

Perancangan Komponen Terprogram
Tugas Kuliah dan Praktikum PKT Project 1
Rotary Encoder



Disusun oleh:
Nurul Akbar Arlan
(07111940000075)

Dosen:
Rudy Dikairono, ST., MT.

Bidang Studi Elektronika
Departemen Teknik Elektro
Institut Teknologi Sepuluh Nopember
Surabaya
2021

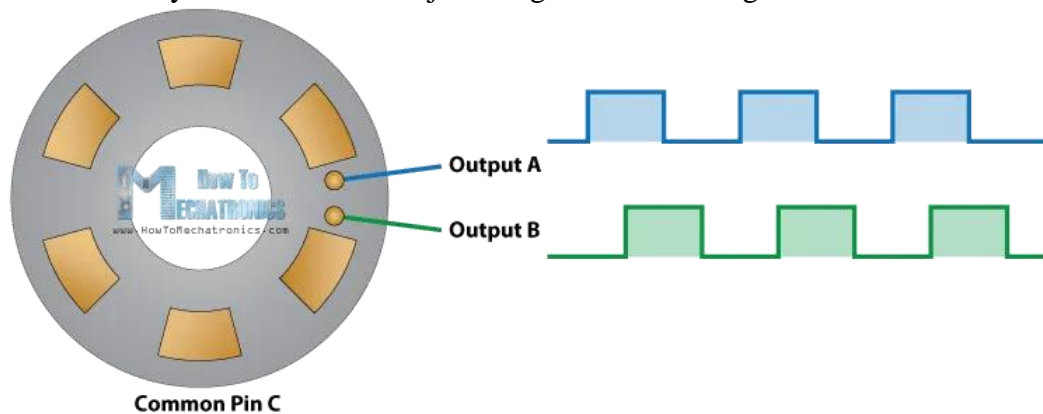
Tugas Kuliah dan Praktikum PKT Project 1

Rotary Encoder

Resume Rotary Encoder

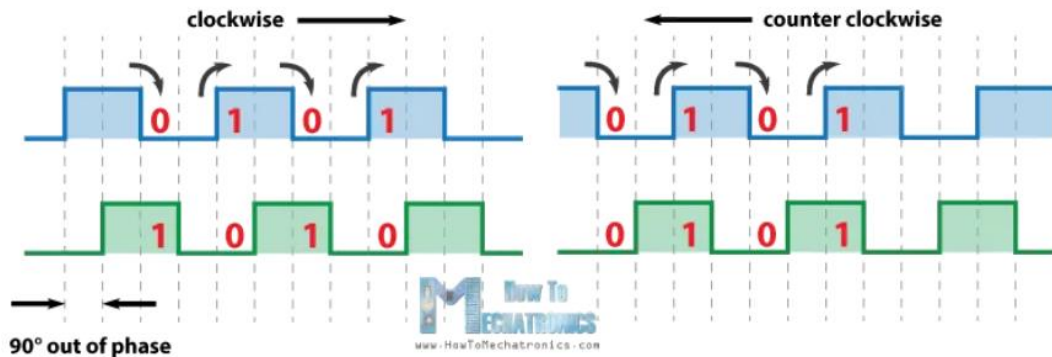
Rotary Encoder adalah sebuah perangkat yang mengubah suatu posisi sudut, tugu, atau gandar menjadi sebuah sinyal elektrik digital atau analog berdasarkan gerak rotasi. Terdapat beberapa jenis rotary encoder berdasarkan sinyal output dan *sensing technology*-nya. Berdasarkan sinyal outputnya rotary encoder dibagi menjadi jenis *Incremental* dan *Absolute*. Berdasarkan *Sensing Technology*-nya dibagi menjadi magnetik, optikal, dan laser.

Rotary Encoder bekerja dengan ilustrasi gambar dibawah ini.,



dimana terdapat sebuah ruang yang terhubung dengan common pin C dan dua kontak pin A dan B. Ketika encodernya berputar, pin A mulai kontak dengan common pin sehingga muncullah sinyal dari kontak antar kedua pin dan common pin.

Dari sinyal output pin A dan pin B, kita dapat menentukan bagaimana arah dari Rotary Encoder. Ketika pin A berubah dan nilai sinyal pin A dan pin B tidak bernilai sama maka encoder berputar searah jarum jam, berdasarkan gambar ilustrasi di atas. Dan ketika pin A berubah dan nilainya bernilai sama dengan pin B maka encoder berputar berlawanan arah jarum jam. Sinyal tersebut dapat diilustrasikan dengan gambar di bawah.



Rangkaian Digital VHDL

Pada rangkaian digital VHDL terdapat bagian inisialisasi port dan sinyal.

```
1  LIBRARY ieee ;
2  USE ieee.std_logic_1164.ALL;
3  USE ieee.numeric_std.ALL;
4  USE IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6  ENTITY rotary_encoder IS
7  |
8  |   PORT      (rot_a, rot_b      : in STD_LOGIC;
9  |              lout2, lout1, lout0 : out STD_LOGIC_VECTOR (6 downto 0));
10 |
11 | END rotary_encoder;
12 |
13 | ARCHITECTURE logic OF rotary_encoder IS
14 | |
15 | |   SIGNAL aprev : STD_LOGIC := rot_a;
16 | |   SIGNAL counter : STD_LOGIC_VECTOR (9 DOWNTO 0) := "0000000000";
17 | |   SIGNAL bcd0 : STD_LOGIC_VECTOR (3 DOWNTO 0) := "0000";
18 | |   SIGNAL bcd1 : STD_LOGIC_VECTOR (3 DOWNTO 0) := "0000";
19 | |   SIGNAL bcd2 : STD_LOGIC_VECTOR (3 DOWNTO 0) := "0000";
20 | |
```

Pada rangkaian ini digunakan lima port, dua port input dan tiga port output. Port input terdiri dari port rot_a dan rot_b, kedua input ini berfungsi sebagai input dari output pin a dan pin b rotary encoder. Port output terdiri dari port lout2, lout1, dan lout0 yang berfungsi sebagai output rangkaian yang menyimpan data seven segment. Lalu digunakan juga lima Signal, aprev untuk menyimpan nilai sinyal rot_a pada iterasi sebelumnya, bcd0, bcd1, dan bcd2 berfungsi untuk menyimpan nilai desimal dari satuan, puluhan, dan ratusan.

```
21 | BEGIN
22 | | PROCESS (rot_a, rot_b)
23 | | | variable Z : STD_LOGIC_VECTOR (21 DOWNTO 0);
24 | | | BEGIN
25 | | |   --memastikan nilai z kembali 0
26 | | |   FOR i in 0 to 21 LOOP
27 | | |     Z(i) := '0';
28 | | |   END LOOP;
29 | | |   --rotary encoder
30 | | |   IF (rot_a /= aprev) THEN
31 | | |     IF (rot_b /= rot_a) THEN
32 | | |       IF (counter < "1111100111") THEN
33 | | |         counter <= counter + 1;
34 | | |       ELSE
35 | | |         counter <= "0000000000";
36 | | |       END IF;
37 | | |     ELSIF (rot_b = rot_a) THEN
38 | | |       IF (counter > "0000000000") THEN
39 | | |         counter <= counter - 1;
40 | | |       ELSE
41 | | |         counter <= "1111100111";
42 | | |       END IF;
43 | | |     END IF;
44 | | |   END IF;
45 | | |   aprev <= rot_a;
```

Selanjutnya adalah proses pertama dari rangkaian VHDL ini, sebelum ke proses pertama pada line 26 sampai 28 terdapat command untuk memastikan nilai Z, yang nantinya digunakan pada proses konversi biner ke bcd, kembali menjadi nol. Proses pertama dari rangkaian VHDL ini merupakan updown counter yang ditrigger oleh arah Gerakan counter tersebut, counter akan bekerja sebagai up counter ketika rotary encoder berputar searah jarum jam, sebaliknya ketika rotary encoder berputar berlawanan arah jarum jam maka counter akan bekerja sebagai down counter.

```

46      --convert ke bcd
47      Z(12 DOWNT0 3) := counter;
48      FOR i IN 0 TO 6 LOOP
49          IF Z(13 DOWNT0 10) > "0100" THEN
50              Z(13 DOWNT0 10) := Z(13 DOWNT0 10) + "0011";
51          END IF;
52          IF Z(17 DOWNT0 14) > "0100" THEN
53              Z(17 DOWNT0 14) := Z(17 DOWNT0 14) + "0011";
54          END IF;
55          Z(21 DOWNT0 1) := Z(20 DOWNT0 0);
56      END LOOP;
57
58      bcd0 <= Z(13 DOWNT0 10);
59      bcd1 <= Z(17 DOWNT0 14);
60      bcd2 <= Z(21 DOWNT0 18);
61  END PROCESS;
62

```

Proses kedua adalah pengonversian nilai biner dari counter menjadi bcd(binary coded decimal) karena dibutuhkan oleh seven segment. Proses ini bekerja dengan cara angka biner digeser(shift right) dan dicuplik empat angka-empat angka, bila nilai dari keempat angka biner itu lebih besar dari 4 maka ditambahkan dengan 3, dan seterusnya hingga semua angka biner tercuplik.

```

63      --seven segment
64      lout0 <= "0000001" when bcd0 = "0000" else
65              "1001111" when bcd0 = "0001" else
66              "0010010" when bcd0 = "0010" else
67              "0000110" when bcd0 = "0011" else
68              "1001100" when bcd0 = "0100" else
69              "0100100" when bcd0 = "0101" else
70              "0100000" when bcd0 = "0110" else
71              "0001111" when bcd0 = "0111" else
72              "0000000" when bcd0 = "1000" else
73              "0000100" when bcd0 = "1001" else
74              "0110000";
75
76      lout1 <= "0000001" when bcd1 = "0000" else
77              "1001111" when bcd1 = "0001" else
78              "0010010" when bcd1 = "0010" else
79              "0000110" when bcd1 = "0011" else
80              "1001100" when bcd1 = "0100" else
81              "0100100" when bcd1 = "0101" else
82              "0100000" when bcd1 = "0110" else
83              "0001111" when bcd1 = "0111" else
84              "0000000" when bcd1 = "1000" else
85              "0000100" when bcd1 = "1001" else
86              "0110000";
87
88      lout2 <= "0000001" when bcd2 = "0000" else
89              "1001111" when bcd2 = "0001" else
90              "0010010" when bcd2 = "0010" else
91              "0000110" when bcd2 = "0011" else
92              "1001100" when bcd2 = "0100" else
93              "0100100" when bcd2 = "0101" else
94              "0100000" when bcd2 = "0110" else
95              "0001111" when bcd2 = "0111" else
96              "0000000" when bcd2 = "1000" else
97              "0000100" when bcd2 = "1001" else
98              "0110000";
99  END logic;

```

Dan proses terakhir adalah mengubah nilai bcd kedalam seven segment.

VHDL Testbench

Pada praktikum pertama ini testbench yang terdapat dua testbench yang digunakan, pertama testbench ketika encoder berputar searah jarum jam.

```
1 add wave rot_a rot_b lout2 lout1 lout0
2 force rot_a 0 0, 1 5ns, 0 15ns -repeat 20ns
3 force rot_b 0 0, 1 10ns -repeat 20ns
4 run 1020ns
```

rot_a diberi sinyal 0-1-1-0 dengan periode 5ns setiap nilai dan rot_b diberi sinyal 0-0-1-1 dengan periode 5ns setiap nilai, diulang terus menerus hingga 1020ns.

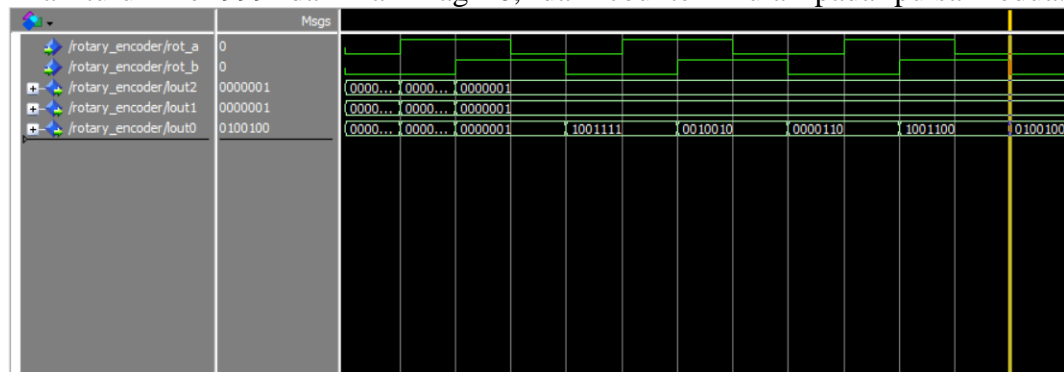
Dan tesbench kedua ketika encoder berputar berlawanan arah jarum jam.

```
1 add wave rot_a rot_b lout2 lout1 lout0
2 force rot_a 1 1, 0 5ns, 1 15ns -repeat 20ns
3 force rot_b 0 0, 1 10ns -repeat 20ns
4 run 1030ns
```

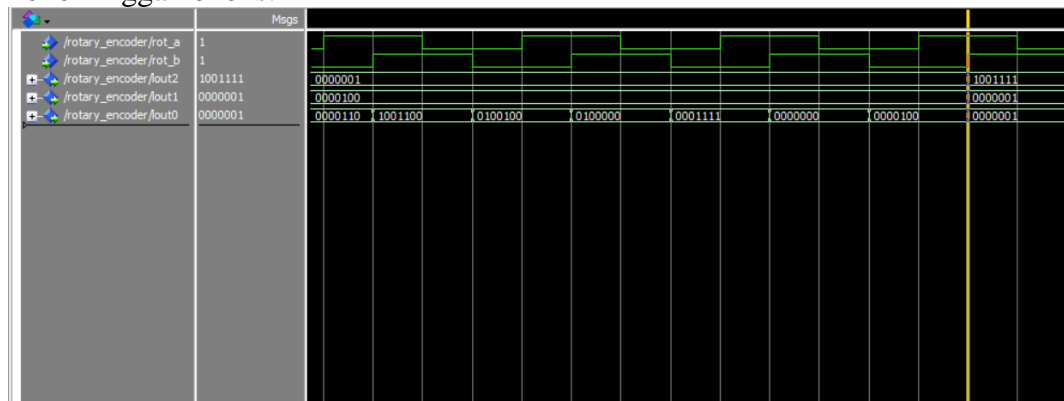
rot_a diberi sinyal 1-0-0-1 dengan periode 5ns setiap nilai dan rot_b diberi sinyal 0-0-1-1 dengan periode 5ns setiap nilai, diulang terus menerus hingga 1030ns.

Hasil Pengujian

Pada testbench pertama, dapat dilihat nilai output seven segment bertambah 2 setiap 1 periode pulsa. Dapat dilihat terdapat error pada pulsa pertama dimana nilai turun ke 999 dan naik lagi 0, dan counter mulai pada pulsa kedua.



dan testbench ini bekerja hingga nilai output seven segment bernilai 100(lout 2 bernilai 1001111, lout1 bernilai 0000001, dan lout0 bernilai 0000001) pada waktu 1010 hingga 1020ns.



Pada testbench kedua, dapat dilihat nilai output seven segment berkurang 2 setiap 1 periode pulsa. Seperti testbench pertama terdapat error pada pulsa pertama.

