1 D4 - TEKKOM B

DECODER 7 SEGMENT

PART 1

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, 13 Mei 2022

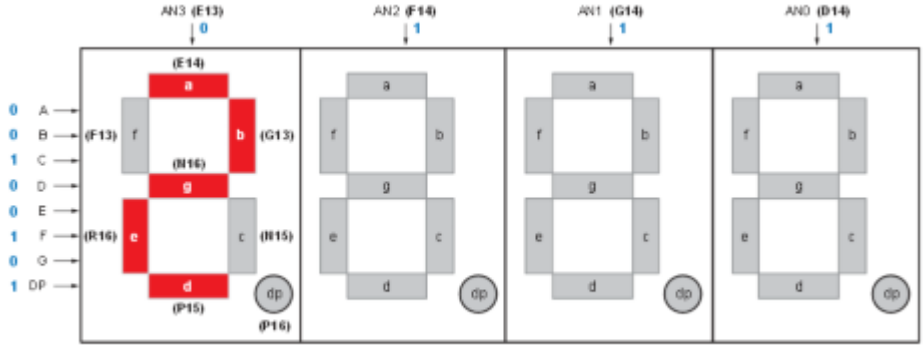
Percobaan 7 – Decoder 7 Segmen

1. Tujuan

- Mampu mengimplementasikan algoritma decoder 7 segmen.
- Dapat menampilkan nilai masukan keluaran pada 7 segmen dalam modul FPGA.

2. Teori

Pada percobaan ini akan mempelajari beberapa rangkaian kombinasional yaitu BCD to 7'segmen. Rangkaian BCD to 7'segmen adalah rangkaian kombinasional yang mengkodekan bilangan biner menjadi bilangan desimal, selanjutnya bilangan tersebut di konfigurasi menjadi tampilan seven segmen. Konfigurasi seven segmen pada board spartan 3 ditunjukan pada gambar 7.1.



Gambar 7. 1 Konfigurasi seven segmen pada board spartan3

Terdapat 4 buah sevensegmen yang disusun secara pararel dengan data 1-g dan dot di rangkai menjadi satu. Sedangkan kendali tampilan di atur melalui pin AN di setiap segmen.

3. Alat dan Bahan

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

4. Langkah Percobaan

A. 7 Segmen Display.

Percobaan ini akan membuat program dengan masukan switch dan keluaran 7segmen.

1. Buatlah new project dengan nama Lab7A.
2. Tambahkan program dibawah ini.

a. 7-segmen Decoder dengan Logic Equation

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity hex7seg is
    Port ( x : in  STD_LOGIC_VECTOR (3 downto 0);
           a_to_g : out  STD_LOGIC_VECTOR (6 downto 0));
end hex7seg;
architecture hex7seg_le of hex7seg is
begin
      a_to_g(6)<=  (not x(3) and not x(2) and not x(1) and x(0))  --a
             or  (not x(3) and x(2) and not x(1) and not x(0))
             or  ( x(3) and x(2) and not x(1) and x(0))
             or  ( x(3) and not x(2) and x(1) and x(0));
      a_to_g(5)<=  (x(2) and x(1) and not x(0))                   --b
             or  ( x(3) and x(1) and x(0))
             or  (not x(3) and x(2) and not x(1) and x(0))
             or  ( x(3) and x(2) and not x(1) and not x(0));
      a_to_g(4)<= (not x(3) and not x(2) and x(1) and not x(0))   --c
             or  (x(3) and x(2) and x(1))
             or  (x(3) and x(2) and not x(0));
      a_to_g(3)<=  (not x(3) and not x(2) and not x(1) and x(0))  --d
             or  (not x(3) and x(2) and not x(1) and not x(0))
             or  (x(3) and not x(2) and x(1) and not x(0))
             or  (x(2) and x(1) and x(0));
      a_to_g(2)<=  (not x(3) and x(0))                            --e
             or  (not x(3) and x(2) and not x(1))
             or  (not x(2) and not x(1) and x(0));
      a_to_g(1)<=  (not x(3) and not x(2) and x(0))               --f
             or  (not x(3) and not x(2) and x(1))
             or  (not x(3) and x(1) and x(0))
             or  (x(3) and x(2) and not x(1) and x(0));
      a_to_g(0)<=  (not x(3) and not x(2) and not x(1))           --g
             or  (x(3) and x(2) and not x(1) and not x(0))
             or  (not x(3) and x(2) and x(1) and x(0));
end hex7seg_le;
```

b. 7-Segmen Decoder Case Statement

```
begin
	process(x)
	begin
		case x is                              --  abcdefg
			when x"0" => a_to_g <= "0000001";--0
			when x"1" => a_to_g <= "1001111";--1
			when x"2" => a_to_g <= "0010010";--2
			when x"3" => a_to_g <= "0000110";--3
			when x"4" => a_to_g <= "1001100";--4
			when x"5" => a_to_g <= "0100100";--5
			when x"6" => a_to_g <= "0100000";--6
			when x"7" => a_to_g <= "0001111";--7
			when x"8" => a_to_g <= "0000000";--8
			when x"9" => a_to_g <= "0000100";--9
			when x"A" => a_to_g <= "0001000";--a
			when x"B" => a_to_g <= "1100000";--b
			when x"C" => a_to_g <= "0110001";--c
			when x"D" => a_to_g <= "1000010";--d
			when x"E" => a_to_g <= "0110000";--e
			when others => a_to_g <= "0111000"; --f
		end case;
	end process;
end hex7segb;
```

Script 7. 2. 7-Segmen Case Statement

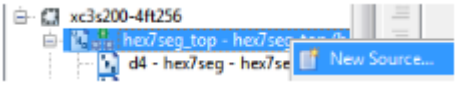
c. 7-Segmen Top Level

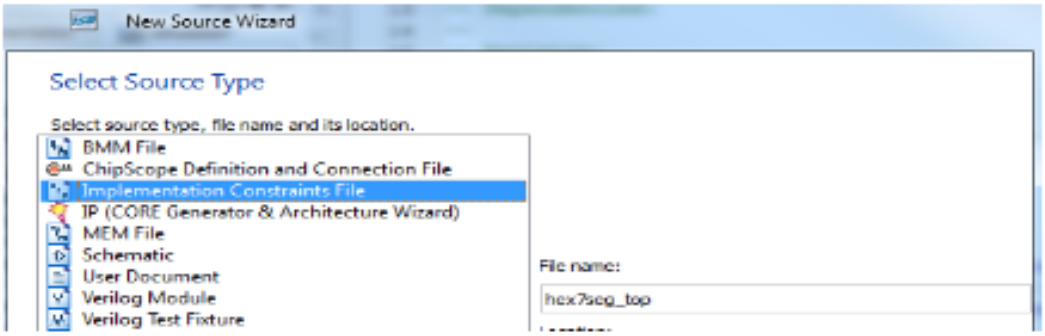
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity hex7seg_top is
    Port ( sw : in  STD_LOGIC_VECTOR (3 downto 0);
           a_to_g : out  STD_LOGIC_VECTOR (6 downto 0);
           an : out  STD_LOGIC_VECTOR (3 downto 0);
           dp : out  STD_LOGIC);
end hex7seg_top;

architecture hex7seg_top of hex7seg_top is
	component hex7segb is
		port(
			x: in STD_LOGIC_VECTOR (3 downto 0);
		   a_to_g : out  STD_LOGIC_VECTOR (6 downto 0)
		);
	end component;
begin
	an <= "0000";		-- all digit on
	dp<= '1';		-- dp off
	d4: hex7seg port map (x=>sw, a_to_g=>a_to_g);
```

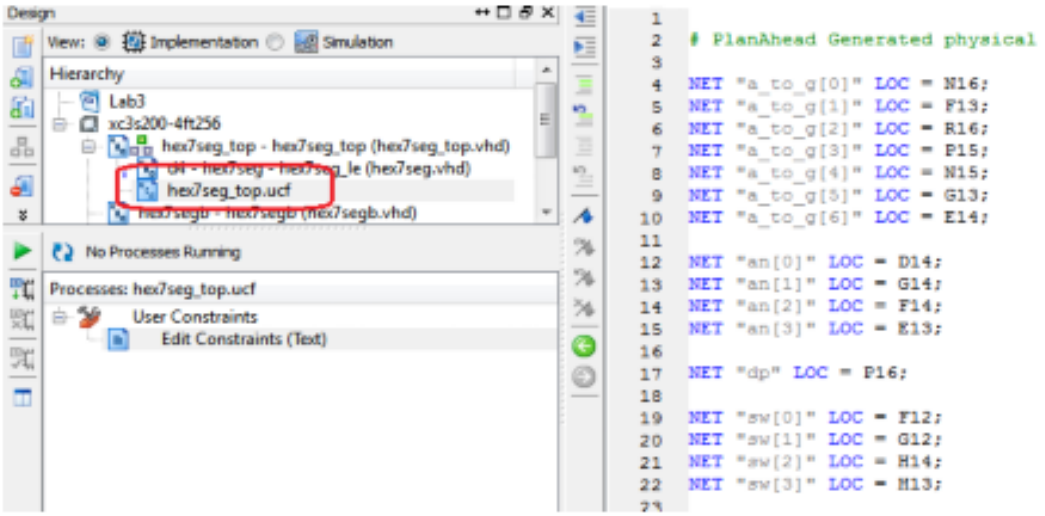
```
end hex7seg_top;
```

Script 7. 3. 7-Segmen Top Level

3. Atur agar program hex7seg_top menjadi Top modul.
4. Lakukan Synthezises-XST, pastikan tidak ada error.
5. Tambahkan konfigurasi pin dengan cara tambahkan new source| Implementation Contraints File. Beri nama hex7seg_top.





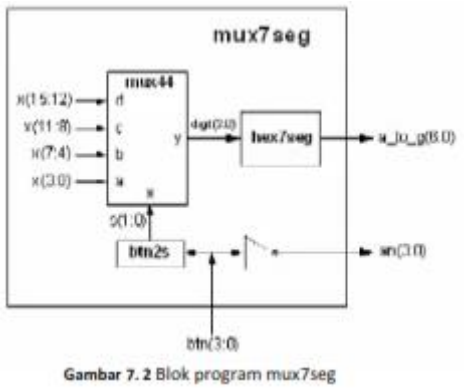


6. Klik Pada jendela hex7seg_top.ucf dan isikan konfigurasi pin seperti gambar diatas.
7. Lakukan SynthezisesXST | Implement Design | Generate programming File sampai dengan programming ke board Spartan-3 FPGA.
8. Atur switch sesuai dengan 1 digit angka terakhir nrp anda.
9. Foto hasil tampilan seven segmen.

B. Multiplexing 7-Segmen

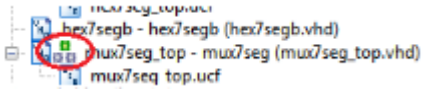
Percobaan ini untuk menampilkan data angka “1234” ke 7segmen dengan pengaturan tampilan dari enable 7segmen melaui push botton.

Blok program ditunjukkan pada gambar 7.2.



Gambar 7. 2 Blok program mux7seg

1. Buatlah new project dengan nama Lab7B.
2. Tambahkan Program baru dengan nama mux7seg_top.
3. Pastikan top modul pada mux7seg_top.vhd.



4. Tulis program mux7seg_top.vhd berikut ini.

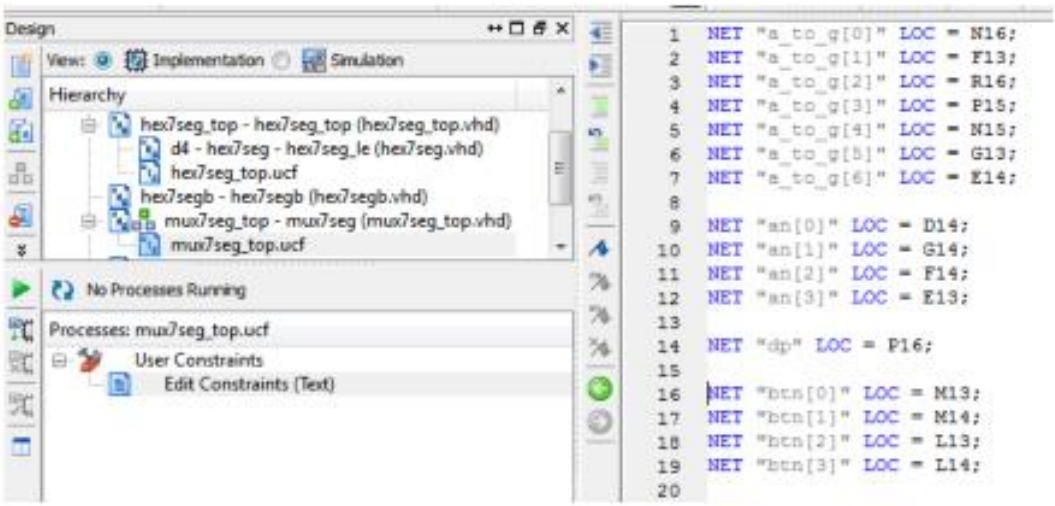
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity mux7seg_top is
    Port ( btn : in  STD_LOGIC_VECTOR (3 downto 0);
           a_to_g : out  STD_LOGIC_VECTOR (6 downto 0);
           an : out  STD_LOGIC_VECTOR (3 downto 0);
           dp : out  STD_LOGIC);
end mux7seg_top;

architecture mux7seg of mux7seg_top is
      signal x: STD_LOGIC_VECTOR (15 downto 0);
      signal s: STD_LOGIC_VECTOR (1 downto 0);
      signal digit: STD_LOGIC_VECTOR (3 downto 0);
```

```
begin
      x <= x"1234";
      an <= not btn;
      s(1) <= btn(2) or btn(3);
      s(0) <= btn(1) or btn(3);
      dp <='1';

-- quad 4 to 1 mux: mux44
      process(s)
      begin
            case s is
                  when "00" => digit <= x(3 downto 0);
                  when "01" => digit <= x(7 downto 4);
```

```
                when "10" => digit <= x(11 downto 8);
                when others => digit <= x(15 downto 12);
            end case;
        end process;

        process(digit)
        begin
            case digit is                -- abcdefg
                when x"0" => a_to_g <= "0000001";  --0
                when x"1" => a_to_g <= "1001111";  --1
                when x"2" => a_to_g <= "0010010"; --2
                when x"3" => a_to_g <= "0000110";  --3
                when x"4" => a_to_g <= "1001100"; --4
                when x"5" => a_to_g <= "0100100";  --5
                when x"6" => a_to_g <= "0100000"; --6
                when x"7" => a_to_g <= "0001111";  --7
                when x"8" => a_to_g <= "0000000";  --8
                when x"9" => a_to_g <= "0000100";  --9
                when x"A" => a_to_g <= "0001000";  --a
                when x"B" => a_to_g <= "1100000"; --b
                when x"C" => a_to_g <= "0110001";  --c
                when x"D" => a_to_g <= "1000010"; --d
                when x"E" => a_to_g <= "0110000";  --e
                when others => a_to_g <= "0111000"; --f
            end case;
        end process;
end mux7seg;
```

Program 4. Multiplexing 7-Segmen

5. Tambahkan file .ucf untuk konfigurasi pin. Seperti dibawah ini.



6. Jalankan programnya dan upload program ke dalam chip FPGA-nya
7. Ubah program agar tampilkan di seven segmen adalah 4 digit terakhir NRP anda.
8. Fotokan hasil tampilannya.

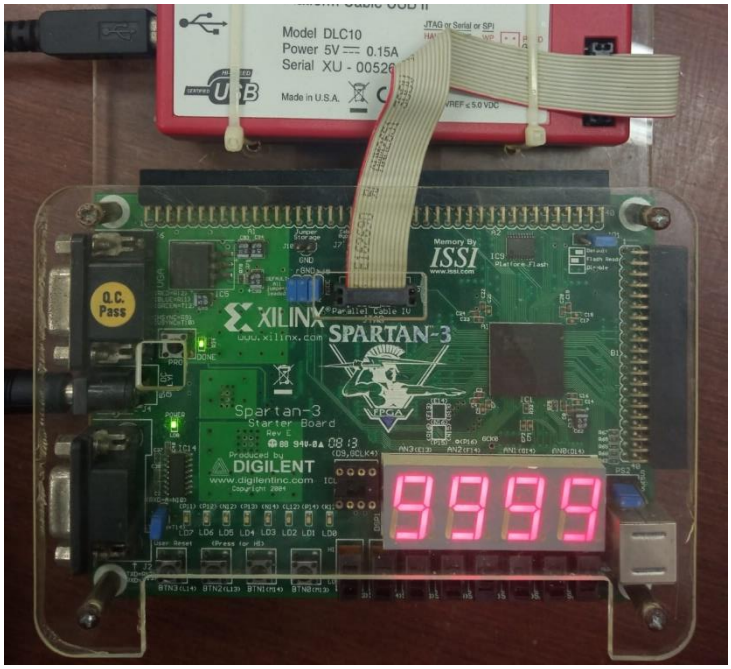
5. Hasil Percobaan

Percobaan A

Definisi Pin

```
1  NET sw(0) LOC = F12;
2  NET sw(1) LOC = G12;
3  NET sw(2) LOC = H14;
4  NET sw(3) LOC = H13;
5
6  NET y(0) LOC = N16;
7  NET y(1) LOC = F13;
8  NET y(2) LOC = R16;
9  NET y(3) LOC = P15;
10 NET y(4) LOC = N15;
11 NET y(5) LOC = G13;
12 NET y(6) LOC = E14;
13
14 NET an(0) LOC = D14;
15 NET an(1) LOC = G14;
16 NET an(2) LOC = F14;
17 NET an(3) LOC = E13;
18
19 NET dp LOC = P16;
```

Tampilan Pada Board



Tampilan 1 digit terakhir NRP 1

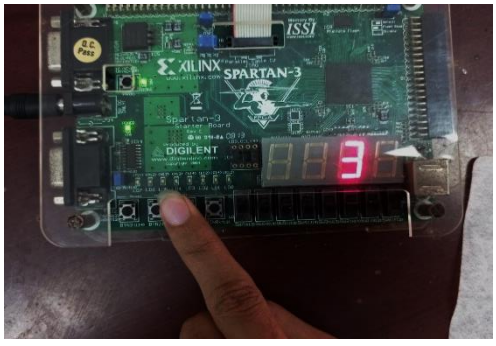
Percobaan B

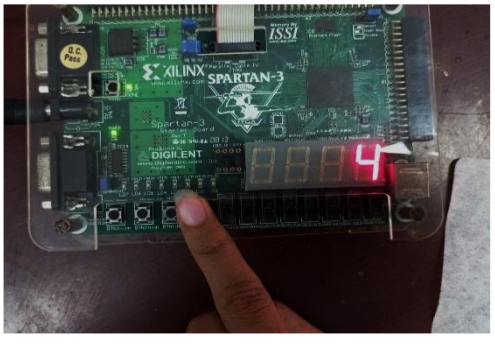
Definisi Pin

```
NET btn(0) LOC = M13;
NET btn(1) LOC = M14;
NET btn(2) LOC = L13;
NET btn(3) LOC = L14;

NET y(0) LOC = N16;
NET y(1) LOC = F13;
NET y(2) LOC = R16;
NET y(3) LOC = P15;
NET y(4) LOC = N15;
NET y(5) LOC = G13;
NET y(6) LOC = E14;

NET an(0) LOC = D14;
NET an(1) LOC = G14;
NET an(2) LOC = F14;
NET an(3) LOC = E13;

NET dp LOC = P16;
```

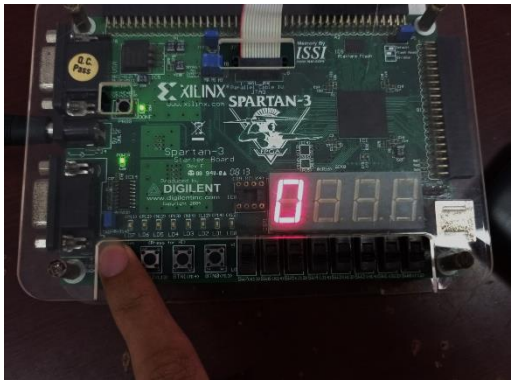
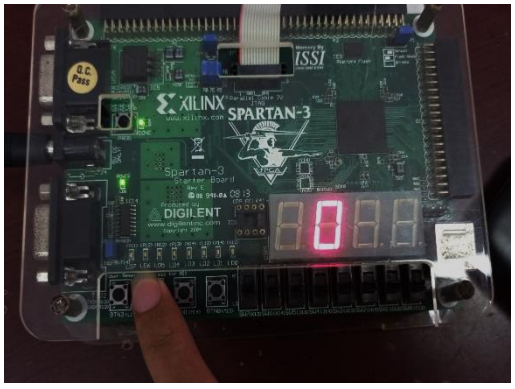
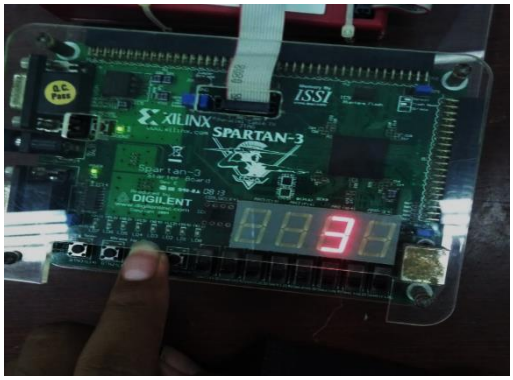
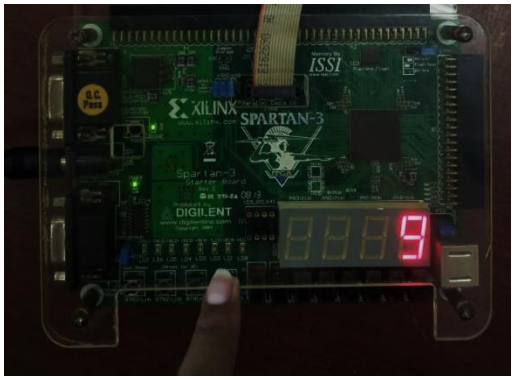
Tampilan pada Board

Output 1234 pada tiap 7 segment





Output 4 digit terakhir dari NRP (0039)

6. Analisa dan Kesimpulan

Date:

Analisa dan kesimpulan

-> Percobaan 1

Pada praktikum tersebut merupakan program rangkaian decoder 7 segment. Rangkaian tersebut biasa digunakan agar inputnya dapat menyalakan display 7 segment. Penggunaan metode Case Statement lebih mudah, seperti when x "0" yang berarti saat x bernilai "0" maka a-to-g akan diberi nilai seperti yang telah diprogram seperti "0000". Untuk Top Level terdapat Entity an yang berguna mengatur digit display dari 7 segment sesuai dengan deklarasi pin.

-> Percobaan 2

Pada praktikum tersebut merupakan program rangkaian Mux 16 to 1 pada 7 seg display. Pada Source Code [x <= x"0039"] berguna sebagai input. Lalu pada bagian [an <= not btn] berguna sebagai output / digit yang akan nyala. Pada -- Quad 4 to 1 mux berguna untuk memilih input, seperti button 1 dan input x = "0039" maka output yang tampil angka 0 pada button 1.