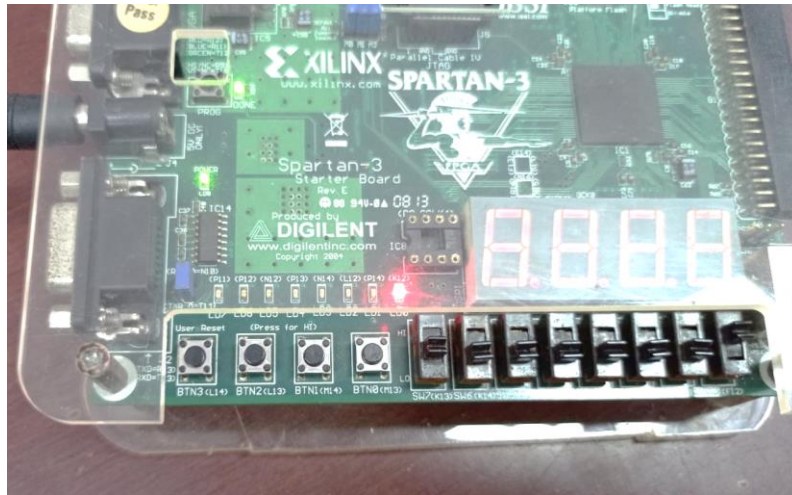
Pin yang difunakan

```

1 NET I (0) LOC=F12;
2 NET I (1) LOC=G12;
3 NET I (2) LOC=H14;
4 NET I (3) LOC=H13;
5
6 NET sw(0) LOC=M13;
7 NET sw(1) LOC=M14;
8
9 NET y LOC=K12;
10

```


Tampilan pada Board



Multiplexer dengan input 0001 dengan switch 00



Multiplexer dengan input 0010 dengan switch 01

5. Demultiplexer If Statement

Source Code

```

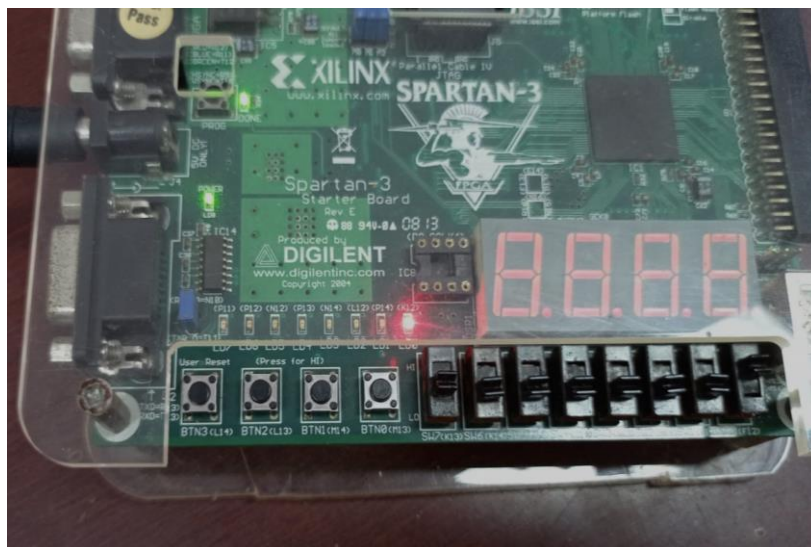
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 entity demux4bit is
24 port ( I : in std_logic;
25       SW : in std_logic_vector (1 downto 0);
26       Y : out std_logic_vector (3 downto 0)
27 );
28
29 end demux4bit;
30
31 architecture Behavioral of demux4bit is
32 begin
33     process (SW,I)
34     begin
35         if SW= "00" then
36             Y <= "000" & I;
37         elsif SW = "01" then
38             Y <= "00" & I & "0";
39         elsif SW= "10" then
40             Y <= "0" & I & "00";
41         else
42             Y <= I & "000";
43         end if;
44     end process;
45
46 end Behavioral;

```

Pin yang digunakan

```
1 NET i(0) LOC=F12;  
2 NET i(1) LOC=G12;  
3 NET i(2) LOC=H14;  
4 NET i(3) LOC=H13;  
5  
6 NET sw(0) LOC=J14;  
7 NET sw(1) LOC=J13;  
8 NET sw(2) LOC=K14;  
9 NET sw(3) LOC=K13;  
10  
11 NET y(0) LOC=K12;  
12 NET y(1) LOC=P14;  
13 NET y(2) LOC=L12;  
14 NET y(3) LOC=N14;  
15
```

Tampilan pada Board



Demux dengan input 1 dan switch 00



Demux dengan input 1 dan switch 10

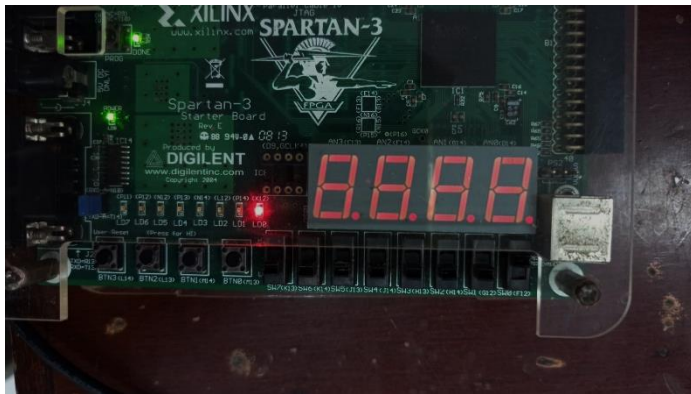
6. Top level Multiplexer dan Demultiplexer

Source Code

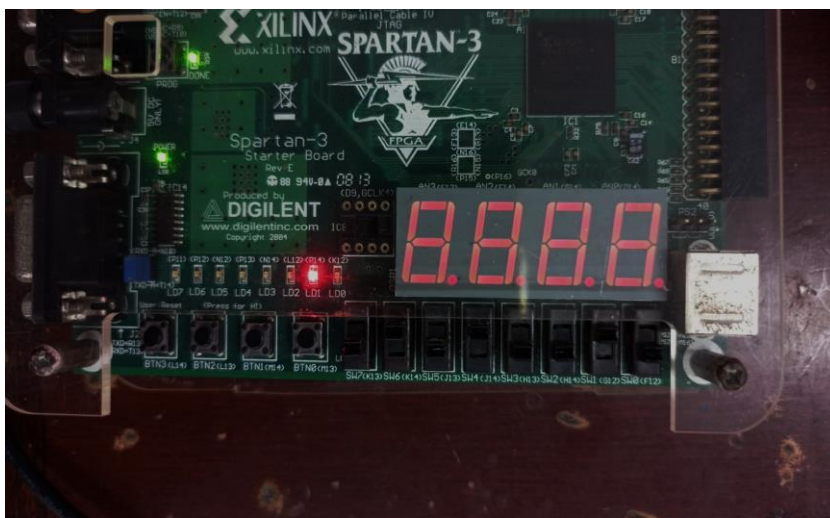
```
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 entity muxDemuxTop is
24 port ( i : in std_logic_vector (3 downto 0);
25       sw : in std_logic_vector (3 downto 0);
26       y : out std_logic_vector (3 downto 0)
27 );
28 end muxDemuxTop;|
29 architecture Behavioral of muxDemuxTop is
30 signal v : std_logic;
31 component mux4bit is
32 port ( I : in std_logic_vector (3 downto 0);
33       sw : in std_logic_vector (1 downto 0);
34       y : out std_logic
35 );
36 end component;
37
38 component demux4bit is
39 port ( I : in std_logic;
40       sw : in std_logic_vector (1 downto 0);
41       y : out std_logic_vector (3 downto 0)
42 );
43 end component;
44 begin
45     mux1 : mux4bit port map( I => i,
46                             sw => sw(1 downto 0),
47                             y => v
48     );
49     demux1 : demux4bit port map( I => v,
50                                 sw => sw(3 downto 2),
51                                 y => y
52     );
53 end Behavioral;
54
```

Pin yang digunakan

```
1 NET i(0) LOC=F12;
2 NET i(1) LOC=G12;
3 NET i(2) LOC=H14;
4 NET i(3) LOC=H13;
5
6 NET sw(0) LOC=J14;
7 NET sw(1) LOC=J13;
8 NET sw(2) LOC=K14;
9 NET sw(3) LOC=K13;
10
11 NET y(0) LOC=K12;
12 NET y(1) LOC=P14;
13 NET y(2) LOC=L12;
14 NET y(3) LOC=N14;
15
```



Mux Demux dengan input 0001 dan switch keduanya 00



Mux Demux dengan input 0010 dan switch keduanya 01

ANALISA DAN KESIMPULAN

No _____

Date _____

Berdasarkan praktikum tersebut program multiplexer 2x1 dengan conditional dan if statement memiliki ~~fungsi~~ output yang berbeda. Multiplexer biasa digunakan utk memilih beberapa sinyal input berdasarkan switchnya.

Pada program Full Adder, yang harus dilakukan membuat komponen dan top level yang disat menjadi file utama. Setiap komponen dipanggil sebanyak 4 kali lalu jalurnya didefinisikan dengan portmap.

Untuk program Multiplexer 4x1 dan demultiplexer 4x1 menggunakan 16 statement yang digunakan utk membuat file Top level. Mux sendiri biasa digunakan utk memilih beberapa sinyal input berdasarkan switch dan output hanya satu. Demultiplexer biasa digunakan utk memilih sinyal output berdasarkan dari switch.

Pada Top Level MuxDemux menggunakan metode Structural Design, dimana merupakan gabungan dari multiplexer 4x1 dan demultiplexer 1x4. Untuk sinyal input berasal dari pin switch F12, G12, H14, H13 dimana J14 dan J13 sebagai multiplexer, dan K14 dan K13 sebagai demultiplexer.